Performance Analysis of the FDDI Token Ring

Randall A. Simonson

Computer Science Report No. TR-88-02
February 1, 1988

This work was sponsored by the Flight Data Systems Branch of NASA--Johnson Space Center, Houston, Texas.

Performance Analysis of the FDDI Token Ring

A Master of Science Thesis by

Randall A. Simonson
Computer Networks Laboratory
Department of Computer Science
Thornton Hall
University of Virginia
Charlottesville, Virginia 22903

Alfred C. Weaver
Thesis Advisor

February 1988

Abstract

The Fiber Distributed Data Interface (FDDI) is an all fiber optic token ring local area network. It is intended for LAN applications requiring high data rates, low latency, high reliability, and freedom from electrical noise. This thesis presents the details of the FDDI architecture and protocol and studies, via simulation, its performance under various network configurations. Measurements of interest include token cycle time, mean packet delivery delay, total throughput, and queue lengths. Performance data are presented for varying packet sizes, numbers of ring stations, ring circumference, and offered loads. The performance of the token ring when handling multiple priority traffic is also studied, and it is shown that implementing FDDI's various priority schemes consistently results in a degradation of overall network performance.

Table of Contents

ABSTRACT i
ACKNOWLEDGEMENTS ii
TABLE OF CONTENTS iii
LIST OF FIGURES vii

1 Local Area Networks 1

1.1 Introduction 1
1.2 IEEE 802.3 Contention Bus 1
1.3 IEEE 802.4 Token Bus 2
1.4 IEEE 802.5 Token Ring 3
1.5 High Speed Local Area Networks 4

1.5.1 SAE AS4074.2 High Speed Ring Bus 5
1.5.2 Fiber Distributed Data Interface 6

2 Fiber Distributed Data Interface Protocol 7

2.1 Introduction 7
2.2 Terms 9
2.3 General Description 9
2.4 Physical Layer Medium Dependent Requirements (PMD) 11

2.4.1 PMD Architecture 11
2.4.2 PMD Services 14

2.4.2.1 PMD to PHY Services 14
2.4.2.2 PMD to SMT Services 15

2.5 Physical Layer Protocol (PHY) 16

2.5.1 PHY Facilities 16

2.5.1.1 Coding and the Symbol Set 16
2.5.1.2 Line States 19

2.5.2 PHY Services 20

2.5.2.1 PHY to MAC Services 20
2.5.2.2 PMD to PHY Services 21
2.5.2.3 PHY to SMT Services 21

2.6 Medium Access Control (MAC) 22

2.6.1 MAC Facilities 22

2.6.1.1 Protocol Data Units 22

2.6.1.2 PDU Fields 22

2.6.1.2.1 Preamble (PA) 22
2.6.1.2.2 Starting Delimiter (SD) 23
2.6.1.2.3 Frame Control (FC) 24
2.6.1.2.4 Destination Address (DA) and Source Address (SA) 25
2.6.1.2.5 Information (INFO) Field 25
2.6.1.2.6 Frame Check Sequence (FCS) 26
2.6.1.2.7 Ending Delimiter (ED) and Frame Status (FS) 26

2.6.1.3 Timers 27

2.6.1.3.1 Token-Holding Timer (THT) 27
2.6.1.3.2 Valid-Transmission Timer (TVX) 27
2.6.1.3.3 Token-Rotation Timer (TRT) 27

2.6.1.4 Frame Counters 27

2.6.2 MAC Operations 28

2.6.2.1 Frame and Token Transmission 28
2.6.2.2 Frame Stripping 28
2.6.2.3 Ring Scheduling 28

2.6.2.3.1 Synchronous Traffic 29
2.6.2.3.2 Asynchronous Transmission 30

2.6.2.4 Ring Monitoring 31

2.6.2.4.1 Claim Token Process 32
2.6.2.4.2 Initialization Process 32
2.6.2.4.3 Beacon Process 33

2.6.3 MAC Services 33

2.6.3.1 MAC to LLC Services 33
2.6.3.2 PHY to MAC Services 35
2.6.3.3 MAC to SMT Services 35

2.7 Station Management (SMT) 36

2.7.1 SMT Services 37

2.7.1.1 MAC to SMT Services 37
2.7.1.2 PHY to SMT Services 37
2.7.1.3 PMD to SMT Services 37

3 The Simulation Environment 38

3.1 Introduction 38
3.2 The FDDI Simulation Process 38

3.2.1 Information Input Phase 38

3.2.1.1 General Network Information 39
3.2.1.2 FDDI Specific Information 40
3.2.1.3 Station Class Information 41

3.2.2 Computation Phase 41
3.2.3 Output Phase 43

3.2.3.1 Network Wide Data 44
3.2.3.2 Priority Data 45

3.3 Justification of Simulation Results 46

3.3.1 Steady State Estimation 46
3.3.2 Confidence Interval Estimation 49
3.3.3 Comparison of Simulation and Analytic Model 51

4 Performance of the FDDI Token Ring 57

4.1 Introduction 57
4.2 Single Priority Operation 57

4.2.1 Reference Configuration 58

4.2.1.1 Token Cycle Time 58
4.2.1.2 Token and Packet Traffic 60
4.2.1.3 Arrival Queue Length 60
4.2.1.4 Throughput 64
4.2.1.5 Packet Delay 64

4.2.2 Effects of Varying Packet Length 64

4.2.2.1 Token Cycle Time 67
4.2.2.2 Token Traffic 67
4.2.2.3 Arrival Queue Length 70
4.2.2.4 Packet Delay 70

4.2.3 Effects of Varying the Number of Stations 70

4.2.3.1 Token Cycle Time 70
4.2.3.2 Throughput 71
4.2.3.3 Arrival Queue Length 71
4.2.3.4 Packet Delay 77

4.2.4 Effects of Varying Ring Circumference 77

4.2.4.1 Token Cycle Time and Throughput 77
4.2.4.2 Packet Delay 78

4.3 Multiple Priority Operation 78

4.3.1 Reference Configuration 78

4.3.1.1 Token Cycle Time and Throughput 83
4.3.1.2 Packet Delay 83

4.3.2 Effects of Varying Asynchronous Priority Thresholds 87

4.3.2.1 Packet Service Delay 87
4.3.2.2 Token Cycle Time 90
4.3.2.3 Throughput 90

4.3.3 Effects of Lowering Target Token Rotation Time 90

4.3.3.1 Token Cycle Time 93
4.3.3.2 Throughput 93
4.3.3.3 Packet Delay 94

4.3.4 Effects of Restricted Token Dialog 98

4.3.4.1 Packet Delay 98
4.3.4.2 Throughput 98

5 Conclusions 101

5.1 FDDI Performance 101

5.1.1 Single Priority Operation 101

5.1.1.1 Varying Mean Packet Length 101
5.1.1.2 Varying the Number of Stations 102
5.1.1.3 Varying Ring Circumference 102

5.1.2 Multiple Priority Operation 102

5.1.2.1 Varying Priority Thresholds 103
5.1.2.2 Lowering Target Token Rotation Time 103
5.1.2.3 Restricted Token Dialog 103

5.2 Suggestions For Future Research 104

Bibliography 105

List of Figures

Figure 1.1	— Dual Redundant Counter Rotating Ring	6
Figure 2.1	— Relationship of FDDI Layers	8
Figure 2.2	— Physical Connection	12
Figure 2.3	— FDDI Topology Example	13
Figure 2.4	— PMD Services	16
Figure 2.5	— PHY Symbol Coding Table	18
Figure 2.6	— Protocol Data Unit Formats	23
Figure 3.1	— General Network Information Input Screen	39
Figure 3.2	— FDDI Specific Input Screen	40
Figure 3.3	— Class Information Input Screen	42
Figure 3.4	— Output Screen Menu	43
Figure 3.5	— General Information Output Screen	44
Figure 3.6	— Priority Information Output Screen	45
Figure 3.7	— Token Cycle Time Stability	47
Figure 3.8	— Queueing Delay Stability	48
Figure 3.9	— Plots of $t_n(1 - \alpha/2)$ for $1 - \alpha = 0.9$ and $1 - \alpha = 0.95$	51
Figure 3.10	— Token Cycle Time, Simulation vs. Analytic Model	54
Figure 3.11	— Token Arrival Queue Length, Simulation vs. Analytic Model	55
Figure 3.12	— Mean Packet Service Delay, Simulation vs. Analytic Model	56
Figure 4.1	— Token Cycle Time, Single Priority Reference	59
Figure 4.2	— Tokens Received and Tokens Accessed, Single Priority Reference	61
Figure 4.3	— Tokens Accessed and Packets Sent, Single Priority Reference	62
Figure 4.4	— Token Arrival Queue Length, Single Priority Reference	63
Figure 4.5	— Throughput, Single Priority Reference	65

Figure 4.6	— Various Packet Delays, Single Priority Reference	66
Figure 4.7	— Token Cycle Time Varying Packet Lengths	68
Figure 4.8	— Tokens Accessed Varying Packet Length	69
Figure 4.9	— Arrival Queue Length Varying Packet Length	72
Figure 4.10	— Network Access Delay Varying Packet Length	73
Figure 4.11	— Token Cycle Time Varying the Number of Stations	74
Figure 4.12	— Throughput Varying the Number of Stations	75
Figure 4.13	— Arrival Queue Lengths Varying the Number of Stations	76
Figure 4.14	— Service Delay Varying the Number of Stations	79
Figure 4.15	— Token Cycle Time Varying Ring Circumference	80
Figure 4.16	— Throughput Varying Ring Circumference	81
Figure 4.17	— Service Delay Varying Ring Circumference	82
Figure 4.18	— Token Cycle Time, Multiple Priority Reference	84
Figure 4.19	— Throughput, Multiple Priority Reference	85
Figure 4.20	— Service Delay, Multiple Priority Reference	86
Figure 4.21	— Service Delay Varying Priority Thresholds From 3.0 ms to 6.5 ms	88
Figure 4.22	— Service Delay Varying Priority Thresholds From 1.0 ms to 4.5 ms	89
Figure 4.23	— Token Cycle Time Varying Priority Thresholds	91
Figure 4.24	— Throughput Varying Priority Thresholds	92
Figure 4.25	— Token Cycle Time Varying Target Token Rotation Time	95
Figure 4.26	— Throughput Varying Target Token Rotation Time	96
Figure 4.27	— Service Delay, 1 ms Target Token Rotation Time	97
Figure 4.28	— Service Delay, Eligible Station vs. Ineligible Station	99
Figure 4.29	— Throughput Varying Restricted Token Dialog Usage	100

Chapter 1

Local Area Networks

1.1. Introduction

Historically, the goal of most computer network research and development has been to improve the means by which data is transmitted. This may seem a fairly straight forward directive, but the long list of network protocols that have failed to meet this goal completely indicates otherwise. One problem is that it is difficult to agree on what constitutes an improvement. To some, faster networks are always better; to others, network reliability is the main issue; and to still others, cost is the primary factor. A common problem network researchers face is that one step forward in one area may cause two steps backwards in another area.

In the past decade, many new network protocols have been introduced, but the "ultimate" or optimal data transmission method has yet to be discovered. Because of the large number of differing network protocols, it was seen early on that industry standards would have to be adopted if any degree of compatibility was to be realized.

One of the first attempts at standardization was the IEEE 802 committee on local area networks. Their goal was to establish a family of local area network standards dealing specifically with internetworking, network management and ISO OSI layers 1 and 2 (physical layer and data link layer). A chronology of the standards adopted by the 802 committee provides a good basis for studying the progression of local area network protocols in the past few years.

1.2. IEEE 802.3 Contention Bus

The 802.3 contention bus [IEEE85a] was the first media access protocol adopted by the 802 committee. 802.3 was basically an adaptation of the Ethernet protocol [METC76]. Although they are not identical, the terms "Ethernet" and "802.3 contention bus" are used interchangeably herein.

The 802.3 protocol specifies a bus topology using a coaxial transmission medium with a 10 Mbps transmission speed. Up to 1000 devices may be attached to the bus, provided that no two devices have a maximum end-to-end separation exceeding 2500 meters.

802.3 utilizes a Carrier Sense Multiple Access with Collision Detect (CSMA/CD) media access method. CSMA/CD specifies that a station may transmit a packet whenever there is no traffic on the bus. A station with a packet to transmit monitors the bus (i.e., carrier sense) and begins transmission when the bus is idle.

Multiple stations may simultaneously attempt packet transmission. When this occurs, the transmitting stations detect the collision and each transmits a jam packet to inform all other stations that a collision has occurred. Since immediate retransmissions of collided packets would result in another collision, each station schedules retransmission according to a binary exponential backoff mechanism. No other packets may be transmitted by stations involved in a collision until the collided packets have been successfully transmitted or until the retransmission process has failed.

In light of the above arguments, 802.3 is not applicable to real-time environments requiring guaranteed message delivery and bounded delivery times. 802.3 was designed for office automation where the bursty nature of the traffic results in few collisions.

1.3. IEEE 802.4 Token Bus

The next step in the progression of local area network protocols was the 802.4 token bus [IEEE85b]. 802.4 was designed for an entirely different environment than the contention bus. Requirements for 802.4 were:

1. allow optional broadband signalling to accommodate both voice and video in addition to data
2. have the capacity to distinguish various priorities of data
3. allow network tuning to guarantee the delivery time of high priority data.

802.4 specifies a token-passing bus topology using a coaxial cable media with data signalling speeds of 1 Mbps, 5 Mbps, and 10 Mbps.

802.4's token passing scheme utilizes a unique sequence of bits known as a token. The token is passed from station to station conferring the right of packet transmission. Stations may transmit packets only when they have the token. Each station on the bus has an address which determines its place in a logical ring. Tokens are passed in order of descending station addresses, with the lowest addressed station passing the token to the highest addressed station.

When a station receives the token and it has packets waiting to be sent, it begins transmission of those packets. Collision detection is not necessary because only one station is transmitting packets at any one time. 802.4 specifies four different message priorities; these are (in order from highest to lowest): Synchronous, Urgent_Asynchronous, Normal_Asynchronous, and Time_Available. Each priority has a timer to control the amount of service it receives relative to the other priorities.

A station may continue packet transmission until it has no more packets to send or until the amount of time the station may keep the token has expired. When a station concludes packet transmission, it addresses the token to its logical successor and transmits it.

1.4. IEEE 802.5 Token Ring

The next step in the progression of local area network protocols brought about a change in the choice of network topology. Since stations in the 802.4 protocol were already in a logical ring, the next step was to organize the stations so that each station's logical successor was also its physical successor. The result was the 802.5 token ring [IEEE85c].

802.5 specifies a token passing ring topology capable of using a variety of transmission media with transmission speeds of 1 Mbps and 4 Mbps. One of the advantages of a ring over a bus is that the actual wiring of a ring is a series of point-to-point connections. This permits use of mixed media such as twisted pair and fiber optics. Another advantage to a ring is that each station's logical successor is also its physical successor. This simplifies the token passing scheme and eliminates the need for an address field in the token. When the ring is heavily loaded, the shorter token allows 802.5 to perform more efficiently than a token bus. One disadvantage to a ring is that a single break in the physical medium causes the entire network to cease operation.

802.5 specifies eight priorities of service governed by a prioritized token scheme. As the token is passed from station to station, its priority may be changed. A token may only be captured for transmission of a priority *n* packet if *n* is greater than or equal to the current priority of the token. Stations with low priority traffic may reserve the token for use after higher priority packets have been serviced.

When a station captures a usable token, it modifies the token to become a *Start_of_Frame* sequence. The capturing station then appends to the modified token, among other things, the address of the destination station(s), the data to be sent, and an *End_of_Frame* sequence. A station continues to send packets until it has no more packets of high enough priority to send, or until it exceeds its token holding time limit. When a station concludes its packet transmission, it generates a new token so that other stations may gain the right of transmission.

On a properly operating ring, bits make one full circumference before being removed by the transmitting station. At any given time the only station transmitting is the one with the token; all other stations are simply repeating the bits they receive. In addition to repeating bits, the station(s) for which the packet was intended copy the bits they receive.

1.5. High Speed Local Area Networks

As local area network protocols continue to evolve it is becoming all too apparent that faster, more reliable performance is not only desirable, it will soon be necessary. The applications of local area networks in business and industry have increased dramatically in the past decade and all indications are that this trend will continue.

A way to improve local area network performance can be found in the type of transmission medium used. One transmission medium that until recently has been far too expensive and difficult to work with is fiber optics. Fiber optic technology utilizes light energy instead of electrical energy. Information is transmitted as pulses of highly focused light emitted from a light source (transmitter), either a light-emitting diode (LED) or injection-laser diode. Hair-thin glass fibers (approximately 0.006 inches in diameter) enclosed in reflective cable, carry light energy to a light detector (receiver), which converts the light pulses to electrical pulses.

In the past few years fiber optic technology has improved so dramatically that it can now be considered a viable alternative to twisted pair and coaxial cable. Fiber optic cable provides the following advantages over conventional cables:

* broader bandwidth (optical cable can carry more than 2 Gbps over ten km.)
* lower transmission losses (typical error rate is one bit in ten billion)
* longer maximum distance between repeaters
* smaller physical size (easy to install)
* lower weight (one km. of copper wire weighs 200 pounds, fiber weighs 30 pounds)
* better electromagnetic immunity (lightwaves do not conduct electricity)
* able to withstand temperatures up to 1000 degrees C.
* better data security (optical cable is difficult to tap unless physically broken)

High speed local area network protocols are emerging to which fiber optic media is well suited. Among the new fiber optic networks, two are quickly becoming standards in the field; they are the SAE AS4074.2 High Speed Ring Bus and the Fiber Distributed Data Interface (FDDI).

1.5.1. SAE AS4074.2 High Speed Ring Bus

The SAE AS4074.2 high speed ring bus [SAE87] is a token-passing ring network intended primarily for real-time transmission of short messages (relative to ring circumference). It specifies a maximum ring circumference of 3 km, to which a maximum of 512 stations may be attached. The SAE standard does not specify a data rate or transmission medium, but suggests implementations using a 40 Mbps coaxial network or 80 Mbps fiber optic network.

SAE was designed with military applications in mind. Hence, it has addressed the problem of breaks in the transmission medium which would render the network useless by introducing a dual redundant counter rotating ring. Figure 1.1 depicts the functionality of the redundant counter rotating ring. In this figure, network A has no breaks in its physical medium; network B depicts the same network with a break between stations 1 and 2. The dual counter rotating ring allows network B to recover completely by simply reconfiguring the ring to utilize the redundant ring. This additional robustness is a major improvement over that of the 802.5 protocol.

Like 802.5, SAE utilizes a prioritized token-passing scheme and eight service priorities. When a station captures a usable token (also known as a *Free* token) it changes the status of the token from *Free* to *Claimed* and appends fields to the modified token in much the same way as 802.5. However, unlike 802.5, only one message may be sent per token access. Because SAE expects primarily short messages, it has the capacity to deal with them very efficiently. Up to 15 short messages may be appended to a *Claimed* token as it circulates around the ring.

1.5.2. Fiber Distributed Data Interface

FDDI is a high speed, general purpose, local area network specifying fiber optics as its only transmission medium. The purpose of this thesis is to examine the FDDI token ring. Chapter 2 provides a description of the FDDI protocol. Chapter 3 discusses the simulation environment developed to analyze the performance of FDDI. Chapter 4 presents the performance of several different FDDI configurations and Chapter 5 discusses the conclusions drawn and the possibilities for further research.

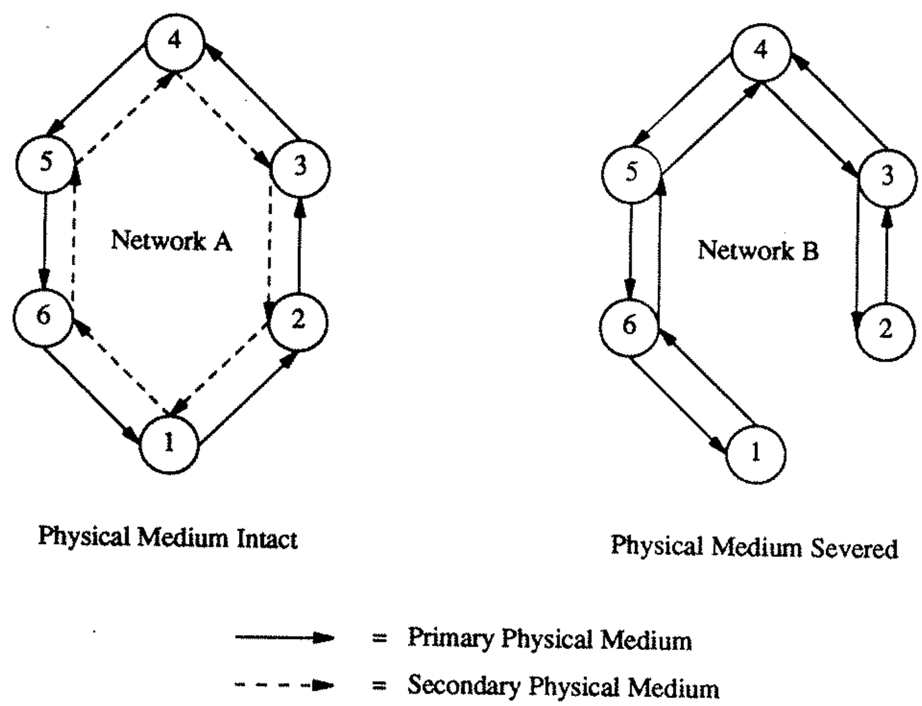


Figure 1.1 — Dual Redundant Counter Rotating Ring

Chapter 2

Fiber Distributed Data Interface Protocol

2.1. Introduction

FDDI is a high bandwidth, general purpose, local area network that specifies fiber optics as its only transmission medium. FDDI signals at a data rate of 125 Mbps, but, due to its 4-into-5 bit encoding scheme, it renders a usable data rate of 100 Mbps. Its architecture specifies a dual (counter rotating) token ring capable of handling up to 1000 stations distributed over a fiber optic path of up to 200 km [WEAV87].

The FDDI standard is divided into four sections:

1. Physical Layer Medium Dependent Requirements (PMD) [FDDI86a]
2. Physical Layer Protocol (PHY) [FDDI86b]
3. Medium Access Control Protocol (MAC) [FDDI86c]
4. Station Management Services (SMT) [FDDI86d]

PMD specifies how a station should communicate with other stations on the ring. PMD translates the encoded electrical data signals to and from optical signals and defines and characterizes the following:

* fiber optic drivers and receivers
* medium dependent code requirements
* cables
* connectors
* power budgets
* optical bypass provisions
* physical hardware related characteristics

PHY provides connection between PMD and the Data Link Layer (DLL). PHY decodes the incoming signal and encodes the outgoing bit stream into grouped transmission codes using a 4-into-5 bit encoding scheme. PHY is also responsible for the synchronization of the incoming and outgoing code-bit clocks and the identification of octet boundaries on the incoming and outgoing bit streams.

MAC provides fair and deterministic access to the medium for all stations. It is responsible for recognition of station addresses and for verification that a message was successfully received by the station(s) to which it was sent. Its primary functions are transmission, repetition, removal and verification of data frames.

SMT works logically beside PMD, PHY, and MAC. Its purpose is to coordinate the activities of these layers so that each station may work cooperatively with other stations on the ring. SMT provides:

* control of station initialization
* configuration management
* fault isolation
* recovery

Figure 2.1 depicts the logical relationship between PMD, PHY, MAC, LLC, and SMT.

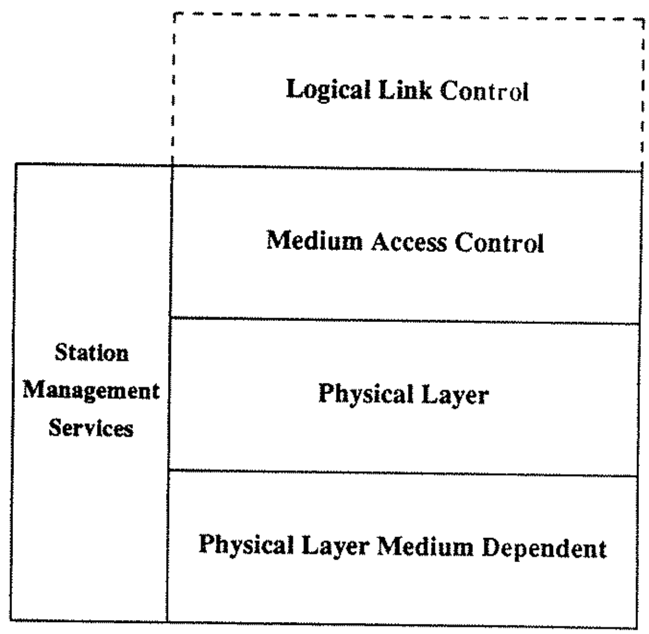


Figure 2.1 — Relationship of FDDI Layers

2.2. Terms

The following terms are used in this chapter to discuss the operation of FDDI.

Concentrator: a dual station containing an additional PHY entity for each single station attached to it.

Data Rate: the speed at which data is transmitted through the network medium (measured in bits per second).

Dual Station: a station consisting of two PHY entities.

Entity: the active functional agent within each FDDI layer (see Figure 2.1).

Fiber Optics: a communication medium whereby light waves are used to transmit data over transparent optical fibers.

NRZI: Non-Return to Zero Inverted.

Octet: a data unit consisting of eight bits (a pair of data symbols).

Primitive: an element of the service interaction between adjacent FDDI layers (see Figure 2.1).

Single Station: a station consisting of a single PHY entity.

4-into-5 Bit Encoding: an encoding scheme whereby four bits of data are translated into five signalled bits to improve transmission reliability.

2.3. General Description

A token ring consists of a set of stations connected by a transmission medium to form a closed loop. Each station serves as the means for attaching one or more devices to the ring, whereafter the attached device(s) are able to communicate with other devices on the ring.

Stations may be in one of two states: transmitting or repeating. Only one station may be transmitting information at any one time. Information on the ring is encoded in the form of symbols (see Section 2.5.1.1) which are transmitted serially and unidirectionally. While one station is transmitting, all other stations are repeating the symbols they receive. The receiving station(s) copies the symbols it receives. Symbols on the ring circulate one full circumference before being removed by the transmitting station.

A station may transmit information only after it has detected and captured a token circulating on the medium. The token is the means by which the right to transmit (as opposed to the normal process of repeating) is passed from one station to another. The token is a unique sequence of symbols that cannot be duplicated by other data on the medium. A station "captures" the token by removing it from the ring. When the capturing station concludes its transmission, it puts a new token back on the ring so another station can have the opportunity to transmit.

Information to be transmitted is bundled in a standard format known as a frame. The frame contains a destination address field and a source address field. A station wishing to transmit a frame(s) must first capture a usable token. As the frame circulates around the ring each station checks the destination address. When a station recognizes the frame's destination address as its own, it begins to copy the frame into an internal buffer. The receiving station performs a validity check on the frame it has copied and modifies a status field at the end of the frame to indicate to the transmitting station whether the frame was successfully copied.

Stations identify frames they have sent by checking the frame's source address field. When a station recognizes one of its own frames, it does not repeat that frame, but emits *Idle* symbols (see Section 2.5.1.1) instead, thus removing the frame from the ring. The transmitting station then checks the status field (modified by the receiving station) to determine if the frame was successfully copied by the receiving station.

A station may continue to send frames until it has no more frames to send, or until its preallocated transmission time expires. Limiting a station to a certain transmission time ensures an upper bound on the time it takes a token to circulate completely around the ring.

Within each station three classes of service are provided: Synchronous, Asynchronous, and Immediate. Synchronous service is used for data that must be delivered in a known and bounded amount of time. Synchronous service would be used for applications such as real-time voice. When the network is initialized, each station negotiates for the percentage of ring bandwidth it may use for synchronous traffic. Asynchronous service is used for data whose bandwidth requirements are less predictable or whose response time requirements are less critical. Asynchronous service would be used for interactive applications. Asynchronous bandwidth is allocated from the pool of remaining unallocated and/or unused ring bandwidth. If the time it takes the token to circulate once around the ring (token cycle time) rises above a predefined level, asynchronous traffic is suspended until the token cycle time returns to an acceptable level. Immediate service is reserved for extraordinary situations such as ring recovery. Frames issued at this level of service pre-empt all frames currently circulating on the ring.

Error detection and recovery are provided to restore ring operation in the event that the ring begins to deviate from normal operation. Detection and recovery functions are common to all stations so that no station is a critical resource.

2.4. Physical Layer Medium Dependent Requirements (PMD)

2.4.1. PMD Architecture

At the heart of FDDI station-to-station communication is the physical connection. As shown in Figure 2.2, a physical connection consists of two stations whose physical layers are connected over the transmission medium by a primary and secondary link. A primary link consists of an output (primary out) of a physical layer communicating over the primary physical medium to the input (primary in) of a second physical layer. A secondary link consists of an output (secondary out) of the second physical layer communicating over the secondary physical medium to the input (secondary in) of the first physical layer.

Two classes of stations are defined: dual attachment stations having two PHY entities, and single attachment stations having only one PHY entity. Physical FDDI rings may consist of only dual attachment stations to accommodate the dual (counter rotating) rings. Certain dual stations can provide additional PHY entities for the attachment of single stations. These dual stations are known as concentrators. A concentrator has an additional PHY entity for each single station attached to it. A concentrator may place offline an attached single station that has failed. In this way, the failure of a single attachment station has no effect on the physical ring and only minimal effect on the logical ring. It should be noted that the logical FDDI ring consists of all attached stations.

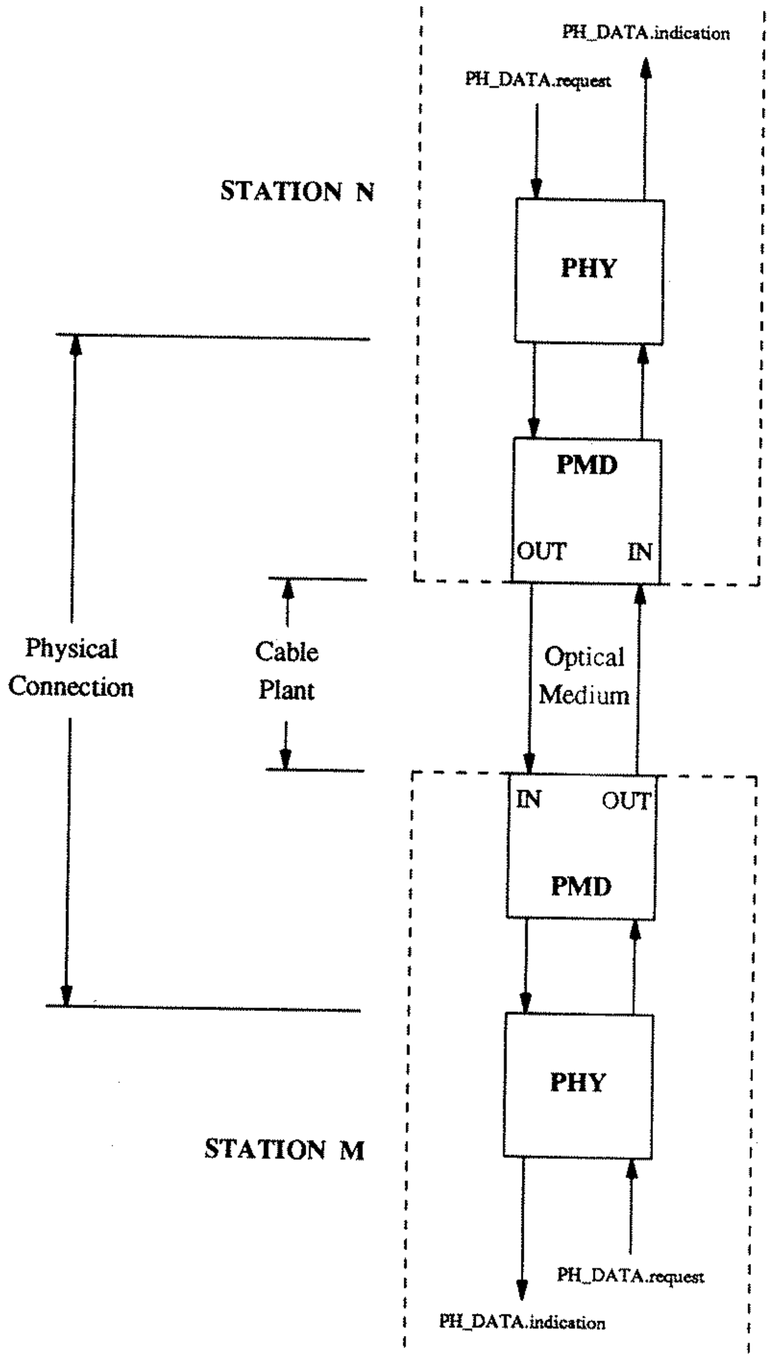


Figure 2.2 — Physical Connection

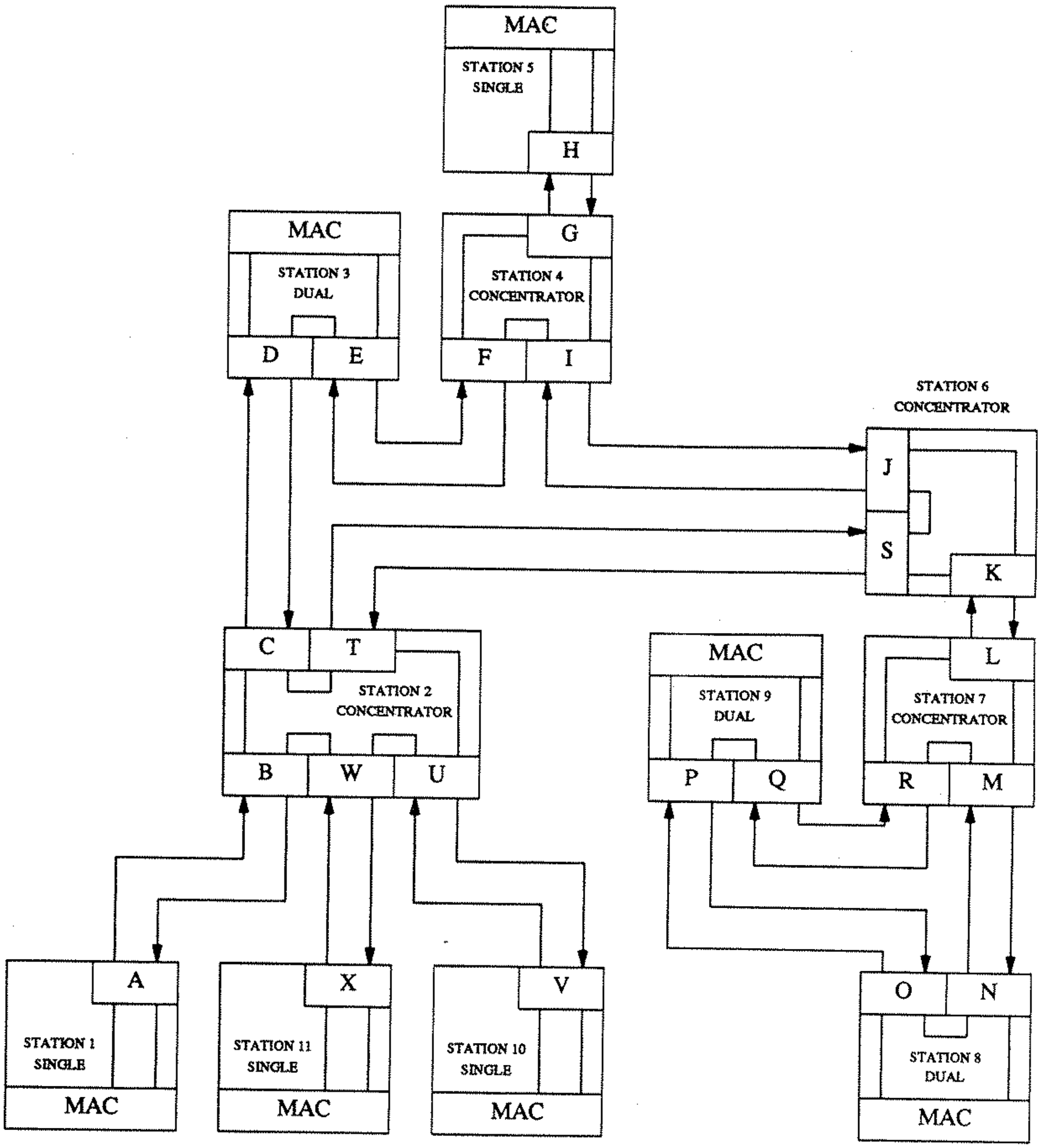


Figure 2.3 — FDDI Topology Example

Figure 2.3 depicts an acceptable FDDI ring topology. Two physical rings are shown here. One is comprised of stations 2, 3, 4, and 6, and the other is comprised of stations 7, 8, and 9. Station 2 is a concentrator for three single attachment stations (stations 1, 10, 11), and station 4 is a concentrator for one single attachment station (station 5). Stations 6 and 7 form what is known as a trunkline between the two physical rings. The numbering of the stations indicates the order in which the stations would be visited by a given symbol circulating on the logical FDDI ring.

2.4.2. PMD Services

PMD provides services to PHY which is logically above it, and to SMT which is logically beside it.

2.4.2.1. PMD to PHY Services

PM_DATA.request (*code*)

This primitive is used in conjunction with the transmitter being enabled. With this primitive, PHY continuously sends to PMD the current NRZI encoding polarity (either logical "0" or logical "1"). PMD responds by transmitting this logical "bit" of information in the form of a low optical level for logical "0" or a high optical level for logical "1".

PM_DATA.indication (*code*)

With this primitive PMD sends to PHY the NRZI encoding for the bit most recently received from the medium.

PM_SIGNAL.indication (*status*)

This primitive is sent to PHY by PMD. It indicates the status of the optical signal being received by PMD. The *status* sent to the PHY indicates whether the incoming optical signal is above (*status* = *On*) or below (*status* = *Off*) the optical signal detection threshold.

2.4.2.2. PMD to SMT Services

Services requested by SMT are performed pre-emptively over those requested by PHY.

SM_PM_CONTROL.request (*action*)

This primitive is sent to PMD by SMT to control the optical transmitter. The *action* parameter has two settings, either *Transmit_Enable* or *Transmit_Disable*. If the *action* requested is *Transmit_Enable* then PMD begins transmitting the signals requested by **PM_DATA.request**. If the *action* requested is *Transmit_Disable* then PMD begins transmitting logical "0" regardless of what **PM_DATA.request** is indicating.

SM_PM_BYPASS.request (*action*)

This primitive is sent to PMD by SMT to indicate whether or not the station wishes to be an active part of the ring. The *action* parameter has two settings, either *Insert* or *Deinsert*. If the *action* requested is *Insert* then PMD will route the inbound optical signal to the optical receiver and the outbound optical signal to the outbound medium. The station becomes an active member of the ring, and resumes normal station operation. If the *action* requested is *Deinsert* then PMD will route the inbound optical signal through the optical switch to the outbound medium. The output of the optical transmitter is also directed back through the optical switch and becomes input to the optical receiver. This process of sending the outbound optical signal back into the station's optical receiver is known as "loopback". Upon receiving the *Deinsert* parameter the station becomes an inactive member of the ring.

SM_PM_SIGNAL.indication (*status*)

With this primitive PMD sends to SMT the status of the optical signal being received. The *status* sent to SMT indicates whether the incoming optical signal is above (*status* = *On*) or below (*status* = *Off*) the optical signal detection threshold.

Figure 2.4 depicts the interaction these six primitives have with PMD in a dual attachment station.

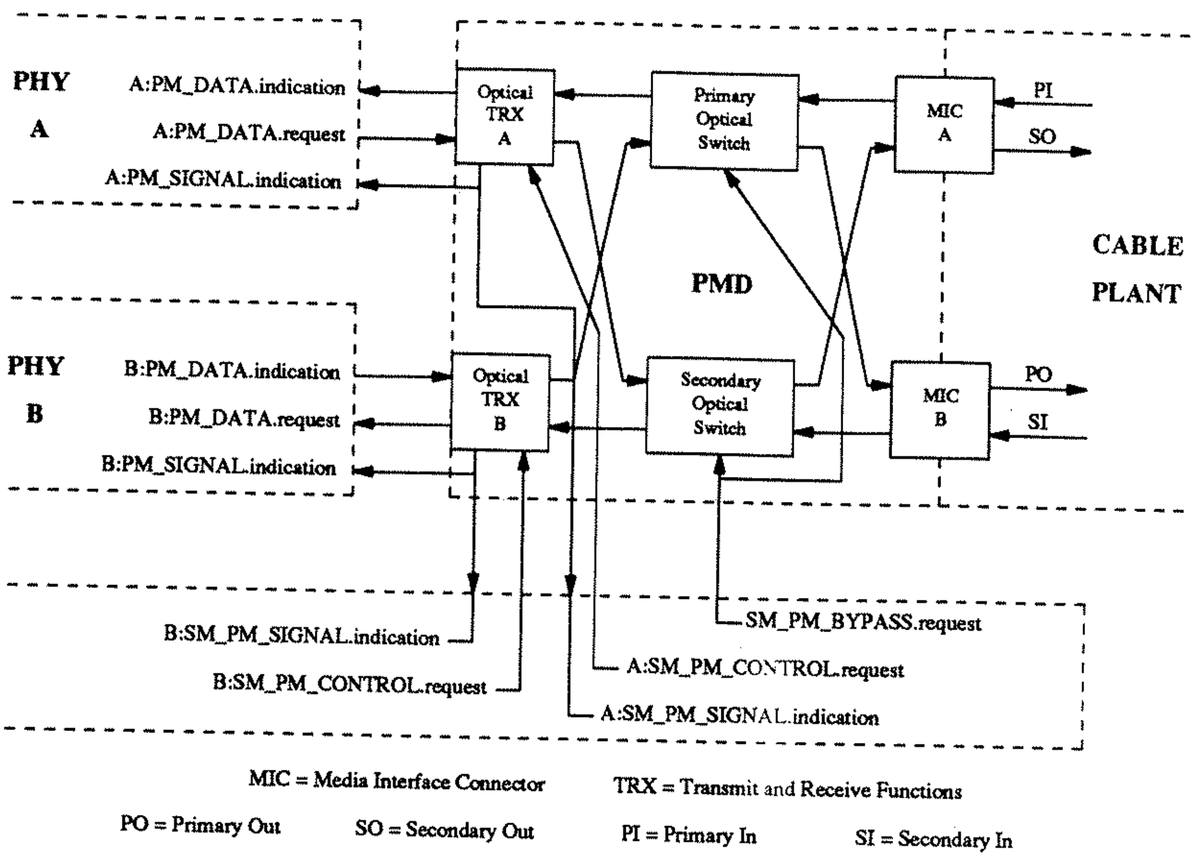


Figure 2.4 — PMD Services

2.5. Physical Layer Protocol (PHY)

2.5.1. PHY Facilities

2.5.1.1. Coding and the Symbol Set

Physical layers communicate with each other via a fixed length code-bit. A code-bit is the smallest signalling entity used by the physical layer. A sequence of five consecutive code-bits is known as a code-group. To increase the reliability of the signalling system, FDDI does not view data as simply 1's and 0's. Data is instead viewed as groups of four bits known as symbols. PHY is responsible for encoding symbols into code-groups, and for decoding code-groups into symbols. It performs this translation using a 4-into-5 bit encoding scheme. Symbols are transmitted over the medium by PMD in their code-group equivalent.

As shown in Figure 2.5, there are 32 possible five bit sequences. Of these possible sequences, 16 are data code-groups used to represent the hexadecimal numbers 0 through F. Eight of these sequences are used as control or indication symbols, and the remaining eight are invalid.

Three of the symbols are line state symbols. These symbols are only found on the medium between valid transmissions. A line state symbol within a frame will cause the termination of that data transmission. The *Quiet* **Q** line state corresponds to the code-group "00000" and is sent when the medium is to be quiet. This code group is appropriate since "0" is signaled with low optical power. The *Halt* **H** line state forces a break in the logical activity of the medium, and the *Idle* **I** line state is the normal signal on the medium between frames.

Three of the symbols are control symbols. These symbols are used to delineate beginning and ending boundaries of valid frames. The starting boundary of a frame is known as the starting delimiter (see Section 2.6.1.2.2). The starting delimiter is comprised of the symbol **J** (represented by code-group "11000"), followed by the symbol **K** (represented by code-group "10001"). The starting delimiter is unique in that the **JK** code-bit sequence can not be duplicated by any other legal symbol sequence. The **JK** sequence is therefore recognizable whether or not it falls on established code-group boundaries. This characteristic is necessary when a new transmission needs to pre-empt the current transmission. Whenever the physical layer encounters the **JK** code-bit sequence, it uses it to establish new code-group boundaries. The ending boundary of a frame is known as the ending delimiter (see Section 2.6.1.2.7). A frame's ending delimiter is comprised of a single **T** symbol and a token's ending delimiter is comprised of a **TT** symbol pair.

Two of the symbols are control indicators. These symbols are used by a receiving station to modify a frame it has copied. For example, the receiving station modifies the frame's "message copied" indicator to indicate to the sending station whether that frame was successfully copied. The *Reset* symbol **R** denotes a logical zero, and would indicate failure to copy the frame successfully. The *Set* symbol **S** denotes a logical one, and would indicate the frame was successfully copied.

DECIMAL	CODE GROUP	SYMBOL	ASSIGNMENT	
CONTROL AND INDICATION SYMBOLS				
00	00000	Q	QUIET Line State	
04	00100	H	HALT Line State	
31	11111	I	IDLE Line State	
24	11000	J	1st Symbol of Starting Delimiter	
17	10001	K	2nd Symbol of Starting Delimiter	
13	01101	T	Used to Terminate the Data Stream	
07	00111	R	Denotes Logical ZERO (*Reset*)	
25	11001	S	Denotes Logical ONE (*Set*)	
DATA SYMBOLS			**HEX**	**BINARY**
30	11110	0	0	0000
09	01001	1	1	0001
20	10100	2	2	0010
21	10101	3	3	0011
10	01010	4	4	0100
11	01011	5	5	0101
14	01110	6	6	0110
15	01111	7	7	0111
18	10010	8	8	1000
19	10011	9	9	1001
22	10110	A	A	1010
23	10111	B	B	1011
26	11010	C	C	1100
27	11011	D	D	1101
28	11100	E	E	1110
29	11101	F	F	1111
INVALID CODE ASSIGNMENTS				
01	00001	V or H		
02	00010	V or H		
03	00011	V		
05	00101	V	Codes 01, 02, 08 and 16 shall be interpreted as Halt when received.	
06	00110	V		
08	01000	V or H		
12	01100	V		
16	10000	V or H		

(12345) = sequential order of code-bit transmission.

Figure 2.5 — PHY Symbol Coding Table

The remaining eight code-groups are violation symbols. These symbols are never transmitted, but they may be received due to transmission errors. If a violation symbol is received, it is generally ignored.

2.5.1.2. Line States

Line states are signaled by the PHY transmitter on request by SMT. They are also detected by the PHY receiver and reported to SMT. The following line state definitions are not exhaustive. Line conditions may arise that do not satisfy the criteria for any defined state. In the event this occurs, the line state is considered unknown. The *Unknown* line state is the default line state.

Quiet_Line-State (QLS)

Quiet_Line-State is initiated by transmitting a continuous stream of **Q** symbols. This line state is entered when a station wishes to break the physical connection with its outbound logical neighbor.

Master_Line-State (MLS)

Master_Line-State is initiated by transmitting a continuous stream of alternating **H** and **Q** symbols. This line state is entered by a master PHY in a concentrator to signal its presence to its outbound logical neighbor. This line state is used by the master PHY to disable and assert control over the physical connection.

Halt_Line-State (HLS)

Halt_Line-State is initiated by transmitting a continuous stream of **H** symbols. This line state is used by a peer or slave PHY to signal its presence to its outbound logical neighbor. This line state disables the associated physical connection without asserting control over it.

Idle_Line-State (ILS)

Idle_Line-State is initiated by transmitting a continuous stream of **I** symbols. This line state is used to establish and maintain clock synchronization with the outbound logical neighbor. During normal ring operation, **I** symbols are transmitted between frame transmissions.

Active_Line-State (ALS)

Active_Line-State is entered upon receipt of a **JK** symbol pair. It is an indication that the inbound logical neighbor has enabled the associated physical connection.

Noise_Line-State (NLS)

Noise_Line-State is initiated when a series of unrelated symbols is received without satisfying the criteria for entry to another line state. A PHY never intentionally transmits a series of symbols that would cause the outbound logical station to enter this state. However, transmission errors make entry into this line state a possibility.

2.5.2. PHY Services

PHY provides services to MAC which is logically above it, to PMD which is logically below it, and to SMT which is logically beside it.

2.5.2.1. PHY to MAC Services

PH_DATA.request (*symbol*)

This primitive is sent to PHY from MAC whenever MAC has a symbol to output. Upon receipt of this primitive, PHY encodes and transmits the symbol.

PHY_DATA.indication (*symbol*)

This primitive is sent to MAC from PHY every time PHY decodes a symbol. This primitive occurs every symbol period (5 code-bit times).

PH_DATA.confirmation (*status*)

This primitive is sent to MAC from PHY to confirm the acceptance of the last symbol sent and to indicate willingness to accept another symbol. MAC uses this primitive to synchronize its delivery of symbols with the medium data rate.

PH_INVALID.indication (*invalid*)

This primitive is sent to MAC from PHY whenever PHY detects a *Quiet*, *Halt*, *Master* or *Noise_Line-State*.

2.5.2.2. PMD to PHY Services

See Section 2.4.2.1.

2.5.2.3. PHY to SMT Services

Services requested by SMT are performed pre-emptively over those requested by MAC.

SM_PH_LINE-STATE.request (*linestate*)

This primitive is sent to PHY by SMT. SMT uses this primitive to command PHY to signal a particular line state. If the *linestate* parameter is *Transmit_Quiet*, PHY begins to signal a continuous stream of Q symbols. If the *linestate* parameter is *Transmit_Idle*, PHY begins to transmit a continuous stream of I symbols. If *linestate* is *Transmit_Master*, PHY signals alternating Q and **H** symbols. If *linestate* is Transmit_PDR, PHY begins to transmit the symbols presented to it on the **PH_DATA.request** interface.

SM_PH_STATUS.indication (*status*)

PHY sends this primitive to inform SMT of line state activity and status changes. The *status* parameter can indicate any of the following line states: *Quiet*, *Halt*, *Master*, *Idle*, *Active*, *Noise*, and *Unknown*. *Unknown* line state is indicated when a defined line state has been exited and conditions for a new line state have not yet been satisfied. In the event *Unknown* line state is entered, this primitive also indicates the most recently known line state.

SM_PH_CONTROL.request (*action*, *status*)

This primitive is used by SMT to control PHY. The *action* parameter commands PHY to do one of the following: *Reset*, *Present_Status*, *Begin_Loopback*, or *End_Loopback*. If the *action* parameter is *Reset*, PHY will perform various reset functions which include transmitting Q symbols and resetting the line state to *Unknown*. If the *action* parameter is *Present_Status*, PHY will return to SMT the current line state in the *status* parameter. If the line state is *Unknown*, it indicates this and also indicates the most recently known line state. If the *action* parameter is *Begin_Loopback*, PHY begins to send the symbols it receives at the **PH_DATA.request** interface back via the **PH_DATA.indication** interface. PHY in effect sends the symbols it receives from MAC back to MAC without putting them on the network medium. A station enters loopback mode to perform local station testing. While in this mode PHY sends PMD a continuous stream of zeros. If the *action* parameter is *End_Loopback*, PHY exits loopback mode.

2.6. Medium Access Control (MAC)

2.6.1. MAC Facilities

2.6.1.1. Protocol Data Units

MAC uses two Protocol Data Units (PDU) formats: tokens and frames. Figures 2.6 depicts the formats of these PDUs in the order of transmission (leftmost symbol transmitted first).

2.6.1.2. PDU Fields

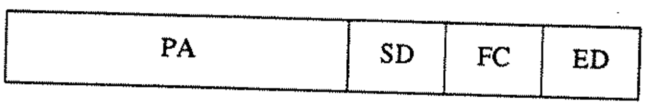
2.6.1.2.1. Preamble (PA)

PA consists of 16 or more *Idle* **I** symbols. The length of PA may vary, consistent with physical layer clocking requirements. A station is required to process a frame preceded by a PA of 12 or more **I** symbols. If fewer than 12 symbols are received the frame is ignored.

2.6.1.2.2. Starting Delimiter (SD)

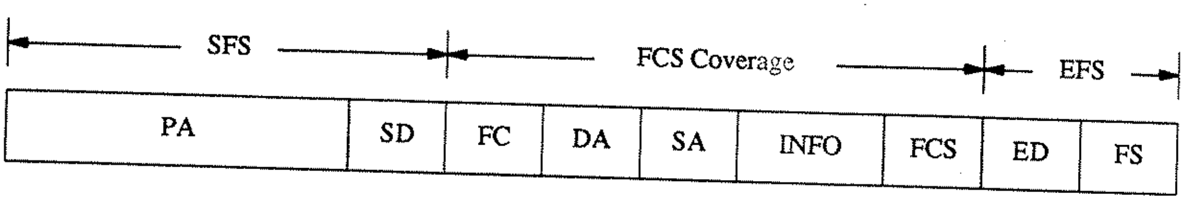
SD consists of a **J** symbol followed by a **K** symbol. No frame or token is considered valid unless it starts with this symbol sequence.

TOKEN



PA = Preamble (16 or more symbols)
SD = Starting Delimiter (2 symbols)
FC = Frame Control (2 symbols)
ED = Ending Delimiter (2 symbols)

FRAME



SFS = Start of Frame Sequence
PA = Preamble (16 or more symbols)
SD = Starting Delimiter (2 symbols)
FC = Frame Control (2 symbols)
DA = Destination Address (4 or 12 symbols)
SA = Source Address (4 or 12 symbols)

INFO = Information (0 or more symbol pairs)
FCS = Frame Check Sequence (8 symbols)
EFS = End of Frame Sequence
ED = Ending Delimiter (1 symbol)
FS = Frame Status (3 or more symbols)

Figure 2.6 — Protocol Data Unit Formats

2.6.1.2.3. Frame Control (FC)

FC is one octet in length. The individual bits of FC take on the following meanings:

C L F F	Z Z Z Z

C = Class Bit

L = Address Length Bit

FF = Format Bits

ZZZZ = Control Bits

The class bit (C) indicates the service class of the frame.

C = 0 indicates an asynchronous frame
C = 1 indicates a synchronous frame

The address length bit (L) indicates the length of the destination and source addresses.

L = 0 indicates a 16 bit address
L = 1 indicates a 48 bit address

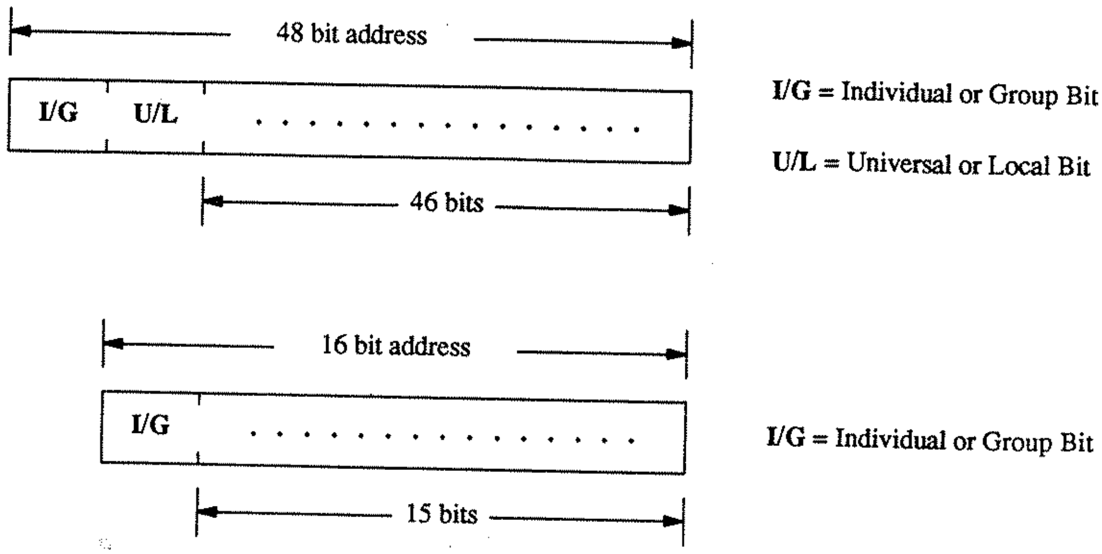
The frame format bits (FF) work in conjunction with the class (C), address length (L), and control (ZZZZ) bits to indicate the frame and token types as listed below.

CLFF ZZZZ to ZZZZ	
0X00 0000	Void frame
1000 0000	Nonrestricted Token
1100 0000	Restricted Token
0L00 0001 to 1111	Station management frame
0L00 0001 to 1111	MAC frame
CL01 r000 to r111	LLC frame
CL10 r000 to r111	Reserved for implementor
CL11 rrrr	Reserved for future standardization

NOTE: X is either 0 or 1 and r is reserved for future standardization and should be set to zero.

2.6.1.2.4. Destination Address (DA) and Source Address (SA)

Each frame contains two address fields: Destination Address (DA) and Source Address (SA). Addresses can be either 16 or 48 bits in length, but all stations have 16 bit address capability.



For both address sizes the first bit transmitted indicates whether it is an individual or group address. An individual address identifies a particular station and a group address is used to address a frame to multiple stations. For 48 bit addresses the second bit transmitted indicates whether the address is locally or universally administered. Universal address administration implies that all network addresses are unique. Two addresses are reserved: an address of all 1s (broadcast address) means all stations on the ring, and an address of all 0s (null address) is not the address of any station.

2.6.1.2.5. Information (INFO) Field

The INFO field contains zero or more symbol pairs whose meaning is determined by the Frame Control (FC) field. The field length is variable, but limited in that the maximum PDU length (including all PDU fields and four symbols for the preamble) is 9000 symbols.

2.6.1.2.6. Frame Check Sequence (FCS)

MAC checks the accuracy of the FC, DA, SA, and INFO fields using a degree 32 standard generator polynomial G(X). The contents of these fields, of length k bits, is called the "message" and is treated as a degree k - 1 polynomial. A 32 bit field is appended to the trailing end of the message. The message, now of length k + 32 bits, is algebraically manipulated in these last 32 (appended) bit positions so that when divided by G(X) there is one and only one 32 bit result, called C(X). When the message is transmitted MAC puts C(X) in the four octets of FCS. In the receiving station the message and its appended 32 bits are divided by G(X). If the result matches FCS then the probability is extremely high that the message was received correctly. If the result differs from FCS then the message was received incorrectly and the frame is discarded.

2.6.1.2.7. Ending Delimiter (ED) and Frame Status (FS)

The only valid symbol for an ED is the **T** symbol. The ED of a token consists of a **TT** symbol pair while the ED of a frame is a single **T** symbol. FS consists of three or more control indicator (**R** or S) symbols and a possible **T** symbol at the end of the control indicators. The additional **T** symbol is appended to FS if there is an even number of control indicators. In this way the single **T** symbol of ED and the symbols of FS are paired evenly.

The three mandatory control indicators (flags) represent (in order of transmission): *Error_Detected*, *Address_Recognized*, and *Frame_Copied*. These are initially transmitted as **R** symbols by the transmitting station. The first station to detect a transmission error on a particular frame (i.e., when the frame check sequence does not match) sets the *Error_Detected* flag by replacing the **R** symbol with an S symbol. If a station recognizes the Destination Address (DA) as its own it sets the *Address_Recognized* flag. After the destination station checks the frame, it sets the *Frame_Copied* flag if it finds it error-free and successfully copies the frame. When the transmitting station receives its own frame, it checks these flags to determine whether the frame was successfully received.

2.6.1.3. Timers

Each station maintains three timers to regulate the operation of the ring: Token-Holding Timer (THT), Valid-Transmission Timer (TVX), and Token-Rotation Timer (TRT).

2.6.1.3.1. Token-Holding Timer (THT)

THT controls how long a station may transmit asynchronous frames. A station may begin transmission of an asynchronous frame if THT has not expired and is less than the *T_Pri* (see Section 2.6.2.3.2) associated with the frame to be transmitted. THT is initialized with the current value of TRT (see Section 2.6.1.3.3) when the token is captured.

2.6.1.3.2. Valid-Transmission Timer (TVX)

The TVX timer is used to detect ring inactivity. TVX is set to the time required for the largest allowable frame and token to circulate around the ring and is reset whenever a valid frame is received. Expiration of TVX indicates that a serious ring error has occurred and causes a station to begin reinitialization procedures.

2.6.1.3.3. Token-Rotation Timer (TRT)

TRT controls ring scheduling during normal operation and assists in detection of serious ring error situations. When TRT expires it is reinitialized with a pre-negotiated operational value (*T_Opr*) and the late token counter (*Late_Ct*) is incremented.

2.6.1.4. Frame Counters

MAC maintains three frame counts: *Frame_Ct*, *Error_Ct*, and *Lost_Ct*. Each station increments *Frame_Ct* when it receives a valid frame. *Error_Ct* is incremented by the first station to detect an error in a particular frame. *Lost_Ct* is incremented when a station in the process of receiving a frame or token detects an error such that the credibility of that PDU can be questioned. The remainder of the PDU that caused *Lost_Ct* to be incremented is not repeated and consequently stripped from the ring.

2.6.2. MAC Operations

2.6.2.1. Frame and Token Transmission

Data needing transmission comes to MAC in the form of a Service Data Unit (SDU). MAC constructs a frame by placing the SDU in the INFO field of the frame. Frames remain queued awaiting the receipt of a suitable token. When a station receives and captures a usable token it begins transmitting its queued frame(s). During transmission, the Frame Check Sequence (FCS) for each frame is generated and appended to the end of the INFO field. A station concludes transmission when it has no more frames to send or when THT has expired.

Upon conclusion of transmission the station may generate a new token or wait for the return of one or more of the frames it sent. While waiting, the station transmits I symbols. This option gives the station the opportunity to examine the Frame Status (FS) fields to determine the reception status of the last message(s) it sent. Because FDDI allows up to 1000 stations connected over a maximum ring circumference of 200 km, the time penalty incurred by waiting for one or more messages to return is large, relative to the speed of the ring. This option is therefore not used with normal data transmission.

2.6.2.2. Frame Stripping

Each station is responsible for stripping from the ring the frames it sent. A station knows to begin stripping when the received frame's Source Address (SA) matches its own address. Because the decision to strip is based on recognition of the SA field, frame remnants consisting of PA, SD, FC, DA, and SA remain on the ring. These remnants cause no harm and are removed from the ring when they encounter a transmitting station.

2.6.2.3. Ring Scheduling

Methods for token access and frame transmission are among the most important aspects of FDDI. PDU transmission is controlled by a "timed token" protocol supporting two major classes of service: synchronous and asynchronous. Synchronous service is for applications requiring guaranteed bandwidth and response time. Asynchronous service is for applications whose bandwidth requirements are less predictable. Asynchronous bandwidth is allocated from the pool of bandwidth unallocated and/or unused by the synchronous class.

To control ring scheduling each station's MAC maintains a Token-Rotation Timer (TRT). This timer keeps track of elapsed time between token arrivals. At ring (re)initialization each station determines the token rotation time it needs to service its synchronous traffic. These values are made known during the claim token process (see Section 2.6.2.4.1) and the smallest of them is chosen to be the Target Token Rotation Time (TTRT). When a station receives a token, TRT is compared with TTRT. If TRT is less than TTRT (i.e., the token is early) then the token may be used for both synchronous and asynchronous messages. If TRT is equal to or greater than TTRT (i.e., the token is late) then the token may only be used for synchronous messages.

It has been shown by Sevcik and Johnson [SEVC85] that this protocol guarantees the average TRT to be no greater than TTRT and the maximum TRT to be no greater than twice TTRT.

2.6.2.3.1. Synchronous Traffic

Each station's synchronous bandwidth allocation is initially zero. Through the exchange of SMT PDUs a station may request a non-zero allocation. The sum of station allocations must not exceed the available bandwidth. Synchronous bandwidth may be expressed as a percentage of TTRT. The maximum available bandwidth is expressed as:

$$Available\_Bandwidth = TTRT - (ring\_latency + token\_time + frame\_max)$$

Available bandwidth calculations for a "nominal" configuration of 100 stations with a 10 km ring would be as follows (assuming 8 symbols or 320 ns internal station latency and a propagation delay of 5085 ns/km):

$$320\ ns\ latency \cdot 100\ stations = 32\ \mu s\ total\ latency$$

$$10\ km\ ring\ length \cdot 5085\ ns/km\ propagation = 50.85\ \mu s\ propagation\ delay$$

$$32\ \mu s\ latency + 50.85\ \mu s\ propagation\ delay = 82.85\ \mu s\ ring\_latency$$

$$6\ symbol\ token + 16\ symbol\ preamble\ =\ 0.88\ \mu s\ token\_time$$

$$9000\ symbol\ frame\ max. + 16\ symbols\ preamble\ =\ 361\ \mu s\ frame\_max\ time$$

$$Available\_Bandwidth\ =\ TTRT - (82.85 + 0.880 + 361\ \mu s)$$

$$Available\_Bandwidth\ =\ TTRT - 444.73\ \mu s$$

$$Available\_Bandwidth\ =\ TTRT - 445\ \mu s$$

$$Available\_Bandwidth\ =\ TTRT - 0.445\ ms$$

Assume the available bandwidth is distributed evenly among the 100 stations and the negotiated TTRT is 9 ms. Each station's synchronous bandwidth allocation is:

$$\frac{(9\ ms - 0.445\ ms)}{100}\ =\ 0.08555\ ms$$

A station would have 100% allocation if it were to transmit for ($TTRT - 0.445\ ms$).

Support for synchronous transmission is optional.

2.6.2.3.2. Asynchronous Transmission

Each station's asynchronous bandwidth allocation is taken from the pool of bandwidth unused by synchronous traffic. This bandwidth is controlled by a two-tier allocation scheme enforced by two classes of tokens: restricted tokens and nonrestricted tokens. Restricted tokens are used to dedicate asynchronous bandwidth to a single extended dialog between specific requesters. Nonrestricted tokens are used for normal asynchronous traffic where bandwidth is divided fairly among all requesters.

Nonrestricted token mode is the normal mode of operation. In this mode multiple asynchronous priorities may be recognized. Associated with each priority is a threshold value *T_Pri(n)*. A nonrestricted token may only be captured for transmission of a priority *n* frame when the current Token-Rotation Timer (TRT) is less than *T_Pri(n)*. Differentiation among priorities is achieved by giving low threshold values to low priority levels and high threshold values to high priority levels.

When a usable token arrives the value of TRT is saved in the Token-Holding Timer (THT), and TRT is reset to time the next token rotation. The difference between the THT and TTRT is then computed. This difference represents the remaining asynchronous bandwidth available to the station. A frame of priority *n* may be transmitted only when THT is less than the associated priority threshold value *T_Pri(n)*.

Multiple priority operation is optional. If it is not implemented all asynchronous frames have an effective threshold value equal to TTRT.

Restricted token mode is used for extended dialog requiring basically all unallocated ring bandwidth. To enter restricted token mode the initiating station captures a nonrestricted token, sends its frames and issues a restricted token. Stations that are to take part in the restricted token dialog are notified by the initiating station. Upon notification these stations enter restricted token mode. Only stations in restricted token mode may access restricted tokens.

Normal Asynchronous traffic is suspended while the stations in restricted token mode pass restricted tokens among themselves. To conclude a restricted dialog a station issues a nonrestricted token and communication reverts to normal.

During restricted token mode asynchronous traffic is suspended. To ensure fairness SMT monitors restricted token dialogs and enforces a maximum restricted dialog time. Synchronous traffic is not interrupted by a restricted token dialog; it responds to both restricted and nonrestricted tokens. In this way a restricted token dialog uses only the asynchronous network bandwidth.

2.6.2.4. Ring Monitoring

Each station monitors the ring for invalid conditions requiring ring (re)initialization. The need for (re)initialization results from either inactivity, or incorrect activity, on the ring, or by SMT request. Inactivity is detected by the expiration of TVX and incorrect activity is detected by counting successive expirations of the Token-Rotation Timer (TRT). Ring monitoring is accomplished by the claim token process, the initialization process, and the beacon process.

2.6.2.4.1. Claim Token Process

Ring (re)initialization is initiated by the claim token process. When this process is entered, one or more stations begin transmitting claim frames, each containing that station's bid for the Target Token Rotation Time (TTRT). As the claim frames circulate, stations compare their bid with the bids they receive. If a station receives a bid higher than its own, the station ignores that bid. A station receiving a bid lower than its own must yield to that bid. Conflicts are resolved as follows:

1. the lowest TTRT bid has precedence;
2. given equal bids, the station with the longest address has precedence (i.e., 48 bit addresses win over 16 bit addresses)
3. given equal address lengths, the numerically highest address has precedence.

When one station receives its own claim frames returning to it, the claim token process is complete and a "winner" has been found. The winning station may then begin ring (re)initialization.

A station entering the claim token process sets its Token-Rotation Timer (TRT) to a value large enough to permit ring recovery. If TRT expires and the station is still in the claim token process, then the process has failed and that station enters the beacon process (see Section 2.6.2.4.3). External intervention is necessary when the claim token process fails.

2.6.2.4.2. Initialization Process

Ring initilization takes place when one station has completed the claim token process and "won" the right to initialize the ring. This station sets the TTRT value equal to its own bid (since it won the bidding contest), resets its TRT, and issues a nonrestricted token. The first token rotation serves to align the TTRT values and TRT timers in all other stations. Stations are not operational during the first token rotation so no station captures the token and no frames are transmitted. This first rotation also serves to put each station into an operational state and to increment each station's late token count (*Late_Ct*).

On the second token rotation synchronous traffic is serviced, but because *Late_Ct* was incremented on the previous token visit, no asynchronous messages are transmitted. Receipt of the second token will clear each station's *Late_Ct* so on the third token rotation the ring is completely functional.

2.6.2.4.3. Beacon Process

Failure of the claim token process or a request from SMT will cause a station to initiate the beacon process. The purpose of the beacon process is to signal to other stations that a major break has occurred and to provide diagnostic assistance for recovery. In most cases the beacon process is entered because the topology of the ring has been altered. A station may have been inserted or deleted, or two rings may have joined into one, or one ring split into two, or the ring may have been broken.

Upon entering the beacon process, a station continuously transmits beacon frames containing its address. Stations receiving beacon frames from upstream neighbors begin repeating those frames instead of their own. If the cause of the problem is a break in the ring, eventually the beacon frames of the first station downstream from the break will be the only frames on the ring. SMT may use this information to configure a path around the break (using the counter-rotating secondary ring). This information may also help an outside agent diagnose the problem. A station receiving its own beacon frames immediately concludes that the problem no longer exists and enters the claim token process.

2.6.3. MAC Services

MAC provides service to LLC which is logically above it, to PHY which is logically below it, and to SMT which is logically beside it.

2.6.3.1. MAC to LLC Services

MA_DATA.request (*FC_value(1), destination_address(1), M_SDU(1),*
requested _service_class(1), stream(1),
.
.
.
FC_value(n), destination_address(n), M_SDU(n),
requested _service_class(n), stream(n), token_class)

With this primitive LLC requests MAC to transmit one or more SDUs. Each set of *FC_value, destination_address, M_SDU, requested_service*, and *stream* parameters specifies one frame for transmission. The *FC_value* supplies the Frame_Control (FC) field for each frame to be transmitted. The *destination_address* supplies an individual or group address for each frame to be transmitted. Each

M_SDU supplies an LLC SDU and *requested_service_class* specifies either synchronous or asynchronous service. If asynchronous service is selected then a *token_class* and *priority* may be optionally specified. The *stream* parameter, if set, indicates that this SDU is part of a stream of messages (i.e., multiple messages). The *stream* parameter, if reset, indicates that this SDU is the last of a stream. A stream of SDUs is transmitted until the *requested_service_class* and TRT indicate that no more frames may be sent. At this time the stream is terminated and a token (restricted or nonrestricted) as indicated by the *token_class* parameter is issued.

MA_DATA.indication (*FC_value, destination_address, source_address, M_SDU, reception_status*)

MAC sends this primitive to LLC indicating the arrival of a frame addressed to this station. The *FC_value* parameter contains the FC field of the frame received. *Destination_address* and *source_address* contain the DA and SA fields of the incoming frame. *M_SDU* contains the SDU of the frame received and *reception_status* indicates whether this was a *Good_Frame* or a *Bad_Frame*. If it was a *Bad_Frame* then the reason for error (*Invalid_FCS, Invalid_Frame_Length,* or *Internal_Error*) is also indicated. *Reception_status* also includes the FC field of the frame received. This field contains the *Error_Detected, Address_Recognized* and *Frame_Copied* flags.

MA_DATA.confirmation (*number_of_SDUs, transmission_status, provided_service_class*)

MAC sends this primitive to LLC after each **MA_DATA.request**. The *number_of_SDUs* parameter indicates the number of SDUs transmitted in the last stream. *transmission_status* (applied to all SDUs transmitted in the last stream) indicates whether all SDUs were acknowledged. The exact meaning of the *transmission_status* parameter is implementor dependent. *Provided_service_class* indicates whether the SDUs were transmitted as synchronous or asynchronous messages.

MA_TOKEN.request (*requested_token_class*)

This primitive is sent by LLC to MAC to request the capture of the next token of the *requested_token_class* (i.e., restricted or nonrestricted). When the token is captured, MAC begins transmitting I symbols until an associated **MA_DATA.request** is received. Because of its adverse affect on throughput this primitive is intended only for rare, time critical operations.

2.6.3.2. PHY to MAC Services

See Section 2.5.2.1.

2.6.3.3. MAC to SMT Services

SMT uses these services to monitor and control the operation of MAC. These services are performed pre-emptively over services requested by LLC or PHY.

SM_MA_INITIALIZE_PROTOCOL.request (*individual_MAC_address(16 bits), individual_MAC_address(48 bits), group_MAC_address, T_Min_value, T_Max_value, T_Reg_value, T_Neg_value, T_Pri_value, indicate_for_own_frame, indicate_for_receive_only_good_frame*)

This primitive is used by SMT to change the operational parameters of MAC. The first three parameters supply the short, long, and group MAC addresses. *T_Min_value* and *T_Max_value* specify the minimum and maximum TTRT this station may accept. *TVX_value* specifies the value of the TVX timer. *T_Reg_value* specifies this station's request (bid) for TTRT and *T_Neg_value* specifies the negotiated TTRT (lowest bid received so far). *T_Pri_value* specifies a set of priority thresholds. *Indicate_for_own_frame*, if set, specifies that MAC should receive and indicate to SMT frames it transmitted. *Indicate_for_receive_only_good_frame*, if set, specifies that MAC should indicate only "good" frames to SMT.

SM_MA_INITIALIZE_PROTOCOL.confirmation (*status*)

MAC sends this primitive to SMT to indicate the success or failure of an initialization request.

SM_MA_CONTROL.request (*control_action, beacon_information, requested_status, requested_condition*)

SMT uses this primitive to control MAC operation. The *control_action* parameter commands MAC to do one of the following: reset, enter beacon process, present its current status, reset its counters, interrupt when certain conditions are detected, or intentionally send a frame with a bad FCS. The remaining three parameters contain information supplemental to *control_action*.

SM_MA_STATUS.indication (*status_report*)

MAC sends this primitive to SMT to indicate any of 15 major status changes. *Status_report* can indicate: receipt of a station's own beacon frames, expiration of TVX or TRT, overflow of frame counter, or occurrence of a condition previously specified by **SM_MA_CONTROL.request.**

SM_MA_DATA.request (*FC_value(1), ..., stream(n), token_class*)

This primitive is used by SMT to transmit one or more SMT SDUs. The format and usage of the parameters follows that used for **MA_DATA.request** (see Section 2.6.3.1).

SM_MA_DATA.indication (*FC_value, ..., reception_status*)

MAC uses this primitive to report any SMT or MAC frames received. The format and usage of the parameters follows that used for **MA_DATA.indication** (see Section 2.6.3.1).

SM_MA_DATA.confirmation (*number_of_SDUs, transmission_status, provided_service_class*)

MAC generates this primitive in response to the **SM_MA_DATA.request**. The format and usage of the parameters follows that used for **MA_DATA.confirmation** (see Section 2.6.3.1).

SM_MA_TOKEN.request (*requested_token_class*)

SMT uses this primitive to request the capture of the next token (restricted or nonrestricted) as indicated by *requested_token_class*. When the token is captured, MAC begins transmitting **I** symbols until an associated **SM_MA_DATA.request** is received. Because of its adverse affects on throughput this primitive is intended only for rare, time critical operations.

2.7. Station Management (SMT)

SMT provides the control necessary to coordinate the activities of the various FDDI layers so that each station may work cooperatively with other stations on the ring.

2.7.1. SMT Services

2.7.1.1. MAC to SMT Services

See Section 2.6.3.3.

2.7.1.2. PHY to SMT Services

See Section 2.5.2.3.

2.7.1.3. PMD to SMT Services

See Section 2.4.2.2.

Chapter 3

The Simulation Environment

3.1. Introduction

A simulation of the FDDI protocol was chosen as the most practical means of analyzing its performance. The FDDI simulator is part of a simulation environment which also handles the SAE AS4074.2 High Speed Ring Bus. The entire simulation system (simulators and user interface code) is written in Pascal on a SUN 3/50 workstation.

3.2. The FDDI Simulation Process

The simulation process is divided into three phases: information gathering, computation, and data display. The information gathering phase allows the user to set up the network configuration to be simulated. Menus guide the user through a series of input screens. As input is entered, bounds checks are performed to ensure that the input complies with limits established by the FDDI standard. The computation phase performs the actual simulation of network activity, and the data display phase allows the user to view the various performance metrics computed by the simulation.

3.2.1. Information Input Phase

The information input phase of the simulation uses menus to guide the user through a series of input screens. Input screens are segregated by the type of information they gather. The first input screen collects general network information common to both FDDI and SAE. The second input screen collects information specific to either FDDI or SAE, depending on which protocol is being simulated. In the case of FDDI, input values such as Target Token Rotation Time (TTRT) are entered on this screen. The third input screen collects information specific to a certain type (class) of station (see Section 3.2.1.3).

```
Enter Global Network Parameters

      1-   Number of stations in the ring: 100

      2- Transmission rate (in Mbits/sec): 100.000

      3-   Ring circumference (in meters): 10000

      4-   Station latency (in bit times): 32

      5- Time to run simulation (seconds): 0.100

      6-    Number of classes of stations: 1

      7-Number of asynchronous priorities: 8

      8-                   Offered load : 85

Enter number of field to change [0 to exit]:
```

Figure 3.1 — General Network Information Input Screen

3.2.1.1. General Network Information

Figure 3.1 depicts the general information input screen. This input is common to both FDDI and SAE simulations. The number of stations, transmission rate, and ring length variables determine some of the major physical characteristics of the network being simulated. FDDI specifies only one medium rate, therefore medium rate is not a true variable here; it is included for generality since it is a variable in the SAE HSRB. Station latency and the number of asynchronous priorities determine some of the physical characteristics of each individual station. Simulation time determines the amount of time being simulated. Target offered load specifies a target for the number of bits to be transmitted during the simulation relative to network capacity. The number of station classes determines how many different station types (classes) this simulation will have (see Section 3.2.1.3).

```
1-             Enter target token rotation time (in msec) : 9.000
2-          Enter mean restricted token duration (in msec) : 0.000
3- Percentage of simulation time in restricted token mode : 0

Distribution types are: Exponential = 1,  Constant = 2,  Uniform = 3

4-                  Restricted token frequency distribution : 1
5-                   Restricted token duration distribution : 1

6-  Are we simulating from an initial configuration?  Y/N : n
7-          Will we have a synchronous message queue?  Y/N : y

Enter asynchronous priority thresholds (in msec)

 8-      (low)  Priority 0 : 1.000
 9-             Priority 1 : 1.500
10-             Priority 2 : 2.000
11-             Priority 3 : 2.500
12-             Priority 4 : 3.000
13-             Priority 5 : 3.500
14-             Priority 6 : 4.000
15-     (high)  Priority 7 : 4.500


        Enter number of field to change [0 to exit]:
```

Figure 3.2 — FDDI Specific Input Screen

3.2.1.2. FDDI Specific Information

Figure 3.2 depicts the input screen for information specific to FDDI. The FDDI simulator does not perform station negotiation to set TTRT, but instead TTRT is controlled by the user and input on this screen. The use of restricted token mode is optional for any simulation. Restricted token mode is governed by the following inputs:

1. Mean duration of a single restricted token dialog.
2. Percentage of total simulation time to be spent in restricted token mode.
3. Restricted token mode frequency distribution (exponential, constant, uniform).
4. Restricted token mode duration distribution (exponential, constant, uniform).

The initial configuration variable allows the user to simulate the special token rotations that occur at (re)initialization (see Section 2.6.2.4.2). Support for synchronous transmission is optional. The synchronous message queue variable allows the user to choose whether this simulation will utilize synchronous message transmission. The service threshold value(s) (*T_Pri(n)*) for each implemented asynchronous priority are the final inputs entered on this screen.

3.2.1.3. Station Class Information

The simulation allows up to ten different station types (classes) in a single simulation. The input for each station class is captured on the class input screen as depicted in Figure 3.3. For each station class the user must specify the following:

1. Class name

2. Number of stations in this class. The sum of stations in all classes must equal the total number of stations as entered on the first input screen (see Figure 3.1).

3. Percentage of offered load which this class contributes to the total offered load. The sum of offered load percentages in all classes must equal 100%.

4. For each implemented priority of service the following information is needed:

    * Percentage of offered load which this priority contributes to this class
    * Mean message length
    * Message length distribution (exponential, constant, uniform)
    * Message frequency distribution (exponential, constant, uniform)

5. Percentage of synchronous bandwidth available to this class. The sum of synchronous bandwidth percentages allocated to all classes must equal 100%.

6. Restricted token eligibility.

3.2.2. Computation Phase

The computation phase utilizes an event driven simulation consisting of two subphases: event set-up and event execution. An event driven simulation encounters events one at a time (as if in chronological order) and reacts to each event by performing a certain function or functions. A typical event in the FDDI simulator would be the arrival of message *x* in queue *p* at station *n*; a subsequent event (if time allowed) would be the transmission of message *x* in queue *p* at station *n* to station *m*.

```
(1)                              Class name: SpaceStation
(2)          Number of stations this class: 100
(3) % of total offered load for this class: 100

Distribution types are:   Exponential = 1,  Constant = 2,  Uniform = 3
                                            Mean MesLen
 Priority      % Off. Load    MesArrDist     In Octets      MesLenDist
 --------      -----------    ----------    -----------     ----------
low   0         (4)   5        (5)   2       (6)   32        (7)   3
      1        (12)  10       (13)   1      (14)  478       (15)   2
      2        (12)  10       (13)   3      (14)  227       (15)   2
      3        (16)  12       (17)   2      (18)  1024      (19)   3
      4        (20)  13       (21)   1      (22)  941       (23)   1
      5        (24)  14       (25)   2      (26)  512       (27)   3
      6        (28)  14       (29)   1      (30)  77        (31)   2
high  7        (32)  12       (33)   2      (34)  2048      (35)   3
Synchronous    (36)  10       (37)   3      (38)  3019      (39)   2

(40)  Percent of available bandwidth used by this class: 100
(41) Is this class eligible for a restricted token? Y/N: n

      Enter number of field to change [0 to exit]:
```

Figure 3.3 — Class Information Input Screen

The event set-up subphase pregenerates a list of events (messages) for each priority in each station. Each pregenerated message has an arrival time which dictates how the messages are ordered in each list (queue). The numerically smallest arrival time is first in the queue and the numerically largest arrival time is last. The event set-up phase is also responsible for determining the times at which a restricted token mode dialog is entered and exited, should that option be used.

The event execution subphase performs the actual simulation of network activity. The simulation begins at time 0 and continues until time *t*, where *t* is the user-specified time limit for this particular simulation. During event execution the pregenerated queues of messages are checked one station at a time according to time constraints established by the FDDI standard. The only messages recognized are those whose arrival time is less than or equal to the time elapsed since the simulation began. As the simulation time advances, those pregenerated messages with later arrival times are eventually recognized and acted upon. The event execution subphase is also responsible for calculating the various performance metrics accumulated by the simulator, such as mean queueing delay of a certain message priority and mean token cycle time.

3.2.3. Output Phase

At the conclusion of a simulation the output screen shown in Figure 3.4 is displayed. From this screen the user can exit the simulation environment, produce a file containing the data collected by the simulator, view network-wide data collected by the simulator, or view data specific to a certain priority and station class. The information provided on the lower portion of this screen is there to remind the user of the basic network configuration used for this simulation.

```
1 = General network information     2 = Specific queue information

3 = Route all output to a file      4 = Exit system

                    ENTER (1 - 4) :

                 Output file name :

           Number of stations in network :  100
           Network ring length in meters :  10000.000
            Medium rate in bits per sec. :  100000000.000
              Simulation time in seconds :  0.100
                 Station latency in bits :  32

           Number of classes of stations :  1
            Number of message priorities :  8
           Target offered load percentage:  85

     Target token rotation time in msec. :  9.000
     Mean restricted token time in msec. :  0.000
Percent of time in restricted token mode :  0
                   Synchronous queue used :  YES
```

Figure 3.4 — Output Screen Menu

3.2.3.1. Network Wide Data

Figure 3.5 depicts the screen displaying the network-wide data collected by the simulator. The data may be viewed for the network as a whole or for individual station classes. The data on this screen produced by the simulator is as follows:

1. packets sent (number of messages sent for the entire simulation)
2. offered load in bits (bits offered relative to medium capacity)
3. fraction of ring capacity, total bits sent, and total bits received
4. throughput (bits received relative to medium capacity)
5. mean number of tokens late per station
6. mean number of tokens accessed per station
7. mean number of tokens received per station
8. mean number of packets sent per station
9. mean and maximum token cycle time
10. maximum arrival and residual queue length

```
                GENERAL INFORMATION FOR :  THE ENTIRE NETWORK

                     Number of stations :  100
               Percentage of offered load :  85
        Percent available sync. band width :  100
                           Packets sent :  8285
                    Offered load in bits :  8664280
                 Fraction of ring capacity :  0.866
                         Total bits sent :  8559125
                     Total bits received :  8540944
                              Throughput :  0.854

            Mean number of tokens late :  0.000
        Mean number of tokens received :  51.430
        Mean number of tokens accessed :  39.270
           Mean number of packets sent :  82.850
                  Mean token cycle time :  1.944e-03

               Maximum token cycle time :  3.694e-03
           Maximum arrival queue length :  4
          Maximum residual queue length :  2

    ENTER class number (0 for network) (99 to EXIT) :
```

Figure 3.5 — General Information Output Screen

3.2.3.2. Priority Data

Figure 3.6 depicts the screen displaying the data collected for each implemented priority of service (including synchronous). This data may be viewed for the network as a whole or for individual station classes. The simulator collects the following data for each implemented priority:

1. mean queueing delay (mean time a message spends in a queue with messages ahead)
2. mean network access delay (mean time a message spends at the head of a queue)
3. mean waiting delay (the sum of queueing delay and network access delay)
4. mean service delay (mean time for a message to get delivered)
5. standard deviation of queueing delay
6. standard deviation of network access delay
7. standard deviation of waiting delay
8. standard deviation of service delay
9. mean arrival queue length and mean residual queue length

```
                PRIORITY INFORMATION FOR :  CLASS 1
                                PRIORITY :  Synchronous
                            Packets sent :  1014

       The following delays are in units of SECONDS

                 Mean queueing delay :   8.329e-05
           Mean network access delay :   9.799e-04
                  Mean waiting delay :   1.063e-03
                  Mean service delay :   1.114e-03

            Std. dev. queueing delay :   3.059e-04
      Std. dev. network access delay :   7.510e-04
             Std. dev. waiting delay :   7.030e-04
             Std. dev. service delay :   7.029e-04

 Mean token arrival queue length :   1.968e-01
      Mean residual queue length :   2.037e-03

      ENTER priority number (0 to 8, 9 to exit) :
 ENTER class number (0 for network information) :
```

Figure 3.6 — Priority Information Output Screen

3.3. Justification of Simulation Results

A simulator is a very powerful tool, but must be used with care to avoid producing invalid or misleading results. Before we draw conclusions for any set of simulations, we must be assured that (1) the simulator has reached steady state and (2) that the results it produces are bounded by a reasonable confidence interval.

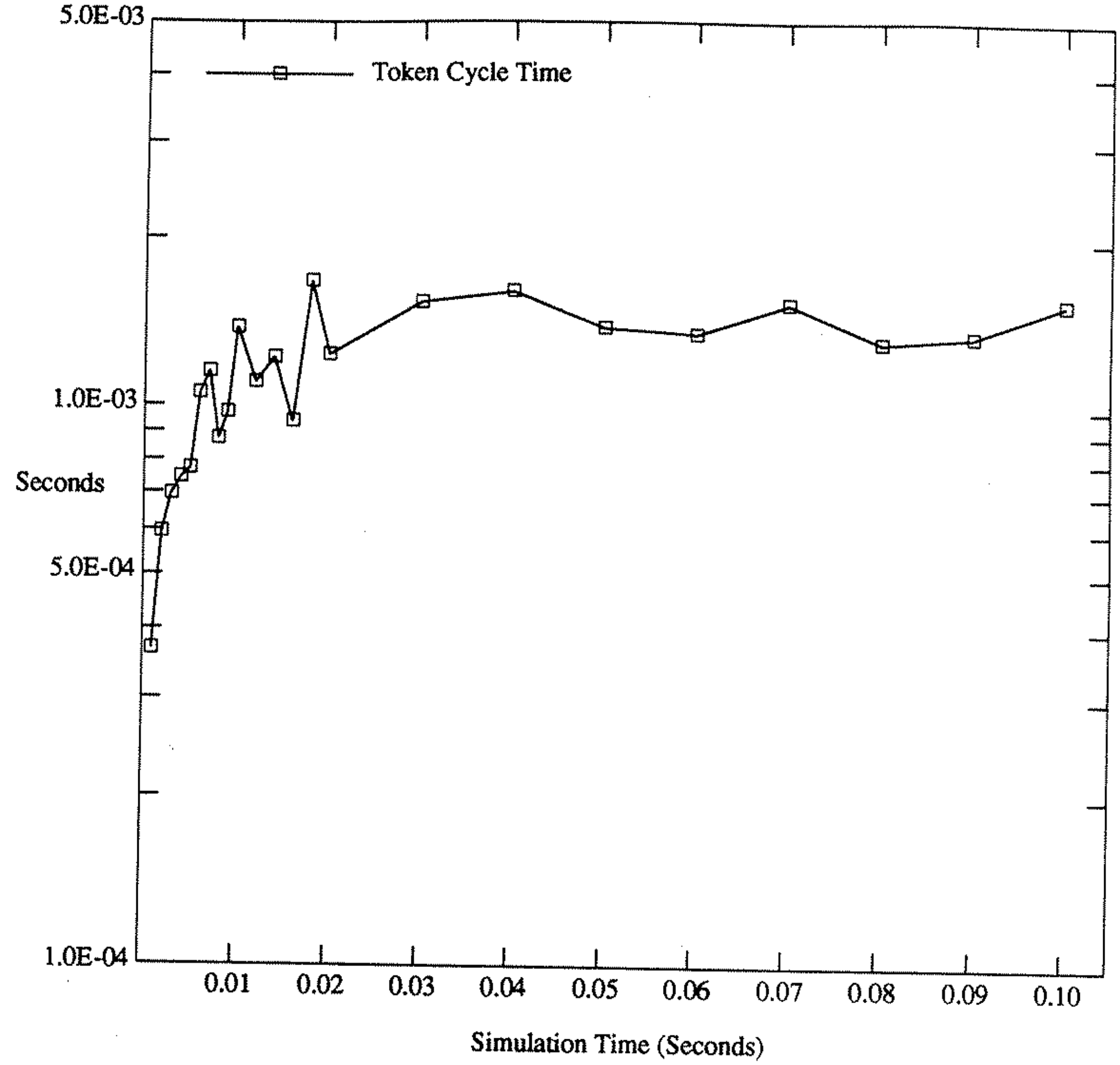
3.3.1. Steady State Estimation

Figures 3.7 and 3.8 present graphs which help us to analyze when the simulator has reached steady state. The network configuration used to produce these graphs is as follows:

100 stations evenly distributed on the ring
100 Mbps medium rate
10 km ring circumference
32 bits internal station latency
8 asynchronous priorities and a synchronous priority
85% offered load
9.0 ms target token rotation time
9.0 ms asynchronous priority thresholds
exponential packet arrival distribution
1024 bit mean packet length distributed exponentially
no restricted token dialog

The independent variable in Figures 3.7 and 3.8 is simulation time which is varied from 1 ms to 100 ms. The performance metrics graphed in these figures were chosen because of their importance to overall network performance.

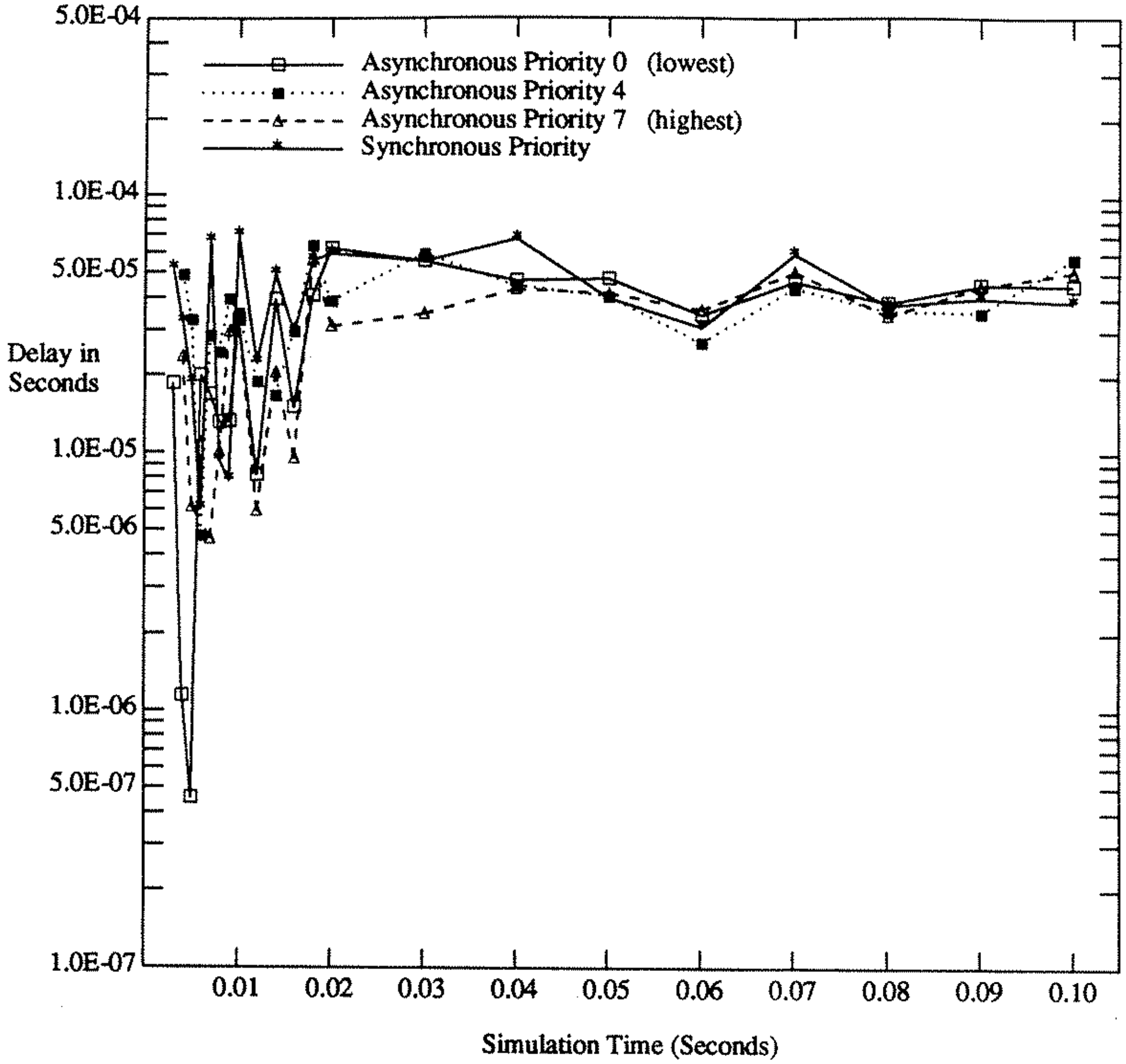
In Figure 3.7 we see that token cycle time begins to stabilize at approximately 30 ms and remains stable as simulation time increases. In Figure 3.8 we see queueing delay for synchronous and asynchronous traffic exhibiting similar characteristics. It should be noted that each point on these graphs represents the average of an increasing number of internal results as simulation time increases. For instance, queueing delay for asynchronous priority 4 at 10 ms simulation time represents the mean queueing delay of approximately 85 packets, whereas queueing delay for the same priority at 100 ms simulation time represents the mean queueing delay of approximately 850 packets.



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
85% Offered Load
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 3.7 — Token Cycle Time Stability



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
85% Offered Load
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 3.8 — Queueing Delay Stability

From observations of these graphs and considering the number of internal results produced, we conclude that 100 ms is sufficient time for the simulator to achieve steady state. In addition, we verified that the simulator remained in steady state out to one second of simulation time.

3.3.2. Confidence Interval Estimation

The question that remains is how can we be assured that the results produced by the simulator are bounded by a reasonable confidence interval. To answer this question we cite the work of Peter D. Welch. In [LAVE83, pp. 281 - 297], Welch presents the following argument for estimating the mean of a steady state random variable.

Suppose we have M independent observations of a steady state random variable (i.e., token cycle time) and each generates a steady state output sequence of length N. We define

V_{mn} = n_{th} *member of the output sequence from the* m_{th} *independent replication of the simulation,* $m = 1, \ldots, M, \; n = 1, \ldots, N.$

On each observation we form the sample mean

$$\hat{\mu}_m = \frac{1}{N} \sum_{n=1}^{N} V_{mn} \tag{3.1}$$

The $\hat{\mu}_m$, $m = 1, \ldots, M$, are independent and

$$E[\hat{\mu}_m] \approx \mu, \tag{3.2}$$

hence a reasonable estimate of μ is

$$\hat{\mu} = \frac{1}{M} \sum_{m=1}^{M} \hat{\mu}_m. \tag{3.3}$$

If we let

$$s^2(\hat{\mu}_m) = \frac{1}{M-1} \sum_{m=1}^{M} (\hat{\mu}_m - \hat{\mu})^2 = \frac{1}{M-1} \sum_{m=1}^{M} \hat{\mu}_m^2 - \frac{M}{M-1} \hat{\mu}^2 \tag{3.4}$$

be the sample variance of the $\hat{\mu}_m$ then

$$\frac{\hat{\mu} - \mu}{s(\hat{\mu}_m)/M^{1/2}} \tag{3.5}$$

has approximately a t-distribution with $M-1$ degrees of freedom. The t-distribution is approximately a normal distribution with zero mean and unit variance for large M. Hence, if we let

$$t_n(x) = (100 \cdot x)^{th} \textit{ percentile of the } t\textit{-distribution with } n \textit{ degrees of freedom}, \text{ and } 0 \le x \le 1, \tag{3.6}$$

we have

$$Prob \left\{ t_{M-1}\left[\frac{\alpha}{2}\right] \le \frac{\hat{\mu}-\mu}{s(\hat{\mu}_m)/M^{1/2}} \le t_{M-1}\left[1-\frac{\alpha}{2}\right] \right\} = 1-\alpha. \tag{3.7}$$

The t-distribution is symmetric about zero, therefore we have

$$t_{M-1}(\alpha/2) = t_{M-1}(1-\alpha/2). \tag{3.8}$$

The following confidence interval and statement of probability is obtained by substituting equation (3.8) into equation (3.7) and reworking the inequality:

$$Prob \ \{ \hat{\mu} - t_{M-1}(\alpha/2)\, s(\hat{\mu}_m)/M^{1/2} \le \mu \le \hat{\mu} + t_{M-1}\,(\alpha/2)\, s(\hat{\mu}_m)/M^{1/2} \} \approx 1-\alpha. \tag{3.9}$$

Figure 3.9 shows curves of $t_n(1-\alpha/2)$ for $1-\alpha=0.90$ and $1-\alpha=0.95$. For large n these curves approach the usual multipliers 1.64 and 1.96 of the normal distribution with zero mean and unit variance. This graph shows that for $M \ge 10$ the function $t_M(1-\alpha/2)$ is approximately constant. This implies that the expected width of the confidence interval for our random variable does not appreciably decrease when more than 10 independent observations of the random variable are averaged.

To generate a sequence of independent random variables Welch suggests either the method of independent replications or the method of batch means. The method of independent replications generates independent sequences by repeated runs of the simulation. The method of batch means generates approximately independent sequences by breaking up the output sequence of a single simulation into subsequences which are each long enough to produce a statistically significant data point. This method is more difficult to implement than independent replications, but it is useful when the time needed to run a single simulation is long.

The FDDI simulator requires approximately five wall clock minutes per simulation on a SUN 3/50, so we concluded that the method of independent replications was the best choice for generating independent sequences of random variables. We also concluded from Welch's confidence interval argument that the average of ten independent replications of a simulation would be sufficient to produce a reasonable confidence interval. All experimental results reported herein were generated by averaging ten independent replications.

3.3.3. Comparison of Simulation and Analytic Model

To increase our confidence in the validity of the performance metrics produced by the simulator we now compare some of those results to analytic models. This comparison is not intended to prove the

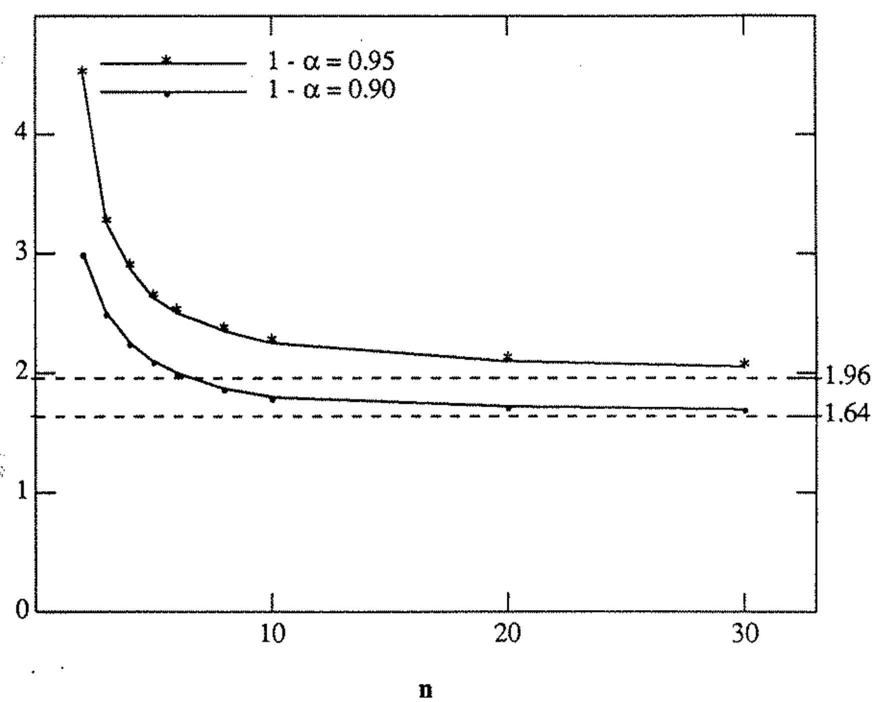


Figure 3.7 — Plots of $t_n(1 - \alpha/2)$ for $1 - \alpha = 0.9$ and $1 - \alpha = 0.95$

correctness of the simulator, but rather to demonstrate that the results produced by the simulator are reasonable estimates of FDDI performance.

Network queueing theory literature provides us with analytic models for basic network performance metrics such as token cycle time and mean packet service delay. To present these models we must define the following:

C ≡ medium capacity in bits per second

$\overline{M}$ ≡ mean packet length in bits

N ≡ number of stations on the ring, numbered $0 \cdots (N-1)$, $N \geq 2$

L_j ≡ station latency in seconds at station j, $0 \leq j \leq N-1$

O_j ≡ protocol overhead at station j, $0 \leq j \leq N-1$

P_j ≡ propagation delay in seconds from station j to station $(j+1)$ *mod* N, $0 \leq j \leq N-1$

λ ≡ mean packet arrival rate in packets per second

μ ≡ mean packet service rate in packets per second $= \frac{C}{\overline{M}}$

$E(TCT)$ ≡ expected token cycle time

$E(Q)$ ≡ expected number of packets enqueued at any station upon token arrival

$E(D)$ ≡ expected packet service delay

One of the most important network performance metrics is token cycle time. Token cycle time is the amount of time necessary for the token to circulate once around the ring. To model token cycle time we introduce terms for the load offered by each station, $\rho = \frac{\lambda}{\mu}$, and ring latency, $\gamma = N(L+P) + N(\rho O)$, where $N(L+P)$ is the total station latency and propagation delay incurred in a single token cycle and $N(\rho O)$ is the total protocol overhead incurred in a single token cycle. Nearly all cyclic service models have these terms in common and FDDI is no exception. As shown by Peden in [PEDE87], equation 3.10 provides an analytic model for expected token cycle time.

$$E(TCT) = \frac{\gamma}{1-N\rho}, \; N\rho < 1 \tag{3.10}$$

The number of packets enqueued at a station when the token arrives is known as the arrival queue length. Peden showed that a model of this performance metric may be derived by a simple modification to the token cycle time model resulting in equation 3.11 for expected arrival queue length.

$$E(Q) = \frac{\lambda\gamma(1-\rho)}{1-N\rho}, N\rho < 1 \qquad (3.11)$$

Another important network performance metric is the delay a packet incurs from the time it is enqueued to the time it is delivered. The total delay a packet experiences is known as packet service delay. Peden shows that equation 3.12 provides an analytic model for expected packet service delay.

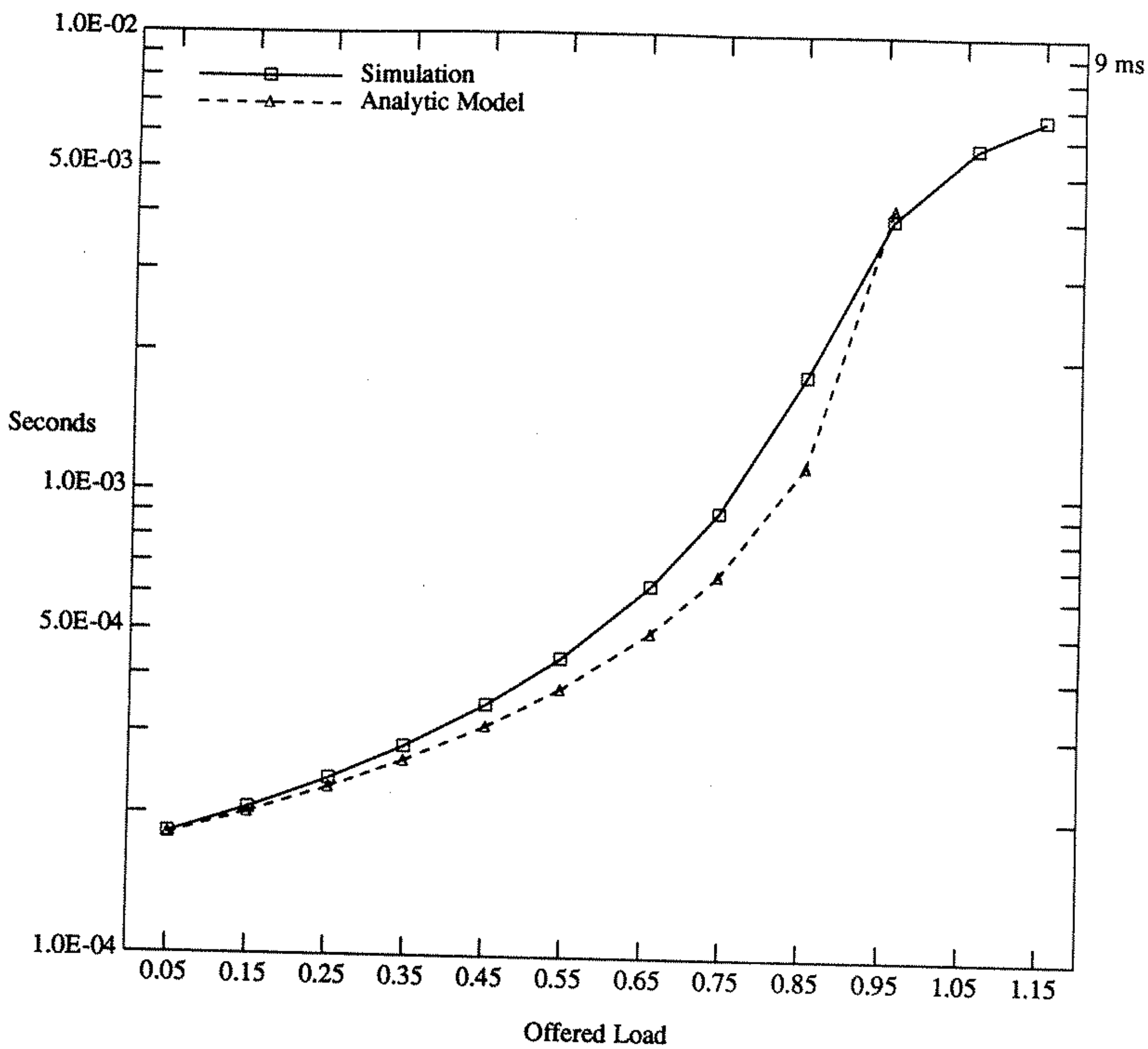
$$E(D) = \frac{N\rho\frac{1}{\mu} + \gamma(1-\rho)}{2(1-N\rho)} + \frac{\gamma\mu+2}{2\mu}, N\rho < 1 \qquad (3.12)$$

With these models our comparison to the simulator was accomplished by selecting a network configuration and comparing the values produced by the simulator for that network to the values produced by the corresponding analytic models. The network configuration we choose for this comparison has the following characteristics:

100 stations evenly distributed on the ring
100 Mbps medium rate
10 km ring circumference
32 bit internal station latency
synchronous priority only
9.0 ms target token rotation time
exponential packet arrival distribution
1024 bit mean packet length distributed exponentially
no restricted token dialog
error free transmission

The motivation for and significance of this network configuration is explained in full detail in the next chapter.

In Figures 3.10, 3.11, and 3.12 we graphically compare the results produced by the simulator for the selected performance metrics to the results produced by our corresponding analytic models. In each of the graphs we see the analytic model differing slightly from the simulator. We attribute this divergence in part to the inability of the analytic model to capture subtle details of the FDDI standard that the simulator can more easily take into consideration, such as station set up time and token transmissions. These differences are, however, very small and we believe that the similarity between the simulator and analytic models for the performance metrics we graphed lends credibility to our belief that the simulator is producing reasonable values for these performance metrics.

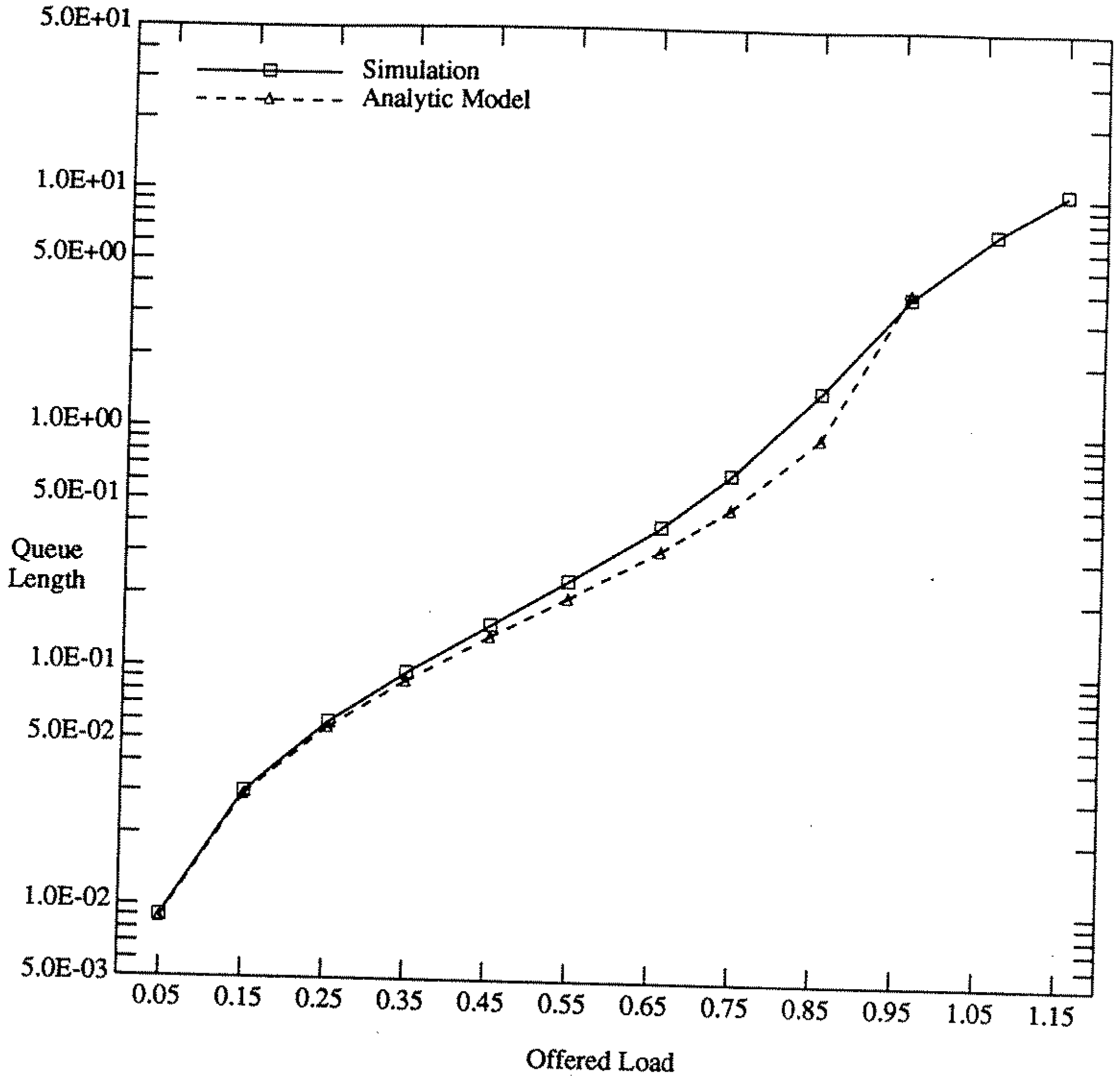


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 3.10 — Token Cycle Time

Simulation vs. Analytic Model

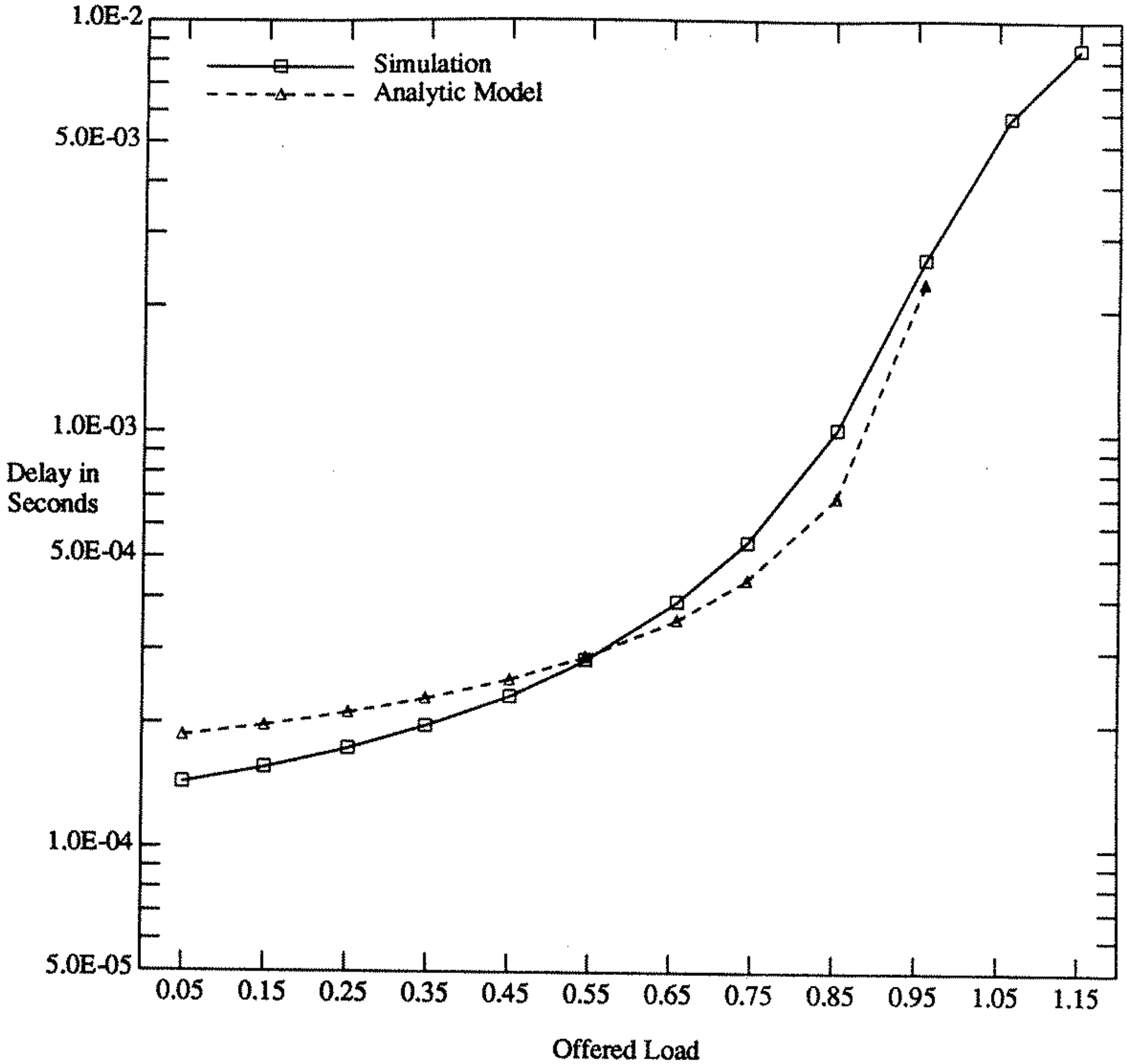


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 3.11 — Token Arrival Queue Length

Simulation vs. Analytic Model



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 3.12 — Mean Packet Service Delay

Simulation vs. Analytic Model

Chapter 4

Performance of the FDDI Token Ring

4.1. Introduction

This chapter presents a performance analysis of the FDDI token ring. Among the performance metrics presented are token cycle time, throughput, queue lengths, and packet delay. The results in this chapter were obtained via simulation and are presented graphically. Each graph has two independent variables; one is the variable of interest (e.g., mean packet length) and the other is always offered load. On all graphs offered load is varied from 5% to 115%. In our context, offered load is measured at the queue level within each station and then summed over all stations on the network, which accounts for the existence of offered loads in excess of 100%. Each graph is accompanied by a brief discussion to enhance the readers' interpretation of the data.

4.2. Single Priority Operation

Since FDDI provides synchronous and asynchronous service it was necessary to determine whether the single priority performance of each was sufficiently different to warrant separate investigation. Through simulation it was observed that no significant differences existed and separate treatment was therefore not justified.

This observation is reinforced by the protocol's description of synchronous and asynchronous transmission. The protocol states that asynchronous transmission uses that portion of bandwidth unused by synchronous transmission. In single priority operation the implemented priority gets all the available bandwidth whether it be synchronous or asynchronous and therefore no significant differences should exist. Differences in these service types arise only when they are used together in the same network configuration. It was concluded that synchronous transmission alone would provide a good indicator of single priority operation.

4.2.1. Reference Configuration

To begin our analysis we must first establish a baseline (reference) configuration against which variations can be compared. Our reference configuration for single priority operation has the following characteristics:

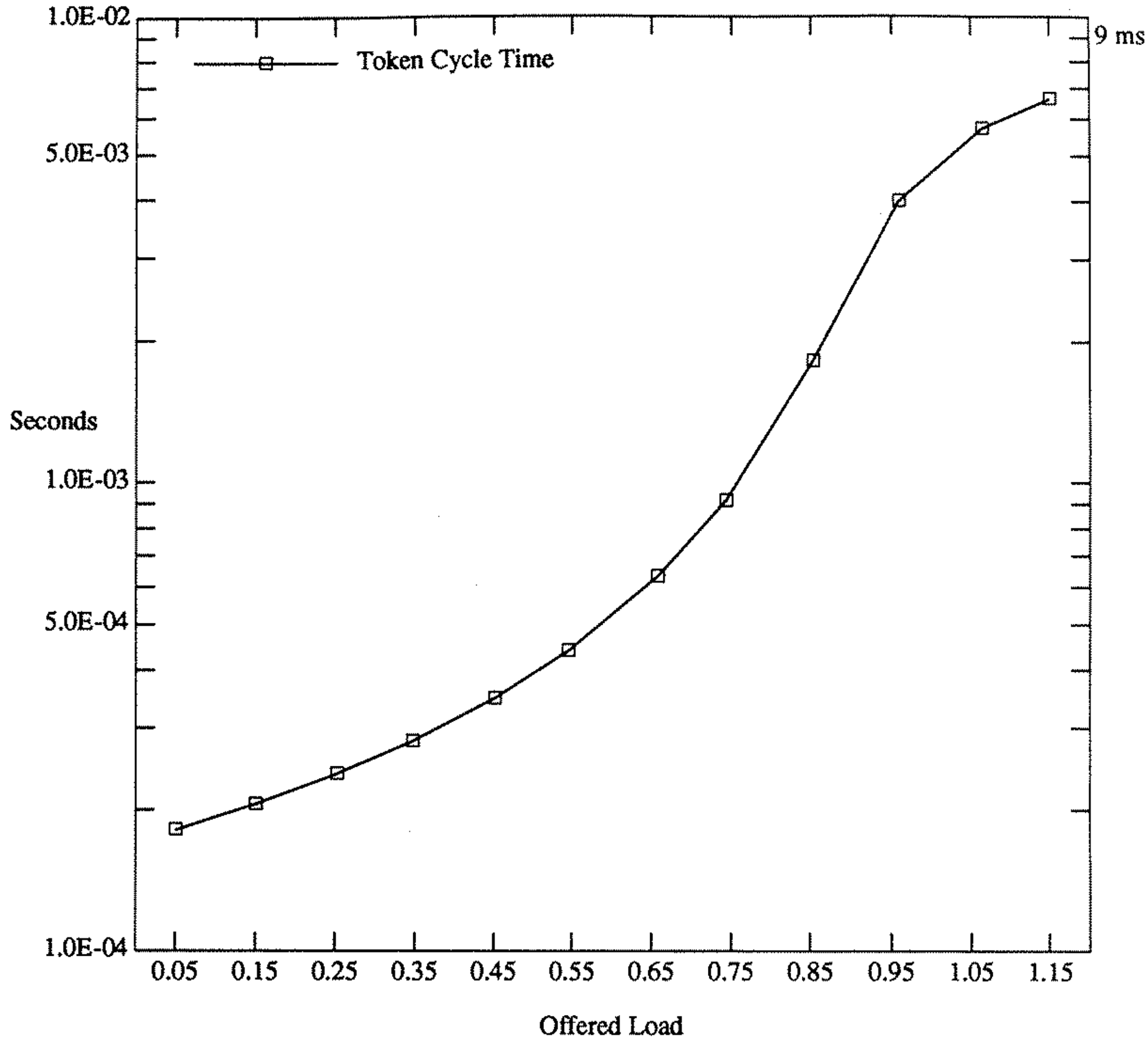
100 stations evenly distributed on the ring
100 Mbps medium rate
10 km ring circumference
32 bit internal station latency
synchronous priority only
9.0 ms target token rotation time
exponential packet arrival distribution
1024 bit mean packet length distributed exponentially
no restricted token dialog
error free transmission

The motivation for this set of network characteristics is simply that it provides a reasonable configuration to which variations can be made and that the target token rotation time is long enough for the network to provide exhaustive service at high offered loads.

4.2.1.1. Token Cycle Time

Token cycle time, also known as token rotation time, is the amount of time necessary for the token to circulate once around the ring. Figure 4.1 shows token cycle time for the single priority reference configuration.

Higher offered loads are achieved by offering more octets per station. Because mean packet length does not vary, generating more octets per station yields more packets per station. Token cycle time increases as offered load increases because there are more packets waiting for transmission per token cycle. Eventually, however, FDDI's "timed token" mechanism (see Section 2.6.2.3) begins to take effect. When this happens, token cycle time stops its steady rise and begins to stabilize as it approaches the target token rotation time (9 ms).



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.1 — Token Cycle Time

Single Priority Reference Configuration

4.2.1.2. Token and Packet Traffic

As the token circulates around the ring it is only captured by stations having packets to transmit. Figure 4.2 plots the number of opportunities a station has to capture the token (tokens received) against the number of tokens it actually captures (tokens accessed). This graph shows that at low offered loads each station receives many tokens, but few of these are actually accessed. As offered load increases, token cycle time goes up and the number of tokens a station receives in a given amount of time begins to decline. The number of tokens accessed rises at first to accommodate increasing offered load, but since a station can access no more tokens than it receives these metrics begin to converge. At very high offered loads a station accesses almost every token it receives.

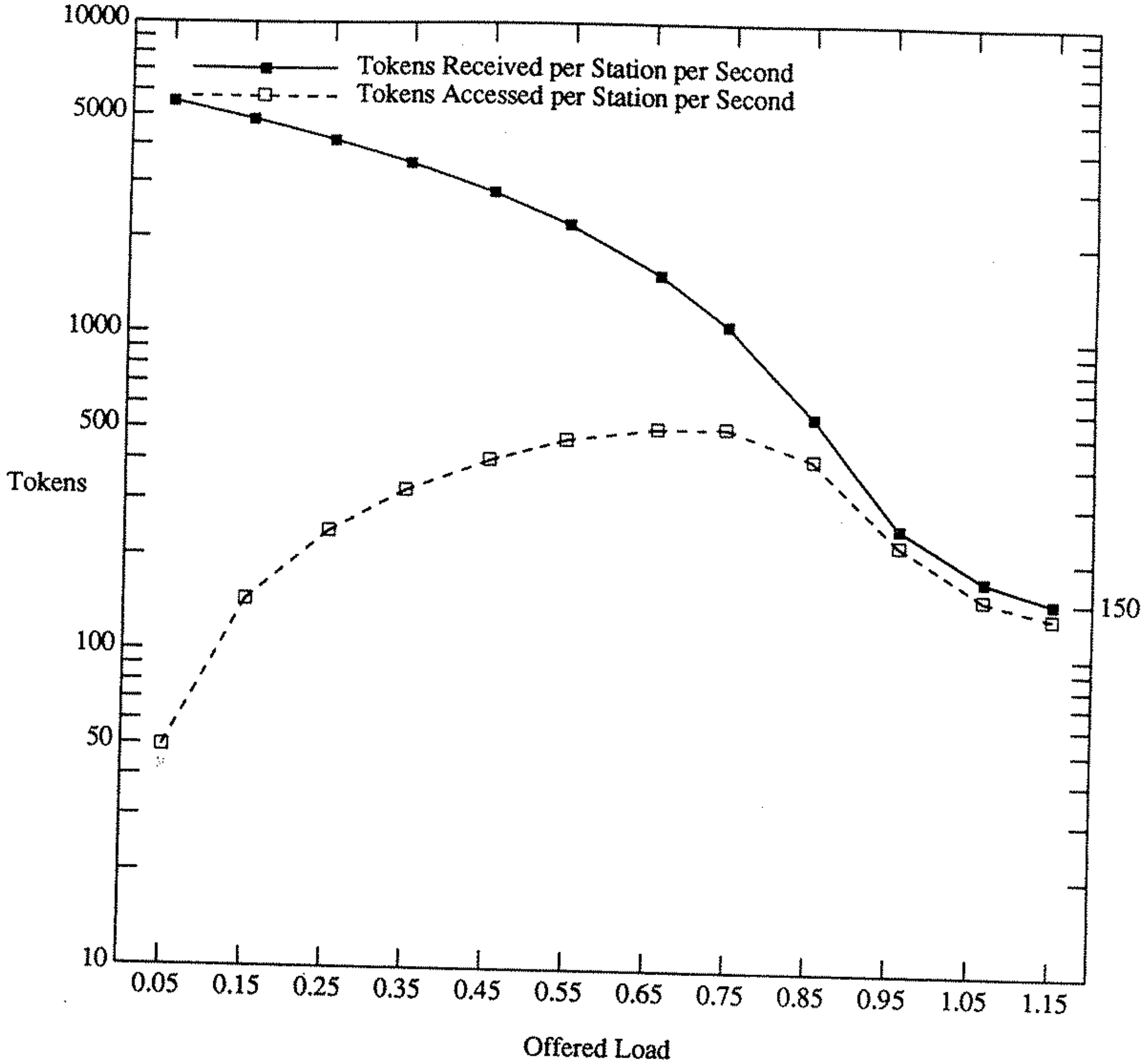
As token cycle time stabilizes, so does the number of tokens each station receives per second. This is why these metrics begin to stabilize at approximately 150 tokens per station per second for the reference configuration.

Figure 4.3 plots the number of tokens accessed per station per second against the number of packets sent per station per second. We have already discussed the tokens accessed curve but notice the difference in the shape of that curve on a linear scale (Figure 4.3) as compared to a logarithmic scale (Figure 4.2).

The number of packets sent per station per second in Figure 4.3 increases linearly until the "timed token" mechanism begins to limit the number of packets each station may transmit per token access. At this point the "packets sent" metric begins to approach a value of 950 packets per station per second.

4.2.1.3. Arrival Queue Length

A station captures a token when it has packets to transmit. Figure 4.4 depicts the mean number of packets waiting when the token arrives at a station. We have previously established that as offered load increases, more packets arrive at the station between token accesses. This is why arrival queue length increases steadily as offered load increases.

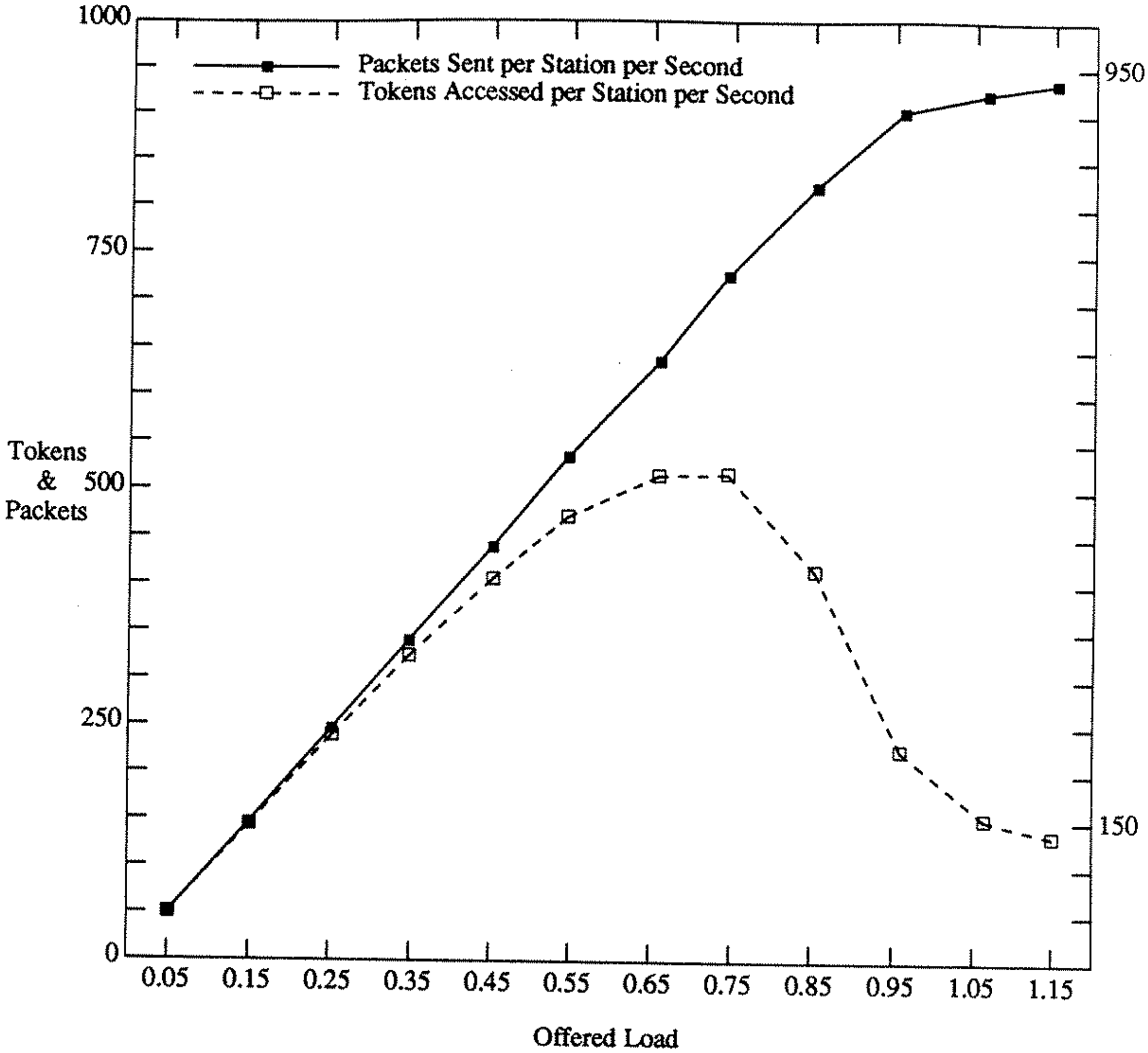


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.2 — Tokens Received and Tokens Accessed

Single Priority Reference Configuration

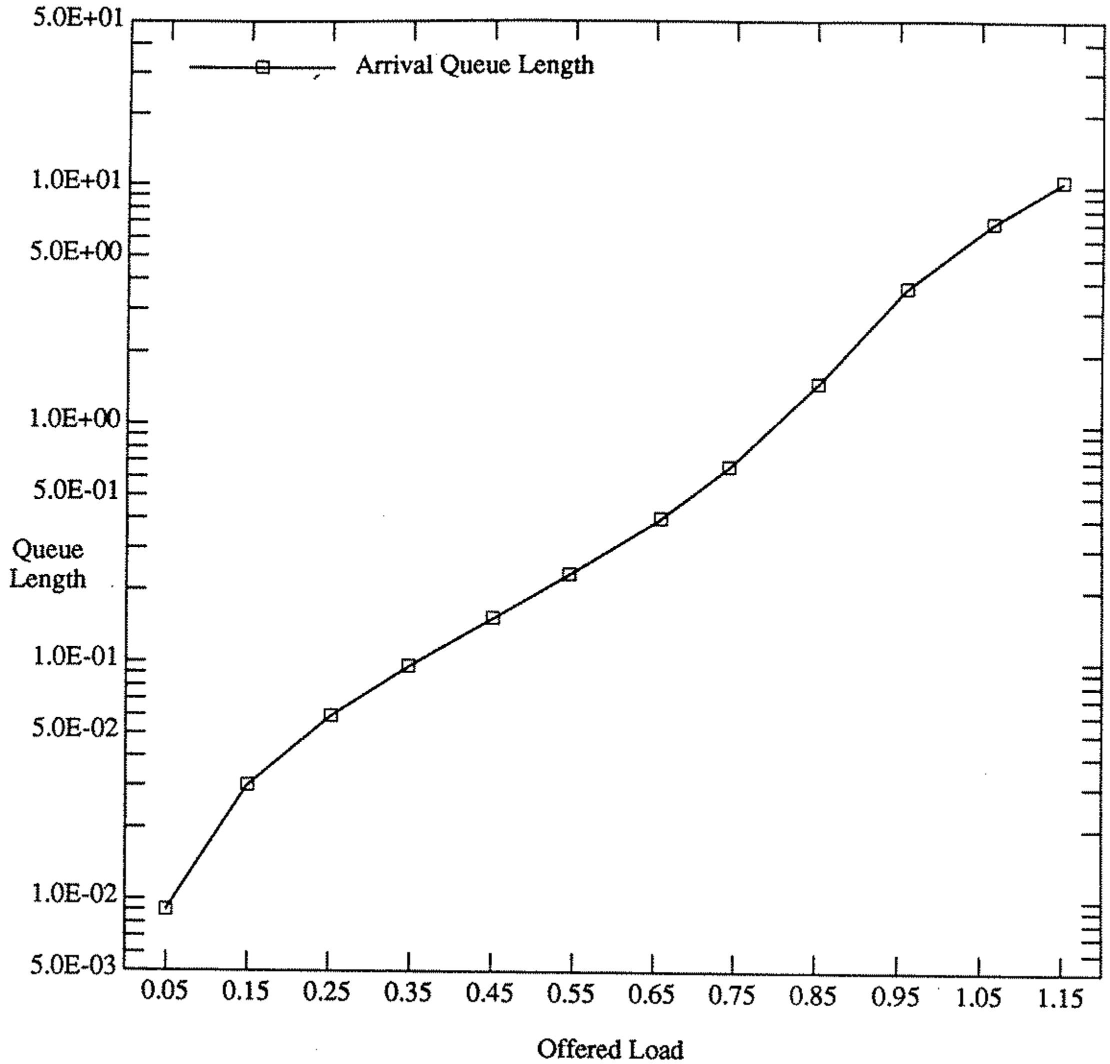


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.3 — Tokens Accessed and Packets Sent

Single Priority Reference Configuration



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.4 — Token Arrival Queue Length

Single Priority Reference Configuration

4.2.1.4. Throughput

Throughput is the number of bits (data plus required framing) transmitted in a given amount of time divided by the medium capacity for that same amount of time. Figure 4.5 shows the throughput of the single priority reference configuration. Throughput rises linearly as offered load is increased and stabilizes when the maximum throughput of the network is reached. For this network configuration the maximum throughput is approximately 96%. The lost 4% is due mainly to token transmission time and related overhead.

4.2.1.5. Packet Delay

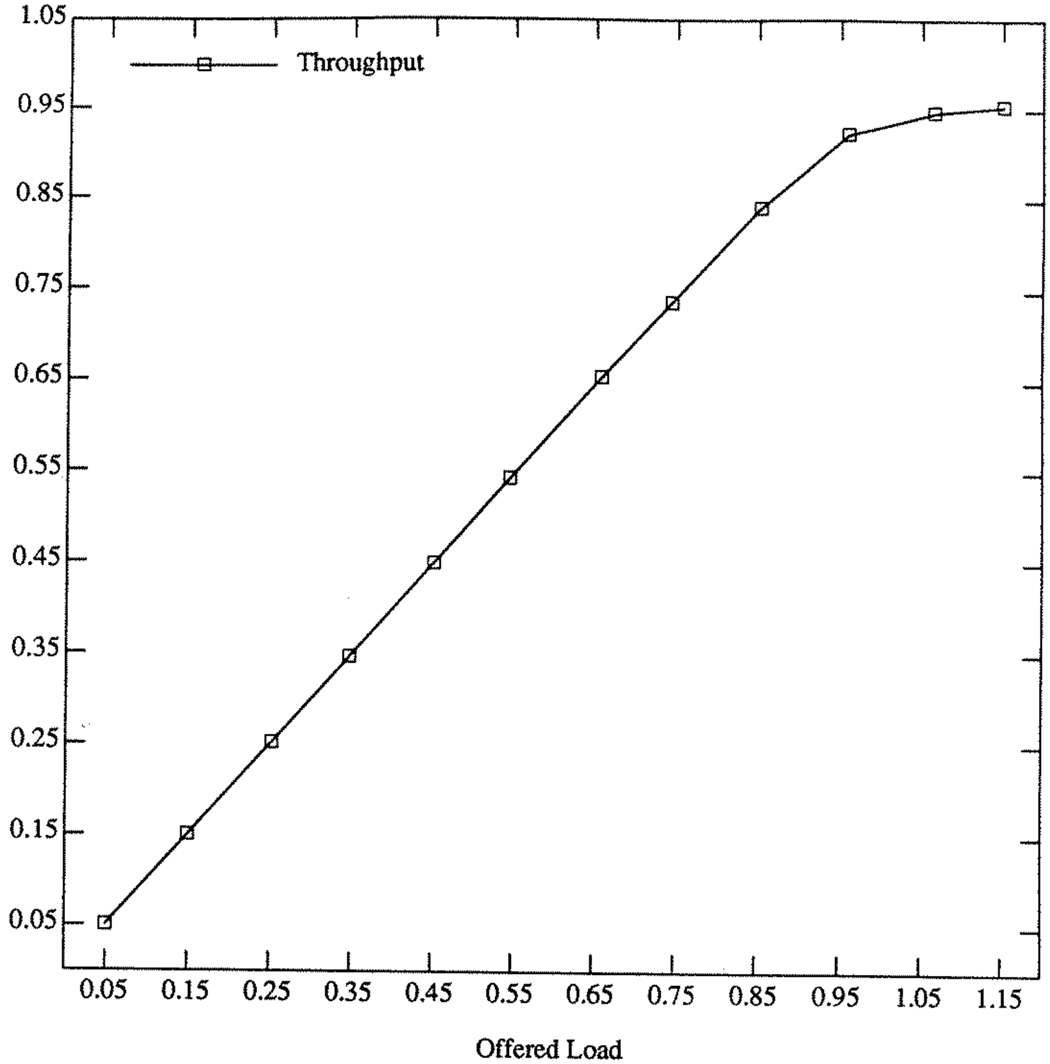
The delay a packet experiences between the time it is enqueued and the time it reaches its destination can be divided into the following four components:

1. Queueing Delay: The time a packet spends in the queue behind other packets.
2. Network Access Delay: The time a packet spends at the head of the queue.
3. Station Delay: The sum of queueing delay and network access delay.
4. Service Delay: The sum of station delay and packet transmission time.

Figure 4.6 depicts these various packet delays. At low offered loads very few packets are generated; therefore, when a packet is enqueued the chances are good that no packets will be ahead of it and network access delay is the major contributor to the service delay of a packet. As offered load increases, packets arrive with greater frequency and this increases the probability that a packet will be enqueued behind packets already in the queue. The result is that as offered load increases, queueing delay contributes more to the service delay of a packet. At approximately 85% offered load queueing delay becomes the major contributor to packet delay. At extremely high offered loads queueing delay accounts for almost all the service delay experienced by a packet.

4.2.2. Effects of Varying Packet Length

The first network parameter of interest is mean packet length. We conducted simulations for mean packet lengths of 32, 64, 128, 256, 512, 1024, and 2048 octets. For these simulations mean packet length is the only network parameter varied.

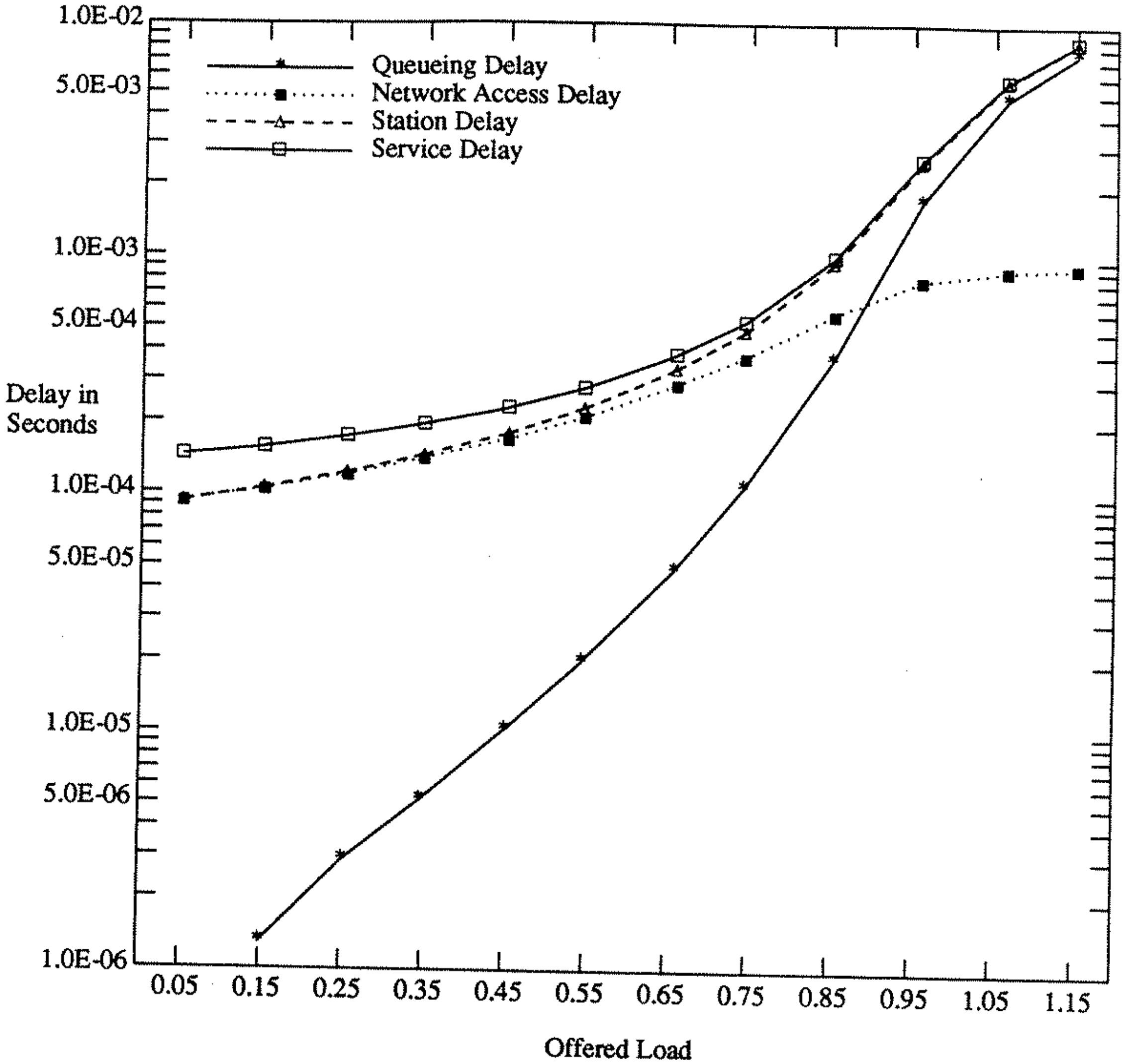


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.5 — Throughput

Single Priority Reference Configuration



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.6 — Various Packet Delays

Single Priority Reference Configuration

4.2.2.1. Token Cycle Time

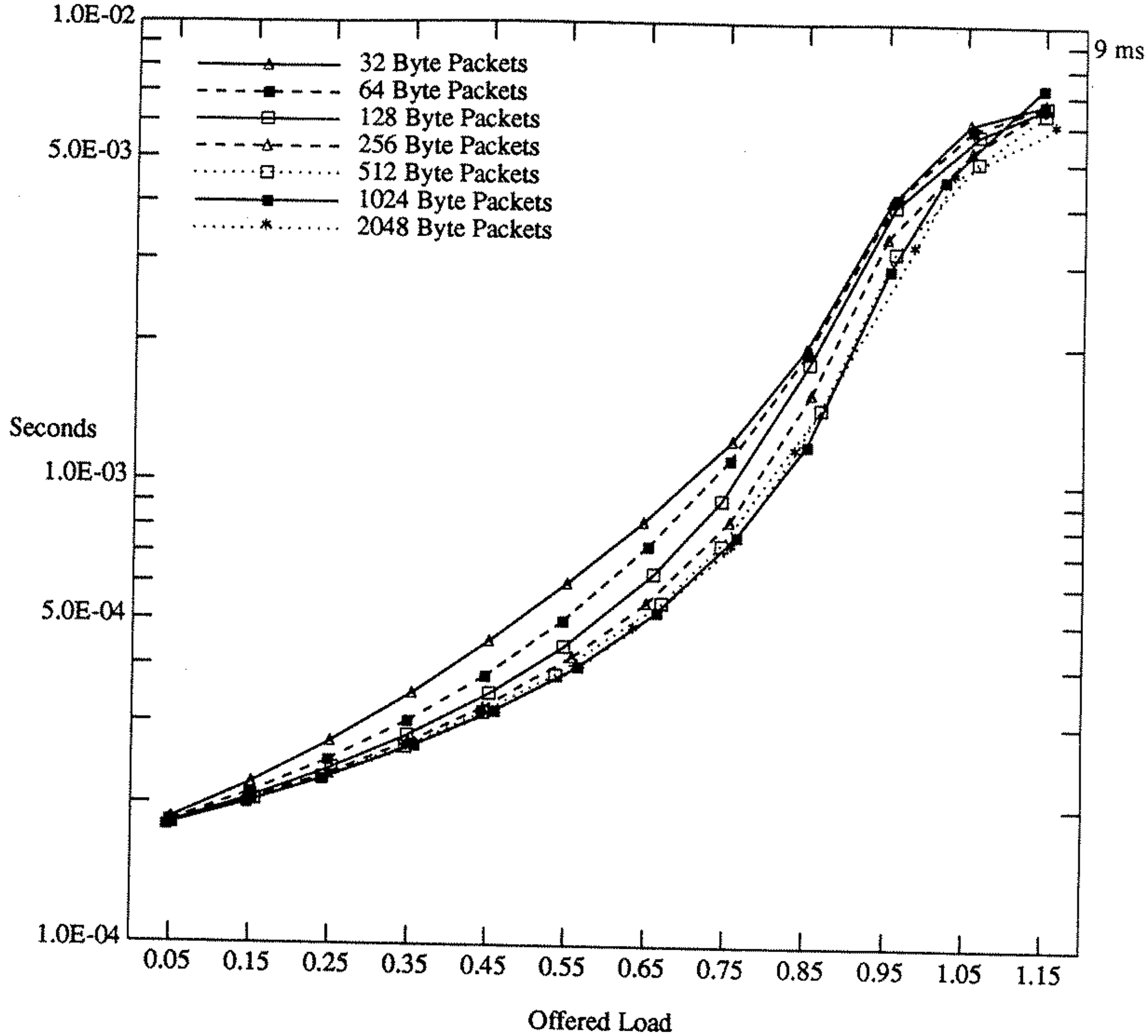
Figure 4.7 shows the effect packet length has on token cycle time. To maintain a certain offered load each station must contribute a certain number of octets to the network traffic. The method used to group these octets is of small consequence. For example, when a token arrives at a station it makes little difference whether two 512-octet packets or sixteen 64-octet packets are enqueued. Each will be serviced in basically the same amount of time and have nearly the same effect on token cycle time.

As we will see in Figure 4.8, small packets over the middle offered loads produce many more token accesses than do large packets. With each token access there is a small amount of overhead. This is why we see small packets having slightly longer token cycle times relative to larger packets over the middle offered loads in Figure 4.7.

4.2.2.2. Token Traffic

Figure 4.8 indicates how variations in packet length affect the number of tokens accessed per station per second. We see from this graph that smaller packets require more token accesses than do larger packets. The reason for this is that as packet size decreases a station must offer more packets to maintain the same offered load. If more packets need to be offered, then the time between packet arrivals will decrease and it follows that the queues will have packets waiting in them more often. If packets are waiting more often, then more tokens will need to be accessed but fewer packets will be serviced per token access.

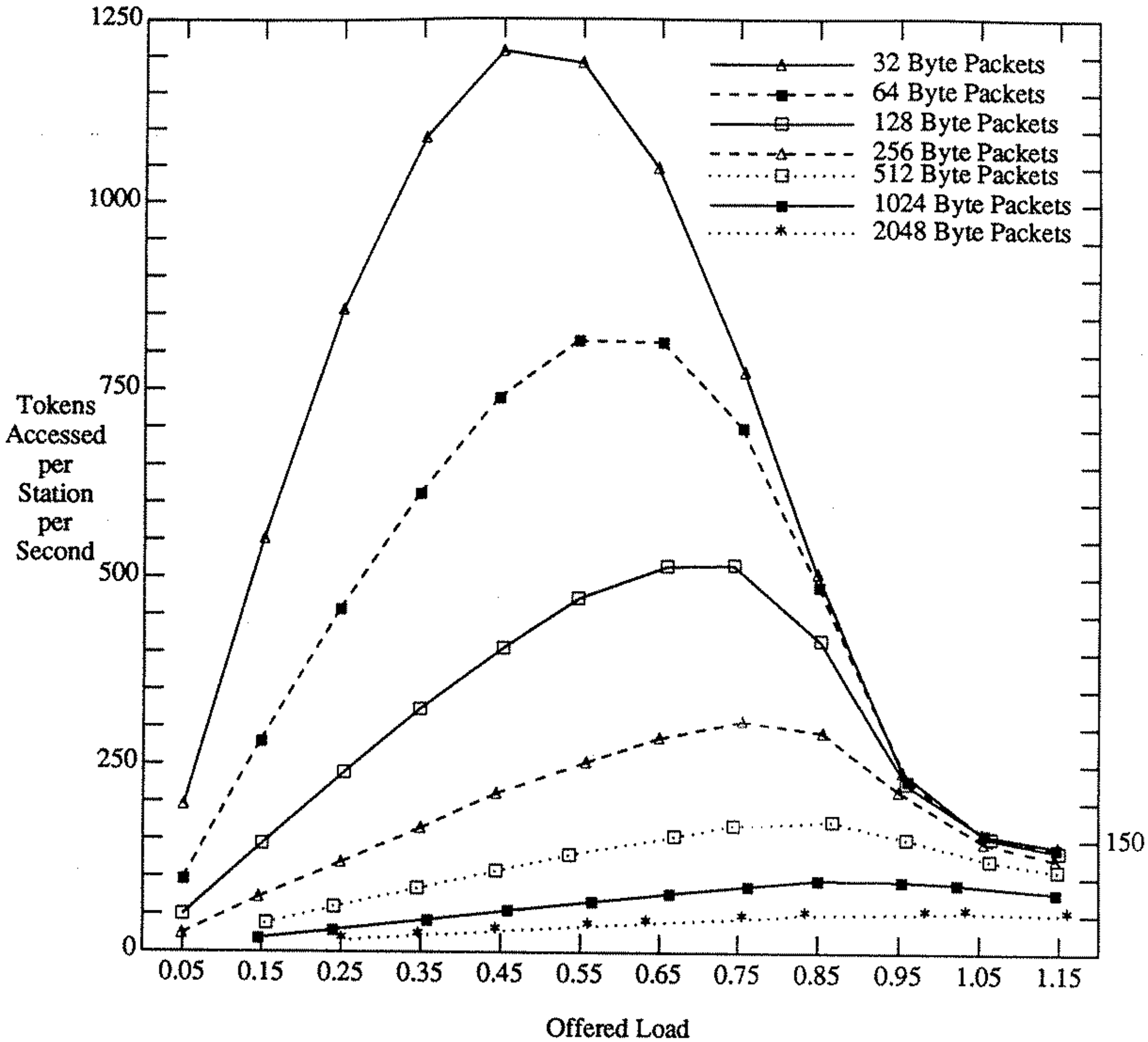
At very high offered loads we again see the number of token accesses begin to stabilize. As before, this occurs because token cycle time has begun to stabilize, and with it, the number of tokens received per station per second has also stabilized. The number of tokens received is the upper bound on the number of tokens accessed and at high offered loads a station accesses almost every token it receives. As a result, these metrics begin to converge.



Configuration:

100 stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
Varying Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.7 — Tokens Cycle Time Varying Packet Length



Configuration:

100 stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
Varying Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.8 — Tokens Accessed Varying Packet Length

4.2.2.3. Arrival Queue Length

Figure 4.9 shows how varying packet lengths affect the number of packets waiting when a token is accessed. As packet size increases the number of packets a station must offer to maintain a certain offered load decreases. As a result, fewer packets arrive between token accesses which is why we see the largest packet lengths producing the shortest arrival queues.

4.2.2.4. Packet Delay

Network access delay is the mean time a packet spends at the head of the arrival queue. Figure 4.10 shows how varying packet length affects this delay metric.

When the token arrives at a station the only packet that has accumulated any network access delay is the first packet in the queue. All packets behind the first packet have no network access delay. Mean network access delay per packet is simply the mean delay incurred by the first packet divided by the mean number of packets serviced per token access. The more packets serviced per token access, the shorter the average network access delay per packet. This is why at high offered loads we see small packets having shorter network access delay relative to large packets.

As offered load increases we eventually reach a point where the maximum number of packets are serviced per token access. When this point is reached network access delay for that particular packet size stabilizes. Figure 4.10 shows that network access delay has stabilized for 32, 64 and 128 octet packets.

4.2.3. Effects of Varying the Number of Stations

The next network parameter of interest is the number of stations on the ring. We conducted simulations for 10, 50, 100, 250, 500, and 1000 stations. One thousand stations is the maximum number of stations FDDI allows. In these simulations the number of stations is the only network parameter varied.

4.2.3.1. Token Cycle Time

In Figure 4.11 we see how variations in the number of stations affect token cycle time. As stations are added to the ring it follows that the token will be accessed more times per token cycle and this will increase token cycle time. This upward trend will continue until token cycle time begins to approach the target token rotation time.

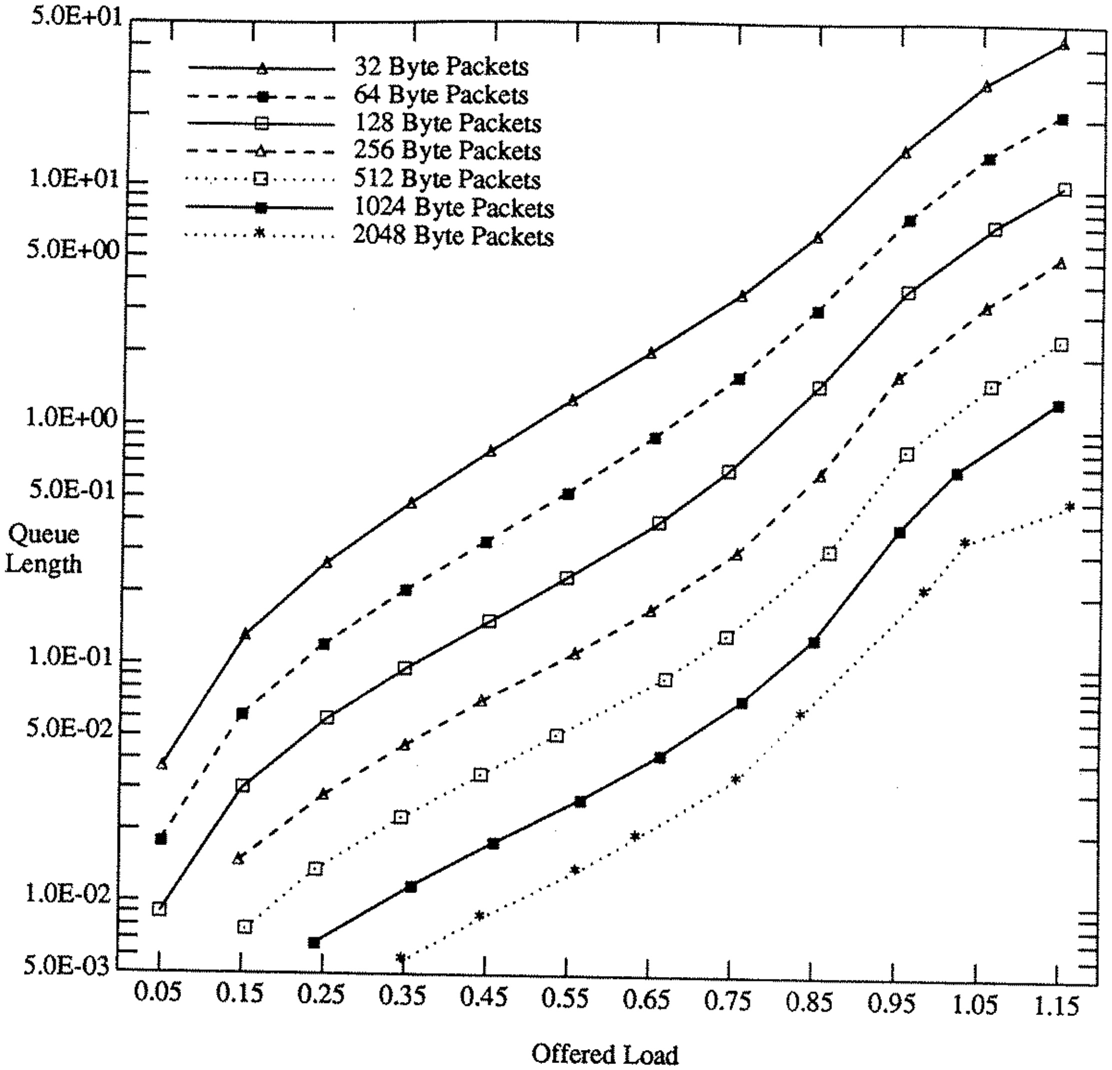
One interesting result presented in this graph is the token cycle time for a ring of 1000 stations. The target token rotation time for this network configuration is 9 ms but token cycle time has slightly exceeded that value. The reason for this is that FDDI allows a station to begin transmission of a packet even if the token hold timer will expire before transmission of that packet is complete. Because of this a station will occasionally release the token late. As the number of stations on the ring increases, it follows that the number of late token releases will also increase. Since all traffic for this configuration is synchronous, there is no asynchronous service to degrade. Consequently, it is difficult to make up for late token releases because at high offered loads each station uses nearly all the bandwidth available to it. An accumulation of late token releases at high offered load results in token cycle time exceeding its target. This accumulated delay is bounded, however. Sevcik and Johnson show in [SEVC85] that the maximum token cycle time is twice the target token rotation time.

4.2.3.2. Throughput

Figure 4.12 shows that increasing the number of stations on the ring results in decreased maximum throughput. The reason this happens is that as the number of stations increases, the token must spend less time at each station per token cycle to maintain the target token rotation time. As a result more bandwidth is consumed by token transmission and related overhead and less is available for packet transmission.

4.2.3.3. Arrival Queue Length

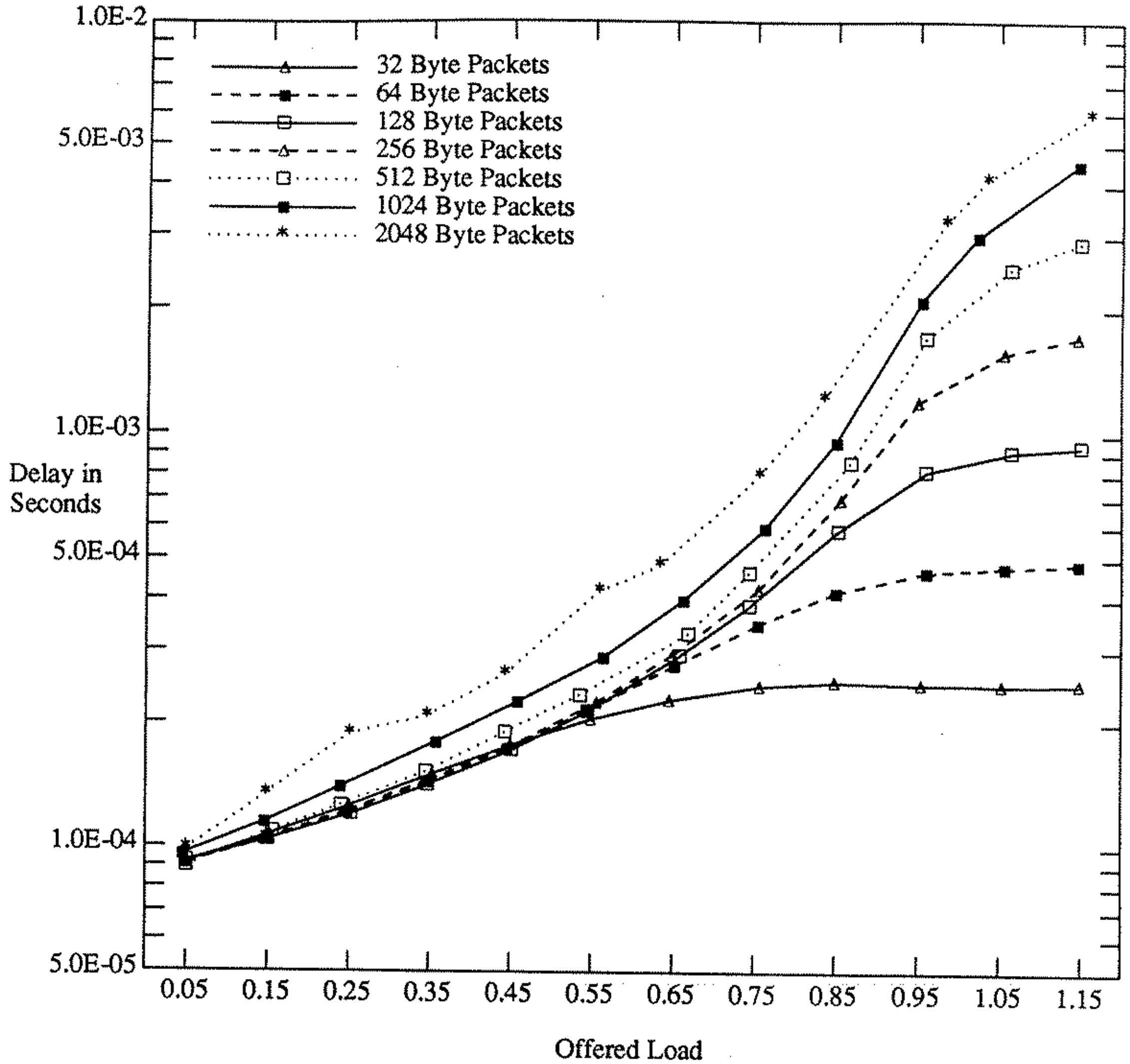
To maintain a certain offered load the network must offer a certain number of packets for transmission. This total number of packets is distributed evenly over all stations on the network. As the number of stations on the network goes down, the number of packets offered by each station must go up. Figure 4.13 shows that having fewer stations results in longer arrival queue lengths. The reason for this is that each station is offering more packets so more packets arrive between token accesses.



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

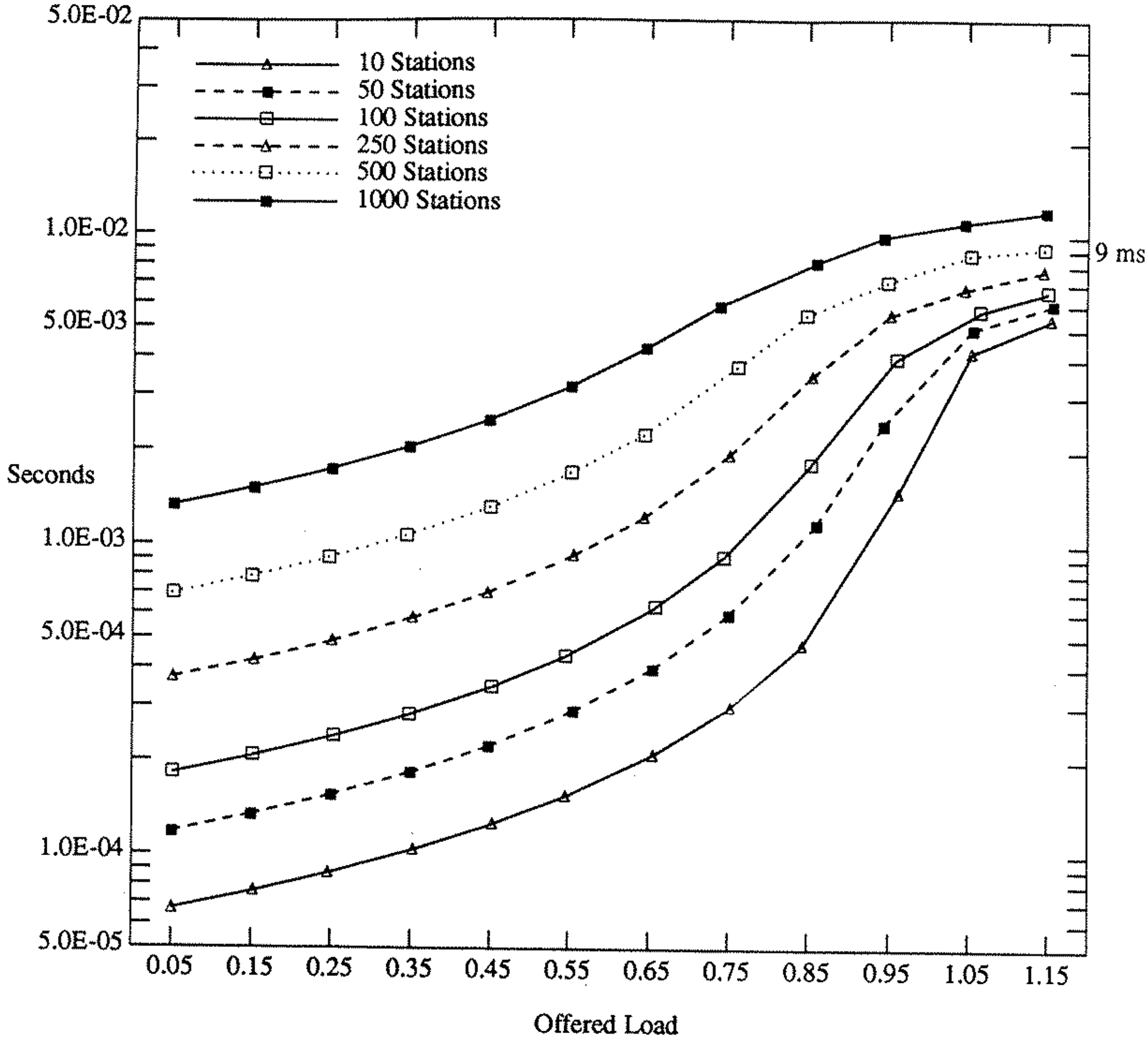
Figure 4.9 — Arrival Queue Lengths Varying Packet Length



Configuration:

100 stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
Varying Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

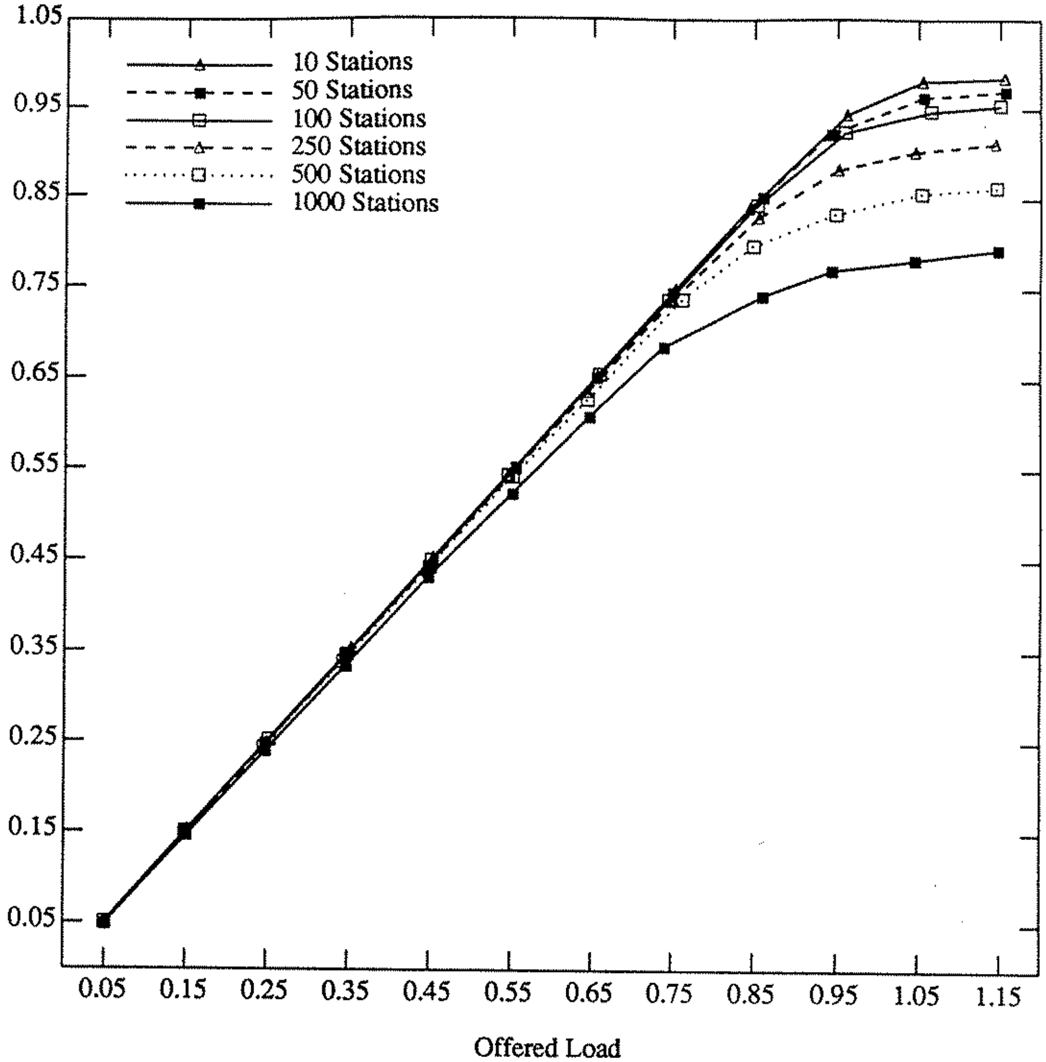
Figure 4.10 — Network Access Delay Varying Packet Length



Configuration:

Varying the Number of Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

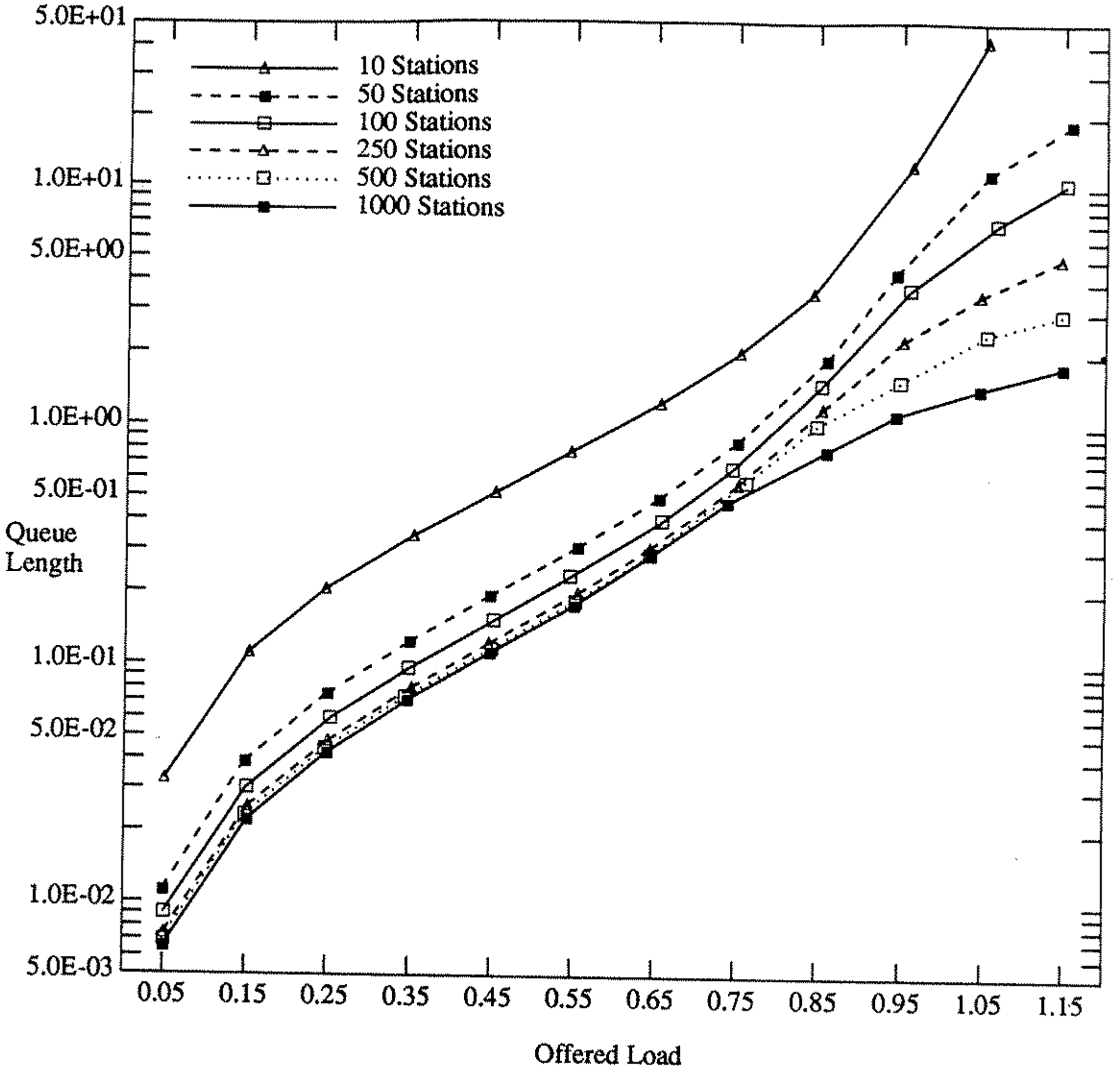
Figure 4.11 — Token Cycle Time Varying the Number of Stations



Configuration:

Varying the Number of Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.12 — Throughput Varying the Number of Stations



Configuration:

Varying the Number of Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.13 — Arrival Queue Lengths Varying the Number of Stations

4.2.3.4. Packet Delay

Figure 4.14 shows how varying the number of stations on the ring affects mean packet service delay. As stations are added to the ring, the mean number of stations a packet must go through to get to its destination (intermediate stations) will also increase. At each intermediate station a packet experiences a small amount of delay in the form of internal station latency. The more intermediate stations there are, the greater the packet's transmission time will be. The result is that service delay increases as the number of stations on the ring increases.

As offered load increases, a packet's queueing delay becomes so large that the delay incurred due to internal station latency becomes negligible. As a result we see the distance between the curves in Figure 4.14 decreases as offered load increases.

4.2.4. Effects of Varying Ring Circumference

The next network parameter of interest is ring circumference. We conducted simulations for 1, 10 and 100 km ring circumferences. Due to the dual counter-rotating ring, the effective ring circumference doubles when a link breaks and the ring goes into loopback mode. To ensure that FDDI's maximum ring circumference (200 km) is not exceeded when a break occurs, the maximum circumference for each of the dual rings is 100 km. For these simulations the only parameter varied is ring circumference.

4.2.4.1. Token Cycle Time and Throughput

Propagation delay is the amount of time it takes a bit to travel from one station to the next. Since stations on the ring are assumed to be equidistant from each other, propagation delay is proportional to ring circumference. The larger the ring, the greater the distance between stations.

Figure 4.15 shows the effect ring circumference has on token cycle time. As expected, the larger the ring, the longer it takes the token to make one complete cycle. The reason for this is that as the ring gets larger, propagation delay grows and it simply takes the token more time to travel between stations.

From Figure 4.16 we see that a larger ring circumference causes maximum throughput to drop. The reason for this is that as the distance between stations grows, the network has more time to wait between token accesses. This additional waiting time consumes bandwidth that would otherwise be used for packet transmission. The result is a reduction to maximum throughput.

4.2.4.2. Packet Delay

Figure 4.17 shows how varying ring circumference affects mean packet service delay. From this graph we see that a larger ring causes service delay to increase. The reason for this is simply that as the ring grows, packets have farther to travel. As a result, the amount of delay a packet experiences will increase as the ring gets larger. It also follows that as the ring gets larger, packet transmission time will account for a greater portion of total packet delay.

4.3. Multiple Priority Operation

FDDI allows one synchronous priority and up to eight asynchronous priorities. The next section of simulation results presents the performance of FDDI when multiple priority levels are used.

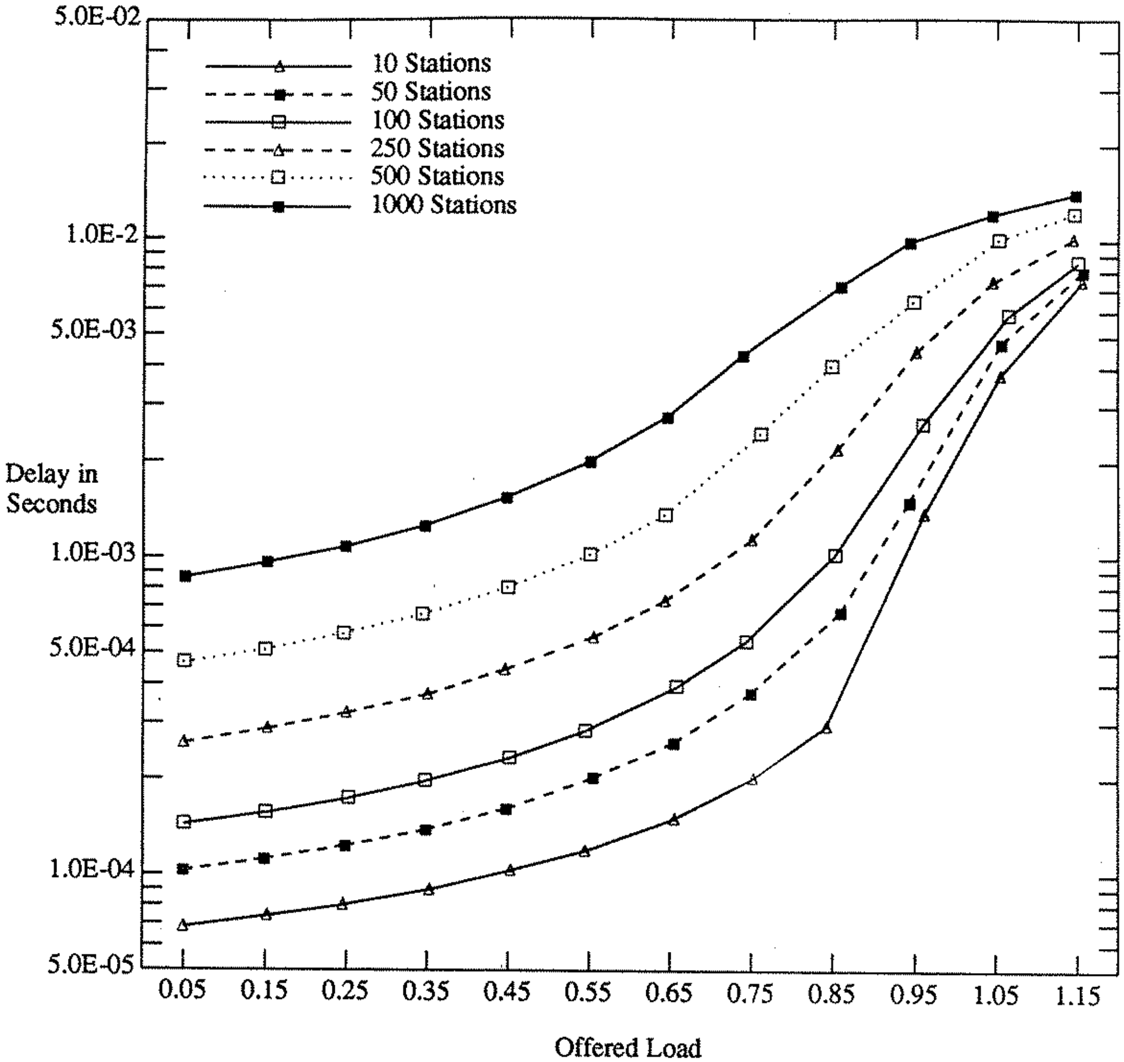
4.3.1. Reference Configuration

The reference configuration for multiple priority operation has the following characteristics:

100 stations evenly distributed on the ring
100 Mbps medium rate
10 km ring circumference
32 bit internal station latency
a synchronous priority and eight asynchronous priorities
9.0 ms target token rotation time
asynchronous priority thresholds held constant at 9.0 ms
exponential packet arrival distribution
1024 bit mean packet length distributed exponentially
no restricted token dialog
error free transmission

As before, the motivation for this configuration was simply that it provide a reasonable configuration to which variations can be made and that target token rotation time is long enough for the network to provide exhaustive service at high offered loads.

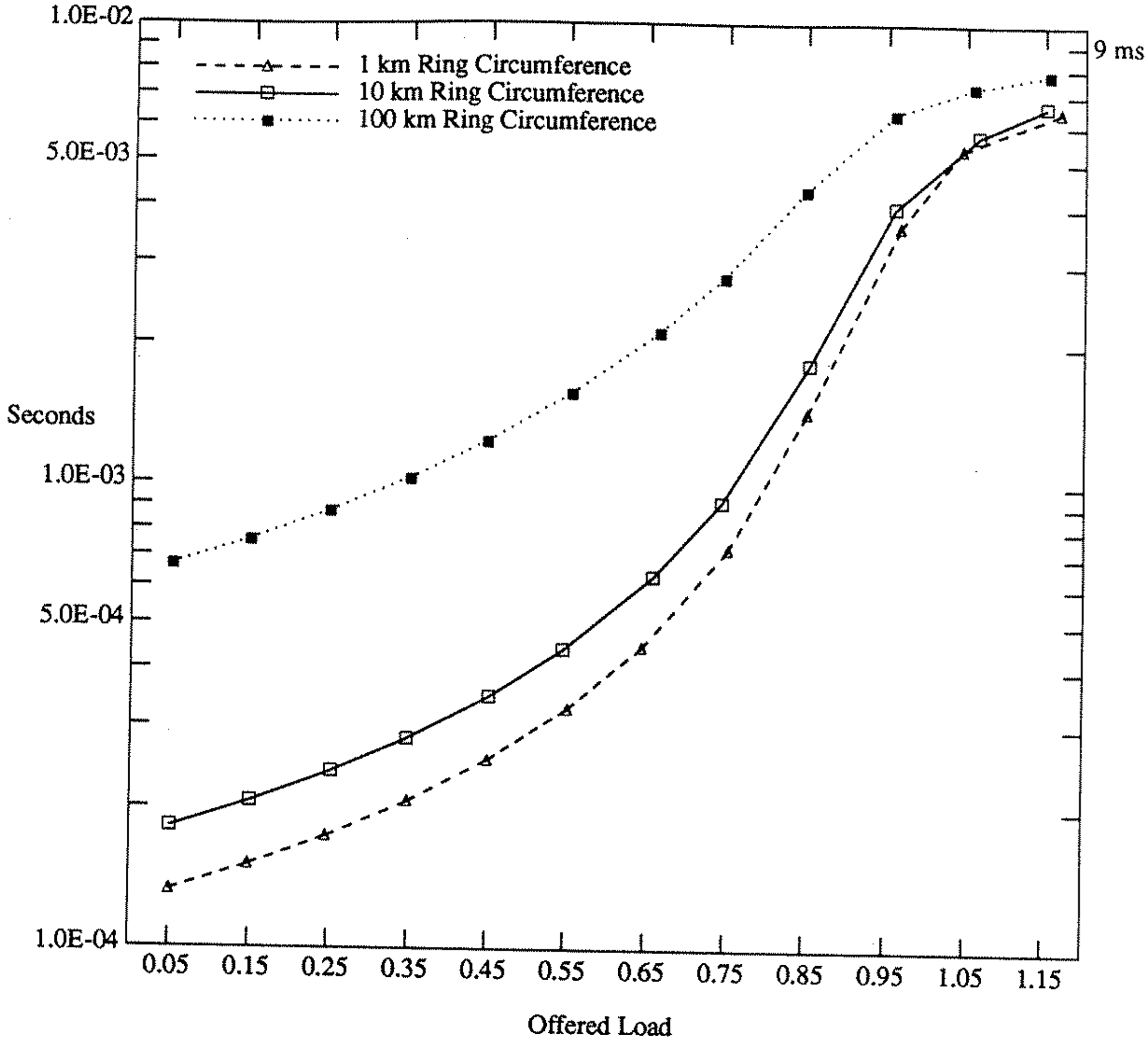
It is important to note that the asynchronous priority threshold values are all set equal to the target token rotation time; by doing this we have placed no artificial limitations on the asynchronous priorities. The only differentiation between the priorities is the order in which they are serviced.



Configuration:

Varying the Number of Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

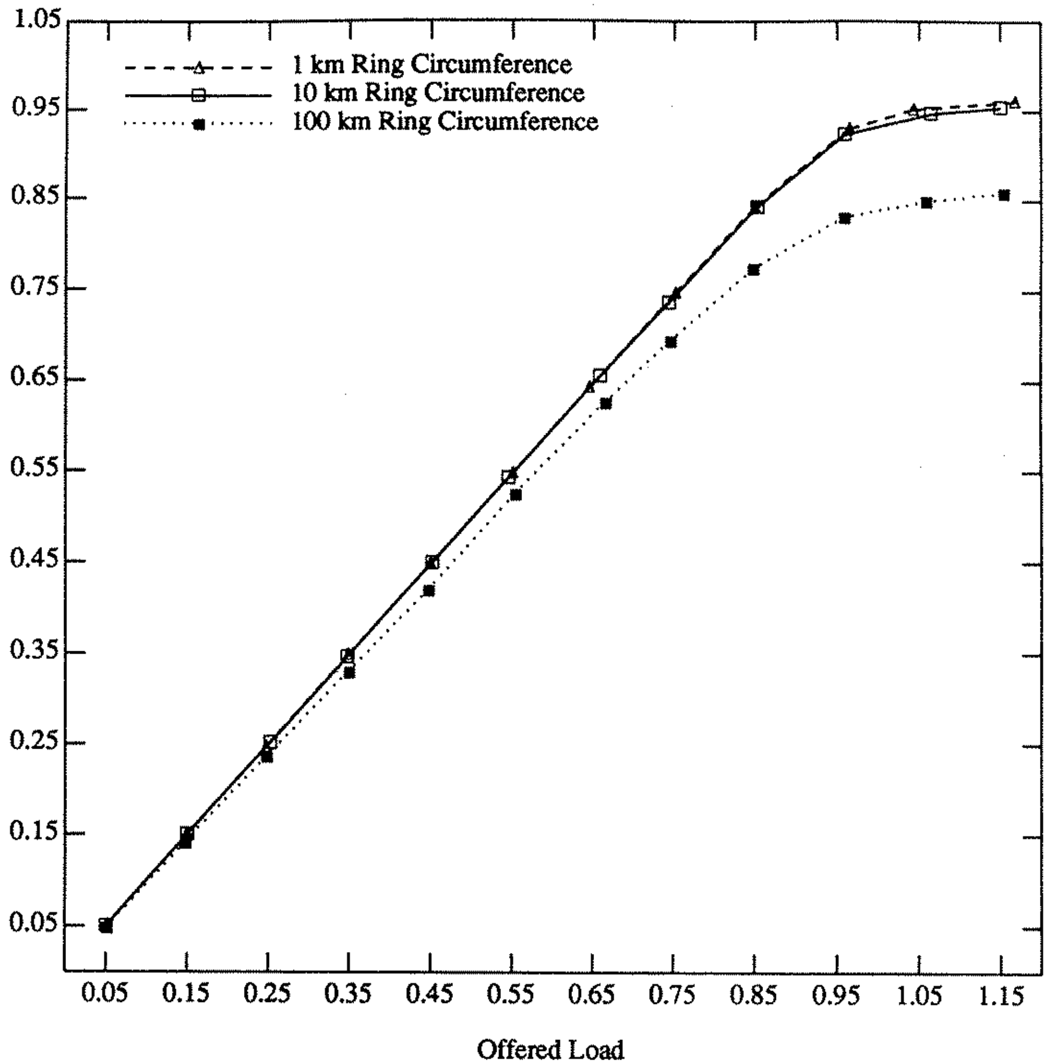
Figure 4.14 — Service Delay Varying the Number of Stations



Configuration:

100 Stations
100 Mbps Medium Rate
Varying Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

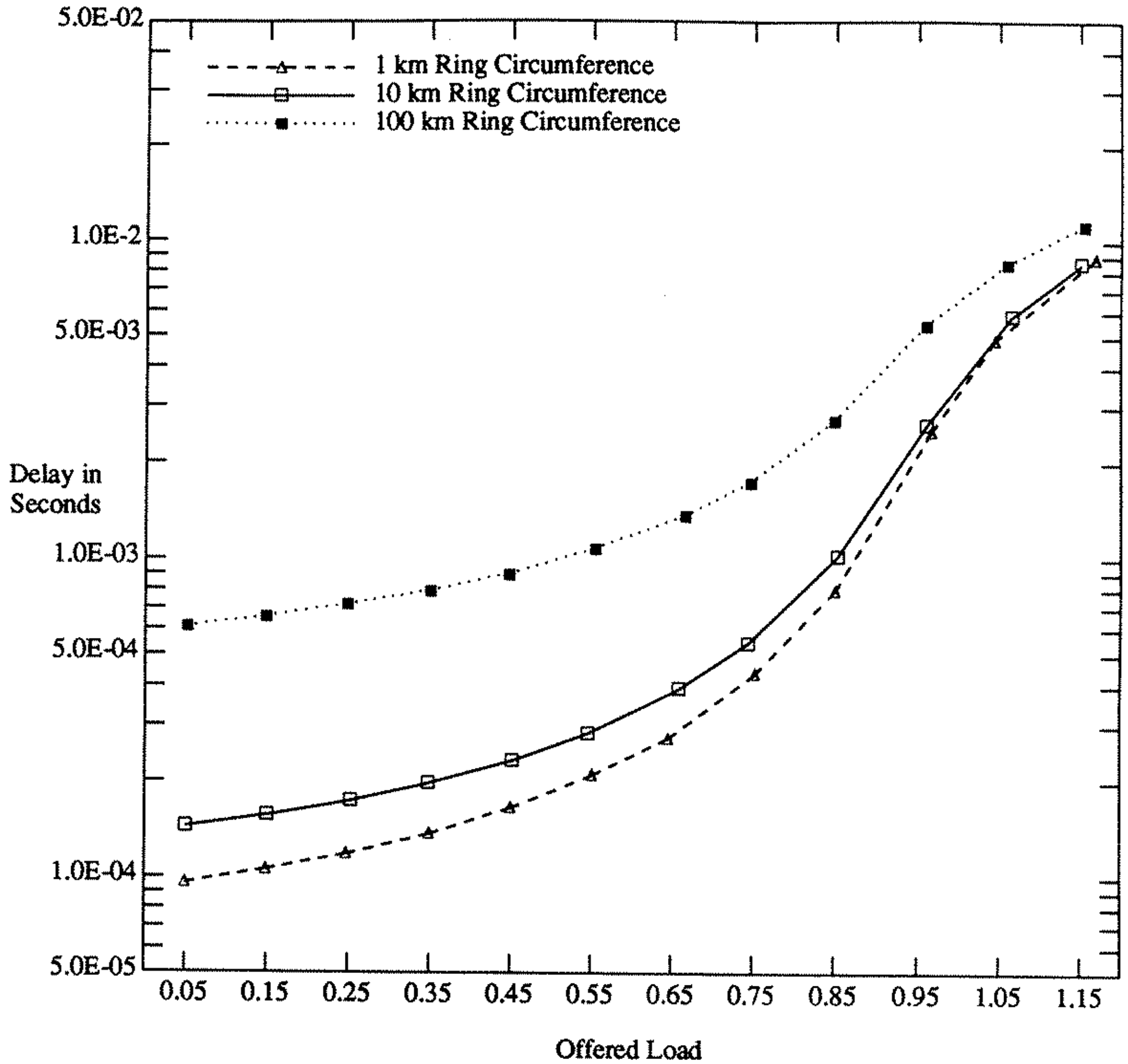
Figure 4.15 — Token Cycle Time Varying Ring Circumference



Configuration:

100 Stations
100 Mbps Medium Rate
Varying Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.16 — Throughput Varying Ring Circumference



Configuration:

100 Stations
100 Mbps Medium Rate
Varying Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority
9.0 ms Target Token Rotation Time
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.17 — Service Delay Varying Ring Circumference

4.3.1.1. Token Cycle Time and Throughput

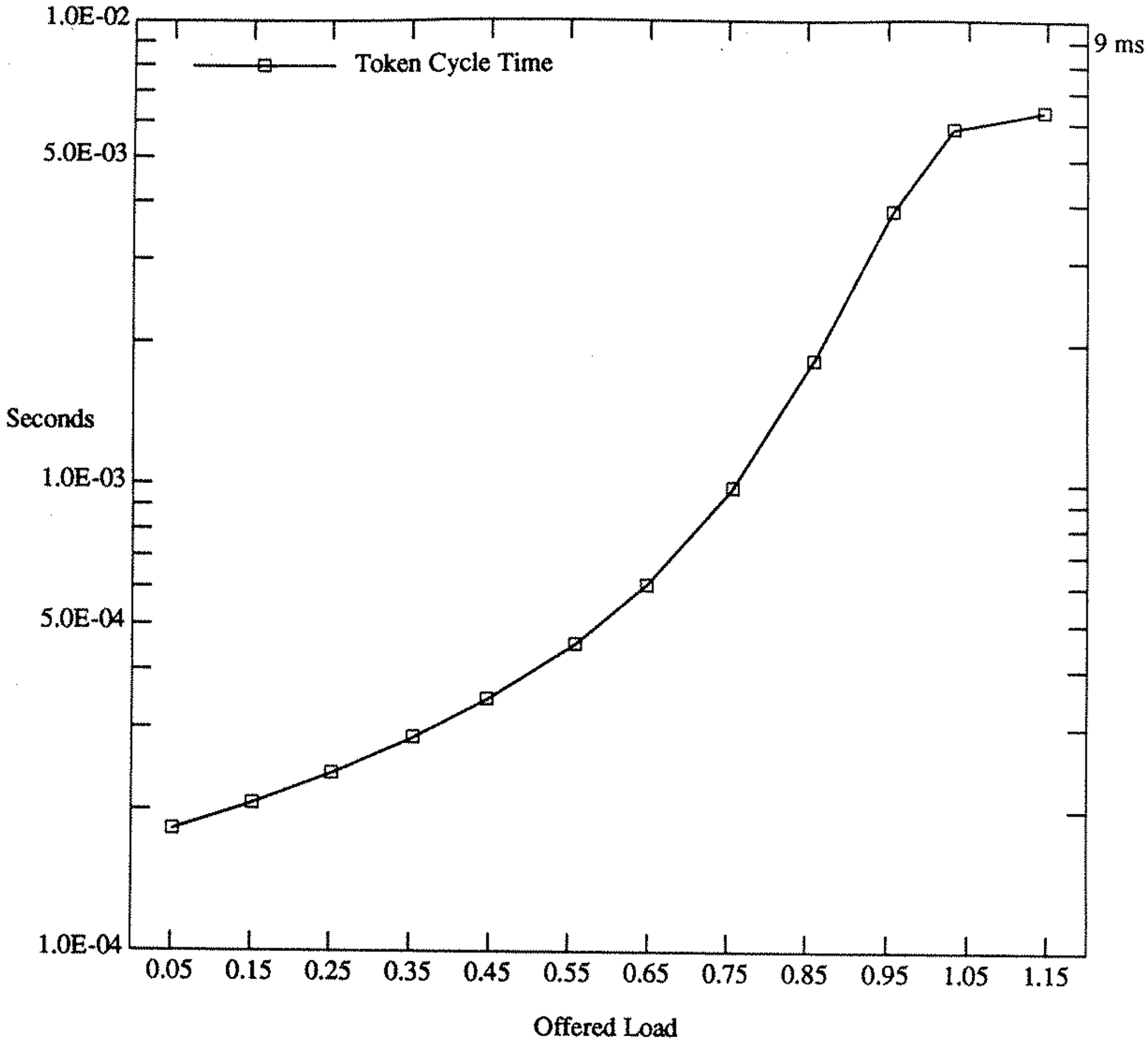
Figures 4.18 and 4.19 depict token cycle time and throughput respectively for the multiple priority reference configuration. These graphs are included to show that multiple priorities yield performance similar to that of a single priority. The graphs produced by the multiple priority reference configuration were so similar to those produced by the single priority reference configuration that inclusion of both sets would be redundant.

The similarity between the reference configurations is due primarily to the fact that no artificial restrictions were placed on the asynchronous priorities. Packet arrivals are assumed to be independent of the priority scheme implemented by the stations. In the first reference configuration all packets were offered at the same priority. In the second reference configuration packet arrivals are evenly distributed among the synchronous and eight asynchronous priorities. Since no artificial restrictions are placed on the asynchronous priorities, all are eligible to transmit packets until the station exhausts its synchronous bandwidth. As a result, asynchronous traffic receives service virtually identical to that of synchronous traffic.

Differences in the performance of the reference configurations would occur if synchronous traffic were heavy enough to consume nearly all the bandwidth available to a station. If this were the case then the need to maintain target token rotation time would result in asynchronous traffic going unserviced.

4.3.1.2. Packet Delay

Figure 4.20 shows service delay for the synchronous and three asynchronous priorities in the multiple priority reference configuration. From this graph we see that all priorities receive virtually the same delay until offered load exceeds the maximum throughput of this configuration. Only then do we see the lower priorities having significantly more delay than the higher priorities. The reason for this is that before offered load reached the maximum throughput there was enough bandwidth available to service all arriving packets. After maximum throughput is reached, there is no longer enough bandwidth available to service all arriving packets, so the packets in the lower priorities have to wait longer.

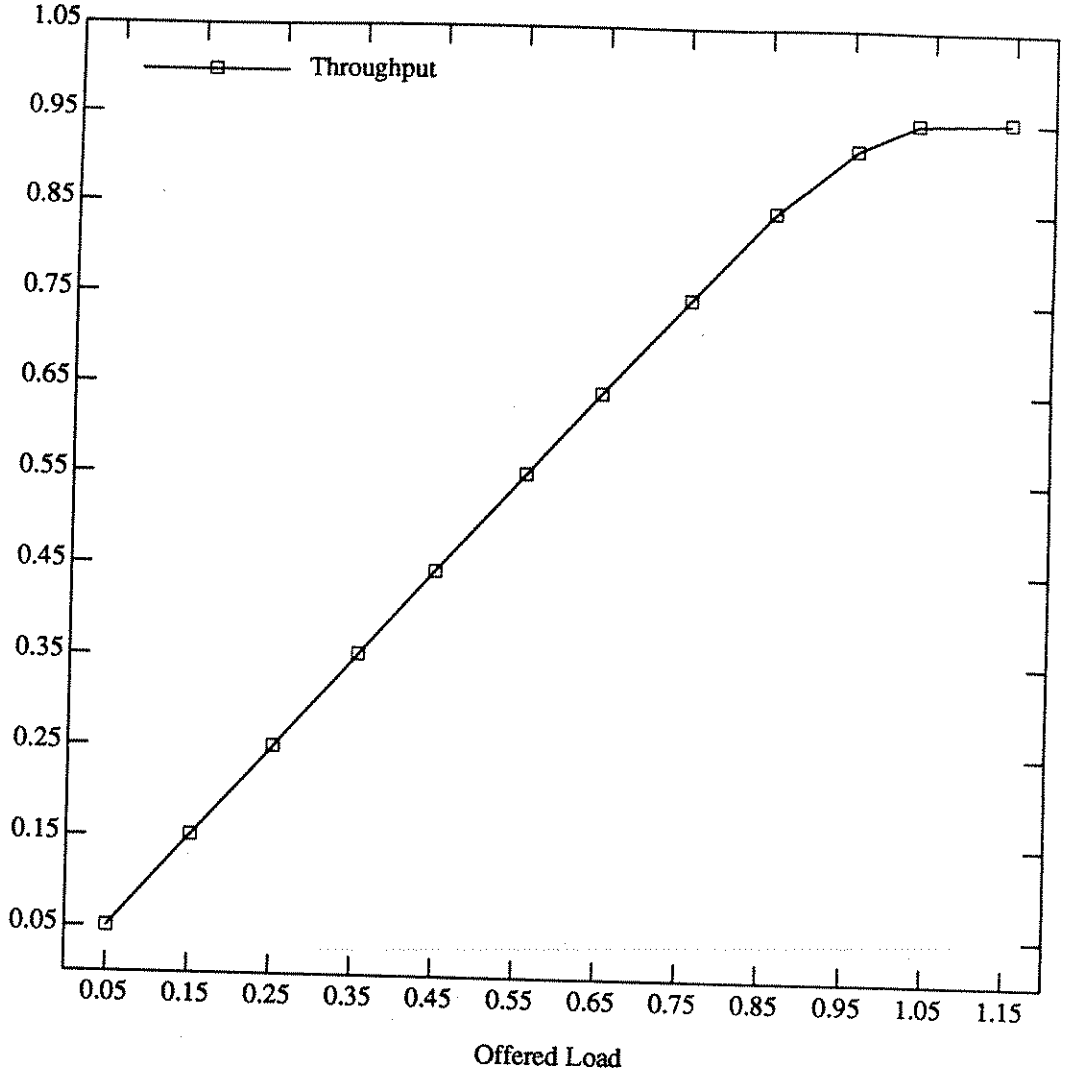


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.18 — Token Cycle Time

Multiple Priority Reference Configuration

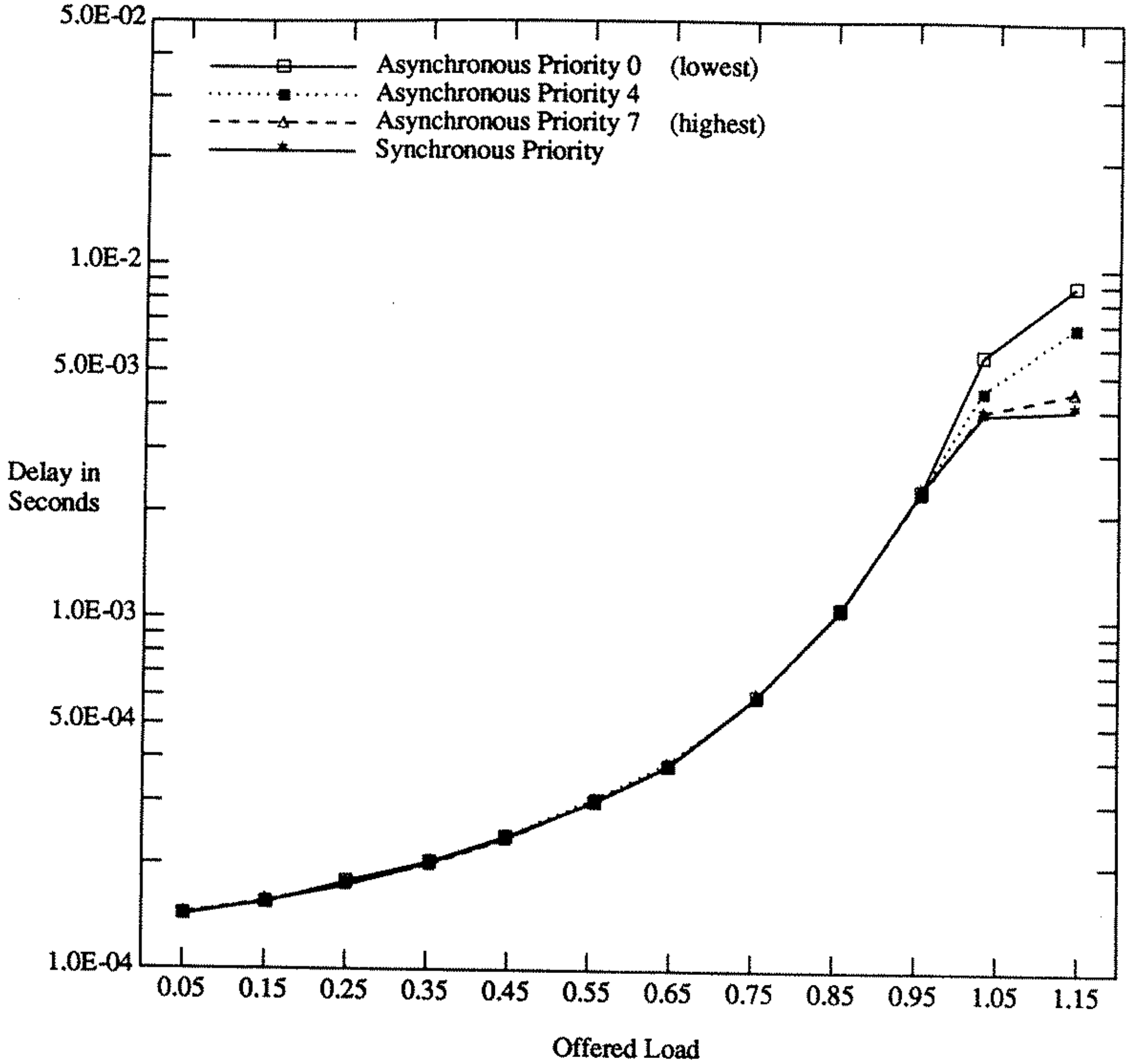


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.19 — Throughput

Multiple Priority Reference Configuration



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.20 — Synchronous And Asynchronous Service Delay

Multiple Priority Reference Configuration

4.3.2. Effects of Varying Asynchronous Priority Thresholds

Asynchronous priority thresholds are the means by which FDDI artificially throttles asynchronous traffic (see Section 2.6.2.3.2). The way they work is that the lowest asynchronous priority gets the lowest priority threshold value and the highest asynchronous priority gets the highest priority threshold value. A packet of a certain priority may only be transmitted if the station's token hold timer is less than the associated priority threshold value. If priority thresholds are not used (as in the multiple priority reference configuration), their default value is the target token rotation time.

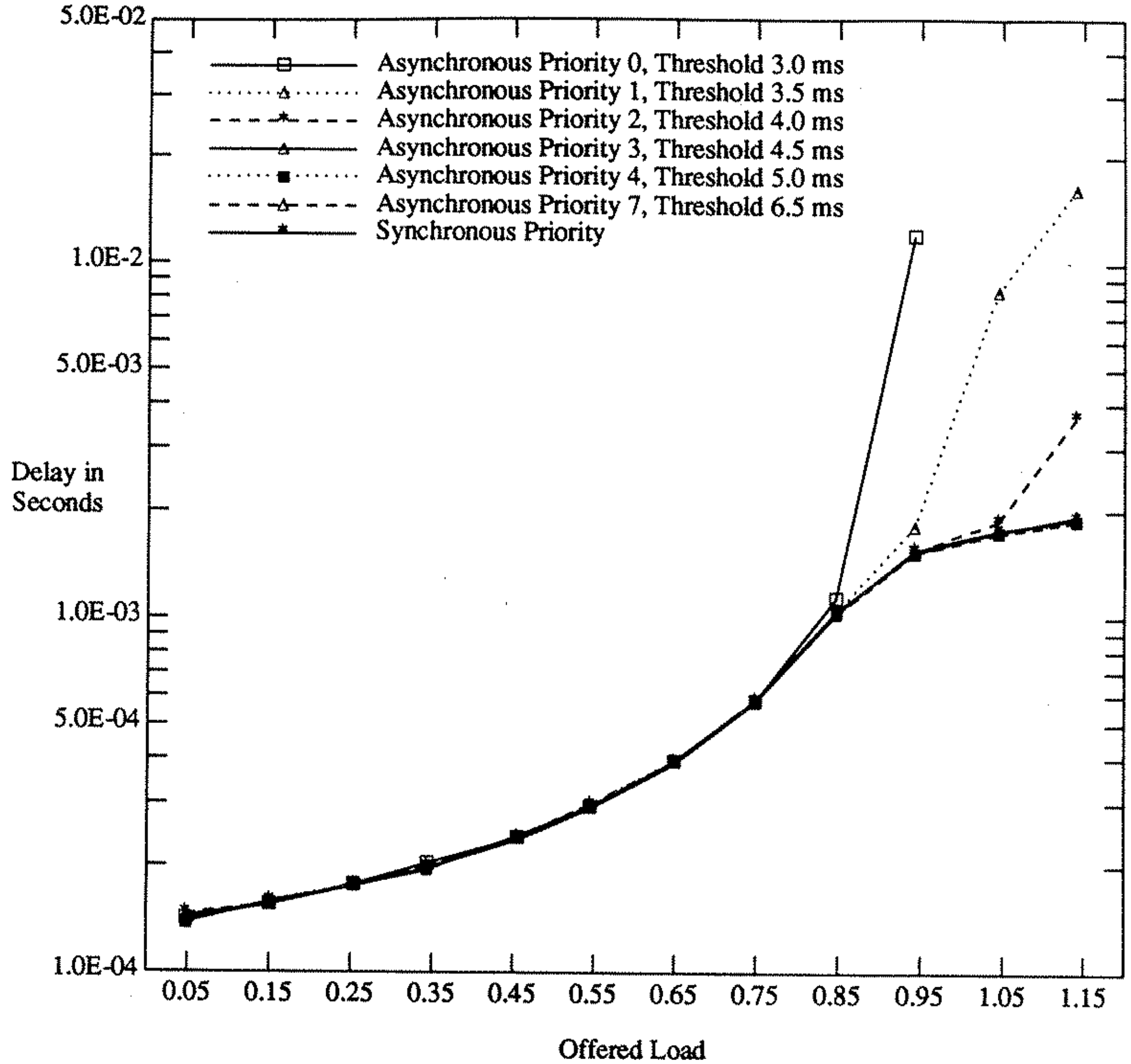
We conducted simulations for priority thresholds held constant at the target token rotation time, varied from 3.0 ms to 6.5 ms by 0.5 ms, and varied from 1.0 ms to 4.5 ms by 0.5 ms.

4.3.2.1. Packet Service Delay

Figures 4.21 and 4.22 show how reducing priority thresholds affects mean packet service delay. Figure 4.21 shows service delays with priority threshold values varied from 3.0 ms to 6.5 ms by 0.5 ms. At low offered loads all priorities receive adequate service. As offered load increases we see service to the lower asynchronous priorities begin to cut off. At 95% offered load the service delay at the lowest asynchronous priority has increased dramatically. The same thing happens to priority one at 105% offered load and to priority two at 115%.

The reason for this is that asynchronous priorities are serviced in order from highest (asynchronous priority 7) to lowest (asynchronous priority 0). By the time packets in asynchronous priority 0 are eligible for service at 95% offered load, the station's token hold timer has advanced to the point where it is greater than priority zero's threshold value. As a result, service to asynchronous priority zero is virtually cut off.

Figure 4.22 shows a similar phenomenon, when the priority thresholds are varied from 1.0 ms to 4.5 ms by 0.5 ms. The difference here is that service cut off to lower priorities begins earlier. On this graph we see service to priorities zero, one and two completely cut off and service to priority three beginning to cut off as offered load increases.

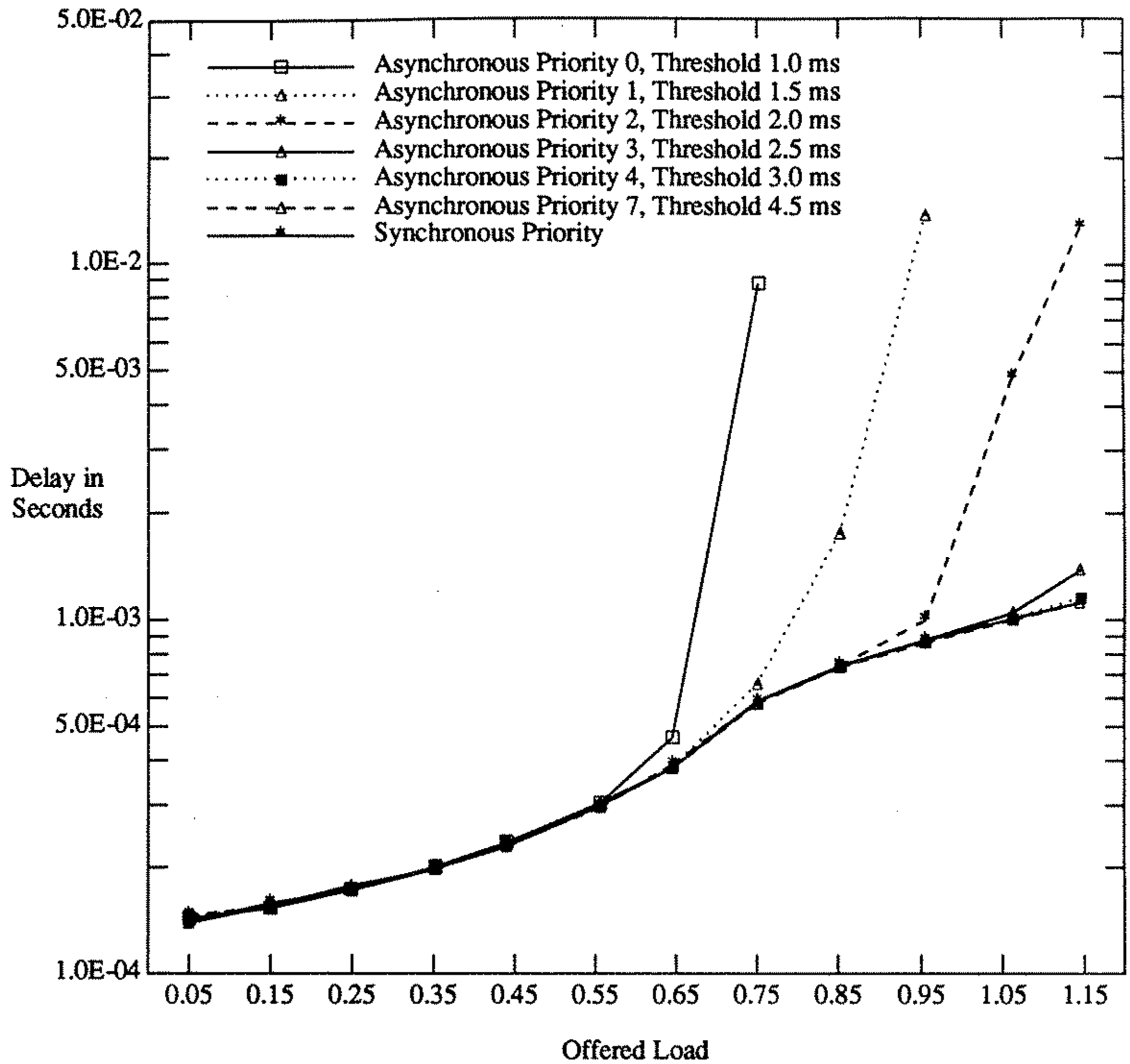


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Varied From 3.0 ms to 6.5 ms by 0.5 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.21 — Synchronous And Asynchronous Service Delay

Priority Thresholds Varied From 3.0 ms to 6.5 ms by 0.5 ms



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Varied From 1.0 ms to 4.5 ms by 0.5 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.22 — Synchronous And Asynchronous Service Delay

Priority Thresholds Varied From 1.0 ms to 4.5 ms by 0.5 ms

4.3.2.2. Token Cycle Time

Figure 4.23 shows how varying priority thresholds affects token cycle time. We saw in Figures 4.21 and 4.22 that as offered load increases, priority thresholds begin to cut off service to the lower asynchronous priorities. When this happens the token finds fewer packets per station that are eligible for transmission, so the station releases the token earlier than it normally would if no priority thresholds were in use. The lower the threshold values are set, the fewer packets the token has to service relative to the number it usually services at that offered load. The result is that lowering priority thresholds causes token cycle time to decrease relative to the reference configuration.

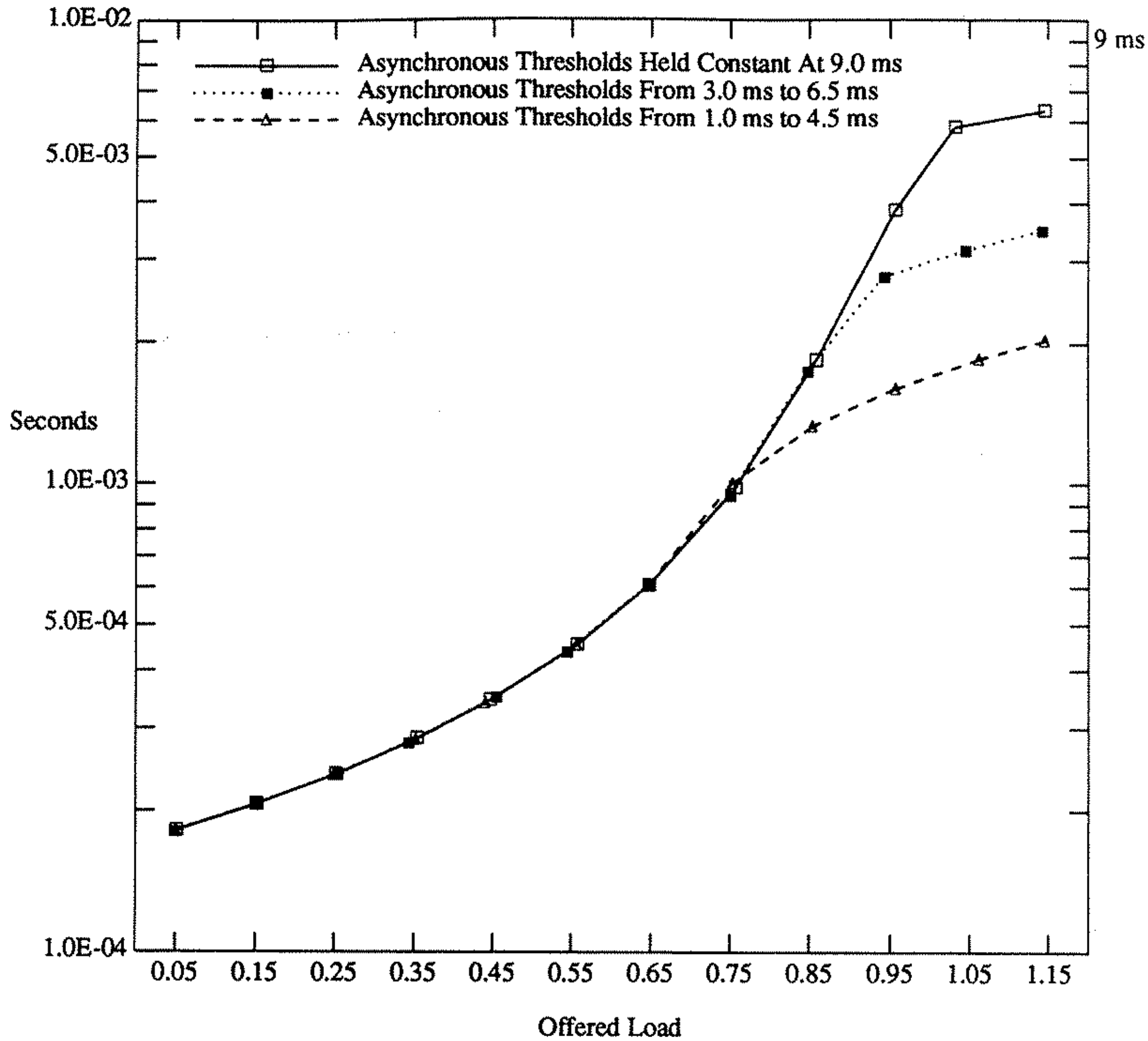
4.3.2.3. Throughput

Lower token cycle time implies that more token transmissions are taking place in a given amount of time. The effect, which can be see in Figure 4.24, is that more bandwidth is consumed by token transmission, resulting in a lower maximum throughput relative to the reference configuration.

4.3.3. Effects of Lowering Target Token Rotation Time

Target token rotation time is a network-wide value negotiated by the stations at ring initilization. It represents the longest token cycle time usable by the station with the most time critical traffic. In our reference configurations, the target token rotation time was set at 9 ms to ensure that the ring would receive exhaustive service even at high offered load. The asynchronous priority thresholds were also set at this value to ensure that the asynchronous priorities were not artificially restricted.

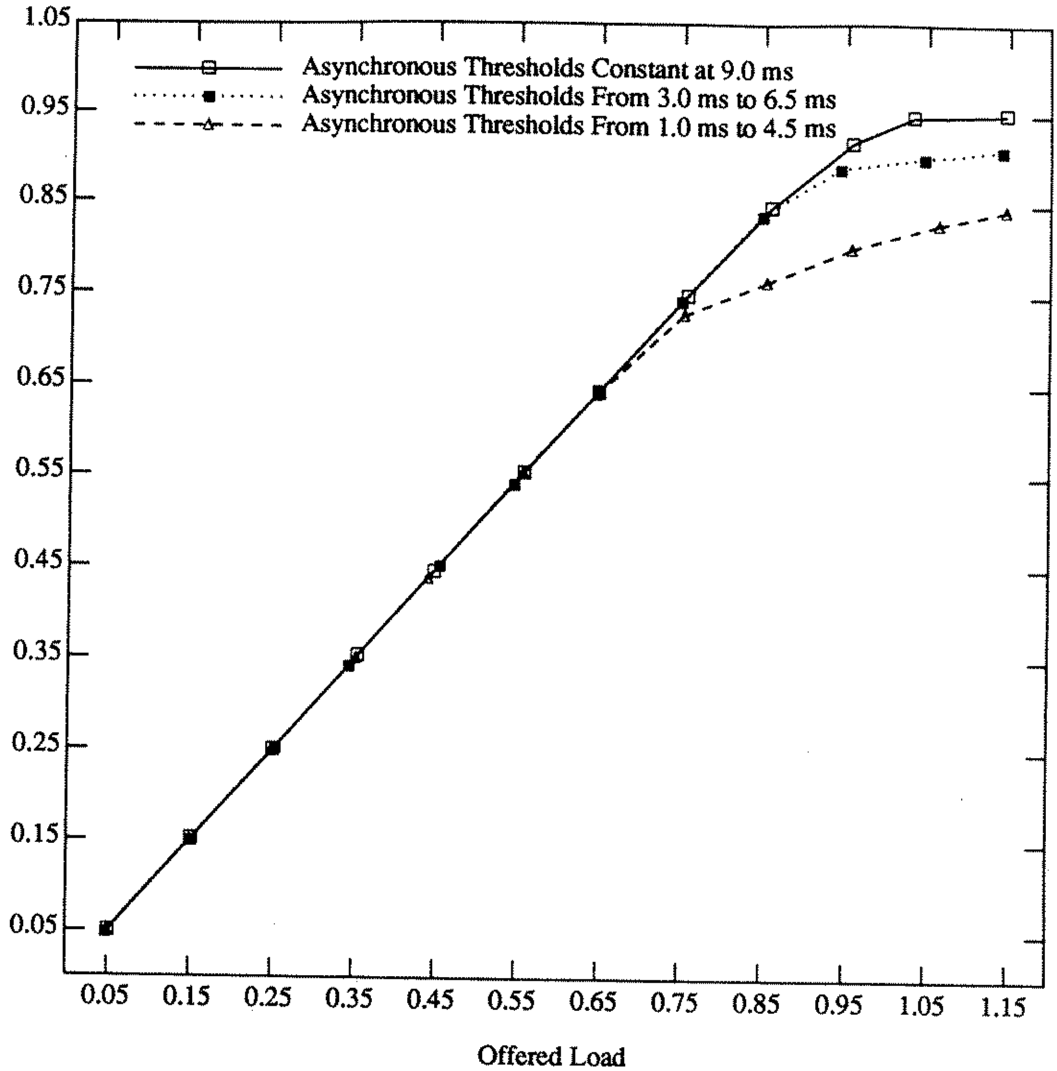
We conducted a series of simulations with the target token rotation time and all asynchronous priority thresholds set to 1 ms. The motivation for this value was that it provided exhaustive service at low offered loads only. In these simulations the target token rotation time and asynchronous priority thresholds were the only parameters varied.



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Varied From 3.0 ms to 6.5 ms
And From 1.0 ms to 4.5 ms by 0.5 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.23 — Token Cycle Time Varying Priority Thresholds



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Varied From 3.0 ms to 6.5 ms
And From 1.0 ms to 4.5 ms by 0.5 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.24 — Throughput Varying Priority Thresholds

4.3.3.1. Token Cycle Time

Figure 4.25 plots the token cycle time of the multiple priority reference configuration against the token cycle time of a network with 1 ms target token rotation time. This graph shows that lowering target token rotation time causes the token cycle time to stabilize at a lower offered load. The reason for this is that lowering the target token rotation time reduces the maximum time the token may spend at each station. The result is that fewer packets need to be waiting when the token arrives to cause it to stay as long as the target token rotation time will allow. As we see in Figure 4.25, the network configuration with a 1 ms target token rotation time reached this limit at approximately 75% offered load.

Because token cycle time is so important to network performance, FDDI monitors the token's progress at each station. When the token arrives late at a given station (i.e., token cycle time is greater than the target token rotation time), that station may not use the token for its asynchronous traffic. In this way, token cycle time is bounded by its target value. As a result, token cycle time rises steadily until it reaches the target token rotation time. The late token arrival mechanism then takes over and causes token cycle time to stabilize quickly.

4.3.3.2. Throughput

Figure 4.26 plots the throughput of the multiple priority reference configuration against the throughput of a network with a 1 ms target token rotation time. This graph shows that lowering the target token rotation time decreases maximum throughput relative to the reference configuration.

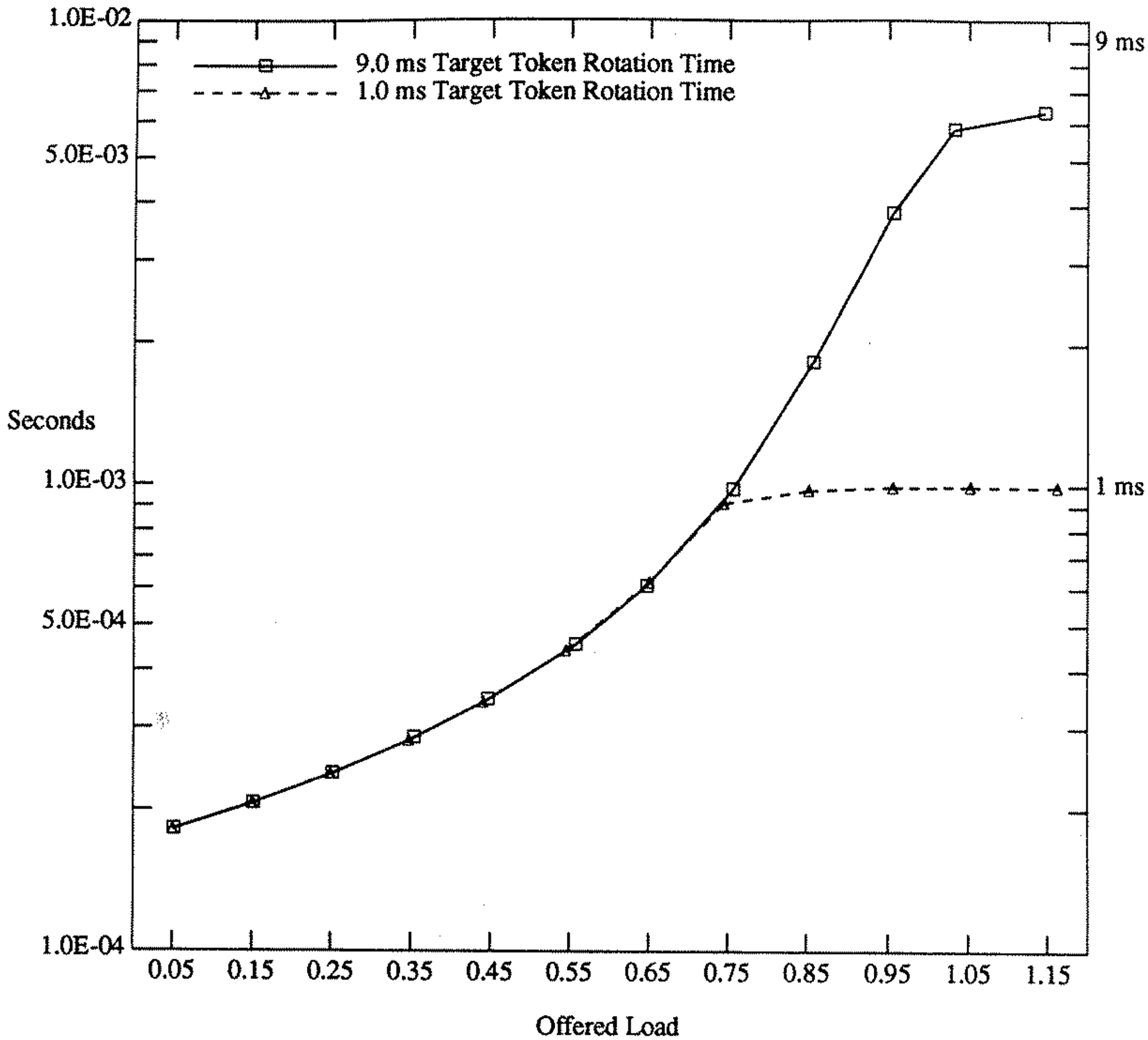
As we saw in Figure 4.25, lowering target token rotation time caused token cycle time to decrease relative to the reference configuration. A lower token cycle time means that more token transmissions are taking place in a given amount of time. Token transmissions and the related overhead consume bandwidth that would otherwise be used for packet transmission. The result is that lowering the target token rotation time will in general lower the maximum throughput.

An effect of FDDI's token cycle time monitoring can be seen by comparing Figures 4.25 and 4.26. At 75% offered load, both network configurations have nearly the same token cycle times, which implies that nearly the same number of token transmissions are taking place. However, throughput for each of the networks is significantly different at 75% offered load. The reason for this is that at 75% offered load, tokens in the network with a 1 ms target token rotation time begin to arrive late. This results in fewer packets being sent per token access, which is why we see a reduction in throughput. At 85% offered load, token cycle time for the network with a 1 ms target token rotation time has risen slightly, resulting in more late tokens and lower throughput as compared to 75% offered load. Throughput eventually stabilizes when token cycle time stabilizes.

4.3.3.3. Packet Delay

Figure 4.27 shows mean packet delay for the synchronous and three asynchronous priorities in a network with a 1 ms target token rotation time. As offered load increases, mean service delay for all priorities remains comparable until the target token rotation time begins to limit token cycle time. At that point (75% offered load), tokens begin to arrive late and mean service delay for all asynchronous traffic is uniformly higher. The reason for the uniform increase is that late token arrivals temporarily suspend service to all asynchronous priorities. The result is a uniform increase in the mean service delay of all asynchronous traffic.

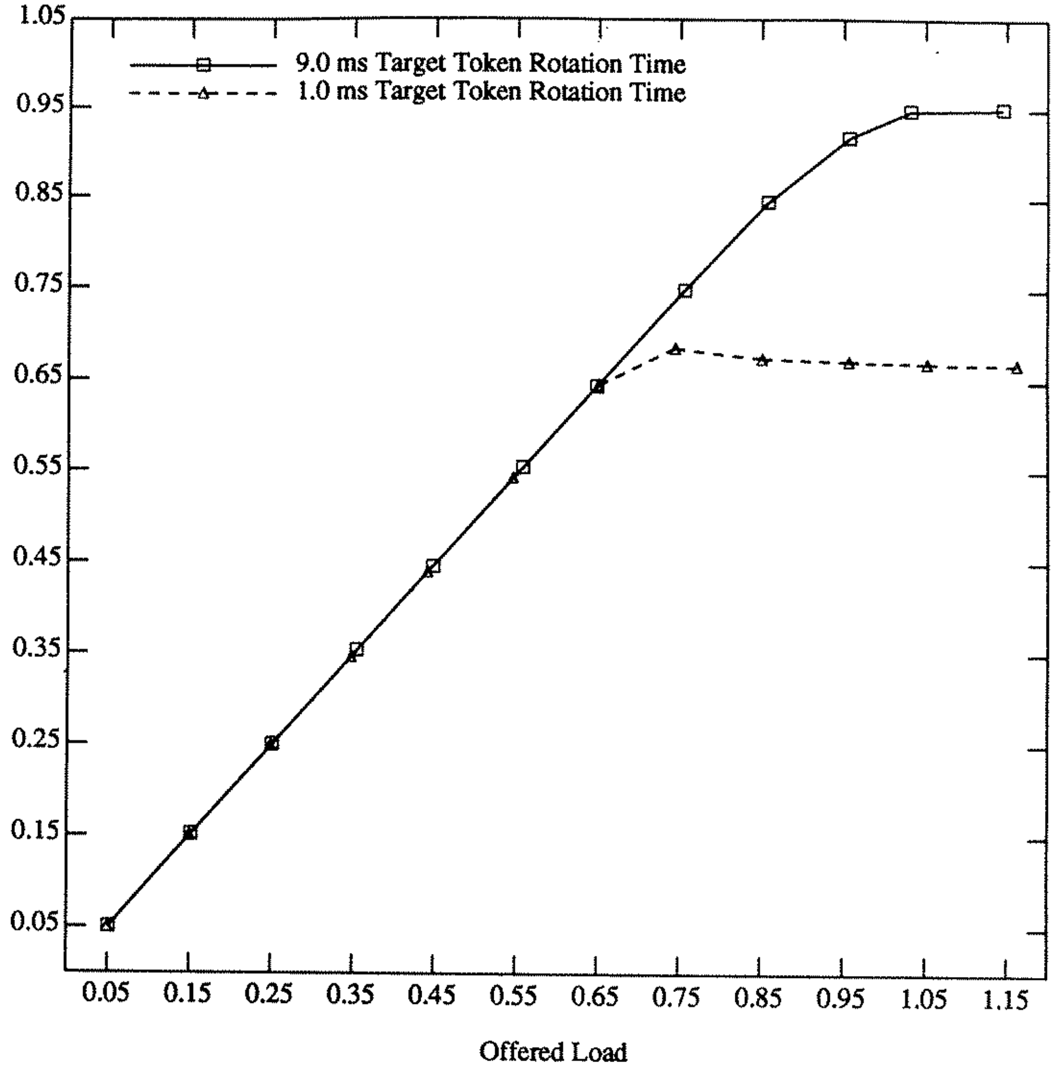
As offered load increases, service to synchronous traffic remains fairly constant, but service to the asynchronous traffic begins to decline. When a station receives a token that is not late, the highest asynchronous priorities are likely to get service, but due to the effect of late tokens, the high asynchronous priorities may have so many packets waiting for transmission that the station's token hold timer may expire before the lower asynchronous priorities get serviced. The result is that lower asynchronous priorities receive even longer mean service delays and eventually get no service at all. Beyond 95% offered load asynchronous priority zero receives service for only a fraction of the packets it offers.



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
Target Token Rotation Time Varied
Asynchronous Priority Thresholds Held Constant at TTRT
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

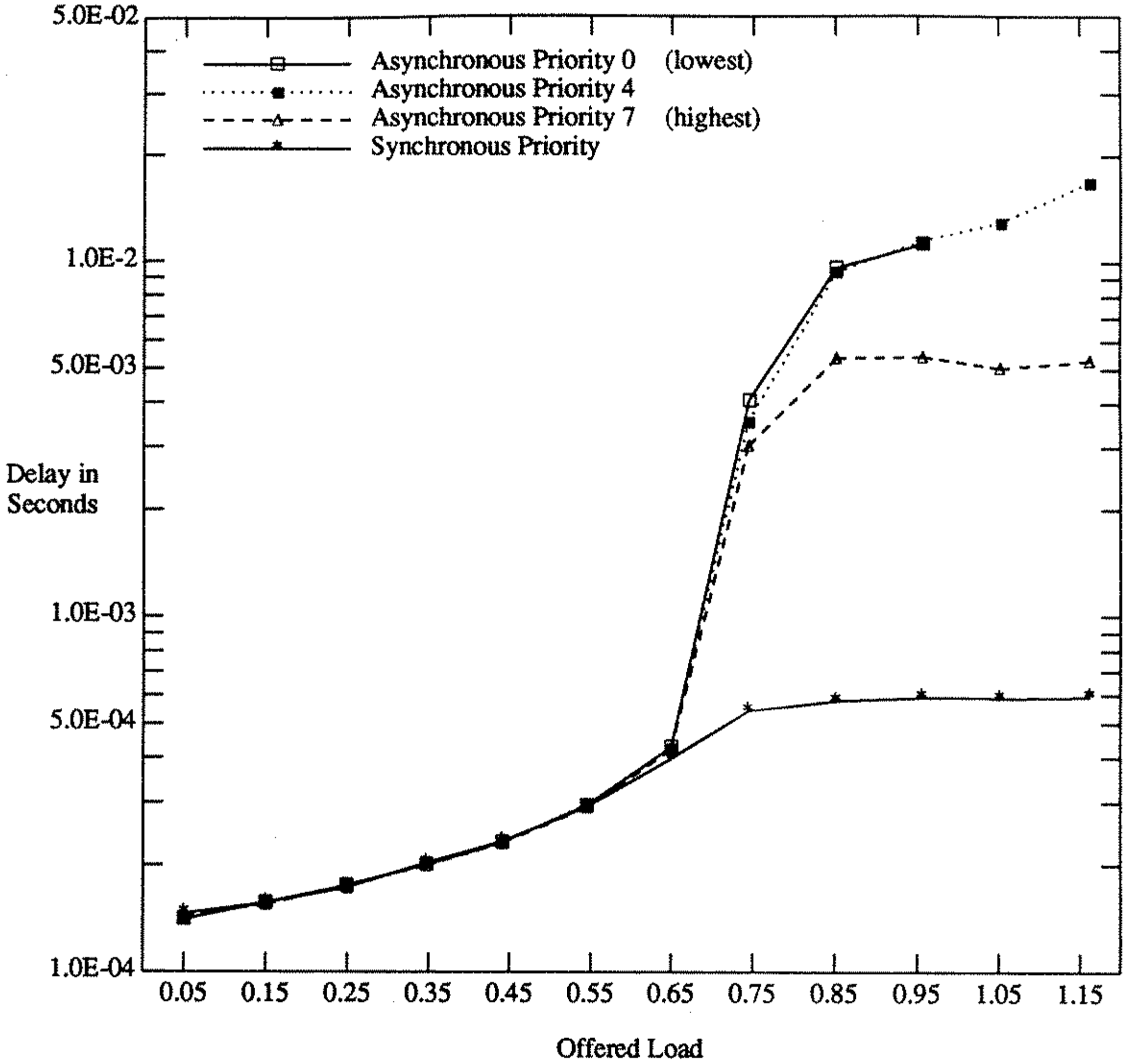
Figure 4.25 — Token Cycle Time Varying Target Token Rotation Time



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
Target Token Rotation Time Varied
Asynchronous Priority Thresholds Held Constant at TTRT
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.26 — Throughput Varying Target Token Rotation Time



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
Target Token Rotation Time Varied
Asynchronous Priority Thresholds Held Constant at TTRT
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
No Restricted Token Dialog

Figure 4.27 — Synchronous And Asynchronous Service Delay

1 ms Target Token Rotation Time

4.3.4. Effects of Restricted Token Dialog

Restricted token dialog is used to dedicate asynchronous bandwidth to a single extended dialog between a certain subset of stations (see Section 2.6.2.3.3). To see the effect restricted token dialog has on asynchronous traffic we have altered the multiple priority reference configuration so that 50 stations are eligible for restricted token dialog and 50 stations are not. Also, for this example we make the unrealistic assumption that the simulation will be in a restricted token dialog 60% of the time. We do this simply to make the effect of restricted token communication very clear.

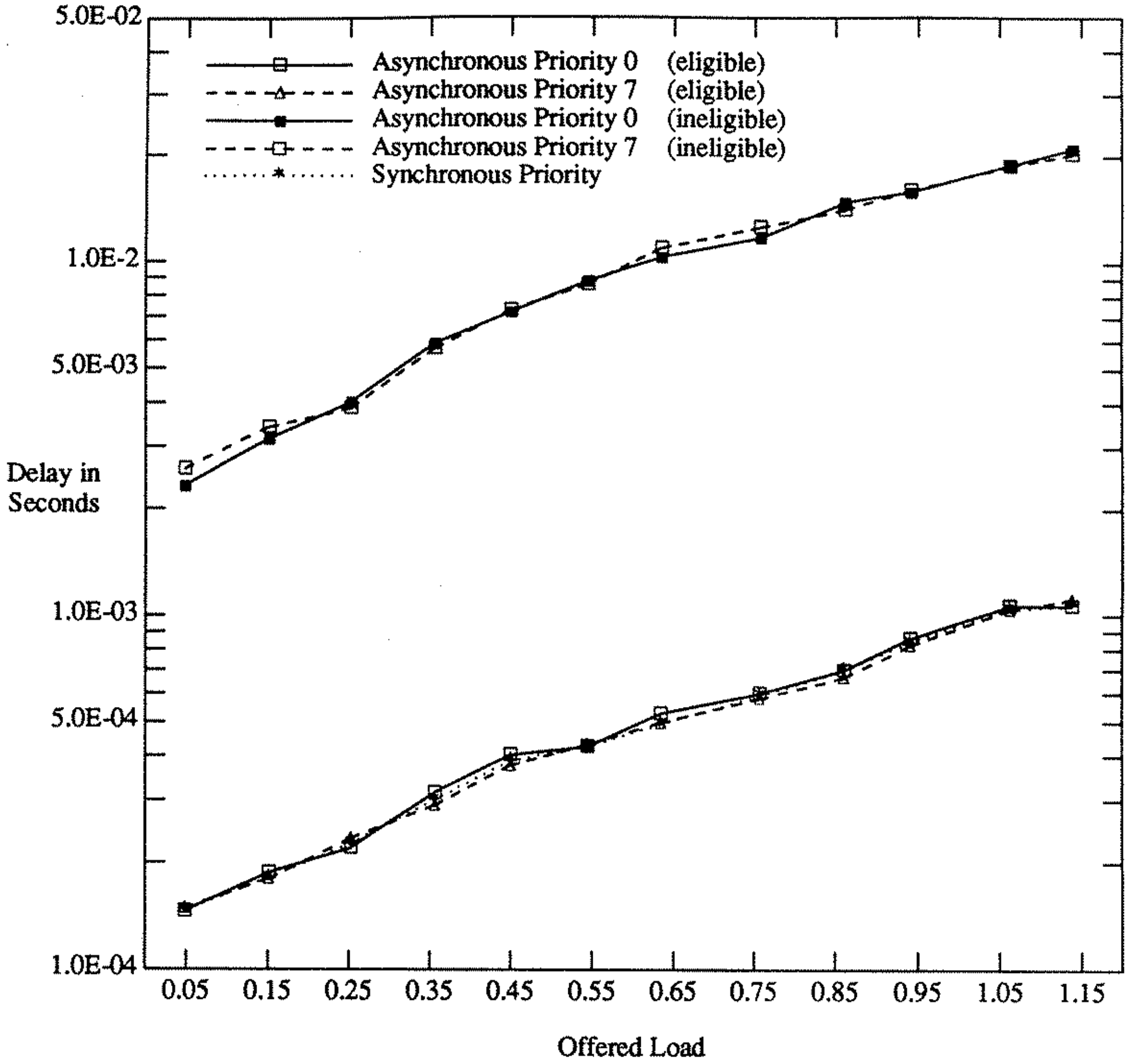
4.3.4.1. Packet Delay

Figure 4.28 shows mean packet service delay for the synchronous priority and the highest and lowest asynchronous priorities of stations that were eligible and ineligible for restricted token dialog. This graph shows that asynchronous priorities for the stations not eligible for a restricted token had uniformly higher service delay.

The reason this occurs is that restricted token dialog occurs in spurts. During these spurts, stations not eligible to participate receive no service for their asynchronous priorities. When the restricted token dialog is finished, service to all stations reverts to normal. The asynchronous packets that were not serviced during the restricted token dialog incurred a certain amount of delay in addition to the delay they would normally receive. This caused the delay of asynchronous priorities in stations not eligible for restricted token dialog to be uniformly higher than the delay incurred by the synchronous priority and the asynchronous priorities in stations eligible for restricted token dialog.

4.3.4.2. Throughput

Figure 4.29 shows that maximum throughput decreases significantly when restricted token dialog is implemented. The reason for this is that during restricted token dialog, stations not participating can only access the token for their synchronous traffic. This implies that ineligible stations hold the token for the short period of time necessary to service their synchronous traffic and then release it. The result is an increase in the number of token transmissions which in turn decreases throughput and the ineligible stations are not generating asynchronous traffic during restricted token dialog.

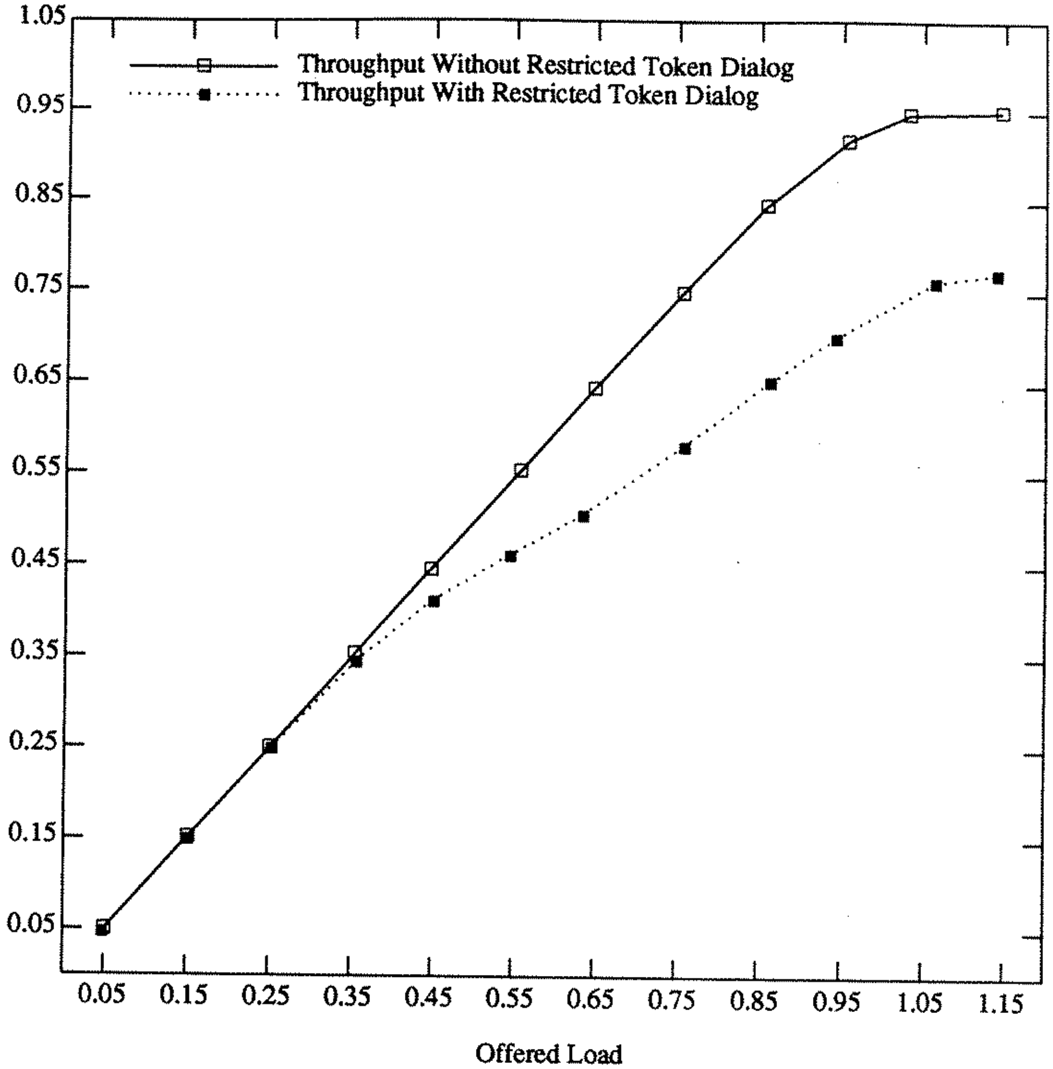


Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
Eligibility For Restricted Token Dialog Varied
60% of Simulation Time in Restricted Token Mode

Figure 4.28 — Synchronous And Asynchronous Service Delay

Eligible Station vs. Ineligible Station



Configuration:

100 Stations
100 Mbps Medium Rate
10 km Ring Circumference
32 Bits Internal Station Latency
A Single Synchronous Priority And Eight Asynchronous Priorities
9.0 ms Target Token Rotation Time
Asynchronous Priority Thresholds Held Constant at 9.0 ms
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
Eligibility For Restricted Token Dialog Varied
60% of Simulation Time in Restricted Token Mode

Figure 4.29 — Throughput Varying Restricted Token Dialog Usage

Chapter 5

Conclusions

5.1. FDDI Performance

5.1.1. Single Priority Operation

In the single priority reference configuration, FDDI's "timed token" mechanism was observed to have a "limiting" effect on token cycle time, tokens received and packets sent. As offered load increased, token cycle time approached the target token rotation time and began to stabilize. When token cycle time stabilized, the number of tokens a station received in a given amount of time also stabilized. The timed token mechanism also affected the "packets sent" metric. As offered load increased, the number of packets serviced per station per second increased until the synchronous bandwidth allocated to each station was no longer sufficient to service all packets eligible for transmission.

We also concluded from this reference configuration that a 100 Mbps transmission medium could transmit packets at such a rate that the delay incurred by actual transmission was negligible. We observed that the difference between station delay (queueing plus network access delay) and service delay for this configuration was not significant. Queueing delay and network access delay contributed nearly all the delay experienced by a packet.

5.1.1.1. Varying Mean Packet Length

Variations in mean packet length showed that small packets produced many more token accesses than large packets. This suggests that larger packets may improve maximum throughput slightly. Total packet delay was also not affected by packet size, but it was shown that network access delay contributed more to a packet's total delay as packet size increased.

5.1.1.2. Varying the Number of Stations

Increasing the number of stations on the ring yielded a longer token cycle time and increased the probability that the target token rotation time would be exceeded. This pointed out that when synchronous traffic dominates the network, FDDI's late token mechanism which throttles asynchronous traffic may be insufficient to control token cycle time if there is insufficient asynchronous traffic to degrade. In this case token cycle time is dependent primarily on the synchronous bandwidth allocated to each station at network initilization.

In addition, it was shown that more stations on the ring caused mean packet service delay to increase. The main reason for this was that as more stations compete for the token, token cycle time increases which in turn increases packet delay. A second reason is that each packet has more intermediate stations to pass through during transmission. This allows FDDI's large latency (32 bit times is a standard FDDI internal station latency) to contribute more to mean packet service delay as the number of stations increases.

5.1.1.3. Varying Ring Circumference

Increasing ring circumference produced the expected results of increasing token cycle time and mean packet service delay. We observed, however, that the extent of these increases was small relative to the differences in ring circumferences that produced them. For example, at 75% offered load, a network with a 1 km ring circumference yielded a mean packet service delay of 0.42 ms; at that same offered load a network with a 100 km ring circumference yielded a mean packet service delay of 1.8 ms. While ring circumference differed by two orders of magnitude, mean packet service delay differed by only a factor of four.

5.1.2. Multiple Priority Operation

It was shown that the performance of multiple priority operation was virtually identical to single priority operation when no artificial restrictions were placed on asynchronous traffic. Mean packet service delay for single priority operation varied from 0.16 ms at 5% offered load to 2.5 ms at 95% offered load. Mean packet service delay for all priorities in the multiple priority configuration varied from 0.14

ms at 5% offered load to 2.3 ms at 95% offered load. We saw that it was pointless to use multiple priorities unless their priority thresholds are set to different values to enforce discrimination.

5.1.2.1. Varying Priority Thresholds

When asynchronous priority thresholds were used it was shown that they produced the desired effect. Service to the synchronous and high asynchronous priorities improved dramatically. However, it was also observed that implementing the priority threshold mechanism caused maximum throughput to drop and resulted in total service cut off to the lower asynchronous priorities. We saw that using the priority threshold scheme had a dramatic effect on overall network performance, and thus should be used with care. The impact of the priority system is a topic for further study.

5.1.2.2. Lowering Target Token Rotation Time

Lowering the target token rotation time had the expected effect of limiting token cycle time. However, the additional token transmissions and late token arrivals that resulted significantly decreased maximum throughput. It was also observed that lowering target token rotation time improved service to synchronous traffic, but the increased number of late tokens caused asynchronous traffic to suffer uniform service degradation. These observations suggest that the negotiation process used to set the target token rotation time must be performed cautiously in order to avoid setting the target token rotation time so low that network performance is seriously degraded.

5.1.2.3. Restricted Token Dialog

Restricted token dialog had the effect of significantly improving service to asynchronous traffic in stations eligible to receive the token while at the same time uniformly degrading service to asynchronous traffic in stations not eligible for the token. This effect, however, resulted in a significant drop in maximum throughput, which suggests that restricted token dialog should itself be restricted or at least monitored to prevent it from significantly degrading overall network performance.

5.2. Suggestions For Future Research

The performance metrics presented in this thesis are strictly MAC layer performance values. They in no way reflect the level of service seen by an application process at the top of an ISO protocol stack. It would be interesting to investigate the impact which a 100 Mbps transmission medium would have on the upper layers of the ISO protocol stack. We believe that the application layer using the ISO protocols would be fortunate to utilize 10% of FDDI's MAC-layer bandwidth.

One very promising area of research involves exploration of new protocol implementations that can effectively utilize a high speed transmission medium. One such implementation is the Protocol Engine project [CHES87] being developed by Dr. Greg Chesson at Silicon Graphics Inc. Chesson is attempting to build a VLSI chip set which will implement the transport, network, and logical link control layers in hardware and then connect at the MAC layer directly to an FDDI chip set. His goal is to provide 100 Mbps bandwidth at the top of the transport layer.

Because FDDI is such a new protocol, an accurate chip set implementation does not yet exist. When one becomes available it would be interesting to compare its performance to the performance metrics produced by the simulator.

This thesis has in no way exhausted the usefulness of the simulator. Many aspects of FDDI such as effects of restricted token dialog and non-exhaustive multiple priority service were only briefly discussed. Two other areas of interest would be the study of heterogeneous traffic, where the load offered by each station varied, and the performance of heterogeneous traffic using multiple priorities.

Bibliography

[CHES87] Greg Chesson, *The Protocol Engine Project*, UNIX Review, September, 1987.

[FDDI86a] ANSI, *FDDI Token Ring Physical Layer Medium Dependent Standard*, draft proposed Standard X3T9.5/84-48, rev. 6, July, 1986.

[FDDI86b] ANSI, *FDDI Token Ring Physical Layer Protocol Standard*, draft proposed Standard X3T9.5/83-15, rev. 13, August, 1986.

[FDDI86c] ANSI, *FDDI Token Ring Media Access Control Standard*, draft proposed Standard X3T9.5/83-16, rev. 10, February, 1986.

[FDDI86d] ANSI, *FDDI Token Ring Station Management Standard*, draft proposed Standard X3T9.5/84-49, rev. 2, September, 1986.

[IEEE85a] IEEE, *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, The Institute of Electrical and Electronics Engineers, Inc., 1985.

[IEEE85b] IEEE, *Token-Passing Bus Access Method and Physical Layer Specifications*, The Institute of Electrical and Electronics Engineers, Inc., 1985.

[IEEE85c] IEEE, *Token Ring Access Method and Physical Layer Specifications*, The Institute of Electrical and Electronics Engineers, Inc., 1985.

[LAVE83] Stephen S. Lavenberg, *Computer Performance Modeling Handbook*, Academic Press, 1983.

[METC76] R. M. Metcalfe and D. R. Boggs, *Ethernet: Distributed Packet Switching for Local Computer Networks*, Communications of the ACM, July, 1976.

[PEDE87] Jeffery H. Peden, *Performance of the IEEE 802.5 Token Ring*, Master's Thesis, Department of Computer Science, University of Virginia, January, 1987.

[SAE87] SAE, *SAE AS4074.2 High Speed Ring Bus*, Final draft Standard, June, 1987.

[SEVC85] Kenneth C. Sevcik and Marjory J. Johnson, *Cycle Time Properties of the FDDI Token Ring Protocol*, RIACS report number TR-85.10, August, 1985.

[WEAV87] Alfred C. Weaver, *Local Area Networks And Busses—An Analysis*, Flight Data Systems, NASA—Johnson Space Center, January, 1987.