PERFORMANCE ANALYSIS OF THE SAE AS4074.2 HIGH SPEED RING BUS

A Thesis

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science (Computer Science)

by

David Wayne Minnich

August 1988

Abstract

In recent years, a new generation of high speed network protocols has begun to emerge. These networks, typically operating in the 100 Mbps range, include the ANSI Fiber Distributed Data Interface (FDDI) and the SAE AS4074.2 High Speed Ring Bus (HSRB). The Computer Networks Laboratory at the University of Virginia, seeing the need for an in-depth understanding of the operation of these protocols, has been conducting simulations and performance analyses of these networks. A general token-ring simulation system has been developed which allows analysis and comparison of these various protocols. The usefulness of this system is three-fold: 1) it allows performance analysis of network protocols for which no hardware currently exists, 2) the results of this analysis may be used to suggest protocol improvements to the design committee, and 3) network implementors may use the system to tune the protocols and determine optimum configurations.

The thesis includes a discussion of the simulation system in general, the SAE HSRB protocol in particular, and the results of the performance analysis.

Acknowledgements

I wish to thank Dr. Alfred C. Weaver for his invaluable guidance and assistance in producing this thesis.

Also, special thanks to my parents, Beverly and Dale Minnich, for their ongoing support and encouragement; Randy Simonson, for his help in co-designing the simulation environment, and for his continual contributions as both a friend and colleague; Jeffery Peden, for his tremendous help throughout the course of this research; and Tim Strayer, for discussions which helped clarify the analysis of the simulation results.

This research was funded, in part, by the Flight Data Systems Branch, NASA — Johnson Space Center, and by the Institute of Information Technology of the Virginia Center for Innovative Technology.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

1.1. INTRODUCTION

We live in a world continually presented with higher degrees of sophistication in computer hardware and associated technology. The growth rate of the industry has been phenomenal: roughly a six-fold increase in 30 years. In such a world, the need for integration is rapidly achieving paramount importance. This integration may take many forms, for example the centralized control of a factory floor, coordination of data-processing activities in an office environment, or the interconnection of diverse computer systems. In all of these applications, the common element is the need for coordinated communication between entities (be they robots, word processors, or mainframe computers) which are physically separated. The communication vehicle which we will examine in this thesis is the Local Area Network (LAN), specifically the SAE AS4074.2 High Speed Ring Bus (HSRB) protocol.

The remainder of Chapter One will establish a context within which to view the HSRB protocol. Chapter Two will deal with the specifics of the protocol itself. Chapters Three and Four will explain the methods by which the protocol will be analyzed and the results of that analysis, and Chapter Five will suggest conclusions to be drawn from the analysis.

1.2. DATA PROCESSING NETWORKS vs. REAL-TIME CONTROL NETWORKS

Local Area Networks (LANs) may be divided into two large classes based on their primary purposes: data processing (general purpose) networks and real-time control networks. Data processing networks are characteristically concerned with the transfer of large blocks of data or files. The primary purpose of these networks is the transfer of data. These are what Berggren [BERG87a] refers to as "resource sharing" systems. If a noticeable delay is incurred in the transfer of the information, this may result in an inconvenience to the user, but it rarely results in more serious repercussions. Furthermore, these systems commonly link many heterogeneous resources, so a great deal of generality must be present in the device interface. Examples of such systems include a network file system, remote procedure call facilities, and remote peripherals such as printers and plotters.

Real-time control networks, on the other hand, are concerned with much more than information transfer; these networks implement distributed control systems. The messages transferred are typically short. Furthermore, messages are typically associated with "hard" deadlines - deadlines which, if not met, may result in catastrophe. Examples of such systems might be a robot control system on a factory floor, the control system for a petro-chemical refinery, or the flight control system of a jet aircraft.

Despite the above differences, both data-processing networks and real-time control networks have certain common characteristics. Both require some form of communication medium. In a LAN, this medium is typically electrical or optical cable. Additionally, both require a means whereby a device may access the medium whenever it has a message to send. The process whereby a device gains control of the medium and sends a message is part of the network protocol.

Obviously, there are many applications for each type of network. Furthermore, a protocol which is suitable for one type of network may not necessarily be suitable for the other. The particular characteristics which make a protocol suitable for one or another type of network are discussed in the following sections. This discussion represents some of the developmental process of the SAE HSRB protocol.

1.3. DEVELOPMENT OF THE SAE AS4074.2 HIGH SPEED RING BUS (HSRB)

The process of developing a protocol for a specific application may be subdivided into several steps. First, one must determine the performance requirements which the protocol must meet. As this relates to the HSRB, these requirements are discussed in Section 1.3.1. Secondly, one must determine if any existing protocol will meet these requirements with little or no modification. If one will, then it makes little sense to design a new protocol. We will work through this process in Section 1.3.2. Lastly, assuming no protocol exists which exactly meets the requirements, one must create a new protocol or modify an existing one to meet the specified requirements. While this thesis does not explicitly treat this process, we see the results of this process in the final draft version of the HSRB protocol [SAE87], which is summarized in Chapter Two.

1.3.1. REQUIREMENTS AND MOTIVATING FACTORS

In order to understand the rationale behind the HSRB protocol decisions, we must first establish a context within which the protocol is intended to operate. This context consists of the physical environment in which it is anticipated that the network will be used, as well as the specific operational requirements imposed on the system by factors such as desired performance.

The SAE HSRB is intended to be the real-time control network for a fairly localized control process, for example a factory floor, a jet airplane, or a ship. This implies a certain set of constraints within which the network must operate. The discussion of these constraints follows, and is derived largely from [BERG87b] and [GEYE87].

1.3.1.1. REAL-TIME RESPONSE

Perhaps foremost among the design considerations of the HSRB is the need for guaranteed, real-time response. Since the penalties for missing a deadline in such a control system may be catastrophic, all possible precautions must be taken to ensure that this does not occur. Real-time performance is dependent on several inter-related factors, including deterministic latencies and high throughput.

1.3.1.1.1. DETERMINISTIC LATENCY

In order to guarantee that a message will arrive within its given deadline, it must be possible to calculate an upper bound on its delivery time. Although the maximum allowable value for this upper bound may vary from application to application, one should be able to determine whether message delivery is guaranteed for a known set of operating characteristics.

1.3.1.1.2. HIGH THROUGHPUT

Another contributor toward real-time response is high throughput. If a network is fundamentally incapable of delivering a given number of messages per unit time, then some of those messages will fail to meet their deadlines. One method of increasing potential throughput is to increase the transmission speed of the network. Obviously, the faster the rate at which the bits are transmitted, the more bits will be received at the destination in a given amount time. As technology improves, the maximum achievable transmission rate increases, and so, potentially, does the throughput of the network. Another method of increasing the potential throughput of the network is to increase the percentage of available bandwidth which may be used to transmit actual data. This percentage (also called normalized throughput or efficiency) may be limited by both the network topology and the overhead inherent in a given communications protocol, so the proper choice of both is essential to achieve high efficiency.

An important factor when considering network throughput is network stability. A network should be capable of maintaining high throughput for values of offered load throughout its operating range. As offered load increases, performance should not peak and then degrade; instead it should achieve some maximum value and maintain that value.

Yet another factor to examine when considering throughput is the expected message length. Short messages are much less efficient than long messages for several reasons. First, to deliver a given number of bits using short messages requires a higher percentage of message overhead than that required for long messages. Secondly, for certain types of protocols, if a message is smaller than some minimum value it incurs some implicit overhead equal to the difference between this minimum value and the message length. For these reasons, a protocol may need to explicitly consider how to handle short messages in order to achieve a high degree of efficiency.

1.3.1.2. RELIABILITY AND FAULT TOLERANCE

Also of paramount concern in the design of the SAE HSRB were the issues of network reliability and fault tolerance. The network should be as error-free as possible, and should recover gracefully in the case of an error. This includes both errors due to noise in the transmission medium and those due to breaks in the transmission medium. The protocol should define methods by which the network may recover from such errors, and such recovery should be rapid and automatic. Furthermore, no single point of failure should disable the network; more severe errors should result in gradual degradation in service, rather than the cessation thereof.

1.3.1.3. HARDWARE INDEPENDENCE

In order to ensure the continued applicability of the HSRB protocol over a period of time, the SAE design committee wished to separate, as much as possible, the details of the protocol from the technological considerations which must be made at implementation. Thus, the HSRB protocol is required to be independent of both the transmission medium and the rate of data transmission. This guarantees that the protocol is capable of encompassing technological advances which will certainly occur after the protocol is finalized.

Obviously, at some level the protocol must address these hardware issues. Therefore, all such issues are specified in appendices to the protocol. The protocol document itself must be specified in terms of parameters and abstractions. For example, timer values must not be given in absolute units such as seconds, but in terms of bit times at the signaling speed. Thus, as technology advances the protocol can take advantage of those advances by merely establishing an appendix suitable for the new implementation.

1.3.1.4. PRIORITIZED ACCESS

The HSRB design requires a system of prioritized access to the communications medium. That is, there must be some network-wide method of determining the order in which messages should be sent. This is different from the concept of priority in some protocols, which establish prioritized access only within a station. In the HSRB, a station with low priority messages to send must not block a station with high priority traffic.

1.3.2. SUITABILITY OF EXISTING PROTOCOLS

As we examine the suitability of existing Local Area Network protocols, those which deserve most attention are those which have been accepted as standards. These include the IEEE 802.X family of protocols and the newly developed ANSI Fiber Distributed Data Interface (FDDI). For various reasons (e.g. relative efficiencies, fault tolerance, hardware independence) we rule out the IEEE 802.3 Contention Bus and the 802.4 Token Bus. For detailed analyses of these protocols see [SCHW87] and [SUMM85]. Consequently, we will concentrate on the IEEE 802.5 Token Ring and the FDDI Token Ring protocols. Detailed analyses of these protocols may be found in [PEDE87] and [SIMO88].

1.3.2.1. IEEE 802.5

On the surface, it seems that the IEEE 802.5 Token Ring is a reasonable candidate for consideration. It has the deterministic latency and high efficiency which are so instrumental in obtaining real-time response. It also specifies a method of ring-wide prioritized access. However, the protocol is bound to a particular transmission rate, and throughout the protocol assumptions are made which are based on this fixed transmission rate. Furthermore, 802.5 makes no allowance for short messages, and hence suffers degradation in service for messages averaging less than the total ring latency in length. Although the protocol specifies methods of error recovery in case of corrupted transmissions, the network is composed of a single physical ring, and thus may be disabled by a single point of failure. For these reasons, it appears that 802.5 is not sufficient to meet the needs of the SAE committee.

1.3.2.2. ANSI FIBER DISTRIBUTED DATA INTERFACE (FDDI)

Again, upon superficial examination of the protocol, FDDI would appear to meet the requirements set forth for the HSRB. It achieves the high degree of efficiency common to token ring protocols, it operates at a very high transmission rate, and it has methods for implementing priorities. Furthermore, it has the necessary fault tolerance which was absent in the IEEE 802.5 protocol. However, FDDI is even more strongly tied to a given transmission rate than is 802.5 (due to its timed token mechanisms), and it is tied to a particular medium as well (fiber optics). Additionally, the priority scheme operates only within a given station. In other words, messages within a station are transmitted in priority order, but the right to transmit passes serially around the ring. Hence, stations with low priority traffic may be allowed to transmit before stations with urgent, high priority traffic.

It was determined that FDDI was unsuited to the needs of the SAE committee. Therefore, the development of a new protocol was, indeed, justified. The protocol which is the result of this development is the SAE AS4074.2 High Speed Ring Bus (HSRB) discussed in Chapter 2.

Chapter 2

SAE HSRB Protocol

2.1. INTRODUCTION

The SAE AS4074.2 High Speed Ring Bus (HSRB) is a high speed, local area network with a token-passing ring topology. In a token ring network, each physical station is connected to its upstream and downstream neighbors by a unidirectional, point-to-point link. Thus, the stations form an actual physical ring (as opposed to the logical ring of the token bus topology).

The HSRB is designed to accommodate either a single physical ring if high reliability and fault tolerance are not pressing concerns, or multiple redundant rings if such issues are of concern (see Figures 2.1 and 2.2). The protocol standard assumes that a dual-redundant HSRB typically will be used, and so defines the appropriate ring reconfiguration procedures to ensure maximum reliability and fault tolerance.

The HSRB protocol is written to be independent of medium, transmission rate, and station separation. The parameters which are dependent on these physical characteristics are defined in appendices to the protocol document ("slash sheets"). Both a 50 megabaud electrical implementation and a 100 megabaud fiber optic implementation are described.

Each station in the network consists of a host device, one or more Ring Interface Units (RIUs), and one or more Ring Interface Modules (RIMs) for each RIU (see Figure 2.3). The RIU serves as the interface between the host machine and a HSRB. It is possible for a host to be connected to multiple HSRBs, in which case the host will have multiple RIUs. The RIU also performs the following functions:

- receives the serial transmission from each RIM, copying and decoding all messages which are addressed to this station, repeating all transmissions which were not originated by this station, and stripping from the ring all transmissions which were originated by this station.

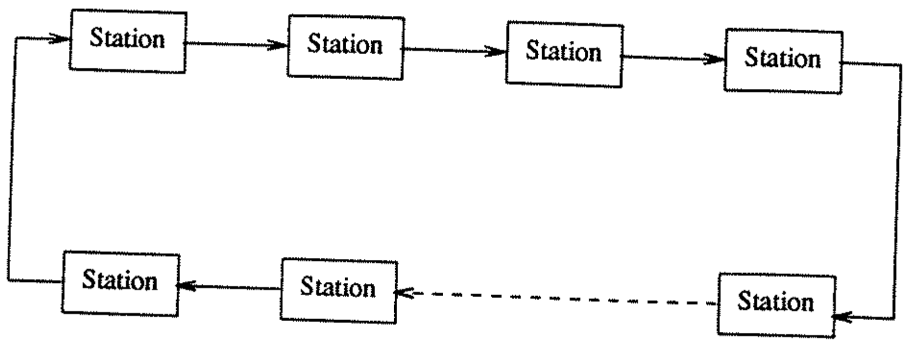


Figure 2.1 — Single Ring

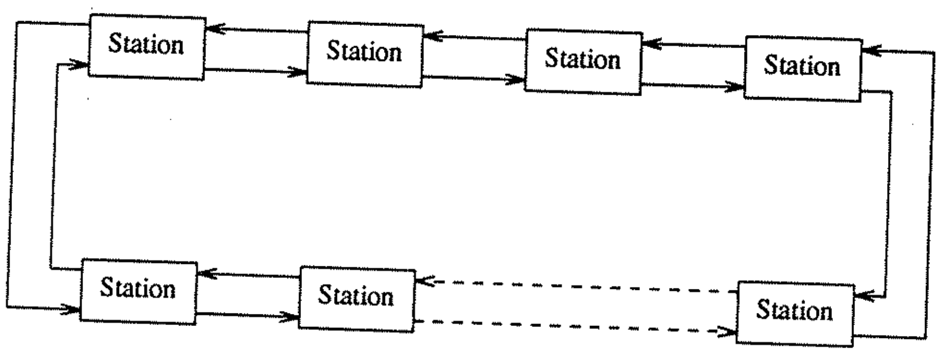


Figure 2.2 — Dual, Counter-rotating Rings

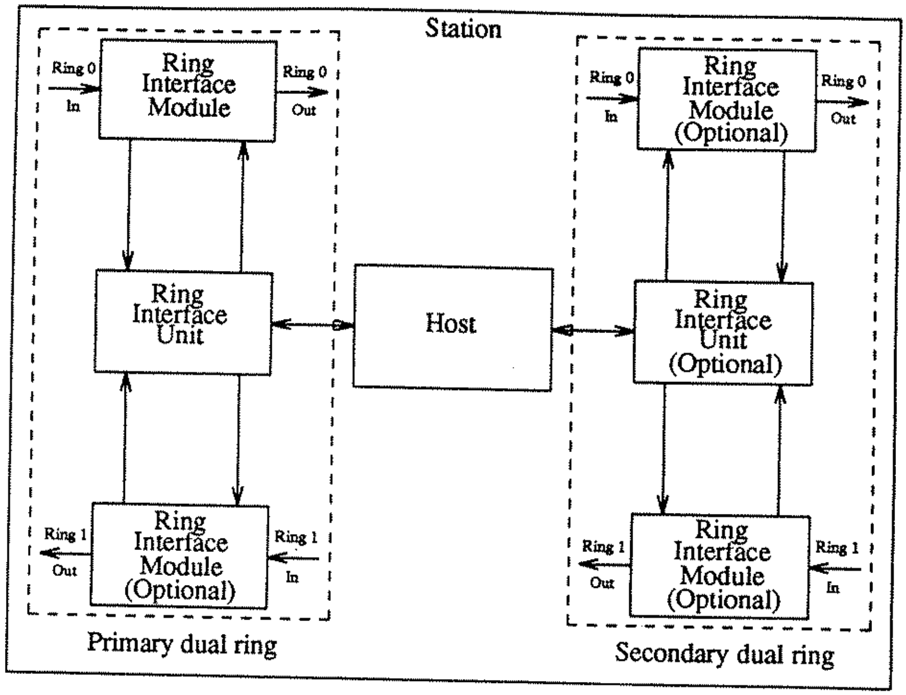


Figure 2.3 — Station Configuration

- performs error checking on tokens, headers, and data.

- captures a Free Token when this station has a message to transmit.

- transmits a Free Token upon completion of message transmission.

The Ring Interface Module serves one of two purposes. If the Ring Interface Unit is active and functional then the RIM connects the RIU to the HSRB. If for some reason the RIU is nonfunctional (no power, hardware fault, disconnected from ring, etc.) the RIM serves as a mechanism to ensure the integrity of the ring, immediately transmitting along the ring any frames received by the RIM, thereby effectively bypassing the station. Thus, the failure of a particular host or RIU will not cause total network failure.

The HSRB protocol allows for up to eight distinct priorities of messages. These priorities are arbitrary classifications by the user which designate the importance of a particular message. The HSRB may neither set nor change these priorities. A queue is maintained by each station for each message priority, and within each queue messages are processed in order of arrival. The protocol does not attempt to set any bounds on message delivery time; instead, it attempts to guarantee a relative order of service.

When a station has a message to send, it must first acquire a "Free" Token (in the manner described in Section 2.4). Once said Token is acquired, the station changes the Token's status from "Free" to "Claimed" and appends to the Token its highest priority message. Upon completion of message transmission the station re-issues a Free Token. Each station may transmit only one message per Token possession. If a station has multiple messages to send, it must nevertheless transmit a Free Token following a message, allowing any other station with a message of equal or greater priority to claim the Token. Thus, under normal operation only one message will exist on the ring at a time.

Optionally, multiple "short messages" may be allowed to coexist on the ring in an effort to increase throughput. A "short message" is defined to be any message which is completely transmitted before its prefacing Token has circumnavigated the ring. The explanation of the short message protocol is contained in Section 2.5.

Control of the HSRB is distributed among all active stations on the ring. All stations are responsible for error detection. If the error is such that message retransmission is sufficient corrective action, then the first station to detect the error notifies the sending station by setting the appropriate error indication bits in the Message Frame. If the error is of a more serious type, the station detecting the error may initiate reconfiguration of the ring or some other appropriate action. At any given time one station is designated as the clock master station, and is responsible for maintaining ring synchronization. In order to accomplish this the clock master station manages an elastic buffer which may vary in size from 1 to 128 bits. Every active station must be capable of performing this function, and failure of the current clock master station results in reconfiguration of the ring.

The HSRB protocol defines two modes of station addressing: physical and logical. In physical addressing mode the protocol allows up to four physical HSRBs connected by bridges, with up to 128 stations per ring, for a maximum of 512 stations. Furthermore, the protocol supports 512 subaddresses per station. In logical addressing mode there are 2^{64} possible addresses. This allows the definition of logical groups of stations and thus allows broadcast (every station) and multicast (multiple station) transmissions. Stations which require specific data may be so grouped; when the data becomes available it is transmitted with a Logical Address which is recognized by all stations needing the data. Since a station may have more than one Logical Address, and since the space of possible addresses is so large, one has the effective capability to "label" different types of data. In a factory floor control network, for example, if a control message is to be sent to all machines of a specified type, the message could be sent with the Logical Address which signifies that particular type of machine. Such a group could be defined for each type of control message which might occur.

2.2. TERMS

Dual redundant — indicates that the backbone of the network consists of two physical rings, each capable of performing the necessary tasks to keep the network running.

Host — the device (computer, sensor, controller, etc.) which is attached to the network through the Ring Interface Unit.

Message — the Protocol Data Unit (PDU) which contains the actual data being transmitted.

Offered Load — the ratio of bits presented to the network for transmission (per unit time) to the maximum number of bits transmitted in that time. **Reservation** — the priority at which the next Free Token will be issued.

Symbol — a group of four bits which is encoded as a 5-bit unit.

Throughput — as used in this thesis, the ratio of data bits transmitted to total bits transmitted per unit time.

Token — the Protocol Data Unit (PDU) which grants the right of transmission to a station which claims it.

Protocol Data Unit (PDU) — any of a number of bit sequences which are recognized by the protocol as having special meaning.

2.3. DATA ENCODING

The HSRB allows transmission of messages from 1 to 4096 words in length, each word being 16 bits long. A word is subdivided into four groups of four bits each. Each four-bit group is encoded as a five-bit symbol (called signaling bits) using a 4-into-5 bit encoding scheme. The 32 possible 5-bit codes and their associated values are shown in Figure 2.4 below.

Because of this encoding scheme, the protocol inherently suffers a twenty percent reduction in effective signaling rate (since only four bits of data are transmitted for each five signaling bits).

The encoded data is transmitted using a Non-Return to Zero Invert (NRZI) scheme. An NRZI encoded stream is derived from a Non-Return to Zero (NRZ) encoded stream in the following manner:

- the bit duration for each is identical.

- an NRZ logical '1' causes a transition in the NRZI signal level.

- an NRZ logical '0' does not cause a transition in the NRZI signal level.

The combination of the 4-into-5 bit encoding and NRZI transmission allows the extraction of a clock signal from the data stream.

2.4. PRIORITY OPERATION

The SAE AS4074.2 High Speed Ring Bus protocol defines eight possible levels of service (priorities). This allows the user to assign a greater degree of importance (and hence a higher level of service) to those messages which are deemed to be more crucial to the system. Thus, in a factory, real-time control data might be assigned a higher priority than purchasing or inventory information, since the time criticality of the first is much higher than that of the second.

The mechanism by which the priority operation is implemented is a reservation scheme. As a Claimed Token traverses the ring, stations with a message to send (the priority value of which exceeds the value of the current Reservation) copy the priority value of their message into the Reservation field. When the sending station receives the Token it issued, it compares the value of the Reservation field with its highest priority message (if any). If the Reservation value is higher or if the station has no more messages to send, then a Free Token is issued with that priority. If, however, the priority value of the message is

Symbol	Code Group	Assignment
0	11110	Data Symbol '0000' Binary
1	01001	Data Symbol '0001' Binary
2	10100	Data Symbol '0010' Binary
3	10101	Data Symbol '0011' Binary
4	01010	Data Symbol '0100' Binary
5	01011	Data Symbol '0101' Binary
6	01110	Data Symbol '0110' Binary
7	01111	Data Symbol '0111' Binary
8	10010	Data Symbol '1000' Binary
9	10011	Data Symbol '1001' Binary
A	10110	Data Symbol '1010' Binary
B	10111	Data Symbol '1011' Binary
C	11010	Data Symbol '1100' Binary
D	11011	Data Symbol '1101' Binary
E	11100	Data Symbol '1110' Binary
F	11101	Data Symbol '1111' Binary
I	11111	IDLE. No information currently being transmitted
J	11000	Used for Token Frame, Message Frame, and Beacon Frame Starting Delimiters
K	10001	Used for Token Frame and Beacon Frame Starting Delimiters
Q	00000	QUIET. No signal being transmitted
T	01101	TERMINATE. Ending delimiter for Token, Message, and Beacon Frames
S	11001	Never transmitted as a symbol, but it may occur in a Token or Frame Status field
V	00001	INVALID. These symbols shall not be used, since they violate conditions of the 4-into-5 bit encoding scheme and the restrictions applied by that scheme
V	00010	
V	00100	
V	01000	
V	10000	
V	00011	
V	00101	
V	00110	
V	00111	
V	01100	

Figure 2.4 — 4-into-5 Bit Encoding Scheme

greater than that of the Reservation the Free Token issued has a priority equal to the message's priority. The Token continues around the ring until claimed by a station with a message of equal or greater priority. This station need not have set the Reservation in order to claim the Token. The claiming station resets the Reservation to the lowest priority and strips the Token Ending Delimiter from the Token. Note that even though a station may have multiple messages of the highest priority to send, it must transmit a Free Token after each of them. Although this may cause a slight degradation in performance at high loads, it eliminates the problem of having long periods of time without the transmission of a Free Token. The placement of the Priority bits, Token Status bits, and Reservation bits in the Token allow minimal delay in executing the Reservation logic.

2.5. SHORT MESSAGE PROTOCOL

One of the goals of the SAE AS4074.2 HSRB designers was the optimization of throughput. If we view the reservation priority scheme with this in mind, we find two cases. If the expected length of a message exceeds the total ring latency, then the Token will have returned to the issuing station before message transmission is completed. Therefore, when the station is ready to issue a Free Token, the Reservation value from the previous Token is already known, and the Token can be issued immediately. This is done with no loss of throughput. If, however, the expected message length is less than the total ring latency, the time spent waiting for the Claimed Token to return to the issuing station (in order to obtain its Reservation value) is effectively wasted. As the transmission speed of the network increases, more bits can be stored on the ring at a time, hence this throughput penalty becomes greater. Obviously, some corrective measures must be taken. The problem is that if a station wishes to transmit a Free Token immediately upon completing transmission of a Short Message, it has no way of knowing at what priority the Token should be issued. Any transmission of the Free Token before receipt of the previous Claimed Token (and, hence, before receipt of the Reservation value) must necessarily violate the priority scheme. As a compromise, the designers offer an optional Short Message Protocol.

If the Short Message Protocol is implemented, multiple Short Messages may coexist on the ring. The value of the Short Message Count field indicates the number of Short Messages immediately preceding the current message. If that value is less than fifteen, and if the current message is a Short Message, then the current station increments the Short Message Count and transmits a Free Token with its priority set to the lowest priority. However, if the Short Message Count equals fifteen, the station must wait until it receives its Claimed Token (i.e. the message is "forced long") before it may transmit a Free Token. The Token is transmitted with a correct priority as described previously, and the Short Message Count is reset to zero. Thus, no more than sixteen Short Messages will be sent before the priority system is reinstated.

In a network in which the Short Message Protocol is not implemented, a station issuing a Free Token always sets the Short Message Count to fifteen, thus "forcing long" each message.

2.6. JOINING THE RING

Either at ring initialization or some time after a ring has been configured a station may wish to join the ring. In either case, an unconnected station must follow the same procedure. Upon application of power to the RIU (or upon reset of the RIU by the host) the RIU performs a self-test procedure. If the self-test fails for both of the counter-rotating rings, the station enters a quiescent state, and the RIU is bypassed. If the self-test succeeds, the station attempts to synchronize itself to the incoming signal. It also transmits IDLE symbols to allow the down-ring station to synchronize. If synchronization is achieved before 1024 IDLE symbols have been transmitted the station enters the Reconfiguration state; else, the station enters a listening state, continuing to attempt to synchronize while attempting no transmission. If the station subsequently achieves synchronization it enters the Reconfiguration state.

2.7. RECONFIGURATION

Reconfiguration is the process whereby a group of independent stations are connected as a network. Reconfiguration is required at startup, whenever a station wishes to enter or leave the ring, and after the occurrence of various types of errors (e.g. a break in the medium). The goal of the Reconfiguration process is to establish a ring which connects a maximum number of stations. This process may consist of simply determining that both of the counter-rotating rings are functioning, and choosing which one to use as the active ring. On the other hand, a break in the medium may result in a more complicated configuration (see Section 2.9.2).

The Reconfiguration procedure proceeds as follows:

1) A station entering Reconfiguration issues a Restart Beacon to cause the other stations to also enter Reconfiguration.

2) The station enters the Vie state and begins to send Vie Beacons every 16 symbol times, separated by IDLE symbols. The fields of the Vie Beacons (Highest Known Address, Beacon Path Indicator, Station Count -- see Section 2.10.3 for details) are updated as the Beacon travels around the ring. Upon receiving a valid Vie Beacon, the registers which keep track of the values of these fields are updated.

3) The station indicated by the Highest Known Address is configured as the Clock Master Station, and all others are configured as Clock Slave Stations. The Clock Master Station, after transmitting a sequence of IDLE symbols for synchronization purposes, issues a Free Token, and the ring begins normal operation.

2.8. WARM START

Warm Start is the recovery procedure used by the HSRB when an error has occurred which may not require reconfiguration of the ring. Upon detecting such an error, a station issues a Warm Start Beacon. Any station receiving a Warm Start Beacon repeats it. Upon issuing or repeating a Warm Start Beacon, a Clock Slave Station resets its Loop Time Counter (see Section 2.9.1.2.1) and proceeds to repeat all symbols received at its input. Other than this the station shall ignore all input other than a Warm Recovery Beacon or a Restart Beacon. A Warm Recovery Beacon causes the station to return to normal operation, while a Restart Beacon causes the station to enter Reconfiguration. If two loop times expire before one of the above is received, the station issues a Restart Beacon and enters Reconfiguration.

Upon repeating or issuing a Warm Start Beacon, the Clock Master Station transmits eight IDLE symbols followed by a Warm Recovery Beacon. Following the Warm Recovery Beacon, and until its return, the Clock Master Station strips all symbols received at its input and transmits IDLE symbols. If the Warm Recovery Beacon is not received by the Clock Master Station within one loop time, the station issues a Restart Beacon to cause Reconfiguration. Otherwise, the station transmits a Free Token with its Priority and Reservation set to the lowest priority, and the Short Message Count set to zero.

2.9. FAULT DETECTION AND RECOVERY

In order for the HSRB to be as tolerant of error as possible, the protocol describes procedures for the detection of, and recovery from, errors which may occur. These errors may be categorized as one of two broad types: transmission errors or hardware faults. Obviously, hardware faults may result in transmission errors, but we will view each from its own perspective. The methods of recovery for each are different, hence the different treatment.

2.9.1. TRANSMISSION ERRORS

For the purpose of this thesis, a transmission error is defined as a transient error which causes corruption of the bit stream. We will consider two types of transmission errors: those which may be detected merely by examining the bit stream, and those the detection of which requires the use of a timer or counter. Errors in the first category include:

-- Field Inconsistency errors

-- Information errors

-- Message Control errors

-- Token Status errors

-- Message Frame Starting Delimiter errors

-- Token format errors

-- Priority too high

-- Priority too low

-- Reservation too high

-- Reservation too low

-- Short Message Count errors

Errors in the second category include:

-- Token Starting Delimiter errors

-- Lost Token

-- Lost Free Token

-- Uncontrolled Transmit

The methods for detecting and recovering from the above-mentioned errors are explained below.

2.9.1.1. BIT STREAM DETECTABLE ERRORS

Several of the fields in protocol Frames are duplicated to ensure that the control information they contain is received correctly. Some examples are the Token Status fields in the Token Starting Delimiter and the subfields of the Frame Status field (see Section 2.10 for details). Under error-free operation, these values will be identical. If a station detects a Field Inconsistency, the appropriate Error Detected bits are set to notify the sender. All the information in such a message is ignored by all receiving stations.

Information errors and Message Control errors are detected by either the receipt of an invalid symbol or an incorrect Frame Check Sequence field. In either case the station detecting the error notifies the sender by setting the appropriate Error Detected bits. Again, all information in a Message Frame which is found to be in error is ignored by all receiving stations.

Token Status errors are detected by a disparity in the Token Status bits. A station sensing the error ignores the Token and initiates Warm Start.

Message Frame Starting Delimiter errors are detected by the transmitting station when its Message Frame returns. Since the receiving station(s) may have ignored the message, it will be requeued for a retry unless is already was a retry. In either case, the error is reported to the host.

Token format errors occur when any of the 1's which should occur in the Control field are missing. These 1's, the positions of which are governed by the protocol, must be present to ensure the extraction of the clock signal from the bit stream. Recovery consists of Warm Start.

If an error results in a valid Free Token with an increased priority, the HSRB recovers in the following way. A station which has a message to send, and which sees two consecutive Free Tokens with no intervening Claimed Token, initiates Warm Start. The only way the above condition will be met is if the circulating Token has a priority higher than any message currently awaiting transmission.

On the other hand, if an error results in a valid Free Token with a diminished priority, no explicit recovery mechanism is necessary. The token will be claimed by either a station with the intended priority message or by some other station. In either case, once the Token is claimed normal operation is re-established.

If an error results in a valid Claimed Token with a corrupted Reservation field, again no explicit recovery is necessary. If a station subsequently sets the Reservation to a higher value than the previous, uncorrupted value, then normal operation is re-established. However, if no such Reservation occurs, the error will propagate either to Priority Too High or to Priority Too Low and be handled in the appropriate manner.

Short Message Count errors (a valid Token with a corrupted Short Message Count) require no explicit recovery. If the count is too high, then a message will be "forced long" unnecessarily, but normal operation will be restored. If the count is too low, then more than sixteen short messages may be transmitted. This will not noticeably affect the latency, since in the worst case (assuming only one such error) only 30 short messages will be transmitted before a long message is enforced and normal operation is restored.

2.9.1.2. TIMER DETECTABLE ERRORS

The HSRB protocol provides two types of timers and two types of counters for the purpose of error detection. They are:

-- Loop Time Counter

-- Uncontrolled Transmit Inhibit Timer

-- Token Starting Delimiter Counter

-- Lost Free Token Counter

These allow the detection of errors which would not be detectable merely by viewing the bit stream. Examples and explanation are provided below.

2.9.1.2.1. LOOP TIME COUNTER

Each station has a Loop Time Counter (LTC) which is used to detect lost Tokens. A Token may be lost in many ways, for example a noise burst on the medium or station failure while a Token is resident at that station. A station's LTC is reset and begins to count down whenever that station views a Free Token (i.e. whenever a station claims, issues, or repeats a Free Token). Upon viewing a Claimed Token, the station stops its LTC. If the LTC expires before a Claimed Token is viewed, this indicates a lost Token, and Warm Start is initiated. The expiration time of the LTC is set equal to one loop time which is, of course, dependent upon the medium rate (and therefore is specified in the appropriate slash sheet to the protocol).

2.9.1.2.2. UNCONTROLLED TRANSMIT INHIBIT TIMER

Each station has an Uncontrolled Transmit Inhibit Timer (UTIT) which aborts transmission of a message of over 1.1 times the maximum message length of 4096 words. The UTIT affects only those transmissions which were originated by the station in which the timer is resident. If the UTIT does expire, the station is locked into repeat mode, and the host is notified. Only the station's host may release the station from this locked state (presumably once the problem is corrected).

2.9.1.2.3. TOKEN STARTING DELIMITER COUNTER

Each station has a Token Starting Delimiter Counter (TSDC) which is responsible for detecting lost Token Starting Delimiters (and, hence, lost Tokens) in the cases which are not detectable by the LTC. The counter is reset upon viewing a valid Token Starting Delimiter. Upon being reset, the TSDC counts the number of bits until viewing the next TSD. If this value exceeds the maximum value (96K bits, to allow margin for error), then the TSD (i.e. the Token) is assumed to be lost, and Warm Start is initiated.

2.9.1.2.4. LOST FREE TOKEN COUNTER

Each station has a counter which counts the number of Token Starting Delimiters viewed by the station. This counter is reset to zero upon viewing a message which has a different Source Address than the previous message, or upon viewing a Free Token. If the value of this counter exceeds two, Warm Start is initiated. Thus, if a Free Token is lost but the TSD remains on the ring, the condition will be detected.

2.9.2. HARDWARE FAULTS

One may conceive of innumerable ways in which the hardware comprising the network may be damaged. One of the goals of the HSRB design committee was to create as fault-tolerant a network as possible. One of the ways in which this is done is the use of counter-rotating ring pairs (nominally one pair, although the protocol allows any number of rings to be used). Since the rings rotate in opposite directions, it is possible not only to use each ring as a separate data channel, but to create more flexible and creative configurations. The ways in which the HSRB may reconfigure to circumvent faults, both in a station and in the medium, are discussed below.

2.9.2.1. STATION FAULTS

Before attempting to join the network each station performs a self-test sequence to ensure proper operation. If the station is determined to be nonfunctional or to function incorrectly, it does not attempt to enter the ring, but rather behaves as described in Section 2.6. If, however, the station fails at some point subsequent to its inclusion in the ring, it is bypassed (see Figure 2.5). In effect, it becomes invisible to the ring, neither receiving nor transmitting data. The protocol requires this bypass to be the default state (i.e. the station is bypassed whenever power is not applied to the Ring Interface Unit). Thus, station failure will not cause a discontinuity in the network medium.

2.9.2.2. MEDIUM FAULTS

The physical connections comprising a network's links may become disabled for any number of reasons, including environmental factors and intentional or inadvertent human interference. Part of the process by which the network routes around faults is called Loopback Mode. A station which is in Loopback Mode effectively links the input of one ring to the output of another (see Figure 2.6). With this capability, the network may use the routing information gathered by the Beacon Frames at Reconfiguration to establish a ring of maximal size. Figures 2.7 through 2.10 illustrate how the network might reconfigure to route around various faults. Note that some configurations may result in the formation of separated rings. Even though this might not be the ideal solution in certain applications, it is certainly to be preferred to total network failure. Of course, in particularly crucial applications more redundancy may be created by the inclusion of additional rings.

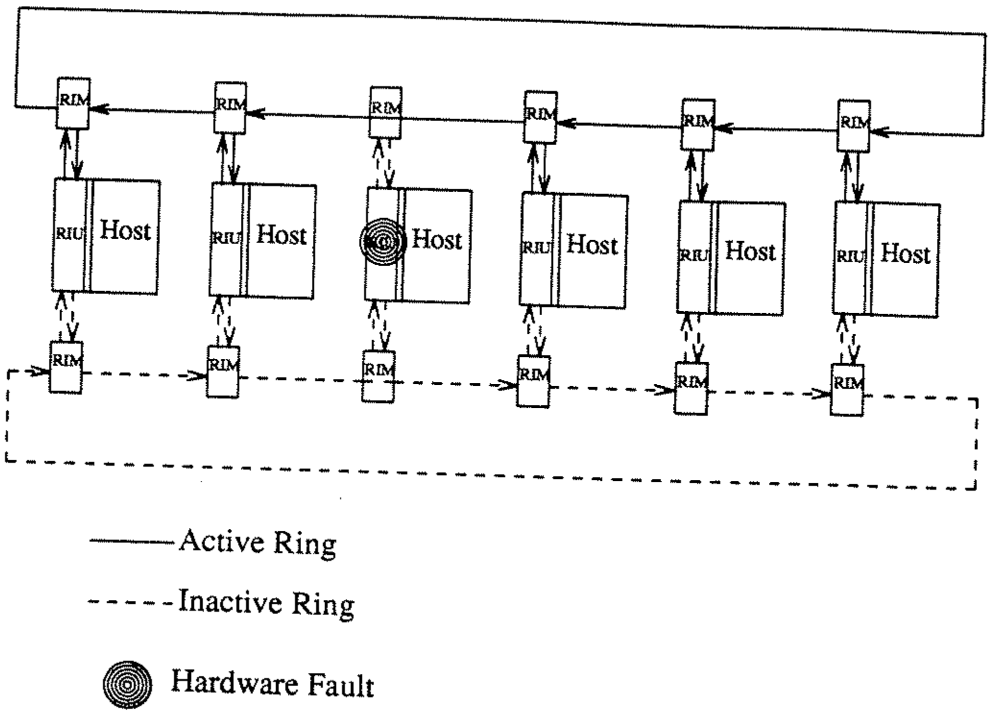


Figure 2.5 — Station Bypass

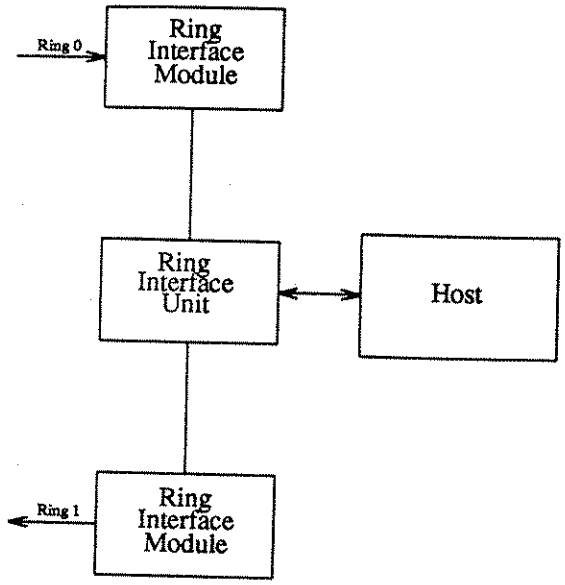


Figure 2.6 — Station with Loop Back

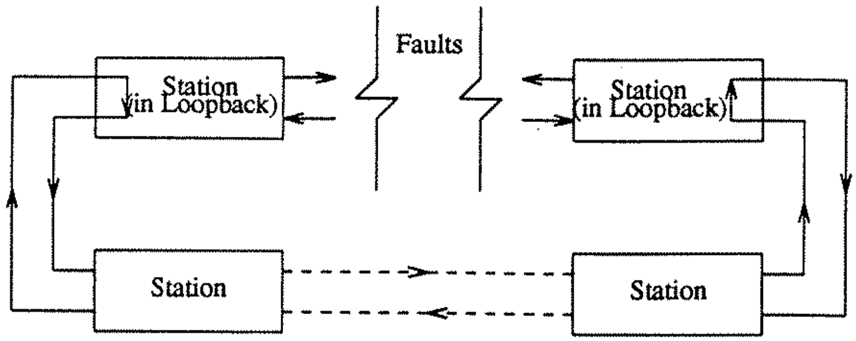


Figure 2.7 — Loop Back Excluding a Fault

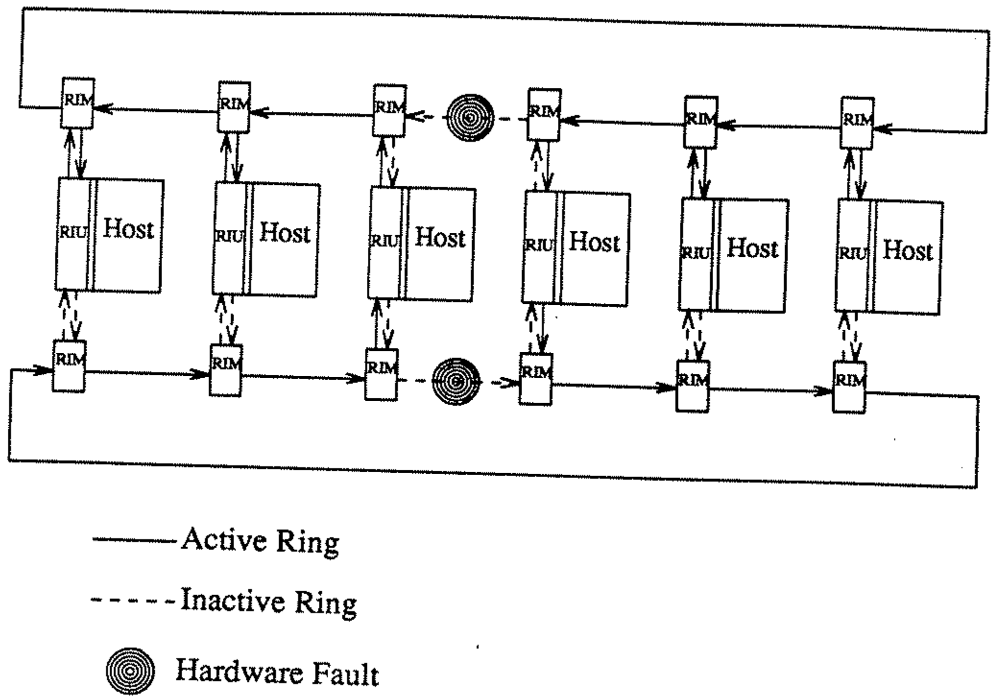


Figure 2.8 — Loopback With All Stations On Same Ring

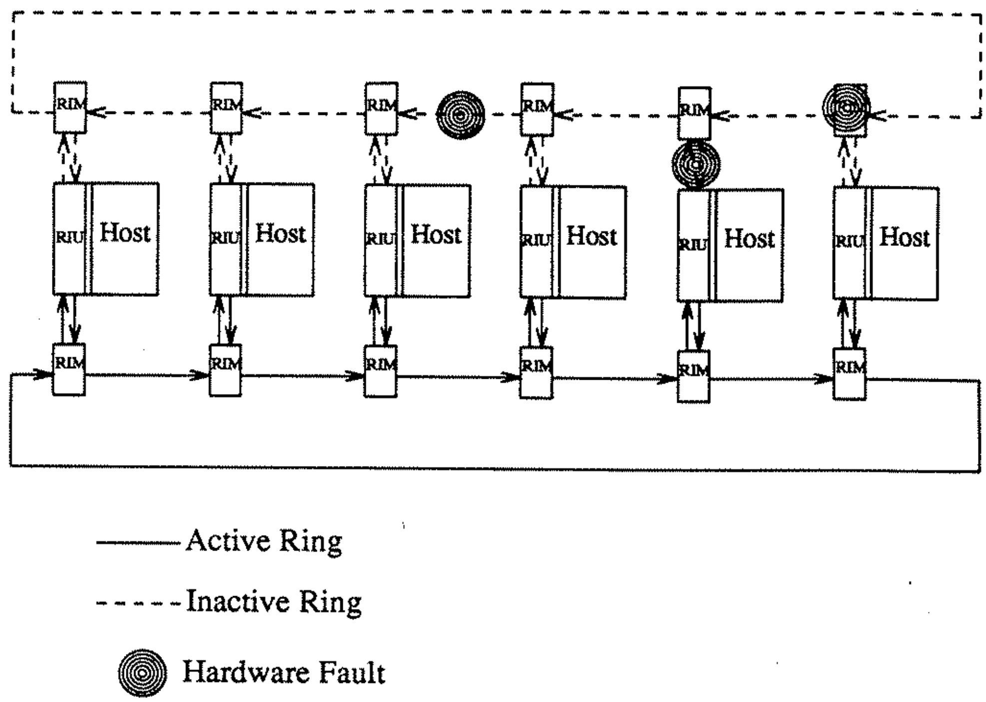


Figure 2.9 — Use of Redundant Ring

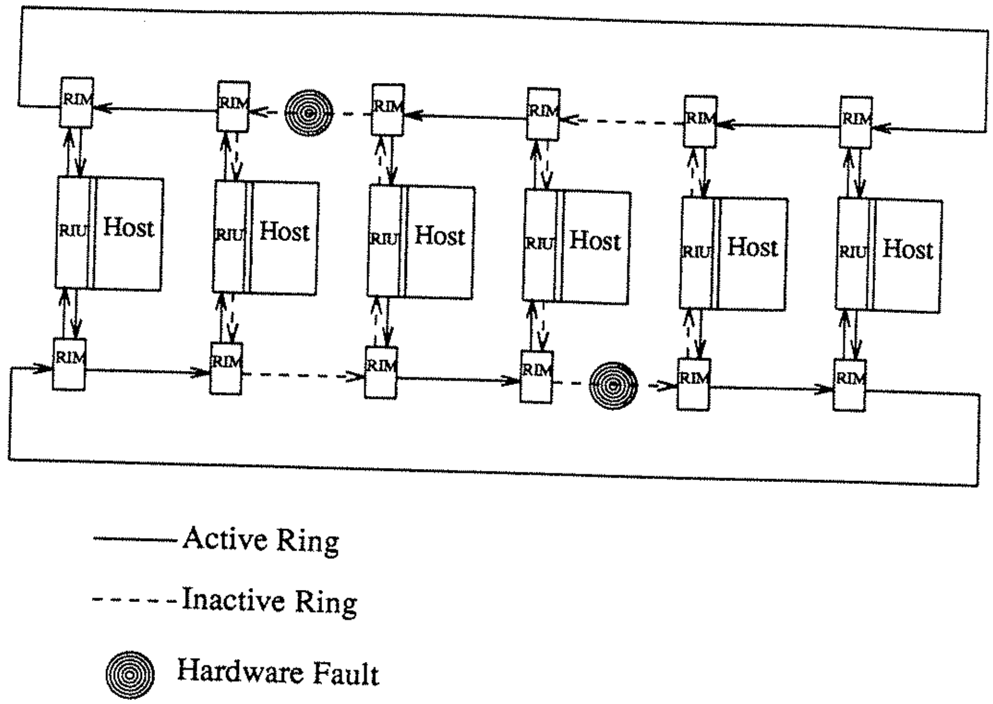


Figure 2.10 — Loopback Forming Separated Ring Networks

2.10. PROTOCOL FRAME FORMATS

Three Protocol Data Units (PDUs) are defined for the HSRB: Token Frames, Message Frames, and Beacon Frames. In the following diagrams, order of transmission is leftmost field first.

2.10.1. TOKEN FRAME

The Token Frame is the control mechanism by which the right to transmit a message (as opposed to merely repeating a message) is transferred from station to station. A Token Frame (also called a Token) may be either "Free" or "Claimed". When in the "Free" state a Token may be claimed by any station having a message to send (subject to the priority rules). A Free Token consists of a Token Starting Delimiter field, a Control field, and a Token Ending Delimiter field. Once claimed, the Token's ending delimiter is removed, and the message is appended.

TSD	CON	TED

TSD Token Starting Delimiter (10 code bits)

CON Control (20 code bits)

TED Token Ending Delimiter (5 code bits)

NOTE: TED is removed by the transmitting station in a Claimed Token

2.10.1.1. TOKEN STARTING DELIMITER (TSD) FIELD

The Token Starting Delimiter field indicates the start of a Token, and consists of a J symbol followed by a K symbol. This symbol sequence is to be recognized regardless of the bit pattern preceding it.

J	K

J: J symbol

K: K symbol

2.10.1.2. CONTROL (CON) FIELD

The Control field consists of five subfields: the Priority (PR) subfield, the Token Status 1 (TS1) subfield, the Short Message Count (SMC) subfield, the Token Status 2 (TS2) subfield, and the Reservation (RES) subfield.

PR	TS1	SMC	TS2	RES

PR Priority (5 code bits)

TS1 Token Status 1 (2 code bits)

SMC Short Message Count (6 code bits)

TS2 Token Status 2 (2 code bits)

RES Reservation (5 code bits)

2.10.1.2.1. PRIORITY (PRI) SUBFIELD

The Priority subfield indicates the priority of the Token and consists of code bits as follows:

1	P_2	P_1	1	P_0

P_2 Priority bit 2 (most significant)

P_1 Priority bit 1

P_0 Priority bit 0 (least significant)

Priority 0 is the highest priority and priority 7 is the lowest.

2.10.1.2.2. TOKEN STATUS 1 (TS1) SUBFIELD

The Token Status 1 subfield shall indicate whether the Token is Claimed or Free. It consists of code bits as follows:

1	T_1

$T_1 = 0$ indicates Claimed Token

$T_1 = 1$ indicates Free Token

2.10.1.2.3. SHORT MESSAGE COUNT (SMC) SUBFIELD

The Short Message Count subfield indicates the number of consecutive short messages transmitted, not including the message following the Claimed Token subframe. Its value may range from 0 to 15, 0 indicating the lowest count, and it consists of code bits as follows:

S_3	1	S_2	S_1	1	S_0

S_3 SMC bit 3 (most significant)

S_2 SMC bit 2

S_1 SMC bit 1

S_0 SMC bit 0 (least significant)

2.10.1.2.4. TOKEN STATUS 2 (TS2) SUBFIELD

The Token Status 2 subfield indicates whether the Token is Claimed or Free. Under error-free operation T2 shall always equal T1 of the TS1 subfield. The TS2 subfield consists of code bits as follows:

T_2	1

$T_2 = 0$ indicates Claimed Token

$T_2 = 1$ indicates Free Token

2.10.1.2.5. RESERVATION (RES) SUBFIELD

The Reservation subfield indicates the highest reservation made (in the previous cycle of the ring) by a station with a message to send. A value of 0 indicates the highest priority and a value of 7 indicates the lowest priority. It consists of code bits as follows:

R_2	1	R_1	1	R_0

R_2 Reservation bit 2 (most significant)

R_1 Reservation bit 1

R_0 Reservation bit 0 (least significant)

2.10.1.3. TOKEN ENDING DELIMITER (TED) FIELD

The Token Ending Delimiter field indicates the end of a Token and consists of a T symbol.

T

T: Terminate symbol

2.10.2. MESSAGE FRAME

A Message Frame is the PDU used to transmit data. It consists of the Token Frame which was claimed by the station (with its Token Ending Delimiter deleted) followed by 14 subfields. The subfields of a Message Frame are:

TSD	CON	PA	MFSD	PRS	WC	SA	AC	GA	DA	MCFCS

INFO	IFCS	MFED	FS

TSD Token Starting Delimiter (10 bits)

CON Control (20 bits)

PA Preamble (20 bits)

MFSD Message Frame Starting Delimiter (10 bits)

PRS Priority and Retry Status (5 bits)

WC Word Count (15 bits)

SA Sending Address (10 bits)

AC Address Control (5 bits)

GA Global Address (5 bits)

DA Destination Address (20, 40, 60, or 80 bits)

MCFCS Message Control FCS (20 bits)

INFO Information (4 to 16,384 symbols in multiples of 4 symbols -- 20 to 81,920 bits)

IFCS Information FCS (20 bits)

MFED Message Frame Ending Delimiter (5 bits)

FS Frame Status (15 bits)

2.10.2.1. PREAMBLE (PA) FIELD

The Preamble field consists of four IDLE symbols.

I	I	I	I

I: I symbol

2.10.2.2. MESSAGE FRAME STARTING DELIMITER (MFSD) FIELD

The Message Frame Starting Delimiter indicates the start of a Message Frame and consists of a J symbol followed by an A symbol.

J	A

J: J symbol

A: A symbol

2.10.2.3. PRIORITY AND RETRY STATUS (PRS) FIELD

The Priority and Retry Status field consists of two subfields: the Message Priority (PRM) subfield and the Retry Status Indicator (RSI) subfield.

PRM	RSI

PRM Message Priority (3 data bits)

RSI Retry Status Indicator (1 data bit)

2.10.2.3.1. MESSAGE PRIORITY (PRM) SUBFIELD

The Message Priority subfield indicates the priority of the current message with priority 0 being the highest priority and priority 7 being the lowest. It is composed of data bits as follows:

PM_2	PM_1	PM_0

PM_2 Priority bit 2 (most significant)

PM_1 Priority bit 1

PM_0 Priority bit 0 (least significant)

2.10.2.3.2. RETRY STATUS INDICATOR (RSI) SUBFIELD

The Retry Status Indicator subfield indicates whether or not the message is an original transmission or a retry. It consists of a data bit as follows:

RSI

RSI = 0 indicates that the Message Frame is not a retry

RSI = 1 indicates that the Message Frame is a retry

2.10.2.4. WORD COUNT (WC) FIELD

The Word Count field consists of 12 4-bit symbol groups and indicates the number of words in the Information (INFO) field. Its value will be between 1 and 4096 words inclusive. It consists of data bits as follows:

WC_{11}	--	WC_0

WC_{11} Word Count bit 11 (most significant)

WC_0 Word Count bit 0 (least significant)

$$WC = \begin{cases} i & indicates\ i\ words, \quad 1 \le i \le 4095 \\ 0 & indicates\ 4096\ words \end{cases}$$

2.10.2.5. SENDING ADDRESS (SA) FIELD

The Sending Address field indicates the station which sent the current message. It consists of eight data bits as follows:

0	SA_6	--	SA_0

SA_6 Sending Address bit 6 (most significant)

SA_0 Sending Address bit 0 (least significant)

2.10.2.6. ADDRESS CONTROL (AC) FIELD

The Address Control field consists of three subfields: the Logical-Physical (LP) subfield, the Address Word Count (AWC) subfield, and the Global Address (GA) subfield.

LP	AWC	GA

2.10.2.6.1. LOGICAL-PHYSICAL (LP) SUBFIELD

The Logical-Physical subfield indicates the addressing mode and consists of data bits as follows:

0	LP

LP = 0 indicates Physical Address Mode

LP = 1 indicates Logical Address Mode

2.10.2.6.2. ADDRESS WORD COUNT (AWC) SUBFIELD

The Address Word Count indicates the number of logical address words transmitted in this message (from 1 to 4). If the message uses Physical Address Mode then the value of AWC is transmitted as 0. AWC consists of data bits as follows:

AW_1	AW_0

AW_1 Address Word Count bit 1 (most significant)

AW_0 Address Word Count bit 0 (least significant)

AWC = 0 indicates 2^{16} logical address space

AWC = 1 indicates 2^{32} logical address space

AWC = 2 indicates 2^{48} logical address space

AWC = 3 indicates 2^{64} logical address space

2.10.2.7. GLOBAL ADDRESS (GA) FIELD

The Global Address field indicates for which ring(s) the message is intended. It consists of either the Global Address Physical (GAP) subfield (in Physical Address Mode) or the Global Address Logical (GAL) subfield (in Logical Address Mode).

2.10.2.7.1. GLOBAL ADDRESS PHYSICAL (GAP) SUBFIELD

The Global Address Physical subfield indicates for which ring the message is intended (in Physical Address Mode). It consists of data bits as follows:

BA	0	GP_1	GP_0

BA Bridge Access bit

GP_1 Global Address Physical bit 1 (most significant)

GP_0 Global Address Physical bit 0 (least significant)

BA = 0, GP = 0 indicates transmission on ring of originating station

BA = 1, GP = n indicates transmission on destination ring n

BA = 0, GP > 0 invalid, not to be transmitted

2.10.2.7.2. GLOBAL ADDRESS LOGICAL (GAL) SUBFIELD

The Global Address Logical subfield indicates for which ring(s) the message is intended (in Logical Address Mode). It consists of data bits as follows:

GL_3	GL_2	GL_1	GL_0

GL_3 Global Address Logical bit 3

GL_2 Global Address Logical bit 2

GL_1 Global Address Logical bit 1

GL_0 Global Address Logical bit 0

$GL_n = 1$ indicates that the Message Frame is intended for ring n

$GL_n = 0$ indicates that the Message Frame is not intended for ring n

2.10.2.8. DESTINATION ADDRESS (DA) FIELD

The Destination Address field indicates for which station(s) the message is intended. It consists of either the Destination Address Physical (DAP) subfield (in Physical Address Mode) or the Destination Address Logical (DAL) subfield (in Logical Address Mode).

2.10.2.8.1. DESTINATION ADDRESS PHYSICAL (DAP) SUBFIELD

The Destination Address Physical subfield indicates for which station the message is intended (in Physical Address Mode). It consists of data bits as follows:

DP_{15} Physical Address bit 6 (most significant)

DP_9 Physical Address bit 0 (least significant)

DP_{15}	--	DP_9	DP_8	--	DP_0

DP_8 Sub-Address bit 8 (most significant)

DP_0 Sub-Address bit 0 (least significant)

2.10.2.8.2. DESTINATION ADDRESS LOGICAL (DAL) SUBFIELD

The Destination Address Logical subfield indicates for which station(s) the message is intended (in Logical Address Mode). It consists of data bits as follows:

$DL_{AWC_{15}}$	--	DL_{AWC_0}	--	$DL_{(AWC-1)_{15}}$	--	$DL_{(AWC-1)_0}$	--

$DL_{0_{15}}$	--	DL_{0_0}

$DL_{AWC_{15}}$ Destination Address Logical (most significant word, most significant bit)

DL_{AWC_0} Destination Address Logical (most significant word, least significant bit)

$DL_{(AWC-1)_{15}}$ Destination Address Logical (second most significant word, most significant bit)

$DL_{(AWC-1)_0}$ Destination Address Logical (second most significant word, least significant bit)

$DL_{0_{15}}$ Destination Address Logical (least significant word, most significant bit)

DL_{0_0} Destination Address Logical (least significant word, least significant bit)

If a station is capable of receiving a logical address of up to N words, then it will not recognize addresses of more than N words. If an address of less than N words is received, then the words received are treated as the low-order bits of the address.

2.10.2.9. MESSAGE CONTROL FRAME CHECK SEQUENCE (MCFCS) FIELD

The Message Control Frame Check Sequence field covers the PRS, WC, SA, AC, GA, DA, and MCFCS fields and uses the generator polynomial:

$$G(X) = X^{16} + X^{12} + X^5 + 1$$

The field consists of data bits as follows:

$MCFCS_{15}$	--	$MCFCS_0$

$MCFCS_{15}$ Message Control Frame Check Sequence bit 15 (most significant)

$MCFCS_0$ Message Control Frame Check Sequence bit 0 (least significant)

2.10.2.10. INFORMATION (INFO) FIELD

The Information field contains that information to be transmitted in the message. It consists of Information Word (IW) and Adjustment (ADJ) subfields.

IW_N	--	IW_{N-256}	ADJ	--	IW_0

IW_N Information Word N

IW_{N-256} 256th Transmitted Information Word

IW_0 Information word 0

The structure of the INFO field is such that an ADJ subfield is inserted at intervals of 256 IW subfields. If the number of words in the message is not a multiple of 256, then the last "block" of words is not followed by an ADJ subfield.

2.10.2.10.1. INFORMATION WORD (IW) SUBFIELD

Each Information Word subfield consists of sixteen Information Bits, the most significant being IB_{15} and the least significant being IB_0.

IB_{15}	--	IB_0

2.10.2.10.2. ADJUSTMENT (ADJ) SUBFIELD

As stated above, the adjustment subfield occurs at intervals of 256 Information Word subfields within the Information field. It is initially transmitted as six IDLE symbols followed by a J symbol followed by an A symbol.

I	I	I	I	I	I	J	A

I: I symbol

J: J symbol

A: A symbol

The ADJ subfield may be adjusted in length by any station to allow initialization of the elastic buffer.

2.10.2.11. INFORMATION FRAME CHECK SEQUENCE (IFCS) FIELD

The Information Frame Check Sequence covers the INFO and IFCS fields and uses the generator polynomial:

$$G(X) = X^{16} + X^{12} + X^5 + 1$$

The field consists of data bits as follows:

$IFCS_{15}$	--	$IFCS_0$

$IFCS_{15}$ Information Frame Check Sequence bit 15 (most significant)

$IFCS_0$ Information Frame Check Sequence bit 0 (least significant)

2.10.2.12. MESSAGE FRAME ENDING DELIMITER (MFED) FIELD

The Message Frame Ending Delimiter field indicates the end of the INFO field. It consists of a T symbol.

T

2.10.2.13. FRAME STATUS (FS) FIELD

The Frame Status field consists of the following subfields:

MCED1 Message Control Error Detected 1

ACK1 Message Acknowledged 1

RCVD1 Message Received 1

IED1 Information Error Detected 1

MCED2 Message Control Error Detected 2

ACK2 Message Acknowledged 2

RCVD2 Message Received 2

IED2 Information Error Detected 2

MCED1	ACK1	RCVD1	IED1	MCED2	ACK2	RCVD2	IED2

2.10.2.13.1. MESSAGE CONTROL ERROR DETECTED 1 (MCED1) SUBFIELD

The Message Control Error Detected 1 subfield indicates whether any difference was found between the received and calculated values of the MCFCS. It consists of two code bits as follows:

MCED	1

MCED = 0 indicates no error detected

MCED = 1 indicates error detected

2.10.2.13.2. MESSAGE ACKNOWLEDGED 1 (ACK1) SUBFIELD

The Message Acknowledged 1 subfield indicates that the message was received, found to be error free, and has been made available to the host. It consists of one code bit as follows:

ACK

ACK = 0 indicates negative acknowledgement

ACK = 1 indicates positive acknowledgement

2.10.2.13.3. MESSAGE RECEIVED 1 (RCVD1) SUBFIELD

The Message Received 1 subfield indicates that a station has received a message containing its address. It consists of two code bits as follows:

RCVD	1

RCVD = 0 indicates not received

RCVD = 1 indicates received

2.10.2.13.4. INFORMATION ERROR DETECTED 1 (IED1) SUBFIELD

The Information Error Detected 1 subfield indicates whether any difference was found between the received and calculated values of the IFCS. It consists of two code bits as follows:

IED	1

IED = 0 indicates no error detected

IED = 1 indicates error detected

2.10.2.13.5. MESSAGE CONTROL ERROR DETECTED 2 (MCED2) SUBFIELD

The Message Control Error Detected 2 subfield indicates whether any difference was found between the received and calculated values of the MCFCS. Under error-free operation MCED2 will be identical to MCED1. It consists of one code bit as follows:

MCED

MCED = 0 indicates no error detected

MCED = 1 indicates error detected

2.10.2.13.6. MESSAGE ACKNOWLEDGED 2 (ACK2) SUBFIELD

The Message Acknowledged 2 subfield indicates that the message was received, found to be error free, and has been made available to the host. Under error-free operation ACK2 will be identical to ACK1. It consists of two code bits as follows:

ACK	1

ACK = 0 indicates negative acknowledgement

ACK = 1 indicates acknowledgement

2.10.2.13.7. MESSAGE RECEIVED 2 (RCVD2) SUBFIELD

The Message Received 2 subfield indicates that a station has received a message containing its address. Under error-free operation RCVD2 will be identical to RCVD1. It consists of two code bits as follows:

RCVD	1

RCVD = 0   indicates not received

RCVD = 1   indicates received

2.10.2.13.8. INFORMATION ERROR DETECTED 2 (IED2) SUBFIELD

The Information Error Detected 2 subfield indicates whether any difference was found between the received and calculated values of the IFCS. Under error-free operation IED2 will be identical to IED1. It consists of three code bits as follows:

IED	1	1

IED = 0   indicates no error detected

IED = 1   indicates error detected

2.10.3. BEACON FRAME

The Beacon Frame is used to transmit network control information during system start-up and reconfiguration. It consists of the Beacon Frame Starting Delimiter (BFSD) field, the Beacon Control (BCON) field, the Highest Known Address (HKA) field, the Station Count (SC) field, the Beacon Frame Check Sequence (BFCS) field, and the Beacon Frame Ending Delimiter (BFED) field.

2.10.3.1. BEACON FRAME STARTING DELIMITER (BFSD) FIELD

The Beacon Frame Starting Delimiter indicates the start of a Beacon Frame. It consists of a K symbol followed by a J symbol (note that this is the inverse of the JK pair which forms the Starting Delimiter for a normal message).

K	J

K: K symbol

J: J symbol

2.10.3.2. BEACON CONTROL (BCON) FIELD

The Beacon Control field consists of the Beacon Type (BT) subfield and the Beacon Path Indicator (BPI) subfield.

BT	BPI

BT Beacon Type (3 bits)

BPI Beacon Path Indicator (1 data bit)

2.10.3.2.1. BEACON TYPE (BT) SUBFIELD

The Beacon Type subfield indicates the type of Beacon transmitted. It consists of data bits as follows:

BT_2	BT_1	BT_0

BT = 0 indicates Warm Start

BT = 1 indicates Warm Recover

BT = 2 indicates Restart

BT = 3 indicates Vie

BT = 4 indicates Configure Ring 0

BT = 5 indicates Configure Ring 1

BT = 6 indicates Configure Loopback

BT = 7 reserved

2.10.3.2.2. BEACON PATH INDICATOR (BPI) SUBFIELD

The Beacon Path Indicator field indicates whether or not the path by which the Highest Known Address (HKA) data has been received contains exclusively Ring 0 or Ring 1 links. It consists of data bits as follows:

BPI

BPI = 0 indicates that the HKA data path has included both Ring 0 and Ring 1 links

BPI = 1 indicates that the HKA data path has included exclusively Ring 0 or Ring 1 links

2.10.3.3. HIGHEST KNOWN ADDRESS (HKA) FIELD

The Highest Known Address field indicates the highest station address known to be active by the station transmitting the Beacon Frame. It consists of eight data bits as follows:

0	HKA_6	--	HKA_0

HKA_6 Highest Known Address bit 6 (most significant)

HKA_i Highest Known Address bit i

HKA_0 Highest Known Address bit 0 (least significant)

2.10.3.4. STATION COUNT (SC) FIELD

The Station Count field indicates the number of stations included in a complete ring. It consists of data bits as follows:

0	SC_6	--	SC_0

SC_6 Station Count bit 6 (most significant)

SC_0 Station Count bit 0 (least significant)

2.10.3.5. BEACON FRAME CHECK SEQUENCE (BFCS) FIELD

The Beacon Frame Check Sequence field covers the BCON, HKA, SC, and BFCS fields and uses the generator polynomial: $G(X) = X^{16} + X^{12} + X^5 + 1$

The field consists of data bits as follows:

$BFCS_{15}$	--	$BFCS_0$

$BFCS_{15}$ Beacon Frame Check Sequence bit 15 (most significant)

$BFCS_0$ Beacon Frame Check Sequence bit 0 (least significant)

2.10.3.6. BEACON FRAME ENDING DELIMITER (BFED) FIELD

The Beacon Frame Ending Delimiter field indicates the end of a Beacon Frame and consists of the T symbol.

T

T: Terminate symbol

Chapter 3

JUSTIFICATION OF SIMULATION

3.1. INTRODUCTION

There are several methods which are typically used to analyze a network protocol. One may gather performance measurements of a particular implementation of the protocol, construct a mathematical model, or use simulation to model the protocol. All of these methods have both advantages and disadvantages. For instance, while performance measurements are the most accurate predictor of the performance a network designer might expect, they perforce require an existing implementation of the protocol. Hence, this technique is not always applicable, especially during the protocol specification process. Mathematical modeling, on the other hand, may be done while the protocol is still being designed. Such a model may state in very precise terms how the protocol operates. However, in order to derive such a model, one may often have to make simplifying assumptions about the protocol's operation. Hence, the mathematical model is limited by its inability to capture all the subtleties of a complex protocol. Simulation, on the other hand, can capture many of the intricate details of protocol operation. Furthermore, since it does not require an existing implementation, simulation may be used as a design tool. Individual network parameters may be changed in order to observe the effect these changes have on the operation of the protocol. Unfortunately, as with any large computer program, complete verification of the correctness of the simulation is generally not possible. In order to generate confidence in the simulation's results, some effort must be given towards validation. Also, because simulation may not capture subtle interactions, and because simulations also make simplifying assumptions, simulation results tend to generate optimistic values, and hence should be used as "best case" statistics.

In conducting a performance analysis of the HSRB protocol, it was decided to use simulation as the modeling method. Since the protocol was still in transition throughout the analysis, there was necessarily no existing implementation from which to make performance measurements. The remainder of this chapter presents the simulation environment which was used for the project and the justification of the simulator's results.

3.2. THE SIMULATOR

The simulation program which was used to gather the HSRB performance data is part of a larger simulation system designed by the author and Randall Simonson [SIMO88]. This encompassing simulation environment is designed to facilitate comparisons between token ring protocols by establishing a generic user interface. This interface allows the user to configure simulations of both the SAE HSRB and the ANSI FDDI protocols in a similar manner. The three phases of the simulation process are:

1) user input and network configuration,

2) computation (the actual simulation), and

3) output.

These will be explained in further detail below.

The simulation system is written in Pascal, and was developed in a 4.2 BSD Unix environment. It consists of approximately 5000 lines of Pascal code, of which about 3200 implement the user interface.

3.2.1. INPUT PHASE

The input phase of the simulation allows the user to define the parameters which determine network configuration. The user is led through a menu driven system which helps him tailor the configuration. All input values are checked to ensure that they meet the specifications of the protocol; invalid values are rejected. The input phase is further divided into three parts: 1) the input of parameters which are common to token ring protocols; 2) the input of protocol-specific parameters; and 3) the input of those parameters which define classes of stations within the network.

3.2.1.1. GLOBAL NETWORK PARAMETERS

Those parameters which are common to token ring networks include transmission rate, ring circumference, number of stations, station latency, simulation time, and others. These are parameters which determine the physical characteristics of the ring, or which are applicable to all stations. The input screen for these parameters is shown in Figure 3.1. Note that some of these variables are assigned fixed or limited

```
Enter Global Network Parameters
        1-   Number of stations in the ring: 100
        2- Transmission rate (in Mbits/sec): 80.000
        3-            Ring Length (in meters): 1000
        4-   Station latency (in bit times): 6
        5- Time to run simulation (seconds): 0.100
        6-     Number of Classes of Stations: 1
        7-Number of Asynchronous Priorities: 8
        8-          Offered load  (Enter %) : 85
        9-  Length of Latency Buffer (bits): 40
Enter number of field to change [0 to exit]:
```

Figure 3.1 — Global Network Parameters Input Screen

values by the HSRB protocol (e.g. number of priorities). Since the interface is to be used for various protocol simulators, this generality is necessary. Where required, the input values are range-checked to ensure conformance with the HSRB protocol.

3.2.1.2. PROTOCOL-SPECIFIC PARAMETERS

The second portion of the input phase deals with protocol-specific parameters. In the case of the SAE HSRB, the only such value is a boolean switch determining whether or not the short message protocol is implemented. Conversely, if the user wishes to simulate FDDI, another entire screen of data must be entered at this point. It deserves mention that for each menu, the user has the option of either entering the values anew or editing a set of system defaults. Once edited, these new values may be saved as the current defaults. This increases the automation of the input process, and decreases the amount of typing required to configure a simulation.

3.2.1.3. STATION CLASS PARAMETERS

The last portion of the input phase consists of defining classes of stations. A class consists of one or more stations with identical traffic characteristics. For each of the number of classes specified in the global network parameters menu the following information is collected:

1) Class name,

```
(1)                                    Class name: sae1
(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)   12        (5)   1       (6)   128       (7)   1
     1           (12)   12       (13)   1      (14)   128      (15)   1
     2           (12)   12       (13)   1      (14)   128      (15)   1
     3           (16)   12       (17)   1      (18)   128      (19)   1
     4           (20)   12       (21)   1      (22)   128      (23)   1
     5           (24)   12       (25)   1      (26)   128      (27)   1
     6           (28)   12       (29)   1      (30)   128      (31)   1
high 7           (32)   16       (33)   1      (34)   128      (35)   1


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

Figure 3.2 — Station Class Parameters Input Screen

2) Number of stations in this class,

3) Percent of the total offered load which is contributed by this class, and

4) Traffic characteristics of each priority in this class.

The input screen for these values is illustrated in Figure 3.2. Note that the traffic characteristics of each priority consist of the percentage of this class' total contribution which is made by a given priority, the message arrival distribution (the mean message arrival rate may be derived from the load multipliers which have already been entered), the mean message length, and the distributions of lengths around that mean.

Classes may be saved in a library of default classes for later reuse. Again, such a default class may be loaded and edited if so desired.

3.2.2. COMPUTATION PHASE

The computation phase of the simulation performs the actual protocol modeling. This is done through event-driven simulation. Such a simulation defines a number of events (e.g. the enqueueing of a message for transmission, the receipt of a free token, or the completion of message transmission). Each event has associated with it a time at which it will occur, so at a given point in the simulation an event may be said to have occurred or to be still in the future. As simulation time elapses, one can follow the execution of these events, the inter-relation of which determines the operation of the simulator.

In particular, the HSRB simulator is based on two major events: the arrival of a free token at a station, and the arrival of a message at a station for transmission. These events are mutually independent, and the handling of each differs slightly.

The arrival of a free token at a station is a unique event -- at most one free token exists at a given time (under error-free operation), so it may be resident in at most one station. Hence, it is necessary only to know where and when the *next* free token will arrive. Message arrival, however, is of a totally different character. Obviously, messages may arrive simultaneously at many different stations, and each station may have many outstanding messages. Hence, some method of storing these events for future processing is necessary.

Therefore, the computation phase of the simulation is subdivided into two parts: 1) the pregeneration of all the message arrivals for the length of the simulation, and 2) the execution of the simulation body. The pregeneration phase includes the allocation of the data structures necessary to store the message arrivals, and the generation of the messages themselves. Each message has associated with it a time of arrival, hence at time t in the simulation messages with arrival times less than or equal to t may be said to have arrived and be awaiting transmission.

The execution of the simulation is driven by token-arrival events. Whenever a free token arrives at a station, the station checks its message queues to determine if a message is waiting to be sent. If so, and if according to the rules of the protocol this station may claim the token, then the message is sent and a free token is issued. Note that the arrival time of the next free token at the next station is determined solely by the time necessary to transmit one message of known length, reissue the token, and allow the token to propagate to the next station. This is easily calculated due to the deterministic nature of these values. This control loop continues until the simulation time reaches some predetermined value, at which point the network statistics are calculated, and the simulation enters its output phase.

3.2.3. OUTPUT PHASE

Upon completion of the computation phase, the simulation system enters an output phase to allow the user to view the results of this run. Initially the user is presented with a menu summarizing the conditions under which this simulation was run (see Figure 3.3). Additionally, the user is presented with several viewing options. These options include viewing those statistics collected on a class-wide basis (with one of the options being network-wide statistics) and viewing those statistics collected on a per-priority basis. The user has the further option of saving all the above data in a file for later perusal. These options are explained below.

3.2.3.1. CLASS-WIDE STATISTICS

In Figure 3.4 we see the screen displaying those statistics collected on a class-wide basis. Notice that one may view these statistics either for the entire network or for a particular station class. The statistics displayed include:

1) Packets sent (for the length of the 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 :

          Simulation Time in Seconds :  0.100
       Number of Stations in Network :  100
       Network Ring Length in Meters :  1000.000
             Station latency in Bits :  6
        Medium Rate in Bits per Sec. :  80000000.000

       Number of Classes of Stations :  1
        Number of Message Priorities :  8

  Short Message Protocol Implemented :  NO
              Latency Buffer (in bits) :  40
```

Figure 3.3 — Output Screen Menu

```
        GENERAL INFORMATION FOR :  THE ENTIRE NETWORK

             Number of stations :  100
      Percentage of offered load :  85

              Total 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

     Number of Short Messages Sent :  0
    Mean Number of Tokens Received :  1002.342
    Mean Number of Tokens Accessed :  54.398
       Mean Number of Packets Sent :  54.398
            Mean Token Cycle Time :  1.944e-05

         Maximum Token Cycle Time :  2.694e-04
     Maximum Arrival Queue Length :  4
    Maximum Residual Queue Length :  3

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

Figure 3.4 — Class-wide Statistics Output Screen

2) Achieved offered load in bits (the number of bits submitted to the network for transmission),

3) Fraction of ring capacity (the achieved offered load expressed as a percentage of the network bandwidth),

4) Bits sent and bits received,

5) Throughput (expressed as a percentage of the network bandwidth),

6) Mean number of free tokens *received* per station,

7) Mean number of free tokens *accessed* per station,

8) Mean number of packets sent per station,

9) Mean token cycle time,

10) Maximum token cycle time,

11) Number of short messages sent (if the short message protocol is implemented),

12) Maximum arrival queue and residual queue lengths (queue lengths at a given station at the arrival/departure of a free token from that station).

3.2.3.2. PRIORITY-WIDE STATISTICS

Figure 3.5 shows the screen displaying those statistics collected on a priority-wide basis. Again, one may choose a view which encompasses either the entire network or only those stations in a given class. The statistics collected by the simulator include:

1) Number of packets sent at this priority,

2) Mean queueing delay (the time a message spends in a priority queue waiting to get to the head of the queue),

3) Mean network access delay (the time a message spends at the head of the queue awaiting transmission),

4) Mean waiting delay (the sum of queueing delay and network access delay),

5) Mean service delay (the time from when a message is enqueued until the last bit of the message is received at the destination station),

6) Standard deviations of each of these delays,

7) Mean token arrival queue and residual queue lengths,

8) Number of short messages of this priority.

Note: the simulator collects statistics only on those messages which have been delivered by the end of the simulation.

3.3. JUSTIFICATION OF SIMULATION RESULTS

A simulator is potentially a very powerful tool, however great care must be taken to ensure the validity of its results. One must demonstrate that the metrics generated by the simulator are not merely products of the randomness inherent in the simulation process, but rather are statistically significant. This

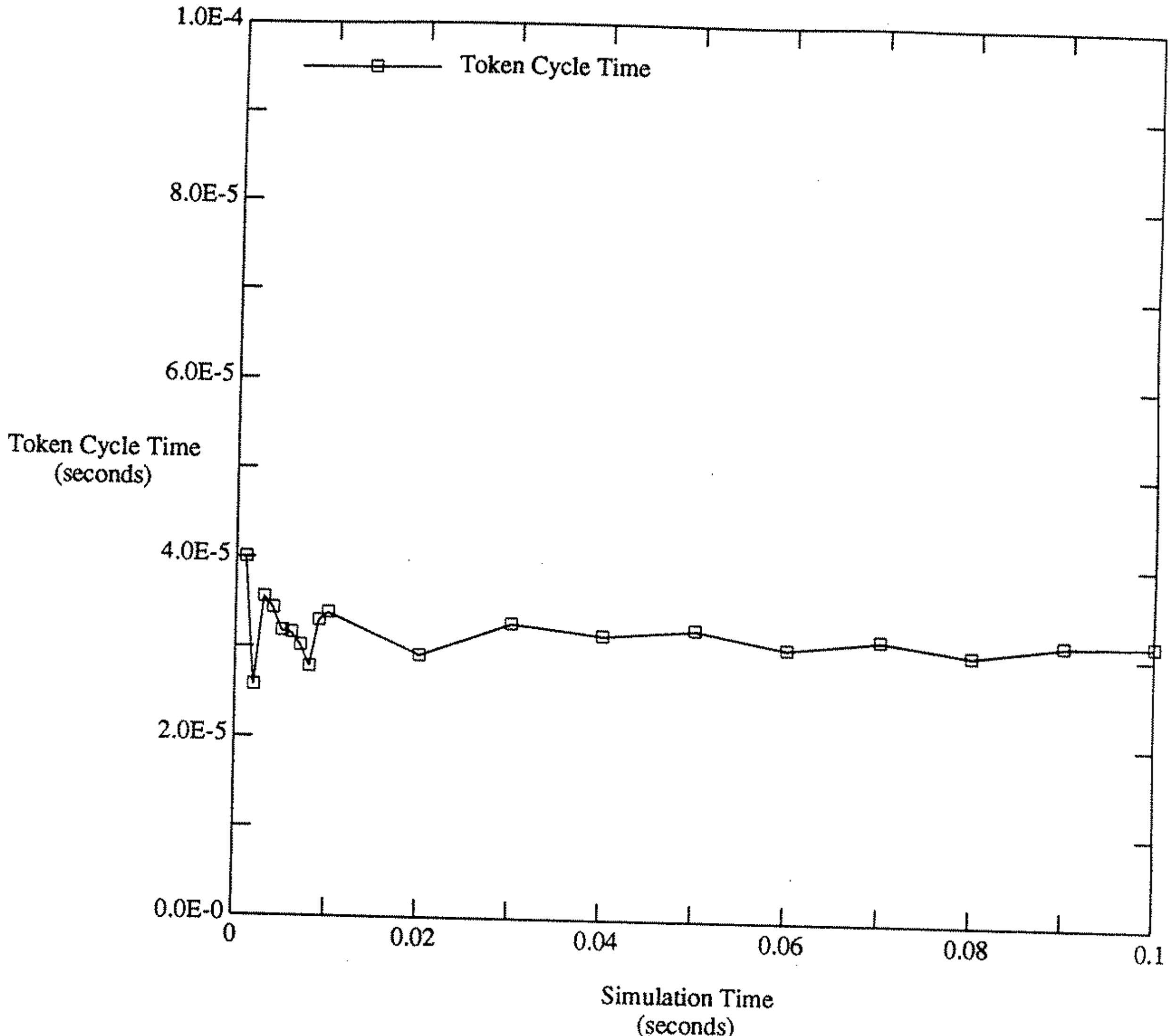
```
                PRIORITY INFORMATION FOR :  CLASS 1
                                PRIORITY :  7
                            Packets sent :  1014

          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 7, 9 to exit) :
      ENTER Class Number (0 for network information) :
```

Figure 3.5 — Priority-wide Statistics Output Screen

argument follows in three parts: 1) showing that the values of a given random variable are collected while the simulator is in a steady state, 2) showing that these same values lie within a given confidence interval of the expected value, and 3) comparing the simulation results with the results of an existing analytic model.

3.3.1. STEADY STATE ESTIMATION

As the simulator is run, it first experiences transient effects due to system startup. As the simulation continues, these startup effects become less significant, and the random variables in the simulation achieve some distribution about their expected values. When the simulator has reached this point it is said to be in steady state. In order to determine how long a simulation must run before it achieves this steady state, a

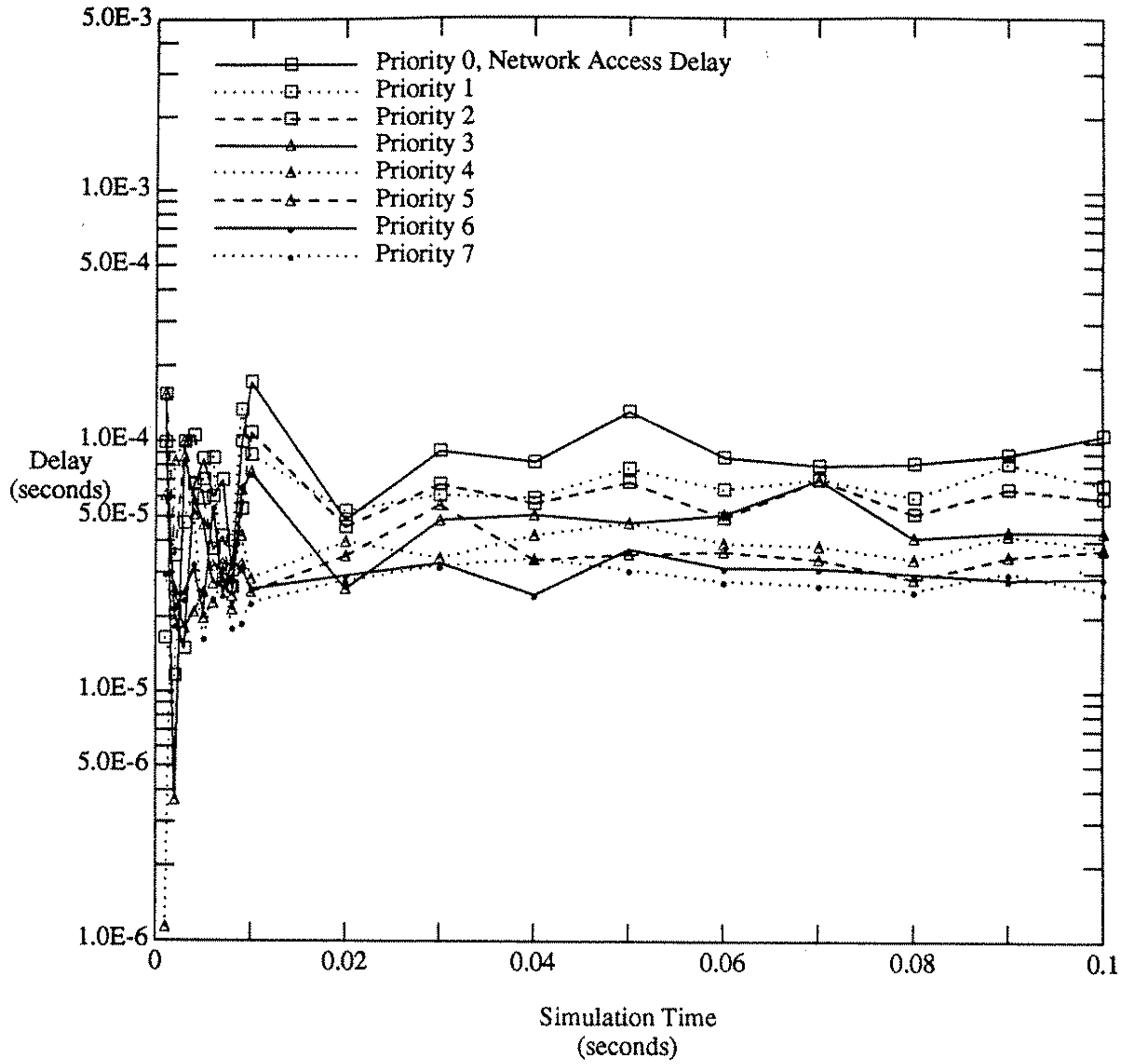


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 2056 Bit Mean Packet Length Distributed Exponentially
- 8 Priorities
- No Short Message Protocol

Figure 3.6 — Token Cycle Time

TIME DEPENDENCE



Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 2056 Bit Mean Packet Length Distributed Exponentially
- 8 Priorities
- No Short Message Protocol

Figure 3.7 — Network Access Delay

TIME DEPENDENCE

series of simulations were run with varying simulation times. Other simulation variables remained constant for all runs. Figures 3.6 and 3.7 show the results of these simulations. Figure 3.6 shows the graph of mean token cycle time vs. length of the simulation. Clearly, by the time 40 ms has elapsed, this value has achieved relative stability. Similarly, Figure 3.7 shows the graph of mean network access delay vs. length of the simulation. Again, within 40 ms the values for the various priorities have become relatively stable.

It should be noted that each point on these two graphs is the average of a number of internal results. For example, the mean token network access delay of priority 7 at 20 ms is the average of 60 internal results, whereas at 100 ms, it is the average of 343 results. It was decided, based on these figures and the graphs shown, that the length of time for which simulations would be run would be 100 ms, well within the steady state of the simulator. All the results shown in Chapter Four are derived from simulations of this length.

3.3.2. CONFIDENCE INTERVAL ESTIMATION

We have yet to show that the results produced by the simulator are bounded by a reasonable confidence interval. The argument for this is derived from [LAVE83].

Given a steady state random variable X (e.g. network access delay), suppose we have M independent observations, each generating a steady state output sequence of length N. We define

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

We know that X has a mean

$$\mu = E[X] \; . \tag{3.1}$$

As a point estimator of μ, we form the sample mean, $\hat{\mu}_m$, on each observation such that

$$\hat{\mu}_m = \frac{1}{N} \sum_{n=1}^{N} X_{mn} \tag{3.2}$$

Since each of the sample means is independent, this point estimator is itself a random variable.

$$E[\hat{\mu}_m] \approx \mu, \tag{3.3}$$

and

$$Var\left[\hat{\mu}_m\right] = \frac{\sigma^2}{M}. \tag{3.4}$$

Thus, a reasonable estimate of μ is $\hat{\mu}$, where

$$\hat{\mu} = \frac{1}{M}\sum_{m=1}^{M}\hat{\mu}_m \tag{3.5}$$

Note that the variance of $\hat{\mu}_m$ decreases as the sample size, M, increases. If we let

$$s^2(\hat{\mu}_m) = \frac{1}{M-1}\sum_{m=1}^{M}(\hat{\mu}_m - \hat{\mu})^2 = \frac{1}{M-1}\sum_{m=1}^{M}\hat{\mu}_m^2 - \frac{M}{M-1}\hat{\mu}^2 \tag{3.6}$$

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

$$\frac{\hat{\mu}-\mu}{s(\hat{\mu}_m)/M^{1/2}} \tag{3.7}$$

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, therefore, if we let

$$t_n(x) = (100 \cdot x)^{th} \textit{ percentile of the } t\textit{-distribution with } n \textit{ degrees of freedom}, \text{ and } 0 \le x \le 1, \tag{3.8}$$

we have

$$Prob\left\{ t_{M-1}\left[\frac{\alpha}{2}\right] \le \frac{\hat{\mu}-\mu}{s(\hat{\mu}_m)/M^{1/2}} \le t_{M-1}\left[1-\frac{\alpha}{2}\right] \right\} = 1-\alpha. \tag{3.9}$$

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

$$t_{M-1}\left[\frac{\alpha}{2}\right] = -t_{M-1}\left[1-\frac{\alpha}{2}\right] \tag{3.10}$$

Substituting equation 3.10 into equation 3.9 and solving the inequality for μ yields

$$Prob\ \{\ \hat{\mu} - t_{M-1}\left[1-\left[\frac{\alpha}{2}\right]\right] s(\hat{\mu}_m)/M^{1/2} \le \mu \le \hat{\mu} + t_{M-1}\left[1-\left[\frac{\alpha}{2}\right]\right] s(\hat{\mu}_m)/M^{1/2}\ \} \approx 1-\alpha. \tag{3.11}$$

Figure 3.8 shows curves of $t_n(1-\alpha/2)$ for $1-\alpha = 0.90$ and $1-\alpha = 0.95$. These curves approach the usual multipliers 1.64 and 1.96 of the normal distribution with zero mean and unit variance for large n. We can see that the function t_n $(1-\alpha/2)$ is approximately constant for $n \geq 10$. Hence, averaging more than 10 independent observations of the point estimator makes no appreciable difference in the expected width of the confidence interval.

There are several ways of generating sequences of variables, perhaps the easiest of which is repeated running of the simulator. To guarantee that runs are independent, the random number generator is seeded with the value of the system clock upon initialization of each run. Since ten such runs appeared to be sufficient to generate a reasonable confidence interval, all the data presented henceforth will be the average

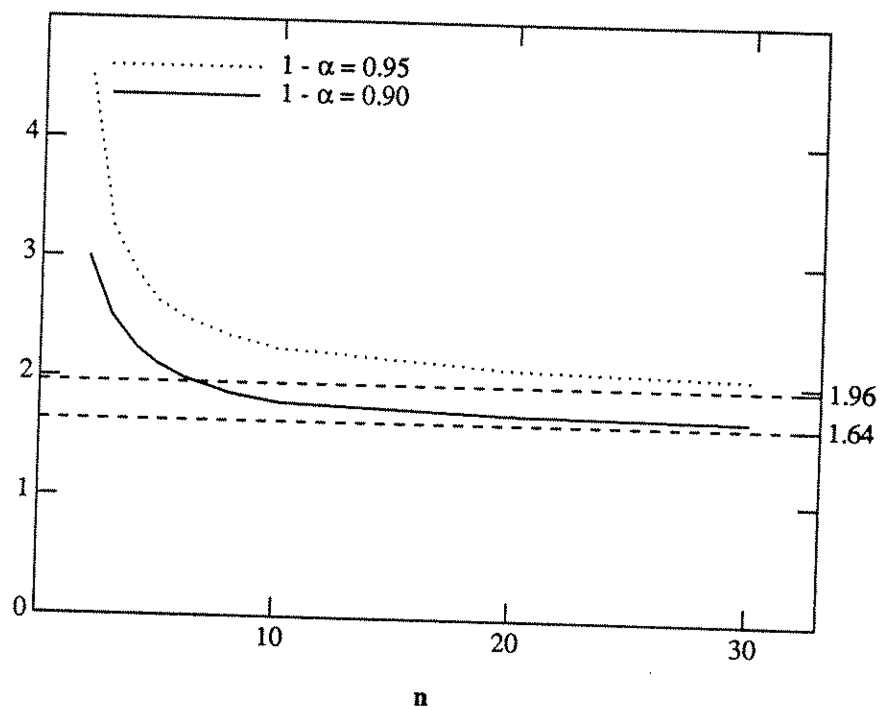


Figure 3.8 — Plots of $t_n(1-\alpha/2)$ for $1-\alpha=0.9$ and $1-\alpha=0.95$

of ten independent replications.

3.3.3. COMPARISON OF SIMULATION VS. ANALYTIC MODEL

Having established that the values produced by the simulator are within an acceptable confidence interval of the means of the random variables being measured, we must still demonstrate that these random variables are consistent with the network protocol. One means of doing this is to predict the expected values through the use of mathematical models and to compare these predictions with the results of the simulator. By comparing metrics for which models may be found, we generate confidence in the results of the simulator as a whole. In particular, we will use the models for token cycle time and station delay presented in [PEDE87], [PEDE88], [KLEI76], and [SCHW87].

These comparisons are not intended to be a formal verification of the simulator; rather, they shall demonstrate that the results of the simulator are reasonable estimates of HSRB performance.

The models below assume inter-arrival times are distributed exponentially and message lengths are constant.

Prior to presenting the analytic models, we must define the following terms:

C = medium capacity in bits per second

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

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

L = station latency in seconds at each station

P = propagation delay in seconds between consecutive stations, assuming stations are spaced equidistant around the ring

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

$E(D)$ = expected packet station delay

One standard measurement of network performance is token cycle time, the time for a free token to complete one circumnavigation of the ring. This is an especially valuable metric for comparison since many other metrics are dependent on token cycle time. Additionally, the terms necessary to calculate token cycle time are common to all token ring networks. Hence, it is very easily modeled. We define two additional terms necessary for the derivation of token cycle time: the load offered by each station, $\rho = \frac{\lambda}{\mu}$, and the ring latency, $\gamma = N(L+P)$, (also called the empty-ring walk time). The empty-ring walk time is the time for a token to cycle the ring in the absence of any load, and consists of the sum of the station latencies and propagation delays incurred in one token cycle. Peden derives the expression for token cycle time to be

$$E(TCT) = \frac{\gamma}{1-N\rho}, \quad N\rho < 1 \tag{3.12}$$

The other metric which we will use for comparison is mean station delay, the delay a packet may expect to incur from its enqueueing until the time its transmission is begun. This consists of time spent waiting in the queue and time waiting at the head of the queue to access the token. Again, we turn to Peden's work, this time from [PEDE88], for the expression

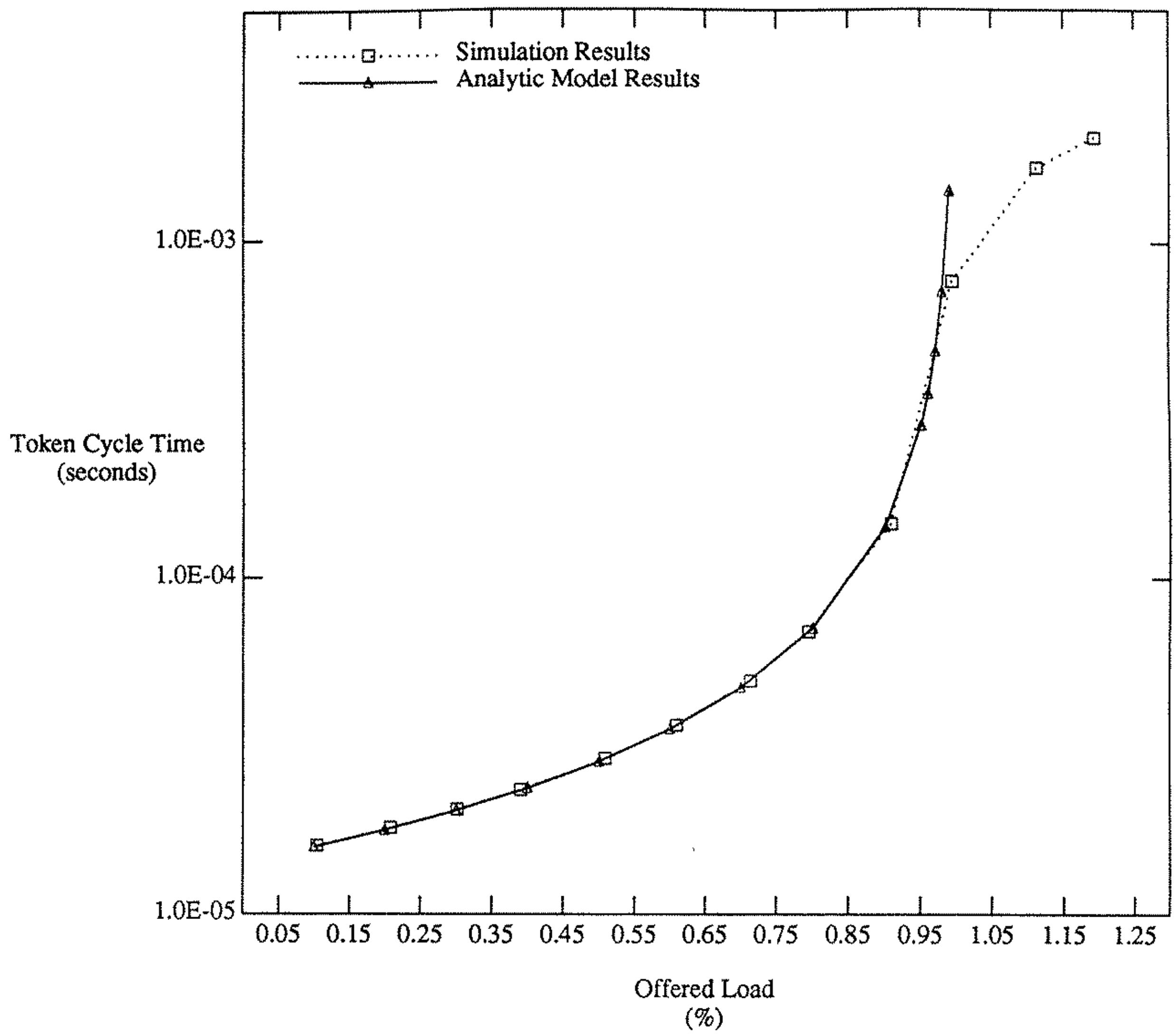
$$E(D) = \left[1 + \frac{\beta}{(1-\beta)^2}\right]\left[\frac{\frac{\lambda}{\mu^2}}{2(1-\rho)} + \frac{(N-1)\rho + \gamma\mu(1-\rho)}{2\mu(1-N\rho)}\right] + \frac{\beta}{(1-\beta)^2}\left[\frac{\gamma}{1-N\rho} + \frac{1}{\mu}\right]$$

$$\textit{where} \quad \beta = 1 - e^{-\lambda\left[\frac{\gamma}{1-N\rho} + \frac{\rho}{\mu}\right]} \tag{3.13}$$

Given these models, we may determine a network configuration for which they will apply. Care must be taken to ensure this applicability, since the models may otherwise yield false or misleading values. The configuration which was chosen for these comparisons is as follows:

100 stations evenly distributed on the ring
80 Mbps medium rate
1000 m ring circumference
6 bit internal station latency
exponential packet arrival distribution
2056 bit constant packet length
error free transmission
short message protocol not implemented

Given this configuration, we generate graphs of the analytic values and compare these to the simulation results. Figures 3.9 and 3.10 show how closely the predicted values match the observed values. This accuracy is due primarily to the validity of the models' underlying assumptions. The HSRB protocol is conceptually very simple. The same determinism which makes the protocol so suited to real-time control also makes it very easy to model. Excluding the short message protocol, the assumptions made in the analytic model are exactly those in the simulated protocol. Hence, the curves for predicted and observed performance match extremely well. This match generates confidence that the simulator indeed produces results which are correct for the HSRB protocol.

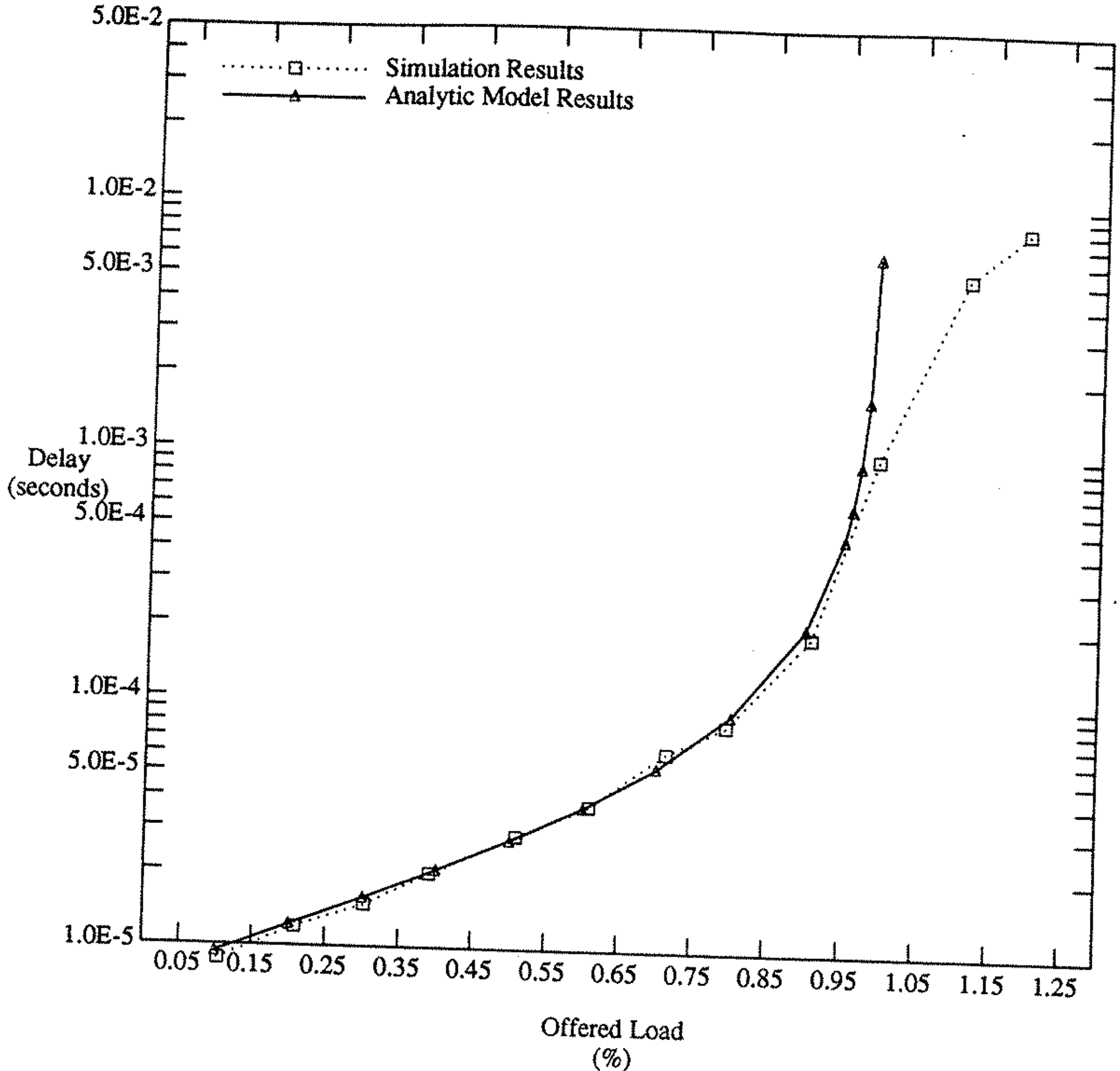


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 2056 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 3.9 — Token Cycle Time

Analytic Model vs. Simulation



Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 2056 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 3.10 — Station Delay

Analytic Model vs. Simulation

Chapter 4

SIMULATION RESULTS

4.1. INTRODUCTION

With a simulator such as the one developed for this project it is easy to generate more data than can reasonably be analyzed in a thesis of this scope. Thus, the results presented in this chapter are by no means a comprehensive analysis of the SAE HSRB. Rather, they summarize several of the important interactions which influence network performance. We will begin by examining the results of several base configurations. These will illustrate the effects that the priority mechanism and the short message protocol have on the performance of the network. We will then examine variations on these base cases, analyzing the effect on the network of such variables as mean message length, ring circumference, and number of stations. We will examine the interactions between these factors and the short message protocol, one of the unique features of the HSRB.

4.2. BASE CONFIGURATIONS

In order to become familiar with the basic performance of the network, we establish four base configurations. This serves two purposes: 1) it establishes a series of baselines with which to compare the simulations in later sections, and 2) it helps us gain insight into how such basic issues as single vs. multiple priority and short message protocol vs. no short message protocol affect network performance. The base cases consist of all four combinations of single/multiple priority and short message/no short message. In all other respects they are identical. The inter-arrival times of the messages and the message lengths are assumed to be distributed exponentially. We first examine the simplest of these cases, the single priority case with no short message protocol. We examine this case in some detail to illustrate the types of metrics which the system is capable of measuring, but our examination of the other cases will not be as detailed.

4.2.1. SINGLE PRIORITY, NO SHORT MESSAGE PROTOCOL

Within this configuration we will examine eight performance metrics. They include token cycle time, tokens received, tokens accessed, arrival queue length, queueing delay, network access delay, service delay, and throughput.

4.2.1.1. TOKEN CYCLE TIME

In Figure 4.1 we see a plot of mean token cycle time as a function of offered load. As the network load increases, more messages are transmitted in a given token cycle. Hence, token cycle time increases to accommodate this message traffic. As the network becomes very heavily loaded, a point is reached at which each station continually has messages to send, hence the token passes in round-robin fashion with each station transmitting in turn. As this point is approached, the token cycle time stabilizes. This effect is seen beyond 80% offered load, in this case.

4.2.1.2. TOKENS RECEIVED AND TOKENS ACCESSED

At low offered loads, many more tokens are received at a station than that station has messages to send. The token circles idly much of the time. As the load increases (and consequently the token cycle time increases), fewer tokens are received in a given time period. The number of tokens accessed, however, increases as a function of offered load. Eventually, the number of tokens received and the number of tokens accessed converge toward the same limiting value. This phenomenon is seen in Figure 4.2.

4.2.1.3. ARRIVAL QUEUE LENGTH

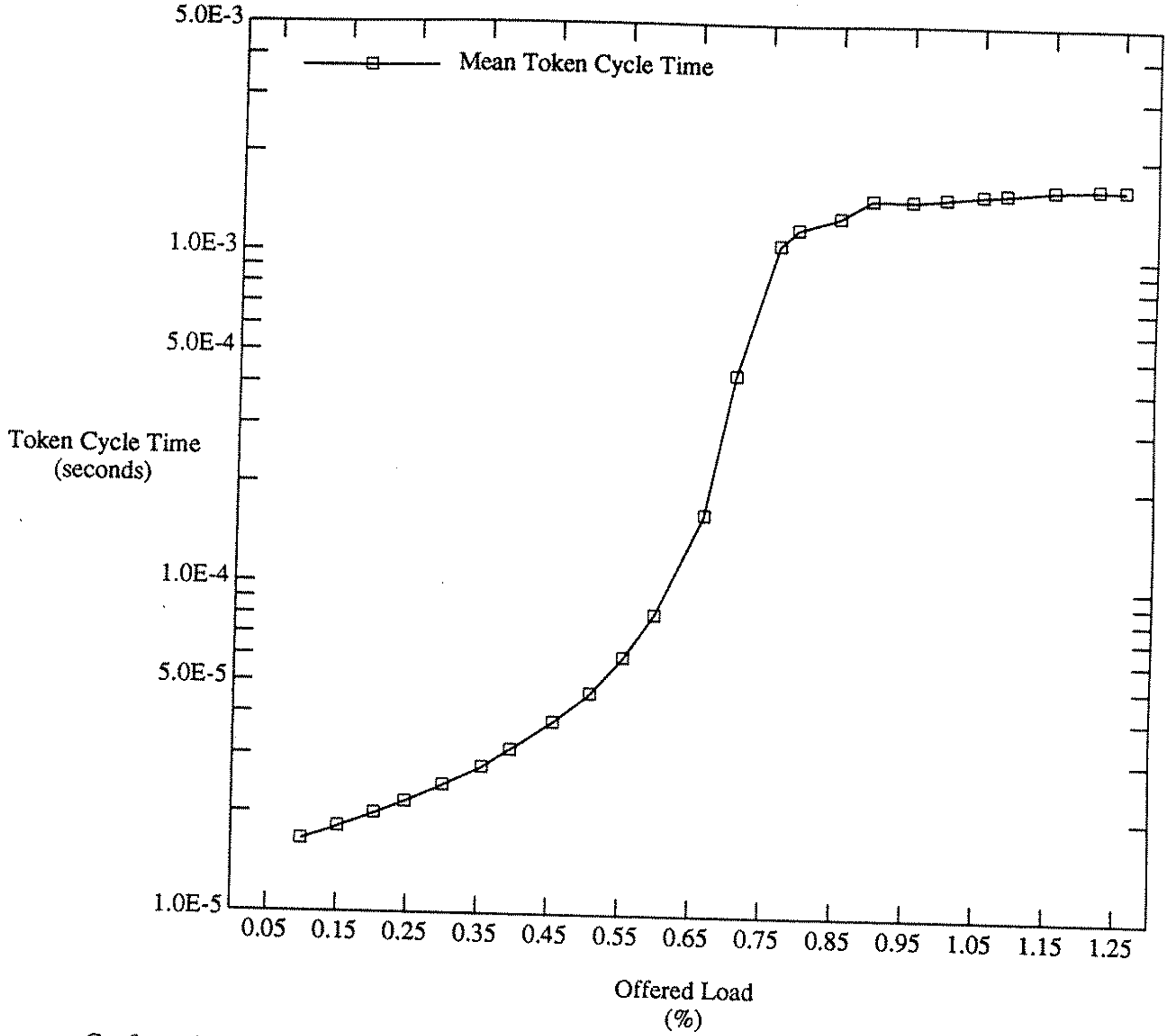
As the load to the network increases, so does the rate of message arrival. As this rate increases, it eventually exceeds the rate at which messages may be serviced. One measure of the relationship between arrival rates and service rates is mean arrival queue length at token claim, the number of messages in the queue upon the claim of a free token by that station. As long as messages are serviced at least as fast as they arrive, the mean arrival queue length will remain stable. When the arrival rate exceeds the service rate, the number of messages in the queue increases in a time dependent manner. In Figure 4.3 we see that the mean arrival queue length remains stable up to roughly 65% offered load. Beyond that point it increases as a function of offered load.

4.2.1.4. DELAY FACTORS

Three delay metrics are collected by the simulator: 1) queueing delay, the time spent in the queue prior to reaching the head of the queue, 2) network access delay, the time spent at the head of the queue waiting to claim the token, and 3) service delay, the time from the enqueueing of a message until the last bit is received at its destination. Service delay is, of course, a function of both queueing delay and network access delay, as well as transmission and propagation delays. In Figure 4.4 we see the interaction of these delays. Note that at low loads service delay is primarily a function of the time necessary to access the token. As load increases, token cycle time stabilizes as we have seen. Hence, network access delay stabilizes as well. However, queueing delay continues to increase, and rapidly becomes the dominant factor in determining service delay. It should again be noted that statistics are collected *only on those messages which are delivered* in the course of the simulation. Messages which remain in the queues at the end of the simulation are ignored. This explains why these delay curves do not have the exponential shape predicted by classical queueing theory. This decision is reasonable, however, since statistics collected on messages remaining in the queues would be meaningless.

4.2.1.5. THROUGHPUT

The last performance metric we will examine for this base configuration is normalized throughput, or efficiency. This is the fraction of the network capacity which is usable for actual data transfer rather than network or protocol overhead. We see this graph in Figure 4.5. We see from this data that the network behaves in a stable fashion when overloaded. Throughput achieves a maximum value and remains at that level. The value of this attained maximum is not meaningful unless we place it in context. Throughput, as with all the above metrics, is very dependent on the configuration simulated. Hence, the usefulness of such general graphs lies in comparison with other base cases. By such comparison we gather insight into the factors influencing performance. This comparison is the subject of the next section.

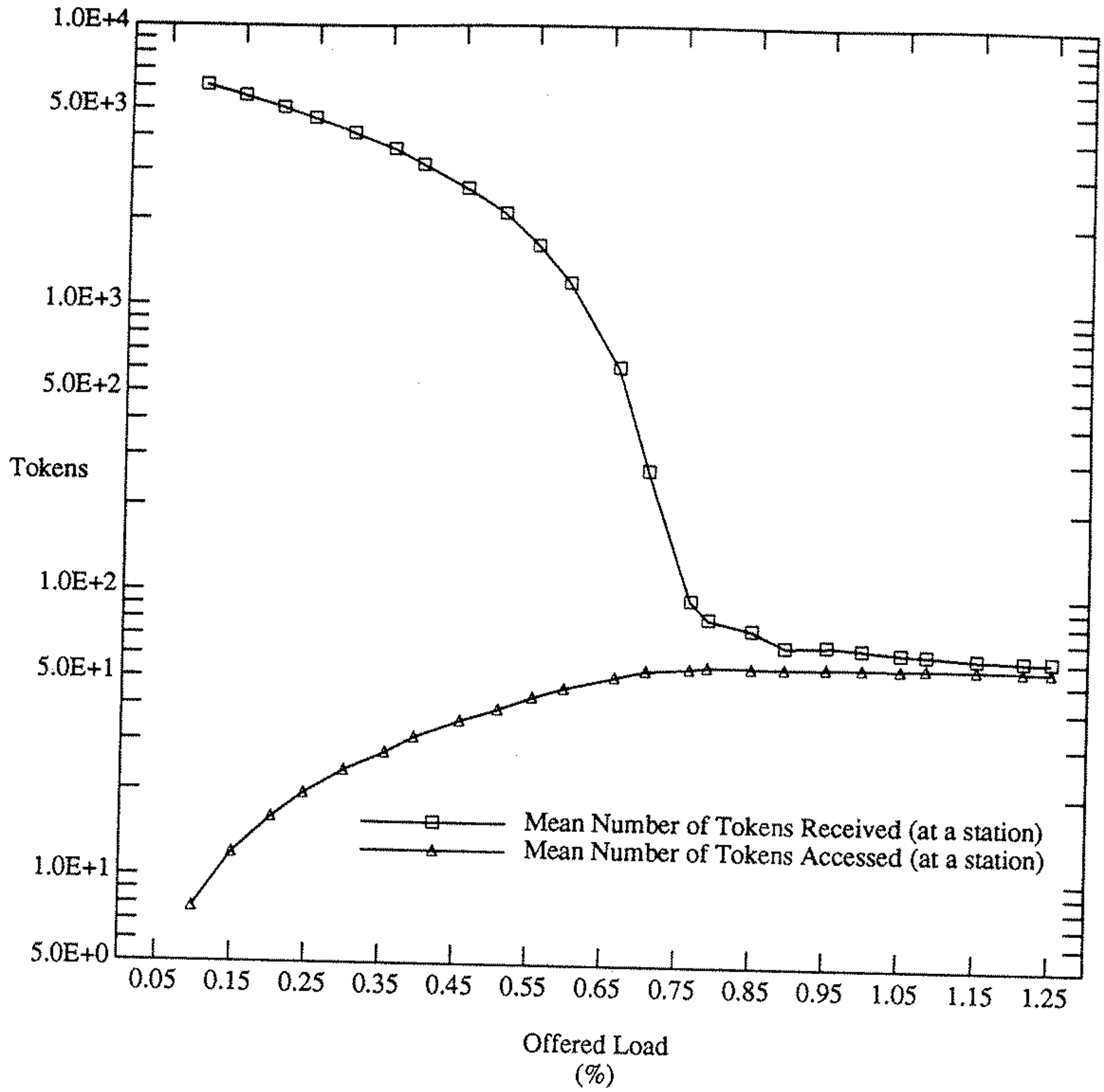


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.1 — Token Cycle Time

Single Priority, No SMP

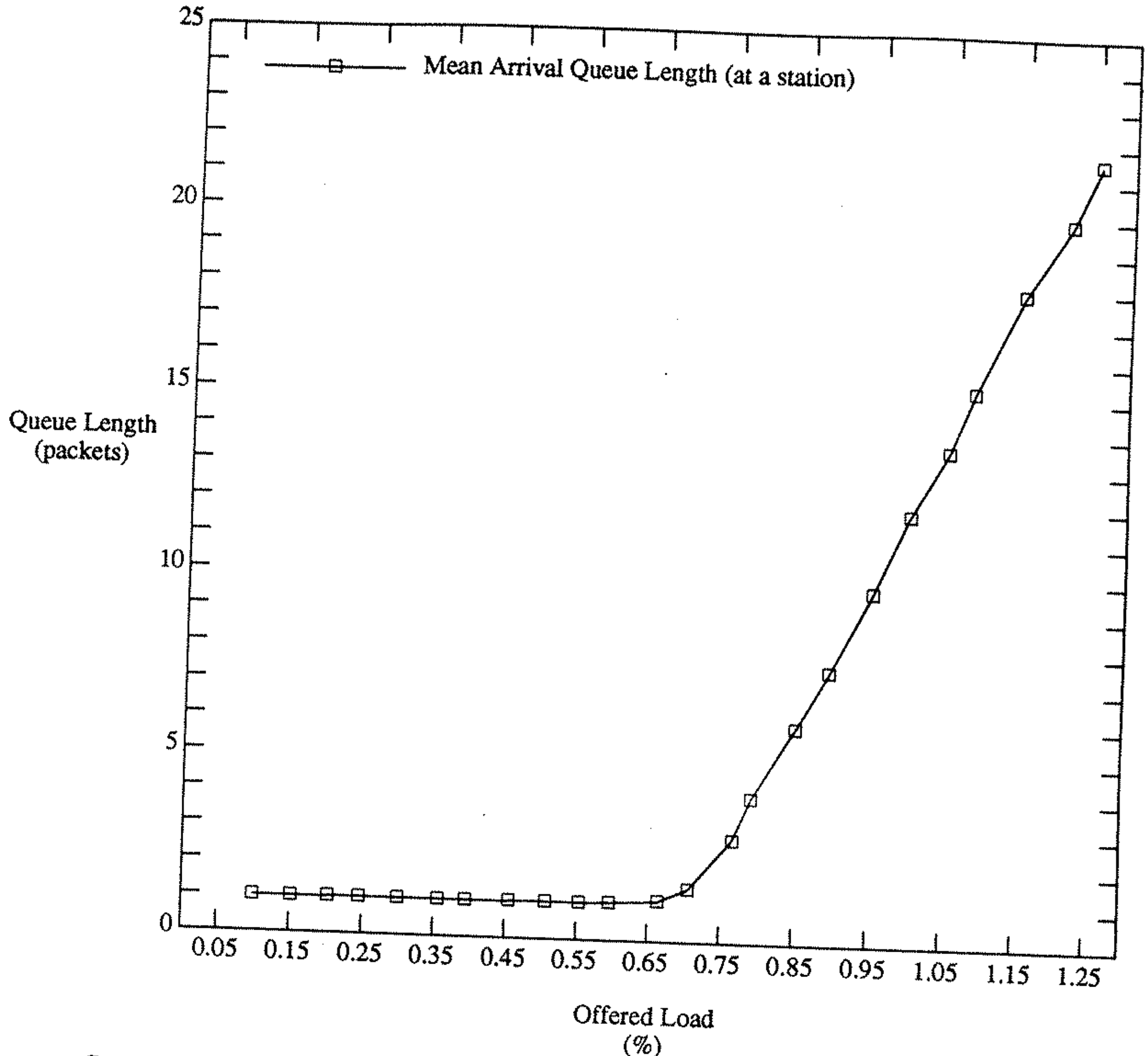


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.2 — Tokens Received and Tokens Accessed

Single Priority, No SMP

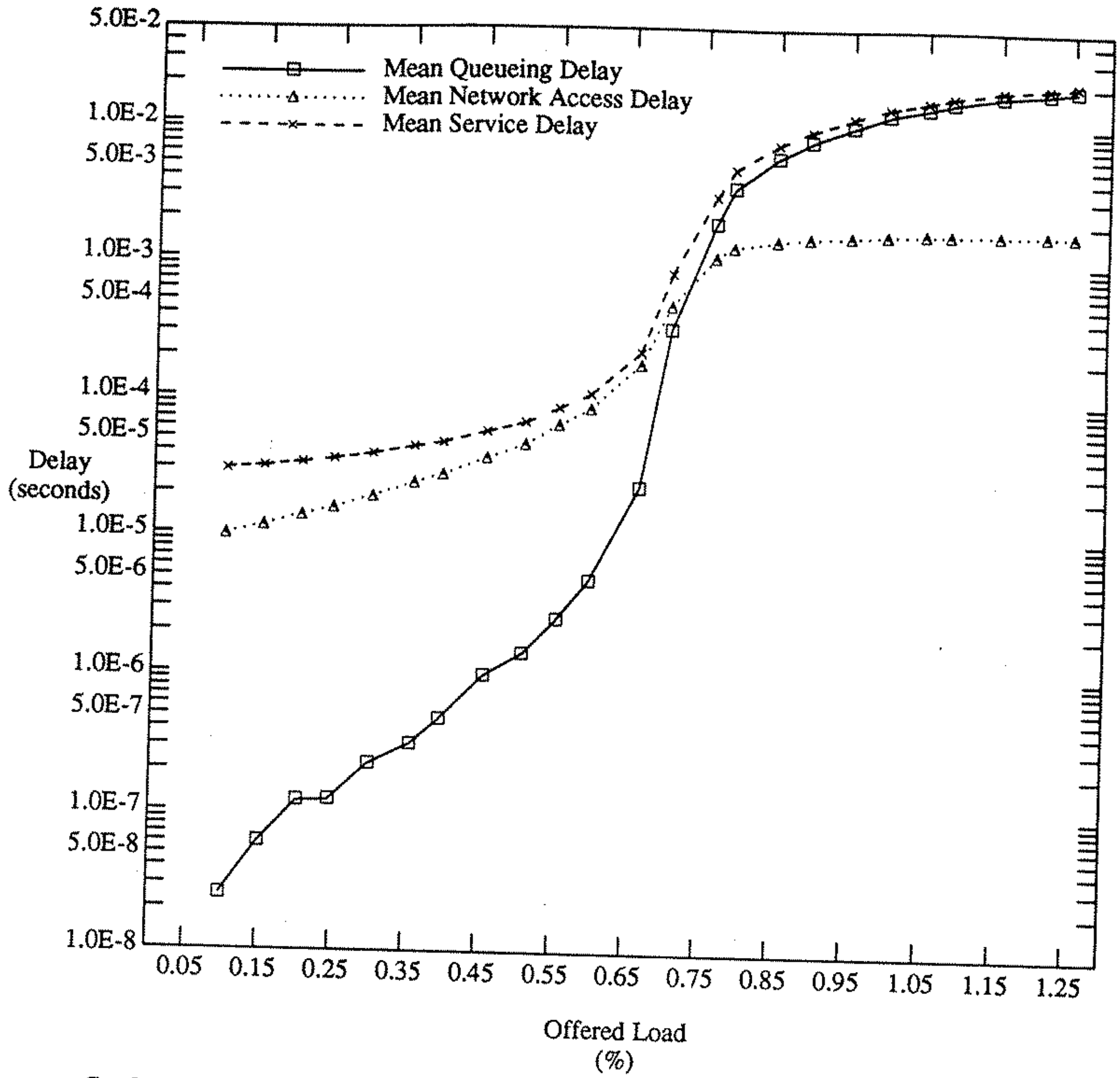


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
1 Priority
No Short Message Protocol

Figure 4.3 — Mean Arrival Queue Length At Token Claim

Single Priority, No SMP

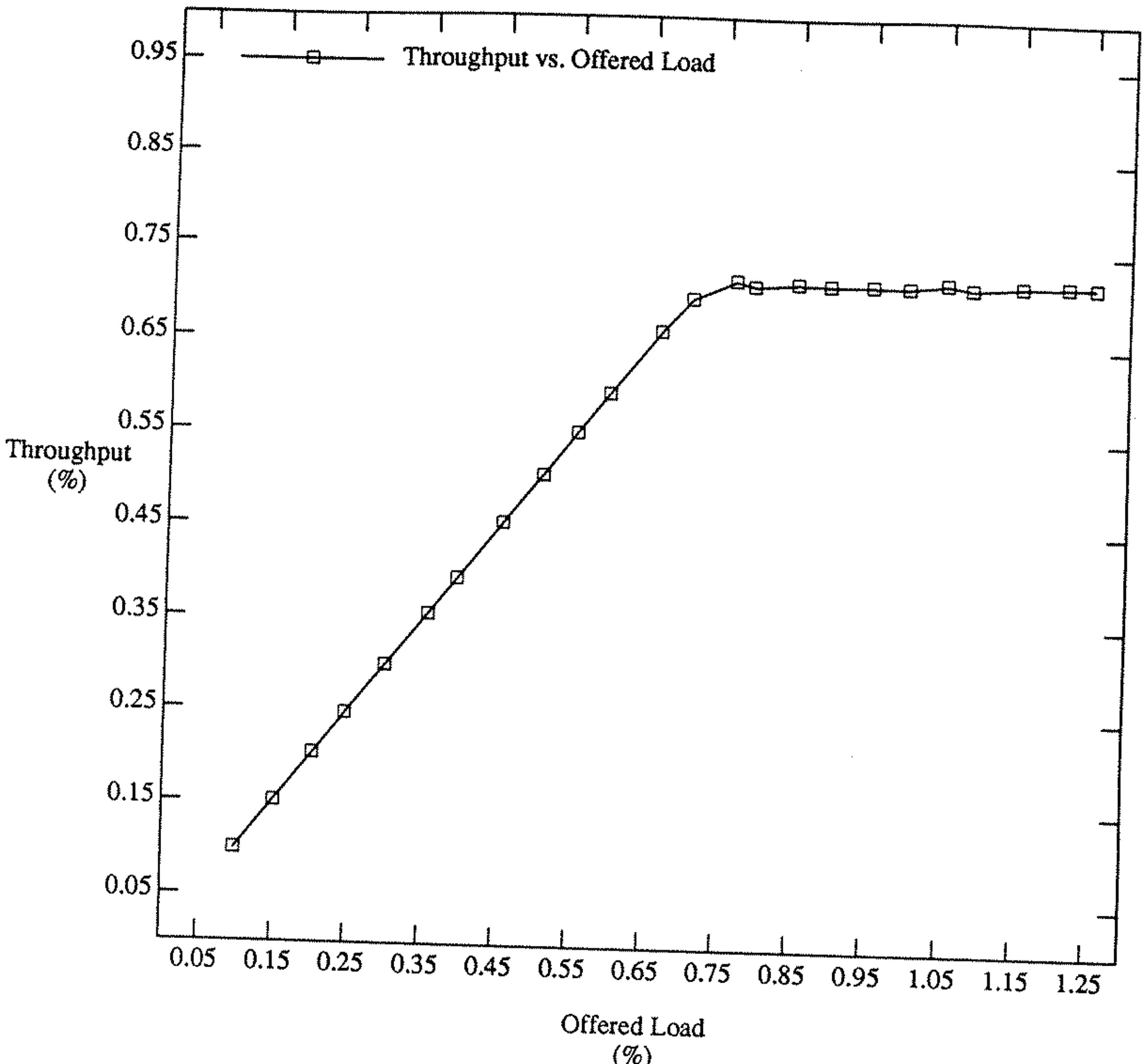


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.4 — Delay Factors

Single Priority, No SMP



Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.5 — Throughput

Single Priority, No SMP

4.2.2. COMPARISON OF BASE CASES

As we compare the results of the four base configurations, we shall focus on three performance metrics: token cycle time, service delay, and throughput. By examining these metrics we increase our understanding of how the protocol behaves, and we are able to predict how other metrics react to perturbations in the network configuration.

4.2.2.1. TOKEN CYCLE TIME

In Figure 4.6 we see a comparison of mean token cycle times for the four base configurations. There are several interesting points to note about this graph. First, the single priority and multiple priority cases perform identically at low offered loads. When few messages are present on the network, token cycle time is unaffected by the number of priorities. Secondly, using the priority scheme results in short token cycle times. Since lower priority messages are delayed if higher priority messages are waiting, a token may pass unaccessed through many stations before it is claimed. Thus, the token rotates more quickly than in the single priority case. Thirdly, the short message protocol affects the single priority case differently. In the single priority case, the short message protocol does not alter the order of message transmission. It serves only to decrease the wasted time between the completion of message transmission and the issuance of a new free token. Thus, the token cycle time decreases as a given number of messages is transmitted in a shorter period of time. On the other hand, in the multiple priority case the short message protocol does affect the order of message transmission. Now, lower priority messages may be transmitted before higher priority messages which may be waiting further around the ring. With a larger number of messages being sent in each token cycle, the token cycle time increases rather than decreases.

4.2.2.2. MEAN SERVICE DELAY

In Figures 4.7, 4.8, and 4.9 we see the comparisons of mean service delay for the various base configurations. Figure 4.7 shows the single priority configurations, both with and without the short message protocol. As one would expect, since the order of message transmission is unchanged and the token cycle time is decreased by implementing the short message protocol, the mean service delay decreases as well. In Figure 4.8 we see the mean service delays of all eight priority queues for the multiple-priority, no-short-message-protocol case. We see the effects of the priority mechanism by the fan-out of the curves.

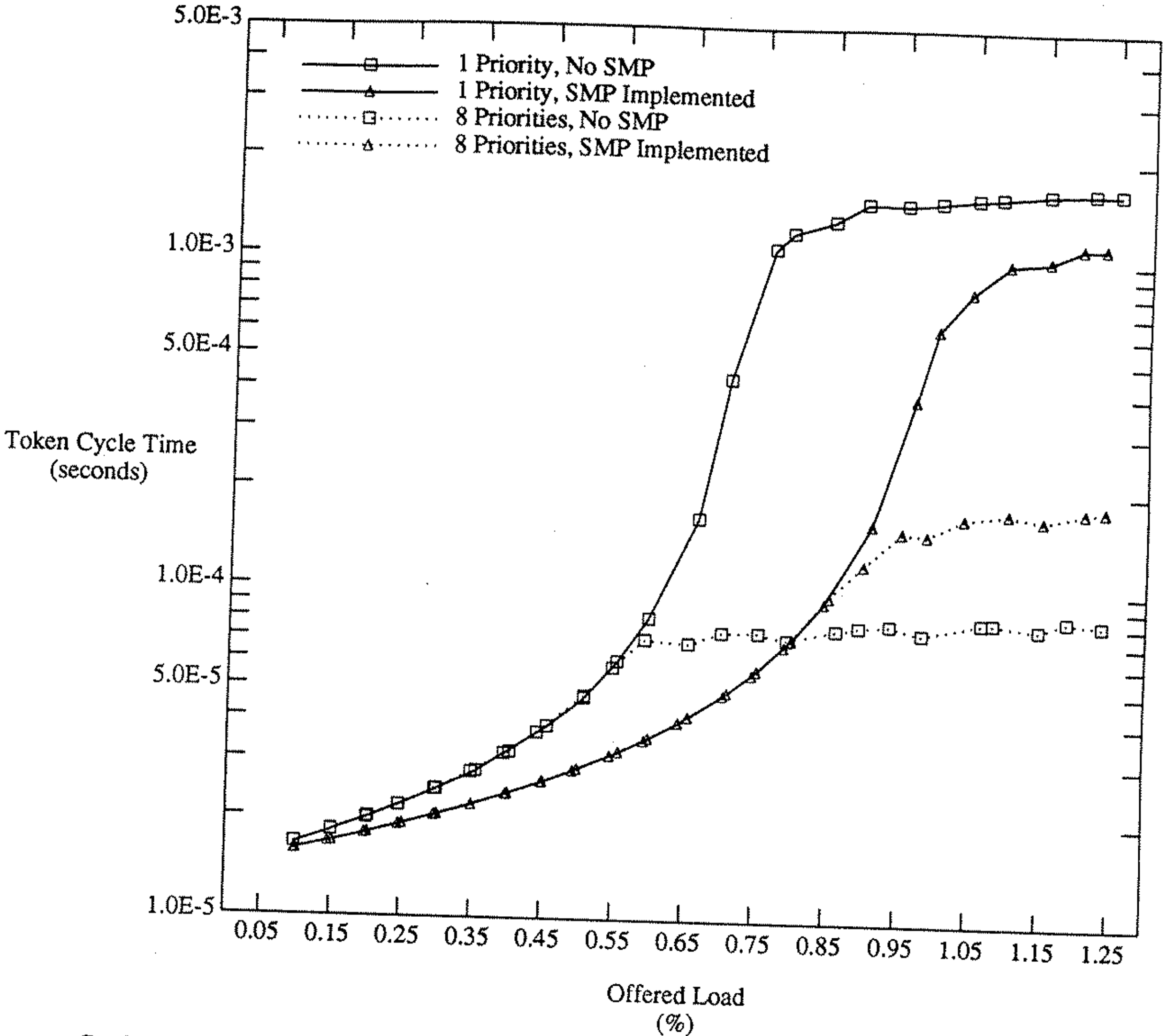
The lower priorities receive poorer service than the higher priorities until, eventually, service to the lower priority queues is shut off entirely. High priority messages continue to receive excellent service throughout the range of offered loads simulated. When the short message protocol is implemented, it has a significant effect on the priority mechanism. As is expected, the differentiation between the priorities becomes much less (see Figure 4.9). For low offered loads (before the queues begin to fill) all priority levels receive essentially the same level of service. As offered load increases, so does the priority differentiation, although this happens only at much higher loads than in the case without the short message protocol.

Note, furthermore, that the distribution of service delays for the multiple priority case varies around the service delay for the single priority case. The higher priorities have lower service delays than the single priority mean, while the lower priorities wait, on the average, longer than they would if all messages were transmitted at the same priority. This is the type of behavior which is the intent of the priority scheme.

4.2.2.3. THROUGHPUT

The effects of the various base configurations on throughput are illustrated in Figure 4.10. The multiple priority case requires more overhead for token traffic, hence the efficiency of the network with these configurations is less than that for single priority configurations.

Additionally, the short message protocol boosts throughput significantly. This happens in several ways. In the single priority case the short message protocol reduces the time wasted waiting at a station for the reservation value to return. In the multiple priority case, it also reduces the percentage of a token cycle which is wasted due to the priority mechanism. Thus, when the short message protocol is implemented, we see a greater throughput gain in the multiple priority case than in the single priority case.

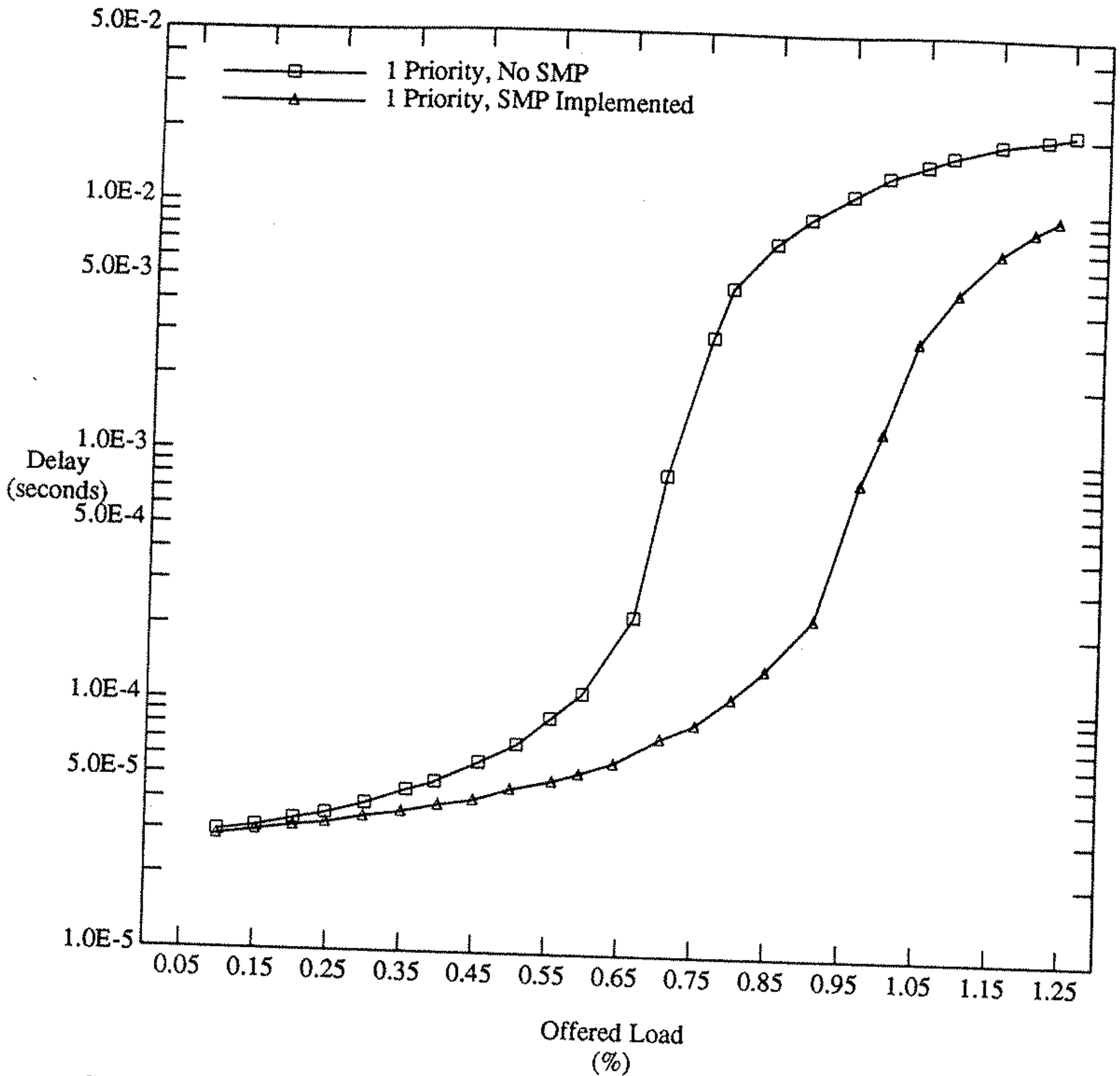


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially

Figure 4.6 — Token Cycle Time

Comparison of Base Cases

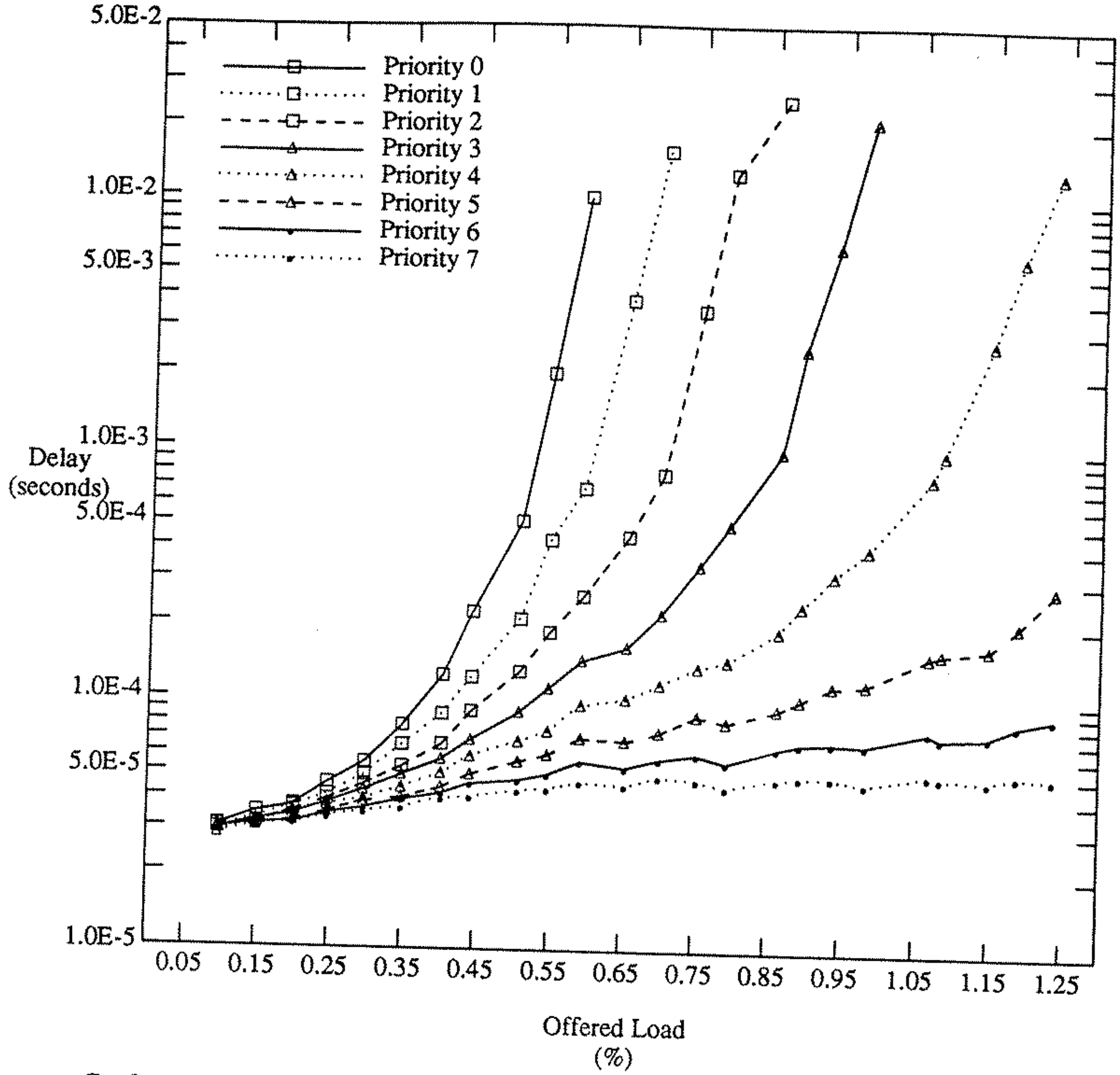


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially

Figure 4.7 — Mean Service Delay

Comparison of Base Cases

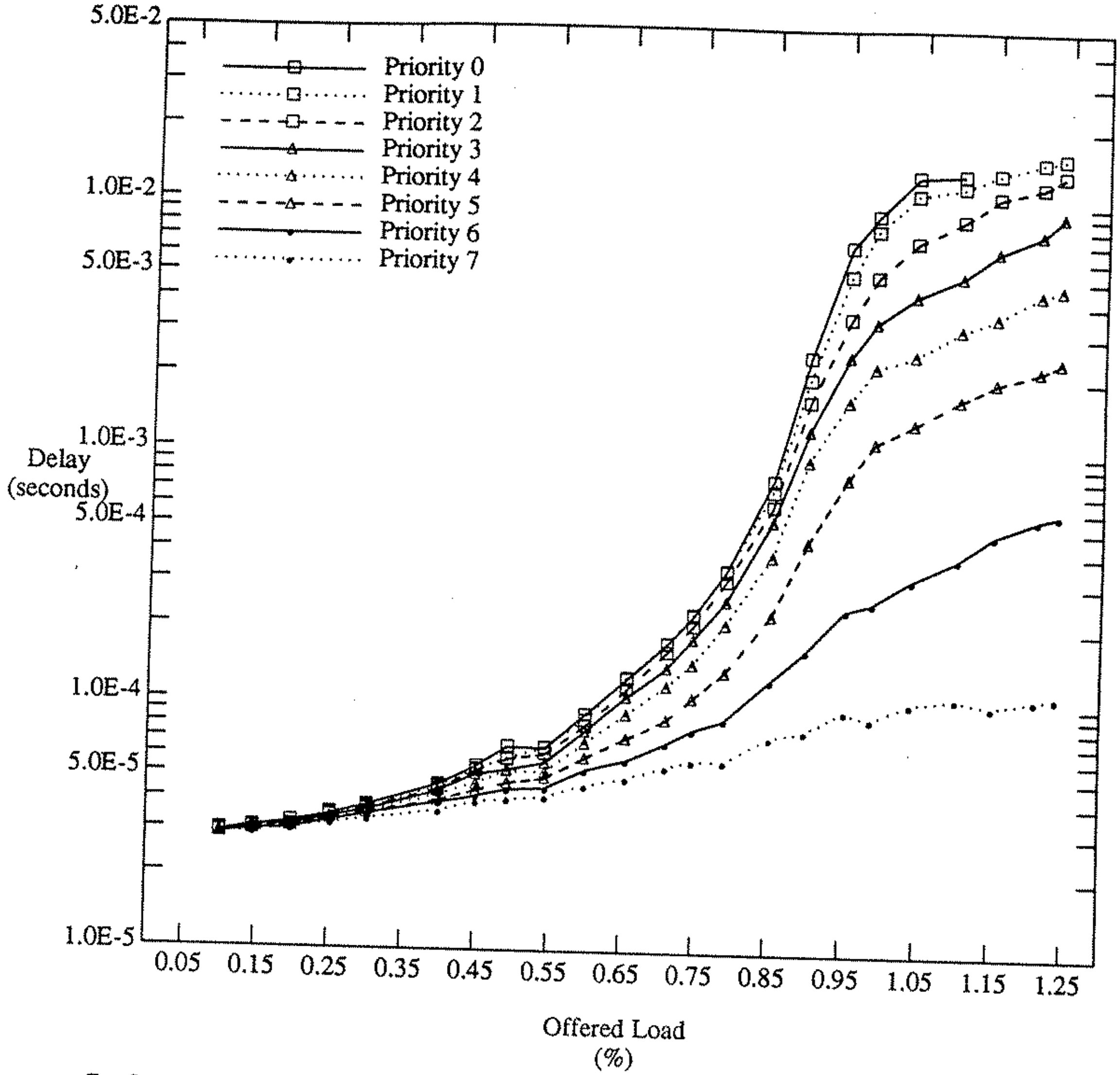


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
8 Priorities
No Short Message Protocol

Figure 4.8 — Mean Service Delay

Multiple Priority, No SMP

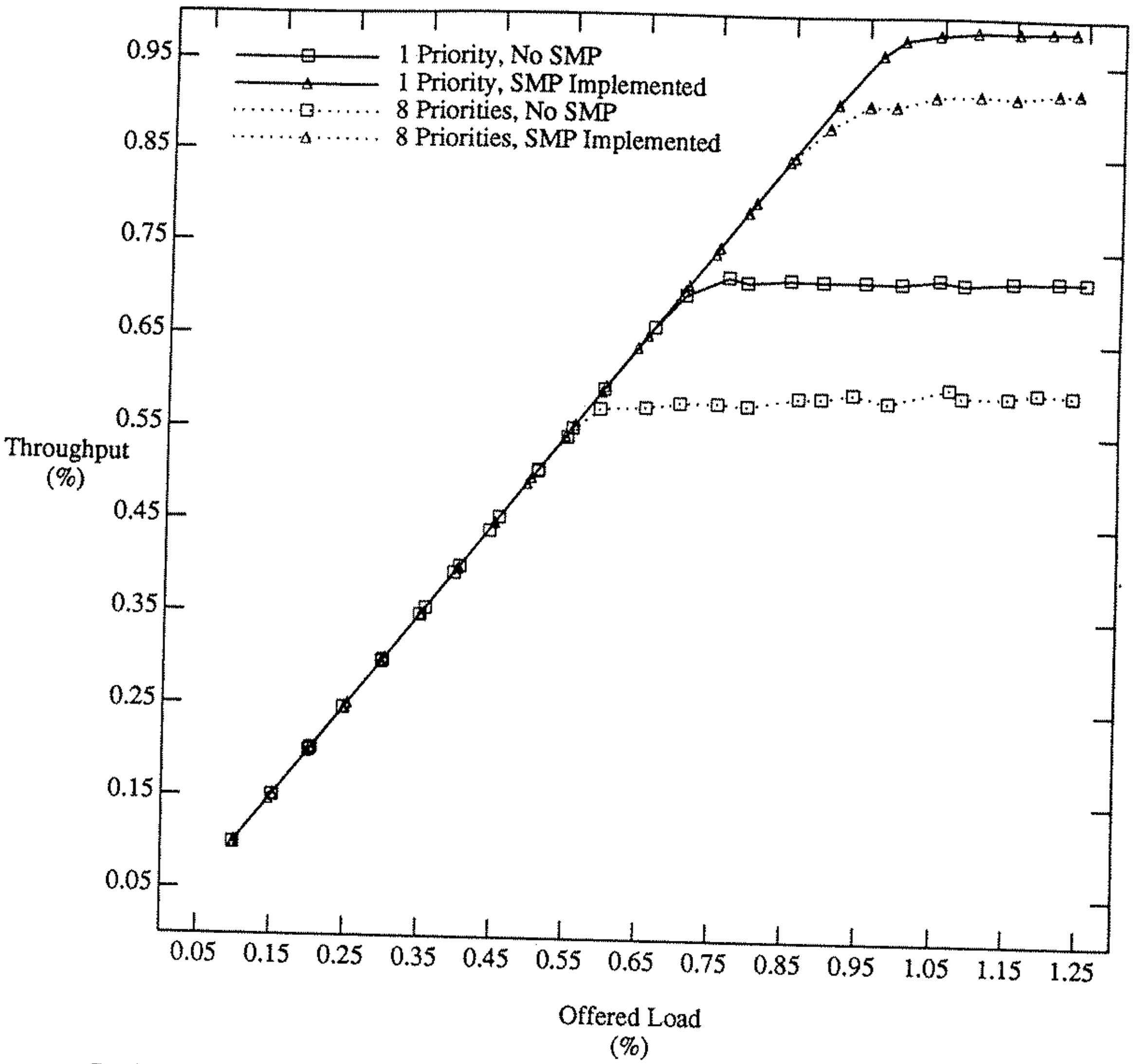


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
8 Priorities
Short Message Protocol Implemented

Figure 4.9 — Mean Service Delay

Multiple Priority, SMP Implemented



Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially

Figure 4.10 — Throughput

Comparison of Base Cases

4.3. VARIATIONS ON BASE CONFIGURATIONS

Having studied the results of the base configurations, we have a general idea of how the protocol behaves in certain situations. We now add another degree of complexity to the analysis by varying the parameters of the base configurations. Specifically, we shall examine the single priority base cases as altered by varying mean message length, ring length, and number of stations. In order to isolate the effects of such changes, we alter a single variable at any given time and observe the results.

4.3.1. VARYING MEAN MESSAGE LENGTH

In Figures 4.11 — 4.16 we see the results of the suite of simulations with varying mean message lengths. The messages range from 32 bytes to 512 bytes in length, yielding data points on both sides of our base case of 128 byte messages.

4.3.1.1. TOKEN CYCLE TIME

In order to understand the following graphs, one point must be remembered. The offered load to the network affects the number of *bits* delivered for transmission, not the number of messages. Consequently, at a given offered load many more short messages than long messages must be enqueued to achieve that load. We see the effect this factor has on token cycle time in Figure 4.11. The message queues fill much faster for shorter messages than for longer ones. Hence, the token cycle time increases much more rapidly for short messages. However, as the network approaches the saturation point (the point at which each station always has a message to send when the token arrives), token cycle time stabilizes. Curiously, the curves for 32 and 64 octet messages appear to stabilize at the same value. Initially, one would anticipate the curve for 32 octet messages to stabilize at a lower value than that of 64 octet messages. In a given token cycle the maximum number of messages which may be sent is equal to the number of stations; therefore, one would expect faster token cycles for short messages if both configurations had reached saturation. Remember, however, that a station must wait a minimum of the ring latency from the time it accesses a free token until the time it emits a new token (i.e. if the short message protocol is not implemented, the station must wait for the return of the reservation field). If a particular configuration generates primarily short messages (as is true in both these cases), then token cycle time will stabilize at this same value. We see that as the mean message length exceeds the ring latency, token cycle time does, indeed, stabilize at progressively higher values.

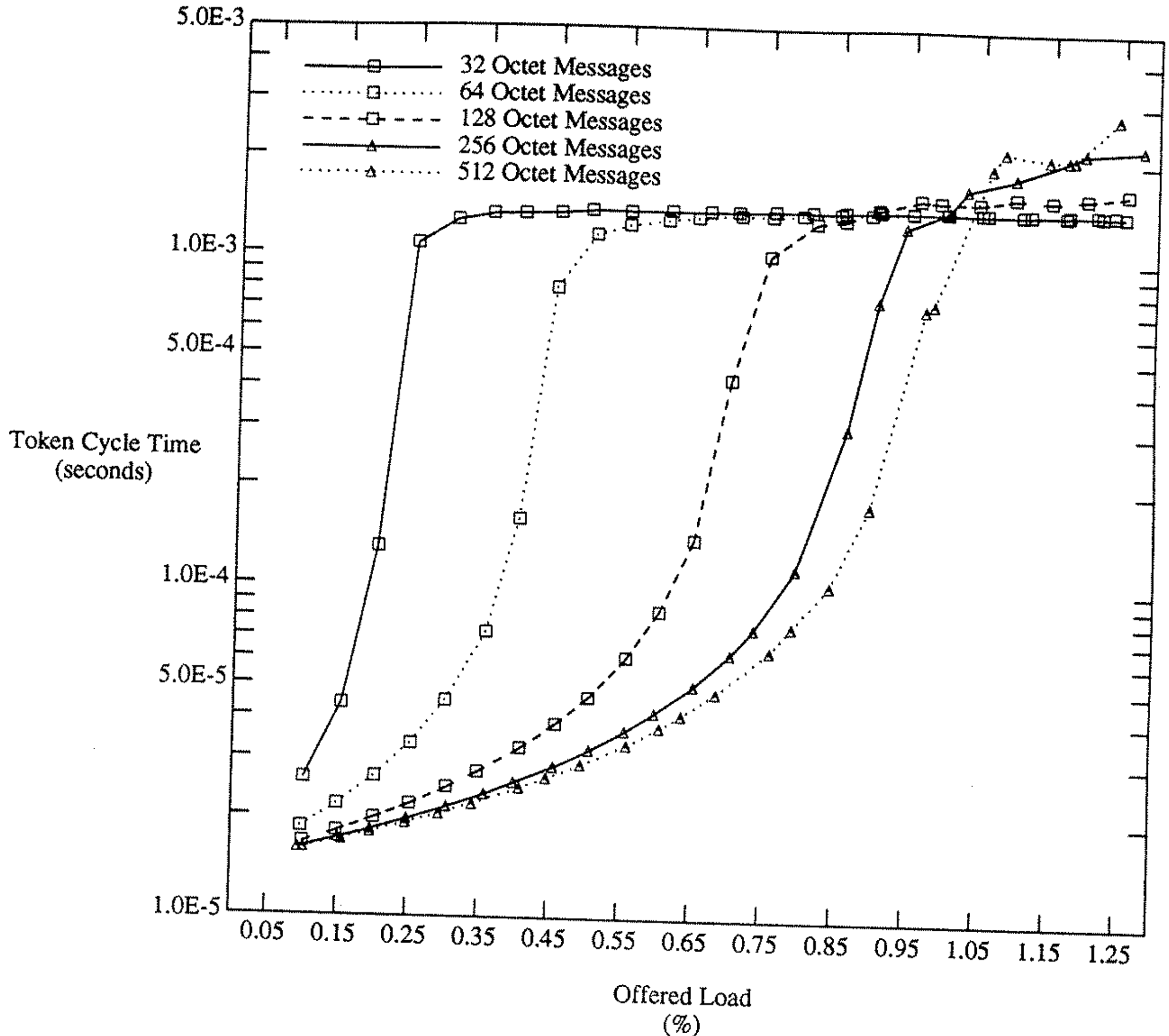
On the other hand, when the short message protocol is implemented (Figure 4.12), we observe the expected behavior. Token cycle time increases more rapidly for short messages, but it stabilizes at a lower level than for longer messages. Even for longer messages, though, the token cycle times are less for the case in which the short message protocol is implemented than for the case in which it is not. This is what we expect from the results of Section 4.2.2.1.

4.3.1.2. MEAN SERVICE DELAY

We look now to the effects of varying message length on mean service delay. We notice in Figure 4.13 that at very low offered loads (before the queues have begun to back up), short messages have less overall delay than long messages. Since the queues are not backing up, queueing delay is negligible, and the delay incurred is primarily a function of the network access delay and the transmission time. At such low loads the network access delay is roughly equivalent for all length messages, hence the transmission time (which is dependent on message length) is the primary delay factor. However, as loads increase, queueing delay rapidly eclipses any other delay factor, and shorter messages suffer greater delays than longer messages. Although this effect is observable in Figure 4.13, it is more readily apparent in Figure 4.14. The short message protocol extends the range throughout which queueing delay is not a factor. This accounts for the interesting crossover effect of the service delay curves.

4.3.1.3. THROUGHPUT

In Figures 4.15 and 4.16 we see the effectiveness with which the short message protocol improves the efficiency of configurations with primarily short messages. Figure 4.15 shows how incredibly inefficient short messages may be. With the implementation of the short message protocol (Figure 4.16), the efficiency of the 32 byte messages is boosted by well over 300%. The gains realized by longer messages are less, but they are, nonetheless, significant.

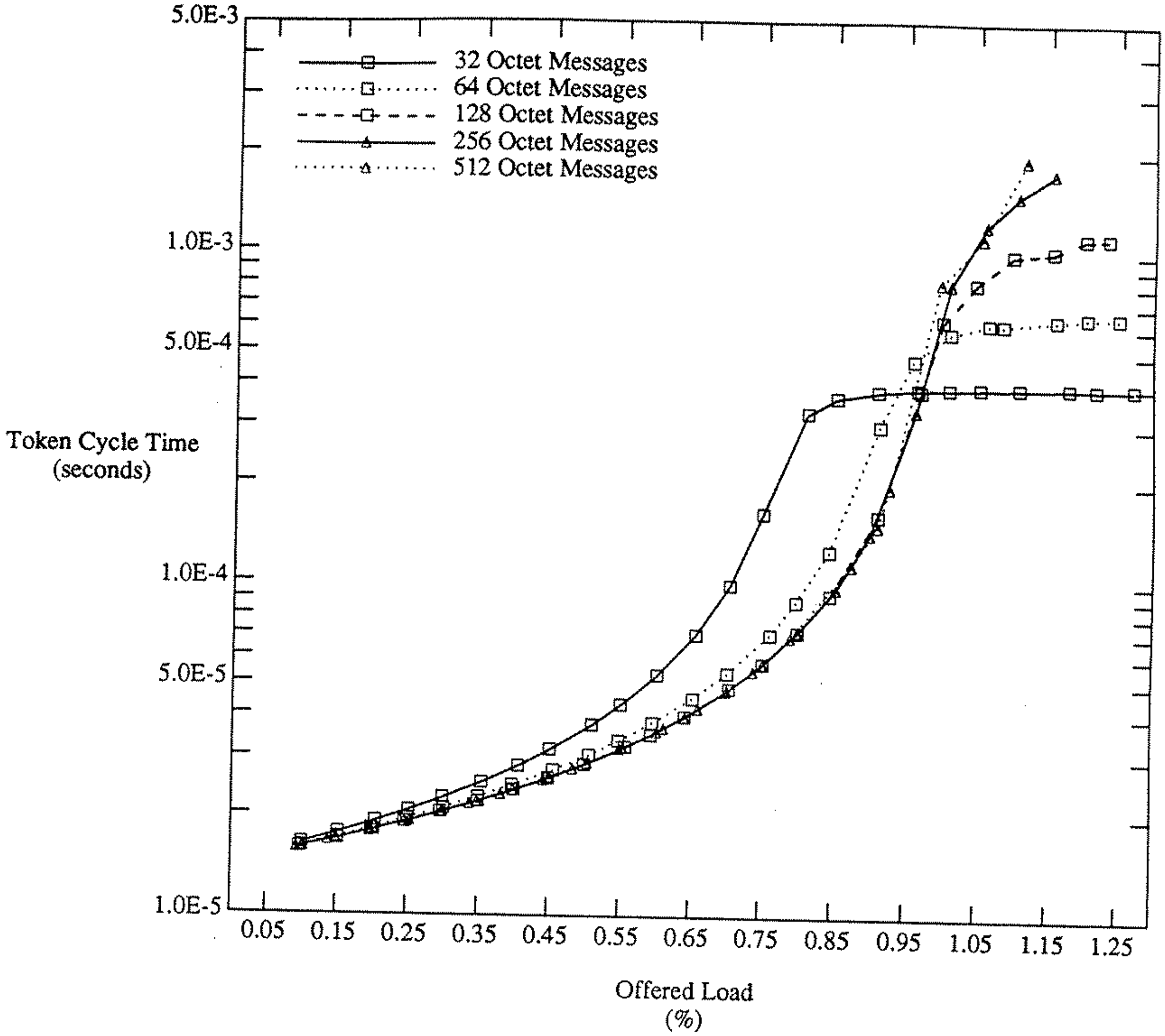


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1 Priority
No Short Message Protocol

Figure 4.11 — Token Cycle Time, Without SMP

Varying Mean Message Length

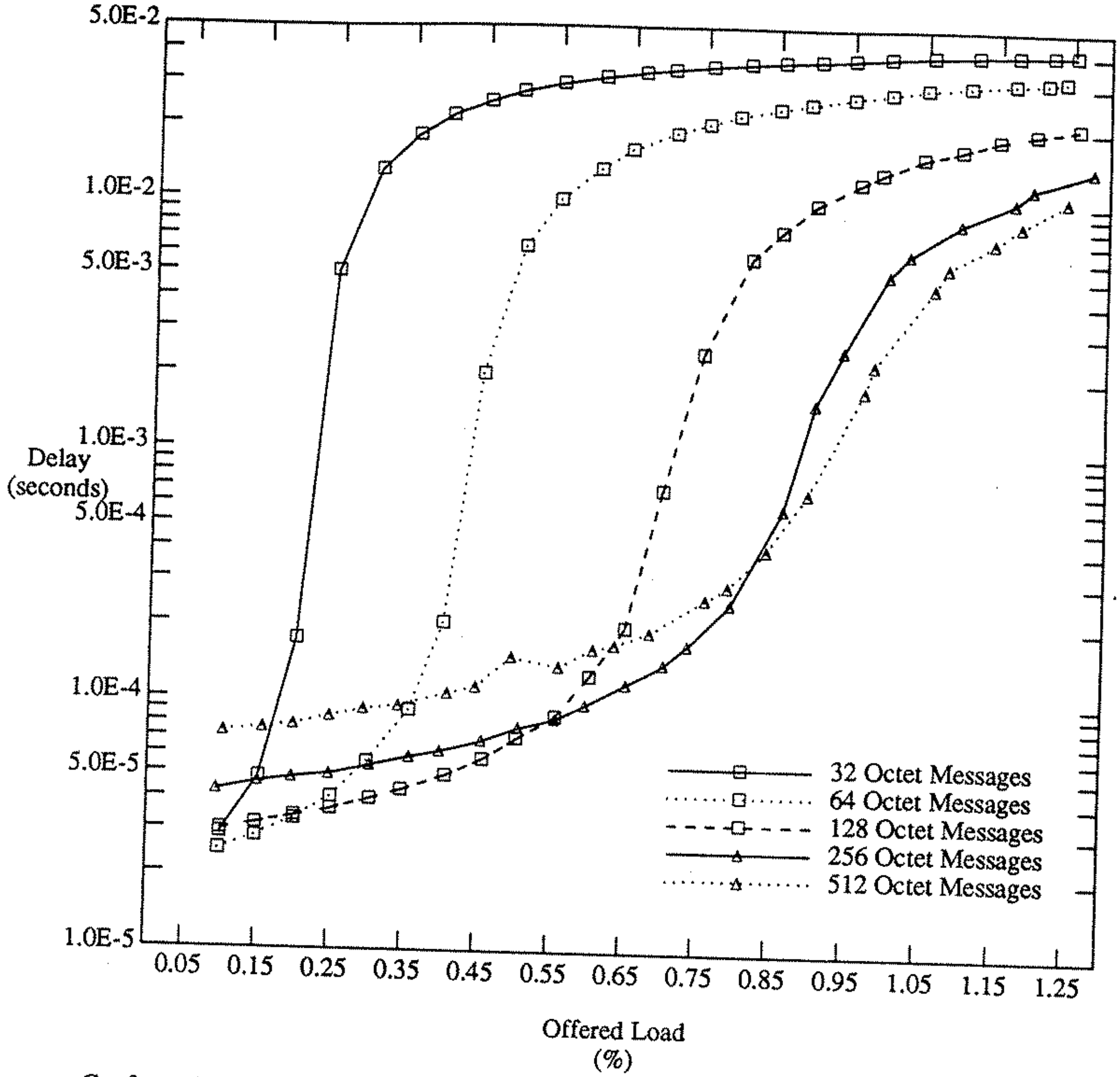


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1 Priority
- Short Message Protocol Implemented

Figure 4.12 — Token Cycle Time, With SMP

Varying Mean Message Length

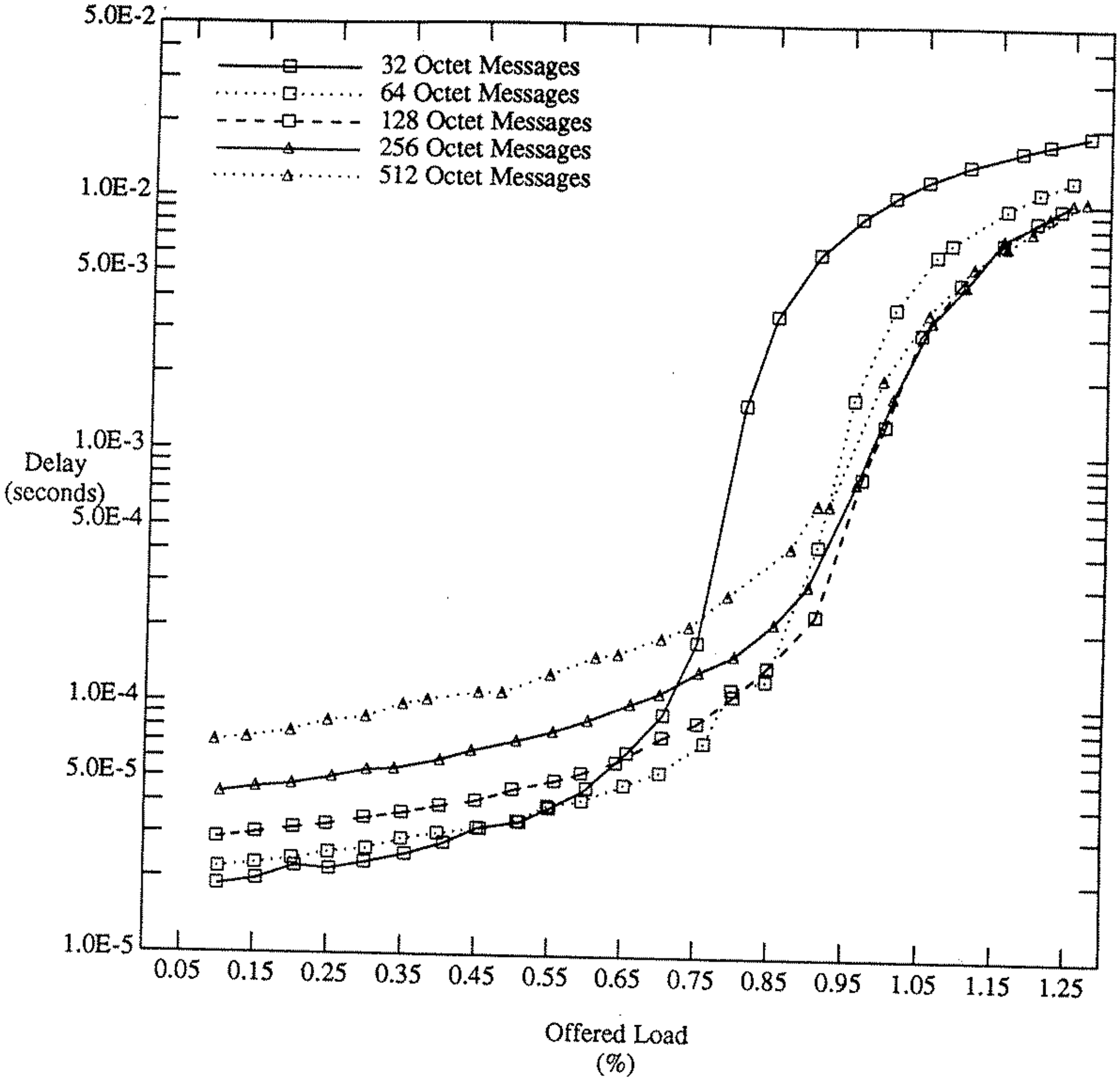


Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1 Priority
No Short Message Protocol

Figure 4.13 — Mean Service Delay, Without SMP

Varying Mean Message Length

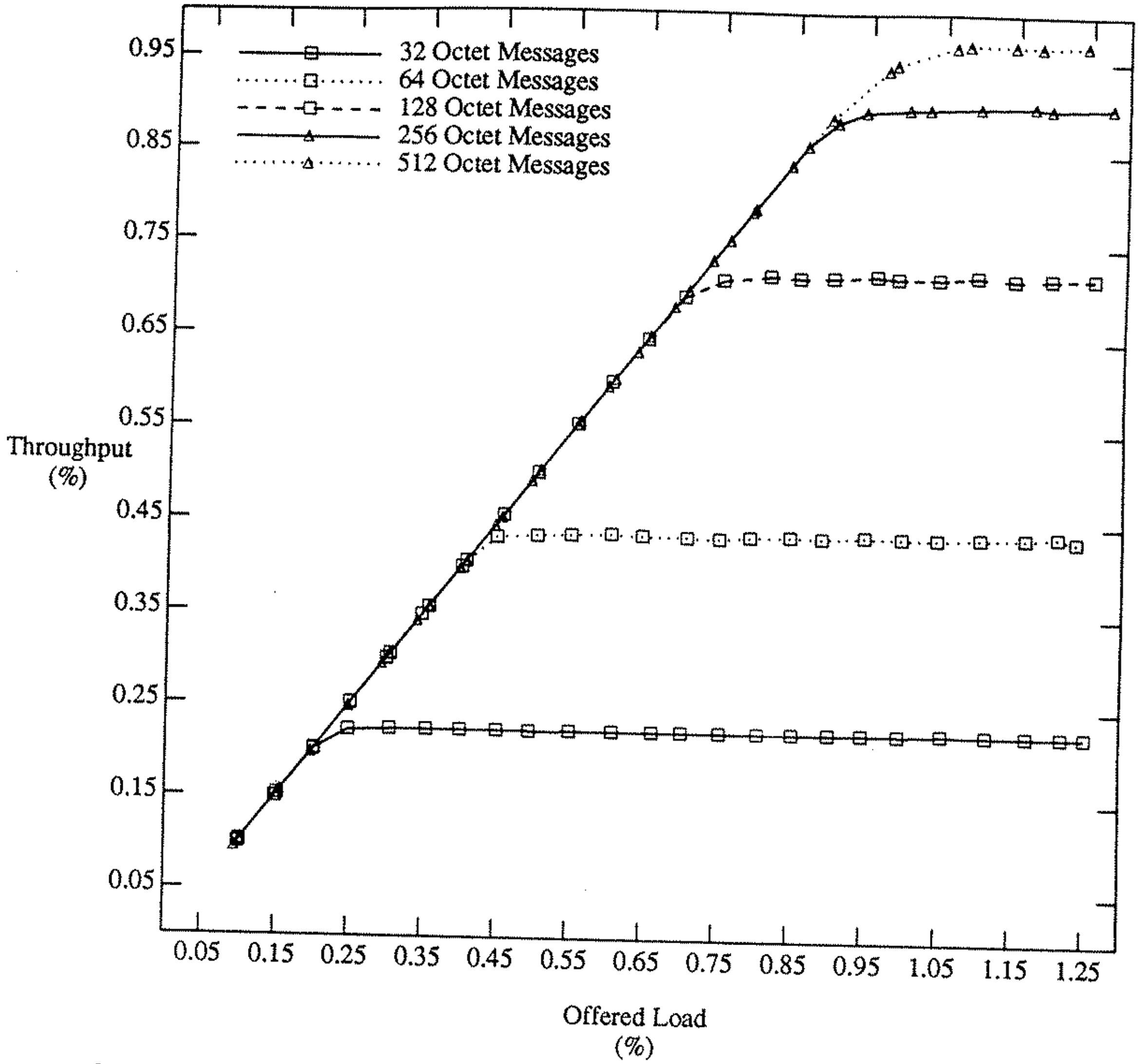


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1 Priority
- Short Message Protocol Implemented

Figure 4.14 — Mean Service Delay, With SMP

Varying Mean Message Length

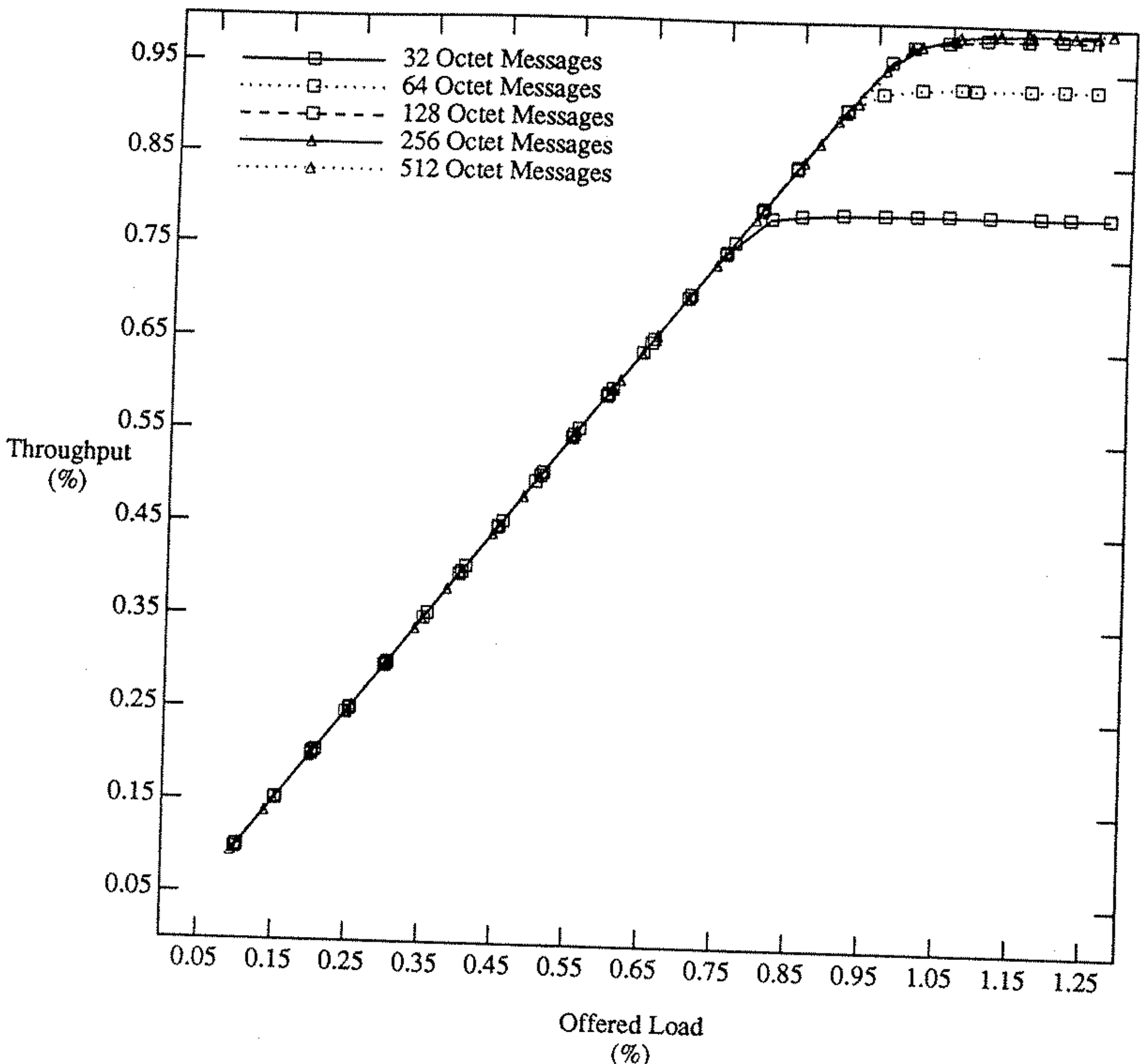


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1 Priority
- No Short Message Protocol

Figure 4.15 — Throughput, Without SMP

Varying Mean Message Length



Configuration:

100 Stations
80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1 Priority
Short Message Protocol Implemented

Figure 4.16 — Throughput, With SMP

Varying Mean Message Length

4.3.2. VARYING RING LENGTH

In Figures 4.17 — 4.22 we see the results of the simulations run for varying values of ring length. The values range from a very short, 10-meter ring to the 3000-meter ring which is the longest allowed by the HSRB protocol.

4.3.2.1. TOKEN CYCLE TIME

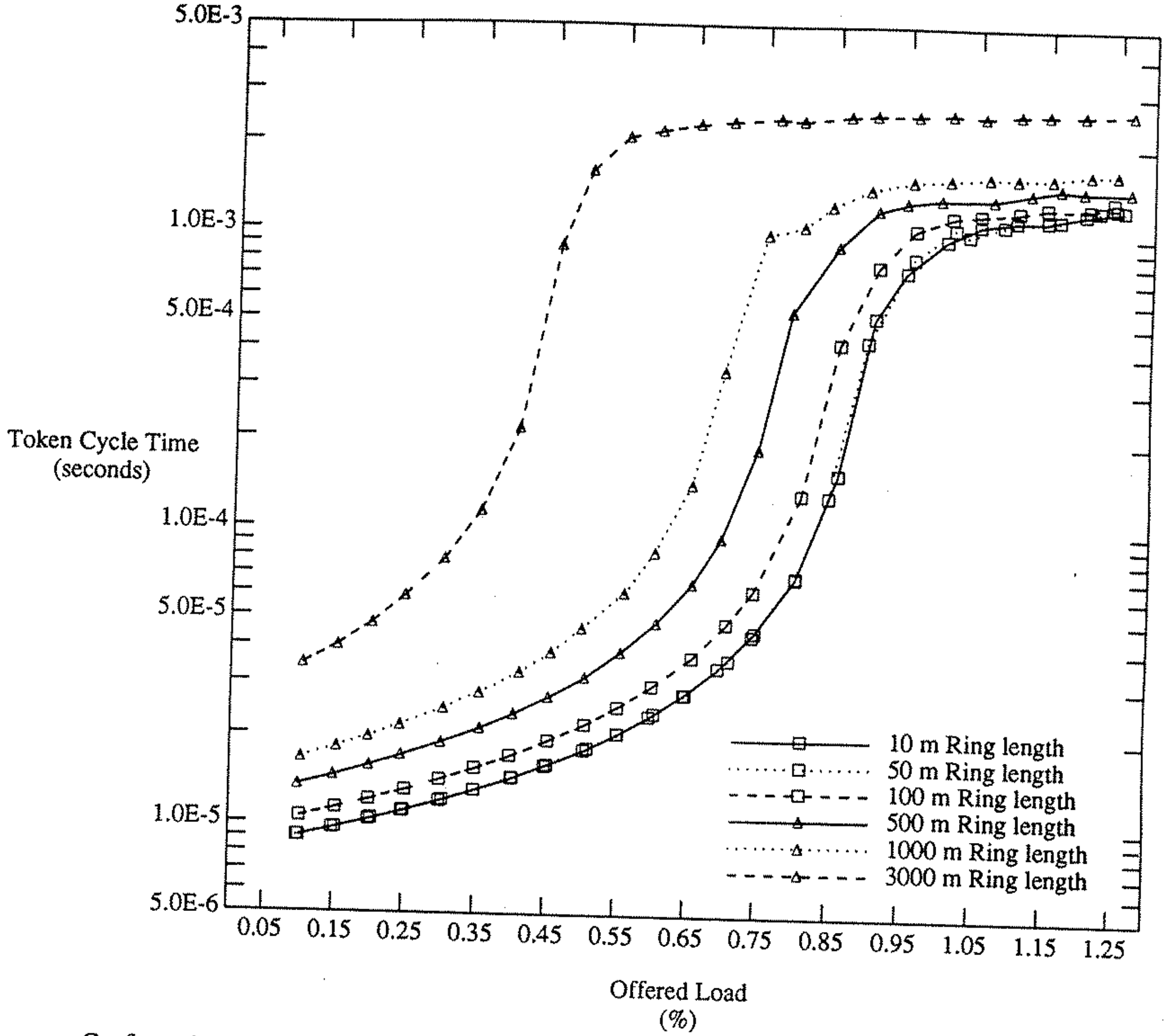
By increasing the length of the ring, one increases the propagation delay and, consequently, the total ring latency. In the case when the short message protocol is not implemented, this added latency is more significant, since each station waits for its message header to return before issuing the token. Since the extra propagation delay is incurred for each message transmission, we see differentiation in the token cycle time curves based on the length of the ring (see Figure 4.17). However, when the short message protocol is implemented (Figure 4.18), this added latency becomes much less of a factor. A station is forced to wait for a reservation much less frequently than before, thus the degree of differentiation is less. Again, we see that the short message protocol holds the token cycle time down for a longer period of time, and causes it to stabilize at a lower level than if the short message protocol was not implemented.

4.3.2.2. MEAN SERVICE DELAY

As we examine Figure 4.19, we see that mean service delay also increases as the ring length increases. This is not surprising, since token cycle time is also rising, which implies network access delay is rising. Again, we see (Figure 4.20) that the degree of differentiation caused by varying ring length is decreased dramatically by implementing the short message protocol.

4.3.2.3. THROUGHPUT

As the ring length increases, so does the length of what is considered a "short" message. Therefore, for a given mean message length, a higher percentage of messages will be considered "short" if the ring length increases. Hence, we see the same type of distribution of throughput curves in Figure 4.21 as we did in Figure 4.15. When the length of a message relative to the ring latency becomes a less significant factor in determining efficiency, so does the ring length itself become less significant (see Figure 4.22).

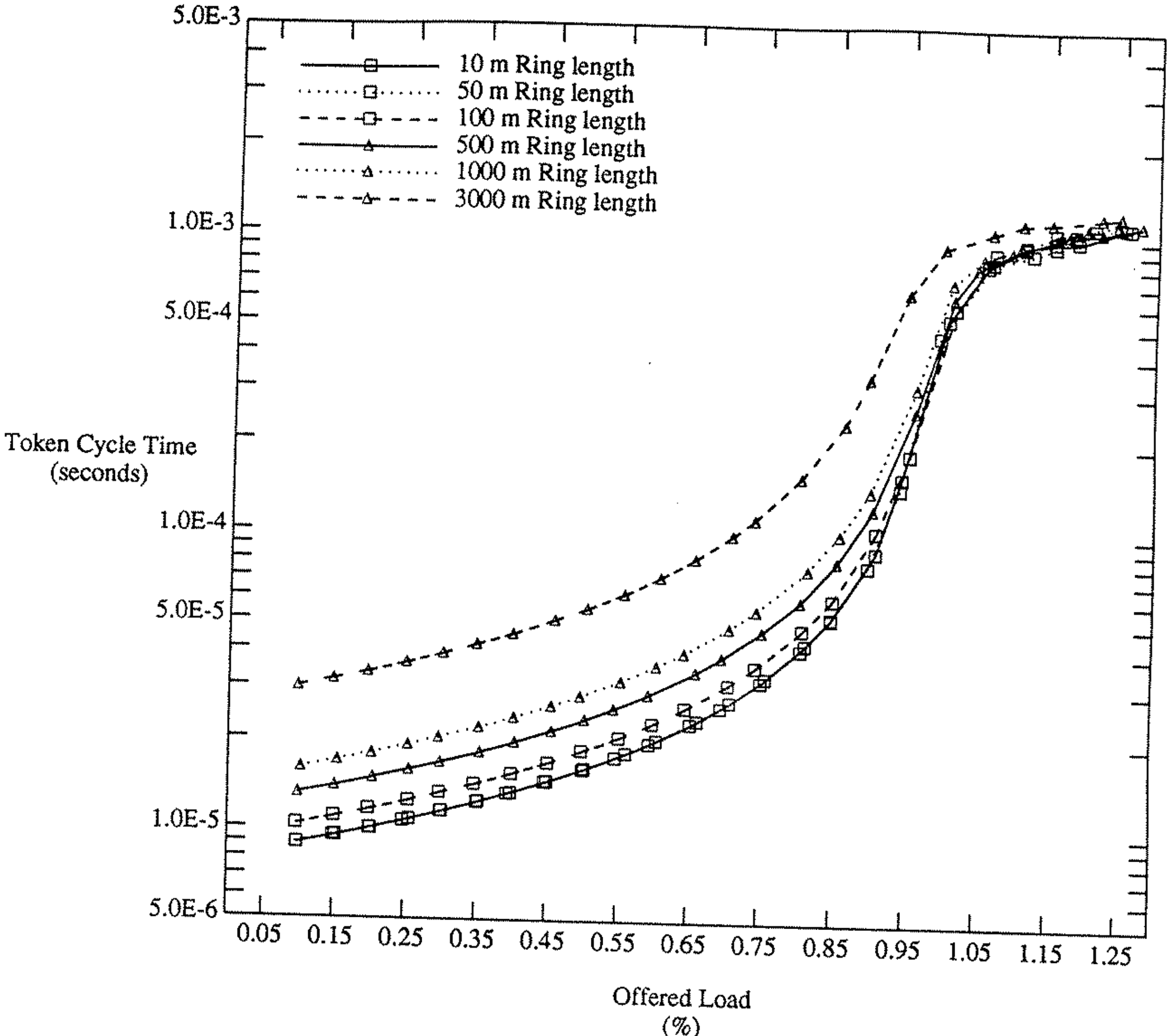


Configuration:

100 Stations
80 Mbps Medium Rate
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
1 Priority
No Short Message Protocol

Figure 4.17 — Token Cycle Time, Without SMP

Varying Ring Length

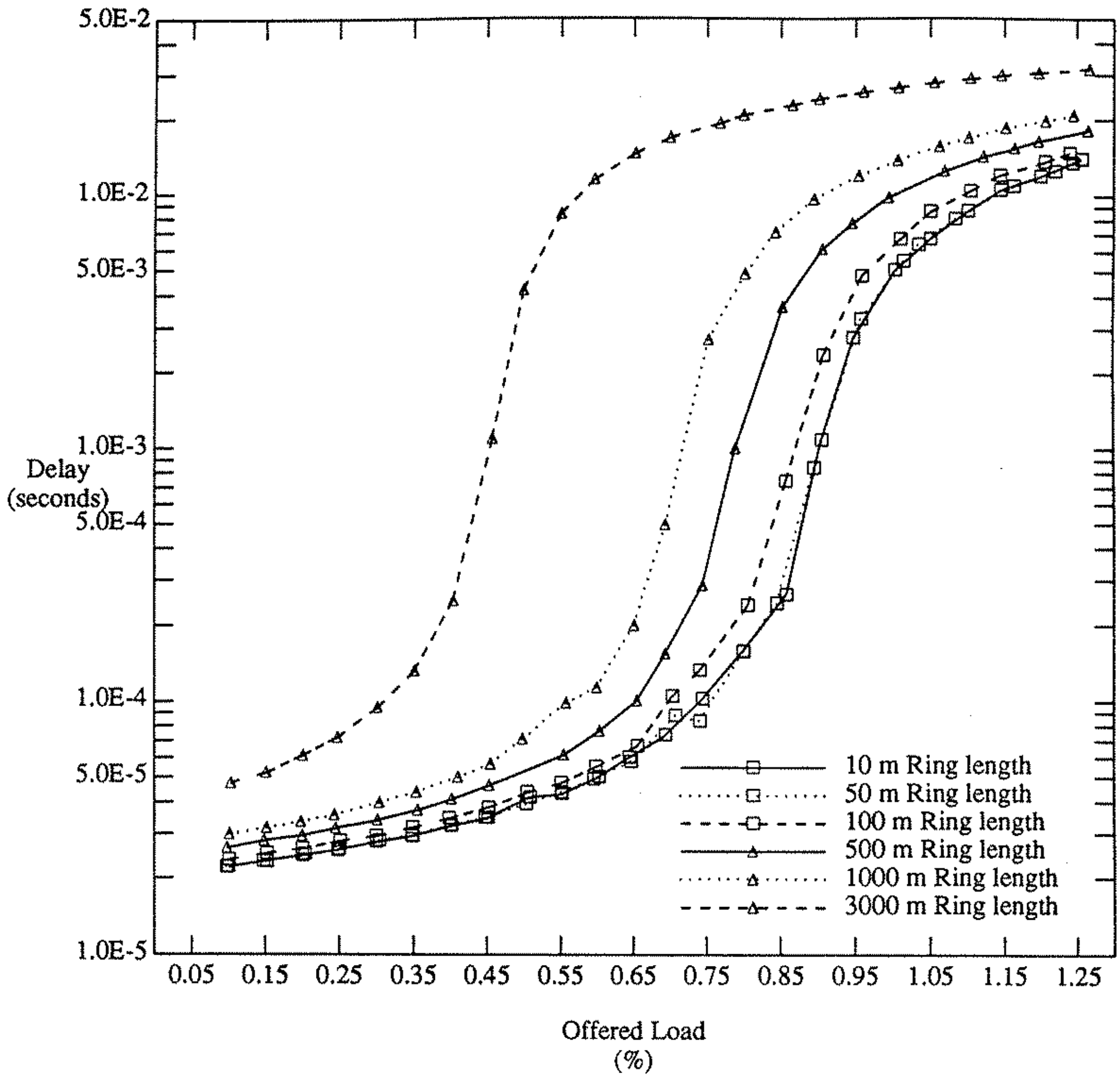


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- Short Message Protocol Implemented

Figure 4.18 — Token Cycle Time, With SMP

Varying Ring Length

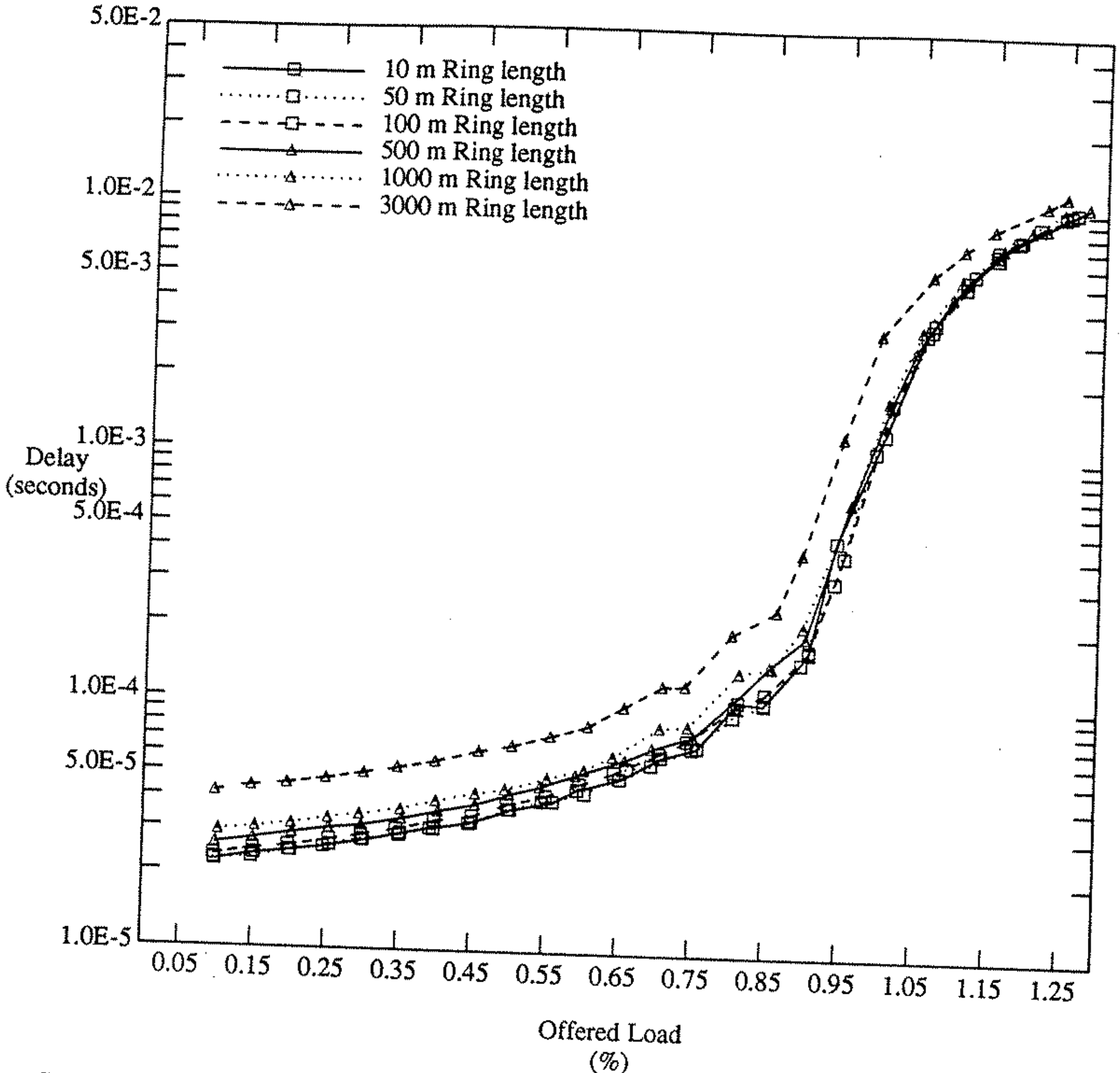


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.19 — Mean Service Delay, Without SMP

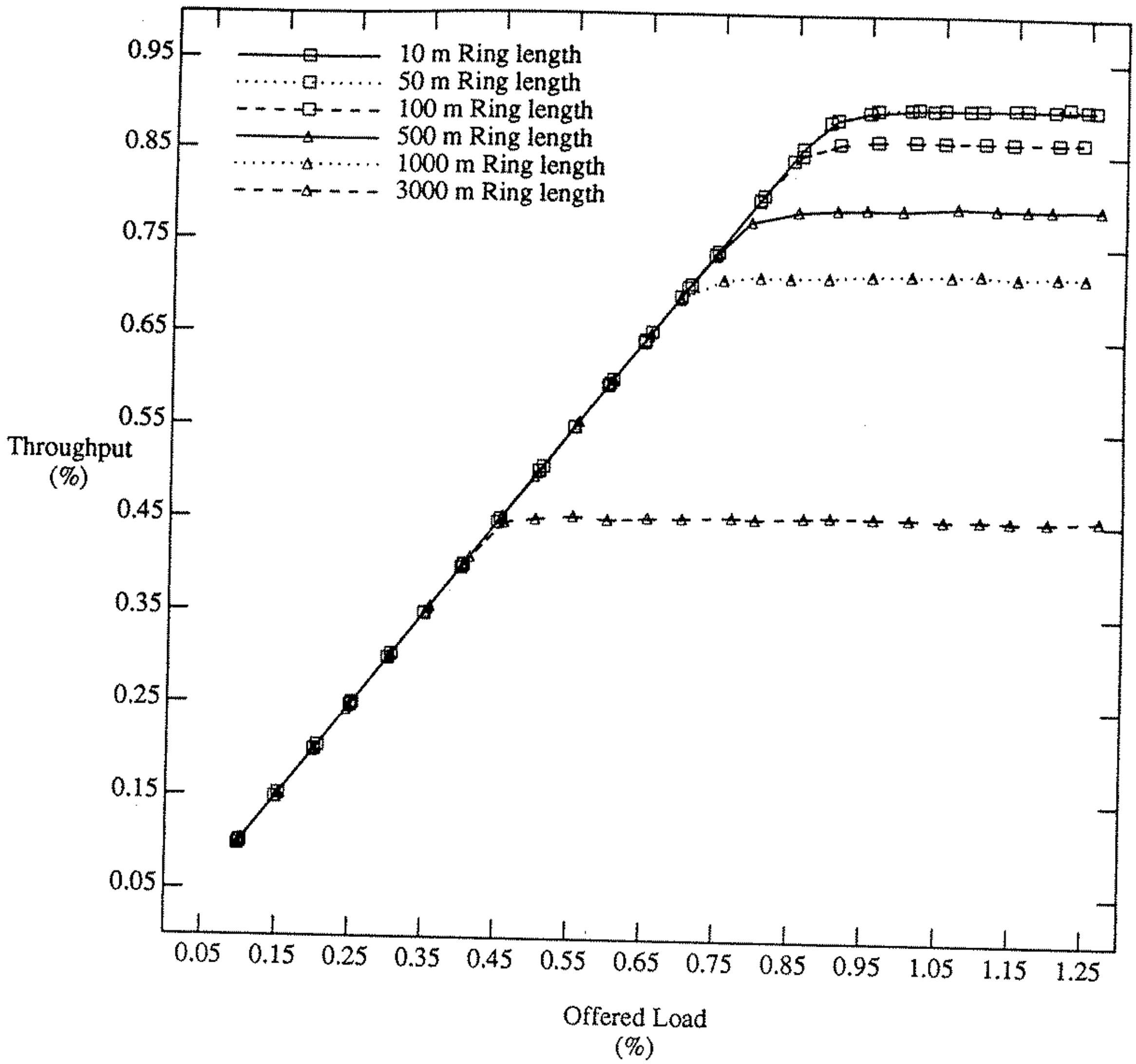
Varying Ring Length



Configuration:
- 100 Stations
- 80 Mbps Medium Rate
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- Short Message Protocol Implemented

Figure 4.20 — Mean Service Delay, With SMP

Varying Ring Length

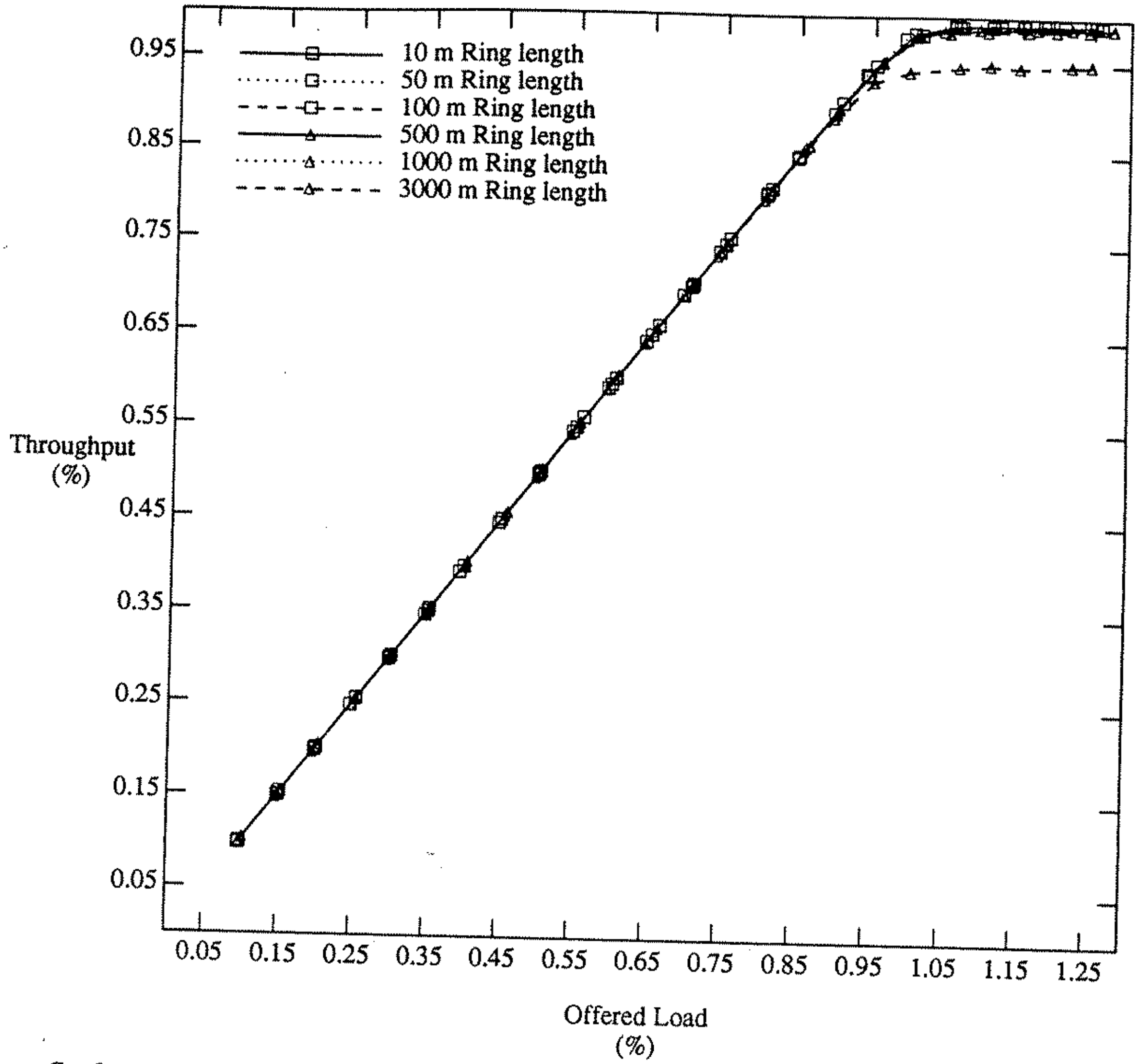


Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.21 — Throughput, Without SMP

Varying Ring Length



Configuration:

- 100 Stations
- 80 Mbps Medium Rate
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- Short Message Protocol Implemented

Figure 4.22 — Throughput, With SMP

Varying Ring Length

4.3.3. VARYING NUMBER OF STATIONS

Figures 4.23 — 4.28 show the results of the suite of simulations run with varying number of stations. The number of stations varies from a minimum of 3 to a maximum of 128, the largest number of stations allowed on a given physical ring.

4.3.3.1. TOKEN CYCLE TIME

As we see in Figures 4.23 and 4.24, as the number of stations increases, so does the token cycle time. Each extra station adds 6 bits of latency to the ring, so this is to be expected. As we have seen repeatedly, the short message protocol increases the range of offered loads throughout which the ring latency is not a significant factor in increasing delays. The mean token cycle time stays lower for higher offered loads than it does if the short message protocol is unimplemented.

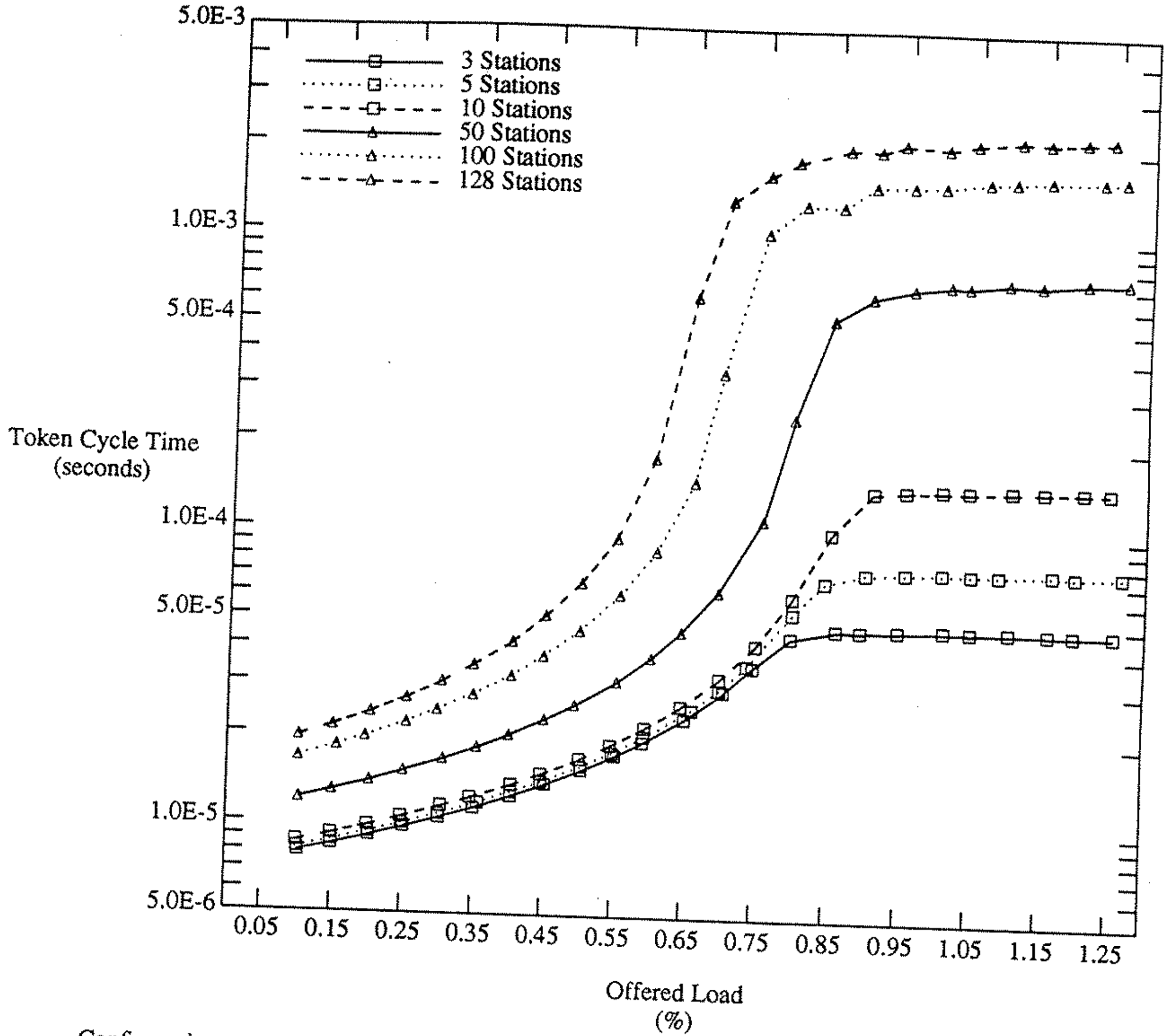
4.3.3.2. MEAN SERVICE DELAY

In Figures 4.25 and 4.26 we again see the interesting crossover effects which we noticed in Figures 4.13 and 4.14. Fewer stations implies a smaller ring latency, but it also implies that the message queues will back up sooner, causing queueing delays to increase. This tradeoff effect causes delays to be smaller for a lesser number of stations at low loads, while at high loads the delays are greater than those incurred by configurations with more stations.

4.3.3.3. THROUGHPUT

At first glance, Figures 4.27 and 4.28 appear to be the same distribution of curves as seen in Figures 4.15, 4.16, 4.21, and 4.22. There are, however, some very interesting differences. In Figure 4.27, we notice that the trend in the curves is not monotonically increasing or decreasing as a function of the number of stations. The maximum throughput is achieved by the configuration with 10 stations. This is caused by a very subtle interaction between the reservation scheme and the ring latency. For small numbers of stations, the ring latency is less than the mean message length, hence a station may transmit the token immediately upon completing transmission of the message. The more times this happens in a token cycle, the higher the throughput will be. Thus for configurations in which the mean message length exceeds the ring latency, the higher the number of stations, the higher the throughput. However, at some point the increasing number of stations causes the ring latency to exceed the mean message length. Now, each station must wait a progressively longer period of time before the reservation value returns and the token may be transmitted. Thus, as the number of stations continues to grow, throughput decreases.

In Figure 4.28, we notice that throughput appears to increase monotonically as a function of the number of stations. Since the short message protocol reduces the effects of the reservation scheme dramatically, this is, indeed, as we would expect from the first part of the preceding argument.

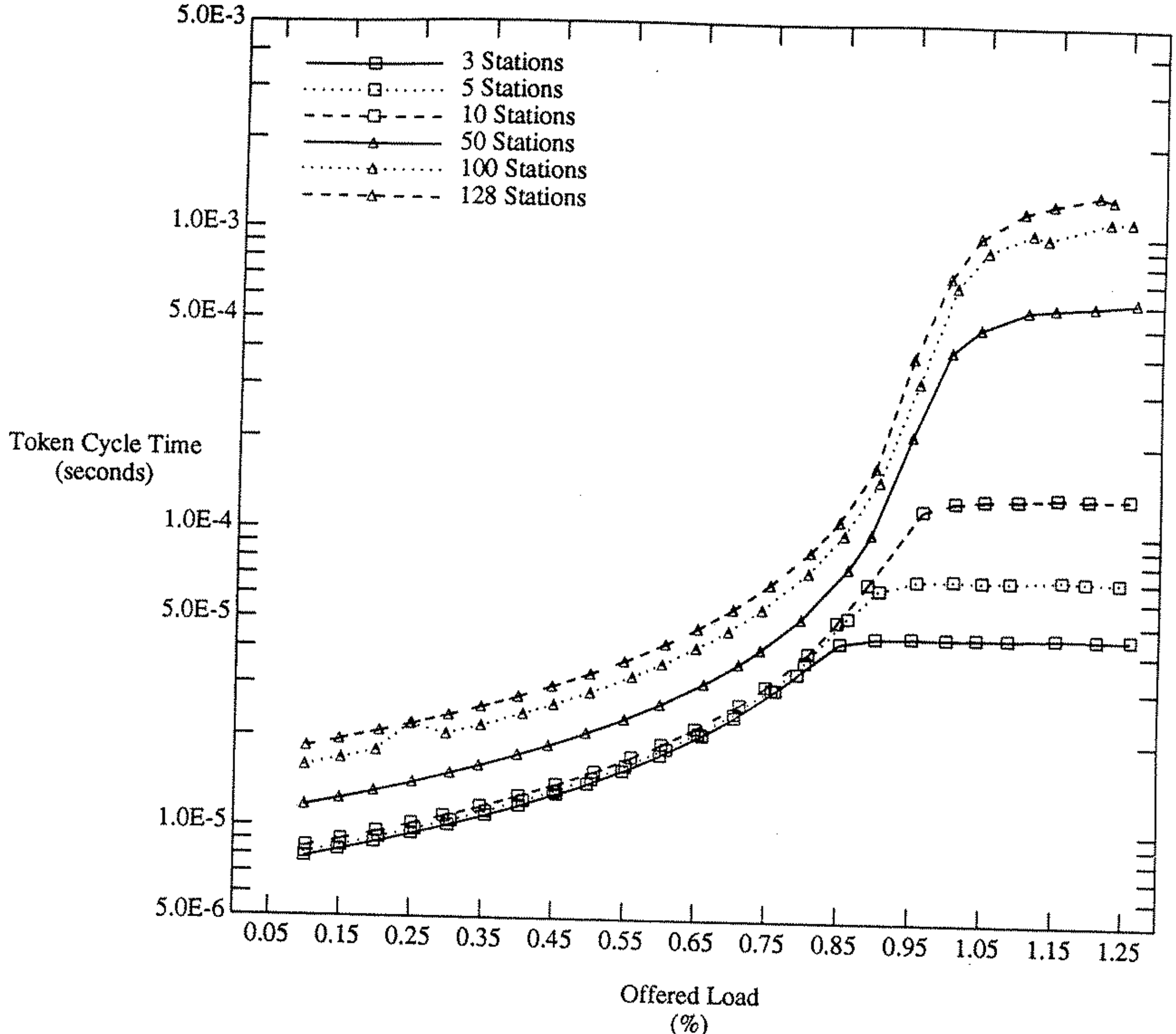


Configuration:

- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- No Short Message Protocol

Figure 4.23 — Token Cycle Time, Without SMP

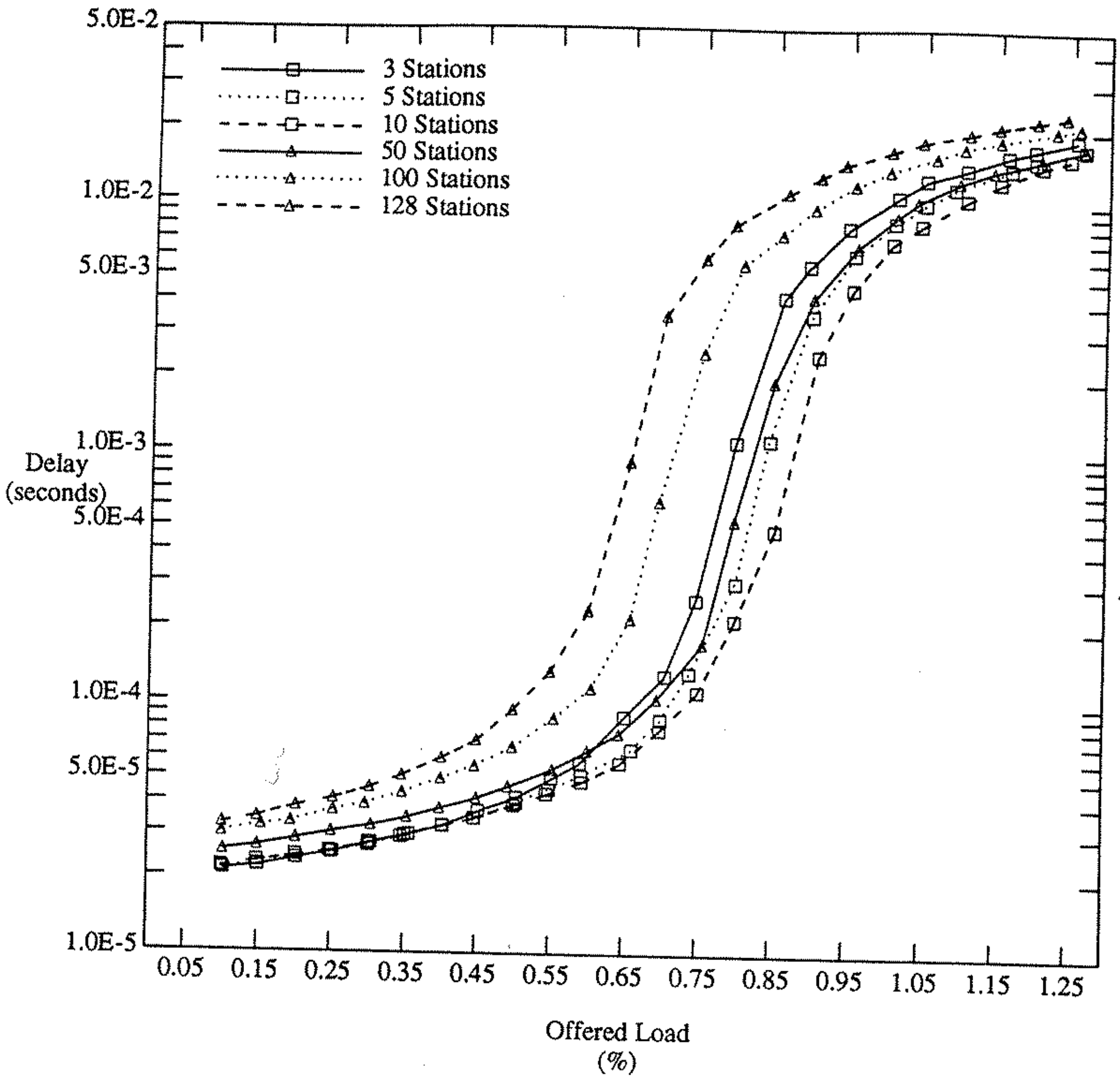
Varying Number of Stations



Configuration:
- 80 Mbps Medium Rate
- 1000 m Ring Circumference
- 6 Bits Internal Station Latency
- 40 Bits Latency Buffer
- Exponential Packet Arrival Distribution
- 1024 Bit Mean Packet Length Distributed Exponentially
- 1 Priority
- Short Message Protocol Implemented

Figure 4.24 — Token Cycle Time, With SMP

Varying Number of Stations

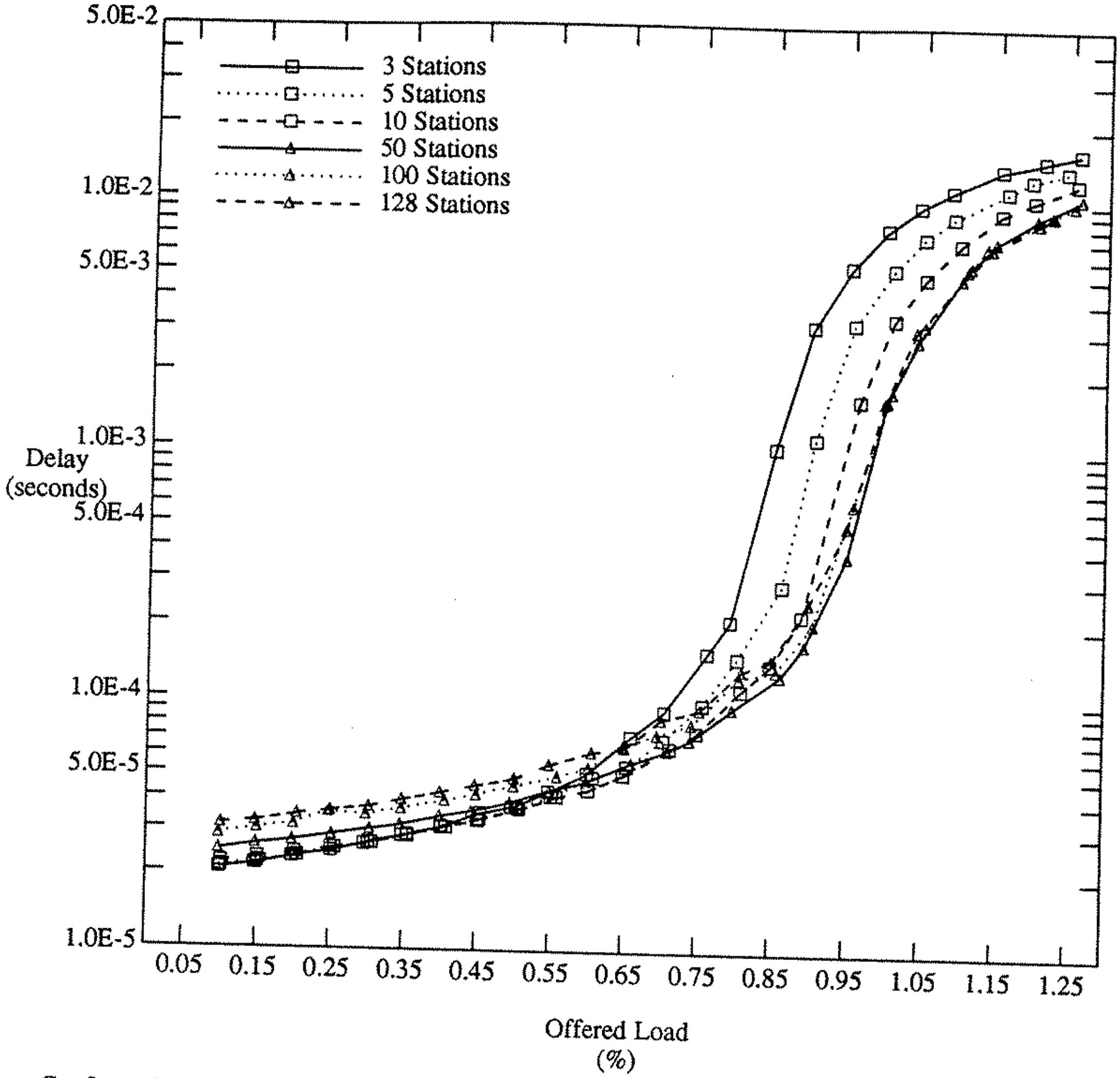


Configuration:

80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
1 Priority
No Short Message Protocol

Figure 4.25 — Mean Service Delay, Without SMP

Varying Number of Stations

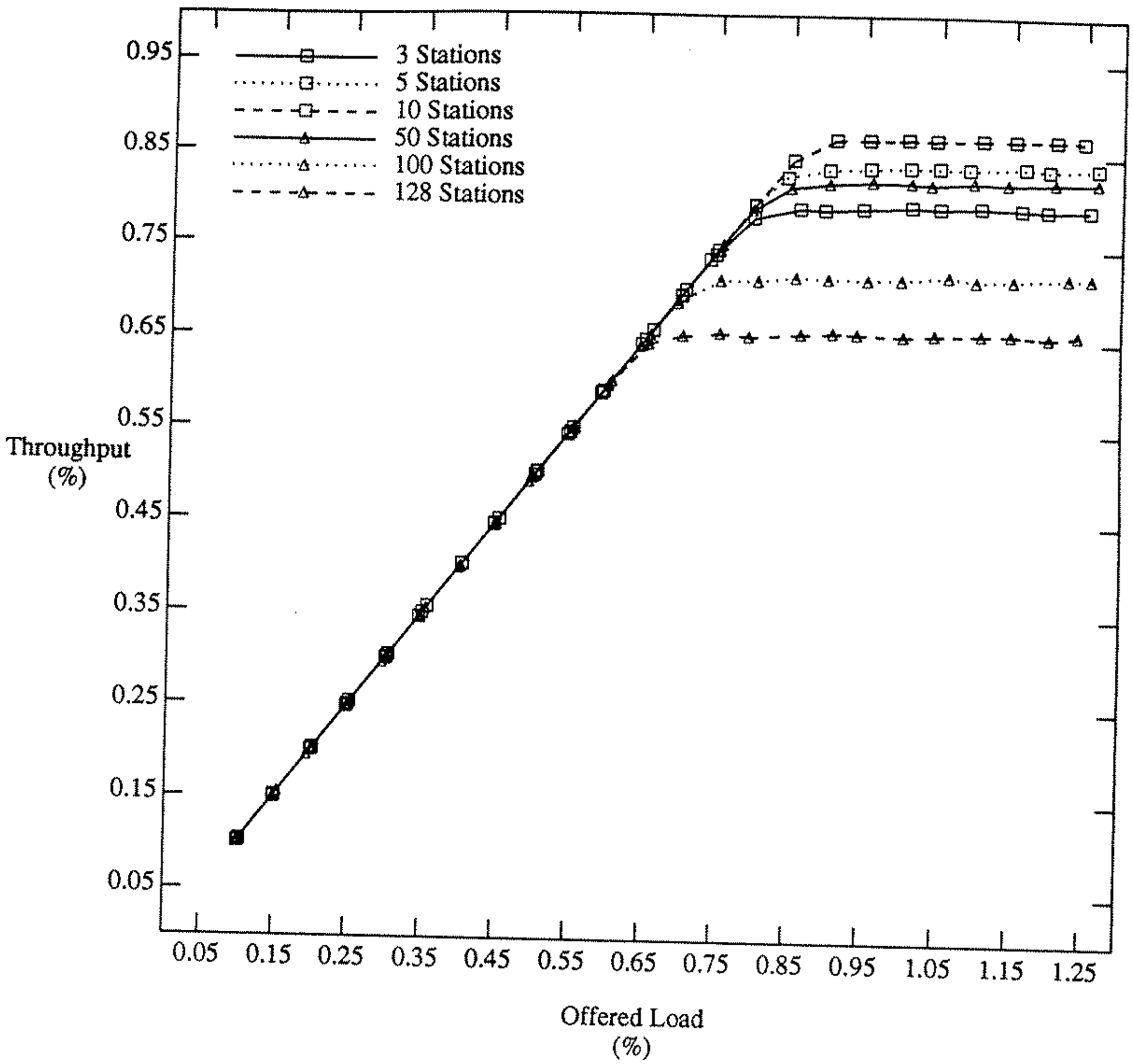


Configuration:

80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
1 Priority
Short Message Protocol Implemented

Figure 4.26 — Mean Service Delay, With SMP

Varying Number of Stations

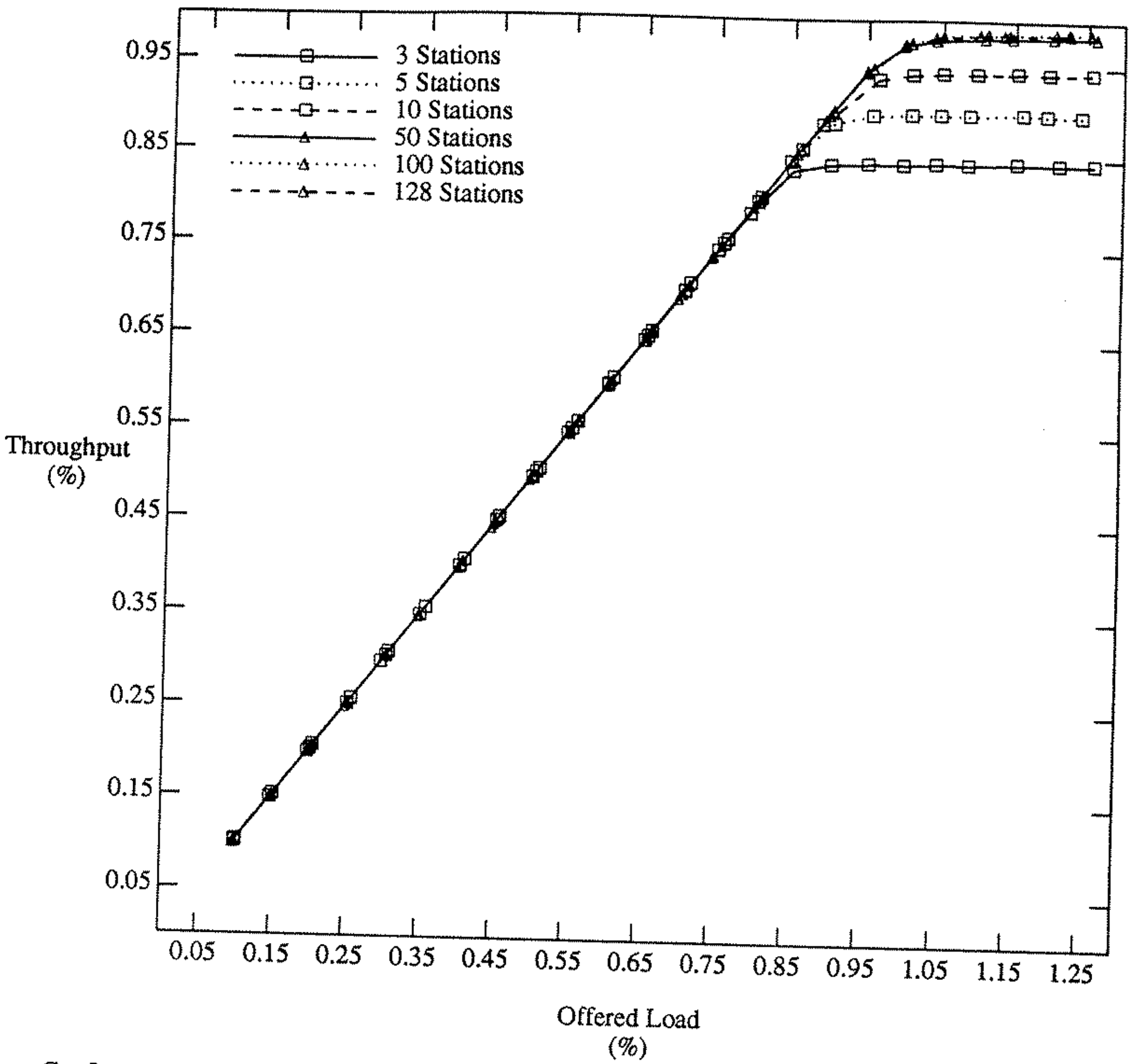


Configuration:

80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
1 Priority
No Short Message Protocol

Figure 4.27 — Throughput, Without SMP

Varying Number of Stations



Configuration:

80 Mbps Medium Rate
1000 m Ring Circumference
6 Bits Internal Station Latency
40 Bits Latency Buffer
Exponential Packet Arrival Distribution
1024 Bit Mean Packet Length Distributed Exponentially
1 Priority
Short Message Protocol Implemented

Figure 4.28 — Throughput, With SMP

Varying Number of Stations

Chapter 5

CONCLUSIONS

5.1. SUMMARY OF ANALYSIS

As we attempt to analyze the HSRB protocol, we must bear in mind the desired performance characteristics discussed in Section 1.3.1. We must view the protocol in this context if we are to accurately judge its performance. Of those four goals (real-time response, reliability and fault tolerance, hardware independence, and network-wide prioritized access), we dealt with two in Chapter Two. We saw that the protocol establishes the mechanisms by which reliability and fault tolerance may be achieved. Furthermore, at all but the lowest layers of the protocol it is written in a hardware-independent manner, hence we assume these goals to have been achieved. The other two goals were the focus of the analysis in Chapter Four, and we summarize the results below.

5.1.1. REAL-TIME RESPONSE

The definition of what constitutes real-time response is very subjective and, of course, varies with the application domain. However, one commonly accepted value is 10 to 20 ms. We saw for all configurations simulated that at low loads service delays were typically several orders of magnitude less than 10 ms. Even at high loads, network access delay stabilized well below this threshold. From this we observe that queueing delay is the predominant factor in cases where service delay exceeds our arbitrary 10 ms threshold. One way to reduce queueing delay (for a certain percentage of the messages) is to implement the priority mechanism. Higher priority messages suffer much less queueing delay, consequently they incur much smaller service delays. If only a portion of the network traffic requires real-time service, this may be guaranteed over very high loads by assigning those messages a higher priority.

We observed that the value at which throughput stabilized varied wildly for different configurations. In all cases, however, efficiency was increased a great deal through the implementation of the short message protocol. The degree to which the short message protocol improved efficiency is not governed by any single factor. Rather, the relationship between the mean message length and the ring latency was the major influence seen. The higher the percentage of messages which are shorter than the ring latency, and the longer the ring latency is in relation to the mean message length, the more of an impact the short message protocol will have on efficiency.

5.1.2. PRIORITY OPERATION

In the base configuration comparisons, we saw examples of the priority scheme in operation. It appears to behave as one would intuitively expect it to. We observed a high degree of differentiation in service delays for the various priorities. Higher priority messages were guaranteed timely delivery while the service to lower priority messages gradually degraded.

At low offered loads, we noted that the mean token cycle time and throughput were the same as for the single priority case, therefore the priority mechanism implementation incurred no performance penalty. However, at high loads the multiple priority cases were less efficient than their single priority counterparts.

Implementing the short message protocol in conjunction with the priority mechanism extended the range of offered loads for which single priority and multiple priority operation were comparable. By doing this we have, in effect, a two-tiered priority scheme: as long as the network is behaving in a stable fashion (i.e. no time-dependent queue backup) we effectively have a single priority system. All priorities receive essentially the same level of service, that of the single priority case. However, as the network starts to be overloaded, the priority mechanism takes effect and results in differing levels of service.

5.1.3. SHORT MESSAGE PROTOCOL

As we have seen repeatedly, the short message protocol is a very effective means of increasing the efficiency of the network. Furthermore, implementing the short message protocol results in lower mean service delays than would otherwise occur. The short message protocol does, however, result in less priority differentiation at low offered loads. But, since all priorities receive high quality service at those loads, this is not a drawback.

We conclude from the above that the short message protocol should, indeed, be implemented. In the simulated cases we observed no negative effects from said implementation, and, in fact, saw significant performance benefits. Indeed, in cases where the mean message length is much less than the ring latency the short message protocol could hardly be considered optional. The network performs so inefficiently in those cases that the short message protocol is necessary to achieve acceptable service.

5.2. FUTURE RESEARCH

Many aspects of the HSRB protocol are not yet fully understood. One such issue is the behavior of the priority mechanism in a wide variety of network configurations. Many questions arise from the use of priorities including 1) how should message priorities be determined, 2) where should those priorities be assigned, and 3) what is the optimal number of priorities to use in a given situation. These questions deserve further study.

The simulations on which this analysis is based assume homogeneous message traffic on the ring. A detailed study of the performance of heterogeneous networks seems in order. The simulation environment is easily capable of modeling such networks, but determining what types of heterogeneity to model is a more complex question. Certainly, network designers who well understand the message traffic which would be present on a given network could easily use the system to determine how the HSRB would perform for that application.

One of the original motivations for this research was comparison of the SAE HSRB with the ANSI FDDI token ring. This comparison has yet to be carried out, but it will surely be completed in the near future. Comparison of these protocols would yield insight about which network to use for a particular application.

At present, no hardware implementation of the HSRB protocol exists. When such an implementation becomes available,, one could measure its performance and compare this to the results of the simulation analysis. One would anticipate that the simulated performance would be better than that of an actual implementation, since a physical implementation must deal with many issues not considered in the simulation.

Finally, it is still uncertain how to propagate the performance benefits of a very fast network through the protocol stack to an application process. Various methods have been proposed, but no definitive answer has yet been reached. At present, certainly, the software overhead inherent in a full protocol stack prevents an application process from receiving the full benefits of very fast transmission rates.

Bibliography

[BERG87a] C. Berggren, Definitions and Requirements for Distributed Real-time Systems, *Communication Requirements for Real-time Systems Workshop, Printed Notes*, National Bureau of Standards, Gaithersburg, MD (June 1987).

[BERG87b] C. Berggren, *Private Communication* (Sept. 1987).

[GEYE87] M. Geyer, *Private Communication* (Sept. 1987).

[KLEI76] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*, John Wiley and Sons, New York (1976).

[LAVE83] S. Lavenberg and P. Welch, *Computer Performance Modeling Handbook*, Steven Lavenberg ed., Academic Press, New York (1983).

[PEDE87] J. Peden, Performance Analysis of the IEEE 802.5 Token Ring, *University of Virginia Computer Science Report No. RM-86-06*, University of Virginia, Computer Science Dept. (June 1986).

[PEDE88] J. Peden, Unpublished work, University of Virginia, Computer Science Dept. (1988).

[SAE87] SAE AE—9B Subcommittee, Final Draft High Speed Ring Bus (HSRB) Standard, *SAE Ref. AS4074.2* (June 1987).

[SCHW87] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley Publishing Company, Reading, MA (1987).

[SIMO88] R. Simonson, Performance Analysis of the FDDI Token Ring, *University of Virginia Computer Science Report No. TR-88-02*, University of Virginia, Computer Science Dept. (Feb. 1988).

[SUMM85] C. Summers and A. Weaver, Performance Analysis of the IEEE 802.4 Token Bus, *University of Virginia Research Memo #RM-85-06*, University of Virginia, Computer Science Dept. (June 1985).