PERFORMANCE EVALUATION AND COMPARISON
OF A TOKEN RING NETWORK
WITH FULL LATENCY STATIONS AND DUAL LATENCY STATIONS

by

EDWARD CHI LUP LO

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
Electrical Engineering

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
May 1988
©Edward Chi Lup Lo, 1988
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of **ELECTRICAL ENG.**

The University of British Columbia
Vancouver, Canada

Date **May 26, 1988**
ABSTRACT

A method of performance improvement of token ring networks is presented, based on the use of stations with two latency states. Station latency is defined as the time delay introduced in passing data through a station. Most token ring protocol standards (e.g. IEEE 802.5 or ANSI X3T9.5) require incoming data to be decoded and encoded in the station before transmission onto the ring. These encoding and decoding operations add significantly to the station latency. The bypassing of the encoding and decoding steps is proposed, which improves the mean message waiting time. A detailed evaluation and comparison of the networks is based on both analytical and simulation results. The performance of identical stations and symmetric traffic is obtained analytically. A discrete event simulation model for a token ring network is written in GPSS for general traffic. Results show a significant reduction in mean waiting time for the dual latency ring, with performance approaching or exceeding that of gated and exhaustive service, for certain ranges of network utilization.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iii</td>
</tr>
<tr>
<td>TABLE OF FIGURES</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>viii</td>
</tr>
</tbody>
</table>

## I. OBJECTIVE AND OUTLINE OF THE THESIS ........................................ 1

## II. INTRODUCTION .................................................................................. 3

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Local Area Networks</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Token Ring Local Area Networks</td>
<td>4</td>
</tr>
<tr>
<td>2.2.1 History and Developments</td>
<td>4</td>
</tr>
<tr>
<td>2.2.2 Basic Operations and Service Disciplines</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Performance Measure</td>
<td>8</td>
</tr>
<tr>
<td>2.4 Importance of Performance Modeling</td>
<td>9</td>
</tr>
</tbody>
</table>

## III. IMPLEMENTATION ........................................................................... 12

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 The IEEE 802.5 Standard</td>
<td>13</td>
</tr>
<tr>
<td>3.2 The Dual Latency Method</td>
<td>16</td>
</tr>
</tbody>
</table>

## IV. ANALYTICAL MODEL ......................................................................... 18

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Assumptions</td>
<td>18</td>
</tr>
<tr>
<td>4.2 Notations</td>
<td>19</td>
</tr>
<tr>
<td>4.3 Stability Requirements</td>
<td>21</td>
</tr>
<tr>
<td>4.4 Analysis of a Completely Symmetric Ring</td>
<td>21</td>
</tr>
<tr>
<td>4.5 Analysis of a General Ring</td>
<td>23</td>
</tr>
<tr>
<td>4.6 Analysis of a Dual Latency Ring</td>
<td>25</td>
</tr>
</tbody>
</table>

## V. SIMULATION MODEL .......................................................................... 29

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Why Simulate ?</td>
<td>29</td>
</tr>
<tr>
<td>5.2 Drawbacks of Using Simulation</td>
<td>30</td>
</tr>
<tr>
<td>5.3 Language</td>
<td>31</td>
</tr>
<tr>
<td>5.4 Models</td>
<td>32</td>
</tr>
<tr>
<td>5.5 Statistical Considerations</td>
<td>58</td>
</tr>
<tr>
<td>5.5.1 Independent Replications of Runs</td>
<td>59</td>
</tr>
<tr>
<td>5.5.2 The Regenerative Method</td>
<td>61</td>
</tr>
<tr>
<td>5.6 Validation</td>
<td>64</td>
</tr>
<tr>
<td>5.7 Features of the Simulation Models</td>
<td>65</td>
</tr>
</tbody>
</table>

## VI. NUMERICAL AND SIMULATION RESULTS ........................................... 69

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Low Speed LANs</td>
<td>70</td>
</tr>
<tr>
<td>6.1.1 Completely Symmetric Cases</td>
<td>71</td>
</tr>
<tr>
<td>6.1.2 Asymmetric Cases</td>
<td>82</td>
</tr>
<tr>
<td>6.2 High Speed LANs</td>
<td>95</td>
</tr>
<tr>
<td>6.2.1 Completely Symmetric Cases</td>
<td>95</td>
</tr>
<tr>
<td>6.2.2 Asymmetric Cases</td>
<td>110</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.1</td>
<td>A Typical Structure of a Token Ring</td>
</tr>
<tr>
<td>3.1</td>
<td>Token Format</td>
</tr>
<tr>
<td>3.2</td>
<td>Frame Format</td>
</tr>
<tr>
<td>3.3</td>
<td>Access Control field, Ending Delimiter field and Frame Status field</td>
</tr>
<tr>
<td>4.1</td>
<td>A Model of Polling Systems</td>
</tr>
<tr>
<td>5.1</td>
<td>Model Segment for Limited-to-one Service Discipline with Multiple-Token Operation and Regenerative Method</td>
</tr>
<tr>
<td>5.2</td>
<td>Model Segment for Gated Service Discipline with Multiple-Token Operation and Regenerative Method</td>
</tr>
<tr>
<td>5.3</td>
<td>Model Segment for Exhaustive Service Discipline with Multiple-Token Operation and Regenerative Method</td>
</tr>
<tr>
<td>5.4a</td>
<td>Model Segment for Single-Token operation with Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.4b</td>
<td>Model Segment for Single-Token operation with Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.4c</td>
<td>Model Segment for Single-Token operation with Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.5a</td>
<td>Model Segment for Single-Packet operation with Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.5b</td>
<td>Model Segment for Single-Packet operation with Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.5c</td>
<td>Model Segment for Single-Packet operation with Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.6a</td>
<td>Model Segment for the Avalanche Simulation with Multiple-token Operation, Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.6b</td>
<td>Model Segment for the Avalanche Simulation with Multiple-token Operation, Limited-to-one Service Discipline and Regenerative Method</td>
</tr>
<tr>
<td>5.7</td>
<td>Model Segment for the Timer Model of the Avalanche Simulation</td>
</tr>
<tr>
<td>5.8</td>
<td>Model Segment for the Turbo mode</td>
</tr>
<tr>
<td>6.1</td>
<td>Performance Comparison of Case 1A for 20 stations</td>
</tr>
<tr>
<td>6.2</td>
<td>Performance Comparison of Case 1A for 100 stations</td>
</tr>
<tr>
<td>6.3</td>
<td>Performance Comparison of Case 1A for 200 stations</td>
</tr>
<tr>
<td>6.4</td>
<td>Performance Comparison of Case 1A for 1000 stations</td>
</tr>
<tr>
<td>6.5</td>
<td>Performance Improvement of Full Latency Ordinary Service to Dual Latency Ordinary Service for Case 1A</td>
</tr>
</tbody>
</table>
Figure 6.6 - Performance Improvement of Full Latency Exhaustive Service to Dual Latency Ordinary Service for Case 1A .................................................. 80
Figure 6.7 - Performance Improvement of Full Latency Exhaustive Service to Dual Latency Ordinary Service for Case 1B .................................................. 81
Figure 6.8 - Performance Improvement of Full Latency to Dual Latency for Case 2, different mean message lengths . . 85
Figure 6.9 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2A .................................................. 86
Figure 6.10 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2B .................................................. 87
Figure 6.11 - Performance Comparison of Full Latency with Different Service Disciplines with the Dual Latency for Case 2C .................................................. 88
Figure 6.12 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2D .................................................. 89
Figure 6.13 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2E .................................................. 90
Figure 6.14 - Performance Improvement of Full Latency to the Dual Latency for Different Message Distributions (Case 3) .................................................. 92
Figure 6.15 - Performance Comparison of Full Latency with Different Message Distributions for Case 3 ................. 93
Figure 6.16 - Performance Comparison of Dual Latency with Different Message Distributions for Case 3 .................. 94
Figure 6.17 - Performance Comparison of Full Latency with Different Message Lengths for Case 4, 20 Stations .......... 98
Figure 6.18 - Performance Comparison of Dual Latency with Different Message Lengths for Case 4, 20 Stations ....... 99
Figure 6.19 - Performance Comparison of Full Latency with Different Message Lengths for Case 4, 100 Stations .... 100
Figure 6.20 - Performance Comparison of Dual Latency with Different Message Lengths for Case 4, 100 Stations ...... 101
Figure 6.21 - Performance Comparison of Full Latency with Different Message Lengths for Case 4, 1000 Stations ..... 102
Figure 6.22 - Performance Comparison of Dual Latency with Different Message Lengths for Case 4, 1000 Stations .... 103
Figure 6.23 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4B, 100 Stations ........................................... 104
Figure 6.24 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4D, 100 Stations ........................................... 105
Figure 6.25 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4B, 1000 Stations .......................................... 106
Figure 6.26 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4D, 1000 Stations ............................................. 107
Figure 6.27 - Performance Improvement of Full Latency Ordinary Service to the Dual Latency Ordinary Service for Case 4A ................................................................. 108
Figure 6.28 - Performance Improvement of Full Latency Ordinary Service to the Dual Latency Ordinary Service for Case 4D ................................................................. 109
Figure 6.29 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency Ordinary Service for Case 5 .................................................... 113
Figure 6.30 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency Ordinary Service for Case 6 .................................................... 114
Figure 6.31 - Performance Improvement of Full Latency Ordinary Service to the Dual Latency Ordinary Service for Case 5 and Case 6 ..................................................... 115
ACKNOWLEDGMENTS

I am always indebted to my supervisors Dr. R.H.S. Hardy and Dr. R.W. Donaldson for their consistent encouragement and support throughout the course of my research.

Special thanks are given to Dr. H.W. Lee and Dr. J.K. Cavers for their helpful comments and suggestions. My research colleague, Mr. Ian Radziejewski, has also given me a lot of help and inspiration.

Finally, financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.
I. OBJECTIVE AND OUTLINE OF THE THESIS

The purpose of this thesis is to investigate and verify the performance improvement that can be made by the use of the dual latency technique [Hard87], [Hard88a]. The performance of the MAC Layer of the OSI model is described by both analytical and simulation models. The physical layer is modeled as a delay caused by the ring medium and the station interface. The analytical model is based upon queueing models for token rings, primarily in the context of polling systems [Taka86]. The existing simulation model was written in GPSS and can be used to simulate the performance of a token passing (token ring or token bus) network. This simulation model enables us to look at the situations where predictions from the analytical model may not be accurate.

The organization of the thesis is as follows:

2. Introduction: This chapter describes the background of Local Area Networks (LANs) and token ring LANs. The importance of performance modeling is also briefly mentioned.

3. Implementation: A brief description of the IEEE 802.5 standard and the implementation of the dual latency method
is presented.

4. Analytical Model: The purpose of this chapter is to give the assumptions and equations used for the performance analysis of a completely symmetric ring.

5. Simulation Model: Various aspects of the simulation models are described. These include: strategy, simulation language used, block diagrams, statistical considerations and difficulties encountered.

6. Results: Numerical and simulation results are presented and the conclusions drawn are explained.

7. Conclusions and Recommendations: The conclusions and recommendations that can be drawn from this research are described.

8. Suggestions for Future Work: Real-time performance modeling becomes a natural extension of the thesis. Further, end-to-end performance modeling may be achieved by building up layer by layer.
II. INTRODUCTION

2.1 Local Area Networks

Local Area Networks (LANs) have become increasingly important in recent years. This trend is mainly due to the increasing use of small single-function systems, such as personal computers, business computer and workstations. These small systems generally provide a more accessible and responsive environment for users. They are easier to learn and more reliable than large central time-sharing systems. LANs are designed for interconnecting these systems and for sharing resources and information (data and/or voice) within a small area.

There are two more driving forces behind this trend. First, progress in Very Large Scale Integration (VLSI) technology makes the implementation of a low cost Network Interface Unit (NIU) or network adapter possible. Second, low cost high speed transmission media, such as optical fiber, are available.
2.2 Token Ring Local Area Networks

2.2.1 History and Developments

The token ring system is probably the oldest ring LAN, first proposed by Farmer and Newhall [Farm69] in 1969 and then referred to as a Newhall ring. Companies such as IBM, Apollo and Sperry seem committed to the token ring. IBM Zurich Research Laboratory, Ruschlikon, Switzerland has been doing intensive research on the token ring and many papers were published about the token ring system by people working there ([Bux81], [Bux83a], [Bux83b], [Bux84], [Kell83], [Wong84]). The importance of the token ring is particularly evident in the fast development of IEEE Computer Society Local Network Committee (Project 802.5) and American National Standard Committee (ANSC) X3T9.5. The IEEE 802.5 token ring can accommodate up to 250 stations with shielded twisted pair at 4 Mbit/s data rate and the FDDI token ring up to 1000 stations with 200 km of fiber at 100 Mbit/s data rate.

Texas Instruments has a joint venture with IBM to develop a NIU for the token ring local network devised by IBM. In addition to the medium access control functions listed in the IEEE 802.5 standard [IEEE802.5], the firmware supports
fine tuning of the star-wired ring link, maintenance of growing networks, error logging and reporting, and multi-ring management. The NIU consists of three very large scale integrated circuits and two medium-scale integrated cable interface chips.

Advanced Micro Devices Corporation recently announced the SUPERNET Family chip set for networks using the ANSI X3T9.5 standard called Fiber Distributed Data Interface (FDDI) [AMD86]. The chip set includes a RAM Buffer Controller (RBC), a Data Path Controller (DPC), a Fiber Optic Ring Media Access Controller (FORMAC), a Encoder/Decoder (ENDEC), and a Transparent Asynchronous Transmitter/Receiver Interface (TAXI). The details of the implementation are not yet available. An introduction to the FDDI standard can be found in [Ross86] and a comparison between the FDDI and IEEE 802.5 standard in [Iyer87].

2.2.2 Basic Operations and Service Disciplines

There are two terms that have to be clarified first before proceeding. Station latency is defined as the time delay introduced in passing data through a station while ring
latency is the sum of the delays around the ring. That is, ring latency is equal to the sum of all the station latencies and propagation delays.

Figure 2.1 depicts a typical structure of a token ring. A token is passed sequentially from station to station around the ring. The token contains a flag indicating whether the token is "free" or "busy". The station which has messages to send must wait until a free token passes by, change the status of the token from free to busy and transmit the data. Physically, a token can be a control bit in the free token or in the header of a message. The transmitting station is responsible for generating a new free token when its transmission is finished.

There are three ways a transmitting station can generate a new free token. The method used can greatly affect the performance. They are: single-packet, single-token and multiple-token operation. For single-packet operation, a transmitting station waits until all its transmitted bits have travelled around the ring and been removed. Then the station releases a new free token. This operation is the most conservative type of the three since it ensures that there is only one transmission at any point in time. Very
few existing token ring LANs use this approach because it does not utilize the token efficiently. Nevertheless, it is the simplest to implement.

Figure 2.1 - A Typical Structure of a Token Ring

For single-token operation, a transmitting station only waits until its busy token or header comes back before releasing a new free token. If, however, the message transmission time is longer than the ring latency, the station will get its busy token before completing data transmission. In this case, a new free token will be released right after data transmission.

Multiple-token operation is the most efficient of the three. A transmitting station releases a new free token right after
its data transmission. If the message transmission time is shorter than the ring latency, it is possible to have several busy tokens and one free token on the ring at the same time, as the name implies. If the message transmission time is longer than the ring latency, multiple-token operation is the same as single-token operation.

SERVICE DISCIPLINES

There are three main service disciplines that are different depending on the number of messages which may be served in a queue after a free token is captured; they are exhaustive, gated, limited service disciplines. Different disciplines are treated in [Hofr86]. For exhaustive service, the server continues to serve the station until the queue is empty. For gated service, the server serves only those messages which are queued at the polling instant. For limited-to-m service, the server serves at most m messages. Therefore, when m is one, it is called limited-to-one service or ordinary service.

2.3 Performance Measure

The mean message response time is the single most important performance measure in most computer communications networks.
The message response time is defined as the time between the start of a request to transmit and the end of its transmission. The usual performance measure used is the mean message waiting time which is the time between the start of a request to transmit and the beginning of its transmission. There is a simple relationship between the mean response time and the mean message waiting time. The mean message response time is equal to the sum of mean message waiting time and mean service time. Throughout the thesis, the mean message waiting time is used as the measure.

2.4 Importance of Performance Modeling

The significance of performance modeling is stressed from the management point of view. Modeling may be involved in the design and development phase of a new system or the configuration and tuning of an existing system. Results from performance modeling should be able to provide "good" recommendations to the designers. A wrong recommendation in design and development will usually affect the organization more than one in configuration and tuning.

There are several ways to obtain performance modeling. The most simple one is to measure the performance directly under a certain system loading for each alternative. Measurement is
essential in performance modeling because it deals with the real system. But there are two drawbacks: it may not be possible or feasible to build alternatives so quickly and it is usually very costly. Therefore, this research project would, eventually, use measurement to validate the model. The research team has already designed the chip set for IEEE 802.5 and some of the chips are ready to be tested.

The second approach is to apply probability and queueing theory to model performance. The main advantage is that it requires less computational effort than a simulation. The model, however, has to make some mathematically tractable assumptions which may not be applicable to the system. For the research interest, the exact analytical solutions for ordinary service asymmetric traffic are not available, so the third approach is stressed more.

The third approach is to apply discrete event simulation. The Simulation is a powerful and flexible approach. The details of the model can be controlled more readily. Nevertheless, it usually requires more computational effort and it is more difficult to prove the correctness. It is the controllability of the details of the model that is important for this
research since the performance can be built layer by layer, which is very important to the design and development phase of the proposed new system.
III. IMPLEMENTATION

In order to test the performance of a dual latency station with realistic hardware parameters, the IEEE 802.5 token ring standard [IEEE802.5] was used as the basis for an implementation of a network interface unit utilizing the dual latency method [Radz88]. The state machine description of the station was converted to a logic description. Additional logic was included to implement the low latency function. The purpose of this implementation is to verify that a dual latency station is feasible to implement while maintaining compatibility with an accepted protocol standard. Furthermore, the order of the latency improvement could be obtained using standard integrated circuit technology.

The implementation is based on the IEEE 802.5 token ring standard rather than the ANSI X3T9.5 standard. The reasons for not implementing the ANSI X3T9.5 standard [ANSI87] are twofold. Firstly, the IEEE 802.5 standard is mature and well documented while the X3T9.5 standard is still evolving. Secondly, testing of the network interface units is easier and cheaper for the IEEE 802.5 standard than for the X3T9.5 standard. The IEEE 802.5 network interface unit can be tested with even telephone twisted
wires while the FDDI has to be tested with optical fibre. Further, the data rate of the IEEE 802.5 standard is only 4 Mbps while that of the X3T9.5 is 100 Mbps.

An IEEE 802.5 network interface unit was designed to operate with a data rate up to 10 Mbps. The logic design was done by my colleague, Mr. Ian Radziejewski, using PCAD which is an IBM based CAD system. The design was simulated using the logic simulation tool in PCAD. For details of the implementation, one should refer to [Radz88]. The QuickChip Facility at Simon Fraser University is used to produce a gate array implementation of the design. Section 3.1 gives a general description of the IEEE 802.5 standard and section 3.2 gives an overview of the dual latency design and the description of the design blocks.

3.1 The IEEE 802.5 Standard

An outline of the IEEE 802.5 token passing ring medium access control method is described for reference; however, full details can be found in [IEEE802.5]. A token ring consists of stations connected serially and information is transferred bit by bit from one station to the next. The station which has the same address as the destination address copies information as it passes through the station. The information, finally, would be removed from the ring by the sending station. The right to
transmit is controlled by a token being passed in the ring. Any station may capture the token by changing the token to a start-of-frame sequence. The maximum amount of time that a station can use the ring before releasing a free token is controlled by a token holding timer set during ring initialization. Eight levels of priority are available and the allocation of priority is by mutual agreement. Error detection and recovery utilize a monitor function which supervises the proper token operation and performs recovery in case of errors. Only one monitor is active at any time on a ring. The selection of the monitor is totally transparent to the users and is determined by the internal addresses of stations.

There are two basic formats used: tokens and frames. The following figures show their field contents, with the left-most bit transmitted first.

**TOKEN FORMAT**

<table>
<thead>
<tr>
<th>SD</th>
<th>AC</th>
<th>ED</th>
</tr>
</thead>
</table>

SD = Starting Delimiter (1 octet)  
AC = Access Control (1 octet)  
ED = Ending Delimiter (1 octet)

Figure 3.1 - Token Format
FRAME FORMAT

<table>
<thead>
<tr>
<th>SD</th>
<th>AC</th>
<th>FC</th>
<th>DA</th>
<th>SA</th>
<th>INFO</th>
<th>FCS</th>
<th>ED</th>
<th>FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD=Starting Delimiter(1 octet)</td>
<td>INFO=Information(0 or more octets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC=Access Control(1 octet)</td>
<td>FSC =Frame-Check-Sequence(4 octets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FC=Frame Control(1 octet)</td>
<td>ED =Ending Delimiter(1 octet)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA=Destination Address(2 or 6 octets)</td>
<td>FS =Frame Status( 1 octet)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA=Source Address(2 or 6 octets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.2 - Frame Format

ACCESS CONTROL FIELD (AC)

<table>
<thead>
<tr>
<th>PPP</th>
<th>T</th>
<th>M</th>
<th>RRR</th>
</tr>
</thead>
</table>

PPP = priority bits  M = monitor bit  
T = token bit  RRR = reservation bits

ENDING DELIMITER (ED)

<table>
<thead>
<tr>
<th>JK1JK1</th>
<th>I</th>
<th>E</th>
</tr>
</thead>
</table>

JK1JK1 = special pattern for ending delimiter recognition
I = intermediate frame bit
E = error-detected bit

FRAME STATUS FIELD (FS)

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
<th>r</th>
<th>A</th>
<th>C</th>
<th>r</th>
<th>r</th>
</tr>
</thead>
</table>

A = address-recognized bits
C = frame-copied bits
r = reserved bits

Figure 3.3 - Access Control field, Ending Delimiter field and Frame Status field

Both token format and frame format have the Starting Delimiter, the Access Control field and the Ending Delimiter.

SECTION 3.1 The IEEE 802.5 Standard 15
The Access Control field has eight bits totally, with 3 bits used for priority, 1 bit for the token, 1 bit for the monitor function and 3 bits for reservation of priority.

The pattern of a free token is shown in the Token Frame while the pattern of a message is shown in the Frame Format. The Token Frame can be converted into the Frame Format by just flipping the Token bit T in the Access Control field.

3.2 The Dual Latency Method

There are several conditions that must be met before a station can enter the low-latency state. Firstly, the station must not be sending a frame; otherwise, the data source is from the interface itself and not from the ring. Secondly, the station which has queued message(s) to send must decode the ring input in order to flip the token bit (T) of a free token in the Access Control (AC) field on time. In the low-latency mode, a station acts simply as a repeater most of the time, i.e. it essentially repeats the data unaltered. Still, the station has to check if the Destination Address (DA) is its address in order to copy data addressed to it. Moreover, the station has to calculate the CRC (cyclic redundancy check) to detect and report any transmission errors. The CRC circuit requires the Frame Check Sequence field for error detection, so the
interface unit has a time period of 7 bits between the FCS and the error bit E in the Ending Delimiter (ED) field to decide if the E bit should be set or not.

The complete network interface unit was implemented using logic elements with timing parameters consistent with the specifications of a commercial 3 micron CMOS gate array process. The entire circuit was simulated using a logic simulator at 10 Mbps. The results show that the full latency is 3.6 bits (360ns), whereas the latency is reduced to 0.6 bits (60ns) in low latency mode. In this case, a factor of 6 reduction in station latency was observed.
IV. ANALYTICAL MODEL

There are a number of papers and books published for the analysis of polling systems and a token ring is one type of polling systems. A polling system consists of a system of multiple queues served by a single server in a cyclic order. The most extensive literature seems to be from Takagi [Taka86]. Takagi has summarized a number of important papers and put the results in one book. His work would, hopefully, unify the notations and terminology used for the analysis of polling systems. Other good work on this topic can also be found in [Klei76], [Schw77], [Schw87], [Hamm86], [Stal87] and Bert[87]. Papers that are of interest can be found in [Wats84], [Ferg85], [Ferg87], [Ever86], [Pang87], [Boxm86], [Boxm87], [Groe87], [Taka85], and [Taka87]. Although the general exact solutions of token rings with ordinary service are not available, there is one special case of token rings that is well analyzed. It is the case of identical stations. Figure 4.1 depicts a polling system model used.

4.1 Assumptions

The assumptions are listed as follows:

* Poisson Arrivals
* Infinite Buffer Size
FCFS (First Come First Serve) for each message in a queue
* Multiple-token operation
* Stable System (see section 4.3)

4.2 Notations

There are \( N \) queues in the system and they are indexed by \( i \) in cyclic order \((i = 1, 2, \ldots, N)\). A Poisson arrival of messages is assumed at a rate \( \lambda_i \) for queue \( i \). The first and second moments of service time at queue \( i \) are \( b_i \) and \( b_i^{(2)} \) respectively. The time it takes to transmit a bit from one station to the next is called switchover time (sometimes called walk time or reply interval). The switchover time is equal to the station latency plus the propagation delay. The mean and variance of the switchover time for queue \( i \) is denoted by \( r_i \) and \( \sigma_i^2 \) respectively. The dual latency method tries to minimize the station latency while the propagation delay is fixed. We define the total utilization \( \rho_0 \) and the total ring latency \( \bar{R} \) as follows:

\[
\rho_0 = \sum_{i=1}^{N} \rho_i = \sum_{i=1}^{N} \lambda_i b_i
\]  

(4.1)
Figure 4.1 - A Model of Polling Systems.
4.3 Stability Requirements

The system is stable in the sense that the mean queue length at every queue is finite, which implies that the mean polling cycle time is also finite. The necessary stability condition is:

* Exhaustive & gated: \( \rho_0 < 1 \) (4.3)

* Ordinary (limited-to-one):
  
  For all \( i \), \( \rho_0 + \lambda_i R < 1 \) (4.4)

In the exhaustive or gated service system, if there is one heavily loaded station, the whole system will be overloaded. In the ordinary or limited-to-one system, however, one heavily loaded station will affect the mean waiting time of that station dramatically rather than that of the whole system. This localization effect provides fairness among stations.

4.4 Analysis of a Completely Symmetric Ring

The completely symmetric case is the case that every station is identical in terms of its arrival rate, service time distribution, and switchover time distribution. Therefore the subscript \( i \) is omitted in the expressions. Derivations of the expressions shown below can be found in either [Taka86] or
[Taka85]. There are three service disciplines that will be considered: limited-to-one (ordinary) service, gated service and exhaustive service. The mean waiting time for limited-to-one service is given by:

\[
E[w]_{limited-to-one} = \frac{\delta^2}{2r} + \frac{N[\lambda b^{(2)} + r(1 + \lambda b) + \lambda \delta^2]}{2[1 - N\lambda(r + b)]}
\]  

(4.5)

where \( E[w] \) is the mean waiting time of messages queued for transmission at a station, \( r \) and \( \delta^2 \) are the mean and variance of the reply interval (the sum of station latency and interstation propagation delay), \( N \) is the number of stations connected to the network, and \( b \) and \( b^{(2)} \) are the first and second moments of the message service time respectively.

The corresponding result for gated and exhaustive service is:

\[
E[w]_{gated(+), exhaustive(-)} = \frac{\delta^2}{2r} + \frac{N[\lambda b^{(2)} + r (1 \pm \lambda b)]}{2(1 - N\lambda b)}
\]

(4.6)

where the '+' term takes the value '-' for gated service and '-' for exhaustive service.
Given equal reply interval and service time statistics, the following inequality holds for a network of \( N \) stations, operating under the three types of service disciplines:

\[
E[w]_{\text{limited}} \geq E[w]_{\text{gated}} \geq E[w]_{\text{exhaustive}}
\]

(4.7)

4.5 Analysis of a General Ring

For a general ring, numerical algorithms developed by Ferguson and Aminetzah [Ferg85] and Watson [Wats84] are available to enable the calculation of mean waiting times in a gated or exhaustive token ring system. In an ordinary service system, the exact expression to calculate the mean waiting time is very difficult and is not yet known at this moment. Only the two-queue case for a general ordinary service ring system case has been solved explicitly [Boxm87], requiring the solution of a Riemann-Hilbert boundary value problem of mathematical physics [Groe87]. Approximation techniques [Groe87] [Boxm86] [Ever86] are available, but they are not incorporated because their accuracy is not guaranteed, especially for a comparison between full latency and dual latency system.

For gated and exhaustive service system, Watson's algorithm [Wats84] can be expressed as follows:
\[ E[w_j]_{\text{exh}} = \frac{1}{2\rho_j^2(1-\rho_j)} \sum_{i=1}^{N} \lambda_i b_i^{(2)} \Delta_i^{(j)} + \frac{\lambda_i b_i^{(2)}}{2(1-\rho_j)} + \frac{\mathcal{R}(1-\rho_j)}{2(1-\rho)} \]

\[ + \frac{1-\rho}{2\mathcal{R}\rho_j^2(1-\rho_j)} \sum_{i=1}^{N} \sigma_i^2 \Delta_i^{(j)} + \frac{\sigma_{j-1}^2(1-\rho)}{2\mathcal{R}(1-\rho_j)} \] (4.8)

\[ E[w_j]_{\text{gated}} = \frac{1+\rho_j}{2\rho_j^2} \sum_{i=1}^{N} \lambda_i b_i^{(j)} \Delta_i^{(j)} + \frac{\mathcal{R}(1+\rho_j)}{2(1-\rho)} \]

\[ + \frac{(1-\rho)(1+\rho_j)}{2\mathcal{R}\rho_j^2} \sum_{i=1}^{N} \sigma_i^2 \Delta_i^{(j)} \] (4.9)

where

\[ \rho = \sum_{j=1}^{N} \rho_j = \sum_{j=1}^{N} \lambda_j b_j \]

\[ \mathcal{R} = \sum_{j=1}^{N} r_j \]

\[ \sigma_0^2 = \sigma_N^2 \]

\[ \Delta_i^{(j)} = \sum_{n=0}^{\infty} \gamma_{i,i-j+1,n}^2, \quad i, j = 1, \ldots, N \]

and where the \( \gamma_{j,k,n} \) are defined by

\[ \gamma_{j,k,n} |_{\text{exh}} = (1-\rho_{k+j-1})^{-1} \left( \sum_{m=1}^{k-1} \rho_{m+j-1} \gamma_{j,m,n-1} \right) \]

\[ + \sum_{m=k+1}^{N} \rho_{m+j-1} \gamma_{j,m,n} \] (4.10)

SECTION 4.5 Analysis of a General Ring
\[ \gamma_{j,k,n} |_{\text{gated}} = \sum_{m=1}^{k} \rho_{m-j-1} \gamma_{j,m,n-1} + \sum_{m=k+1}^{N} \rho_{m-j-1} \gamma_{j,m,n} \] (4.11)

for

\[ k = 1, \ldots, N \]
\[ \gamma_{j,1,-1} = 1 \]
\[ \gamma_{j,m,-1} = 0 \text{ for } m = 2, \ldots, N \]
\[ \rho_k = \rho_{k-N} \text{ for } k > N \]
\[ \gamma_{j,k,n} = \gamma_{j,N+k,n} \text{ for } k \leq 0; \ n = 0, 1, 2, \ldots \]

4.6 Analysis of a Dual Latency Ring

The evaluation of polling system performance under conditions of dependent statistical distributions is difficult in general. Ferguson [Ferg86] has developed bounds and approximations to the mean waiting time for a token ring with station dependent overheads and exhaustive service. For the case of limited-to-one service, no such approximate solutions are yet available.

However, in the case of a polling system having a reply interval with two values, dependent on the state of the
message queue, a relatively straightforward approach is possible. We could approximate the solution by first assigning each station a fixed reply interval, equal to the sum of the low station latency value and the interstation propagation delay. We then add the additional latency, the difference between the full and low latency values, to the message service time. This approach is valid because each station operates with full latency only when it has a message to transmit. This approach is based on the assumption that the latency at a station with a waiting message can be expressed as the sum of two independent random variables, one with the same distribution as the latency for a station with no waiting message and one representing the additional delay. For this case, we apply Takagi's result [Taka86] by adding the additional latency component to the message service time and by setting the switchover time equal to the low latency. That is,

\[ b_{\text{Dual latency}} = b_{\text{Full latency}} + \Delta r \]
\[ b_{\text{Dual latency}}^{(2)} = b_{\text{Low latency}}^{(2)} + 2b\Delta r + (\Delta r)^2 \]
\[ r_{\text{Dual latency}} = r_{\text{Low latency}} \]

where \( \Delta r = r_{\text{Full latency}} - r_{\text{Low latency}} \)

(4.12)
In order to investigate the advantage of the dual ring latency further, I make use of the equation 4.5 from Takagi [Taka85] for the mean waiting time of a completely symmetric ring with ordinary service and constant switchover time. The equation 4.12 is substituted into this equation and yield:

\[
E[w]_{\text{Dual, limited-to-one}} = \frac{N[\lambda b^{(2)}_{\text{Dual latency}} + r_{\text{Dual latency}}(1 + \lambda b_{\text{Dual latency}})]}{2[1 - N\lambda(r_{\text{Dual latency}} + b_{\text{Dual latency}})]} = \frac{N[\lambda b^{(2)}_{\text{Dual latency}} + r(1 + \rho)] - d \rho(1 - \rho - \lambda r)}{2[1 - N(\rho + \lambda r)]} \tag{4.13}
\]

The equation above gives the mean waiting time as a function of station utilization, \( \rho \), first and second moment of the message length, \( b \) and \( b^{(2)}_{\text{Dual latency}} \), switchover time, \( r \) and number of station \( N \). It assumes limited-to-one service and a completely symmetric ring.

The ratio then becomes:

\[
\frac{E[W]_{\text{full}}}{E[W]_{\text{Dual}}} = \frac{\lambda b^{(2)}_{\text{Dual latency}} + r(1 + \rho)}{\lambda b^{(2)}_{\text{Dual latency}} + r(1 + \rho) - \Delta r(1 - \rho - \lambda r)} \tag{4.14}
\]

From the equations above, there are observations that can be made. Firstly the ratio is always greater than or equal to one. Secondly, the ratio increase as the \( \Delta r \) increases. From
the problem definition, \( \Delta r < r \), that leads to an upper limit on the result of the ratio (i.e., \( \Delta r = 0 \)). Thirdly, the ratio decreases as the variance of the message length increases. That makes constant message length more attractive to the dual latency method.
V. SIMULATION MODEL

In this section, the simulation aspects of the performance modeling are described. Computer simulation is a very powerful tool and can be arbitrarily detailed. The simulation software developed can be used in general cases.

5.1 Why Simulate?

There are two general reasons for using a computer simulation model. Firstly, an exact mathematical analysis for a token ring with limited-to-one service is not yet possible. Boxma and Groenendijk [Boxm87] recently explicitly solved a two-queue polling system with limited-to-1 service discipline, which already involved the solution of a very difficult problem in mathematical physics. For a polling system with more than two stations, the solution would be much more difficult. Therefore, approximation techniques [Groe87] [Boxm86] [Ever86] are available for limited-to-one service discipline. Since the accuracy of the approximation techniques is not guaranteed, it may be inappropriate to compare results with different parameters, as in the case of full latency and dual latency, when the error of one case may be a lot higher or lower than that of the other case.
Mathematical analysis sometimes has assumptions that are not valid for real world situations. For instance, assumptions of multiple-token operation, symmetric traffic, Poisson arrivals, etc. may not be good approximations to real world situations. In fact, the IEEE 802.5 standard uses single-token operation rather than multiple-token operation. Small changes in the operation may have dramatic impact on a mathematical model. But a simulation program can be modified relatively easily. Most mathematical models only deal with steady state situations. Simulation models, however, can be used to examine the transient behavior as well as steady state behavior. Good texts on computer simulation are [Shan75], [Gord78], [Saue81], [Lave83] and [Mac85].

5.2 Drawbacks of Using Simulation

There are disadvantages of using computer simulation modeling. Firstly, it potentially contains programing errors, like any software program. It is very difficult to prove the correctness of any reasonably sized program. Time and cost for simulation software development may be high and computational expenses of simulation are usually more than that of mathematical analysis.
5.3 Language

Two discrete event simulation languages were considered for use: GPSS (General Purpose Simulation System) and SIMSCRIPT. GPSS was chosen because of its compact structure that allows minimal changes in source code when simulating different system parameters. The simulation program was originally developed in GPSS/PC on an IBM PC. The GPSS/PC compiler was very user friendly and had a lot of visual aid for debugging programs. But later I found that it was too slow to run the simulation program, so I switched to GPSS/H on the MTS operating system. The program running on MTS is a few hundred times faster than the PC version. Furthermore, the GPSS/H compiler has additional facilities and block statements to reduce the source code required and is upward compatible with GPSS V. Descriptions of GPSS language can be found in [Schr74] and [Gord75].

A pre-processor was developed in PASCAL to allow convenient entry of different system parameters. The pre-processor automatically generates source codes for the GPSS/H compiler. The system parameters include service discipline (ordinary, gated or exhaustive), token transmission operation (multiple-token, single-token or single-packet), latency types (full latency or dual latency), service time distribution (exponen-
tial, constant or mixed of both), arrival distribution (exponential or constant), and running method (regenerative method or independent replications of runs, refer to section 5.5.2).

5.4 Models

This section will describe a few important model segments used in the simulation models. Since GPSS was used to implement the simulation, a basic understanding of the language is required. The block diagrams of the model segments are listed in Figure 5.1 to 5.8. The standard GPSS block notions and the concise meaning of each block can be found in [Gord75] and [Schr74]. The models were implemented in the GPSS/H compiler by Wolverine Software Corporation (refer to [GPSS/H]). The GPSS/H compiler is faster and easier to use than the GPSS/360 and the GPSS/V compilers by IBM. For the models I ran, the GPSS/H compiler was about 6 to 7 times faster than the GPSS/360. Further, the GPSS/H compiler provides convenient features such as free input formats and interactive debugging.

The models described are limited-to-one, gated, and exhaustive service disciplines, multiple-token, single-token and single-packet operations, the regenerative method and the independent replications of runs, and the avalanche situation
simulation and the turbo mode. For each type of token operations, three service disciplines are possible; the avalanche simulation or turbo mode can be used on top of the models. For illustration purpose, the figures show the models with regenerative method; the models with independent replications of runs will be very similar. The main difference is that for independent replications of runs, the data is collected in a number of runs; the number is determined by the user. Thus, the user has to look at the output and calculate the data manually. As will be shown in section 5.5.1, the calculation is fairly simple.

The models used for different service disciplines will be explained first, followed by the models used for different token operations and finally the avalanche simulation model and the turbo model. Although the block diagrams show only the most important model segments, it is enough to understand the basic operations without too much emphasis on the programming details.

Figure 5.1 to Figure 5.3 depict the models used for the limited-to-one, gated and exhaustive service disciplines respectively. Each station is represented by a model segment. The model segment shown is just for one station and the dashed rectangle represents the portion for each station. A circle in
the block diagrams represents a block location. The first block statement for each station is a GENERATE block to generate its own traffic based on some distributions. The software supports two interarrival distributions: exponential and constant. It is not difficult to enter other distributions if necessary. After a message is generated, it will go to a QUEUE block to start to gather waiting timing statistics. The message (or transaction) is then put into the user chain of that station. That makes the message scan-inactive and saves computing time dramatically. There is a token circulating around the ring in a cyclic order. When the token finds a message waiting in a station, it will release one message (or transaction) from the user chain of that station. The token is then destroyed by the TERMINATE block. The message released from the user chain then seizes the facility (or single server) named as "TOKEN". The message will depart the queue and record the statistics. The server will be used for a certain amount of time depending on the random number generator and the service time distribution.

The software supports three different service distributions: exponential, constant and a combination of exponential and constant distributions. Other distributions can be entered easily. Finally the server is released in the RELEASE block. For a dual latency station, the time difference between the
full latency and the dual latency has to be added as a delay. For the regenerative method, the model tests if the whole system is empty. If it is, the statistics will be recorded as a regenerative cycle before the system will be reset. For the independent replications of runs, the statistics collected for the first M messages specified by the user are discarded and then the data afterward will be used as useful data.

In a gated service system, at the instant the token detected the station has message(s) to send, it will store the number of messages in the queue to a variable named "USERQ". Therefore, every time the server serves a message, the variable "USERQ" is decreased by one. The server will continue to serve until the "USERQ" becomes zero. In an exhaustive service system, the server will continue to serve until the queue of the station becomes empty. Thus, one would expect a longer running time for gated service compared with exhaustive service, assuming everything else the same, because there are more block statements involved in running a program for gated service.
Figure 5.1 - Model Segment for Limited-to-one Service Discipline with Multiple-Token Operation and Regenerative Method.
Figure 5.2 - Model Segment for Gated Service Discipline with Multiple-Token Operation and Regenerative Method.
MULTIPLE-TOKEN, EXHAUSTIVE SERVICE, REGENERATIVE METHOD.

Figure 5.3 - Model Segment for Exhaustive Service Discipline with Multiple-Token Operation and Regenerative Method.
The multiple-token operation with different service disciplines is just described. For the cases of the single-token and the single-packet operations, the situations are more complicated and require more computing effort. Figures 5.4a, 5.4b and 5.4c show the block diagram of the single-token operation and Figures 5.5a, 5.5b and 5.5c show the block diagram of the single-packet operation. For the purpose of illustration, ordinary service discipline and the regenerative method are assumed. Using the full latency method, the ring latency is known. (The ring latency is equal to the sum of all station latencies plus propagation delays.) But the ring latency is not known beforehand if the dual latency method is used. Therefore, using the dual latency method for the single-token and the single-packet requires special attention.

For the single-token operation, if the full latency method is used, the operations of the blocks become very similar to the multiple-token. The only difference is that the service time for the last message will be compared with the sum of the ring latency and the header time. If the service time is larger, the message will proceed as usual; otherwise, the time difference between the two has to added as additional delay induced. For ordinary service, the last message served per token is also the only message served per token.
If the dual latency method is used, the station will generate messages as normal. But after a message seizes the token and goes through a DEPART block, another transaction has to be generated for the dual latency method and it is called the created transaction. The created transaction is generated by the SPLIT block. The created transaction is necessary in order to check the time it takes for the header to go back to the same station. Since for the dual latency method the ring latency is not known beforehand, the created transaction is to be circulated around the ring. The ring latency experienced by the created transaction depends on the queues of other stations. After some time, the created transaction goes back to its sending station and checks if the logic switch "RING1" is set. If the logic switch "RING1" is not set, it will be held in the GATE block waiting for the logic switch to be set by the original transaction. It the logic switch is set, the created transaction will proceed to set the logic switch "RING2" before finally being destroyed.

The original transaction will go through the normal path to the ADVANCE block and then the RELEASE block. In order to keep the memory for the system manageable by the GPSS/H compiler, two logic switches "RING" and "RING2" are used to control the flow of the two transactions. The original transaction will proceed as normal, but after the RELEASE block, it will set
the logic switch "RING1" and then test if the created transaction has returned to its own station. If the created transaction has returned, the message transaction will proceed; otherwise, it will wait for the created transaction to return to the LOGIC block to set the logic switch "RING2". The setting of the logic switch "RING2" is controlled only by the created transaction and the setting of the logic switch "RING1" is controlled only by the original transaction. The reason for using the logic switches is to avoid using the TEST block alone, which produces a large amount of computing overhead.

The tricky part is when the station uses service disciplines other than ordinary service. For instance, if the station uses exhaustive service, it does not know how many messages that will be served in advance. That is the main reason for using the created transaction. Only the original transaction of the last message for a token will open the logic switch "RING1" and allows it to check if all the created transactions have returned to the sending station. The operations described above assume that a station can switch from low latency to high latency between messages in the cases of exhaustive or gated service disciplines. In terms of implementation, when a station has a message arrived and waits for a free token, it will not be able to switch to high latency if it is repeating
message bits sending by other stations. The station will have to wait until it finds the ending delimiter of the message and then switch to high latency.

For the single-packet operation, if the full latency is used, the operations of the blocks become very similar to the multiple-token operation. The main difference is that the service time of the last message will be increased by the amount equal to the ring latency. The ring latency is a constant and can be calculated in the pre-processor.

The block diagram using the dual latency method is shown in Figures 5.6a, 5.6b and 5.6c. The message transaction will proceed as in the case of the single-token operation until it reaches the RELEASE block. For the single-packet, the entire message has to be received by the sending station rather than only the header. Therefore, I make use of the transit time to calculate the ring latency of the last message (as for exhaustive service). After the last message is served, we save the absolute clock using the MARK block. The transit time for each created transaction is stored in the variable named "CHECK", so that the variable "CHECK" will finally store the transit time of the last created message after all created transactions are destroyed. In the mean time, the last message may be waiting for the last created transaction to finish.
Therefore, the additional delay is equal to the transit time of the last created transaction minus the time the last message waiting for all created transactions to finish.
SINGLE-TOKEN, LIMITED-TO-ONE SERVICE, REGENERATIVE METHOD.

Figure 5.4a - Model Segment for Single-Token operation with Limited-to-one Service Discipline and Regenerative Method.
Figure 5.4b - Model Segment for Single-Token operation with Limited-to-one Service Discipline and Regenerative Method.
Figure 5.4c - Model Segment for Single-Token operation with Limited-to-one Service Discipline and Regenerative Method.
SINGLE-PACKET, LIMITED-TO-ONE SERVICE, REGENERATIVE METHOD.

Figure 5.5a - Model Segment for Single-Packet operation with Limited-to-one Service Discipline and Regenerative Method.
Figure 5.5b - Model Segment for Single-Packet operation with Limited-to-one Service Discipline and Regenerative Method.
Figure 5.5c - Model Segment for Single-Packet operation with Limited-to-one Service Discipline and Regenerative Method.
The model segment used for the avalanche simulation is shown in Figure 5.6a and 5.6b. Figures 5.6a and 5.6b present the model segment for each station while Figure 5.7 presents the timer model used in the avalanche simulation. The avalanche model tries to simulate the real-time behavior when the ring system experiences many messages arrived at a short period. Although the avalanche arrivals may be large, the normal arrivals may be fairly small. Hence, there are two things to investigate. One is that when the system finishes serving all the avalanche messages and another one is that when the system returns to its normal operating state. There are two measures proposed: relaxation time and empty time. Relaxation time here is defined as the time it takes the system to finish all the avalanche messages. Empty time is defined as the time it takes the system to empty queues of all stations. Obviously, the empty time is larger than or equal to the relaxation time.

There are two model segments shown to illustrate the operations of the avalanche model. The first one is the model segment used for each station and is shown in Figures 5.6a and 5.6b. The second one is the model for the timer to open or close the gate which allows or disallows the avalanche messages to enter a station. Further, it will record the relaxation time and the empty time. In order to distinguish between avalanche messages and normal messages, a group entity
is used. All avalanche messages belong to the group named "AVALAN". When the logic switch "LOCK" is reset, it allows the avalanche messages to go to the station. When the logic switch is set, all avalanche messages generated are destroyed. A messages proceeds as usual until it seizes the token. Then the message is checked if it belongs to the group "AVALAN". If it is, then collect statistics on the tables; otherwise, skip the statistics collection part. In order to determine when the last avalanche message finishes being served, the relaxation time for every avalanche message that finishes transmission is stored in the variable "KEEP". Therefore, when the system is empty, the variable "KEEP" will store the relaxation time for all the avalanche messages that have finished their transmission.

The timer model starts with storing the time that the avalanche period ends in the variable "END". The logic switch is reset and after the avalanche period is past, the logic switch will be set. That makes all the avalanche messages generated after setting be destroyed immediately without going to the queues of the stations. This model segment also collects the statistics for the relaxation time and the empty time when the system becomes empty.
AVAVANCHE
SIMULATION,
MULTIPLE-TOKEN,
LIMITED-TO-ONE
SERVICE,
REGENERATIVE
METHOD.

Figure 5.6a - Model Segment for the Avalanche Simulation with Multiple-token Operation, Limited-to-one Service Discipline and Regenerative Method.
Figure 5.6b - Model Segment for the Avalanche Simulation with Multiple-token Operation, Limited-to-one Service Discipline and Regenerative Method.
TIMER FOR AVALANCHE SIMULATION

SAVEVALUE
END

LOGIC
RESET
LOCK

ADVANCE
AVALANCHE PERIOD
SET
LOCK

LOGIC
IF NOT

E
IF NOT

TEST
IF NOT

E
IF NOT

TABULATE
RECORD RELAXATION TIME
DONE

TABULATE
RECORD EMPTY TIME
EMPTY

TRANSFER
UNCONDITIONAL

Figure 5.7 - Model Segment for the Timer Model of the Avalanche Simulation.
The model segment for the turbo mode is shown in Figure 5.8 and is important to saving computing time. Normally, without the turbo mode, the computing cost increases when the system utilization goes down. It is a common problem for discrete-event simulation because every time the simulated time moves forward, it is considered by the computer as an event and there is some computing overhead for every event. Hence, for the token ring LAN, the transfer of a free token increases the simulated time and thus is regarded as one event. But that is not useful to the data we are interested in. As the channel utilization becomes smaller, a larger portion of computing time is wasted in the transfer of the free token around the ring. In order to alleviate the problem, when the system becomes empty, the token is held in the TEST block. The scanning algorithm of GPSS then schedules the next event which will be the earliest message arrival. After a message arrival, the free token is put back to the ring randomly allowing messages a chance to capture it. It should be noted that the message first arrived does not guarantee the capture of the token, since another message may arrive later but could seize the token on the way the token goes to the station of the first message. There is a BUFFER block has to added to the model segment for each station. The BUFFER block should be put right after the QUEUE block and is used to explicitly force a restart of the current events chain scan.
Otherwise, after a message is put into the user chain (opposed to the current event chain) the compiler will misinterpret that there is no more active transaction and quit running.
Figure 5.8 - Model Segment for the Turbo mode.
5.5 Statistical Considerations

Since computer simulation is random in nature, the results may be very different by using different sets of random number generator seeds. In order to overcome this problem, statistical analysis is frequently used in simulation. In situations where a sequence has a limiting distribution, i.e. the sequence goes through a transient phase which depends on the initial conditions, and has an essentially unchanging distribution afterward, the sequence has what we call a steady state behavior. The sequence could be the mean waiting time of a system, the total utilization of a system, etc. In contrast, characteristics of a simulation sequence depend on the initial conditions and on the point at which they occur either in a sequence or in time. The sequence has a transient state behavior. It could be the mean waiting time of the 5th customer, the mean waiting time of the 10th customer, the mean absolute time that the 5th customer finished his service, etc.

In this section, I will concentrate on the statistical considerations for steady state behavior rather than for transient behavior. Good material on this subject can be found in [Klej74], [Fish79] and [Lave83]. There are common problems associated with simulation of steady state behavior. The first common problem for steady state simulation is that the
duration of a transient phase is usually unknown. If the transient phase duration is overestimated, then computing time is wasted. If the duration is underestimated, the data collected is biased because of the transient data included. A general technique to determine the duration of a transient is not known. In practice, an experimenter could estimate the duration by looking at the plot of the sequence and subjectively determine that a certain period is good enough to terminate a transient phase.

After the transient duration is estimated, there remains the problem of analyzing the output data. The difficulty lies in the fact that the output sequence is not independent. The two approaches described below both produce independent blocks of data so classical statistics can be applied.

5.5.1 Independent Replications of Runs

The most common and easiest to understand approach is to run the simulation using the method of independent replications. In order to make each replication independent of each other, we can change the seeds of random number generators for each replication, discard the data from the transient phase and collect the data afterward for each replication.
Suppose we want to estimate the true mean $\mu$ and the confidence intervals of the estimated mean $\hat{\mu}$. Let $M$ be the number of replications, $n$ be the sample size of each run, $y_{ij}$ be the $i$th observation in the $j$th run, and the sample mean of the $j$th run be $Y_j$. Therefore, for each run:

$$Y_j = \frac{1}{n} \sum_{i=1}^{n} y_{ij} \quad \text{(5.1)}$$

Since $Y_{1}, Y_{2}, \ldots, Y_{M}$ are independent, an approximately unbiased estimate of the mean is

$$\hat{\mu} = \frac{1}{M} \sum_{j=1}^{M} Y_j \quad \text{(5.2)}$$

Hence, traditional statistical analysis can be used to calculate the confidence intervals. The $(1-\alpha)$ confidence interval may be obtained by

$$\hat{\mu} \pm \frac{t_{\frac{\alpha}{2}} s_{Y}}{\sqrt{M}} \quad \text{(5.3)}$$

where
\[ s^2 = \frac{1}{M-1} \sum_{i=1}^{M} (Y_i - \hat{\mu})^2 \]  (5.4)

and

\[ t^{a/2}_{M-1} \]

is the upper \(a/2\) percentile of the Student's \(t\)-statistics with \(M-1\) degree of freedom.

If the degree of freedom is over 30, then the standard normal statistics can be used. In general, \(M\) should be kept small and \(N\) large to minimize unused data during the transient state and any residual bias caused by the slow convergence of \(\mu\).

5.5.2 The Regenerative Method

The regenerative method is based on the idea of independent cycles in a single run. Crane and Iglehart did a lot of original research on this topic [Cran74a], [Cran74b], [Cran75a] and [Cran75b] and a more application oriented text can be found in [Cran77]. The advantages of using the regenerative method are listed as follows:

1. It does not need to know the duration of a transient phase.
2. It collects useful data right at the beginning. We don't need to discard any useful data in the transient phase.

3. It produces independent sub-sequences of data in a single run.

Limitations

The regenerative method applies to discrete-event simulations that can be modeled as regenerative processes. The basic regenerative assumption is that there exists an regeneration time, \( T_1 \), such that the continuation of the process beyond \( T_1 \) is a probabilistic replica of the whole process starting at time 0. This property implies that \( T_2, T_3, \ldots \) exist and they have the same property as \( T_1 \). Further, finite regeneration times are assumed.

The typical situation which satisfies the regenerative requirement is when the system hits \( T_1 \), the system can proceed without any knowledge of its past history.

The procedures of obtaining an approximate \( 100(1 - \alpha) \) confidence interval for the parameter \( r \) are the following:

1. Run the simulation for \( n \) regenerative cycles.
2. For each regenerative cycle, compute the sum of the observed data in the jth cycle $Y_j$ and the number of observations made in the jth cycle $V_j$.

3. After the simulation run, compute the following sample statistics:

$$f = \frac{\bar{Y}}{\bar{V}}$$

where

$$\bar{Y} = \frac{1}{n} \sum_{j=1}^{n} Y_j, \quad \bar{V} = \frac{1}{n} \sum_{j=1}^{n} V_j$$

$$s^2 = s_{11} - 2f s_{12} + f^2 s_{22}$$

where

$$s_{11} = \frac{1}{n-1} \sum_{j=1}^{n} Y_j^2 - \frac{1}{n(n-1)} \left( \sum_{j=1}^{n} Y_j \right)^2$$

$$s_{22} = \frac{1}{n-1} \sum_{j=1}^{n} V_j^2 - \frac{1}{n(n-1)} \left( \sum_{j=1}^{n} V_j \right)^2$$

$$s_{12} = \frac{1}{n-1} \sum_{j=1}^{n} Y_j V_j - \frac{1}{n(n-1)} \left( \sum_{j=1}^{n} Y_j \right) \left( \sum_{j=1}^{n} V_j \right)$$

4. Finally, the confidence interval can be computed.
\[ P \pm \frac{z^*_a s}{\sqrt{V/n}} \]

where

\[ z^*_a = \phi^{-1} \left( 1 - \frac{a}{2} \right) \]

\[ \phi \] is the standard normal distribution function.

5.6 Validation

The validity of the simulation models was first inspected by using the interactive debugging facilities in GPSS/H. The program was monitored to determine if it behaved as expected. Afterward, the validation was then checked by comparing known results with the simulation results. The two examples of Ferguson's paper [Ferg85] were chosen for comparison and the simulation results with 90% confidence intervals covered most of the data points of the known theoretical results.

When the simulations for unknown theoretical results were run, there were three things that had to be checked. Firstly, I would observe if the results were too reasonable. If they were, the systems parameters would be inspected again very carefully. Secondly, when the utilization went down to 0.1, the mean message waiting time should be close to but not
smaller than half of the ring latency for Poisson arrivals. Thirdly, the mean message waiting time of the full latency simulations should be more than that of the dual latency simulations for same utilization. Finally, when the utilization went up close to the stability limit, the simulation results should experience a sharp increase in mean message waiting time. The simulation should be stopped because the memory requirements and the computing cost would rise dramatically. The simulation might become too expensive to run.

5.7 Features of the Simulation Models

The simulation model consists of a pre-processor, a processor and a post-processor. The pre-processor was written in PASCAL and it automatically generates the source code for the GPSS/H processor after the system parameters are entered. If the regenerative method is specified, a FORTRAN post-processor is available for calculation and output of confidence intervals for the mean message waiting time and other statistical data such as the number of regenerative cycles, number of messages, etc.

The pre-processor is the heart of the simulation software. It allows the user to enter his parameters for his ring system.
relatively easily. Also, it checks if the input parameters are unrealistic. If they are, it will output a warning message, but a GPSS/H program will still be generated. This will prevent the user from wasting computing money on wrong entries and at the same time give the user the freedom to run such a system. The post-processor is a relatively straightforward program for statistical analysis of the regenerative method. The reason for using FORTRAN is that it can be directly linked with GPSS/H.

It should be noted that the simulation model can be used for either a token ring or token bus network. Of course, the input parameters will be different. For a token ring, the switchover time is just the propagation time from one station to the next plus the station latency. For token bus, the switchover time would become the time to transmit an explicit token frame, which is 152 bits in the IEEE 802.4 Token Bus standard [IEEE802.4] plus the propagation delay between the two stations, which may be physically quite far apart. Finally the response time of the station has to be added.

As the utilization becomes lower, the computing cost rises considerably. The reason is that for discrete-event simulation the computing cost is directly proportional to the number of events initiated within the simulated time period but not
simulated time itself. Therefore, in low utilization the transfer of a free token dominates the transfer of messages. Consequently, each transfer of a free token is an event to the computer but there is no statistically useful data associated with such an event. In order to remedy this problem, a turbo mode is available for running low utilization. In turbo mode, after each message is served, the free token released is suspended until the next message arrives. Then the free token is randomly put back into the ring. The time a free token circulated around the ring would be very minimal. The problem with this approach is that for some cases the assumption would be invalid. For instance, if stations have constant arrivals, the capture of a token may depend on the ordering of stations rather than random placement of a token.

There is another type of simulation option available. It is called avalanche simulation and is used for cases when a number of messages arrive only in an avalanche fashion, i.e., they arrive in a period of time and then disappear and arrive ... The application of this kind of avalanche simulation can be thought of as the simulation of a process control application, such as a nuclear power plant. When the alarm is on and a number of sensors send out messages roughly around the same time, the system is under an overloaded situation. This situation occurs only once in a while but not all the
time. The software has already been debugged, but it still needs some modifications. The difficulty in the avalanche simulation is that the confidence intervals of the parameters of interest are too big for a reasonable computing cost.

In summary, the input parameters are listed as follows:

- Service Disciplines: exhaustive, gated, or limited-to-one.

- Token transmission types:
  - multiple-token (e.g. FDDI, IEEE 802.4),
  - single-token (e.g. IEEE 802.5) or single-packet.

- Latency Types: Full latency or dual latency.

- Interarrival Time: exponential or constant or a given distribution function.

- Service Time: exponential, constant, or a mix of both.

- Switchover Time: constant for both latencies.

- Number of Stations: 2 - 100.

- Running Methods: the Regenerative Method or Replicated Runs.

- Turbo mode available for running low utilization.

- Avalanche simulation option
VI. NUMERICAL AND SIMULATION RESULTS

This section presents the numerical and simulation examples in order to discuss some general characteristics of the dual latency method compared with the full latency. Since there can be so many combinations of the parameters, it was difficult to choose a so-called "typical" LAN to investigate. Thus, I used the values for the low latency and the full latency based on the implementation [Radz88]. Section 6.1 describes the LANs running at low speed, such as the IEEE 802.5 standard. The values used for the latencies were based on the IEEE 802.5 implementation that runs at 10 Mbps data rate. The station low latency is 0.6 bits and the station full latency is 3.6 bits. Section 6.2 describes the LANs running at high speeds, such as the FDDI standard. The values used for the latencies were based on the AMD implementation [AMD86] of the FDDI standard that runs at 100 Mbps. The station low latency is 62 bits and the station full latency is 24 bits [Hard88b].

Each subsection is further divided into completely symmetric cases and asymmetric cases. For the asymmetric cases, it is assumed that the LAN has 20 stations, 2 of them heavily load and 18 of them lightly loaded. This asymmetric loading was studied in [Bux83a], [Bux84], [Ferg85], and [Ever86] and was chosen to represent a class of asymmetric traffic. Each of the two heavily
loaded stations contributes 40% of the total utilization while each of the eighteen lightly loaded stations shares the remaining traffic utilization evenly.

6.1 Low Speed LANs

Highlights of the implementation for the dual latency method is presented in section 3. The station latencies were estimated based on the IEEE 802.5 implementation [Radz88] operating at 10 Mbps. The values for the station low latency and the station full latency are 0.6 and 3.6 bits respectively. The propagation delay is assumed to be 5 \( \mu s/km \) cable length. Further, the propagation delay per station is assumed to be 0.9 bits which is equivalent to about 18 m cable length. The total cable length of the ring is about 360 m. Therefore, the final low latency (the propagation delay per station + the station low latency) is equal to 1.5 bits and the final full latency (the propagation delay per station + the station full latency) is equal to 4.5 bits. Thus, the difference between the final full latency and the final low latency is equal to 3 bits.
6.1.1 Completely Symmetric Cases

The parameters used to run the completely symmetric ring are listed below. The cases 1A and 1B were chosen to demonstrate the effect of the number of stations and the effect of using either constant or exponential message distribution. The 176 bits mean message length was picked because as will be seen in section 6.1.2, the mean message length plays an important role in determining how attractive the dual latency method is. The mean message length here is 176 bits as in Case 2C in section 6.1.2, which is $1/8k$ (exp.) bits + 48 bits overhead rather than either completely constant or completely exponential. The number is used because it is large enough to show the significance of the dual latency method but it is not too small to exaggerate the results. For real-time applications, the messages are typically only a few bytes in length [Hutch87]. The message length chosen is well suitable for these kinds of applications.
Completely Symmetric Case (Case 1)

<table>
<thead>
<tr>
<th>Number of stations</th>
<th>20, 100, 200, or 1000.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant switchover time (in bits)</td>
<td>4.5 for full latency. 1.5 for low latency.</td>
</tr>
<tr>
<td>Mean message length (in bits)</td>
<td>Case</td>
</tr>
<tr>
<td>case 1A</td>
<td>176</td>
</tr>
<tr>
<td>case 1B</td>
<td>0</td>
</tr>
</tbody>
</table>

Figures 6.1 to 6.4 show the performance comparison of different service disciplines for different number of stations. The results were obtained using the equations 4.5 and 4.6. Note that the gap between the full latency and dual latency becomes greater and greater as the number of stations increases. The 20 station figure also shows the intersection of the dual latency (ordinary service) curve and full latency exhaustive service curve. This intersection exists for all cases but may occur at a much higher utilization for some cases. The reason is that the mean waiting time of the dual latency tends to increase faster than the exhaustive service in high utilization. This can be explained by the stability criterion mentioned in section 4.3:

SECTION 6.1.1 Completely Symmetric Cases 72
* Exhaustive & gated: \[ \rho_0 < 1 \] (4.3)

* Ordinary (limited-to-one):

For all i, \[ \rho_0 + \lambda_i R < 1 \] (4.4)

Therefore, for a completely symmetric ring,

\[
\max(\rho_0) = \frac{b}{b+R/N} = \frac{b}{b+r}
\]

(6.1)

where \( \rho_0 \) is the total utilization, \( \lambda_i \) is the Poisson arrival rate for station i, \( b \) is the mean service time, and \( R \) and \( r \) are the ring latency and station switchover time respectively.

Note that the maximum utilization is independent of N. But why does Figure 6.1 exhibit an intersection and Figures 6.2 to 6.3 do not. The explanation for this is that as the number of stations increases, the chances of a station goes into the low latency state becomes higher for the same utilization. So, there are more time savings for using the dual latency. That makes the intersection occur at higher utilization. Also note the full latency exhaustive service is always stable for \( \rho < 1 \).
Figure 6.5 presents the ratio of the mean waiting time of the full latency and that of the dual latency, both using ordinary service. The first observation is that the ratio is always higher than one. For this case, the dual latency produces a 20% improvement in mean waiting time for 20 stations even at 0.9 utilization. In Figure 6.6, the ratio of the mean waiting time of the full latency exhaustive service and that of the dual latency ordinary service is shown. Note that the ratio falls below the break even line at high utilization as expected. The ratio approaches zero when the dual latency ring becomes unstable. For this case, the $\max[p]$ is equal to roughly 0.975. This confirms the results obtained above. Similar behavior can be found in Figure 6.7 for case 1B. The difference between case 1A and case 1B is that case 1A uses exponential distribution for message length while case 1B uses constant message length. The performance improvement of case 1A is less than that of case 1B. This can be supported by equation 4.14 which gives the ratio of the mean waiting time of the full latency to the mean waiting time of the dual latency.
Figure 6.1 - Performance Comparison of Case 1A for 20 stations.
Figure 6.2 - Performance Comparison of Case 1A for 100 stations.
Figure 6.3 - Performance Comparison of Case 1A for 200 stations.

SECTION 6.1.1 Completely Symmetric Cases
Figure 6.4 - Performance Comparison of Case 1A for 1000 stations.

SECTION 6.1.1 Completely Symmetric Cases 78
Figure 6.5 - Performance Improvement of Full Latency Ordinary Service to Dual Latency Ordinary Service for Case 1A.
Figure 6.6 - Performance Improvement of Full Latency Exhaustive Service to Dual Latency Ordinary Service for Case 1A.
Figure 6.7 - Performance Improvement of Full Latency Exhaustive Service to Dual Latency Ordinary Service for Case 1B.
6.1.2 Asymmetric Cases

The parameters chosen for the asymmetric case are listed below. Cases 2A to 2E were chosen to illustrate the effect of message length on dual latency while cases 3A to 3E were chosen to illustrate the effect of message length distribution from totally exponential gradually to purely constant. The traffic pattern chosen was used as an illustration in many papers [Bux83a], [Bux84], [Ferg85], and [Ever86] and might be of interest to other researchers. They used 1K (exp.) bits + 48 bits overhead as the mean message length. I ran a few points using this mean message length and concluded that this mean message length was too big to show the significance of the dual latency. Hence, I tried to reduce the exponential part by one half to become case 2A and one half of 2A to become 2B and so on to case 2E. But I kept the overhead the same (48 bits).
### Asymmetric Cases (Case 2 & Case 3)

<table>
<thead>
<tr>
<th>Number of stations</th>
<th>20.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic pattern</td>
<td>( \rho_1 = \rho_8 = 40% \rho_0 ), ( \rho_2 = \ldots = \rho_7 = \rho_9 = \ldots = \rho_{20} ).</td>
</tr>
<tr>
<td>Constant switchover time (in bits)</td>
<td>4.5 for full latency.</td>
</tr>
<tr>
<td>Mean message length (in bits)</td>
<td>1.5 for low latency.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean message length (bits)</th>
<th>Exponentially distributed Information (bits)</th>
<th>Constant information (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>case 2A</td>
<td>512</td>
<td>+ 48</td>
<td></td>
</tr>
<tr>
<td>case 2B</td>
<td>256</td>
<td>+ 48</td>
<td></td>
</tr>
<tr>
<td>case 2C</td>
<td>128</td>
<td>+ 48</td>
<td></td>
</tr>
<tr>
<td>case 2D</td>
<td>64</td>
<td>+ 48</td>
<td></td>
</tr>
<tr>
<td>case 2E</td>
<td>32</td>
<td>+ 48</td>
<td></td>
</tr>
<tr>
<td>case 3A</td>
<td>0</td>
<td>+ 176</td>
<td></td>
</tr>
<tr>
<td>case 3B</td>
<td>48</td>
<td>+ 128</td>
<td></td>
</tr>
<tr>
<td>case 3C</td>
<td>88</td>
<td>+ 88</td>
<td></td>
</tr>
<tr>
<td>case 3D</td>
<td>128</td>
<td>+ 48</td>
<td></td>
</tr>
<tr>
<td>case 3E</td>
<td>176</td>
<td>+ 0</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.8 shows the performance improvement expected for case 2 with different mean message lengths. Note that for the asymmetric cases only the mean waiting time for station 1 is shown. Stations 1 and 8 are heavily loaded and of primary concern. The behavior of the mean waiting time for station 1 and station 8 are very similar. From the figure,
the first observation is that the improvement increases rapidly as the utilization goes up. Secondly, when the message length increases the advantage of using the dual latency method decreases. It is due to the fact that long messages have a lower switchover time overhead than short messages.

Figures 6.9 to 6.13 show the performance comparison of full latency with different service disciplines to the dual latency for various mean message lengths. The curves for exhaustive and gated service disciplines were calculated based on Watson's algorithms and were put in to be consistent with case 1. For a mean message of 512 exp.+48 bits (case 2A), the dual latency is better than full latency with gated service up to 0.8 utilization and full latency with exhaustive service up to 0.5. For a message with 128exp.+48 bits (case 2C), the dual latency is better than the gated service up to 0.7 and exhaustive service up to around 0.5 utilization. When the mean message length decreases to 64exp.+48 bits, the dual latency is still better than the gated service up to 0.65 and exhaustive service up to 0.5 utilization. Also, the gap between the dual latency and the full latency becomes greater as the mean message length becomes smaller.
Figure 6.8 - Performance Improvement of Full Latency to Dual Latency for Case 2, different mean message lengths
Figure 6.9 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2A
Figure 6.10 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2B.
PERFORMANCE COMPARISON

CASE 2C (SIMULATION)

Figure 6.11 - Performance Comparison of Full Latency with Different Service Disciplines with the Dual Latency for Case 2C.
Figure 6.12 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2D
Figure 6.13 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 2E
Case 3 tries to investigate what would happen when the mean message length is dominated by either the constant field or the exponential field as in the cases of practical systems. Figure 6.14 shows that the performance improvement becomes more significant as the mean message length has more constant field. Figures 6.15 and 6.16 show the performance comparison among case 3A to case 3E for full latency stations and for dual latency stations respectively. The more constant the mean message length, the smaller the mean waiting time. If we try to compare the two figures, they behave very similar except that figure 6.16 (dual latency) looks like shifted utilization curves of figure 6.15 (full latency). In other words, the dual latency will delay the onset of the instability of the system compared with the full latency.

In real systems like the IEEE 802.5 and the FDDI standards, they usually have a constant header field. That will make the dual latency more appealing than random message lengths. For real-time applications, the system often has a lot of short and constant messages that are time-dependent. That makes the dual latency even more attractive for time-critical applications.
Figure 6.14 - Performance Improvement of Full Latency to the Dual Latency for Different Message Distributions (Case 3)
PERFORMANCE COMPARISON

CASE 3 - STATION 1 (SIMULATION)

Figure 6.15 - Performance Comparison of Full Latency with Different Message Distributions for Case 3
Figure 6.16 - Performance Comparison of Dual Latency with Different Message Distributions for Case 3
6.2 High Speed LANs

High speed LANs have become more important because of the increasing demand for more services and the technological advances in communications. The latencies used in the examples were based on the AMD implementation [AMD86] of the FDDI standard. The station full latency was estimated to be between 62 bits and 45 bits and the low latency was estimated to be 24 bits [Hard88b]. If we assume that the propagation delay for optical fiber is 5 μs/km and the interstation fiber length is 10 m long. Then the propagation delay per station becomes 50 ns which is equivalent to 5 bits time. The variation of the full latency value can be 17 bits and well covers the 5 bits propagation delay. The original estimation of the low latency value was 8.6 bits. Since the FDDI standard specifies that the minimum station delay be three octets, the 24 bits value has to be used. The 5 bits propagation delay is assumed to contribute part of the low latency value from 8.6 bits to 24 bits. Therefore, the 5 bits propagation delay is assumed to be included in both the 62 bits full latency and the 24 bits low latency.
6.2.1 Completely Symmetric Cases

The parameters chosen for the completely symmetric case are listed below. The 312 bits overhead was calculated based on the FDDI standard. The 4K bits (case 4A) exponential field was chosen so that the probability of the messages exceeding the maximum message size (35840 bits) specified in the FDDI standard is very small. Then 4K is further reduced by a factor of 4 to become 1K (case 4B) and so on finally to 64 bits (case 4D).

<table>
<thead>
<tr>
<th>Number of stations</th>
<th>Completely Symmetric Case (Case 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant switchover time (in bits)</td>
<td>62 for full latency.</td>
</tr>
<tr>
<td>Mean message length (in bits)</td>
<td>24 for low latency.</td>
</tr>
<tr>
<td></td>
<td>Case</td>
</tr>
<tr>
<td></td>
<td>case 4A</td>
</tr>
<tr>
<td></td>
<td>case 4B</td>
</tr>
<tr>
<td></td>
<td>case 4C</td>
</tr>
<tr>
<td></td>
<td>case 4D</td>
</tr>
</tbody>
</table>

Figure 6.17 to 6.22 show the performance comparison for different mean message lengths. The performance of the full
latency stations and that of the dual latency stations are very similar except that the performance curve of the dual latency station appears shifted to lower values of utilization. Further, for short messages like case 4D, the mean message waiting time originally is smaller for low utilization, but as the utilization increases, the waiting time becomes bigger. When the utilization is low, there is less chance that a short message will have to wait for another short message. But when the utilization is high, the chances of a short message having to wait for another short message is greater because of the increased switchover time overhead for short messages.

Figures 6.23 to 6.26 show the performance comparison of full latency with different service disciplines to the dual latency ordinary service. Again, it is observed that the gap between the full latency and the dual latency is greater for smaller mean message length. Further, the gap increases as the number of stations increases.

Figure 6.27 and 6.28 show the effect of number of stations on the performance improvement using the dual latency method. As we noted before for short message length the performance improvement is more significant than for long message length.
EFFECT OF MESSAGE LENGTHS

CASE 4 (THEORETICAL)

Figure 6.17 - Performance Comparison of Full Latency with Different Message Lengths for Case 4, 20 Stations.

SECTION 6.2.1 Completely Symmetric Cases
EFFECT OF MESSAGE LENGTHS

CASE 4 (THEORETICAL)

Figure 6.18 - Performance Comparison of Dual Latency with Different Message Lengths for Case 4, 20 Stations.

SECTION 6.2.1 Completely Symmetric Cases
EFFECT OF MESSAGE LENGTHS

CASE 4 (THEORETICAL)

Figure 6.19 - Performance Comparison of Full Latency with Different Message Lengths for Case 4, 100 Stations.

SECTION 6.2.1 Completely Symmetric Cases
Figure 6.20 - Performance Comparison of Dual Latency with Different Message Lengths for Case 4, 100 Stations
EFFECT OF MESSAGE LENGTHS

CASE 4 (THEORETICAL)

Figure 6.21 - Performance Comparison of Full Latency with Different Message Lengths for Case 4, 1000 Stations.
Figure 6.22 - Performance Comparison of Dual Latency with Different Message Lengths for Case 4, 1000 Stations.
Figure 6.23 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4B, 100 Stations.
Figure 6.24 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4D, 100 Stations.
COMPARISON OF PERFORMANCE

CASE 4B (THEORETICAL)

Figure 6.25 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4B, 1000 Stations.

SECTION 6.2.1 Completely Symmetric Cases
Figure 6.26 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency for Case 4D, 1000 Stations
Figure 6.27 - Performance Improvement of Full Latency Ordinary Service to the Dual Latency Ordinary Service for Case 4A.
EFFECT OF # OF STATIONS

CASE 4D (THEORETICAL)

Fig 6.28 - Performance Improvement of Full Latency Ordinary Service to the Dual Latency Ordinary Service for Case 4D.

SECTION 6.2.1 Completely Symmetric Cases
6.2.2 Asymmetric Cases

The parameters chosen for the asymmetric cases are listed in case 5 and case 6 below. The difference between the two cases is that the full latency is 62 bits and 45 bits for case 5 and case 6 respectively while the low latency is unchanged (24 bits). That gives us the improvement expected when the full latency decreases. The message length was chosen to be 256 bits in the exponential field so that it is the same as case 4C and the information field size is comparable to the overhead field size.

<table>
<thead>
<tr>
<th></th>
<th>Asymmetric Case (Case 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of stations</td>
<td>20.</td>
</tr>
<tr>
<td>Traffic pattern</td>
<td>$\rho_1 = \rho_8 = 40% \rho_0$ , $\rho_2 = ... = \rho_7 = \rho_9 = ... = \rho_{20}$ .</td>
</tr>
<tr>
<td>Constant switchover</td>
<td>62 for full latency.</td>
</tr>
<tr>
<td>time (in bits)</td>
<td>24 for low latency.</td>
</tr>
<tr>
<td>Mean message length</td>
<td>Exponentially distributed Information (bits)</td>
</tr>
<tr>
<td>(in bits)</td>
<td>256</td>
</tr>
</tbody>
</table>
Asymmetric Case (Case 6)

<table>
<thead>
<tr>
<th>Number of stations</th>
<th>20.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic pattern</td>
<td>(\rho_1 = \rho_5 = 40% \rho_0), (\rho_2 = \ldots = \rho_7 = \rho_9 = \ldots = \rho_{20}).</td>
</tr>
<tr>
<td>Constant switchover time (in bits)</td>
<td>45 for full latency. 24 for low latency.</td>
</tr>
<tr>
<td>Mean message length (in bits)</td>
<td>Exponentially distributed Information (bits)</td>
</tr>
<tr>
<td></td>
<td>256</td>
</tr>
</tbody>
</table>

Figures 6.29 and 6.30 show the performance comparison for case 5 and case 6 respectively. Note that for asymmetric cases only the mean waiting time for station 1 is shown. Since it is the heavily loaded station, it is of primary concern. Since the full latency is greater in case 5 than in case 6, a longer mean message waiting time is expected for case 5. The figures confirm this rule. Both cases indicate that the dual latency performs better than the gated and exhaustive services up to around 0.4-0.5 utilization. Compared with the completely symmetric cases of the same system, the dual latency is better than the gated and exhaustive services up to much higher utilization. The reason is that for an asymmetric case the traffic is much more concentrated and queues can be built up easily. That
makes the time savings using the dual latency less significant since the time savings has to divided by the number of messages in the queues.

Figure 6.31 shows the range of performance improvement expected if the full latency changes from 62 bits to 45 bits. The performance improvement for case 5 and 6 is fairly constant and it rises rapidly as the utilization goes up. This sharp increase in performance can be explained by the fact that the heavily loaded stations start to build up queues. When the queues build up, the time savings of using dual latency starts to produce multiplying effect.
COMPARISON OF PERFORMANCE

CASE 5 - STATION 1 (SIMULATION)

MEAN WAITING TIME IN BITS

UTILIZATION (256exp.*312 bits message)

- Full latency
- Dual latency
- Exhaustive service
- Gated service

Figure 6.29 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency Ordinary Service for Case 5

SECTION 6.2.2 Asymmetric Cases 113
Figure 6.30 - Performance Comparison of Full Latency with Different Service Disciplines to the Dual Latency Ordinary Service for Case 6.
PERFORMANCE IMPROVEMENT

CASE 5 & 6 - STATION 1 (SIMULATION)

Figure 6.31 - Performance Improvement of Full Latency Ordinary Service to the Dual Latency Ordinary Service for Case 5 and Case 6.
VII. CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary

The performance measure used for this study is the mean message waiting time. The study shows that a dual latency ring always performs better than a full latency ring. The degree of improvement depends on a number of parameters: number of stations, arrival statistics, message length statistics, and utilization of the system. The improvement is more significant when the LAN has a large number of stations, short messages or constant messages.

7.2 Conclusions

The conclusions drawn below are useful for the performance up to the MAC layer in a token ring LAN.

1) Performance of the dual latency ring system is always better than that of the full latency system. The following table shows the percentage reduction of the mean waiting time if the dual latency ring is used instead of the full latency ring for the cases described in section 6.
<table>
<thead>
<tr>
<th>CASE</th>
<th>NUMBER OF STATIONS</th>
<th>RANGE OF UTILIZATION</th>
<th>PERCENTAGE REDUCTION IN MEAN MESSAGE WAITING TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE 1A</td>
<td>20</td>
<td>0 - 0.9</td>
<td>67% - 15%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.9</td>
<td>67% - 40%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.9</td>
<td>67% - 63%</td>
</tr>
<tr>
<td>CASE 1B</td>
<td>20</td>
<td>0 - 0.9</td>
<td>67% - 25%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.9</td>
<td>67% - 50%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.9</td>
<td>67% - 65%</td>
</tr>
<tr>
<td>CASE 2A</td>
<td>20</td>
<td>0 - 0.5 - 0.8</td>
<td>67% - 15% - 23%</td>
</tr>
<tr>
<td>CASE 2E</td>
<td>20</td>
<td>0 - 0.3 - 0.6</td>
<td>67% - 56% - 74%</td>
</tr>
<tr>
<td>CASE 3A</td>
<td>20</td>
<td>0 - 0.4 - 0.7</td>
<td>67% - 42% - 54%</td>
</tr>
<tr>
<td>CASE 3E</td>
<td>20</td>
<td>0 - 0.4 - 0.7</td>
<td>67% - 32% - 46%</td>
</tr>
<tr>
<td>CASE 4A</td>
<td>20</td>
<td>0 - 0.85</td>
<td>61% - 10%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.85</td>
<td>61% - 29%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.85</td>
<td>61% - 55%</td>
</tr>
<tr>
<td>CASE 4D</td>
<td>20</td>
<td>0 - 0.85</td>
<td>61% - 49%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.85</td>
<td>61% - 58%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.85</td>
<td>61% - 61%</td>
</tr>
<tr>
<td>CASE 5</td>
<td>20</td>
<td>0 - 0.5</td>
<td>61% - 88%</td>
</tr>
<tr>
<td>CASE 6</td>
<td>20</td>
<td>0 - 0.5 - 0.6</td>
<td>47% - 60% - 92%</td>
</tr>
</tbody>
</table>

2) Performance of the dual latency system is better than that of the gated and exhaustive service systems up to medium range of utilization. The table below lists the percentage reduction in mean message waiting time when the dual latency ring is better than the exhaustive service ring for the cases described in section 6.
<table>
<thead>
<tr>
<th>CASE</th>
<th>NUMBER OF STATIONS</th>
<th>RANGE OF UTILIZATION</th>
<th>PERCENTAGE REDUCTION IN MEAN MESSAGE WAITING TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>20</td>
<td>0 - 0.8</td>
<td>67% - 5%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.9</td>
<td>67% - 20%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.9</td>
<td>67% - 51%</td>
</tr>
<tr>
<td>1B</td>
<td>20</td>
<td>0 - 0.8</td>
<td>67% - 16%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.9</td>
<td>67% - 33%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.9</td>
<td>67% - 54%</td>
</tr>
<tr>
<td>2A</td>
<td>20</td>
<td>0 - 0.4</td>
<td>67% - 8%</td>
</tr>
<tr>
<td>2E</td>
<td>20</td>
<td>0 - 0.5</td>
<td>67% - 10%</td>
</tr>
<tr>
<td>3A</td>
<td>20</td>
<td>0 - 0.5</td>
<td>67% - 6%</td>
</tr>
<tr>
<td>3E</td>
<td>20</td>
<td>0 - 0.5</td>
<td>67% - 10%</td>
</tr>
<tr>
<td>4A</td>
<td>20</td>
<td>0 - 0.7</td>
<td>61% - 15%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.7</td>
<td>61% - 32%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.7</td>
<td>61% - 37%</td>
</tr>
<tr>
<td>4D</td>
<td>20</td>
<td>0 - 0.85</td>
<td>61% - 0%</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0 - 0.85</td>
<td>61% - 22%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0 - 0.85</td>
<td>61% - 34%</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>0 - 0.4</td>
<td>61% - 9%</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>0 - 0.3</td>
<td>47% - 12%</td>
</tr>
</tbody>
</table>

3) Dual latency becomes more attractive when there are many stations. In case 1A, the percentage reduction in mean message waiting time is 15% for 20 stations, 40% for 100 stations and 63% for 1000 stations at 0.9 utilization (refer to the table in conclusion #1). In case 1B, the percentage reduction is 25% for 20
stations, 50% for 100 stations and 65% for 1000 stations at 0.9 utilization. In case 4A, the percentage reduction is 10% for 20 stations, 29% for 100 stations and 55% for 1000 stations at 0.85 utilization. In case 4B, the percentage reduction is 49% for 20 stations, 58% for 100 stations and 61% for 1000 stations at 0.85 utilization.

4) Short messages are very suitable for the dual latency method. Case 4A has the longest mean message length and case 4D has the shortest one. For 20 stations, the percentage reduction in mean message waiting time is 10% in case 4A and 49% in case 4D at 0.85 utilization (refer to the table in conclusion #1). For 100 stations, the percentage reduction is 29% in case 4A and 58% in case 4D at 0.85 utilization. For 1000 stations, the percentage reduction is 55% in case 4A and 61% in case 4D at 0.85 utilization.

5) Constant messages make the dual latency method more favorable than other types of messages. Case 1A has exponentially distributed messages while case 1B has constant messages. For 20 stations, the percentage reduction in mean message waiting time is 15% in case 1A and 25% in case 1B at 0.9 utilization (refer to the table in conclusion #1). For 100 stations, the
percentage reduction is 40% in case 1A and 50% in case 1B at 0.9 utilization. For 1000 stations, the percentage reduction is 63% in case 1A and 65% in case 1B at 0.9 utilization.

6) The shorter the message length, the smaller the delay for low utilization. But the shorter the message length, the higher the delay for high utilization.

7) For asymmetric traffic, the mean waiting times of the heavily loaded stations become unstable when the network utilization goes up. The dual latency approach allows higher level of traffic utilization before the onset of instability. This explains the trend of increasing percentage reduction in mean message waiting time of the dual latency ring to the latency ring in some cases described in section 6. In case 2A, the percentage reduction is 67% at zero utilization, 15% at 0.5 utilization and 23% at 0.8 utilization (refer to the table in conclusion #1). In case 2E, the percentage reduction is 67% at zero utilization, 56% at 0.3 utilization and 74% at 0.6 utilization. In case 3A, the percentage reduction is 67% at zero utilization, 42% at 0.4 utilization and 54% at 0.7 utilization. In case 3E, the percentage reduction is 67% at zero utilization, 32%
at 0.4 utilization and 46% at 0.7 utilization. In case 5, the percentage reduction increases from 61% to 88% when the utilization grows from zero to 0.5. In case 6, the percentage reduction increases from 61% to 92% when the utilization grows from zero to 0.6.

7.3 Recommendations

1) Since the performance of the dual latency method is always better than or equal to the full latency, the incorporation of the dual latency into the Network Interface Unit would be very desirable. The cost of introducing the dual latency circuitry should be investigated further.

2) The dual latency method is very attractive for real-time applications because their message lengths are often small. The other characteristics and requirements of real-time applications should be explored more to determine if the dual latency method can fulfil them.

3) When the chip implementation of the IEEE 802.5 standard network interface unit for the dual latency method is ready, the performance improvement can then be experimentally measured and compared with the simulation results.
8.1 Real-Time Applications

Many existing LANs have been developed for the office environment. However, there is a demand of LANs for real-time situations. It is unclear if the currently available protocols would be suitable to handle real-time applications. Examples are defense systems, industrial process control systems and integrated telecommunications systems. There are three major requirements identified for real-time LANs [Lann85]. They are timing, robustness and flexibility requirements. The same research group also stresses the importance of modeling of avalanche traffic (e.g. alarm messages) for real-time systems [Boud87]. Since the dual latency method is more suitable for real-time applications, avalanche simulation may be able to reveal other important properties of the dual latency method. Hence, modeling of avalanche simulation should be investigated further.

8.2 End-to-end Performance Modeling

The methods available for performance modeling of the physical layer, and MAC layer are quite successful in general and
proven useful in practice. Nevertheless, the performance of the first two layers may not be the performance experienced by a user. That makes the end-to-end performance very useful; however, universal techniques for end-to-end performance modeling are not yet available [Reis86]; some proposed methods can be found in [Mitc86] and [Mura87]. The end-to-end performance modeling may involve analysis of a network of queues. Since the simulation software was developed for the first two layers, higher layer models may be developed and added on top of the software to predict the end-to-end performance.
REFERENCES


REFERENCES


REFERENCES


