

## INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

# U·M·I

University Microfilms International  
A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
313/761-4700 800/521-0600

**Order Number 9007869**

**Performance analysis of token bus protocols for integrated  
control system networks**

**Hong, Seung Ho, Ph.D.**

**The Pennsylvania State University, 1989**

**U·M·I**  
300 N. Zeeb Rd.  
Ann Arbor, MI 48106

The Pennsylvania State University

The Graduate School

PERFORMANCE ANALYSIS OF TOKEN BUS PROTOCOLS  
FOR INTEGRATED CONTROL SYSTEM NETWORKS

A Thesis in

Mechanical Engineering

by

Seung Ho Hong

©1989 Seung Ho Hong

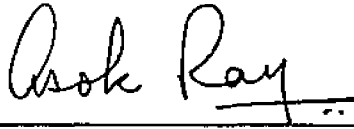
Submitted in Partial Fulfillment  
of the Requirements  
for the Degree of

Doctor of Philosophy

August 1989

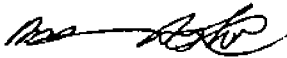
We approve the thesis of Seung Ho Hong.

Date of Signature



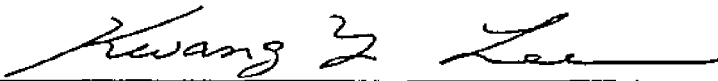
April 7, 1989

Asok Ray  
Associate Professor of Mechanical  
Engineering  
Chair of Committee  
Thesis Adviser



April 10, 1989

Ashok Belegundu  
Assistant Professor of Mechanical  
Engineering




April 10, 1989

Kwang Y. Lee  
Associate Professor of Electrical  
Engineering



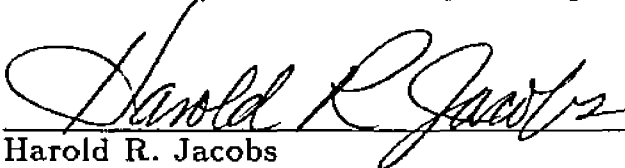
April 10, 89

Uri Tsach  
Assistant Professor of Mechanical  
Engineering



April 10, 1989

James C. Wambold  
Professor of Mechanical Engineering



April 12, 1989

Harold R. Jacobs  
Professor of Mechanical Engineering  
Head of the Department of  
Mechanical Engineering

## ABSTRACT

Integrated Communication and Control System (ICCS) networks require accommodation of heterogeneous traffic of real-time and non-real-time data. The token bus protocol is one of the most widely accepted Medium Access Control (MAC) protocols for ICCS networks because of its performance characteristics such as bounded data latency and high throughput. In addition, token bus protocols have a priority mechanism that is capable of handling heterogeneous traffic.

In this thesis, an analytical model which evaluates the performance of the priority scheme in token bus protocols has been developed. The analytical model is based on the concepts of the conditional token circulation time and conditional effective service time, and it provides a direct relationship between the network parameters (e.g., number of stations in the network, message length, message interarrival time and priority timer setting) and the network performance measures (e.g., statistical characteristics of message data latency for all priority classes). The analytical model has been validated by simulation experiments under different conditions of network traffic.

Performance of the priority scheme in token bus protocols can be predicted by using the analytical model. The use of the analytical model has been illustrated for initial design and optimization of ICCS networks. On the basis of this research, a systematic methodology for ICCS design can be developed.

## TABLE OF CONTENTS

LIST OF FIGURES .....	vi
LIST OF TABLES .....	viii
LIST OF SYMBOLS .....	ix
ACKNOWLEDGEMENTS .....	xii
Chapter 1. INTRODUCTION .....	1
1.1. Introduction to Integrated Communication and Control Systems (ICCS) .....	1
1.2. ICCS Network and Design Problems .....	3
1.2.1. Taxonomy of Network Access Protocols .....	5
1.2.2. Design Problems in ICCS Networks .....	7
1.3. Priority Scheme in ICCS Networks .....	7
1.4. Research Objectives .....	9
1.5. Organization of the Thesis .....	12
Chapter 2. DESCRIPTION OF TOKEN BUS NETWORK PROTOCOL .....	14
2.1. Token Bus Protocol .....	14
2.2. Priority Scheme in the Token Bus Protocol .....	17
Chapter 3. DEVELOPMENT OF AN ANALYTICAL MODEL FOR THE PRIORITY SCHEME OF TOKEN BUS PROTOCOL .....	20
3.1. Analysis of Conditional Token Circulation Time .....	27
3.2. Analysis of Conditional Effective Service Time .....	39
3.3. Performance Analysis of Priority Scheme .....	51
Chapter 4. PETRI NET MODEL FOR PRIORITY SCHEME AND DEVELOPMENT OF A SIMULATION MODEL .....	54
4.1. Petri Net Model for Priority Scheme .....	54
4.2. Development of a Simulation Model .....	64

Chapter 5. COMPARISON OF ANALYTICAL MODEL WITH SIMULATION EXPERIMENT .....	71
Chapter 6. PRELIMINARY DESIGN OF ICCS NETWORKS .....	92
6.1. Preliminary Design Criteria of ICCS Network .....	92
6.2. An Example of ICCS Network Design .....	95
Chapter 7. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK .....	103
7.1. Summary .....	103
7.2. Conclusions .....	106
7.3. Recommendations for Future Work .....	107
REFERENCES .....	108
Appendix A. DERIVATION OF APPROXIMATE WAITING TIME FOR THE PRIORITY SCHEME .....	112
A.1. State Distribution at Scan Instants .....	113
A.2. State Distribution at Departure Instants .....	116
A.3. Waiting Time Analysis .....	117
Appendix B. SOLUTION APPROACH FOR THE OPTIMIZATION PROBLEM .....	119

## LIST OF FIGURES

1.1	A Network Architecture for Integration of Design and Manufacturing System . . . . .	4
2.1	Schematic Diagram of Token Bus Protocol . . . . .	15
3.1	Model for the Analysis of Token Bus Protocol with Four Levels of Priority . . . . .	22
3.2	An Epoch of Message Transmission Opportunity at a Priority $i$ Queue . . . . .	30
3.3	Probability Density Function of $T'_i$ without Experiencing $TRT_i$ Expiration . . . . .	42
3.4	Probability Density Function of $T'_i$ with Experiencing $TRT_i$ Expiration Once . . . . .	44
4.1	Petri Net Submodel for Channel . . . . .	57
4.2	Petri Net Submodel for Station $j$ . . . . .	58
4.3	Petri Net Submodel for Priority 0 queue . . . . .	60
4.4	Petri Net Submodel for Priority $i$ queue . . . . .	61
4.5	Event Interaction Diagram of Discrete-Event Simulation . . . . .	67
5.1	Average Data Latency for Asymmetric Traffic ( $G=0.2$ ) . . . . .	75
5.2	Average Data Latency for Asymmetric Traffic ( $G=0.5$ ) . . . . .	76
5.3	Average Data Latency for Asymmetric Traffic ( $G=0.8$ ) . . . . .	77
5.4	Probability Density Function of $T_r$ at Traffic Condition of Case 1 . . . . .	79



5.5	Kuehn's Approximation for Asymmetric System . . . . .	80
5.6	Average Data Latency for Symmetric Traffic ( $G=0.2$ ) . . .	83
5.7	Average Data Latency for Symmetric Traffic ( $G=0.5$ ) . . .	85
5.8	Average Data Latency for Symmetric Traffic ( $G=0.8$ ) . . .	87
5.9	Probability Density Function of $T_r$ at Traffic Condition of Case 2 . . . . .	90
6.1	Perturbation of the Sum of Data Latency Variances of Real-Time Data with respect to $TRT_i^*$ . . . . .	98
6.2	Perturbation of the Sum of Effective Service Time Variances of Real-Time Data with respect to $TRT_i^*$ . .	99

**LIST OF TABLES**

6.1	Average Data Latencies at Medium Traffic Load . .	100
6.2	Average Data Latencies at High Traffic Load . . . . .	101

## LIST OF SYMBOLS

$c_i$	compensation coefficient for the average data latency of the priority $i$ class
$D_i$	data latency of the priority $i$ class
$f_{T'_{ri}}(t)$	probability density function of $T'_{ri}$
$f_{T''_{ri}}(t)$	probability density function of $T''_{ri}$
$f_{T'_i}(t)$	probability density function of $T'_i$
$f_1(t)$	probability density function of $T'_{ri}$ during which $TRT_i$ is not expired
$f_2(t)$	probability density function of $T'_{ri}$ during which $TRT_i$ is expired
$F_{T'_{ri}}(TRT_i)$	probability distribution function of $T'_{ri}$
$G$	total offered traffic
$G_i$	offered traffic of the priority $i$ class
$K$	the lowest priority level
$L_i$	message transmission time of the priority $i$ class
$M$	bound of the priority level which accommodates real-time data
$M_{kj}$	number of the priority $k$ queues transmitting their messages during $T'_{rj}$
$N_i$	number of the priority $i$ queues in the network
$P_i$	probability that the token serves a message when it arrives at a priority $i$ queue
$P'_{ij}$	conditional probability that the token serves a message when it arrives at a priority $i$ queue during $T'_{rj}$

$P_{ij}''$	conditional probability that the token serves a message when it arrives at a priority $i$ queue during $T_{rj}''$
$Q_i$	average queue length of the priority $i$ class
$R$	total idle time during a token circulation
$T_i$	effective service time of the priority $i$ class, which is defined as the time interval from the instant a queue has an opportunity to transmit a message till the next instant the same queue has an opportunity to transmit a message
$T_i'$	conditional effective service times of a priority $i$ queue during which a message is not served from the queue
$T_i''$	conditional effective service times of a priority $i$ queue during which a message is served from the queue
$\overline{T_i'}$	average of $T_i'$
$\overline{T_i''}$	average of $T_i''$
$\overline{T_i'^2}$	second moment of $T_i'$
$\overline{T_i''^2}$	second moment of $T_i''$
$T_r$	token circulation time which is defined as the elapsed time between two consecutive instants at which a transmitter queue captures the token
$T_{ri}'$	conditional token circulation times of a priority $i$ queue during which a message is not served from the queue
$T_{ri}''$	conditional token circulation times of a priority $i$ queue during which a message is served from the queue

$\overline{T'_{ri}}$	average of $T'_{ri}$
$\overline{T''_{ri}}$	average of $T''_{ri}$
$\overline{T'^2_{ri}}$	second moment of $T'_{ri}$
$\overline{T''^2_{ri}}$	second moment of $T''_{ri}$
$\overline{T'^a_r}$	average value of $T'_{ri}$ during which $TRT_i$ is not expired
$\overline{T'^e_r}$	average value of $T'_{ri}$ during which $TRT_i$ is expired
$TRT_i$	token rotation timer value of the priority $i$ class
$TRT_i^*$	optimal token rotation timer value of the priority $i$ class
$w_i$	design safety factor for the priority $i$ class
$W_i$	queueing delay of the priority $i$ class
$\lambda_i$	average message arrival rate at the priority $i$ queue
$\mu_i$	probability that $TRT_i$ is not expired during $T_r$
$\mu'_{ij}$	conditional probability that $TRT_i$ is not expired during $T'_{rj}$
$\mu''_{ij}$	conditional probability that $TRT_i$ is not expired during $T''_{rj}$
$\delta_i$	bound of average data latency at the priority $i$ class
$\Phi_{T'_{ri}}(s)$	moment generation function of $T'_{ri}$
$\Phi_{T'_i}(s)$	moment generation function of $T'_i$
$\sigma'_j$	variance of $T'_j$
$\sigma''_j$	variance of $T''_j$
$\sigma^2_{T'_{ri}}$	variance of $T'_{ri}$
$\sigma^2_{T''_{ri}}$	variance of $T''_{ri}$
$\tau_i$	average message interarrival time at the priority $i$ queue

## ACKNOWLEDGEMENTS

I sincerely appreciate Professor Asok Ray for his continuous guidance and encouragement during my Ph.D. program. I appreciate comments of the committee members on the thesis. I would like to thank the Bendix Flight Systems Division for its financial support during the research.

I am very indebted to my parents. Thank you, father and mother.

## Chapter 1

### INTRODUCTION

#### 1.1. Introduction to Integrated Communication and Control Systems

A revolutionary development in sensor technology and the associated enhancement of information transmission capabilities confront the designer of complex engineering systems with an overwhelming amount of information about the environment and the process to be controlled. For example, in an automated factory, a variety of sensors delivers information about position and spatial relations among robots, machine tools and objects in the workspace. Data are generated not only by these sensors but also by Computer-Aided Design (CAD) models of various objects. Designing the architecture of an information processing system that will fuse the information from all these sources to obtain a consistent view of the workspace is an important problem. Similar problems exist in other complex systems, such as aircraft and spacecraft control, military communications, and office automation.

One omnipresent class of architectures that has received much attention from several disciplines is that of networks. Computers used to control such large-scale systems are themselves connected to form computer communication networks. The sensor data and the process information generated from a variety of distributed systems are exchanged through the network. Networks form the backbone of Integrated Communication and Control Systems (ICCS) in complex dynamical processes.

There are many reasons why ICCS using computer networks have experienced such significant advances during the past few years. One of the most important factors is the dramatic and continuing decrease in computer hardware costs, accompanied by an increase in computer hardware capacity. Today's microprocessors have speeds, instruction sets, and memory capacities comparable to medium scale minicomputers. This trend has brought a number of changes in the way information is collected, processed, and used in organizations. The large-scale systems are decomposed into several distributed single-function systems and intelligent workstations to make them more accessible and user-friendly. The reduction in hardware cost correspondingly decreases hardware life cycles, which aggravates software conversion problems. These conversion costs can be reduced by decomposing large-scale systems into smaller, separate components.

All these factors lead to an increased number of interconnected subsystems at a single site. Examples are automated factory, advanced aircraft and spacecraft, and the operation center of a large organization. The major motivation for creating an interconnected system is:

1. Exchange of data between individual systems.
2. Provision of backup facilities in real-time applications.
3. Sharing expensive and scarce resources.

A major challenge in large scale manufacturing automation is to integrate the "factors of productions" such as various design and production processes, the material inventory, sales marketing, purchasing, administration and engineering processes into a single, closed-loop, control system. Computer Integrated Manu-



facturing (CIM) organizes a number of computing machines from the mainframe and minicomputers in the office environment to microcomputer workstations at the cell control level, and microprocessors and programmable logic controllers at the factory floor level [1]. A high degree of flexibility and modularity in manufacturing automation can be achieved by partitioning the shop level facilities into several virtual cells.

Essential to this distributed total manufacturing system is the communication network over which the necessary information will flow. As proposed by Ray [1], a network architecture for integration of design and manufacturing system via a single high-speed medium is illustrated in Figure 1.1. The mainframe computers at the design and administration office are responsible for Manufacturing Management such as Manufacturing Requirements Planning (MRP), Tool Management (TM), CAD/CAM functions, and interface with the cell control workstations. Each workstation at the cell control level is connected to its own Data Control Point (DCP). Information of different cells are directly exchanged with each other through a common network medium to share the common resources.

## 1.2. ICCS Network and Design Problems

Just as it is in any field, the development of computer network systems is subject to a number of constraints. The basic features are simplicity, flexibility and reliability. Since the environments in question are generally characterized by a large number of devices that require interconnections, they call for networks with simple topologies and low-cost interfaces that provide considerable flexibility

Engineering Design, Data Processing, Scheduling, and Accounting

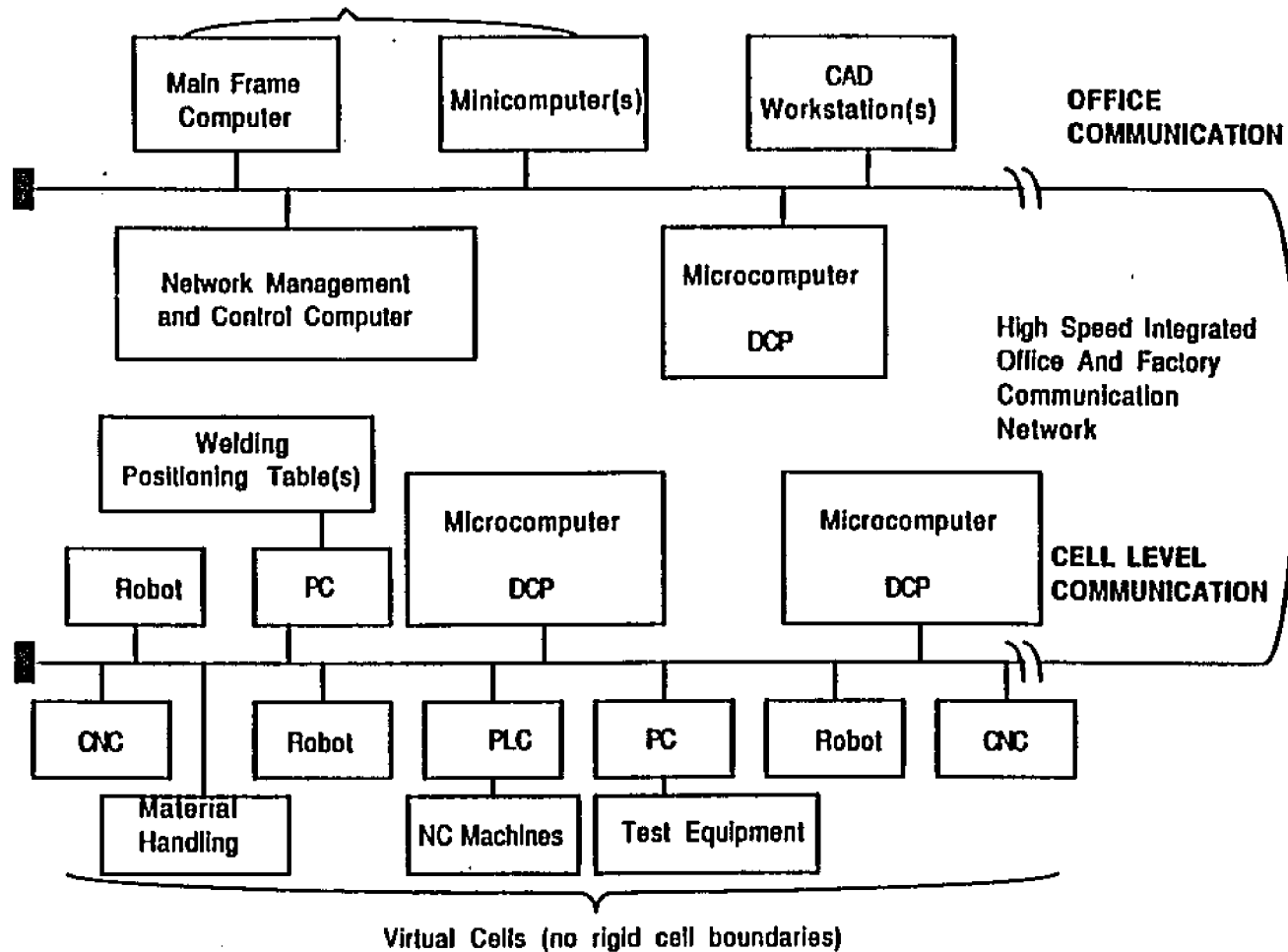


Figure 1.1: A Network Architecture for Integration of Design and Manufacturing System.

for accommodating the variability in the environment and ensure the desired reliability. *Multiple-access* protocols with the broadcast communication capabilities eliminate complex topological design problems, which cannot be solved by using conventional *point-to-point store-and-forward* network protocols.

Since the traffic in the computer communication is usually bursty, it is more efficient to provide an available communication bandwidth as a single high-speed channel which is shared by the contending users, rather than furnishing dedicated low-speed channels to individual users [2]. This solution conveys with it the attendant benefits of the law of large numbers, which states that, with a very high probability, the aggregate demand placed on the channel is equal to the sum of the average individual demands.

### 1.2.1. Taxonomy of Network Access Protocols

Multiple-access protocols with broadcast communication can be broadly divided into two categories [3]: (1) random access protocols (i.e., CSMA and CSMA/CD) and (2) cyclic service protocols (i.e., token bus and token ring)

In *random access protocols*, the users transmit at any time they desire; when conflicts occur, the conflicting users reschedule transmission of their packets. At light traffic, this technique is effective. However, as the traffic load increases, the risk of a packet collision increases. At heavy load, random access protocols show poor channel utilization and have the potential for very large delays and complete disruption of the message transmission process. Since many real-time control loops exist in ICCS, the network-induced delays should be kept within *a priori* specified

limits. Therefore, random access protocols are not considered suitable for ICCS [4].

In *cyclic service protocols*, the users are served in a cyclic order by a travelling server (i.e., token, empty slot, polling signal, etc.). Cyclic service protocols exhibit a bounded delay and high throughput characteristics even in the high traffic load, thus suitable for ICCS.

Ray et al. [4,5] analyzed the performance characteristics (delay, throughput, network flexibility and reliability, etc.) of several multiple-access protocols such as SAE linear token bus [6], SAE token ring [7], and MIL-STD-1553B [8] in view of the ICCS network design requirements using the combined discrete-event and continuous-time simulation technique. This study indicates that SAE linear token bus and SAE token ring show better performance over MIL-STD-1553B in view of low data latency and high throughput.

Although the token ring protocol exhibits performance characteristics similar to that of the token bus protocol, it has a limitation on the system reliability. If an interface fails in a ring network, the whole network system may stop operating. Another limitation is that expanding the ring is complex; expansion requires hardware modifications. For these reasons, token bus is selected as the most suitable network protocol for ICCS. The Manufacturing Automation Protocol (MAP) [9] which has been widely accepted as a standard for automated factory communication network by the international community is based on the IEEE 802.4 Token Passing Bus Protocol [10].

### 1.2.2. Design Problems in ICCS Networks

The advantages of multiple-access protocol over point-to-point connections include reduced wiring and power requirements, flexibility of operations, evolutionary design process, and ease of maintenance, and diagnostics and monitoring. However, since several users share the common channel, if the network traffic is not carefully controlled, undesirable effects such as congestion and monopolization arise. The congestion and monopolization increase data latency, and decrease throughput. Even worse, they could make the network operations unstable.

In the token bus protocol, a computer network may be thought of as a productive resource whose capacity must be shared dynamically by a community of competing users (or, more generally, processes). A computer network system usually supports a large number of heterogeneous applications such as real-time data, real-time voice, file transfer, station management data, etc. To enhance utilization of the network, and to provide timely communications between the processes, the channel bandwidth should be allocated according to various service requirements of different applications. Priority scheme which is discussed in the next section offers solutions to this problem.

### 1.3. Priority Scheme in ICCS Networks

In ICCS networks, allowable data latency at each user is limited to its maximum value. Some users (e.g., real-time data and voice transmission) must have smaller allowable data latency than others (e.g., file transfer), and, thus, they are more time constrained. These users should receive preferential treatment at the

expense of others which can tolerate larger data latency. A Priority scheme is essential to decrease the data latency for time-critical users.

As an example, consider a CIM which consists of several different workcells that have their own control loops. Some control loops may have faster dynamics than others, and are thus more sensitive to data latency. Also, the system and network management data should be transmitted within a bounded time interval in order to operate the totally distributed systems in a stable state. To achieve a smaller data latency, these real-time data should have more frequent opportunity of transmission than others.

In a multiple-access protocol, the goal of priority scheme can be achieved by providing a better opportunity of message transmission to the time-critical users. A straightforward approach for implementing message priorities in multiple-access protocols is obtained by a simple *reservation access mechanism* [11]. In this approach, each station schedules a message for transmission in the scheduling period corresponding to the message priority. However, when the number of users is large or the scheduling overhead becomes nonnegligible, the scheduling penalty may become prohibitively high. In addition, this scheme tends to increase the probability of errors due to channel corruption.

An approach which attempts to reduce this scheduling overhead is the *timing mechanism* which is introduced in token bus protocols. In this approach, except for the highest priority queue, each prioritized queue has a timer called Token Rotation Timer (*TRT*). The highest priority queue can transmit its message whenever the token arrives. When the token arrives at the lower priority queues,

they check their timer values. If the timer is not expired, the waiting message can be transmitted. If the timer is expired, the queue defers transmission and passes the token to the next queue. The timer is reset and restarted whenever the queue captures the token. Several priority levels can be achieved by choosing different timer values for different priority queues. The timer value of the higher priority queue is larger than that of the lower priority queue; thus the higher priority queue has a lower probability of timer expiration and, therefore, a higher probability of message transmission.

#### 1.4. Research Objectives

In ICCS, a number of different kinds of service requirements are imposed on the capacity-limited communication network. Therefore, appropriate control of medium access is demanded for each application, i.e., the access of users into the network must be effectively limited according to their functional characteristics. This is the motivation for this research. This goal can be achieved by a *priority scheme*, and the rigorous analysis for the performance of this scheme is required.

While the need for providing prioritized service is evident, the amount of rigorous analytic studies in this field has been surprisingly small. For random access protocols (CSMA/CD), priority functions were proposed, and their performances have been evaluated [12,13]. However, very little work has so far been reported for analysis of priority schemes in token bus protocols. This is mainly because of the mathematical complexity that arises from deriving the service time distribution at each priority level. Jayasumana and Fisher [14,15] developed an analytical model

which can find approximate throughput characteristics of each priority class for the IEEE 802.4 Token Passing Bus Protocol. But they did not consider the delay characteristics, which is often more important in the design of ICCS networks.

Analytical studies for obtaining the delay of cyclic service systems without priority scheme have been studied by several investigators [16-19]. These analyses are based on the assumptions of *exhaustive service system* and *gated service system* [16]. In the exhaustive service system, a station receiving the token transmits all messages waiting in the queue until it becomes empty. In the gated service system, a token receiving station transmits all messages that are currently in the queue when the token arrives. Exhaustive and gated service polling systems have the tendency to monopolize the server by one station, especially at high loads. Therefore, these assumptions are not suitable for application to ICCS networks, which deal with real-time systems where the allowable data latency of a message at each station is limited to its maximum value. From the practical point of view, an ICCS network may be viewed to consist of several control loops. Each control loop consists of sensor computer(s) and controller computer(s). The data generated from sensor and controller computers need to be transmitted on a timely basis.

The exact analysis of nonexhaustive services discipline in a cyclic service system is not available due to its complexity even without a priority scheme (for example, the two-queue system in [20]). Ibe and Cheng [21], and Boxma and Meister [22] obtained the approximate analysis of asymmetric token passing system (i.e., the message length and message generation interval at each station are



different from each other) with an assumption of the *single service system* in which a station is allowed to transmit only one message whenever it receives the token. An important study of asymmetric single service system is due to Kuehn [23] where the mean waiting time was approximately obtained based on the concept of conditional cycle times. Recently, Boxma and Groenendijk [24,25] obtained an expression for a weighted sum of the mean waiting time at various stations for nonexhaustive service systems. The exact analytical expressions for the individual mean waiting time at each station have not yet been found.

Several simulation studies have been reported for the performance evaluation of the priority scheme in token bus protocols (especially IEEE 802.4 Token Passing Bus Protocol) [26-29], but no rigorous analytical study has been proposed so far. Although the simulation technique can be used to analyze the priority scheme, the performance analysis using a simulation model may be extremely time consuming, and does not provide an exhaustive means for arriving at a conclusion. One simulation run generates the network performance of only one operational condition, and there are so many different operational conditions. Due to the stochastic property of the network system, several independent simulation runs are needed with different seed numbers of the random number generator for each set of input parameters. This procedure often proves to be cumbersome. For this reason, simulation techniques are generally not adequate for optimizing network parameters when a large number of alternative system design options are available.

The objective of this research is to formulate an analytical model for evaluating the performance characteristics of the priority scheme in token bus protocols.

The analysis builds upon the concept of the Kuehn's model [23], and provides a direct relationship between the network parameters (i.e., number of stations in the network, message arrival rate, message length, priority timer, etc.) and the network performance measures (i.e., average queueing delay, average data latency, average queue length, etc.)

One of the major requirements of ICCS network design is that the data latency at each station must be bounded. The maximum allowable data latency at each station is determined by the station's functional characteristics. The analytical model developed in this thesis can be used as a design tool for ICCS networks. As the first step in ICCS network design, design criteria which optimize the Token Rotation Timer (*TRT*) parameters for a given traffic condition have been introduced.

## 1.5. Organization of the Thesis

The thesis is composed of seven chapters and two appendices. Description of token bus protocol and its features are given in Chapter 2. Development of the analytical model is given in Chapter 3. A Petri net model for the token bus protocol is developed in Chapter 4. The simulation program is developed on the basis of the Petri net model. The analytical model is validated by comparison with simulation experiments in Chapter 5. Chapter 6 discusses an approach for designing ICCS networks using the analytical model. Chapter 7 presents the summary, conclusions and recommendations for future work. Based on the Kuehn's model, the expected value of waiting time under the priority scheme is derived in

Appendix A. An approach for optimization of  $TRT$  parameters for ICCS networks is described in Appendix B.

## Chapter 2

### DESCRIPTION OF TOKEN BUS NETWORK PROTOCOL

The token bus network is considered to be suitable for Integrated Communication and Control Systems (ICCS) networks due to bounded data latency, high throughput, high reliability and flexibility [4,5]. In Section 2.1, the characteristics of token bus protocol and its basic features are discussed. Section 2.2 describes the priority scheme in the token bus protocol.

#### 2.1. Token Bus Protocol

The token bus protocol consists of a set of stations connected by a broadcast transmission medium. A message transmission at each station is heard by all other stations. The token bus protocol establishes and maintains a logical ring of stations independent of their physical locations as shown in Figure 2.1. Each station in the logical ring maintains the addresses of its predecessor (i.e., the station which logically precedes this station) and its successor (i.e., the station which logically succeeds it).

The token is passed in a round robin fashion from the lowest logical address to the highest logical address and back to the lowest. Access of a station to the medium is controlled by the token, which is explicitly passed from the station holding the token to its successor. A station receiving the token gains the right to use the medium. If the station receiving the token has any waiting message(s), it transmits its message(s) for a predetermined maximum amount of time, deter-

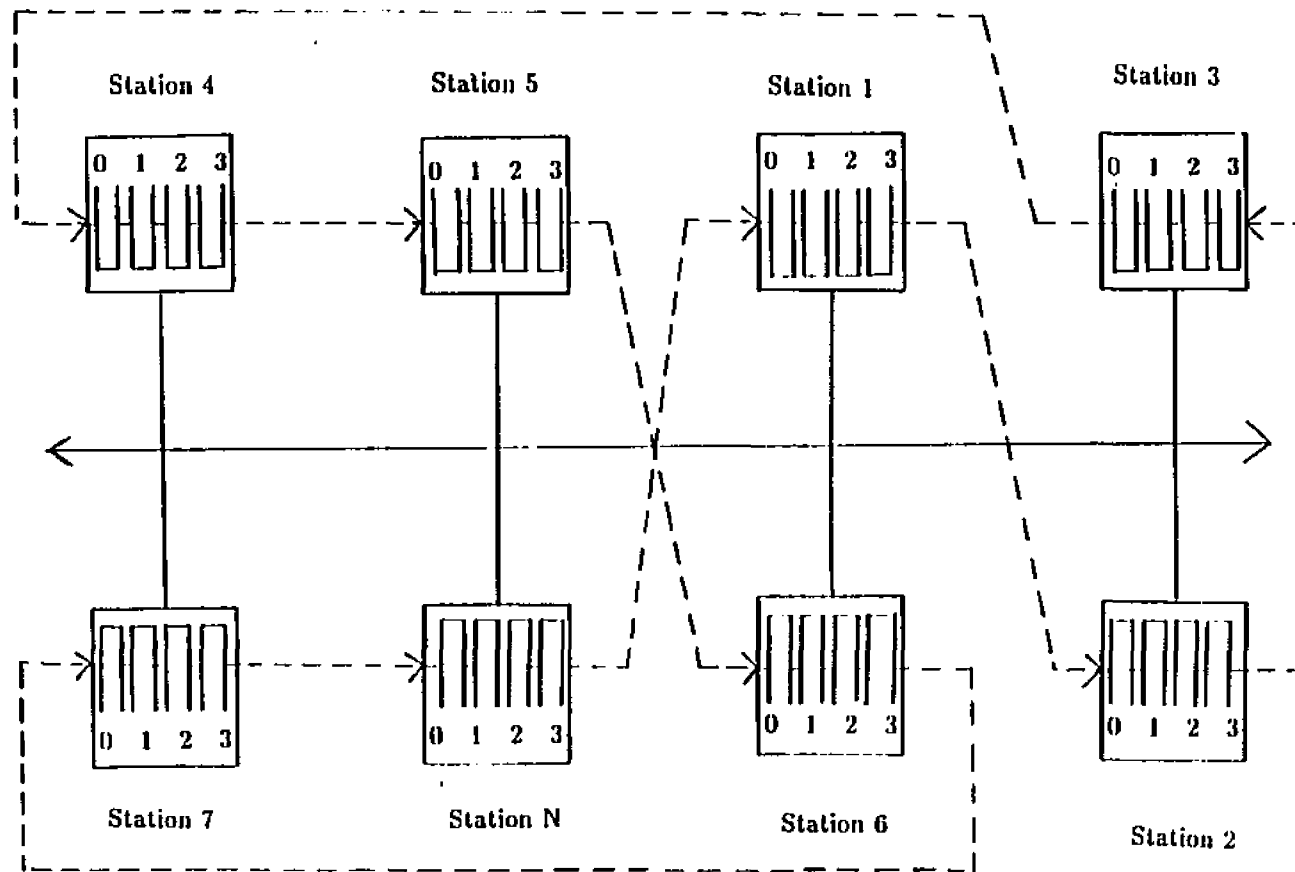


Figure 2.1: Schematic Diagram of Token Bus Protocol; — Physical Bus, - - - Logical Ring.

mined by the Token Holding Timer (*THT*). When this timer has expired or the station has sent all of its messages, the token is forwarded to its successor. If the *THT* expires during message transmission, the station finishes the ongoing transmission of a message and then passes the token to its successor. If the station receiving the token does not have waiting message, the token is immediately forwarded to its successor.

Each station checks if the token is successfully passed immediately after passing the token. During normal operations, a station passes the token to its current successor and then monitors the bus. If a station fails to see a valid token being passed by its successor within a predetermined time interval, called Token Passing Time (*TPT*), it retransmits the token to its successor on the assumption that the successor did not receive the token in the first attempt. If, after two attempts, the successor station still does not respond to the token pass, it is bypassed and the station increments the successor address by one and begins to hunt for a new successor. The bypassed station, although deleted from the logical ring, will continue to be polled for reinsertion during the periodic *station insertion* cycles. This scheme provides a high level of fault tolerance and rapid failure recovery.

Without disrupting network operations, the logical ring is reconfigured every time a station removes itself from the ring. Periodically, stations perform the node admission procedure which allows new stations an opportunity to be admitted into the logical ring. Stations which are just powered up or stations which may have dropped off of the bus due to momentary power failure or other anomaly are allowed to rejoin the network.

The steady-state network operation consists of waiting for an addressed token, transmitting data upon receipt of the token, passing the token to a successor, and monitoring the token pass. Access to the channel during steady-state network operation is achieved in a deterministic manner, thus guaranteeing an upper bound to network access time. By using a bidirectional bus system, all transmissions are truly broadcast. This simplifies the tasks of detecting and recovering from network failures. A detailed description of the token bus protocol is given in [6,10].

## 2.2. Priority Scheme in the Token Bus Protocol

Another highlight of the token bus protocol is its support of a message-based priority mechanism. The priority of each message is assigned when the Logical Link Control (LLC) sublayer requests the Medium Access Control (MAC) sublayer to send a data frame. The MAC sublayer offers four levels of priority classes. In the SAE linear token bus protocol, the priority classes are named 0, 1, 2 and 3, with 0 corresponding to the highest priority and 3 to the lowest. In the IEEE 802.4 token-passing bus protocol, the priority classes are called 0, 2, 4 and 6, with 6 corresponding to the highest priority, and 0 to the lowest. Each station can have maximum of four separate priority queues. As shown in Figure 2.1, each priority queue acts as a virtual substation in that token is passed internally from the highest priority queue to the lowest, through all priority queues before being passed to successor.

With a priority scheme, data transmission access on the bus is controlled through a system involving a Token Holding Timer (*THT*) and three Token Ro-

tation Timers (*TRT*). During normal operations, each station resets its *THT* to the (user programmed) maximum value when the token is received. This value is the maximum allowable amount of time for which the highest priority queue may use the transmission medium during a cycle of token passing. *THT* prevents the highest priority queue from monopolizing the network. After *THT* is reset and restarted, the station immediately resumes transmitting the highest priority message(s).

After the *THT* is expired or if there is no more highest priority messages, the station starts serving the queue of next priority class. The three *TRT*s are reset and restarted when the corresponding priority queue begins serving, and the remaining *TRT* timer values, prior to reset, are stored in the registers. The stored values are used to bound the amount of time available for message transmission at the lower priority levels, i.e., if the token returns to the lower priority queue prior to the *TRT* expiration, the waiting messages in that queue are transmitted until the remaining *TRT* is expired. If the *TRT* has expired by the time the token returns, the corresponding priority queue is not allowed to send any message.

During normal bus operations, situation may arise when a message traffic deferral is requested due to *TRT* expiration. In the event of a heavy traffic, a station may receive the token and, upon checking his *TRT*'s find that one or more of the *TRT*'s in the lower priority queue has expired. In this case, the station must defer the lower priority message (usually, non-real-time data) transmission corresponding to those expired timers and transmit only the higher priority messages (usually, real-time data) with unexpired timers. This procedure allows only



higher priority messages to be transmitted under high traffic and the data latency for the higher priority message can be bounded to a desired value. The allocation of channel capacity to various priority levels is controlled by the *TRT*'s. The responsibility of setting these values lies with the station management which is usually resident above the link layer in the protocol hierarchy.

### Chapter 3

## DEVELOPMENT OF AN ANALYTICAL MODEL FOR THE PRIORITY SCHEME OF TOKEN BUS PROTOCOL

The priority scheme investigated in this thesis is very similar to that specified by the standards of SAE token bus protocol [6] and IEEE 802.4 token bus protocol [10]. The SAE and IEEE token bus protocols have four priority levels. However, the number of priority levels can be increased by setting different values of Token Rotation Timer ( $TRT$ ). The model developed in this thesis is assumed to have  $(K + 1)$  priority levels, i.e., each station has  $(K + 1)$  queues for each priority level. The priority classes are designated as 0, 1, 2, ... ,  $K$ , with 0 corresponding to the highest priority and  $K$  to the lowest.

The analysis in this thesis is limited to the practical case of single-service discipline, i.e., serving exactly one message string at a time. For the highest priority queue, i.e., priority 0, the opportunity to transmit a waiting message is given whenever the token arrives. For the lower priority queues, i.e., priority 1 to  $K$ , the opportunity is given if the corresponding Token Rotation Timer ( $TRT$ ) is not expired when the token arrives after circulating through the logical ring of all active stations in the network. The  $TRT_i$  is reset and restarted whenever the token arrives at a priority  $i$  ( $i=1$  to  $K$ ) queue. The higher priority queue has larger  $TRT_i$  value so that the probability of  $TRT_i$  expiration becomes smaller, and the probability of message transmission becomes larger.

As mentioned in Section 2.2, the objective of Token Holding Timer ( $THT$ ) is

to prevent any priority 0 queue from monopolizing the network. Since the analysis is based on a single-service discipline, and a message is completely transmitted even if  $THT$  is expired during the transmission, the value of  $THT$  does not affect the performance of the analytical model developed in this thesis.

The ICCS network consists of several different kinds of stations which perform diverse and largely independent functions. On this basis, the message arrival process at each individual queue is assumed to have a Poisson distribution. Furthermore, since the ICCS network is decomposed into several single-function subsystems, and the messages generated from the single-function subsystems are usually packetized to a fixed length, the message generated from each queue is assumed to have a constant length. In this analysis, the case of messages being rejected due to queue saturation is not considered. Thus, the queue capacity is assumed to be infinite. Also, it is assumed that messages (at all stations) belonging to the same priority class have identical (average) message arrival rate and message length.

Although the token is passed from one station to another, it is convenient to consider all priority queues in a station as separate sub-stations where the token is passed from one sub-station to another. Thus, the appropriate model is a system of multiple of different priority queues attended by a single server in a cyclic order. Figure 3.1 shows a schematic diagram of the priority scheme with four priority levels. The key notations used in this analysis are listed in page viii.

The performance of the priority scheme is dependent on stochastic variables, such as token circulation time and effective service time, that are directly related

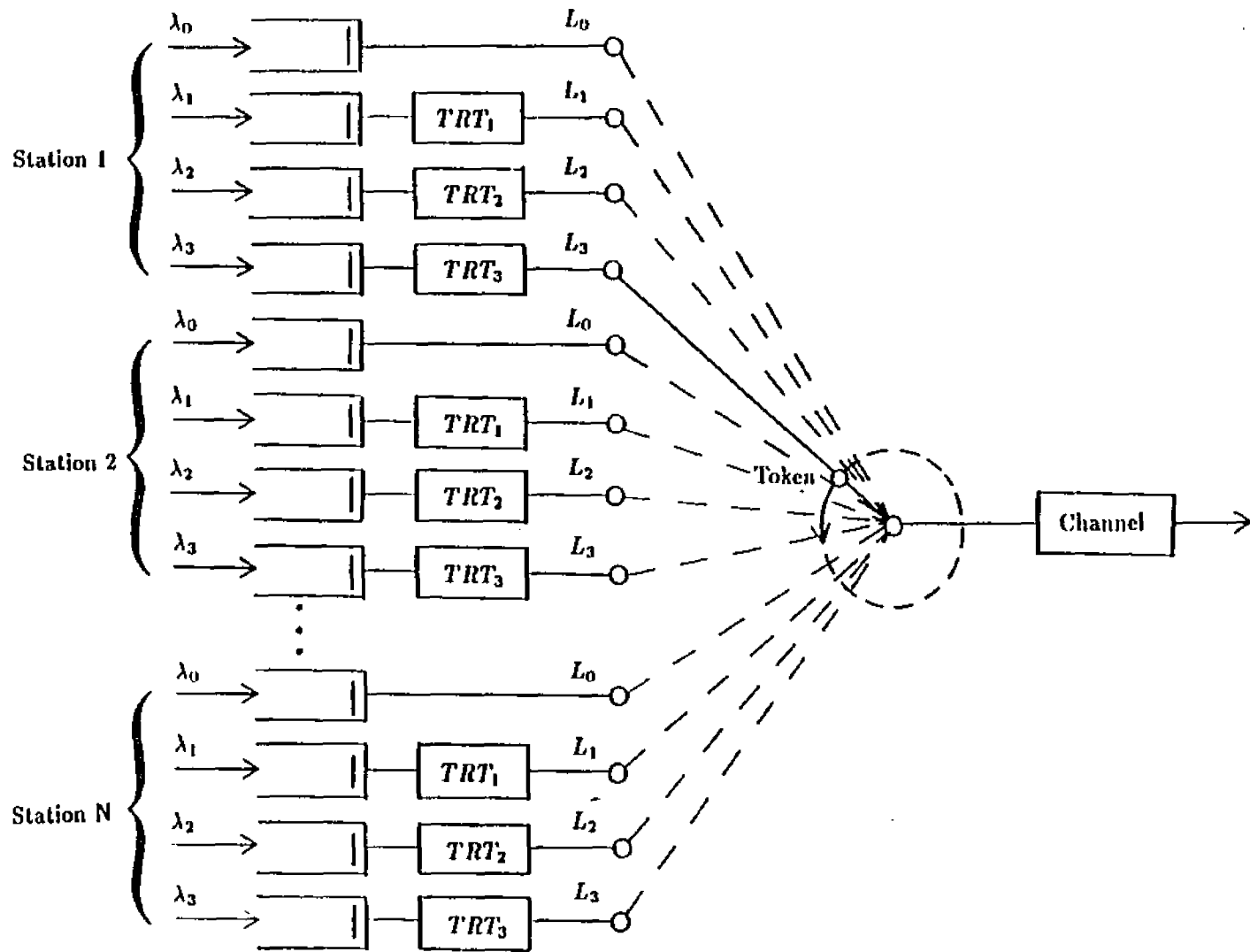


Figure 3.1: Model for the Analysis of Token Bus Protocol with Four Levels of Priority.

to network traffic. Pertinent variables that are to be used in the analytical model are defined below.

**Definition 3.1:** Token Circulation Time, which is denoted as  $T_r$ , is the elapsed time between two consecutive instants at which a transmitter queue captures the token. ■

**Remark 3.1:**  $T_r$  is a random variable and the expected value of  $T_r$  is denoted as  $\overline{T_r}$ , which is identical relative to any queue. ■

**Definition 3.2:** Effective Service Time at a priority  $i$  queue, which is denoted as  $T_i$ , is the time interval from the instant a priority  $i$  queue has an opportunity to transmit a message till the next instant the same queue has an opportunity to transmit a message. ■

**Remark 3.2:** For priority 0 queue,  $T_0 = T_r$  because priority 0 queue transmits a waiting message whenever the token arrives regardless of the setting of  $TRT$ . ■

**Remark 3.3:** For priority  $i$  ( $i=1$  to  $K$ ) queue, an opportunity to transmit a message is given if the corresponding  $TRT_i$  is not expired at the instant of token arrival. ■

**Remark 3.4:**  $T_i$  is a random variable and the expected value of  $T_i$  is denoted as  $\overline{T_i}$ , which is identical relative to any queue belonging to priority  $i$  class. ■

The statistical analysis in the development of the model is not exact. For an exact analysis, the state of the network system at a time  $t$  has to be defined such that all past history is summarized in it so that the system states can be completely predicted. In the present case, the system state could be described by a vector  $[n_1(t), n_2(t), \dots, n_N(t), c_1(t), c_2(t), \dots, c_M(t), L(t), A(t)]$ , where

$n_j(t)$  denotes the number of waiting messages in queue  $j, j = 1, 2, \dots, N$ , and  $N$  is the total number of queues in the network,  $c_j(t)$  represents the remainder of  $TRT$  after it is reset and restarted in queue  $j, j = 1, 2, \dots, I$ , in which  $I$  is the total number of lower priority (priority 1 to  $K$ ) queues,  $L(t)$  points to the present location of the token within a cycle, and  $A(t)$  specifies the *age* of the current service (or propagation) phase of the token. Since there are too many transitions of states even for a small number of queues in the network, exact analysis is mathematically intractable. Eisenberg [20] derived a mathematical model of the system without a priority scheme which has only two queues that are served alternatively. In his analysis, even without a priority scheme, some steps in the solution still have not been proven due to the difficult nature of analysis.

In a cyclic queueing system, several queues share a single server, i.e., token, to transmit their messages. Therefore, the state of a queue is influenced by the states of other queues. However, it is very difficult, if not impossible, to mathematically describe an exact relationship of the processes among the queues in a single-server network system [23]. Because of the mathematical intractability of most cyclic queueing problems, several approximate methods were suggested [23,30]. Fundamental assumption for the analysis is that the message waiting process at each queue is ergodic with probability one. The approximate methods usually rely on certain simplifying assumptions such as the *independence assumption* under which the processes within a particular queue are more or less independent of the processes within the other queues. The analytical model developed in this thesis is also based on the independence assumption.

Token circulation time  $T_r$  depends on whether a message from queue  $j$  is served or not during a circulation. From this observation, Kuehn [23] considered conditional token circulation times for the analysis of asymmetric single-service system without a priority scheme. Since Kuehn did not consider a priority scheme, every queue is allowed to transmit a waiting message whenever the token arrives. Thus, queue  $j$  has exactly one opportunity to transmit during one  $T_r$  (note that  $T_r$  is the effective service time for all queues if the priority scheme is not considered).

The concept of conditional token circulation time reduces the effect of independence assumption which was proposed by Hashida and Ohara [30]. However, Kuehn's approach is still approximate since the conditional token circulation times are assumed to be independent and identically distributed (i.i.d.) variables; the expression for their probability density functions are only approximation, too. Under this assumption, the mean waiting time of a queue  $j$ ,  $\overline{W}_j$ , in an asymmetric single service system (without a priority scheme) was determined by Kuehn [23] as a function of the first two moments of the conditional token circulation times.

A lower priority queue (priority 1 to  $K$ ) can transmit its message only if the corresponding  $TRT$  is not expired at the instant of token arrival (see Remark 3.3). Analysis of the effective service time under a priority scheme is more complex than that without a priority scheme because the effect of  $TRT$  expiration on the message waiting process at priority 1 to  $K$  queues has to be taken into account.

Similar to the Kuehn's model, token circulation time  $T_r$  as observed at a specific priority  $j$  queue is classified under the two conditions of whether the queue transmits or not.

**Definition 3.3:** The token circulation time  $T_{rj}$  for a given priority  $j$  queue is denoted as  $T'_{rj}$  if a message is not served from the queue during the token circulation; otherwise,  $T_{rj}$  is denoted as  $T''_{rj}$ . ■

Effective service time  $T_j$  is also classified under two similar conditions.

**Definition 3.4:** The effective service time  $T_j$  for a given priority  $j$  queue is denoted as  $T'_j$  if a message is not served from the queue; otherwise,  $T_j$  is denoted as  $T''_j$ . ■

In the priority scheme, priority  $i$  queue has exactly one opportunity to transmit during one  $T_i$ . By replacing the conditional token circulation times in Kuehn's formula into the conditional effective service times for priority  $i$  class, the queueing delay of a priority  $i$  queue,  $\overline{W}_i$ , is expressed as

$$\overline{W}_i = \frac{\overline{T_i'^2}}{2\overline{T_i'}} + \frac{\lambda_i \overline{T_i''^2}}{2(1 - \lambda_i \overline{T_i''})} \quad (3.1)$$

where  $\lambda_i$  is the average message arrival rate at priority  $i$  queue, and  $\overline{T_i'}$ ,  $\overline{T_i''}$ ,  $\overline{T_i'^2}$  and  $\overline{T_i''^2}$  are the first and second moments of  $T_i'$  and  $T_i''$ . Using the procedure given in [23], the detailed derivation of equation (3.1) is presented in Appendix A.

To obtain  $\overline{W}_i$ , the first two moments of conditional effective service times should be determined. In Section 3.1., the conditional token circulation times of a network under the priority scheme have been analyzed. Based on the results of Section 3.1., the conditional effective service times of each priority class are determined in Section 3.2. The performance analysis of the priority scheme in token bus protocols is presented in Section 3.3.



### 3.1. Analysis of Conditional Token Circulation Time

This section describes the analysis of the conditional token circulation times of a queue which belongs to the priority  $j$  class. As mentioned earlier, analysis of the exact relationship of the processes among all the queues in the network is mathematically intractable. However, the influence of all the other queues on the process in a priority  $j$  queue can be expressed by the conditional token circulation times  $T'_{rj}$  and  $T''_{rj}$ .

Let  $\mu_i$  be the probabilities that  $TRT_i$  is not expired at the instant when the token arrives at a priority  $i$  queue, i.e.,

$$\mu_i = \Pr[T_r \leq TRT_i], \quad i = 1, \dots, K \quad (3.2)$$

$\mu_0 = 1$  because priority 0 messages are transmitted regardless of the  $TRT$  value.

From Definition 3.3, the corresponding conditional probabilities can be stated as:

$$\mu'_{ij} = \Pr[T'_{rj} \leq TRT_i] \quad (3.3)$$

$$\mu''_{ij} = \Pr[T''_{rj} \leq TRT_i] \quad (3.4)$$

The following analysis is based on the independence assumption, i.e., the processes within a queue are independent of the processes within the other queues. Let  $P_i$  be the probability that the token serves a message at the instant when it arrives at a priority  $i$  queue.

According to Kuehn [23], the probability  $\alpha_0$  that the token meets at least one message when it arrives at a priority 0 queue is

$$\alpha_0 = \lambda_0 \overline{T_r} \quad (3.5)$$

For a priority 0 queue, if the token finds a waiting message at the instant of arrival at this priority 0 queue, it immediately serves the waiting message at the same token arrival instant. Thus, the probability  $P_0$  that the token serves a message at the instant when it arrives at a priority 0 queue equals to the probability that the token finds a waiting message at the instant of arrival at priority 0 queue, i.e.,

$$P_0 = \lambda_0 \overline{T_r} \quad (3.6)$$

The token can serve a lower priority (priority 1 to  $K$ ) message only if at least one message is waiting and the corresponding  $TRT_i$  is not expired at the instant when it arrives at a lower priority queue. From this observation,  $P_i$  is obtained using the assumption that, at the instant of token arrival, the message waiting process at a priority  $i$  queue and  $TRT_i$  expiration process at the same queue are independent of each other. However, the message waiting process and  $TRT_i$  expiration process at the same queue could be mutually dependent only when the token serves a message from the queue during a particular token circulation (or, message transmission at the queue contributes to the increment of token circulation time  $T_r$ ). If the traffic is light, this effect becomes negligible. Based on this assumption, the probability  $P_i$  that the token serves a message at the instant when it arrives at a priority  $i$  ( $i=1$  to  $K$ ) queue is determined in the following way.

The probability  $\alpha_i$  that the token meets at least one message at the instant when it arrives at a priority  $i$  queue is given in [23] as

$$\alpha_i = \lambda_i \overline{T_r} \quad (3.7)$$

When the token arrives at a lower priority (priority 1 to  $K$ ) queue, it may not be able to serve a waiting message at the instant of token visit due to  $TRT_i$  expiration, i.e., the token may have to visit more than once to serve a waiting message at a lower priority queue.

Let  $\omega_i$  be an epoch that the token have a chance to serve a message from a priority  $i$  queue, i.e.,  $TRT_i$  is not expired when the token arrives at a priority  $i$  queue. As shown in Figure 3.2, during each epoch, the token may experience  $TRT_i$  expiration  $(n - 1)$  times, where  $n = 1$  to  $\infty$ . (note that, for priority 0 class,  $\omega_0$  is every instants of token arrival at a priority 0 queue because the token can serve a priority 0 message whenever it arrives at a priority 0 queue regardless of  $TRT$ ).

Suppose that, during the time interval between  $\omega_i$ , shown in Figure 3.2, the token experiences  $TRT_i$  expiration at the first to  $(n - 1) - th$  visits and does not experience  $TRT_i$  expiration at the  $n - th$  visit. Then the token serves the message at the  $n - th$  visit. The message which is served at the  $n - th$  token visit may have arrived at the first token visit, or at the second token visit, ... , or at the  $n - th$  token visit. Therefore, the probability  $\phi_i^n$  that a message at a priority  $i$  queue is served at the  $n - th$  visit under the condition that the token experiences  $TRT_i$  expiration  $(n - 1)$  times becomes

$$\phi_i^n = \sum_{j=0}^{n-1} \alpha_i (1 - \mu_i)^j \mu_i = \alpha_i [1 - (1 - \mu_i)^n] \quad (3.8)$$

The probability of the event that the token undergoes  $(n - 1)$  consecutive expirations of  $TRT_i$  and does not experience  $TRT_i$  expiration at the  $n - th$  visit

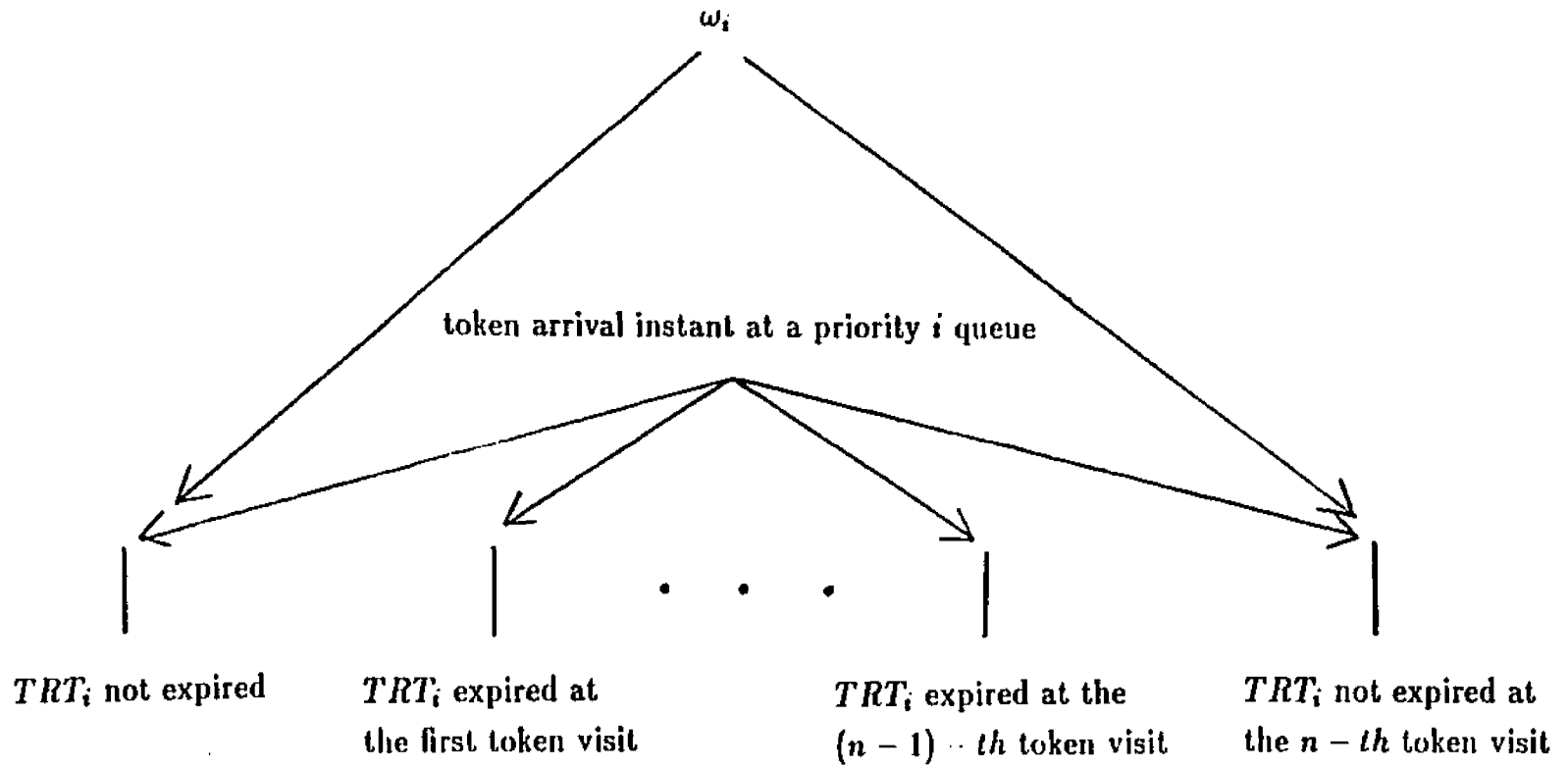


Figure 3.2: An Epoch of Message Transmission Opportunity at a Priority  $i$  Queue.

is

$$\eta_i^n = (1 - \mu_i)^{n-1} \mu_i \quad (3.9)$$

Considering an equivalent probability that the token serves a message at a priority  $i$  queue, the probability  $P_i$  is obtained by averaging  $\phi_i^n$  with a weighted term  $\eta_i^n$ , i.e.,

$$\begin{aligned} P_i &= \sum_{n=1}^{\infty} \phi_i^n \eta_i^n \\ &= \sum_{n=1}^{\infty} \alpha_i [1 - (1 - \mu_i)^n] (1 - \mu_i)^{n-1} \mu_i \\ &= \frac{\lambda_i \overline{T_r} \mu_i}{1 - (1 - \mu_i)^2} \end{aligned} \quad (3.10)$$

**Remark 3.5:**  $P_i$  is a monotonically increasing function of both  $\mu_i$  and  $\lambda_i$ . Specifically, if  $\mu_i = 1$  ( $i=1$  to  $K$ ), the probability  $P_i$  is identical to the case without a priority scheme, i.e.,  $P_i = \lambda_i \overline{T_r}$ . ■

**Remark 3.6:** This approximation is based on the independence assumption, i.e., processes within a particular queue is independent of the processes within the other queues. In fact, the state at a queue (i.e., number of waiting messages, remainder of  $TRT_i$  value, current location of the token, etc.) is dependent upon the state of all the other queues. As mentioned earlier, exact analysis on this basis is not feasible. ■

If the token circulation time is  $T'_{rj}$ , the conditional probability  $P'_{ij}$  is obtained by replacing  $\mu_i$  and  $T_r$  in expression for  $P_i$  by  $\mu'_{ij}$  and  $T'_{rj}$  in (3.3), respectively.

$$P'_{ij} = \frac{\lambda_i \overline{T'_{rj}} \mu'_{ij}}{1 - (1 - \mu'_{ij})^2} \quad (3.11)$$

Similarly  $P''_{ij}$  is obtained by replacing  $\mu_i$  and  $T_r$  in expression for  $P_i$  by  $\mu''_{ij}$  and  $T''_{rj}$  in (3.4), respectively.

$$P''_{ij} = \frac{\lambda_i \overline{T''_{rj}} \mu''_{ij}}{1 - (1 - \mu''_{ij})^2} \quad (3.12)$$

Let  $M_{kj}$  be the number of the priority  $k$  queues transmitting their messages for a given token circulation time during which one priority  $j$  queue does not transmit its message. Since the processes at each queues are assumed to be independent of each other,  $M_{kj}$  is a binomial random variable. The binomial distribution of  $M_{kj}$  is obtained as follows.

$$\Pr[M_{kj} = \theta] = \binom{N_k - \delta_{kj}}{\theta} P_{kj}^\theta (1 - P_{kj})^{N_k - \delta_{kj} - \theta}, \quad (3.13)$$

$$0 \leq \theta \leq N_k - \delta_{kj}.$$

where  $N_k$  is the number of priority  $k$  queue in the network,  $\delta_{kj}$  is the Kronecker's delta, i.e.,  $\delta_{jj} = 1$ ;  $\delta_{kj} = 0$  for  $k \neq j$ , and  $P_{kj}$  denotes the probability that a message is served at the instant when the token arrives at a priority  $k$  queue during a given token circulation time, i.e., if a given token circulation time is  $T'_{rj}$  or  $T''_{rj}$ , then  $P_{kj}$  is equal to  $P'_{kj}$  or  $P''_{kj}$ , respectively.

For the binomial random variable  $M_{kj}$ , the conditional token circulation times  $T'_{rj}$  and  $T''_{rj}$  are expressed as

$$T'_{rj} = \sum_{k=0}^K M_{kj} L_k + R \quad (3.14)$$

$$T''_{rj} = \sum_{k=0}^K M_{kj} L_k + R + L_j \quad (3.15)$$

where  $j$  and  $k$  denote the priority levels,  $L_k$  points a message transmission time at a priority  $k$  queue, and  $R$  is the total idle time during one token circulation, which includes the sum of token holding delay at each station and the sum of the total propagation delay of the token during one circulation around the logical ring.

The analysis for the derivation of  $T''_{rj}$  is identical to that of  $T'_{rj}$  except that  $P''_{kj}$  and  $R+L_j$  are replaced by  $P'_{kj}$  and  $R$ , respectively. Therefore, in the following derivation, only  $T'_{rj}$  is considered.

The moment generation function of  $M_{kj}L_k$  is

$$\Phi_{kj}(s) = \sum_{\theta=0}^{N_k - \delta_{kj}} e^{\theta L_k s} \Pr[M_{kj} = \theta] \quad (3.16)$$

From the independence assumption, the moment generation function of  $T'_{rj}$  in (3.14) becomes

$$\begin{aligned} \Phi_{T'_{rj}}(s) &= e^{Rs} \prod_{k=0}^K \Phi_{kj}(s) \\ &= \underbrace{\sum_{a=0}^{N_0 - \delta_{0j}} \dots \sum_{b=0}^{N_K - \delta_{Kj}}}_{K+1} \underbrace{\pi'_{0j} \dots \pi'_{Kj}}_{K+1} e^{\underbrace{(L_0 a + \dots + L_K b + R)s}_{K+1}} \end{aligned} \quad (3.17)$$

where

$$\pi'_{kj} = \Pr[M_{kj} = \theta] = \binom{N_k - \delta_{kj}}{\theta} P'_{kj}{}^\theta (1 - P'_{kj})^{N_k - \delta_{kj} - \theta} \quad (3.18)$$

From  $\Phi_{T'_{rj}}(s)$  in (3.17), the probability density function (pdf) of  $T'_{rj}$  can be obtained by using the inverse Laplace-Stieltjes Transformation [31] as

$$f_{T'_{rj}}(t) = \underbrace{\sum_{a=0}^{N_0 - \delta_{0j}} \dots \sum_{b=0}^{N_K - \delta_{Kj}}}_{K+1} \underbrace{\pi'_{0j} \dots \pi'_{Kj}}_{K+1} U_0[t - \underbrace{(L_0 a + \dots + L_K b + R)}_{K+1}]. \quad (3.19)$$

where  $U_0[\bullet]$  is the unit impulse function.

**Remark 3.7:** When the token circulation time is  $T''_{rj}$ , the probability density function of  $T''_{rj}$  is determined by replacing  $P'_{kj}$  and  $R$  in (3.19) to  $P''_{kj}$  and  $R + L_j$ , respectively, i.e.,

$$f_{T''_{rj}}(t) = \underbrace{\sum_{a=0}^{N_0 - \delta_{0j}} \cdots \sum_{b=0}^{N_K - \delta_{Kj}}}_{K+1} \underbrace{\pi''_{0j} \cdots \pi''_{Kj}}_{K+1} U_0[t - (\underbrace{L_0 a + \cdots + L_K b}_{K+1} + R + L_j)]. \quad (3.20)$$

where

$$\pi''_{kj} = \Pr[M_{kj} = \theta] = \binom{N_k - \delta_{kj}}{\theta} P''_{kj}{}^\theta (1 - P''_{kj})^{N_k - \delta_{kj} - \theta} \quad (3.21)$$

■

From (3.13), the moment generation function of  $M_{kj}L_k$  given in (3.16) reduces to

$$\Phi_{kj}(s) = [1 - P'_{kj}(1 - e^{L_k s})]^{N_k - \delta_{kj}}. \quad (3.22)$$

Using the moment generation functions in (3.22), the moment generation function for  $T'_{rj}$  given in (3.17) is equivalently expressed as

$$\begin{aligned} \Phi_{T'_{rj}}(s) &= e^{Rs} \prod_{k=0}^K \Phi_{kj}(s) \\ &= e^{Rs} \prod_{k=0}^K [1 - P'_{kj}(1 - e^{L_k s})]^{N_k - \delta_{kj}} \end{aligned} \quad (3.23)$$

From (3.23), the mean token circulation time  $\overline{T'_{rj}}$  is

$$\begin{aligned} \overline{T'_{rj}} &= \left. \frac{d}{ds} \Phi_{T'_{rj}}(s) \right|_{s=0} \\ &= \sum_{k=0}^K P'_{kj} N_k L_k - P'_{jj} L_j + R \end{aligned} \quad (3.24)$$



Similarly, the mean token circulation time  $\overline{T''_{rj}}$  is

$$\begin{aligned}\overline{T''_{rj}} &= \frac{d}{ds} \Phi_{T''_{rj}}(s) \Big|_{s=0} \\ &= \sum_{k=0}^K P''_{kj} N_k L_k - P''_{jj} L_j + R + L_j\end{aligned}\quad (3.25)$$

Substituting  $P'_{kj}$  in (3.11) to (3.24),  $\overline{T'_{rj}}$  is determined as

$$\overline{T'_{rj}} = \frac{R}{1 - S'_j} \quad (3.26)$$

where

$$S'_j = \sum_{k=0}^K \frac{\lambda_k \mu'_{kj} L_k N_k}{1 - (1 - \mu'_{kj})^2} - \frac{\lambda_j \mu'_{jj} L_j}{1 - (1 - \mu'_{jj})^2} \quad (3.27)$$

Similarly,

$$\overline{T''_{rj}} = \frac{R + L_j}{1 - S''_j} \quad (3.28)$$

where

$$S''_j = \sum_{k=0}^K \frac{\lambda_k \mu''_{kj} L_k N_k}{1 - (1 - \mu''_{kj})^2} - \frac{\lambda_j \mu''_{jj} L_j}{1 - (1 - \mu''_{jj})^2} \quad (3.29)$$

**Remark 3.8:** The average values of  $T'_{rj}$  and  $T''_{rj}$  express the influence of message arrival process ( $\lambda_k$ ) and  $TRT$  expiration process ( $\mu'_{kj}$ ) of all the other queues ( $k = 0$  to  $K$ ) on the process of a queue which belongs to the priority  $j$  class. ■

Transposing (3.26) and (3.28) into (3.11) and (3.12),  $P'_{kj}$  and  $P''_{kj}$  are determined as

$$P'_{kj} = \frac{\{\lambda_j \mu'_{kj} / [1 - (1 - \mu'_{kj})^2]\} R}{1 - S'_j}, \quad (3.30)$$

and,

$$P''_{kj} = \frac{\{\lambda_j \mu''_{kj} / [1 - (1 - \mu''_{kj})^2]\} (R + L_j)}{1 - S''_j}. \quad (3.31)$$

The second moment of  $T'_{rj}$  is

$$\overline{T'^2_{rj}} = \frac{d^2}{ds^2} \Phi_{T'_{rj}}(s) \Big|_{s=0} \quad (3.32)$$

The variance of  $T'_{rj}$  is

$$\begin{aligned} \sigma_{T'_{rj}}^2 &= \overline{T'^2_{rj}} - \overline{T'_{rj}}^2 \\ &= \sum_{k=0}^K P'_{kj}(1 - P'_{kj})N_k L_k^2 - P'_{jj}(1 - P'_{jj})L_j^2. \end{aligned} \quad (3.33)$$

Similarly, the variance of  $T''_{rj}$  is

$$\begin{aligned} \sigma_{T''_{rj}}^2 &= \overline{T''^2_{rj}} - \overline{T''_{rj}}^2 \\ &= \sum_{k=0}^K P''_{kj}(1 - P''_{kj})N_k L_k^2 - P''_{jj}(1 - P''_{jj})L_j^2. \end{aligned} \quad (3.34)$$

**Remark 3.9:** The variances of  $T'_{rj}$  and  $T''_{rj}$  express the influence of message arrival process ( $\lambda_k$ ) and  $TRT$  expiration process ( $\mu'_{kj}$ ) of all the other queues ( $k = 0$  to  $K$ ) on the process of a queue which belongs to the priority  $j$  class. ■

Since  $T_0 = T_r$  (see Remark 3.2), from (3.26) and (3.28), the average conditional effective service times for a priority 0 queue,  $\overline{T'_0}$  and  $\overline{T''_0}$ , is determined as,

$$\overline{T'_0} = \frac{R}{1 - S'_0} \quad (3.35)$$

and

$$\overline{T''_0} = \frac{R + L_0}{1 - S''_0} \quad (3.36)$$

From (3.33) and (3.34), the second moment of conditional effective service times for a priority 0 queue,  $\overline{T'^2_0}$  and  $\overline{T''^2_0}$ , are also determined as

$$\overline{T'^2_0} = \sigma_{T'_{r_0}}^2 + \overline{T'_0}^2 \quad (3.37)$$

and,

$$\overline{T_0''^2} = \sigma_{T_0''}^2 + \overline{T_0''}^2 \quad (3.38)$$

For  $i = 1$  to  $K$ ,  $\mu'_{ij}$  is defined in (3.3) as

$$\mu'_{ij} = \text{Pr}[T'_{rj} \leq TRT_i] = F_{T'_{rj}}(TRT_i) \quad (3.39)$$

where  $F_{T'_{rj}}(TRT_i)$  is a probability distribution function of  $T'_{rj}$ . By integrating  $f_{T'_{rj}}(t)$  in (3.19) from  $t = 0$  to  $TRT_i$ ,  $\mu'_{ij}$  can be determined as

$$\begin{aligned} \mu'_{ij} &= \int_0^{TRT_i} f_{T'_{rj}}(t) dt \\ &= \underbrace{\sum_{a=0}^{N_0 - \delta_{0j}} \dots \sum_{b=0}^{N_K - \delta_{Kj}}}_{K+1} \underbrace{\pi'_{0j} \dots \pi'_{Kj}}_{K+1} U_1[TRT_i - \underbrace{(L_0 a + \dots + L_K b + R)}_{K+1}] \end{aligned} \quad (3.40)$$

where  $U_1[\bullet]$  is the unit step function and  $\delta_{kj}$  is the Kronecker's delta.

For a given  $j$ , by replacing  $P'_{kj}$  in (3.30) into (3.18) and substituting  $\pi'_{kj}$  ( $k=0$  to  $K$ ) into (3.40),  $\mu'_{ij}$  in (3.40) is expressed as  $K$  non-linear simultaneous equations for  $\mu'_{1j}$  to  $\mu'_{Kj}$ . By repeating this procedure from  $j=0$  to  $K$ , total  $K(K+1)$  unknowns of  $\mu'_{ij}$  ( $i=1$  to  $K$ , and  $j=0$  to  $K$ ) can be determined. These non-linear simultaneous equations have been solved by using the IMSL subroutine ZSCNT which numerically solves a set of non-linear equations. However, any other appropriate techniques can be used to solve these non-linear simultaneous equations.

For  $T''_{rj}$ , by integrating (3.20),  $\mu''_{ij}$  may be expressed as follows:

$$\mu''_{ij} = \underbrace{\sum_{a=0}^{N_0 - \delta_{0j}} \dots \sum_{b=0}^{N_K - \delta_{Kj}}}_{K+1} \underbrace{\pi''_{0j} \dots \pi''_{Kj}}_{K+1} U_1[TRT_i - \underbrace{(L_0 a + \dots + L_K b + R + L_j)}_{K+1}]. \quad (3.41)$$

The  $\mu''_{ij}$  in (3.41) is obtained on the assumption that the conditional token circulation time  $T''_{rj}$  is an independent and identically distributed (i.i.d.) random variable. In this case, when  $TRT_i < R + L_j$ ,  $\mu''_{ij}$  always becomes zero, i.e., priority  $i$  queues become unstable.

In a real system, token circulation time with respect to a particular queue does not consist of two independent and identically distributed random variables of  $T'_{rj}$  and  $T''_{rj}$ , but successive token circulation times are dependent of each other. For a more exact analysis, covariance of token circulation time with respect to each queue should be considered. The analysis of covariance of token circulation time is extremely complex because the state of each token circulation time with respect to a particular queue depends on the states of all the queues at that token circulation. As mentioned earlier, the analysis of a relationship of the states among all the queues at each token circulation is not mathematically tractable.

To take account of the stability of priority  $i$  queues during a particular token circulation  $T''_{rj}$ , an approximate approach is considered for the determination of  $\mu''_{ij}$ . It is assumed that, during a particular token circulation of  $T''_{rj}$ , all the queues belonging to priority  $j$  class transmit their messages with a probability of  $P''_{jj}$  at the instant of token arrival. Using this assumption, the  $\mu''_{ij}$  given in (3.41) is modified by replacing  $R + L_j$  and  $N_j - 1$  by  $R$  and  $N_j$ , respectively, i.e.,

$$\mu''_{ij} \approx \underbrace{\sum_{a=0}^{N_0} \dots \sum_{b=0}^{N_K}}_{K+1} \underbrace{\pi''_{0j} \dots \pi''_{Kj}}_{K+1} U_1[TRT_i - \underbrace{(L_0 a + \dots + L_K b + R)}_{K+1}]. \quad (3.42)$$

The approximation of  $\mu''_{ij}$  in (3.42) considers the stability of priority  $i$  queues. To account for a message service from one priority  $j$  queue during  $T''_{rj}$ , the prob-

ability  $\mu_j''$  that  $TRT_j$  is not expired when the token arrives at a priority  $j$  queue during  $T_{rj}''$  is approximated as follows. During a particular token circulation of  $T_{rj}''$ ,  $TRT_j$  of one priority  $j$  queue is not expired with a probability of one, while the  $TRT_j$  of the rest  $(N_j - 1)$  priority  $j$  queues is not expired with a probability of  $\mu_{jj}''$  which is obtained from (3.42). Thus,  $\mu_j''$  is obtained as

$$\mu_j'' \approx \mu_{jj}''(N_j - 1)/N_j + 1/N_j \quad (3.43)$$

These approximations will be examined by comparing with the simulation results (cf. Chapter 5).

### 3.2. Analysis of Conditional Effective Service Time

In this section, first the probability density functions (pdf) of  $T_i'$  and  $T_i''$  (see Definition 3.4),  $f_{T_i'}$  and  $f_{T_i''}$ , are determined. Then, the first and second moments of conditional effective service times of the priority  $i$  class are obtained from their respective probability density functions.

Let  $\mu_i'$  be the probability that  $TRT_i$  is not expired when the token arrives at a priority  $i$  queue during  $T_{ri}'$ . Since the  $TRT_i$  expiration process at all queues belonging to the priority  $i$  class is identical, i.e.,  $T_{ri}'$  and  $TRT_i$  are same for all priority  $i$  queues,  $\mu_{ii}'$  determined from (3.40) equals to  $\mu_i'$ . The analysis of  $T_i''$  follows that of  $T_i'$  with a replacement of  $T_{rj}'$  and  $\mu_i'$  by  $T_{rj}''$  and  $\mu_i''$ , respectively. The probability distribution function (PDF) of  $T_{ri}'$  is given as follows.

$$\begin{aligned} \Pr[T_{ri}' \leq TRT_i] &= F_{T_{ri}'}(TRT_i) \\ &= \sum_{j=1}^{u_i} p_j U_1[TRT_i - t_j], \end{aligned} \quad (3.44)$$

$$\begin{aligned}
\Pr[T'_{ri} > TRT_i] &= 1 - F_{T'_{ri}}(TRT_i) \\
&= \sum_{k=1}^{v_i} q_k U_1[t_k - TRT_i],
\end{aligned} \tag{3.45}$$

where  $U_1[\bullet]$  is the unit step function.  $u_i$  denotes the number of token circulations when  $T'_{ri}$  does not exceeds  $TRT_i$  and  $v_i$  denotes the number of token circulations when  $T'_{ri}$  exceeds  $TRT_i$ . Correspondingly,  $p_j$  and  $q_k$  are the probabilities that the token circulation time  $T'_{ri}$  equals  $t_j$  and  $t_k$ , respectively.

**Remark 3.10:** The numbers  $u_i$  and  $v_i$  in (3.44) and (3.45) depends on the distribution of token circulation time and the value of  $TRT_i$ . ■

**Lemma 3.1:** The probability  $\mu'_i$  that  $TRT_i$  is not expired when the token arrives at a priority  $i$  queue is

$$\mu'_i = \sum_{j=1}^{u_i} p_j \tag{3.46}$$

and the probability  $1 - \mu'_i$  that  $TRT_i$  is expired when the token arrives at a priority  $i$  queue is

$$1 - \mu'_i = \sum_{k=1}^{v_i} q_k \tag{3.47}$$

**Proof:** From (3.44), for  $T'_{rj} \leq TRT_i$ ,

$$\mu'_i = \Pr[T'_{ri} \leq TRT_i] = \sum_{j=1}^{u_i} p_j$$

and from (3.45), for  $T'_{rj} > TRT_i$ ,

$$1 - \mu'_i = \Pr[T'_{ri} > TRT_i] = \sum_{k=1}^{v_i} q_k$$

■

The probability density functions (pdf) of (3.44) and (3.45) are given as follows

$$f_1(t) = \sum_{j=1}^{u_i} p_j U_0[t - t_j], \quad 0 \leq t \leq TRT_i, \quad (3.48)$$

$$f_2(t) = \sum_{k=1}^{v_i} q_k U_0[t - t_k], \quad t > TRT_i \quad (3.49)$$

where  $U_0[\bullet]$  is the unit impulse function.

Let a priority  $i$  queue have a chance to transmit a message (i.e.,  $TRT_i$  is not expired when the token arrives) at a time instant  $t = t_0$ . The token comes back to the same queue at  $t = t_1$ . If  $T'_{ri} = t_1 - t_0 < TRT_i$  (i.e.,  $TRT_i$  is not expired), the effective service time  $T'_i$  for this queue equals to  $T'_{ri}$  at this instant. The probability that priority  $i$  queue has a chance to transmit without experiencing  $TRT_i$  expiration becomes

$$\Pr[T'_i = T'_r] = \Pr[T'_{ri} \leq TRT_i] \quad (3.50)$$

where  $T'_r$  denotes that  $TRT_i$  is not expired during that  $T'_{ri}$ . The pdf of  $T'_i$  for this case is

$$f_{T'_i}^{(1)}(t) = f_1(t) = \sum_{j=1}^{u_i} p_j U_0[t - t_j], \quad 0 \leq t_j \leq TRT_i, \quad (3.51)$$

Figure 3.3 illustrates an example of  $f_{T'_i}^{(1)}(t)$

Let a priority  $i$  queue have a chance to transmit a message at a time instant  $t = t_0$ . If a priority  $i$  queue misses a chance to transmit at a token arrival when  $t = t_1$  because  $TRT_i$  is expired, and has the chance on the next token arrival at  $t = t_2$ , the effective service time  $T'_i$  for this queue is equal to  $t_2 - t_0 = T'_r + T'_a$ , where  $T'_r$  denotes  $TRT_i$  is expired during that  $T'_{ri}$ . The probability that priority

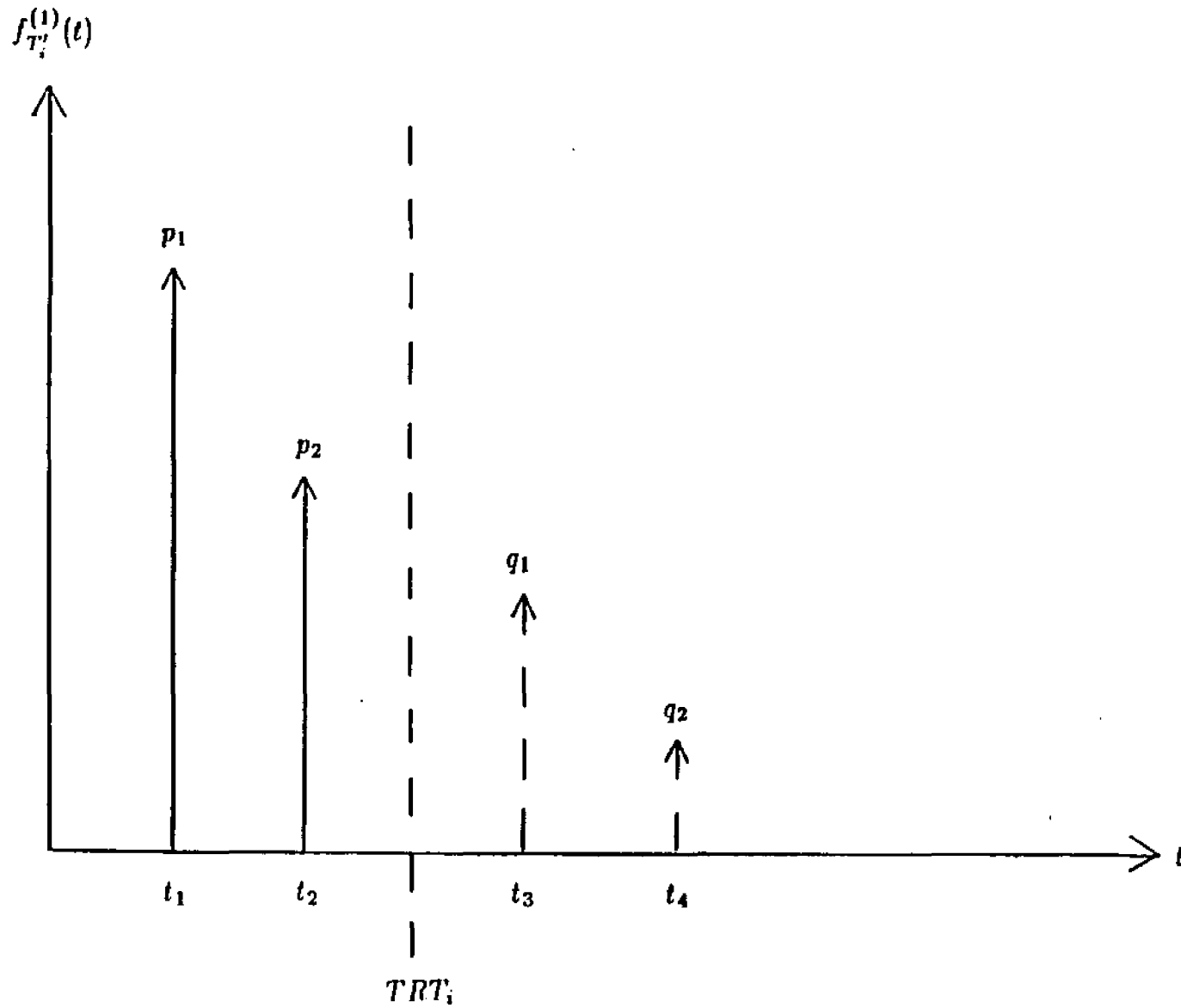


Figure 3.3: Probability Density Function of  $T'_i$  without Experiencing  $TRT_i$  Expiration.



$i$  queue has a chance to transmit after  $TRT_i$  is expired once becomes

$$\Pr[T'_i = T_r^e + T_r^a] = \Pr[T'_{ri} > TRT_i] \Pr[T'_{ri} \leq TRT_i] \quad (3.52)$$

since the two events are independent by the initial postulation.

The pdf of  $T'_i$  for this case is

$$f_{T'_i}^{(2)}(t) = f_2(t) f_1(t) \quad (3.53)$$

which is equivalent to

$$f_{T'_i}^{(2)}(t) = \sum_{j=1}^{u_i} \sum_{k=1}^{v_i} q_k p_j U_0[t - (t_k + t_j)]. \quad (3.54)$$

where

$$0 \leq t_j \leq TRT_i, \quad t_k > TRT_i$$

Figure 3.4 illustrates an example of  $f_{T'_i}^{(2)}(t)$ . This figure shows that the probability that  $T'_i = t_3 + t_1$  is the product of the probability  $q_1$  that token circulation time becomes  $t_3$  ( $TRT_i$  is expired) at the first instant of token arrival and the probability  $p_1$  that token circulation time becomes  $t_1$  ( $TRT_i$  is not expired) at the next instant of token arrival.

In general, the probability that priority  $i$  queue has a chance to transmit after  $TRT_i$  is expired  $(n - 1)$  times is

$$\begin{aligned} \Pr[T'_i = \underbrace{T_r^e + \dots + T_r^e}_{(n-1)} + T_r^a] \\ = \underbrace{\Pr[T'_{ri} > TRT_i] \dots \Pr[T'_{ri} > TRT_i]}_{(n-1) \text{ times}} \Pr[T'_{ri} \leq TRT_i]. \end{aligned} \quad (3.55)$$

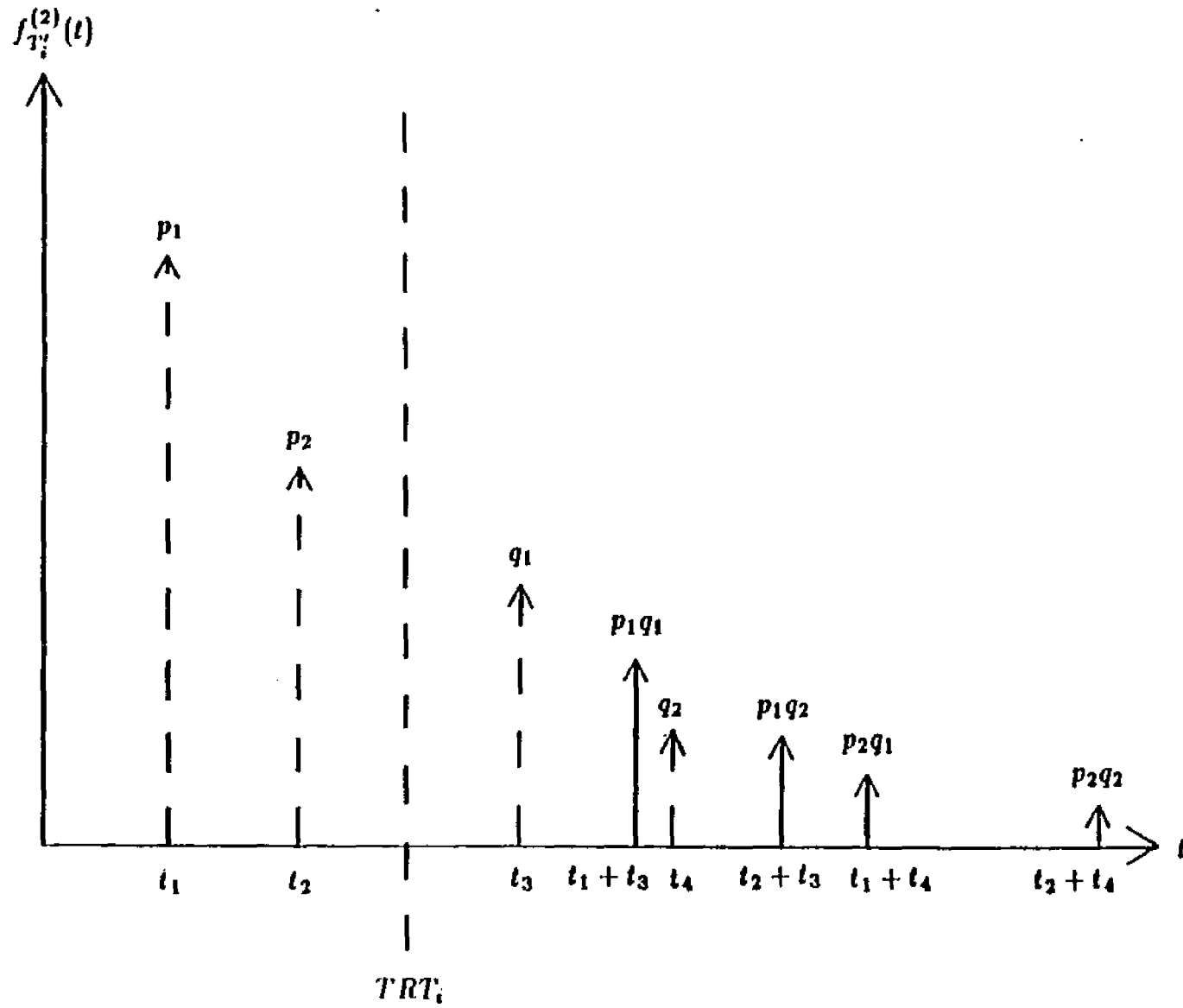


Figure 3.4: Probability Density Function of  $T'_i$  with Experiencing  $TRT_i$  Expiration Once.

The pdf of  $T'_i$  for this case is

$$f_{T'_i}^{(n)}(t) = \underbrace{f_2(t) \dots f_2(t)}_{(n-1)\text{ times}} f_1(t) \quad (3.56)$$

which is equivalent to

$$f_{T'_i}^{(n)}(t) = \sum_{j=1}^{u_i} \underbrace{\sum_{k=1}^{v_i} \dots \sum_{l=1}^{v_i}}_{(n-1)} \underbrace{q_k \dots q_l}_{(n-1)} p_j U_0[t - (t_j + \underbrace{t_k + \dots + t_l}_{(n-1)})]. \quad (3.57)$$

in which

$$0 \leq t_j \leq TRT_i, \underbrace{t_k > TRT_i, \dots, t_l > TRT_i}_{(n-1)}.$$

Finally, the pdf of  $T'_i$  becomes

$$f_{T'_i}(t) = \sum_{n=1}^{\infty} f_{T'_i}^{(n)}(t) = \sum_{j=1}^{\infty} Z_j U_0[t - t_j] \quad (3.58)$$

where  $Z_j$  denotes the probability that  $T'_i$  equals to  $t_j$ , i.e.,  $Z_j = \Pr[T'_i = t_j]$ .

From (3.58), the moment generation function for  $T'_i$  is

$$\begin{aligned} \Phi_{T'_i}(s) &= \int_0^{\infty} f_{T'_i}(t) e^{st} dt \\ &= \sum_{j=1}^{\infty} Z_j \int_0^{\infty} e^{st} U_0[t - t_j] dt \\ &= \sum_{j=1}^{\infty} Z_j e^{st_j}. \end{aligned} \quad (3.59)$$

The average of  $T'_i$  is

$$\begin{aligned} \overline{T'_i} &= \left. \frac{d}{ds} \Phi_{T'_i}(s) \right|_{s=0} \\ &= \sum_{j=1}^{\infty} t_j Z_j \\ &= \sum_{j=1}^{\infty} t_j \Pr[T'_i = t_j]. \end{aligned} \quad (3.60)$$

The next proposition gives the convergence of  $\overline{T'_i}$ .

**Proposition 3.1:**  $\overline{T'_i}$  converges to

$$\overline{T'_i} = \frac{\overline{T'_{ri}}}{\mu'_i} \quad (3.61)$$

**Proof:** Let  $\overline{T'_{ra}}$  denote the average value of  $T'_{ri}$  during which  $TRT_i$  at a given priority  $i$  queue is not expired, and  $\overline{T'_{re}}$  represent the average value of  $T'_{ri}$  during which  $TRT_i$  at a given priority  $i$  queue is expired. Following (3.48) and (3.49),  $\overline{T'_{ra}}$  and  $\overline{T'_{re}}$  are expressed as

$$\overline{T'_{ra}} = \sum_{j=1}^{u_i} p_j t_j \quad (3.62)$$

$$\overline{T'_{re}} = \sum_{k=1}^{v_i} q_k t_k \quad (3.63)$$

$\overline{T'_i}$  given in (3.60) is equivalently expressed in view of (3.58) as

$$\overline{T'_i} = \sum_{n=1}^{\infty} \overline{T'^{(n)}_i}$$

where  $\overline{T'^{(n)}_i}$  is determined from (3.57) as

$$\begin{aligned} \overline{T'^{(n)}_i} &= \sum_{j=1}^{u_i} \underbrace{\sum_{k=1}^{v_i} \dots \sum_{l=1}^{v_i} \sum_{m=1}^{v_i}}_{(n-1)} \underbrace{q_k \dots q_l q_m}_{(n-1)} p_j (t_j + \underbrace{t_k + \dots + t_l + t_m}_{(n-1)}) \\ &= \sum_{j=1}^{u_i} p_j t_j \underbrace{\sum_{k=1}^{v_i} q_k \dots \sum_{l=1}^{v_i} q_l \sum_{m=1}^{v_i} q_m}_{(n-1)} \end{aligned}$$

$$(n-1) \left\{ \begin{array}{l} + \sum_{k=1}^{v_i} q_k t_k \sum_{j=1}^{u_i} p_j \underbrace{\sum_{l=1}^{v_i} q_l \cdots \sum_{m=1}^{v_i} q_m}_{(n-2)} \\ \cdot \\ \cdot \\ + \sum_{m=1}^{v_i} q_m t_m \sum_{j=1}^{u_i} p_j \underbrace{\sum_{k=1}^{v_i} q_k \cdots \sum_{l=1}^{v_i} q_l}_{(n-2)} \end{array} \right.$$

Using (3.46), (3.47), (3.62) and (3.63), the above equation yields

$$\overline{T_i^{(n)}} = \overline{T_r^a} (1 - \mu_i')^{n-1} + (n-1) \overline{T_r^e} \mu_i' (1 - \mu_i')^{n-2}$$

Since  $\mu_i' \leq 1$ , equation (3.61) is justified as

$$\begin{aligned} \overline{T_i'} &= \sum_{n=1}^{\infty} \overline{T_i^{(n)}} \\ &= \overline{T_r^a} \sum_{n=1}^{\infty} (1 - \mu_i')^{n-1} + \overline{T_r^e} \mu_i' \sum_{n=1}^{\infty} (n-1) (1 - \mu_i')^{n-2} \\ &= \frac{\overline{T_r^a} + \overline{T_r^e}}{\mu_i'} \\ &= \frac{\overline{T_{ri}'}}{\mu_i'}. \end{aligned}$$

■

**Corrolary to Proposition 3.1:**

$$\overline{T_i''} = \frac{\overline{T_{ri}''}}{\mu_i''} \quad (3.64)$$

**Proof:** The proof follows directly from Proposition 3.1. ■

**Remark 3.11:** If  $\mu_i' = 1$  and  $\mu_i'' = 1$ ,  $\overline{T_i'}$  and  $\overline{T_i''}$  are equal to  $\overline{T_{ri}'}$  and  $\overline{T_{ri}''}$ , respectively. Priority  $i$  queue acts as a priority 0 queue. ■

Using (3.26) and (3.61),  $\overline{T'_i}$  is determined as

$$\overline{T'_i} = \frac{R/\mu'_i}{1 - S'_i} \quad (3.65)$$

Similarly,

$$\overline{T''_i} = \frac{(R + L_i)/\mu''_i}{1 - S''_i} \quad (3.66)$$

From (3.59), the second moment of  $T'_i$  is

$$\begin{aligned} \overline{T'^2_i} &= \left. \frac{d^2}{ds^2} \Phi_{T'_i}(s) \right|_{s=0} \\ &= \sum_{j=1}^{\infty} t_j^2 Z_j \\ &= \sum_{j=1}^{\infty} t_j^2 \Pr[T'_i = t_j]. \end{aligned} \quad (3.67)$$

The next proposition shows the convergence of  $\overline{T'^2_i}$ .

**Proposition 3.2:**  $\overline{T'^2_i}$  converges to

$$\overline{T'^2_i} = \frac{\overline{T'^2_{ri}}}{\mu'_i} + \frac{2\overline{T'_{ri}} \overline{T'^e_r}}{\mu'^2_i} \quad (3.68)$$

where  $\overline{T'^e_r}$  is given in (3.63).

**Proof:**  $\sigma_a$ ,  $\sigma_e$ ,  $\sigma_{ae}$  and  $\sigma_{ee}$  are defined as

$$\sigma_a = \sum_{j=1}^{u_i} p_j t_j^2 \quad (3.69)$$

$$\sigma_e = \sum_{k=1}^{v_i} q_k t_k^2 \quad (3.70)$$

$$\sigma_{ae} = \overline{T'^a_r} \overline{T'^e_r} = \sum_{j=1}^{u_i} p_j t_j \sum_{k=1}^{v_i} q_k t_k \quad (3.71)$$

$$\sigma_{ee} = \overline{T_r^{\prime 2}} = \sum_{k=1}^{v_i} q_k t_k \sum_{l=1}^{v_i} q_l t_l \quad (3.72)$$

Using (3.67),  $\overline{T_i^{\prime 2}}$  is expressed as

$$\overline{T_i^{\prime 2}} = \sum_{n=1}^{\infty} \overline{T_i^{\prime(n)2}}$$

where  $\overline{T_i^{\prime(n)2}}$  is determined from (3.57) as

$$\begin{aligned} \overline{T_i^{\prime(n)2} &= \sum_{j=1}^{u_i} \underbrace{\sum_{k=1}^{v_i} \sum_{l=1}^{v_i} \dots \sum_{m=1}^{v_i} \sum_{r=1}^{v_i}}_{(n-1)} \underbrace{q_k q_l \dots q_m q_r}_{(n-1)} p_j (t_j + t_k + t_l + \dots + t_m + t_r)^2 \\ &= \sum_{j=1}^{u_i} \underbrace{\sum_{k=1}^{v_i} \sum_{l=1}^{v_i} \dots \sum_{m=1}^{v_i} \sum_{r=1}^{v_i}}_{(n-1)} \underbrace{q_k q_l \dots q_m q_r}_{(n-1)} p_j (t_j^2 + t_k^2 + t_l^2 + \dots + t_m^2 + t_r^2 \\ &\quad + \underbrace{t_j t_k + t_j t_l + \dots + t_j t_m + t_j t_r}_{2(n-1)} \\ &\quad + \underbrace{t_k t_l + t_k t_m + \dots + t_k t_r + \dots + t_r t_k + t_r t_l + \dots + t_r t_m}_{(n-2)}) \\ &= \sum_{j=1}^{u_i} p_j t_j^2 \underbrace{\sum_{k=1}^{v_i} q_k \sum_{l=1}^{v_i} q_l \dots \sum_{m=1}^{v_i} q_m \sum_{r=1}^{v_i} q_r}_{(n-1)} \\ &\quad + \sum_{k=1}^{v_i} q_k t_k^2 \underbrace{\sum_{j=1}^{u_i} p_j \sum_{l=1}^{v_i} q_l \dots \sum_{m=1}^{v_i} q_m \sum_{r=1}^{v_i} q_r}_{(n-2)} \\ &\quad \cdot \\ &\quad \cdot \\ &\quad + \sum_{r=1}^{v_i} q_r t_r^2 \underbrace{\sum_{j=1}^{u_i} p_j \sum_{k=1}^{v_i} q_k \sum_{l=1}^{v_i} q_l \dots \sum_{m=1}^{v_i} q_m}_{(n-2)} \end{aligned}$$

(n-1) times

$$\left. \begin{aligned}
 & + \sum_{j=1}^{u_i} p_j t_j \sum_{k=1}^{v_i} q_k t_k \underbrace{\sum_{l=1}^{v_i} q_l \cdots \sum_{m=1}^{v_i} q_m \sum_{r=1}^{v_i} q_r}_{(n-2)} \\
 & \cdot \\
 & + \sum_{j=1}^{u_i} p_j t_j \sum_{r=1}^{v_i} q_r t_r \underbrace{\sum_{k=1}^{v_i} q_k \cdots \sum_{l=1}^{v_i} q_l \sum_{m=1}^{v_i} q_m}_{(n-2)}
 \end{aligned} \right\} 2(n-1) \text{ times}$$

$$\left. \begin{aligned}
 & + \sum_{k=1}^{v_i} q_k t_k \sum_{l=1}^{v_i} q_l t_l \sum_{j=1}^{u_i} p_j \underbrace{\sum_{m=1}^{v_i} q_m \cdots \sum_{r=1}^{v_i} q_r}_{(n-3)} \\
 & \cdot \\
 & + \sum_{m=1}^{v_i} q_m t_m \sum_{r=1}^{v_i} q_r t_r \sum_{j=1}^{u_i} p_j \underbrace{\sum_{k=1}^{v_i} q_k \cdots \sum_{l=1}^{v_i} q_l}_{(n-3)}
 \end{aligned} \right\} (n-1)(n-2) \text{ times}$$

From (3.69) to (3.72), and (3.46) and (3.47) in Lemma 3.1, it follows that

$$\begin{aligned}
 \overline{T_i^{(n)2}} &= \sigma_a (1 - \mu'_i)^{n-1} + (n-1) \sigma_e \mu'_i (1 - \mu'_i)^{n-2} \\
 &\quad + 2(n-1) \sigma_{ae} (1 - \mu'_i)^{n-2} + (n-1)(n-2) \sigma_{ee} \mu'_i (1 - \mu'_i)^{n-3}
 \end{aligned}$$



Since  $\mu'_i \leq 1$ ,

$$\begin{aligned}
\overline{T_i'^2} &= \sum_{n=1}^{\infty} \overline{T_i'^{(n)2}} \\
&= \sigma_a \sum_{n=1}^{\infty} (1 - \mu'_i)^{n-1} + \sigma_e \mu'_i \sum_{n=1}^{\infty} (n-1)(1 - \mu'_i)^{n-2} \\
&\quad + 2\sigma_{ae} \sum_{n=1}^{\infty} (n-1)(1 - \mu'_i)^{n-2} + \sigma_{ee} \mu'_i \sum_{n=1}^{\infty} (n-1)(n-2)(1 - \mu'_i)^{n-3} \\
&= \frac{\overline{T_{ri}'^2}}{\mu'_i} + \frac{2\sigma_{ae}}{\mu_i'^2} + \frac{2\sigma_{ee}}{\mu_i'^2} \\
&= \frac{\overline{T_{ri}'^2}}{\mu'_i} + \frac{2\overline{T_{ri}'} \overline{T_r'^e}}{\mu_i'^2}.
\end{aligned}$$

■

**Corrolary to Proposition 3.2:**

$$\overline{T_i''^2} = \frac{\overline{T_{ri}''^2}}{\mu_i''} + \frac{2\overline{T_{ri}''} \overline{T_r''^e}}{\mu_i''^2}. \quad (3.73)$$

**Proof:** The proof follows directly from Proposition 3.2. ■

### 3.3. Performance Analysis of Priority Scheme

Now the first two moments of conditional effective service times  $\overline{T_i'}$ ,  $\overline{T_i''}$ ,  $\overline{T_i'^2}$  and  $\overline{T_i''^2}$  are determined in (3.61), (3.64), (3.68) and (3.73). The average queuing delay for priority  $i$  queue,  $\overline{W_i}$ , can be determined from (3.1).  $\overline{W_i}$  can be used to determine the average data latency which is one of the most important parameters for evaluation of the network performance. Data latency is defined as follows:

**Definition 3.5:** Data latency is defined as the time interval between the instant of arrival of a message at the transmitter buffer of the source station and the instant when the last bit of the same message reaches the receiver buffer of the destination station. ■

Neglecting propagation delay, the average data latency for priority  $i$  class is

$$\overline{D}_i = \overline{W}_i + L_i \quad (3.74)$$

From the Little's theorem [31], the average queue length, i.e., average number of messages waiting in a priority  $i$  queue is

$$\overline{Q}_i = \lambda_i \overline{W}_i \quad (3.75)$$

The network system is stable if the mean waiting time (or, equivalently, the mean queue length) of any priority message is finite. The following proposition gives the stability conditions for a network system, and each individual priority  $i$  queue.

**Proposition 3.3:** The single-service network system with a priority scheme is stable if

$$\max_i [(\lambda_i/\mu_i'')(R + L_i) + S_i''] < 1, \quad i = 0, \dots, K \quad (3.76)$$

where  $S_i''$  is given in (3.29). The stability condition for the priority  $i$  queue is

$$(\lambda_i/\mu_i'')(R + L_i) + S_i'' < 1. \quad (3.77)$$

**Proof:** From (3.1) the mean waiting time for priority  $i$  queue  $\overline{W}_i$  is finite if

$$\lambda_i \overline{T_{ri}''} = (\lambda_i/\mu_i'')(R + L_i)/(1 - S_i'') < 1 \quad (3.78)$$

From (3.78), (3.76) and (3.77) is evident. ■

Under a stable condition, every message which arrives at the transmitter queue is eventually transmitted. Therefore, throughput is equal to the offered traffic, both of which essentially determine steady-state performance.

The analysis in this chapter is based on the independence assumptions. The analytical results are verified in Chapter 5 by comparison with the results of simulation experiments which do not incur approximations due to the independence assumptions.

## Chapter 4

### PETRI NET MODEL FOR PRIORITY SCHEME AND DEVELOPMENT OF A SIMULATION MODEL

This chapter describes the development of a simulation model for the priority scheme in token bus protocols. First, Petri net model is developed in order to investigate the structure and dynamic behavior of the priority scheme in token bus protocols. Based on the Petri net model, a simulation model is developed using SIMAN. The simulation model measures the network performances such as queueing delay, data latency, throughput and bus utilization for each priority level. Description of the Petri net model is given in Section 4.1. Section 4.2 presents the simulation model.

#### 4.1. Petri Net Model for Priority Scheme

Petri nets are one of the extensively used system modeling techniques which reveal important information about the structure and dynamic behavior of the modeled system. A network system consists of several separate and interacting components which exhibit *concurrency*, i.e., several activities of components in the system may occur simultaneously. The concurrent nature of activities in a system creates difficulties in modeling a system. The activities of the interacting components must be synchronized to assure correctness of information exchange. Petri nets are designed specifically to model systems with interacting concurrent components [32].

Although Petri net models have been used by several authors for the description and analysis of communication network protocols such as packet-switched networks and CSMA/CD network [33-35], the description of the priority scheme in token bus protocols using Petri net has not apparently been reported. In this section, the priority scheme of token bus protocols is modeled using a Timed Petri Net (TPN). TPN is an advanced version for Petri net modeling, which can describe the actions or operations that occur after some finite time or require some finite time to be executed [36].

The TPN model of this thesis is structured to facilitate the development of the discrete-event simulation model of priority scheme. The propagation delay of token or message is extremely short compare to the queueing delay or message transmission time. Thus, the effect of propagation delay on the network performance (especially data latency) is negligible. In this thesis, abnormal operating conditions such as lost token, multiple tokens and babbling station are not considered. Thus, the network system under consideration is assumed to operate under normal condition. According to the standards of SAE and IEEE token bus protocols, the TPN model developed in this thesis is assumed to have four priority levels, i.e., each station has four queues for each priority level. The priority classes are designated as 0, 1, 2 and 3, with 0 corresponding to the highest and 3 to the lowest according to the SAE token bus protocol. To summarize, the TPN model has been developed under the following assumptions.

1. Propagation delay is negligible relative to queueing delay and message transmission time.

2. Network operates under normal condition.
3. Each station has four priority level queues.

The Petri net token is called as P-token to distinguish it from the protocol token. The TPN model of the priority scheme consists of four submodels; channel, station, priority 0 queue and priority  $i$  ( $i= 1, 2$  and  $3$ ) queue.

As shown in Figure 2.1 of chapter 2, token bus protocols consist of a set of stations connected by a broadcast transmission medium. The token is continuously passed around the logical ring via physical bus. Figure 4.1 shows the TPN submodel of channel. This submodel represents token passing along the logical ring through the channel (immediate transitions  $tkn\_pas\_S_j$ ,  $j =1$  to  $N$ ), and message transmission via the channel (places  $cha\_msg\_xmt$ ). It is noted that the channel submodel has a modular structure which comprises  $N$  station submodels.

The TPN submodel for a station is shown in Figure 4.2. The submodel for station  $j$  consists of two parts, separated by the dashed line. The upper part represents the station connection to the channel, which consists of two immediate transitions,  $tkn\_pas\_S_j$ ,  $tkn\_pas\_S_{j+1}$ , and a place  $cha\_msg\_xmt$ . They are identical to those shown in the channel submodel given in Figure 4.1. Neglecting the propagation delay, the immediate transitions  $tkn\_pas\_S_j$  and  $tkn\_pas\_S_{j+1}$  represent the token passing along the logical ring to station  $S_j$  and  $S_{j+1}$ , respectively. The place  $cha\_msg\_xmt$  represents a message transmission from station  $j$  via the channel.

The lower part of a station submodel represents the station behavior. When the token arrives at station  $j$ , P-token is marked in place  $tkn\_arv\_S_j$  which rep-

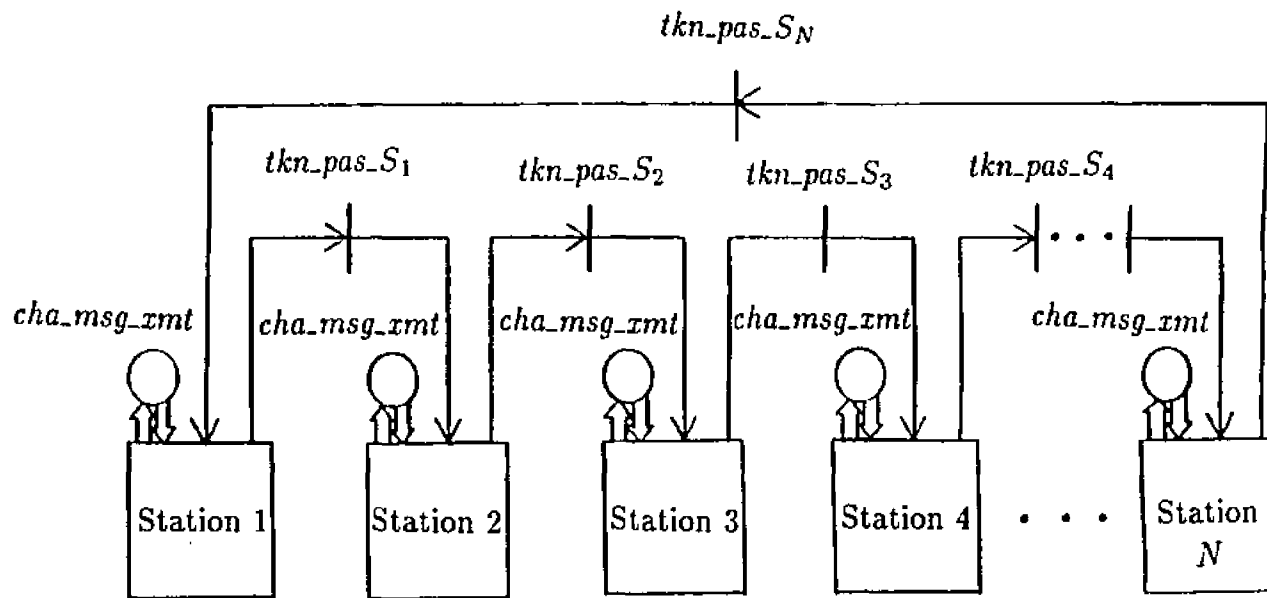


Figure 4.1: Petri Net Submodel for Channel.

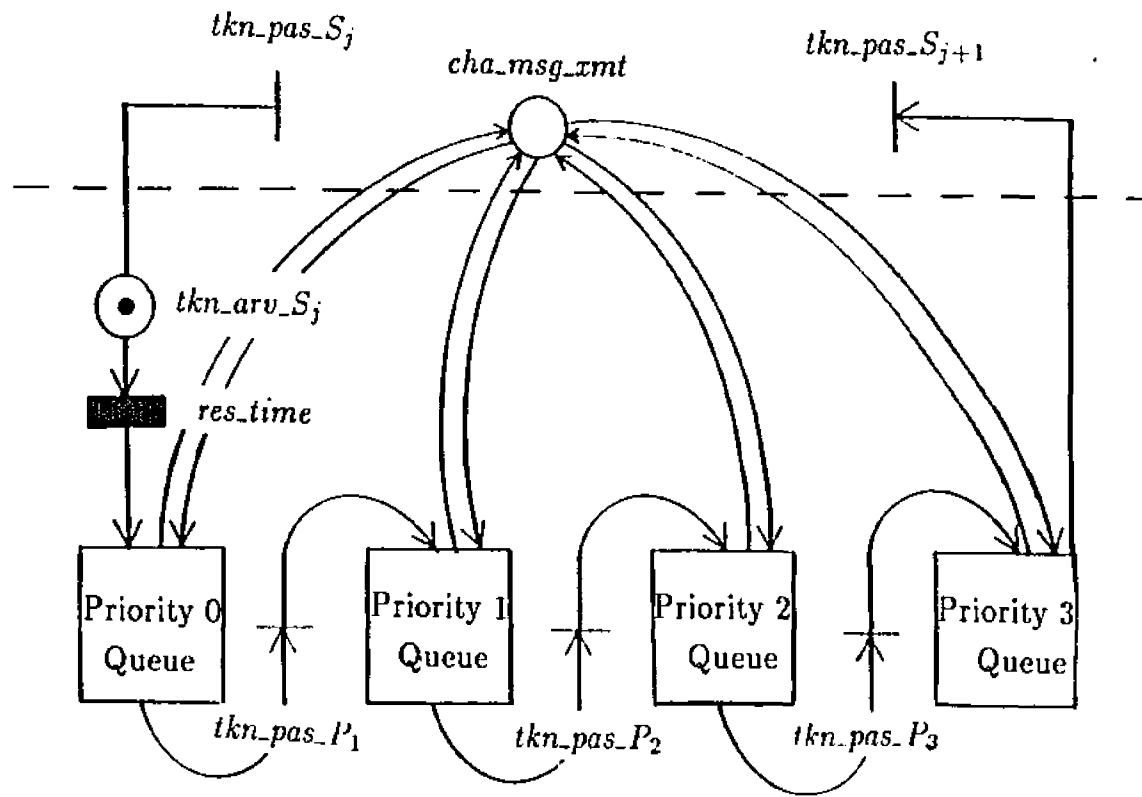


Figure 4.2: Petri Net Submodel for Station  $j$ .



resents the token arrival at station  $j$ , and fires deterministically timed transition  $res\_time$ , which models response time of a station. At the end of the response time, the token begins to serve priority 0 queue. Each station has four queues for priority 0, 1, 2 and 3. Therefore, the station submodel has four queue submodels and three immediate transitions  $tkn\_pas\_P_k$  ( $k=1, 2$  and  $3$ ) which represent token passing to the lower priority queue. Since the behavior of priority 1, 2 and 3 queues is identical, only two queue submodels for priority 0 and priority  $i$  ( $i=1, 2$  and  $3$ ) are described below.

The TPN models for priority 0 and priority  $i$  ( $i=1, 2$  and  $3$ ) queues are presented in Figures 4.3 and 4.4, respectively. The queue submodels consist of two subnets; message generation subnet and protocol subnet, which are divided by the dashed line.

The message generation subnet for priority 0 queue consists of two places,  $P_0\_msg\_gen$  and  $P_0\_queue$ , and an exponentially timed transition,  $P_0\_msg\_arv$ . The place labeled  $P_0\_msg\_gen$ , which is initially marked with one P-token, models a message generation state. The marking of place  $P_0\_msg\_gen$  enables the firing of the exponential transition  $P_0\_msg\_arv$ , and a new message is continuously generated with the Poisson distribution. A newly generated message is queued, which is modeled by a place labeled  $P_0\_queue$ .

The protocol subnet of priority 0 queue consists of three immediate transitions,  $P_0\_que\_emp$ ,  $P_0\_msg\_stt$  and  $tkn\_pas\_P_1$ , two deterministically timed transitions,  $res\_time$  and  $P_0\_msg\_end$ , and three places,  $P_0\_tkn\_cap$ ,  $P_0\_srv\_end$  and  $cha\_msg\_zmt$ , out of which the transitions  $res\_time$  and  $tkn\_pas\_P_1$ , and a place

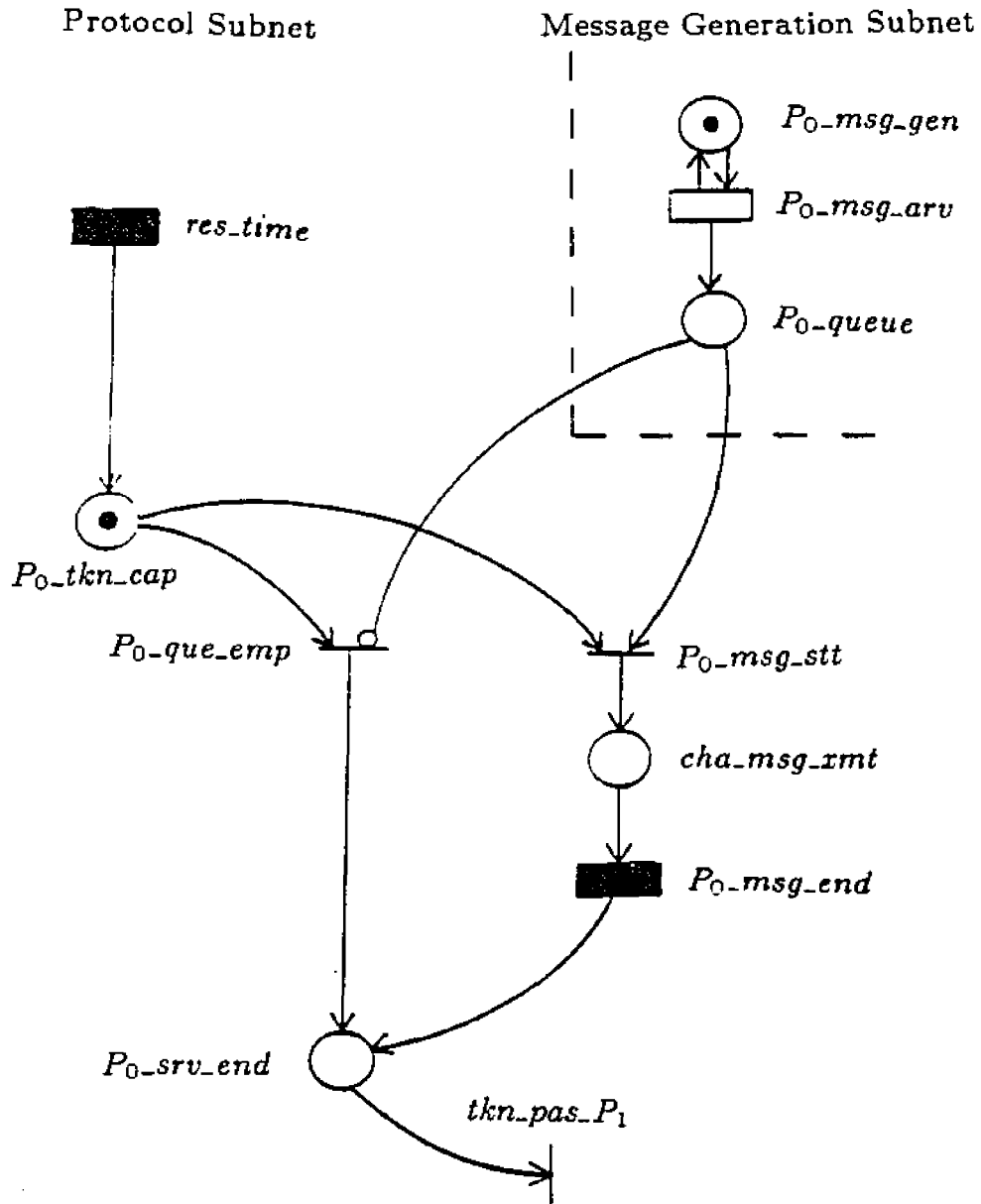


Figure 4.3: Petri Net Submodel for Priority 0 Queue.

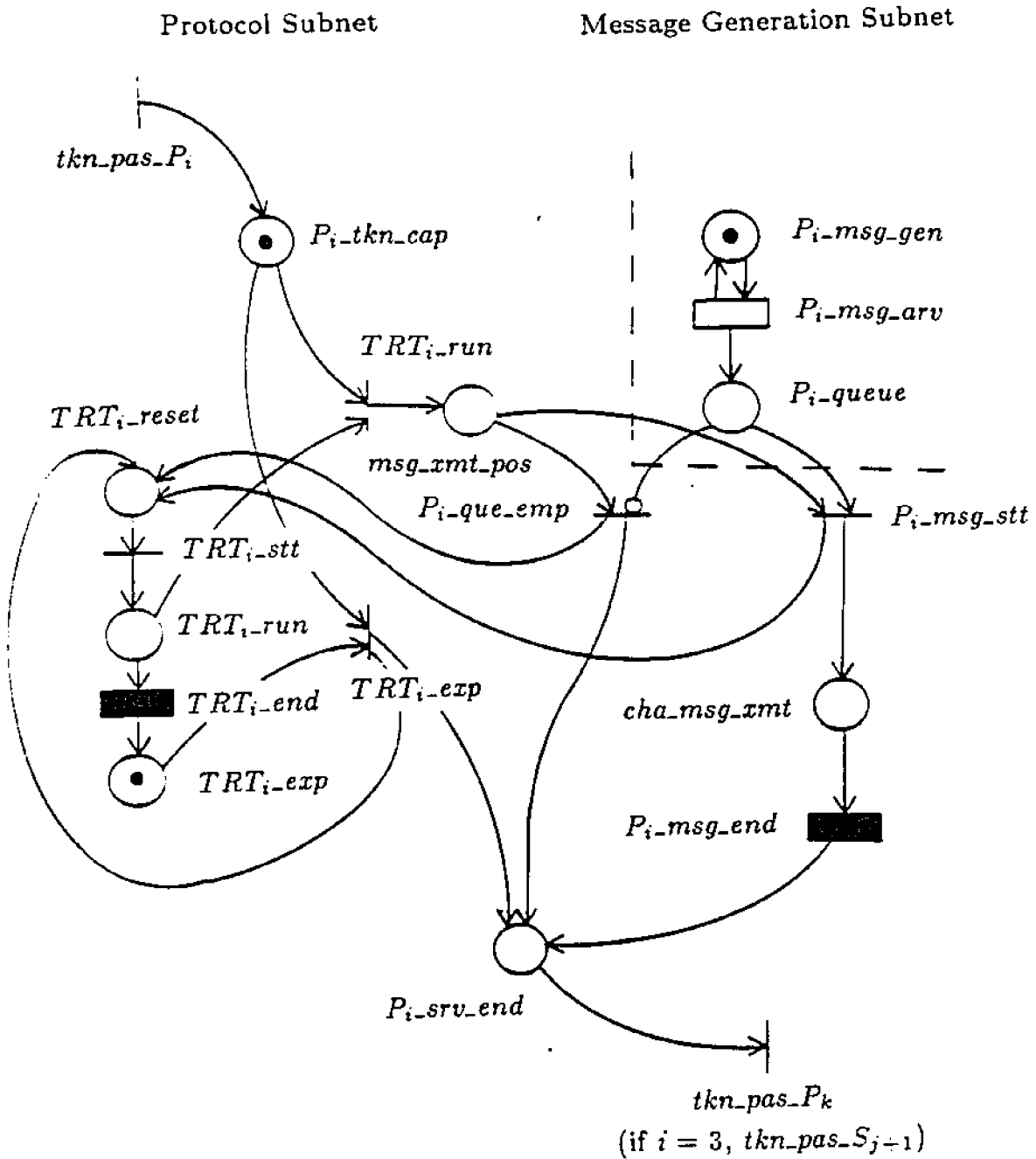


Figure 4.4: Petri Net Submodel for Priority  $i$  Queue.

*cha\_msg\_xmt* are identical to those shown in the station submodel of Figure 4.2.

At the end of the response time, P-token moves into place  $P_0\_tkn\_cap$ , which models token capture at priority 0 queue. At this moment, if there is no P-token in place  $P_0\_queue$ , i.e., if the priority 0 queue is empty, the immediate transition  $P_0\_que\_emp$  is enabled and fires, putting the P-token in place  $P_0\_srv\_end$  that represents end of priority 0 queue service, and fires the immediate transition  $tkn\_pas\_P_1$ , passing the token to priority 1 queue.

However, if the place  $P_0\_queue$  is not empty when P-token is marked in place  $P_0\_tkn\_cap$ , the immediate transition  $P_0\_msg\_stt$  (start of priority 0 message transmission) is enabled and fires, putting the P-token in place *cha\_msg\_xmt* representing a priority 0 message transmission through the channel. At the end of message transmission period, deterministically timed transition  $P_0\_msg\_end$  (end of priority 0 message transmission) moves P-token to  $P_0\_srv\_end$ , and the service of priority 0 queue is finished.

The TPN model for priority  $i$  ( $i=1, 2$  and  $3$ ) queue which is given in Figure 4.4 also has message generation subnet and protocol subnet. The message generation subnet consists of two places,  $P_i\_msg\_gen$  and  $P_i\_queue$ , and an exponentially timed transition,  $P_i\_msg\_arv$ . The behavior of message generation subnet for priority  $i$  queue is exactly same as that for priority 0 queue, which is described above.

The protocol subnet of priority  $i$  queue consists of seven immediate transitions,  $tkn\_pas\_P_i$ ,  $TRT_i\_stt$ ,  $TRT_i\_run$ ,  $TRT_i\_exp$ ,  $P_i\_que\_emp$ ,  $P_i\_msg\_stt$  and  $tkn\_pas\_P_k$  (or,  $tkn\_pas\_S_{j+1}$ ), two deterministically timed transitions,  $TRT_i\_end$

and  $P_i\_msg\_end$ , and eight places,  $P_i\_tkn\_cap$ ,  $TRT_i\_reset$ ,  $TRT_i\_run$ ,  $TRT_i\_exp$ ,  $msg\_xmt\_pos$ ,  $cha\_msg\_xmt$  and  $P_i\_srv\_end$ , where the transitions  $tkn\_pas\_P_i$  ( $i=1$  to 3),  $tkn\_pas\_P_k$  ( $k=2$  and 3) and  $tkn\_pas\_S_{j+1}$ , and the place  $cha\_msg\_xmt$  are identical to those given in the station submodel of Figure 4.2.

Transition  $tkn\_pas\_P_i$  models that the token is passed from the queue which is one level higher priority than priority  $i$  queue. When the transition  $tkn\_pas\_P_i$  is fired, the P-token is immediately moved in place  $P_i\_tkn\_cap$ , which models token capture at priority  $i$  queue. Token Rotation Timer for priority  $i$  queue ( $TRT_i$ ) is always either in running state which is modeled by a place labeled  $TRT_i\_run$ , or in expired state which is labeled by  $TRT_i\_exp$ . Initially P-token is marked in place  $TRT_i\_exp$ , representing  $TRT_i$  is in expired mode at the beginning of network operation.

If the  $TRT_i$  is expired (a marking of P-token in place  $TRT_i\_exp$ ) when priority  $i$  queue captures the token (a marking of P-token in place  $P_i\_tkn\_cap$ ), the immediate transition  $TRT_i\_exp$  is enabled and fires, putting one P-token in place  $P_i\_srv\_end$  that represents end of priority  $i$  queue service due to  $TRT_i$  expiration, and another P-token is moved in place  $TRT_i\_reset$  which models reset of  $TRT_i$ . This enables firing of transition  $TRT_i\_stt$ , and  $TRT_i$  is restarted for the next instant of token capture. When P-token is marked in place  $P_i\_srv\_end$ , the immediate transition  $tkn\_pas\_P_k$  (if current queue is priority 1 or 2) or  $tkn\_pas\_S_{j+1}$  (if current queue is priority 3) fires, and the token is passed to the lower priority queue or to its successor station in the logical ring, respectively.

However, if  $TRT_i$  is running (a marking of P-token in place  $TRT_i\_run$ ) when

priority  $i$  queue captures the token (a marking of P-token in place  $P_i\_tkn\_cap$ ), the immediate transition  $TRT_i\_run$  is fired, moving P-token in place  $msg\_xmt\_pos$  which represents the transmission of waiting message is possible. At this moment, if there is no P-token in place  $P_i\_queue$ , i.e., if the priority  $i$  queue is empty, the immediate transition  $P_i\_que\_emp$  is enabled and fires, putting the P-token in  $P_i\_srv\_end$  that represents end of priority  $i$  queue service due to empty queue. Also another P-token is moved in place  $TRT_i\_reset$  to reset and restart  $TRT_i$  again.

On the other hand, if the place  $P_i\_queue$  is not empty when P-token is marked in place  $msg\_xmt\_pos$ , the immediate transition  $P_i\_msg\_stt$  (start of priority  $i$  message transmission) is enabled and fires, putting the P-token in  $cha\_msg\_xmt$  representing a priority  $i$  message transmission through the channel, and another P-token is immediately moved in place  $TRT_i\_reset$  to reset and restart  $TRT_i$  again. At the end of a message transmission period, the deterministically timed transition  $P_i\_msg\_end$  moves P-token from  $cha\_msg\_xmt$  to  $P_i\_srv\_end$ , and the service of priority  $i$  queue is finished.

Based on the TPN model of priority scheme described above, a simulation model is developed in the following section.

## 4.2. Development of a Simulation Model

Simulation is a numerical experiment technique which imitates the operations of various kinds of real-world facilities or processes by using a digital computer. This technique involves certain types of mathematical and logical models that describe the behavior of a system over extended periods of real time. Since a

network system is categorized as an *event-driven system*, the discrete-event simulation technique is used to model the priority scheme of token bus protocols [37]. *Discrete-event simulation* concerns the modeling of a system that evolves over time, in which the state variables change only at a countable number of points in time. These points in time are the ones at which an event occurs, where an *event* is defined to be the instantaneous occurrence which may change the state of a system. In the discrete-event simulation, a *system* is defined as a collection of entities which are joined in some regular interaction or independence. Entities are characterized by data values which are called *attributes*, and these attributes are part of the system states for a discrete-event simulation model. Although a discrete-event simulation could conceptually be done by hand calculation, the amount of data that must be stored and manipulated for the system dictates that discrete-event simulation be done on a digital computer.

In the network system, the entities of the system are messages. Messages are characterized by the following attributes;

1. Time of generation: the instant when a message arrives at the transmitter queue.
2. Message length (or, message transmission time).
3. Source queue: the queue from which a message is generated.
4. Destination queue: potentially any queue other than the source queue.
5. Message priority: priority of the queue from which a message is generated.

Similar to the TPN model described in Section 4.1, the discrete-event simulation model for the priority scheme of token bus protocols consists of two sub-

models; message generation submodel and protocol submodel. Message generation submodel drives the entering of newly generated messages from the environment into the system. Protocol submodel describes the activities occurring within the network system.

The network-operating assumptions for the simulation model are exactly identical to that for analytical model, which include single-service discipline, Poisson distribution of message arrival, constant message length and infinite queue capacity. The same priority messages are assumed to have identical message arrival rate and message length. The simulation model is also assumed to have four priority levels according to the standard of SAE token bus protocol.

The simulation model has a modular structure which consists of several events. In the following, the events of the simulation model are described and the relationship between the TPN model and the simulation model is explained. The interaction diagram among the events is shown in Figure 4.5.

1. Initialization: Initialization event reads the simulation and network parameters for a given traffic condition.
2. Power\_up: In this event, the first message generations in all queues are scheduled. This event corresponds with the marking of P-token in place  $P_j\text{-msg-gen}$  ( $j=0, 1, 2$  and  $3$ ) of all stations in the TPN model at the beginning of network operation. The first token pass is also scheduled at this instant. In the TPN model, this event is modeled as putting a P-token in place  $tkn\_arv\_S_1$  of the first station in the logical ring sequence.
3. Message\_generation: As new messages are created to be put into queues,



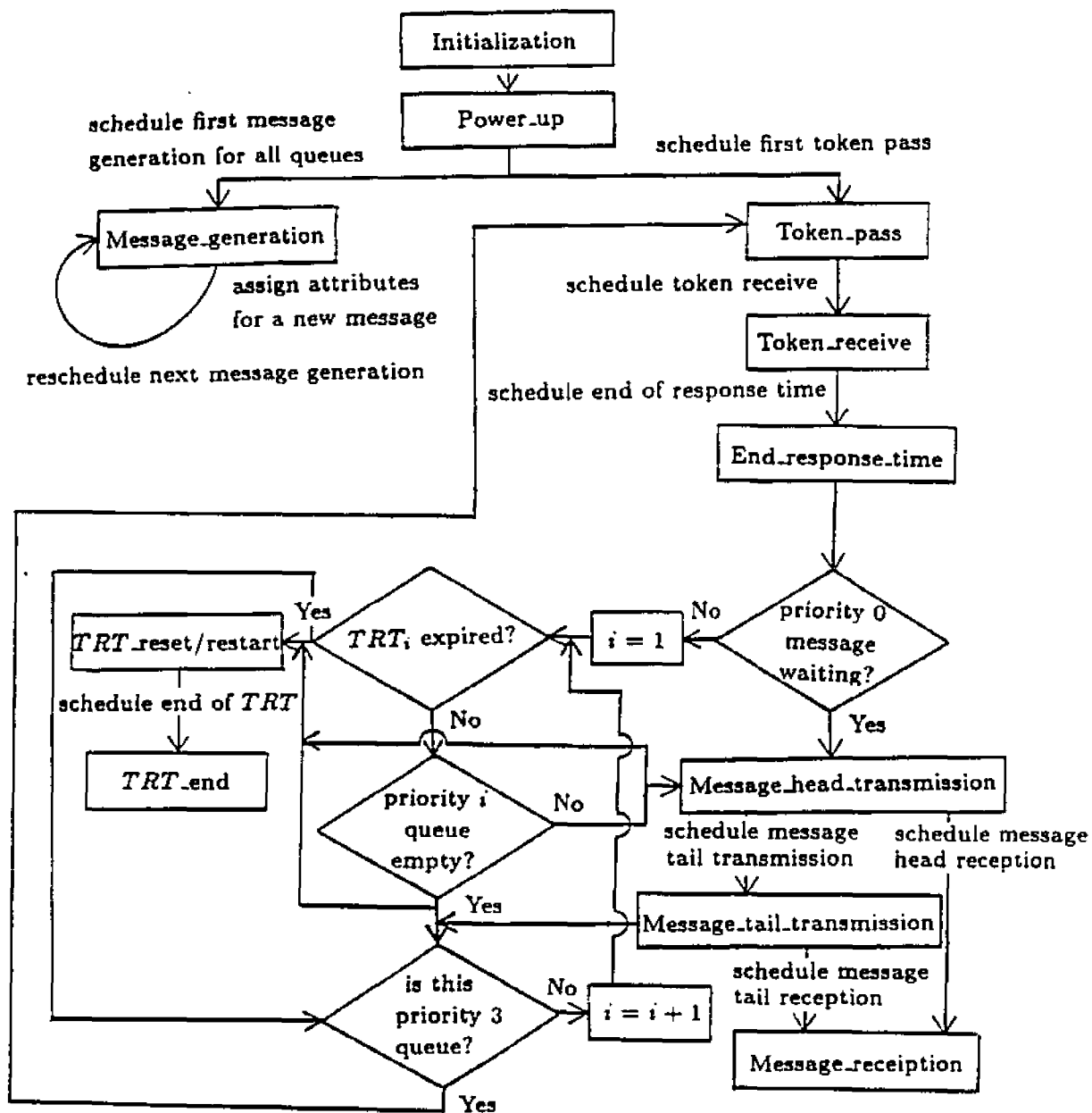


Figure 4.5: Event Interaction Diagram of Discrete-Event Simulation.

this event reschedules next message generation according to the given message interarrival time. In the TPN model, the exponentially timed transition  $P_j\_msg\_arv$  ( $j=0, 1, 2$  and  $3$ ) continuously generates new messages with the Poisson distribution, and put a P-token into place  $P_j\_queue$ . To distinguish each generated message, attributes for the message is assigned at this moment.

4. **Token\_pass**: At this moment, token is passed to the successor in the logical ring (fire immediate transition  $tkn\_pas\_S_j$ ). This event schedules the **Token\_receive** event after a propagation delay (if applicable).
5. **Token\_receive**: In the TPN model, this behavior is represented as a place  $tkn\_arv\_S_j$ . This event schedules the end of response time.
6. **End\_response\_time**: At the end of response time, the transmission of a message from priority 0 queue is possible (P-token in place  $P_0\_tkn\_cap$ ). This station checks if priority 0 queue has waiting messages. If it has (place  $P_0\_queue$  is not empty), it immediately transmits the waiting message (fire immediate transition  $P_0\_msg\_stt$ ) by calling **Message\_head\_transmission** event. If not (place  $P_0\_queue$  is empty), priority 0 message cannot be transmitted (fire immediate transition  $P_0\_que\_emp$ ) and the service of priority 0 queue is finished (P-token in place  $P_0\_srv\_end$ ). It immediately passes the token to priority 1 queue (fire immediate transition  $tkn\_pas\_P_1$ ).

When priority  $i$  ( $i=1, 2$  and  $3$ ) queue captures the token (P-token in place  $P_i\_tkn\_cap$ ) it checks if the corresponding  $TRT_i$  is expired. If the  $TRT_i$  is expired (P-token in place  $TRT_i\_exp$ ), the priority  $i$  queue cannot transmit its message (fire transition  $TRT_i\_exp$ ). The  $TRT_i$  is reset and restarted for the

next token capture by calling *TRT\_reset/restart* event.

On the other hand, if the  $TRT_i$  is not expired (P-token in place  $TRT_i-run$ ) when a priority  $i$  queue captures the token (P-token in place  $P_i-tnk-cap$ ), the waiting message can be transmitted (P-token in place  $msg-xmt-pos$ ). At this instant, if there is no waiting message (no P-token in place  $P_i-queue$ ), the service of this queue is completed (P-token in place  $P_i-srv-end$ ). If this queue is of priority 1 or 2, the token is passed to priority 2 or 3 queue, respectively. If this queue is priority 3, the token is passed to its successor in the logical ring. At the same time the  $TRT_i$  is reset and restarted for the next token capture.

However, if there is waiting message in priority  $i$  queue (place  $P_i-queue$  is not empty) when it has a chance to transmit a message (P-token in place  $msg-xmt-pos$ ), the queue resets and restarts its  $TRT_i$  and begins to transmit a message (fire transition  $P_i-msg-stt$ ) by calling *Message\_head\_transmission* event.

7. *TRT\_reset/restart*: This event resets and restarts the  $TRT$  (P-token in place  $TRT_i-reset$ ). This event schedules end of  $TRT_i$  event,  $TRT_end$ . When  $TRT$  is reset, the scheduled event of  $TRT_end$  is eliminated from the event calendar.
8. *TRT\_end*: This event indicates expiration of corresponding  $TRT_i$  (P-token in place  $TRT_i-exp$ ).
9. *Message\_head\_transmission*: At this event, head of message is transmitted. This event schedules reception of the head of message. Transmission of the tail of message is also scheduled at this moment to take into account the

message transmission time.

10. *Message\_tail\_transmission*: This event transmits tail of message. When a message transmission is completed, service of this queue is completed (P-token in place  $P_i\text{-srv\_end}$ ), and a message transmission from the next priority queue is examined if  $i=0, 1$  and  $2$ , or the token is passed to its successor if  $i=3$ . This event schedules reception of the tail of message.
11. *Message\_reception*: After a message is completely received, the statistical data such as mean and standard deviation of queueing delay, data latency, throughput and bus utilization for each priority class are collected.

Commonly used languages for discrete-event simulation include FORTRAN, GPSS [38], SIMSCRIPT II.5 [39], SIMULA [40], SLAM [41] and SIMAN [42]. SIMAN was selected as the language for simulation in view of the following requirements:

1. Programming flexibility.
2. Modularity and structured programming.
3. Built-in data analysis and real-time event scheduling capacities.
4. Verification and run-time debugging.
5. Combined discrete-event and continuous-time simulation capacities.
6. Program portability.

Further details of SIMAN and its comparison with other simulation languages are provided in [43].

## Chapter 5

### COMPARISON OF ANALYTICAL MODEL WITH SIMULATION EXPERIMENT

This chapter provides a comparison of the results derived from analytical and simulation models presented in Chapter 3 and Chapter 4, respectively. The analysis was performed under the following assumptions;

1. The processes within a queue are independent of the processes within the other queues.
2. When the token arrives at a priority  $i$  ( $i=1$  to  $K$ ) queue, the message waiting process and  $TRT_i$  expiration process at that queue are independent.

Accuracy of the analytical model is found to be better at low traffic since the above assumptions are asymptotically exact for zero arrival rates. The analytical results are less accurate at heavy traffic. This is because the processes within a queue are dependent upon those within the other queues. In addition, as the traffic increases,  $TRT_i$  expiration process and message waiting process at a queue becomes more dependent of each other.

The offered traffic is an important parameter for the measurement of network performance, and is defined as follows [5].

**Definition 5.1:** Offered traffic for the priority  $i$  class is defined as the expected value of the total message transmission time of the priority  $i$  class traffic on the network per unit time, and can be expressed as;

$$G_i = \frac{N_i L_i}{\tau_i} \quad (5.1)$$

where,

$N_i$ = Number of the priority  $i$  queue in the network.

$L_i$ = Average transmission time of the priority  $i$  message in unit of time, i.e., length of the message in bits divided by the data latency in bit/unit time.

$\tau_i$ = Average message interarrival time of the priority  $i$  queue ( $\tau_i = 1/\lambda_i$ ). ■

**Definition 5.2:** Total offered traffic is the sum of the individual offered traffic of each priority level, and is expressed as

$$G = \sum_{i=0}^K G_i \quad (5.2)$$

■

In this thesis, two different traffic conditions are considered. The first is the important case of *asymmetric system* in which the message length and message generation interval at each priority level are different from each other. In the ICCS, high priority messages (real-time data such as sensor and controller information) usually have short lengths and high message interarrival rates, and low priority messages (non-real-time data such as those for CAD information and file transfer) are usually long and rather infrequently transmitted. Real-time data have a smaller and tighter upper bound on the data latency than that of non-real-time data. Thus, real-time data should receive preferential treatment at the expense of the increased data latency of non-real-time data which can tolerate larger data latency.

Next, a *symmetric system* is considered, where the message length and message generation interval for all priority classes are identical. Since the data latency

is the most important factor for the network performance, the performance of priority scheme is analyzed on the basis of data latency.

### Case 1: Asymmetric Traffic

For the asymmetric case, three different traffic loads, low ( $G=0.2$ ), medium ( $G=0.5$ ) and high ( $G=0.8$ ), and four levels of priority are considered. Offered traffic of each individual priority class is assumed to be identical, i.e.,  $G_0 = G_1 = G_2 = G_3$ . The traffic condition is given as follows.

1. Message transmission time:  $L_0 = 0.1msec$ ,  $L_1 = 0.2msec$ ,  $L_2 = 0.3msec$ ,  
 $L_3 = 0.4msec$ .

2. Number of queues:  $N_0 = N_1 = N_2 = N_3=4$ .

3. Average message interarrival time

$$G=0.2: \tau_0 = 8msec, \tau_1 = 16msec, \tau_2 = 24msec, \tau_3 = 32msec,$$

$$G=0.5: \tau_0 = 3.2msec, \tau_1 = 6.4msec, \tau_2 = 9.6msec, \tau_3 = 12.8msec,$$

$$G=0.8: \tau_0 = 2msec, \tau_1 = 4msec, \tau_2 = 6msec, \tau_3 = 8msec.$$

4. Total idle time due to station response and propagation delay at all stations during one token cycle:  $R=0.006 msec$ .

5.  $TRT_2 = 2TRT_3, TRT_1 = 3TRT_3$ .

The simulation results are dependent upon the seed number of the random number generator. That is, even under the same traffic condition, the results could be changed if different seed numbers are used. Because of the randomness of data latency, simulation results should be measured based on the sufficient collection of statistical data. Therefore, 10 different experiments were performed with different seed numbers. Statistical data obtained from these 10 experiments were used to

obtain 95 % confidence interval.

Average data latencies obtained from both simulation and analytical models at this traffic condition for offered traffic  $G=0.2, 0.5$  and  $0.8$  are shown in Figures 5.1 to 5.3, respectively. These figures illustrate the change of average data latencies of priority 0, 1, 2 and 3 messages as the values of  $TRT_1, TRT_2$  and  $TRT_3$  increase. The abscissa in each figure is  $TRT_3$  values, and  $TRT_2$  and  $TRT_1$  are set at  $2TRT_3$  and  $3TRT_3$ , respectively. In Figures 5.1 to 5.3, data latencies determined from the analytical model are represented by solid, long dashed, dashed and dotted lines for priority 0, 1, 2 and 3, respectively. The data latencies obtained from the simulation model are given as circle ( $\circ$ ), cross ( $\times$ ), triangle ( $\triangle$ ) and square ( $\square$ ) for priorities 0, 1, 2 and 3, respectively. The simulation results indicate the intervals between symbols to represent 95 % confidence.

Figure 5.1 shows that, at low offered traffic ( $G=0.2$ ), data latencies for all priority classes are practically independent of the  $TRT$  values in contrast to those for medium ( $G=0.5$ ) and high ( $G=0.8$ ) offered traffic in Figures 5.2 and 5.3, respectively. The rationale is that, at low offered traffic, most of the queues are empty when the token arrives. Since the token circulation time is relatively short at low traffic,  $TRT$ 's are usually in running state, i.e., not expired, when the token arrives. Therefore, the unexpired  $TRT$ 's have no bearing on the data latencies of all priority classes at low traffic.

Figures 5.2 to 5.3 show that, for  $G=0.5$  and  $0.8$ , the dependence of data latency on the respective  $TRT$  values exhibits a close agreement between simulation and analytical results. When the  $TRT_i$  ( $i=1, 2$  and  $3$ ) are set to very small values,



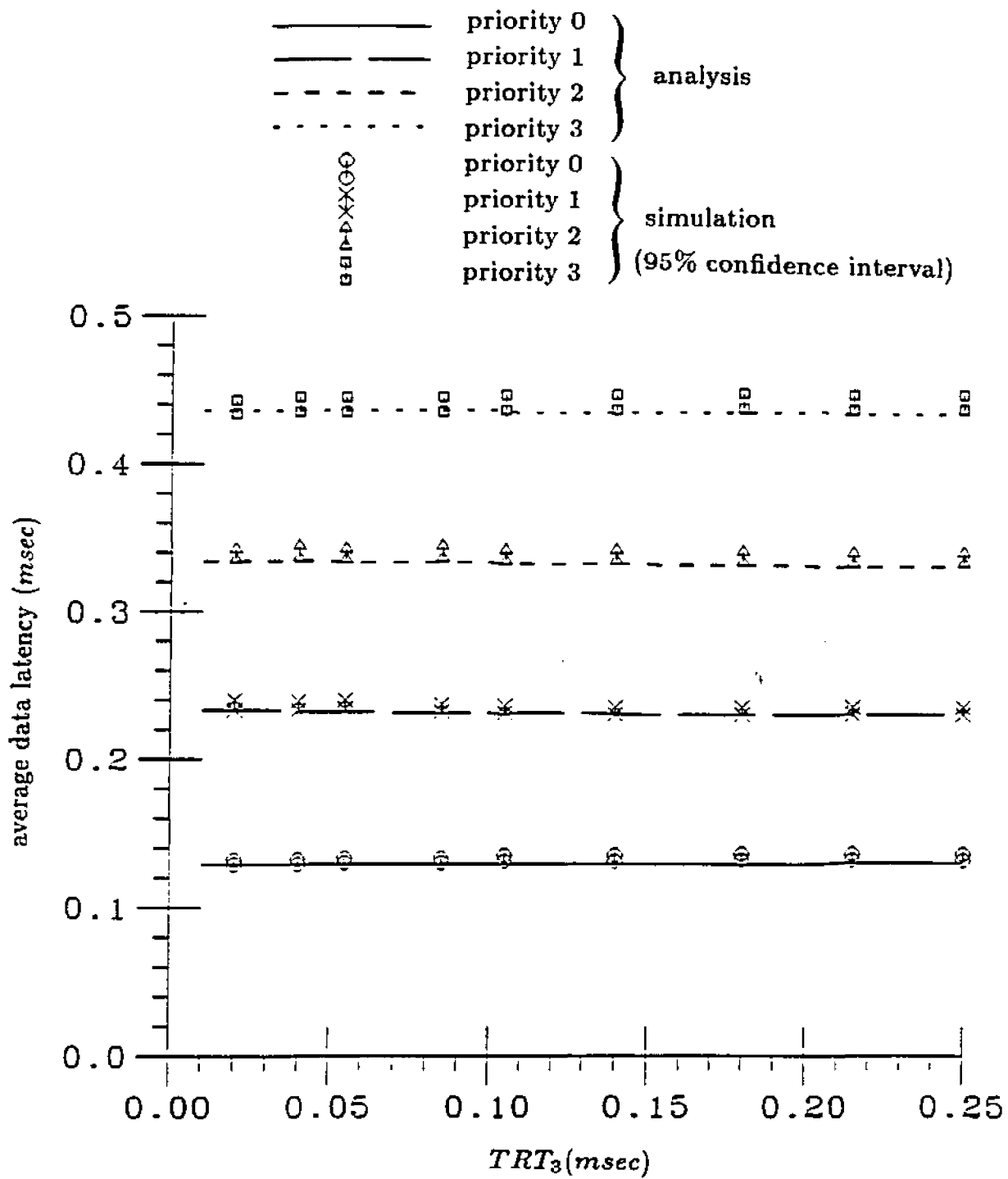


Figure 5.1: Average Data Latency for Traffic Condition 1 ( $G=0.2$ ).

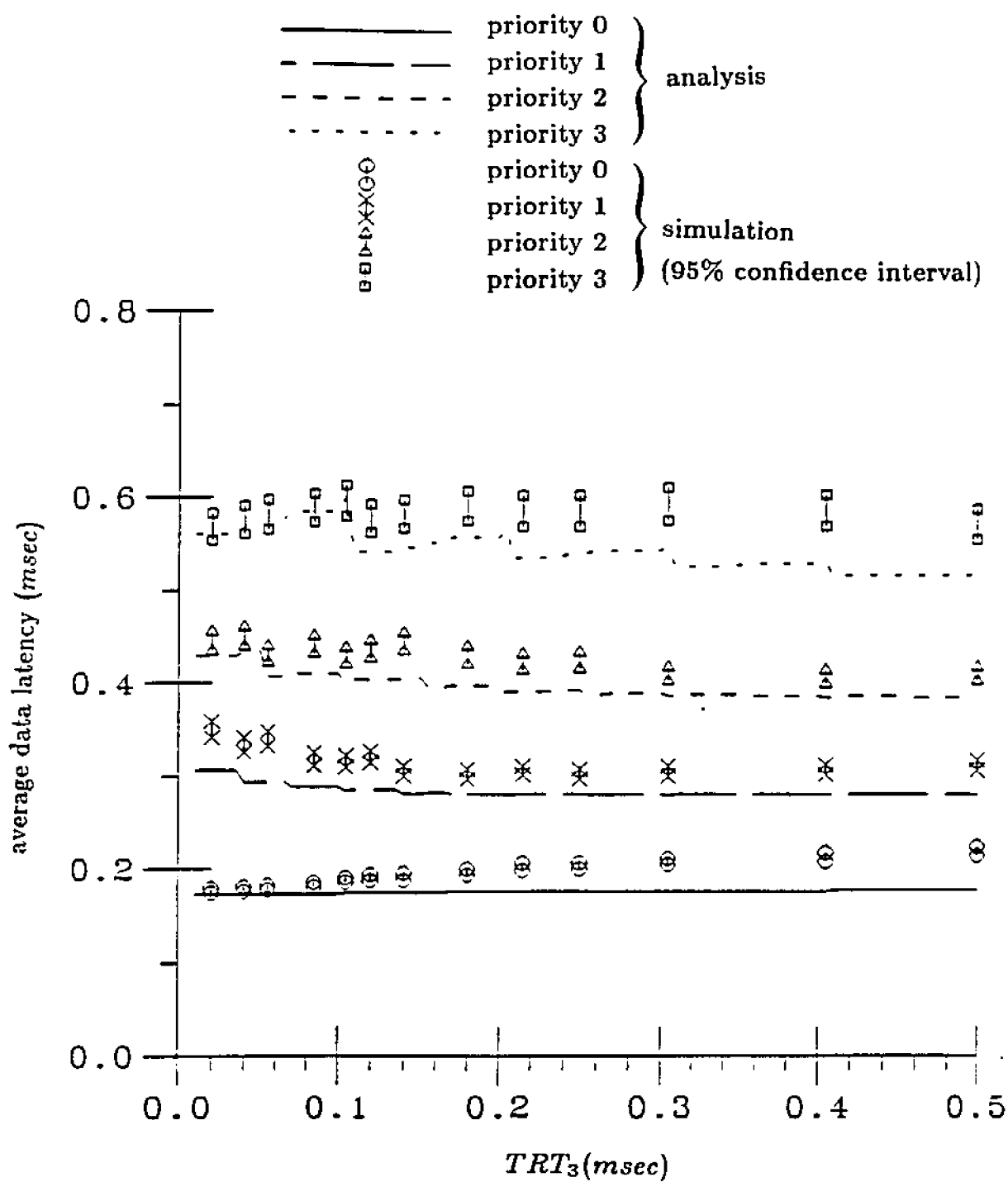


Figure 5.2: Average Data Latency for Traffic Condition 1 ( $G=0.5$ ).

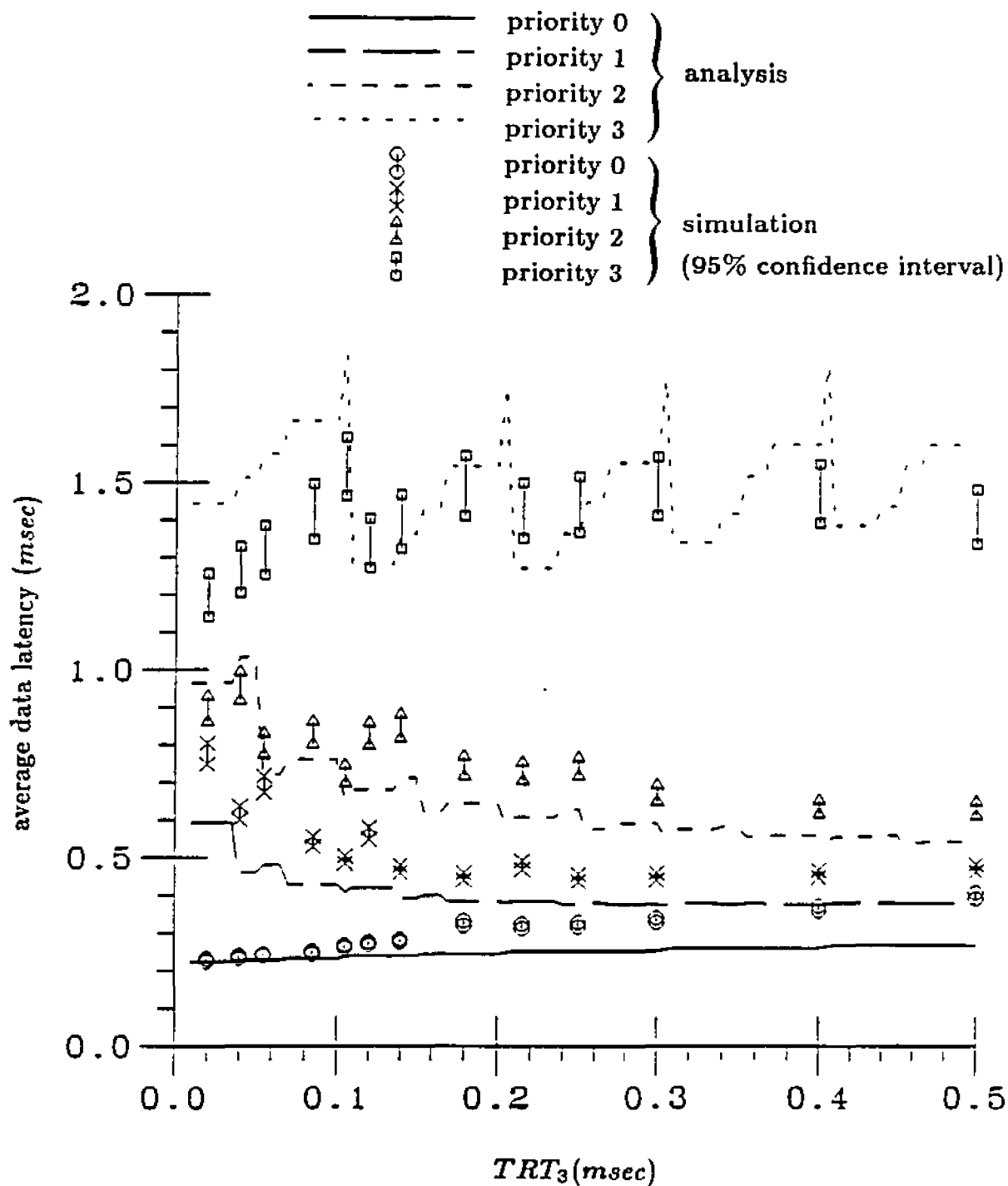


Figure 5.3: Average Data Latency for Traffic Condition 1 ( $G=0.8$ ).

i.e.,  $TRT_3=10msec$ , data latency for priority 0 is reduced to a low value. This is because the  $TRT$ 's are set so small that most of the priority 1, 2 and 3 messages experience  $TRT$  expiration, and the channel capacity is practically dedicated to priority 0 class, and the priority 1, 2 and 3 queues are allowed to transmit using the left-over channel capacity. Data latency for priority 0 increases as  $TRT_i$  ( $i=1, 2$  and  $3$ ) increases, because the the transmission of lower priority (priority 1, 2 and 3) messages are less dependent on the respective  $TRT$  expiration. Thus, more channel capacity is assigned to the lower priority messages. This causes an increase in the data latency of priority 0 messages.

For a given traffic condition, the possible token circulation times are  $n_0L_0 + n_1L_1 + n_2L_2 + n_3L_3 + R$ , ( $n_i = 0, \dots, 4$ ). Since  $L_j$  ( $j=0, 1, 2$  and  $3$ ) are multiple of  $0.1 msec$  in this traffic condition, the token circulation times are distributed with an interval of  $0.1 msec$ . Figure 5.4 shows the probability density function of token circulation time,  $T_r$ , for  $G=0.5$  and  $TRT_3=0.055 msec$ . Data latency for priority  $i$  class decreases when  $TRT_i$  is set just beyond a possible token circulation time. This is the reason why data latency for priority  $i$  ( $i=1, 2$  and  $3$ ) class decreases with the  $TRT_i$  interval of  $0.1 msec$ .

Figures 5.2 and 5.3 show that, as  $TRT_i$  ( $i=1$  to  $3$ ) increases, the analytical model generally underestimates data latency. This is because, as  $TRT_i$  increases, messages are less subject to the  $TRT_i$  expiration and the priority scheme model is closer to the Kuehn's model (without the priority scheme).

The results obtained from Kuehn's model (without the priority scheme) under the same traffic condition is given in Figure 5.5 (Kuehn's model may be viewed that

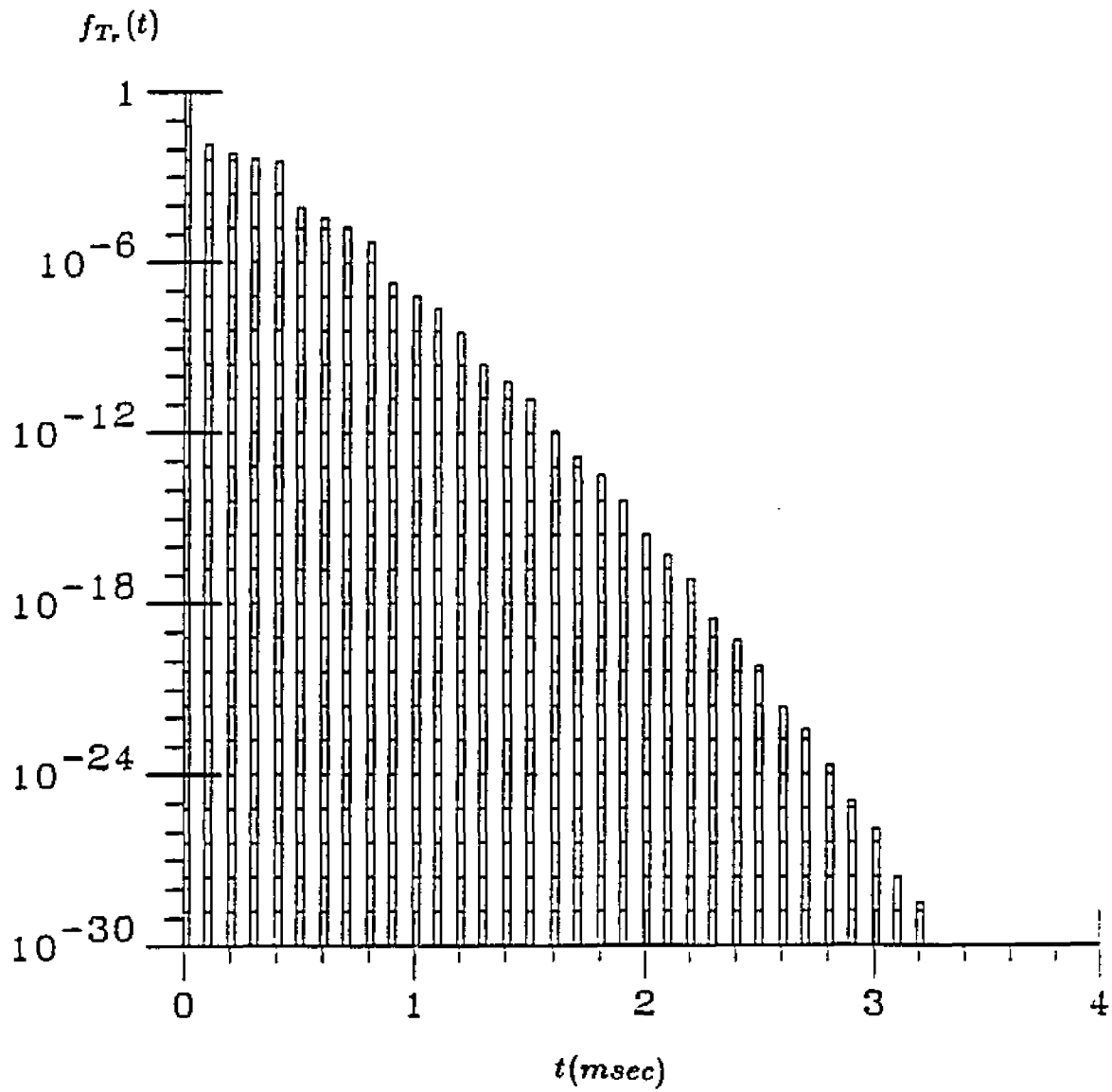


Figure 5.4: Probability Density Function of  $T_r$  at Traffic Condition 1.

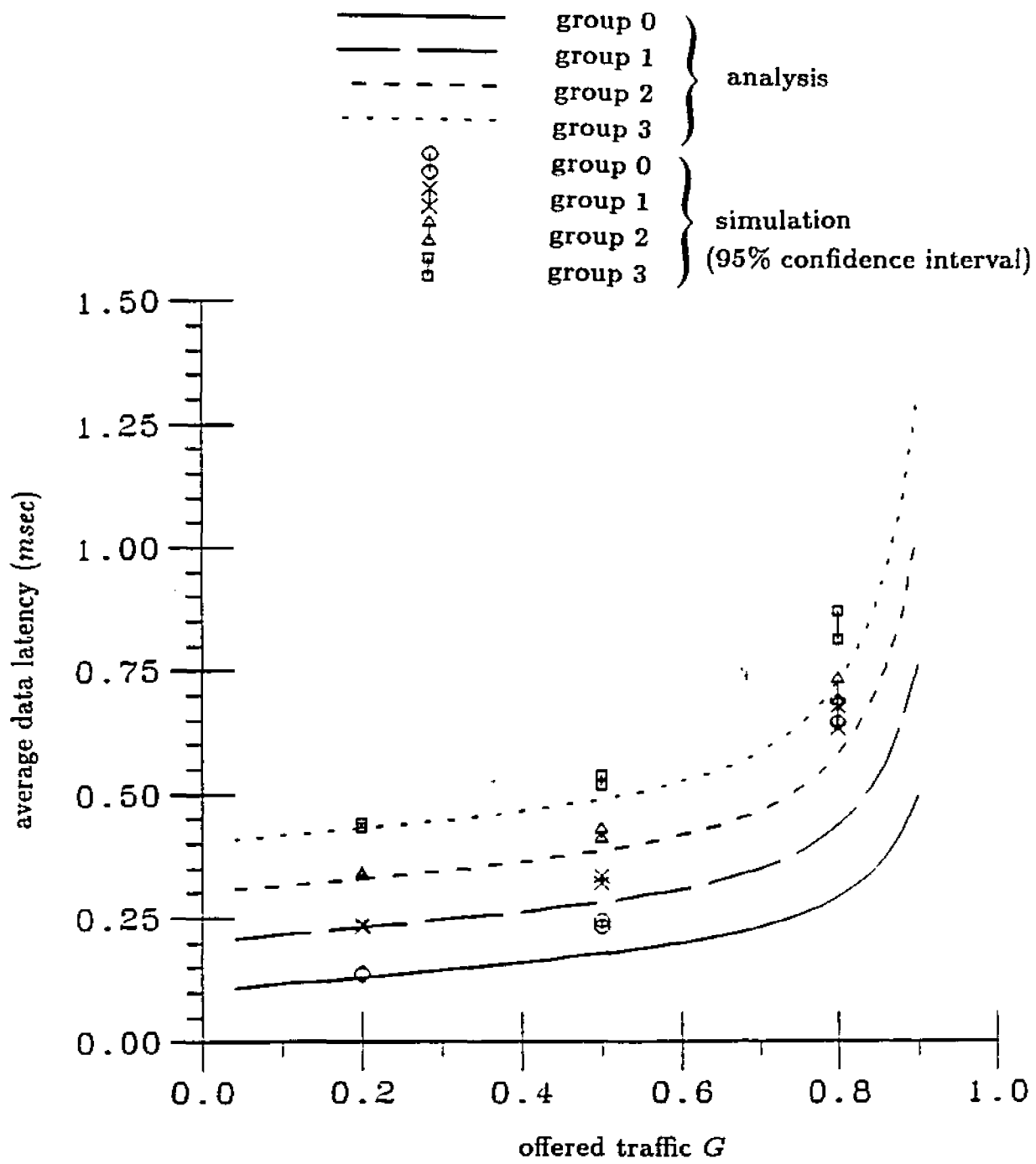


Figure 5.5: Kuehn's Approximation for Asymmetric System.

all  $TRT$ 's are set to infinity). Stations in the network are divided into four groups (group 0, 1, 2 and 3) according to their message transmission time ( $L_i$ ). Traffic condition of group  $i$  is identical to that of priority  $i$  class given in Case 1. Figure 5.5 exhibits data latency as a function of offered traffic  $G$  for each group. The change of data latencies determined from Kuehn's model are represented as solid, long dashed, dashed and dotted lines for the groups 0, 1, 2 and 3, respectively. The data latencies obtained from the simulation model are given as circle ( $\circ$ ), cross ( $\times$ ), triangle ( $\triangle$ ) and square ( $\square$ ) for groups 0, 1, 2 and 3, respectively, with 95 % confidence intervals. This figure shows that Kuehn's model underestimates the data latency, especially at high offered load ( $G=0.8$ ) for the group 0. This error in data latency is inevitable as the variance of the token circulation time is inadequately evaluated due to the independence assumption [23]. A comparison of the results of Figures 5.3 and 5.5 (at  $G=0.8$ ) shows that the results of priority scheme model are significantly superior to those of the Kuehn's model. This is because the  $\mu''_j$  and  $\mu''_j$  given in (3.42) and (3.43) of Chapter 3 allow more accurate estimation of the variance of the effective service time  $T''_j$ . This largely alleviates the problem of erroneous estimation of data latency, which is unavoidable whenever the independence assumption is used.

A qualitative assessment of Figures 5.1 to 5.3 is that accuracy of the analytical model is good for low ( $G=0.2$ ) to medium ( $G=0.5$ ) traffic. However, when the traffic is high ( $G=0.8$ ), the analytical model becomes less accurate. This is because, at high offered traffic, more messages are built up in the queue and the state of a queue is more dependent upon the states of the other queues. Also,

$TRT_i$  expiration process at a queue is more dependent upon the message waiting process at the same queue. Since the analysis is based on the independence assumptions, the accuracy of the analytical model is degraded at high traffic.

### Case 2: Symmetric Traffic

For a symmetric system, three different offered loads, low ( $G=0.2$ ), medium ( $G=0.5$ ) and high ( $G=0.8$ ), are considered. Offered traffic of each individual priority class is assumed to be identical, i.e.,  $G_0 = G_1 = G_2 = G_3$ . The traffic condition is given as follows.

1. Message transmission time:  $L_0 = L_1 = L_2 = L_3 = L=0.05 \text{ msec}$ .
2. Number of queues:  $N_0 = N_1 = N_2 = N_3=4$ .
3. Average message interarrival time

$$G=0.2: \tau_0 = \tau_1 = \tau_2 = \tau_3 = 4\text{msec},$$

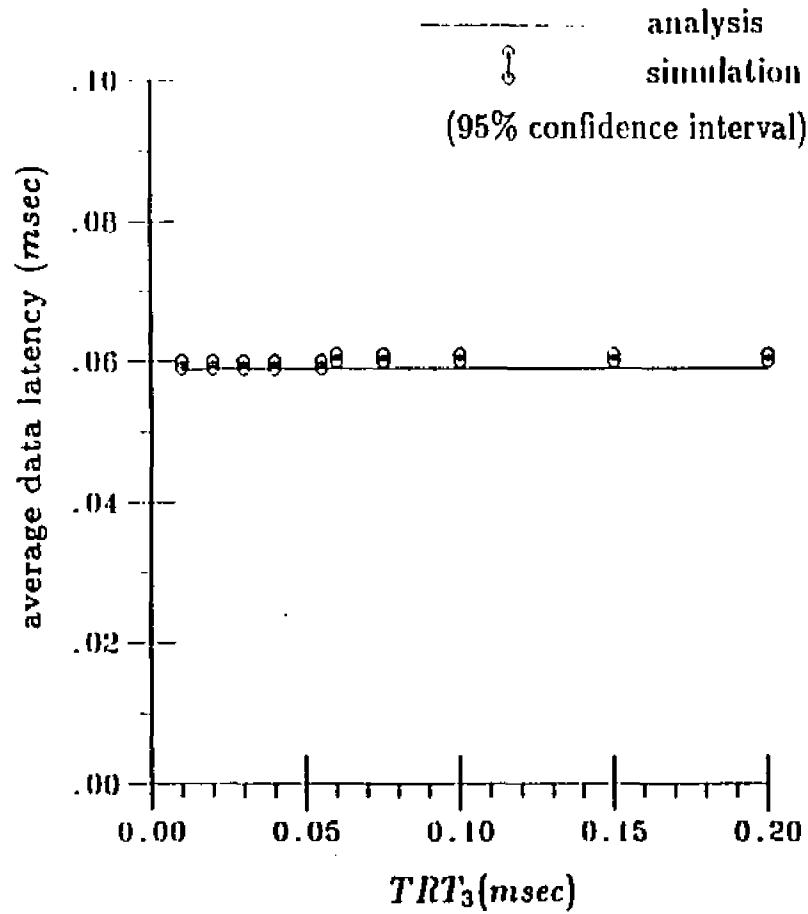
$$G=0.5: \tau_0 = \tau_1 = \tau_2 = \tau_3 = 1.6\text{msec},$$

$$G=0.8: \tau_0 = \tau_1 = \tau_2 = \tau_3 = 1\text{msec},$$

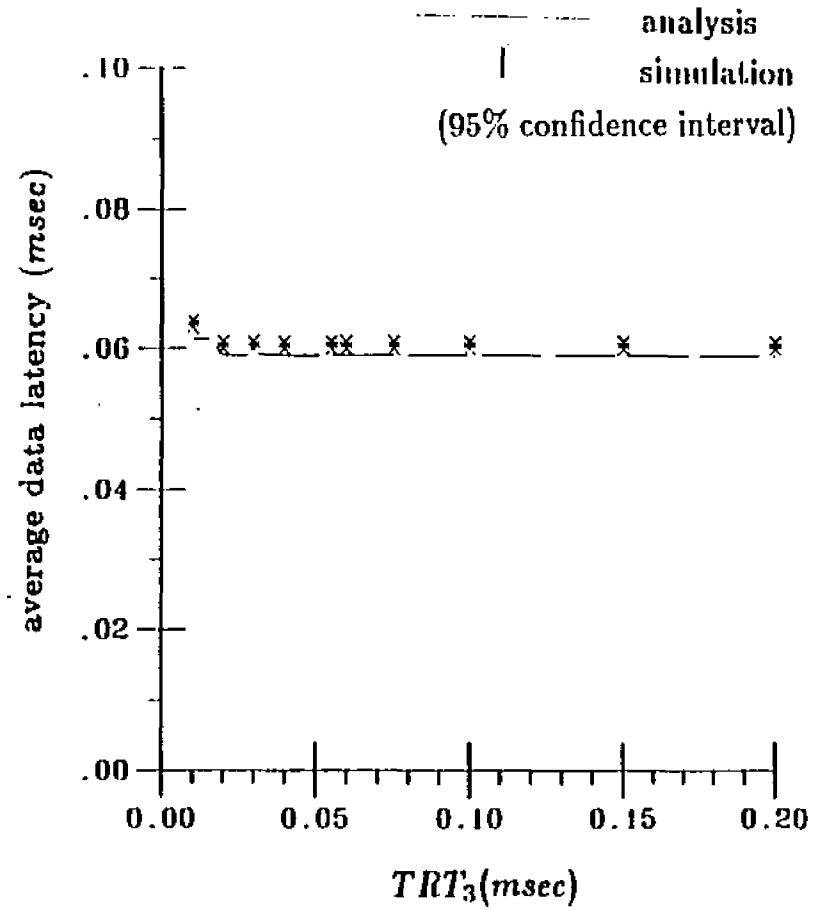
4. Total idle time:  $R=0.006 \text{ msec}$ .
5.  $TRT_2 = 2TRT_3, TRT_1 = 3TRT_3$ .

The simulation results were obtained on the basis of 95 % confidence interval, and the results from both simulation and analytical models for  $G=0.2, 0.5$  and  $0.8$  are given in Figures 5.6 to 5.8, respectively, with data latencies of priority 0, 1, 2 and 3 messages versus the respective values of  $TRT_i$ . The lines and symbols which represent the data latencies obtained from analytical and simulation models for each priority class are the same as those in Figures 5.1 to 5.3.



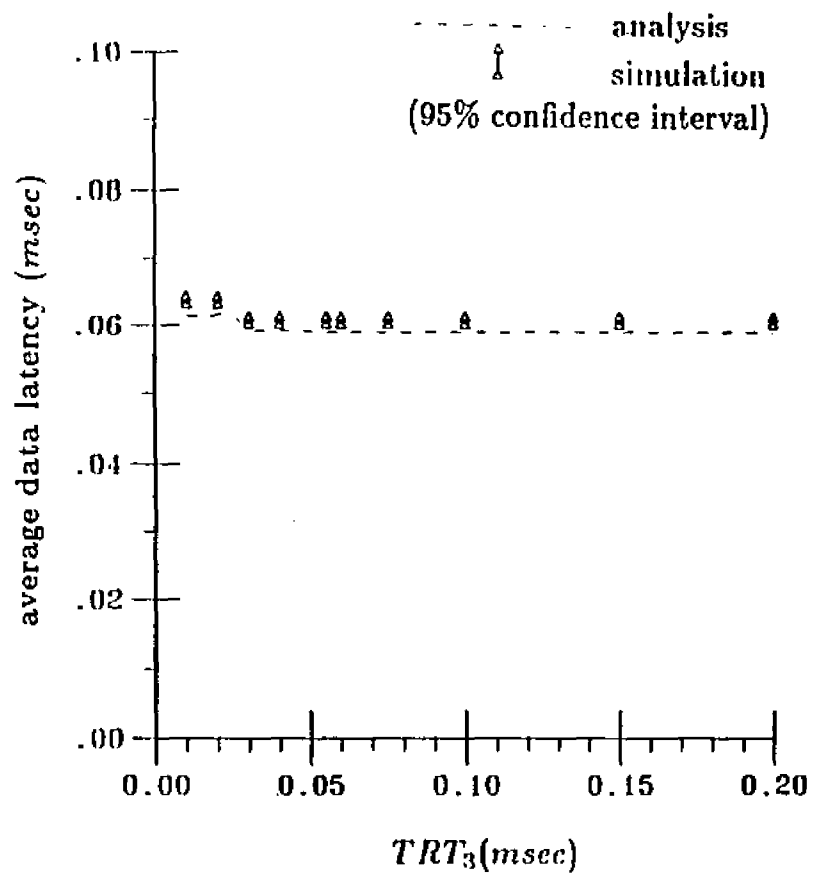


priority 0

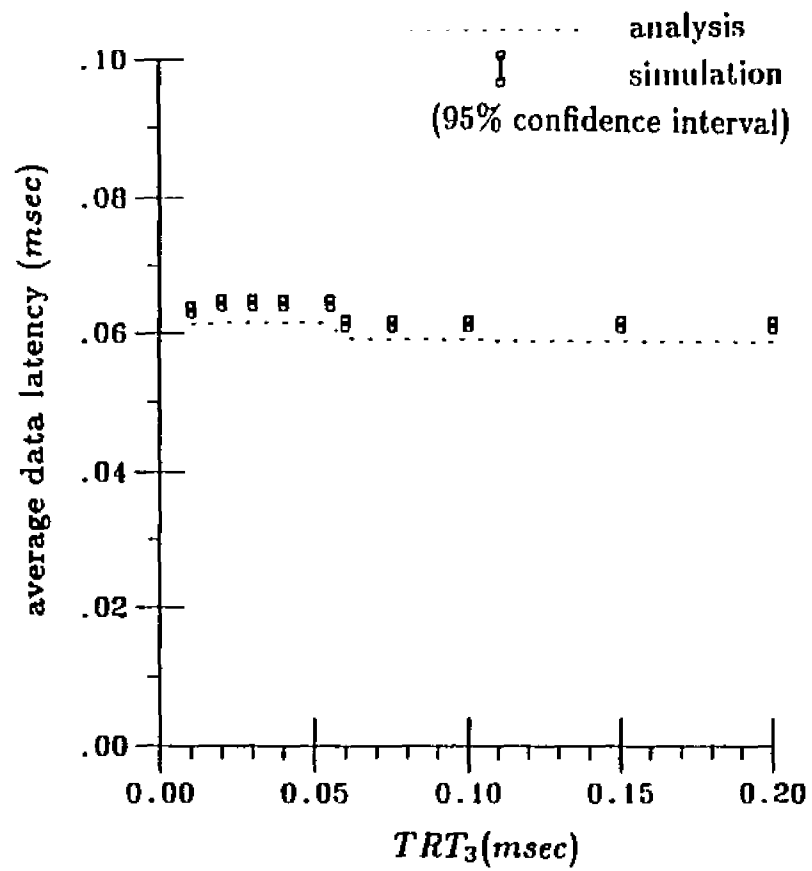


priority 1

Figure 5.6: Average Data Latency for Symmetric Traffic ( $G=0.2$ ) (cont. on next page).



priority 2



priority 3

Figure 5.6: (cont.)

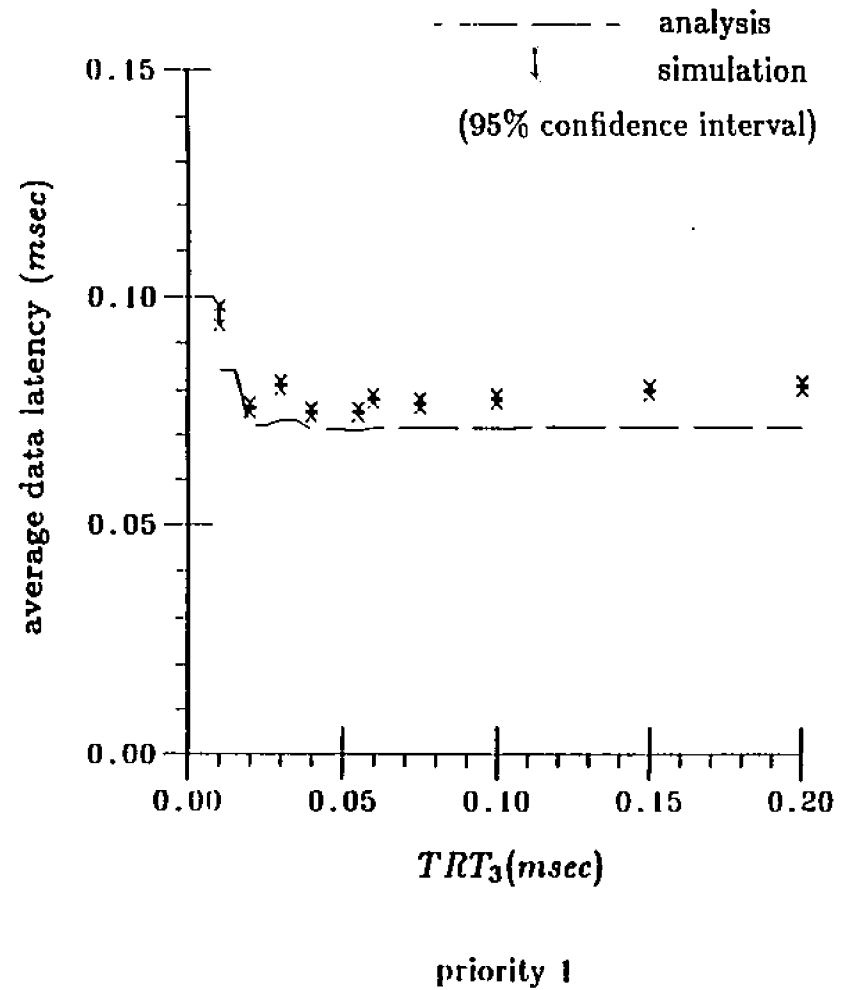
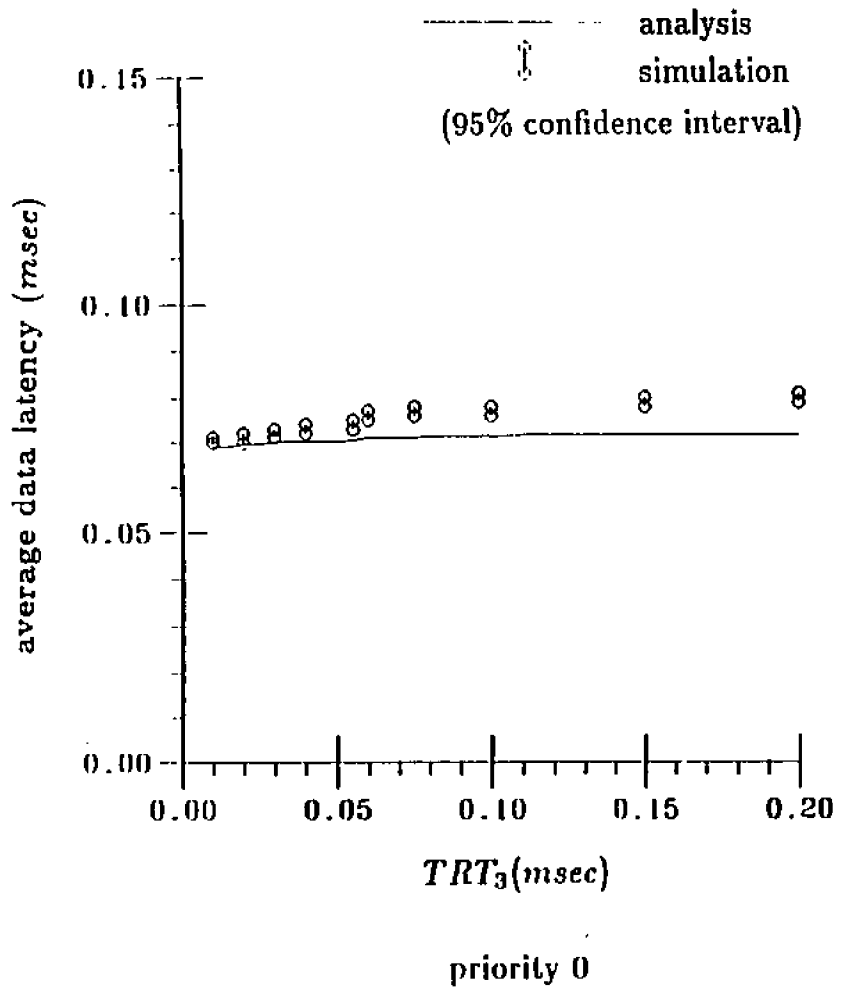
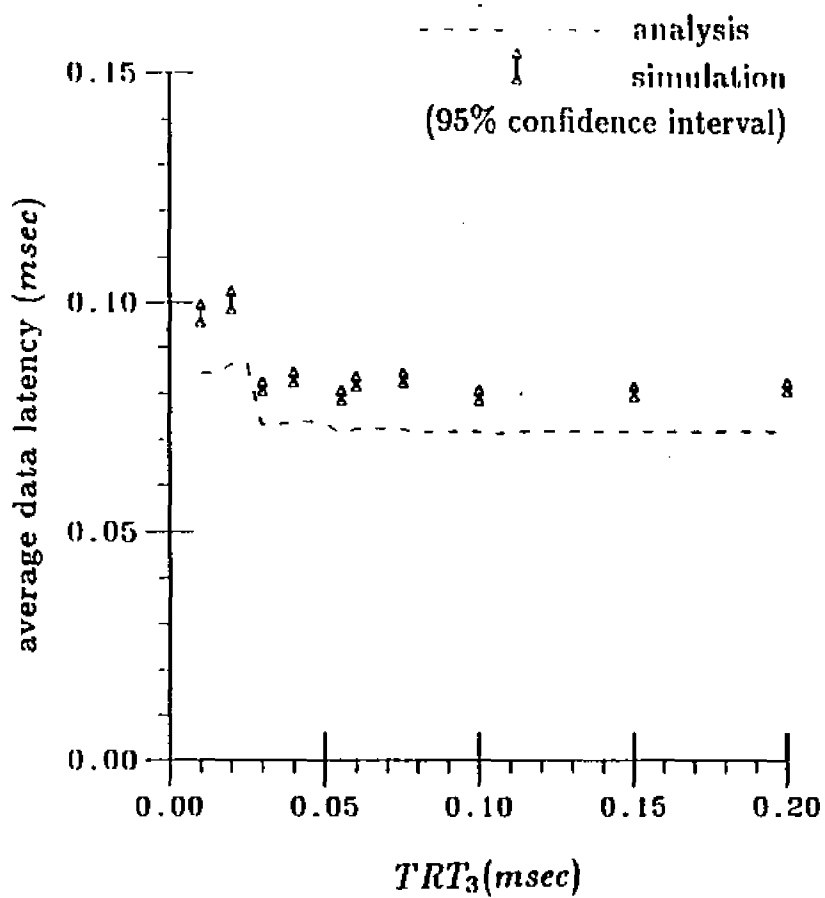
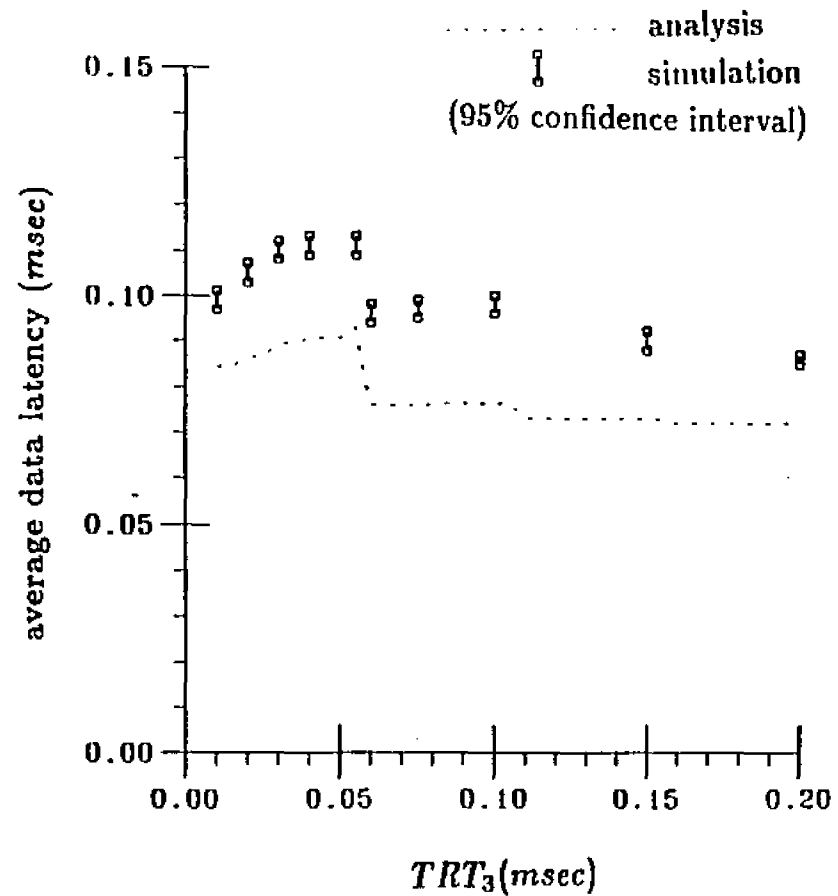


Figure 5.7: Average Data Latency for Symmetric Traffic ( $G=0.5$ ) (cont. on next page).

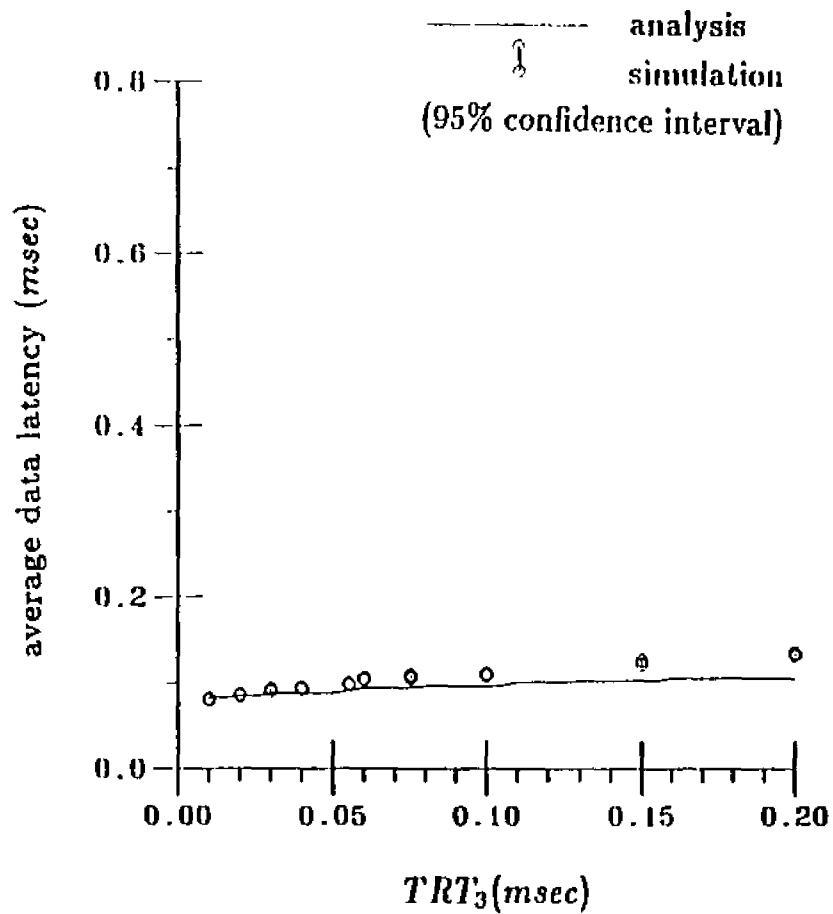


priority 2

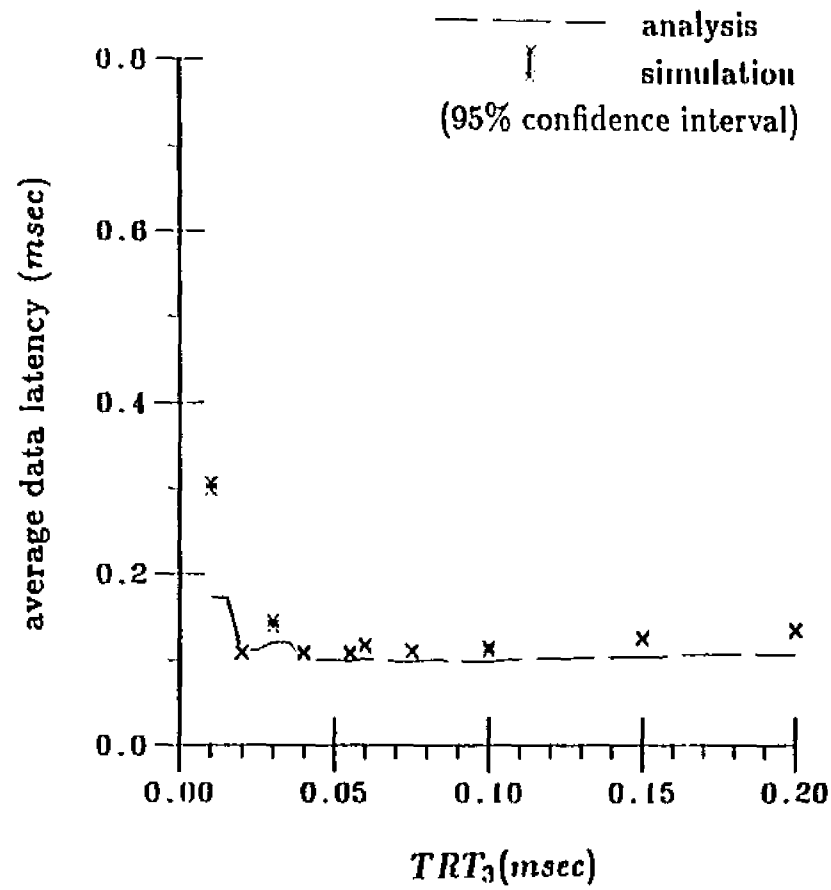


priority 3

Figure 5.7: (cont.)

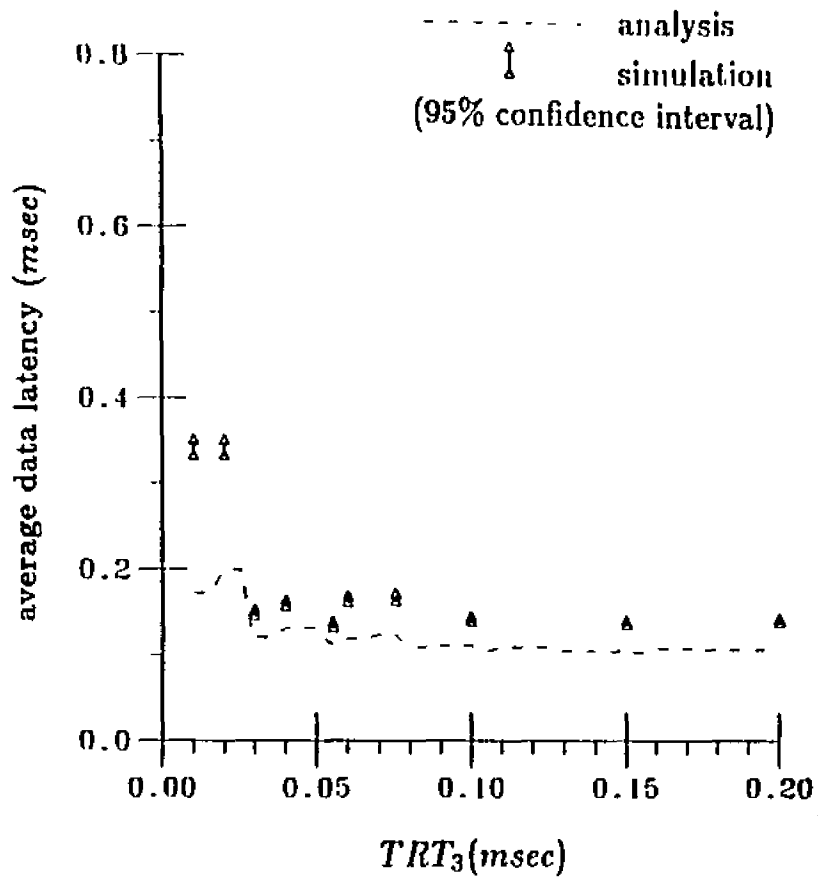


priority 0

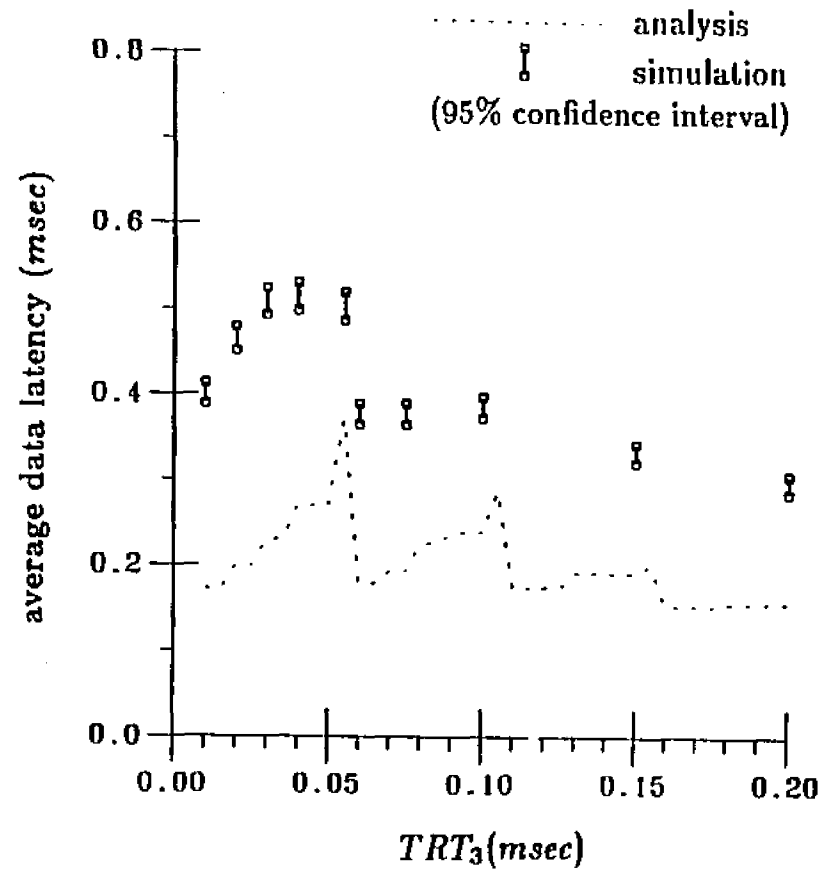


priority 1

Figure 5.8: Average Data Latency for Symmetric Traffic ( $G=0.8$ ) (cont. on next page).



priority 2



priority 3

Figure 5.8: (cont.)

The characteristics of data latency for the symmetric case are the same as those for the asymmetric case, except that the period of data latency decrement for priority 1, 2 and 3 levels is 0.05 msec. A probability density function of token circulation time for  $G=0.5$  and  $TRT_3=0.055$  msec is given in Figure 5.9. Since the system is symmetric, the possible token circulation times  $T_r$  are  $nL + R$  ( $n = 0, \dots, 16$ ). Thus, the data latency for priority 1, 2 and 3 levels decreases with a  $TRT_i$  interval of  $L$  (0.05 msec). When the data latency of a priority level decreases, the data latencies of all the other priority levels tend to increase.

When  $TRT_i$  ( $i=1, 2$  and  $3$ ) are set to large values, e.g.,  $TRT_3 = 0.2$  msec, data latency differences (or, queueing delay differences) among priority 0, 1, 2 and 3 classes become smaller. This is because  $TRT$ 's are set sufficiently large so that most of the priority 1, 2 and 3 messages are transmitted before the expiry of the respective  $TRT$ .

Figure 5.8 shows that, at high offered traffic ( $G=0.8$ ), when  $TRT_i$  ( $i=1, 2$  and  $3$ ) are set to small values, accuracy of the data latency of lower priority levels (priority 1, 2 and 3) is reduced. This is because more messages are built up in the queues due to  $TRT_i$  expiration at this traffic condition. Thus, the state of a queue is more dependent upon the states of the other queues. Since the analytical model is developed on the basis of the independence assumptions, the accuracy is reduced at these traffic conditions. This phenomenon is similar to that for asymmetric traffic in Case 1.

Accuracy of the analytical model is reduced especially for priority 3 queues under high traffic ( $G=0.8$ ) because of the independence assumptions. At high

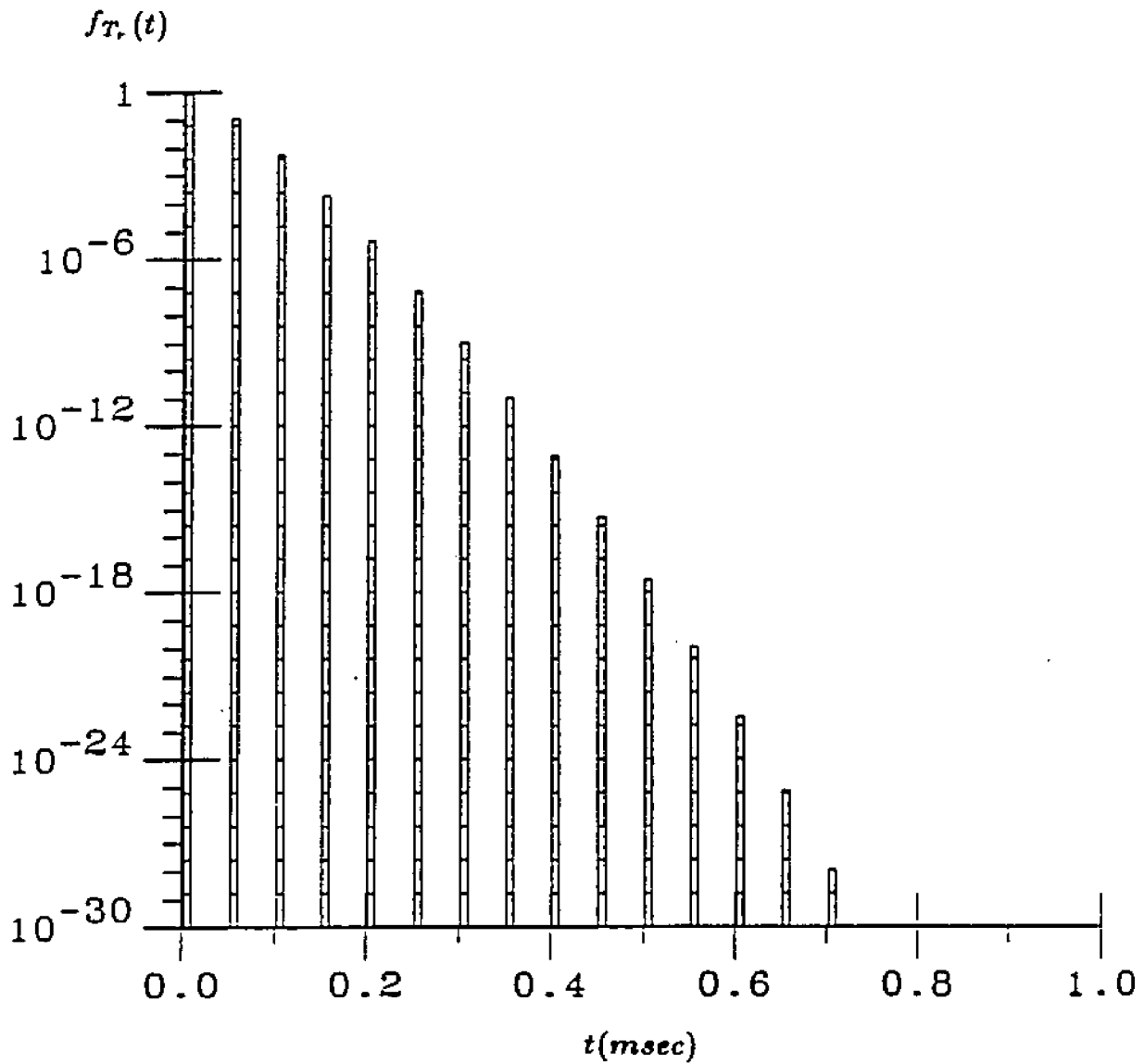


Figure 5.9: Probability Density Function of  $T_r$  at Traffic Condition 2.



traffic, more messages are built up in the priority 3 queues and the effect of independence assumptions becomes more significant. However, accuracy of the priority 3 data latency is not so significant from the view point of ICCS network design and operation, because the priority 3 class is usually assigned to non-real-time data which is not time-critical.

Although the analytical model developed in this thesis is approximate, its comparison with the simulation experiments shows that the model closely reflects the properties of the priority scheme in token bus protocols.

## Chapter 6

### PRELIMINARY DESIGN OF ICCS NETWORKS

In ICCS networks, data latency at each station is designed to have a maximum bound. The maximum allowable data latency is determined by the station's functional characteristics. For example, certain control loops in the ICCS network may have faster dynamics than others, thus the real-time data (such as sensor and controller information) generated from these stations must have smaller data latency. To accomplish this design requirement, the priority scheme in token bus protocols has been introduced.

This chapter introduces an approach for preliminary design of ICCS networks which use token bus protocols with the priority scheme. Design criteria for ICCS networks are described in Section 6.1. Section 6.2 illustrates an example of ICCS network design.

#### 6.1. Preliminary Design Criteria of ICCS Network

In the ICCS network which has a priority mechanism, access of medium is controlled by *TRT* timers. As described in Chapter 5, data latency is heavily dependent upon the *TRT* values as well as the message length and interarrival time. When the ICCS network is designed, the message length and interarrival time at each station are usually determined *a priori* according to their functional characteristics. Furthermore, controlling of medium access can be easily accomplished by simply adjusting the *TRT* timer values. To efficiently design and operate a

network system, optimal  $TRT$  values should be determined.

Determination of optimal  $TRT$  values using the simulation technique is time-consuming and very cumbersome when a large number design options are available. The analytical model developed in Chapter 3 can be directly used to determine at least sub-optimal values of  $TRT$  with little effort.

As mentioned earlier, the ICCS network has to be designed such that data latency at each station is bounded by the pre-determined value with a given confidence. To achieve this goal, it is necessary to obtain the average and variance of data latency and preferably higher moments. As an extension of this thesis, the variance of data latency at each priority class need to be investigated. However, at this stage of preliminary ICCS networks design, no information regarding the variance of data latency need to be generated. Intuitively, if the variance of effective service time (see Definition 3.2 of Chapter 3) decreases, the variance of data latency will also decrease (this postulation will be reviewed in Section 6.2). In the ICCS, data latency of real-time data is critical, while it is not so significant for non-real-time data. Therefore, as an initial design step of ICCS network, an alternative design criteria is introduced, which minimize the sum of the variances of effective service times of all the priority levels which accommodate real-time data such that the average data latency of each priority class is bounded by a pre-determined value with a specified confidence interval.

Based on this design criteria, formulation of the objective function and constraint equations for optimization of  $TRT$  parameters is suggested as follows;

$$\begin{aligned}
& \text{Minimize } \sum_{j=0}^M [\sigma_j'(\mathbf{TRT}) + \sigma_j''(\mathbf{TRT})] \\
& \text{Subject to } D_i(\mathbf{TRT}) \leq (w_i/c_i)\delta_i, \quad i = 0, \dots, K \\
& \mathbf{TRT} = [TRT_1, TRT_2, \dots, TRT_K]
\end{aligned} \tag{6.1}$$

where

$\sigma_j'(\mathbf{TRT}), \sigma_j''(\mathbf{TRT})$  = variance of the conditional effective service time for priority  $j$  class,

$M$  = bound of the priority level which accommodates real-time data,

$D_i(\mathbf{TRT})$  = average data latency of the priority  $i$  class,

$w_i$  = design safety factor for the priority  $i$  class,

$c_i$  = compensation coefficient for average data latency of the priority  $i$  class,

$\delta_i$  = bound of average data latency of the priority  $i$  class,

$K$  = the lowest priority level.

Using the notations in Chapter 3, the variances of conditional effective service times  $\sigma_j'$  and  $\sigma_j''$  are determined as follows;

$$\sigma_j' = \overline{T_j'^2} - \overline{T_j'}^2 \tag{6.2}$$

$$\sigma_j'' = \overline{T_j''^2} - \overline{T_j''}^2 \tag{6.3}$$

$w_i$  denotes the safety factor which allows a safety margin of network design.  $c_i$  is used to compensate the difference between analytical model and simulation experiment. However, only  $w_i/c_i$  is needed for optimization. A comparison with simulation experiments given in Chapter 5 shows that the analytical model generally underestimates data latency. Therefore, it is desirable to set  $w_i/c_i$  sufficiently

smaller than one to take into account a safety margin of network design and underestimation of the analytical model.

Since the objective and constraint equations are formulated as nonlinear functions, nonlinear programming technique offers a solution for this optimization problem. In this optimization problem, the constraint function (average data latency of the priority  $i$  class) is a combination of step functions with respect to the design variables ( $TRT_j$ ,  $j=1$  to  $K$ ), i.e., the constraint functions are not continuously differentiable with respect to the design variables. To avoid this problem, a multiplier method [44] is used in this thesis, which transforms the constrained optimization problem to unconstrained optimization problem. The unconstrained optimization problem is solved by using the Hook-Jeeve method [45] which is one of the multi-dimensional search methods that do not require derivatives. Detailed solution approach for the optimization problem is described in Appendix B.

## 6.2. An Example of ICCS Network Design

In this section, an example of ICCS network design is illustrated. The network is required to provide communication services to two types of subscribers, i.e., real-time and non-real-time data. Although occasional losses of real-time data packets can be tolerated, network-induced delays are critical for real-time operations and must not exceed specified bounds. In contrast non-real-time data do not have to be processed within specified time-constraints but need the assurance of accurate delivery. Thus, the real-time data should receive preferential treatment at the expense of the increased data latency of non-real-time data.

As shown in Figure 5.1 of Chapter 5, at low offered traffic, data latencies for all priority classes are almost unaffected by the change of  $TRT$  values. Thus, optimization of  $TRT$  have no significant bearing at low offered traffic. ICCS networks are not usually operated under heavy traffic because of network stability. An example of ICCS network design for the practical case of medium traffic ( $G=0.48$ ) is presented below.

**Example:** An ICCS network consists of four stations. Each station has four priority queues, priority 0, 1, 2 and 3, with 0 corresponding to the highest priority and 3 to the lowest. Priority 0 and 1 are assigned to accommodate real-time data, and priority 2 and 3 are to accommodate non-real-time data. Lengths of the priority 0, 1, 2 and 3 messages are packetized by 0.05 msec, 0.15 msec, 0.32 msec and 0.5 msec, respectively. Average message interarrival times for the priority 0, 1, 2 and 3 messages are 1.86 msec, 6 msec, 9 msec and 15 msec, respectively. Average data latencies for the priority 0, 1, 2 and 3 classes are assumed to be bounded to 0.25 msec, 0.47 msec, 0.9 msec and 1.6 msec, respectively. Design parameters  $w_i/c_i$  for all priority classes are set to 0.5.

The objective is to find the optimal  $TRT_i$  ( $i=1, 2$  and  $3$ ) values which minimizes the sum of data latency variances of real-time data such that the average data latency for each priority class is bounded to the values given above. ■

Using a computer program developed on the basis of the analytical model in Chapter 3 and the optimization algorithms in Appendix B, the optimal  $TRT$  for the priority 1, 2 and 3,  $TRT_1^*$ ,  $TRT_2^*$  and  $TRT_3^*$ , are determined as 0.2 msec, 0.09 msec and 0.05 msec, respectively.

Minimization of the sum of data latency variances of real-time data is examined by perturbing design variables  $TRT_i$  ( $i=1$  to 3). Figure 6.1 exhibits the change of the sum of data latency variances of real-time data which is obtained from the simulation model with respect to the perturbations of  $TRT_i^*$  ( $i=1$  to 3). The solid line with circle ( $\circ$ ) represents the change of the sum of data latency variances of real-time data when  $TRT_1$  is perturbed from 0.05 msec to 0.5 msec, while  $TRT_2^*$  and  $TRT_3^*$  are fixed to 0.09 msec and 0.05 msec, respectively. The dashed line with cross ( $\times$ ) shows the change of the sum of data latency variances of real-time data when  $TRT_2$  is perturbed from 0.05 msec to 0.5 msec, while  $TRT_1^*$  and  $TRT_3^*$  are fixed to 0.2 msec and 0.05 msec, respectively. The dotted line with triangle ( $\triangle$ ) represents the change of the sum of data latency variances of real-time data when  $TRT_3$  is perturbed from 0.05 msec to 0.5 msec, while  $TRT_1^*$  and  $TRT_2^*$  are fixed to 0.2 msec and 0.09 msec, respectively.

Similarly, the change of the objective function of this design procedure (sum of effective service time variances of real-time data) with respect to the perturbations of  $TRT_i^*$  ( $i=1$  to 3) is obtained from the analytical model, and illustrated in Figure 6.2. The lines and symbols which represent the perturbations of the objective function are the same as those described in Figure 6.1.

Figures 6.1 and 6.2 show similar pattern of change with respect to the perturbations of  $TRT_i^*$  ( $i=1$  to 3), i.e., when the sum of data latency variances of real-time data increases (decreases) the sum of effective service time variances of real-time data also increases (decreases). Thus, the the sum of effective service time variances is another candidate for the objective function of the preliminary

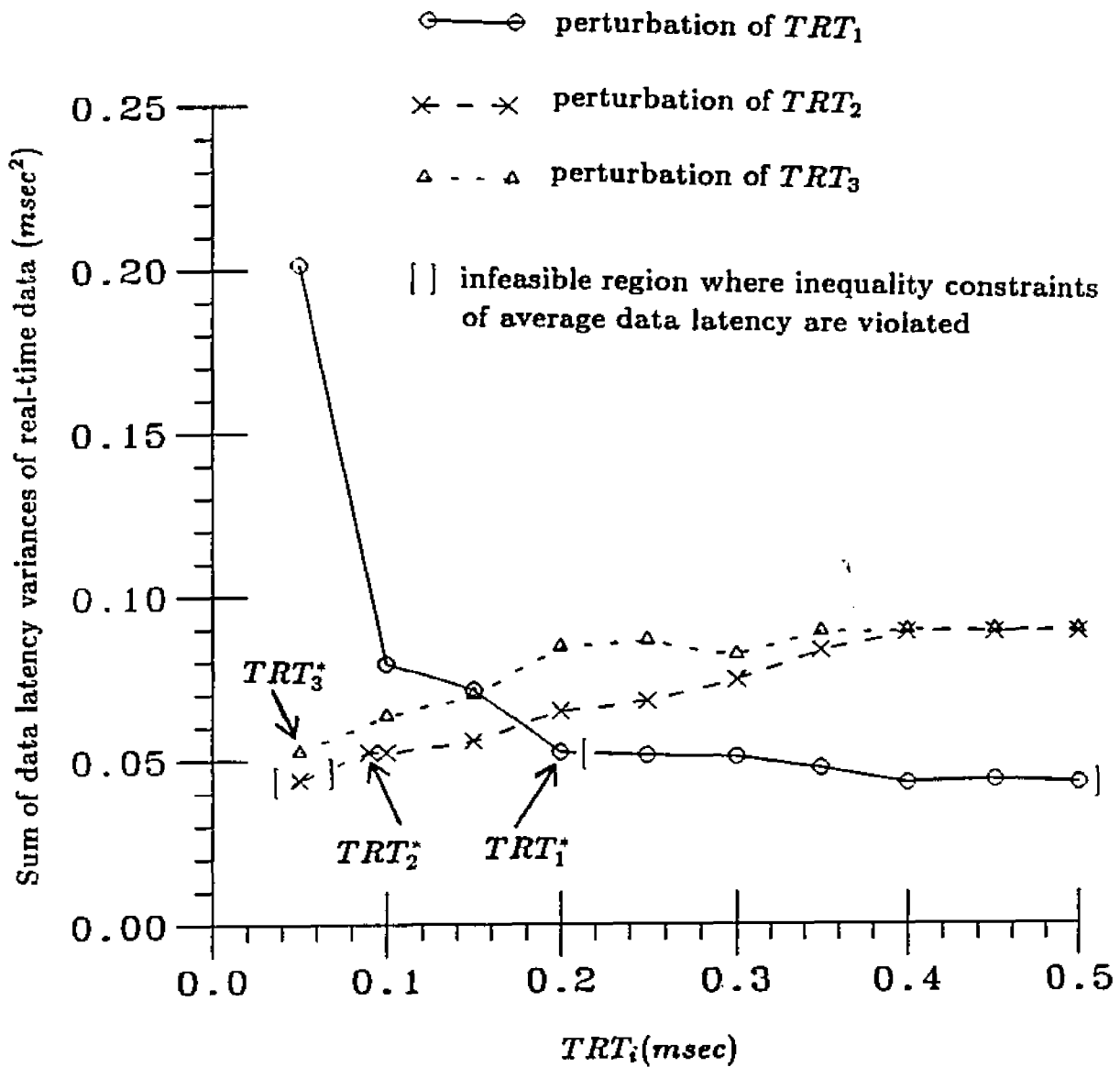


Figure 6.1: Perturbation of the Sum of Data Latency Variances of Real-Time Data with respect to  $TRT_i^*$ .



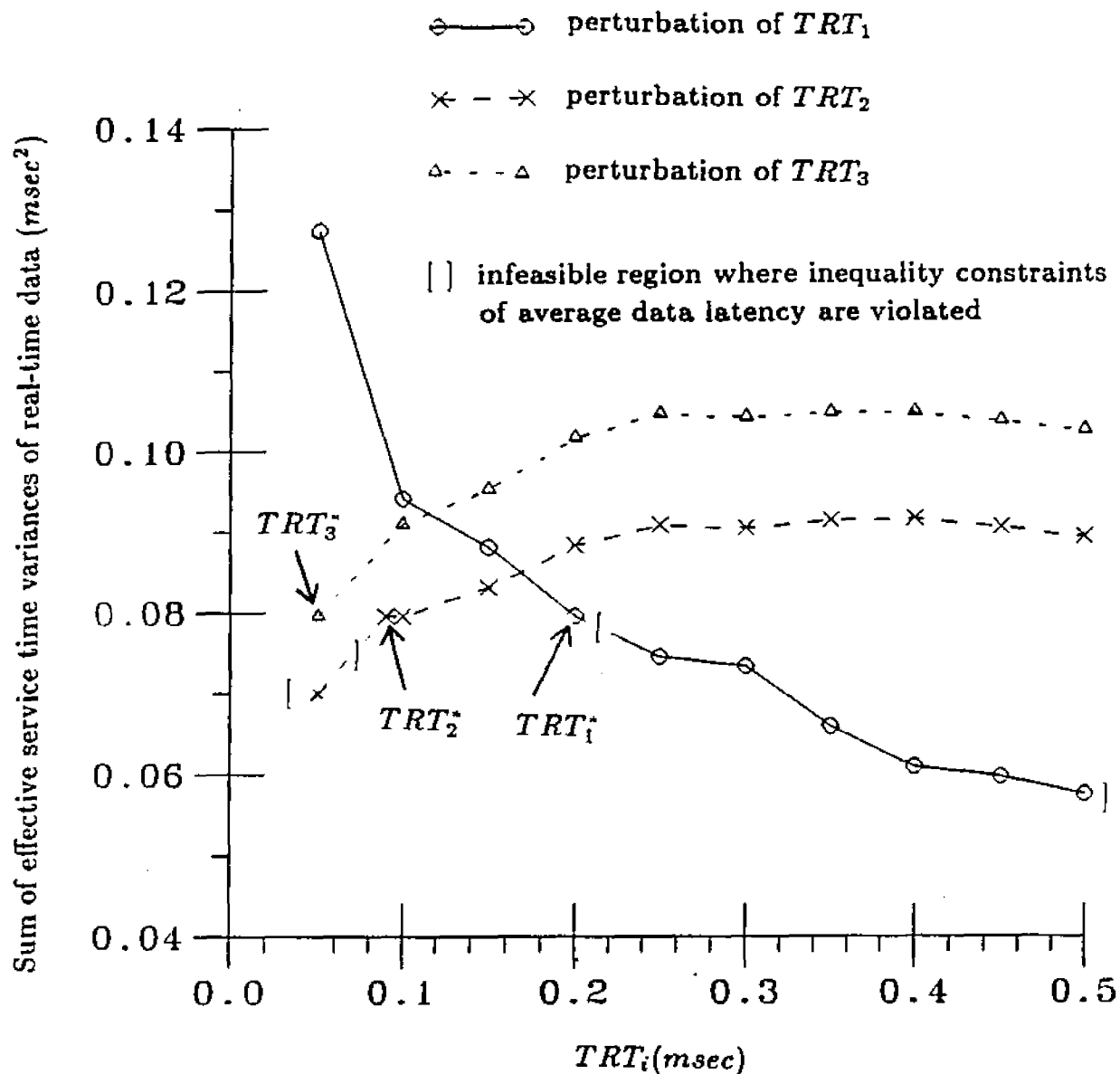


Figure 6.2: Perturbation of the Sum of Effective Service Time Variances of Real-Time Data with respect to  $TRT_i^*$ .

design of ICCS networks.

Figure 6.1 shows that, if  $TRT_1$  is set to a value which is greater than  $TRT_1^*$  (0.2 msec) and  $TRT_2$  is set to a value which is smaller than  $TRT_2^*$  (0.09 msec), the sum of data latency variances of real-time data is more reduced. However, these conditions violate the constraints of average data latencies which are imposed on the analytical model. Infeasible region where inequality constraints of average data latency given in (6.1) are violated (note that the design parameters  $w_i/c_i$  is set to 0.5) is marked in Figure 6.1.

$TRT_i^*$  must be very close to the actual optimal  $TRT_i$  values because, as shown in Figure 6.1,  $TRT_1^*$  and  $TRT_2^*$  are very close to the minimum points of the curves ( $TRT_3^*$  is located in the minimum point of the curve). The average data latencies for priority 0, 1, 2 and 3 at these  $TRT_i^*$  ( $i=1$  to 3) values are obtained from analytical and simulation models as follows:

Table 6.1: Average Data Latencies at Medium Traffic Load

priority	average data latency (msec)		$(w_i/c_i) \times$ bound of average data latency (msec)
	simulation	analysis	
0	$0.134 \pm 0.002$	0.125	0.125
1	$0.271 \pm 0.005$	0.233	0.235
2	$0.476 \pm 0.008$	0.443	0.45
3	$0.699 \pm 0.014$	0.799	0.8

The optimal  $TRT_i^*$  values satisfy the constraints for the average data latencies of all priority classes.

Figure 5.3 of Chapter 5 shows that the analytical model do not closely agree with the simulation results at high traffic. Optimization of  $TRT$  values at high traffic is illustrated as follows. Length of the priority 0, 1, 2 and 3 messages are the same as those given in the previous example. Average message interarrival times of the priority 0, 1, 2 and 3 messages are adjusted to 1 msec, 3 msec, 6.4 msec and 10 msec, respectively, and the offered traffic  $G$  is increased to 0.8. Average data latencies for the priority 0, 1, 2 and 3 classes are bounded to 0.354 msec, 0.72 msec, 2.4 msec and 4.4 msec, respectively. Design parameters  $w_i/c_i$  for all priority classes are set to 0.5. The optimal  $TRT$  of the priority 1, 2 and 3 classes for the given traffic condition were obtained as 0.13 msec, 0.1 msec and 0.03 msec, respectively.

For these optimal  $TRT$  values, the average data latencies of priority 0, 1, 2 and 3 at high traffic were determined from analytical and simulation models as follows:

Table 6.2: Average Data Latencies at High Traffic Load

priority	average data latency (msec)		$(w_i/c_i) \times$ bound of average data latency (msec)
	simulation	analysis	
0	$0.205 \pm 0.002$	0.176	0.177
1	$0.598 \pm 0.010$	0.359	0.36
2	$0.858 \pm 0.020$	1.188	1.2
3	$1.728 \pm 0.070$	2.164	2.2

A comparison of Tables 6.1 and 6.2 shows that, accuracy of the analytical model is not significantly reduced at high offered traffic from the perspectives of network design.

$TRT_i^*$  which are obtained from this design procedure are not actual optimal  $TRT_i$  values because our analytical model is approximate. Therefore, the design procedure provided above requires a careful examination before being applied for the final design of ICCS networks. However, this procedure can be used as an initial step for ICCS networks design, i.e., the  $TRT_i^*$ 's obtained from this design procedure can be used as initial values of perturbation analysis. The actual optimal  $TRT$  values should be obtained by perturbing  $TRT$ 's in the simulation model. Since the analytically determined  $TRT_i^*$ 's are expected to be close to the actual optimal  $TRT$  values, the number of comparison of alternative design points can be considerably reduced.

## Chapter 7

### SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

In this chapter, summary, conclusions and recommendations for future work are described in Sections 7.1, 7.2 and 7.3, respectively.

#### 7.1. Summary

The Integrated Communication and Control Systems (ICCS) network requires accommodation of heterogeneous traffic of real-time data and non-real-time data. Real-time data have a smaller and tighter upper bound on the data latency than that of non-real-time data. Thus, it should receive preferential treatment at the expense of the increased data latency of non-real-time data which can tolerate larger delays. The priority scheme provides different levels of privilege of medium access so that the data latency of each priority message remains within its bound.

The token bus protocol is one of the most widely accepted Medium Access Control (MAC) protocols for ICCS network because of its performance characteristics such as bounded data latency and high throughput. Token bus protocols also provide certain advantages such as flexibility of operations, evolutionary design process, and ease of maintenance, diagnostics and monitoring. In addition, token bus protocols have a priority mechanism which allows for the differential treatment for different message priorities.

Performance analysis of the priority scheme in token bus protocols is essential for effective design and operation of ICCS networks that use such protocols. This problem can be solved by using discrete-event simulation technique. However, the simulation technique suffers from heavy computations which prove to be costly and time-consuming. A mathematical model could save the cost of numerous simulation runs and provide more direct insight into the problem.

In this thesis, an analytical model which approximately evaluates the performance of the priority scheme is developed. From this analytical model, the relationships between the key network parameters, i.e., number of stations, message interarrival time, message length and priority timer, and the network performances, i.e., average queueing delay, average data latency and average queue length, have been determined for all priority classes.

The basic network-operating assumptions used in this thesis include the Poisson distribution of message arrival, constant message length, infinite queue capacity and single-service system (i.e., one message transmission at a time). The restriction of constant message length can be lifted if the complete statistics of the message length are available. Because of the mathematical intractability, approximation using the aforesaid assumptions of independence has been adopted. This implies that processes at each queue are independent of each other, and also the message waiting process and *TRT* expiration process at the same queue are independent.

The analytical model determines the probability that a message is served at the instant when the token arrives at a priority  $i$  queue. The moment genera-

tion functions of the conditional token circulation times for priority  $i$  class are determined on this basis. By using the inverse Laplace-Stieltjes Transformation, the probability density functions of the conditional token circulation times are obtained. The probability density function is integrated from 0 to  $TRT_i$  to determine the probability that the Token Rotation Timer is not expired when the token arrives at a priority  $i$  queue. Based on this probability and the conditional token circulation times, the first and second moments of the conditional effective service times of each priority class are obtained. The average queueing delay for each priority level is determined from the first two moments of the conditional effective service times. Average data latency and average queue length for each priority class are also determined along with the stability conditions of the network. As described in the Remark A.1 of Appendix A, any higher moments of the queueing delay at each priority level can be evaluated from the moment generation function of queueing delay given in (A.24) of Appendix A and the moment generation function of the conditional effective service time given in (3.59) of Chapter 3.

A Petri net model has been developed to investigate the structure and dynamic behavior of the priority scheme in token bus protocols under normal operating conditions. Based on the Petri net model, a simulation model is developed using SIMAN [42]. The analytical model is verified by the simulation experiments.

A comparison of the analytical model with simulation experiments exhibits that (1) accuracy of the analytical model is good for low to medium traffic, and is degraded at high traffic; and (2) accuracy of the analytical model is reduced as the priority level becomes lower. The first item is not a serious problem because ICCS

networks are seldom operated under heavy traffic because of network stability. The second item is not any major concern because a lower priority class usually accommodates non-real-time data which is not time-critical.

Albeit the fact that the analytical model is not exact, the trend of average data latency with respect to  $TRT$  is accurate for all priority levels. Therefore, the analytical model closely reflects the properties of the priority scheme in token bus protocols.

The analytical model developed in this thesis can be used as a preliminary design tool for ICCS networks. An approach for the initial phase of network design has been presented.

## 7.2. Conclusions

The major conclusions derived from the analytical and simulation models that have been developed in this thesis are given below;

1. The effects of priority scheme becomes more dominant as the traffic load increases.
2. The data latency of the priority  $i$  class decreases as  $TRT_i$  is set larger than a possible token circulation time.
3. For a given offered traffic, as the data latency of a priority  $i$  class decreases, the data latencies of all the other priority classes tend to increase.
4. In general, accuracy of the analytical model is good for low to medium traffic load, and monotonically deteriorates as the traffic increases.



5. Accuracy of the analytical model is degraded at high traffic load especially at the lowest priority. This is the effect of the independence assumptions.

### **7.3. Recommendations for Future Work**

The topics for future research are delineated below.

1. Formulation of the analytical relationship between queueing processes during a token circulation, and the relationship between successive token circulation times with respect to each queue. This is essential for improving the accuracy of the current analytical model.
2. Development of an analytical model with finite queue length. In an actual network system, each queue has a fixed queue capacity.
3. Extension of the analytical model for the exhaustive and gated service systems instead of the single-service system.
4. Development of a perturbation analysis model for prediction and optimization of ICCS network performance.
5. Development of a systematic methodology that provides an interface between the disciplines of the communication and control systems engineering for ICCS design.

## REFERENCES

1. Ray, A., "Networking for Computer-Integrated Manufacturing," IEEE Network, Vol. 2, No. 3, pp. 40-47, May 1988.
2. Tobagi, F. A., "Multiaccess Protocols in Packet Communication Systems," IEEE Transactions on Communications, Vol. COM-28, No. 4, April, 1988.
3. Sachs, R. A., "Alternative Local Area Network Access Protocols," IEEE Communications Magazine, Vol. 26, No. 3, March 1988.
4. Ray, A., Halevi, Y., Hong, S. H. and Lee, S., "Communication Networks for Autonomous Manufacturing and Process Control," Proc. ASME Conference on International Computers in Engineering, New York, NY, August 1987.
5. Ray, A., Hong, S. H., Lee, S., Egbelu, P. J., "Discrete-Event/Continuous-Time Simulation of Distributed Digital Control Systems," Transactions of the Society for Computer Simulation, Vol. 5, No. 1, January 1988.
6. SAE Linear Token Passing Multiplexed Data Bus Standard, 1987.
7. SAE high Speed Ring Bus Standard, Issue 1, November 1985.
8. MIL-STD-1553B Protocol Manual, United States Government Printing Office, February 1980.
9. Manufacturing Automation Protocol Reference Specification 2.2, Society of Manufacturing Engineers, Dearborn, MI, August 1986 and Draft of Version 3.0, July 1987.
10. ANSI/IEEE Standard 802.4 - 1985: Token Passing Bus Access Method and Physical Layer Specifications, IEEE, 1985.
11. Chlamtac, I., Ganz, A. and Koren, Z., "Prioritized Demand Assignment Protocols and Their Evaluation," IEEE Transactions on Communications, Vol. COM-36, No. 2, pp. 133-143, February 1988.
12. Choudhury, C. L. and Rappaport, S. S., "Priority Access Schemes Using CSMA /CD," IEEE Transactions on Communications, Vol. COM-33, No. 7, pp. 620-626, March 1982.

13. Shen, S. H. and Liu, T. H., "Dynamic CSMA/CD with a Priority Scheme," Proc. IEEE INFOCOM, pp. 258-263, April 1986.
14. Jayasumana, A. P., "Performance Analysis of a Token Bus Priority Scheme," Proc. IEEE INFOCOM, pp. 46-54, March 1987.
15. Jayasumana, A. P. and Fisher, P. D., "Performance Modeling of IEEE 802.4 Token Bus," Proc. IEEE-NBS Workshop on Factory Communication. pp. 221-252.
16. Rubin, I. and de Moraes, L. M., "Message Delay Analysis for Polling and Token Multiple-Access Schemes for Local Communication Networks," IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No.5, pp. 935-947, November 1983.
17. Ferguson, M. J. and Aminetzah, Y. J., "Exact Results for Nonsymmetric Token Ring Systems," IEEE Transactions on Communications, Vol. COM-33, No. 3, pp.223-231, March 1985.
18. Takagi, H., Analysis of Polling Systems, M.I.T. Press, Cambridge, MA, 1986.
19. Ferguson, M. J., "Computation of the Variance of the Waiting Time for Token Ring," IEEE Journal on Selected Areas in Communications, Vol. SAC-4, No. 6, pp. 775-782, September 1986.
20. Eisenburg, M., "Two Queues with Alternating Service," SIAM Journal of the Applied Mathematics, Vol. 36, No. 2, pp. 287-303, April 1979.
21. Ibe, O. C. and Cheng, X., "Approximate Analysis of Asymmetric Single-Services Token Passing Systems," Proc. IEEE International Communications Conference '87, Vol. 1, pp. 17.6.1-17.6.6, June 1987
22. Boxma, O. J. and Meister, B., "Waiting-Time Approximations for Cyclic-Service Systems with Switch-Over Times," Performance Evaluation, Vol. 7, No. 4, pp. 299-308, November 1987.
23. Kuehn, P. J., "Multiqueue Systems with Nonexhaustive Cyclic Service," The Bell System Technical Journal, Vol. 58, No. 3, pp. 671-689, March 1979.
24. Boxma, O. J. and Groenendijk, W. P., "Pseudo-Conservation Laws in Cyclic Service Systems," Journal of Applied Probability, Vol. 24, pp. 949-964, 1987.

25. Boxma, O. J. and Groenendijk, W. P., "Waiting times in Discrete-Time Cyclic-Service Systems," *IEEE Transactions on Communications*, Vol. COM-36, No. 2, pp. 164-170, February 1988.
26. Jayasumana, A. P. and Jayasumana, G. G., "Simulation and Performance Evaluation of 802.4 Priority Scheme," *Proc. IEEE/ACM Symposium on the Simulation of Computer Networks*, August 1987.
27. Piementel, J. R., "Performance Simulation of the IEEE Token Passing Bus Protocol Using SIMAN," *Proc. Workshop on Analytic and Simulation Modeling of IEEE 802.4 Token Bus LAN*, NBS, pp. 5-34, April 1985.
28. Sastry, A. P. and Atkinson, M. W., "Simulation of the IEEE 802.4 Token Passing Bus Using SIMSCRIPT," *Proc. Workshop on Analytic and Simulation Modeling of IEEE 802.4 token Bus LAN*, NBS, pp. 52-61, April 1985.
29. Janetzky, D. and Watson, K. S., "Performance Evaluation of MAP Token Bus in Conjunction with LLC Protocols," *Proc. IEEE-NBS Workshop on Factory Communication*.
30. Hashida, O. and Ohara, K., "Line Accommodation Capacity of a Communication Control unit," *Review of the Electrical Communication Laboratories, Nippon Telegraph and Telephone Public Corporation*, Vol. 20, pp. 231-239, March-April 1972.
31. Kleinrock, L., Queueing Systems, John Wiley & Sons, 1975
32. Peterson, J. M., Petri Net Theory and the Modeling of Systems, Prentice-Hall, 1981.
33. Diaz, M., "Modeling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Models," *Computer Network* 6, pp. 419-441, December 1982.
34. Gressier, E., "A Stochastic Petri Net for Ethernet," *Proc. Int. Workshop on Timed Petri Nets*, IEEE, Torino, Italy, July 1985.
35. Marsan, M. J., Chiola, G. and Fumagalli, A., "An Accurate Performance Model of CSMA/CD Bus Lan," Rozenberg, G. ed., "Advances in Petri Nets 1985," Vol. 222, Springer-Verlag, NY, 1986.

36. Proc. Int. Workshop on Timed Petri Nets, IEEE, Torino, Italy, July 1985.
37. Law, A. M. and Kelton, W. D., Simulation Modelling and Analysis, McGraw-Hill, 1982.
38. Schriber, T., Simulation Using GPSS, John Wiley, New York, 1974.
39. Bulgren, W. G., Discrete System Simulation, Prentice Hall, 1982.
40. Birtwistle, G., DEMOS-A System on Discrete Event Modeling on SIMULA, MacMillan, New York, 1979.
41. Pritsker, A. A. B. and Pegden, C. D., Introduction to Simulation and SLAM, Halstead Press, 1979.
42. Pegden, C. D., Introduction to SIMAN, System Modeling Corporation, State College, PA, 1986.
43. Banks, J. and Carson II, J. S., "Process-Interaction Simulation Language," *Simulation*, Vol. 44, No. 5, pp. 225-235, May 1985.
44. Bertsekas, D. P., "Multiplier Methods: A Survey," *Automatica*, Vol. 12, pp. 133-145, Pergamon Press, 1976.
45. Bazaraa, M. S. and Shetty, C. M., Nonlinear Programming: Theory and Algorithm, John Wiley & Sons, 1979.
46. Powell, M. J. D., "A Method for Nonlinear Constraints in Minimization Problems," in Optimization (R. Fletcher, ed.), Chapter 19, Academic Press, London and New York, 1969.

## Appendix A

### DERIVATION OF APPROXIMATE WAITING TIME FOR THE PRIORITY SCHEME

In this appendix, the expected value of waiting time at each priority level is derived. The analytical technique builds upon the concept of conditional effective service times which are assumed to be independent and identically distributed (i.i.d.) variables. The conditional cycle time in Kuehn's model [23] is replaced by the conditional effective service time. The assumptions and notations used in this appendix are same as those given in Chapter 3. In this thesis, only the state of the number of waiting messages of a particular priority  $i$  queue is considered. The analysis does not apply to continuous time but is restricted to a set of time epoches which are the scan instants and the departure instants of the token at the queue under consideration. The scan instant for priority 0 is defined as the instant when the token arrives at a priority 0 queue. For the priority 1 to  $K$  queues, the scan instant is defined as the instant when the token arrives at a priority  $i$  ( $i=1$  to  $K$ ) queue and the corresponding  $TRT_i$  is not expired. Departure instant defines the time when the token leaves the queue after serving a message.

The time intervals between the scan instants of a priority  $i$  queue are the conditional effective service times  $T'_i$  and  $T''_i$ ;  $T'_i$  is the effective service time during which the priority  $i$  queue under consideration does not transmit a message, and  $T''_i$  is the effective service time during which the priority  $i$  queue does transmit a message. The influence of all other queues on the queue length process in priority

$i$  queue is completely expressed by those effective service times. The following imbedded Markov chain solution is approximate since  $T_i'$  and  $T_i''$  are assumed to be independent and identically distributed (i.i.d.) variables.

The following analysis gives the results for a particular priority  $i$  queue,  $i=1$  to  $K$ . For a priority 0 queue,  $T_0=T_{r,0}$  and therefore Kuehn's approximation holds true [23].

### A.1. State Distribution at Scan Instants

Let  $J$  be the number of waiting messages at the scan instant of a priority  $i$  queue. The stationary distribution of  $J$  for priority  $i$  queues becomes

$$p_n^i = \Pr\{J = n\}, \quad n = 0, 1, 2, \dots, \quad i = 1, \dots, K \quad (A.1)$$

For brevity, the superscript  $i$  specifying the priority is omitted in subsequent steps until Section A.3.

Because of the memoryless property of the arrival process, the system state of the considered queue forms an imbedded Markov chain at the discrete set of scan instants (renewal points). If  $t_1, t_2, \dots$  are the successive scan instants,  $n_k$  is the state of the system at  $t_k$ . Then, by the theorem of total probability, it can be written as

$$\Pr\{n_{k+1} = n\} = \sum_{m=0}^{\infty} \Pr\{n_{k+1} = n | n_k = m\} \Pr\{n_k = m\} \quad (A.2)$$

$$(n = 0, 1, \dots; k = 1, 2, \dots)$$

The transition probability  $\Pr\{n_{k+1} = n | n_k = 0\}$  is

$$p_{0n} = \Pr\{n_{k+1} = n | n_k = 0\} = \int_0^{\infty} \frac{(\lambda_i t)^n}{n!} e^{-\lambda_i t} dT_i'(t), \quad (n \geq 0) \quad (A.3)$$

Similarly, when  $n_k = m > 0$ , the transition probability becomes

$$\begin{aligned}
 p_{mn} &= \Pr\{n_{k+1} = n | n_k = m\} \\
 &= \begin{cases} \int_0^\infty \frac{(\lambda_i t)^{n-m+1}}{(n-m+1)!} e^{-\lambda_i t} dT_i''(t), & \text{if } m \geq 0, n \geq m+1; \\ 0 & \text{otherwise.} \end{cases} \quad (A.4)
 \end{aligned}$$

Clearly, the distribution of the number of messages found at scan instants is different for each  $k$ , and the dependence of this distribution on  $k$  diminishes as  $k$  increases. On the basis of ergodic assumption, it follows that  $\lim_{k \rightarrow \infty} \Pr\{n_k = n\} = p_n$  should exist for the imbedded Markov chain if the system is stable. The distribution  $\{p_n\}$  is a statistical equilibrium distribution;  $p_n$  is the probability that an arbitrary scan instant found  $n$  messages in the priority  $i$  queue that has been operating for a sufficiently long period of time.

From (A.2) to (A.4), the stationary distribution of the number of waiting messages satisfies the equation

$$p_n = p_0 p_{0n} + \sum_{m=1}^{n+1} p_m p_{mn}, \quad n = 1, 2, \dots \quad (A.5)$$

Together with the normalizing condition

$$\sum_{n=0}^{\infty} p_n = 1 \quad (A.6)$$

the stationary probabilities of state at the scan instants are completely determined by the set of equations (A.3), (A.4), (A.5) and (A.6). The probability generating function of the state distribution  $p_n$ , ( $n = 0, 1, \dots$ ) is

$$G(x) = \sum_{n=0}^{\infty} p_n x^n. \quad (A.7)$$



Substituting (A.5) into (A.7),

$$G(x) = p_0 \sum_{n=0}^{\infty} p_{0n} x^n + \sum_{n=0}^{\infty} \sum_{m=1}^{n+1} p_{mn} p_m x^n \quad (\text{A.8})$$

Denotes the double sum in (A.8) by  $S$  as

$$S = \sum_{n=0}^{\infty} \sum_{m=1}^{n+1} p_{mn} p_m x^n \quad (\text{A.9})$$

By reversing the order of summation in (A.9),

$$\begin{aligned} S &= \sum_{m=1}^{\infty} p_m \sum_{n=m-1}^{\infty} p_{mn} x^n \\ &= \sum_{m=1}^{\infty} p_m \sum_{l=0}^{\infty} p_l x^{l-m+1} \end{aligned} \quad (\text{A.10})$$

where the second equality in (A.10) follows from the substitution  $l = n - m + 1$ .

Hence, (A.10) can be written as

$$\begin{aligned} S &= x^{-1} \left( \sum_{m=1}^{\infty} p_m x^m \right) \left( \sum_{l=0}^{\infty} p_l x^l \right) \\ &= x^{-1} [G(x) - p_0] \sum_{l=0}^{\infty} p_l x^l, \quad |x| \leq 1, x \neq 0. \end{aligned} \quad (\text{A.11})$$

Substituting (A.11) into (A.8),

$$G(x) = p_0 \sum_{n=0}^{\infty} p_{0n} x^n + x^{-1} [G(x) - p_0] \sum_{l=0}^{\infty} p_l x^l \quad (\text{A.12})$$

The first summation in the right hand side of (A.12) is

$$\begin{aligned} \sum_{n=0}^{\infty} p_{0n} x^n &= \sum_{n=0}^{\infty} \int_0^{\infty} e^{-\lambda_i t} \frac{(\lambda_i t x)^n}{n!} dT'_i(t) \\ &= \int_0^{\infty} e^{-tz} dT'_i(t) \\ &= \Phi'(z) \end{aligned} \quad (\text{A.13})$$

where  $z = \lambda_i(1 - x)$ .  $\Phi'(z)$  is the Laplace-Stieltjes transformation (LST) of the effective service time  $T'_i$ . Similarly, the second summation in the right hand side of (A.12) becomes

$$\sum_{l=0}^{\infty} p_l x^l = \Phi''(z) \quad (\text{A.14})$$

$\Phi''(z)$  is the LST of the effective service time  $T''_i$ . Transposing (A.13) and (A.14) into (A.12) and rearranging,

$$G(x) = p_0 \frac{x\Phi'(z) - \Phi''(z)}{x - \Phi''(z)} \quad (\text{A.15})$$

Note that  $G(x)$  is completely expressed by  $p_0$  and the LSTs of the two effective service times. Using the identity  $G(1) = 1$ ,  $p_0$  is obtained from (A.15) through evaluation of  $\lim_{x \rightarrow 1} G(x)$  by L'Hospital's rule

$$p_0 = \frac{1 - \lambda \overline{T''_i}}{1 - \lambda_i(\overline{T''_i} - \overline{T'_i})} \quad (\text{A.16})$$

The expected number of waiting messages at the scan instant of a priority  $i$  queue follows from

$$\bar{J} = \left. \frac{d}{dx} G(x) \right|_{x=1}$$

This results in

$$\bar{J} = p_0 \lambda_i \frac{\lambda_i \overline{T'_i}^2 (1 - \lambda_i \overline{T''_i}) + \overline{T'_i} (\lambda_i^2 \overline{T''_i}^2 + 2 - 2\lambda_i \overline{T''_i})}{2 - (1 - \lambda_i \overline{T''_i})^2} \quad (\text{A.17})$$

## A.2. State Distribution at Departure Instants

Let  $J^*$  be the number of waiting messages within the considered queue which are left behind by a departing message of that queue with distribution

$$p_n^* = \Pr[J^* = n], \quad n = 0, 1, \dots \quad (\text{A.18})$$

and generation function

$$G^*(x) = \sum_{n=0}^{\infty} p_n^* x^n \quad (A.19)$$

The probability  $p_n^*$  can be expressed through the probability of having  $m$  messages at the scan instant given that the considered queue is not empty,  $p_m/(1-p_0)$ , and the probability of  $n-m+1$  new arrivals in that queue during the subsequent transmission time of one message. Hence,

$$p_n^* = \sum_{m=1}^{n+1} \frac{p_m}{1-p_0} \int_0^{\infty} e^{-\lambda_i t} \frac{(\lambda_i t)^{n-m+1}}{(n-m+1)!} dL_i(t), \quad n = 0, 1, \dots \quad (A.20)$$

where  $L_i(t)$  is the probability density function (pdf) of message transmission time. Substituting (A.20) into (A.19) and interchanging the order of summation and integration,

$$G^*(x) = \frac{1}{(1-p_0)} \frac{G(x) - p_0}{x} \Phi_L(z) \quad (A.21)$$

where  $\Phi_L(z)$  is the LST of  $L_i$  and  $z = \lambda_i(1-x)$ .

Therefore, it follows for the expected number of messages at the departure instant:

$$\bar{J}^* = \left. \frac{d}{dx} G^*(x) \right|_{x=1} = \frac{\bar{J}}{1-p_0} - 1 + \lambda_i \bar{L}_i \quad (A.22)$$

$\bar{J}^*$  is also considered as the expected number of messages which arrived during the sojourn (waiting+transmission) time of the departing message.

### A.3. Waiting Time Analysis

Let  $W_i$  be the waiting time which an arbitrary message of the considered priority  $i$  queue has to undergo with pdf  $W(t)$  and LST  $\Phi_W(s)$ .  $p_n^{i*}$  defined in

Section A.2 can be alternatively considered as the distribution of the number of arriving message during the sojourn time  $D_i$  of the considered message. Since  $D_i = W_i + L_i$  and since  $W_i$  and  $L_i$  are independent of each other, the pdf of  $D_i$  is the convolution of  $W_i(t)$  and  $L_i(t)$ , symbolized by  $W_i(t) \otimes L_i(t)$ . Hence,

$$p_n^i = \int_0^\infty e^{\lambda_i t} \frac{(\lambda_i t)^n}{n!} d(W_i(t) \otimes L_i(t)), \quad n = 0, 1, \dots \quad (\text{A.23})$$

Applying (A.19) in (A.23),  $G_i^*(x)$  is determined as  $G_i^*(x) = \Phi_{W_i}(z)\Phi_{L_i}(z)$ , where  $z = \lambda_i(1 - x)$ , which, with (A.21), results in

$$\Phi_{W_i}(s) = \frac{1 - \lambda_i \overline{T_i''}}{\overline{T_i'}} \frac{1 - \Phi_i'(s)}{s - \lambda_i[1 - \Phi_i''(s)]} \quad (\text{A.24})$$

where  $\Phi_i'(s)$  and  $\Phi_i''(s)$  are the moment generating functions of the conditional effective service times  $T_i'$  and  $T_i''$ , respectively.

From (A.24), finally mean waiting time is determined as

$$\overline{W_i} = \left. \frac{d}{ds} \Phi_{W_i}(s) \right|_{s=0} = \frac{\overline{T_i'^2}}{2\overline{T_i'}} + \frac{\lambda_i \overline{T_i''^2}}{2(1 - \lambda_i \overline{T_i''})} \quad (\text{A.25})$$

Equation (A.25) reveals that the mean waiting time depends basically on the first and second moments of the conditional effective service times at each priority level.

**Remark A.1:** Any moment of the waiting time can be determined by differentiating (A.24), i.e.,

$$\overline{W_i^n} = \left. \frac{d^n}{ds^n} \Phi_{W_i}(s) \right|_{s=0} \quad (\text{A.26})$$

provided that  $\Phi_{W_i}(s)$  is analytic in the neighborhood of  $s = 0$ . The  $n$ -th moment given in (A.26) requires the computation of the first to  $(n + 1)$ -th moments of the conditional token circulation times  $T_i'$  and  $T_i''$ . These moments can be obtained (although it may be complex) by using the similar procedure given in the Proposition 3.1 and Proposition 3.2 of Section 3.2. ■

## Appendix B

### SOLUTION APPROACH FOR THE OPTIMIZATION PROBLEM

In Chapter 6, the objective and constraint equations for the optimal design of ICCS networks have been formulated. This appendix presents a solution approach for the optimization problem. Since the objective function and the constraint equations are nonlinear, nonlinear programming offers a solution for this optimization problem. The optimization problem given in Chapter 6 can be expressed as a typical nonlinear programming problem:

$$\text{Minimize } f(\mathbf{x})$$

$$\text{Subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \quad (B.1)$$

where  $\mathbf{x} \in R^n$  is the vector of  $n$  design variables,  $f$  and  $\{g_i\}$  are the objective and constraint functions, and  $m$  is the number of constraint functions.

Almost all the nonlinear programming problems are based on the following philosophy: Let  $\mathbf{x}_k$  be the design at the current  $k$ -th iteration. Then, a new design  $\mathbf{x}_{k+1}$  is found from the expression

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (B.2)$$

where  $\alpha_k$  is a step size parameter and  $\mathbf{p}_k$  is a direction vector. The procedure is continued until the optimality conditions are satisfied.

In the optimization problem of priority scheme given in Chapter 6, constraint functions are average data latencies of priority  $i$  ( $i=0$  to  $K$ ) classes. As described in Chapter 5, average data latency of the priority  $i$  class suddenly changes whenever the design variables  $TRT_k$  ( $k=1$  to  $K$ ) is set just beyond a possible token circulation time. Therefore, the constraint function (average data latency of the priority  $i$  class) is formulated as a combination of step functions with respect to the design variables ( $TRT_k$ ,  $k=1$  to  $K$ ). Primal methods such as recursive quadratic programming and gradient projection methods [45] cannot be used because these algorithms require computation of gradient vectors of the objective and constraint functions.

Transformation methods such as penalty function, barrier function [45] and multiplier methods [44] solve the constraint optimization problem given in (B.1) by transforming it into one or more unconstrained optimization problems. These unconstrained optimization problems are solved by using the multi-dimensional search methods such as cyclic coordinate and Hook-Jeeve methods [45], which do not use derivatives.

In the penalty function method, a penalty term is added to the objective function for any violation of the constraints. This method generates a sequence of infeasible points whose limit is an optimal solution to the original problem [45]. In the barrier function method, a barrier term that prevents the points generated from leaving the feasible region is added to the objective function. The method generates a sequence of feasible points whose limit is an optimal solution to the

original problem [45]. The transformation function is expressed as [45]

$$\psi(\mathbf{x}, r) = f(\mathbf{x}) + P(g(\mathbf{x}), r) \quad (B.3)$$

where  $r$  is a controlling parameter and  $P$  is a real-valued function whose action of imposing the penalty is controlled by  $r$ .

A drawback of these methods is that they may cause computational difficulties especially if  $r$  is large [44]. When  $r$  is large, the penalty and barrier functions tend to be ill-behaved near the boundary of the constraint where the optimum point usually lies. To alleviate these computational difficulties, *multiplier methods* [44] have been developed. In multiplier methods, there is no need for the controlling parameter  $r$  to go infinity. As a result, the transformation function  $\psi(\mathbf{x}, r)$  has good conditioning without singularities. Furthermore, multiplier methods are globally convergent and have been proved to process faster rates of convergence than penalty function or barrier function methods [44].

In this thesis, multiplier method is used to transform the constrained optimization problem given in Chapter 6 to unconstrained optimization problem. The penalty function  $P$  in (B.3) is formulated as follows [46];

$$P(g(\mathbf{x}), r, \theta) = 1/2 \sum_{i=1}^m r_i [(g(\mathbf{x}) + \theta_i)^+]^2 \quad (B.4)$$

where

$$(g(\mathbf{x}) + \theta_i)^+ = \max[0, (g(\mathbf{x}) + \theta_i)] \quad (B.5)$$

and  $r_i$  and  $\theta_i$  are the controlling parameters associated with the  $i$ -th constraint. The unconstrained optimization problem is solved by using the Hook-Jeeve method

[45] which is one of the multi-dimensional search methods that do not require derivatives. An algorithm for the method of Hook-Jeeve using line search is given in the following:

**Initialization Step:** Let  $\mathbf{d}_1, \dots, \mathbf{d}_n$  be the coordinate directions. Choose a scalar  $\varepsilon > 0$  to be used in terminating the algorithm. Choose starting point  $\mathbf{x}_1$ , let  $\mathbf{y}_1 = \mathbf{x}_1$ , let  $k = j = 1$ , and go to main step.

**Main Step:**

1. Let  $\lambda_j$  be an optimal solution to the problem to minimize  $f(\mathbf{y}_j + \lambda \mathbf{d}_j)$  subject to  $\lambda \in E_1$ , and let  $\mathbf{y}_{j+1} = \mathbf{y}_j + \lambda_j \mathbf{d}_j$ . If  $j < n$ , replace  $j$  by  $j + 1$ , and repeat step 1. Otherwise, if  $j = n$ , let  $\mathbf{x}_{k+1} = \mathbf{y}_{n+1}$ . If  $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon$ , stop; otherwise, go to step 2.
2. Let  $\mathbf{d} = \mathbf{x}_{k+1} - \mathbf{x}_k$ , and let  $\lambda^*$  be an optimal solution to the problem to minimize  $f(\mathbf{x}_{k+1} + \lambda \mathbf{d})$  subject to  $\lambda \in E_1$ . Let  $\mathbf{y}_1 = \mathbf{x}_{k+1} + \lambda^* \mathbf{d}$ , let  $j = 1$ , replace  $k$  by  $k + 1$ , and repeat step 1.



## VITA

Seung Ho Hong

I was born on May 31, 1956, in Seoul, Korea. I earned a B.S. degree in Mechanical Engineering at Yonsei University in 1982 and an M.S. degree in Mechanical Engineering at The Texas Tech University in 1985. The M.S. thesis research was directed at analyzing dynamic behavior of four-bar mechanical linkage which is flexibly supported and spring-constrained. The thesis title was "Dynamic Analysis of Spring-Constrained and Flexibly Supported Four-Bar Mechanical Linkage." During the Master program I worked as a teaching assistant for the course of Numerical Analysis in Mechanical Engineering.

I earned a Ph.D. degree in Mechanical Engineering at The Pennsylvania State University. My research areas during the Ph.D. program involved performance analysis and optimization of computer network systems, simulation of computer network performance and investigation of the effect of network-induced delay on the control systems. My thesis research was directed at developing an analytical model which evaluates the performance of the priority scheme in token bus protocols. During the Ph.D. program I worked as a research assistant. The research area included performance analysis and simulation of network protocols and control systems in the advanced aircraft (supported by Bendix Flight Systems Division).