Studies
on
Performance Evaluation of Token Ring Networks

Masayuki Murata

Osaka University
Osaka, Japan
February 1988

Studies

on

Performance Evaluation of Token Ring Networks

Masayuki Murata

Osaka University

Toyonaka, Osaka, Japan

February 1988

Studies

on

Performance Evaluation of Token Ring Networks

Masayuki Murata

Submitted in Partial Fulfillment of the

Requirement for the Degree of

DOCTOR of ENGINEERING

(Information and Computer Science)

at

Osaka University

Toyonaka, Osaka, Japan

February 1988

Preface

Recent advances in technology and diversity of local area networks have proved that they offer high speed data-exchanges or resource sharing located at a single site, such as office building, factory, campus, etc. A wide use of LANs has been accelerated by standard protocols, which are mainly developed by the Institute of Electrical and Electronics Engineers (IEEE), and partially by other international standards organization such as the European Computer Manufacture Association (ECMA). The IEEE 802 standard is concerned primarily with the physical and medium access control (MAC) layer of the Open System Interconnection. For the MAC layer protocol (such as CSMA/CD protocols and token passing ring/bus protocols), performance characteristics of those have been studied exhaustively in many literatures. Local area networks are now going into a new stage, i.e., the integrated service local area network which supports multifarious traffic. In such systems, various traffic is transmitted by a priority operation. We consider a new sort of a nonpreemptive priority queueing system in order to analyze token ring networks with priority operation. We also consider interconnected local area networks, in which local area networks are interconnected by a bridge or gateway as there is a limit to the number of stations that can be attached to a single network and to the distance the network can span. In this doctoral dissertation, we consider performance evaluation of the following three themes: (i) nonpreemptive reserved priority disciplines which are applicable to the token ring local area network with the priority operation which is adopted by the IEEE standard protocol, (ii) a two-layer model for local area networks which combines the Transport protocol and the token ring protocol which is one of the MAC layer protocol in the OSI Reference Model, and (iii) token ring networks which are interconnected by a finite capacity bridge.

First, we consider a new sort of nonpreemptive priority queueing systems which are applicable to the

reservation priority-mode operation of the token ring network. In the case where no switchover time is needed between two successive message services, we consider a system where the class of message to be served next is the highest priority class waiting at the beginning of the current message service. Where some switchover time is needed between services, we consider the three types of systems depending on the instance when the next service class is determined. That is, the next service class is given to the highest priority class message waiting (i) at the beginning of the current message service (called *type I*), (ii) at the end of the current service (*type II*), and (iii) at the end of the switchover time following the current message service (*type III*). In each type of systems, we obtain the mean message waiting time for each class. We also compare the degree of discrimination among the classes for each type.

Next, we build a two-layer performance model of token ring networks. Our performance model consists of a MAC (Media Access Control) layer submodel and Transport layer submodels. We employ a multiple-queue, cyclic-service model for the token passing MAC layer, and a closed queueing network model for the Transport layer. To deal with acknowledgement traffic with priority over data messages, we have revised a Mean Value Analysis (MVA) priority approximation for the case where the priority can change at each queue. We propose an iterative solution algorithm for this two-layer performance model. We apply our proposed method to the following five communication systems: a symmetric-load, piggybacked acknowledgement model, a symmetric-load, explicit acknowledgement model, full-duplex communication models (with and without priorities), and a client/server model. System performance measures such as throughput and mean message delay are derived from our analysis.

Lastly, we consider a token ring network consisting of two kinds of stations: single-buffer stations and a station with finite capacity. We first present a detailed analysis for this model. Next we give a simplified analysis for the same model based on the detailed analysis. Such an analysis makes it possible to obtain performance measures in the case of a middle/large number of terminals. We apply our methods to an interconnected token ring network model, each of which consists of single-buffer terminals and a finite capacity bridge. System performance measures (mean message waiting time, throughput and loss probability at the bridge) obtained through our analysis are compared to the simulation results. For almost all tried cases, our approximate analysis is shown to be within 95% confidence intervals in the mean message waiting time and within a few percent error in the throughput and loss probabiltiy.

Acknowledgements

I would like to express my sincerely appreciation to Professor Hideo Miyahara of Osaka University, who introduced me to the area of computer communications including the subjects in this thesis. I also thank his encouragement and invaluable comments in preparing this thesis.

I am heartily gratitude to Professor Toshio Fujisawa, Professor Tadao Kasami, Professor Koji Torii, Professor Nobuki Tokura, Professor Kenichi Taniguchi of Osaka University for their invaluable and patient suggestions.

Special thanks are due to the late Professor Kensuke Takashima and Dr. Takeshi Nishida. They were my supervisors when I was a under- and over-graduate school student of Osaka University. Their infectious enthusiasm had a monumental impact on my entering the area of the computer communications.

All works of this thesis would not have been possible without the support of Dr. Hideaki Takagi (Tokyo Research Laboratory, IBM Japan, Ltd.). It gives me great pleasure to acknowledge his assistance. He was constant sources of encouragement and advice throughout my studies and preparation of this manuscript.

Thanks are also due to many friends and colleagues in Tokyo Research Laboratory of IBM Japan and in the Department of Information and Computer Science of Osaka University who gave me useful suggestions.

To my parents, I give my gratitude for the principles they instilled in me and for the opportunities they afforded me. Finally, I would like to thank my wife, Teruko, for her careful proofreading of the thesis and for her infinite patience, encouragement and understanding.

Contents

List of Illustrations

Chapter 1. Introduction

1.1. Local Area Networks Overview

Local Area Networks (LANs) have proved effective in providing the vital link through which not only microcomputers, workstations, and the like, but also other sophisticated equipment such as data storage devices, printers, voice and video handling devices, etc, communicate with each other. This communication facility permits the implementations of electronic mail and the sharing of expensive resources at a single site, offices, laboratories, etc. Indeed, since the Ethernet was developed by Digital, Intel, and Xerox, numerous other LAN technologies have been proposed and implemented. For an introduction to LANs in general and related discussions on definitions, design, technologies and protocols, see, for example, [Cheo82] and [Stal84].

Leading organizations of LAN standardizations are the Institute of Electrical and Electronics Engineers (IEEE) project 802 and the European Computer Manufacture Association (ECMA) 24. They have adopted a LAN architecture model that describes the relationships of the LAN architecture and the Open System Interconnection (OSI) reference model [ISO83] as shown in Figure 1.1. In the LAN architecture model, the OSI data link layer is splitted into the two sublayers; Logical Link Control (LLC) and Medium Access Control (MAC) sublayers. The various type of LAN access protocols, such as CSMA/CD [IEEE85a], a token-passing bus protocol [IEEE85b] and a token ring protocol [IEEE85c], are then restricted to the MAC layer and the Physical layer, which provide necessary logic for gaining access to network for frame

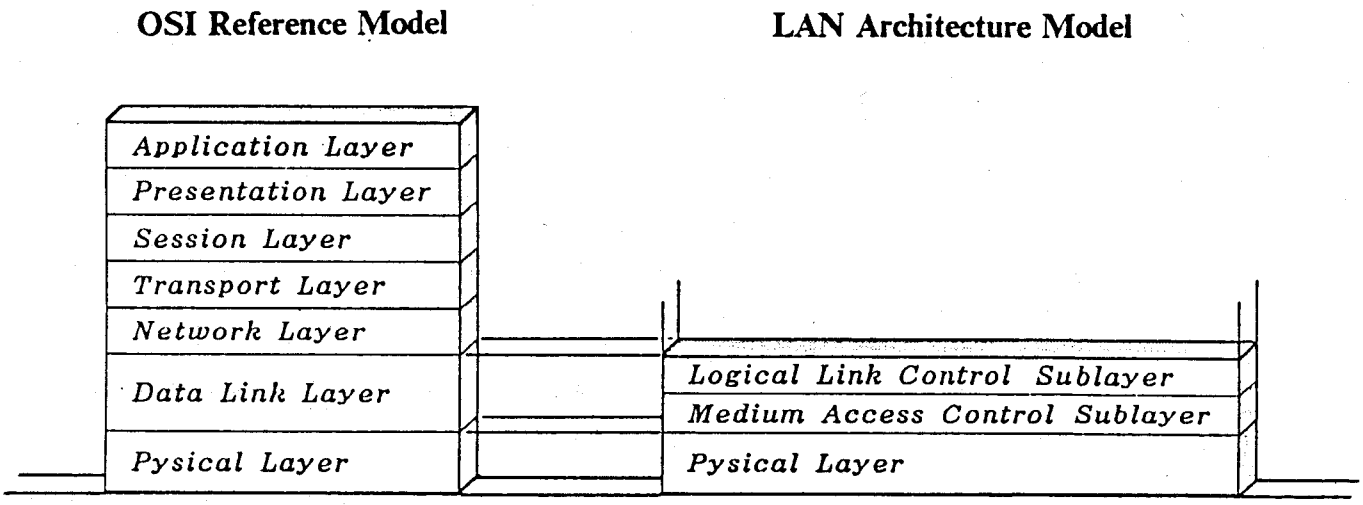
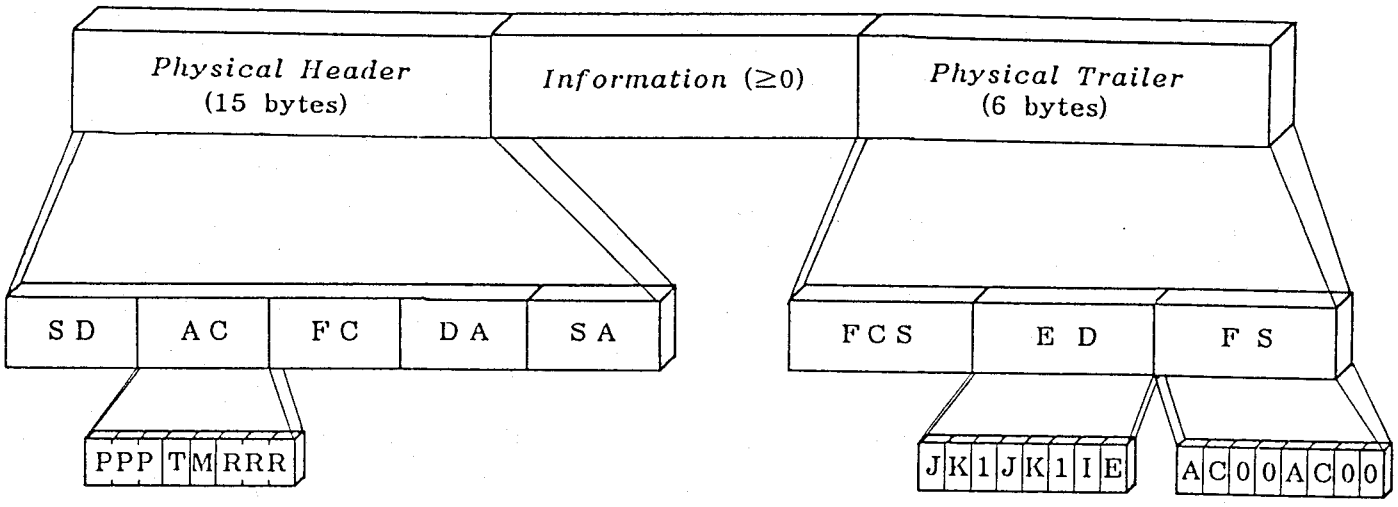


Figure 1.1. Relationships between OSI Reference Model and LAN Architecture Model

transmission and reception, and functions of data encapsulation (framing, addressing, error detection). On the other hand, the LLC sublayer is medium-independent and performs those functions normally associated with the Data Link layer in the OSI Reference Model, i.e., end-to-end error control, flow control and sequencing. The higher layers (Network layer through Application layer) provides the functions in much the same way as those in the OSI. Consequently, the quality of service in the LAN differs among different MAC protocols. From a viewpoint of performance, many literatures concerning with performance evaluation of access protocols (e.g., [Klei75] [Toba77] for CSMA/CD and [Taka86a] for token ring/bus protocols) and a comparison among them [Bux81b] have already been published. A tutorial survey for performance aspects of several standard protocols is found in [Bux84]. However, in the framework of OSI Architecture Layers Model, various performance studies have been devoted to each layer protocol as pointed out in [Reis86]. In the case of LANs, MAC layer protocols have been studied exhaustively because they characterize mainly the quality of service, i.e., delay-throughput characteristics in the LAN. However, only few works have been published for performance study where two or more layers are combined [Gihr86] [Bond87] [Mura87a] [Mura87b]. Such a performance model is important for network users to predict performance of network functions imbedded in their applications, which is one of the subjects in this thesis.

In parallel to the LAN standardizations, there has been a significant interest in the integration of various traffic (such as data, voice and image) and in expanding the methods and technologies accumulated in the LAN designs to Metropolitan Area Networks (MAN). In LAN systems, some priority-mode operation is required to support various traffic. For example, in integrated LAN systems, voice traffic should be transmitted with a higher priority because of its real-time transmission constraint while a lower priority may be assigned to the traffic in file transfers. Here, we briefly introduce a priority operation adopted by IEEE 802.5 for the token ring. The original token ring network (*without* any priority operation) is based on the use of a single token that circulates around the ring when all stations attached to the ring are idle. A station wishing to transmit must wait until it detects a *free* token passing by. It, then, changes the token from free to *busy*. The station transmits a frame immediately following the busy token. See Figure 1.2 for the frame format of the token ring network. The frame on the ring is removed by the transmitting station after the receiving station on the ring copies that frame. For more detailed descriptions



Starting Delimiter (SD): A unique 8-bit pattern used to start each frame.

Access Control (AC): Has the format 'PPPTMRRR' where 'PPP' and 'RRR' are 3-bit priority and reservation variables. 'M' is the monitor bit and 'T' indicates whether this is a token or data frame. In the Case of a token frame, the only additional is ED.

Frame Control (FC): Indicates whether this is an LLC data frame.

Destination Address (DA): Specifies the station(s) for which the frame is intended (6 bytes).

Source Address (SA): Specifies the station which sent the frame (6 bytes).

Frame Check Sequence (FCS): A 32-bit cyclic redundancy check value.

Ending Delimiter (ED): Contains the Intermmediate Frame (I) bit and the Error Detection (E) bit.

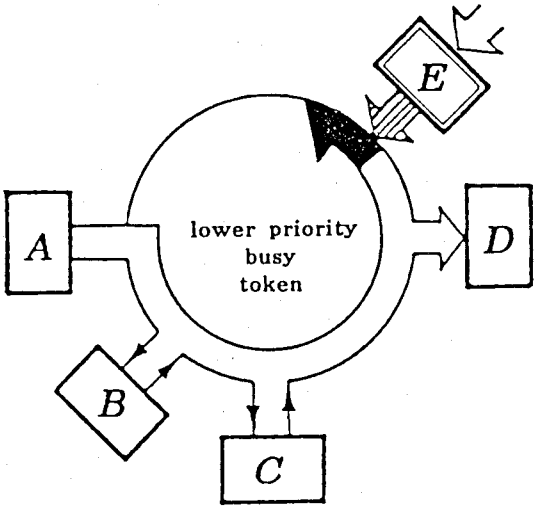
Frame Status (FS): Contains the Address Recognized (A) bits and the Frame Copied (C) bits, which are duplicated because they are not protected by FCS.

Figure 1.2. Token Ring Network Frame Format

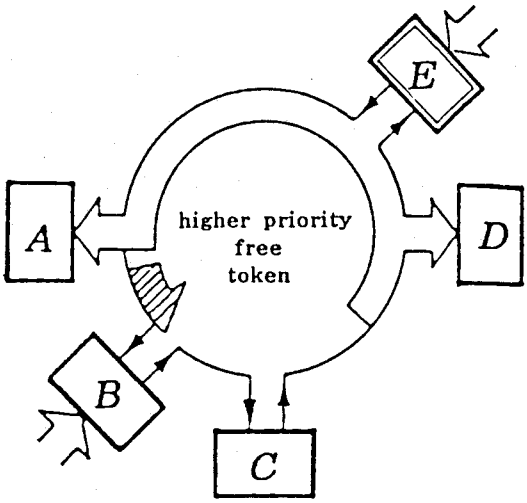
of the token ring protocol and discussions on other considerations such as reliability, the readers refer to [Bux81b], [Dixo83] and [Stro83].

The priority-mode operation for the token ring can be realized by modifying the original token ring protocol as follows. The Priority (**PPP**) and reservation (**RRR**) indicators are used in realizing the access mechanism. Each station may be assigned a priority level according to the traffic generated at that station. Selected stations with same priority levels can capture any free token that has a priority level setting equal to or less than those assigned priority. When the token is busy, the requesting station may set its priority in the reservation field of the header if the priority of that station is higher than the current reservation request. The current transmitting station must examine the reservation field and release the next free token with the same priority level as the reservation field. A requesting station can, then, use the free priority token if the priority level is equals to or less than its priority. Other stations assigned that priority can also transmit a frame when that free token passes by to implement a uniform access among stations with same priority level. When the station, which originally released the priority free token, capture a free token at that priority, it releases a free token at the original priority level. Figure 1.3 shows an example of such a priority-mode operation of the token ring [Stro83].

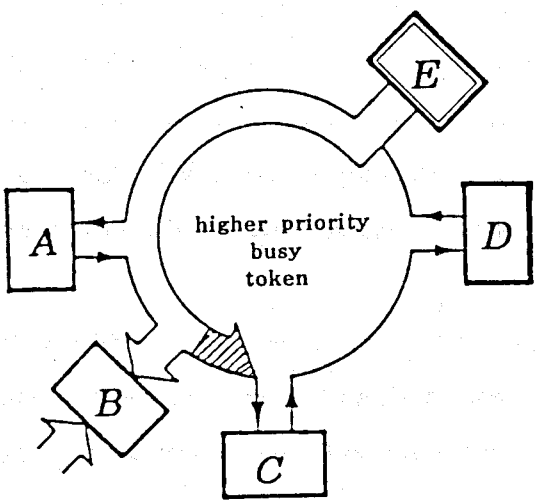
From a performance's point of view, the token ring protocol (and the token-passing bus protocol) has been modelled by a multiple-queue cyclic-service queueing systems with (server's) switchover times, or the *polling system* for short. Various analytical results are available for this type of queueing systems. In this kind of systems, three types of service for each station have been considered: (i) exhaustive, (ii) gated and (iii) limited. In an exhaustive service system, each station is served until it empties before the server switches over to the next station. In a gated service system, only those messages which are found at the instant of polling are served in the current round. In the limited (nonexhaustive) service system, a station is served until either 1) it empties, or 2) the first fixed number, say L, of messages are served, whichever occurs first. For the first two services, exact analytical results have been already obtained by [Ferg85]. For the last service, approximate results are available in several papers (e.g., [Kueh79] and [Boxm86]), which is the IEEE standard by setting $L = 1$ for the normal operation of the token ring network protocol.



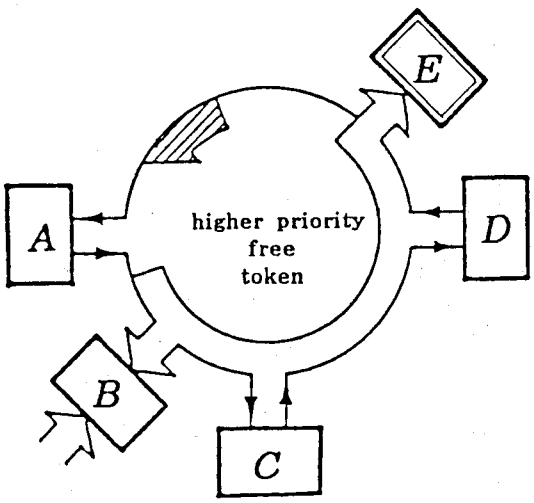
(1) *A* is sending the frame to *D*. *E* makes a higher priority level reservation.



(2) *A* generates a higher priority level free token and saves the original priority level. *B* assigned a lower priority level cannot use the higher level token at this point.



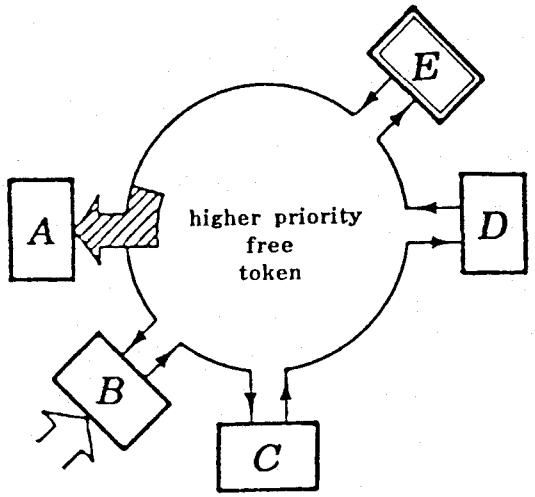
(3) *E* sends the frame to *B* using the free token at a higher priority level.



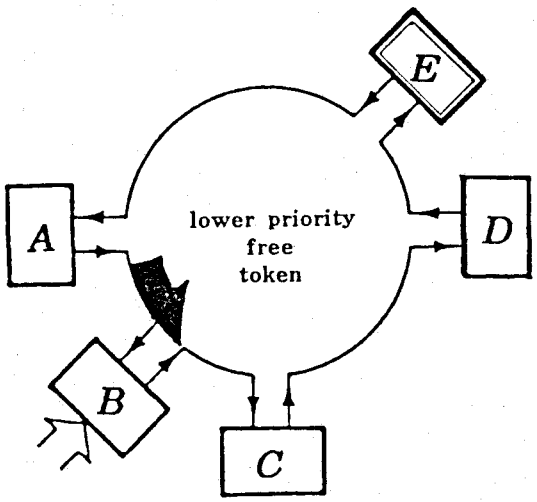
(4) *E* generates a free token at a current (higher) priority level.

The station *E* is assigned the higher priority level while stations *A*, *B*, *C* and *D* are assigned the lower priority level in the figure.

Figure 1.3. An Example of Token Ring Operation with Priority



(5) *A* receives the higher priority level *free* token and, then, generates a free token at the saved preempted (lower) priority level.



(6) *B* can now send the frame by using the free token at a lower priority level.

Figure 1.3. An Example of Token Ring Operation with Priority (Cont'd)

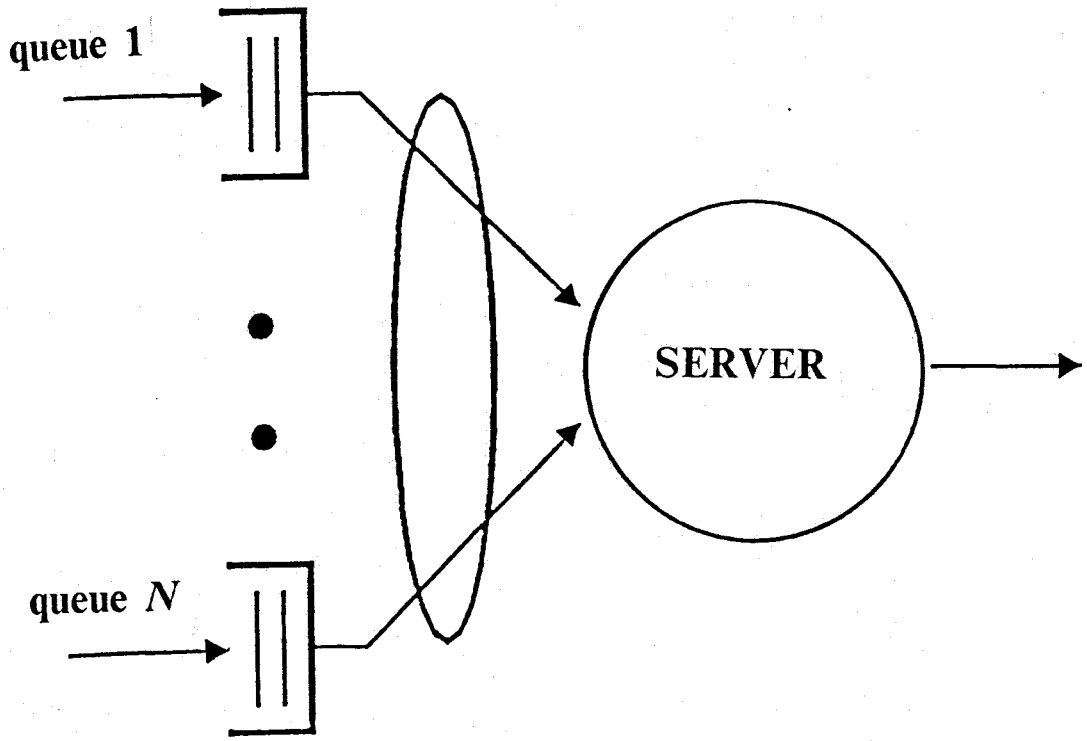


Figure 1.4. Model for Polling Systems

However, there is very little work on the performance evaluation of the priority-mode operation except [Taka86b] and [Mura86], which is the basic motivation of the study done in Chapter 2.

LANs are truly effective in permitting high-speed communication between devices located within a building or building complex. However, there is a limit to both the number of devices that can be attached to the network and the distance the network can span. The introduction of interconnected LANs where two or more LANs are interconnected through a *gateway* or a *bridge* is one of most effective solutions to relax such limitations. The gateway is a station with special functions such as routing and store-and-forward operation for inter-messages. Note that the term *bridge* is generally used for the gateway which interconnects homogeneous networks, i.e., networks with identical MAC layer protocols. In interconnected LANs, some priority operation is required to give a higher priority to a gateway than other stations. For the priority-mode operation in interconnected token ring networks, two methods can be considered [Bux85]. One is the token reservation scheme which is already described earlier in this chapter. Another scheme is that the bridge exhaustively transmits frames in its buffer while other stations transmit at most one frame per token possession as in the case of an ordinary operation. For exhaustive service, we can use an Intermediate Frame (I) bit which indicates that one more frames follow the current frame. The latter approach is considered in [Taki86] and [Mura87c]. In Chapter 4, we consider such a priority-mode operation to interconnect the token ring networks and evaluate performance of such a system.

1.2. Outline of the Dissertation

As discussed in the previous section, LANs are now proceeding to a new stage. In this thesis, we treat the following three subjects.

In Chapter 2, we consider a new sort of nonpreemptive priority queueing systems which are applicable to the reservation priority-mode operation of the token ring network. As described in the previous section, when the reservation priority-mode operation is used in the token ring network, the priority class to be

served next is determined as the highest priority class that exists at the moment when the header of the current message is transmitted on the ring. We consider such queueing systems that the service is given to the message with highest priority class which have already arrived at the beginning of the current message service. First, we consider the case with no switchover times between two successive message services, which corresponds to zero walking times between adjacent stations in token ring networks. Next, we treat the case where the finite switchover time is required between two successive message services. In that case, the three types of systems are considered depending on the time when the next service is reserved, i.e., the next service class is given to the highest priority class existing (i) at the beginning of the current message service, (ii) at the end of the current message service, and (iii) at the end of the switchover times. We call those systems *type I*, *type II* and *type III*, respectively. Note that the type I system corresponds to the above-mentioned reservation priority-mode operation of the token ring networks with finite switchover times. For each type of systems, we obtain the mean message waiting time for each class and compare the degree of discrimination among the classes.

Next, we build a two-layer performance model of token ring networks in Chapter 3. Our performance model consists of a MAC (Media Access Control) layer submodel and Transport layer submodels. We employ a multiple-queue, cyclic-service model for the token passing MAC layer, and a closed queueing network model for the Transport layer. To deal with acknowledgement traffic with priority over data messages, we revise an MVA priority approximation [Brya84] for the case where the priority can change at each queue. We propose an iterative solution algorithm for this two-layer performance model, and apply our proposed method to the following five communication systems: a symmetric-load, piggybacked acknowledgement model, a symmetric-load, explicit acknowledgement model, full-duplex communication models (with and without priorities), and a client/server model. System performance measures such as throughput and mean message delay in those five models are derived.

In Chapter 4, we consider a token ring network consisting of two kinds of stations: single-buffer stations and a station with finite capacity. First a detailed analysis for this model is derived. Next we give a simplified analysis which makes it possible to treat a middle/large number of terminals because the detailed analysis does not allow such a range of parameters by its excessive computational time and space. Our

methods is, then, applied to an interconnected token ring network model, each of which consists of single-buffer terminals and a finite capacity bridge. System performance measures (mean message waiting time, throughput and loss probability at the bridge) obtained through our analysis are compared to the simulation results.

Chapter 5 is devoted to concluding remarks and future research topics.

Chapter 2. Queueing Analysis of Nonpreemptive Reserved Priority Disciplines

2.1. Introduction

The study of nonpreemptive priority queueing systems in this chapter was motivated by the priority-mode operation of the token ring protocols in LANs. If there is no priority structure among stations on the token ring, after a station transmits a frame (called *message* below), the transmission rights are transferred to the next station in the downstream. In the priority mode, a higher priority station A (say) reserves transmission rights in the header of a message (sent by a lower priority station B (say)) when the header passes through A. Thus, unless yet higher or equal priority stations override the reservation, the next transmission right is given to A. Only after all transmissions from higher priority stations have been completed, the transmission right is passed to the station in the immediate downstream of B.

An interesting feature involved in the above-mentioned priority-mode operation is that the priority class to be served next is determined as the highest priority class that exists at the beginning of the current service time, i.e., at the moment when the header of the current message is transmitted. This is different from the classical nonpreemptive head-of-line priority queueing system (e.g., [Cox61], [Klei76], [Haye84]) where the next service class is determined at the end of the current service. Another aspect in the token ring operation is that a finite time is required to switch the service from one message to another (actually, it consists of the signal propagation delay and the sum of bit latencies at each station). For systems with

service switchover times, we consider the three types of systems depending on the time when the next service class is determined. That is, the next service is given to the highest priority class message waiting (i) at the beginning of the current message service, (ii) at the end of the current message service, and (iii) at the end of the switchover time following the current service. We call these three systems *type I*, *type II* and *type III*, respectively.

The purpose of this chapter is to derive the mean message waiting time in each type of reservation disciplines with/without switchover times. Note that when there are no switchover times type II and III systems reduce to the classical head-of-line queueing model. Also, the mean waiting time for type III with switchover times can be found by using the result by [Heym69] for a nonpreemptive priority $M/G/1$ queue with server's vacations.

2.2. Model without Service Switchover Times

We assume that arriving messages belong to one of a set of P different priority classes, indexed by subscript p ($p = 1, 2, \ldots, P$). Our convention is that the smaller the index value associated with a class is, the higher is the priority of that class. We assume that messages from priority class p arrive in a Poisson stream at rate λ_p (independent of arrival processes of other classes). We define the total arrival rate by

$$\lambda \triangleq \sum_{p=1}^{P} \lambda_p \,. \tag{2.1}$$

The service time distribution of each message is assumed to be identical for all priority classes; let $B^*(s)$, b and $b^{(2)}$ be the LST (Laplace-Stieltjes transform) of the distribution function, mean and the second moment, respectively, for the service time. The above definitions are used throughout the paper.

The service discipline is given as follows. Let us consider the successive i th and $i + 1$ st services administered to messages of all classes. The class of messages to which the $i + 1$ st service is given is the

highest priority class (if any existing) at the beginning of the i th service. If there are no messages waiting at the beginning of the i th service, a message from the highest priority class existing at the end of the i th service is chosen for the $i + 1$ st service. If there are no messages in the system at the end of the i th service (in this case, the server becomes idle), a message which arrives first gets the $i + 1$ st service. Each service is given in nonpreemptive fashion. In Section 2.2, we assume no switchover times between successive services. Thus our system is work-conservative [Klei76].

We are interested in the mean message waiting time W_p (the mean time from the arrival instant to the beginning of service) for arbitrary message of priority class p ($p = 1, 2, \ldots, P$). Since we have a nonpreemptive work-conserving system, Kleinrock's conservation law [Klei76] must be satisfied; in our case, it takes the form of the load-weighted mean:

$$\overline{W} \triangleq \sum_{p=1}^{P} \frac{\lambda_p}{\lambda} W_p = \frac{\lambda b^{(2)}}{2(1 - \lambda b)} . \tag{2.2}$$

2.2.1. Number of Messages at Service Start Instants

Let n_{ip} be the number of class p messages in the waiting room at the beginning of the i th service. Let a_p be the number of class p messages to arrive during a service time. We now express $n_{i+1,p}$ in terms of n_{ip} and a_p in the following $P + 1$ mutually exclusive and exhaustive cases.

- *Case* p ($p = 1, 2, \ldots, P$): $n_{ij} = 0$ for $j = 1, 2, \ldots, p - 1$, and $n_{ip} > 0$.

In this case, the $i + 1$ st service is given to a class p message which exists at the beginning of the i th service. We have

$$n_{i+1,j} = \begin{cases} a_j & j = 1, 2, \ldots, p - 1 \\ n_{ij} - 1 + a_j & j = p \\ n_{ij} + a_j & j = p + 1, p + 2, \ldots, P \end{cases} \tag{2.3}$$

• *Case* 0: $n_{ij} = 0$ for $j = 1, 2, \ldots, P$.

In this case, there are no messages in the waiting room at the beginning of the i th service. This case is subdivided into the following $P + 1$ mutually exclusive and exhaustive subcases.

- *Subcase* 0-p ($p = 1, 2, \ldots, P$): $a_j = 0$ for $j = 1, 2, \ldots, p - 1$, and $a_p > 0$.

In this case, there are no arrivals of class j messages for $j = 1, 2, \ldots, p - 1$, and there is at least one arrival of class p messages during the i th service time. It follows that $i + 1$ st service is given to a class p message. Thus,

$$n_{i+1\,j} = \begin{cases} 0 & j = 1, 2, \ldots, p - 1 \\ a_j - 1 & j = p \\ a_j & j = p + 1, p + 2, \ldots, P \end{cases} \tag{2.4}$$

- *Subcase* 0-0: $a_j = 0$ for $j = 1, 2, \ldots, P$.

In this case, there are no arrivals of any class messages during the i th service time, and there is an idle time between the i th and $i + 1$ st services. The $i + 1$ st service is given to the first arriving message of any class during the idle time. We have

$$n_{i+1\,j} = 0 \qquad j = 1, 2, \ldots, P \tag{2.5}$$

Let us denote by π_p the probability of *case* p above ($p = 0, 1, \ldots, P$). On the condition that *case* 0 occurs, the probability of *subcase* 0-p is given by

$$\begin{aligned} &\text{Prob}\,[\textit{subcase}\ 0\text{-}p \mid \textit{case}\ 0] = B^*\Big(\sum_{j=1}^{p-1}\lambda_j\Big) - B^*\Big(\sum_{j=1}^{p}\lambda_j\Big), \qquad p = 1, 2, \ldots, P \\ &\text{Prob}\,[\textit{subcase}\ 0\text{-}0 \mid \textit{case}\ 0] = B^*(\lambda)\,. \end{aligned} \tag{2.6}$$

Note that in *subcase* 0-0 the $i + 1$ st service is given to a class p message with probability λ_p / λ with

which a class p message arrives breaking the idle time. Now, we may equate the probability that a randomly chosen service is given to a class p message (this occurs in *case p*, in *subcase* 0-p, and in *subcase* 0-0 with probability λ_p / λ) to λ_p / λ which is the long-time fraction of class p messages to all messages:

$$\pi_p + \pi_0 \left[\frac{\lambda_p}{\lambda} B^*(\lambda) + B^*\left(\sum_{j=1}^{p-1} \lambda_j\right) - B^*\left(\sum_{j=1}^{p} \lambda_j\right) \right] = \frac{\lambda_p}{\lambda}, \qquad p = 1, 2, \ldots, P. \tag{2.7}$$

To find π_0, we observe that the system becomes empty at the end of a service if and only if there are no waiting messages at the beginning of the service and there are no arrivals during the service time. However, the number of all messages in our system is equivalent to that of an $M/G/1$ queue with arrival rate λ. Therefore, the probability that our system becomes empty at the end of a service is equal to the probability that the system is empty at any instant. Thus, we have

$$\pi_0 B^*(\lambda) = 1 - \lambda b. \tag{2.8}$$

Solving (2.7) and (2.8), we get

$$\begin{aligned} \pi_0 &= (1 - \lambda b) \,/\, B^*(\lambda) \\ \pi_p &= \lambda_p b - (1 - \lambda b) \left[B^*\left(\sum_{j=1}^{p-1} \lambda_j\right) - B^*\left(\sum_{j=1}^{p} \lambda_j\right) \right] \,/\, B^*(\lambda), \qquad p = 1, 2, \ldots, P. \end{aligned} \tag{2.9}$$

We proceed to express the relationship in (2.3)-(2.5) using the steady-state probability generating function $G(z_1, z_2, \ldots, z_P)$ for the distribution of n_{ip} $(p = 1, 2, \ldots, P)$, which is defined by

$$G(z_1, z_2, \ldots, z_P) = \lim_{i \to \infty} E\left[\prod_{p=1}^{P} (z_p)^{n_{ip}} \right]. \tag{2.10}$$

We may express π_p in terms of $G(\bullet)$ as

$$\begin{aligned} \pi_0 &= G(0, \ldots, 0) \\ \pi_p &= G(0, \ldots, 0, 1, 1, \ldots, 1) - G(0, \ldots, 0, 0, 1, \ldots, 1), \qquad p = 1, 2, \ldots, P. \end{aligned} \tag{2.11}$$

On the right-hand side of this equation, 1 first appears in the p th and $p+1$ st positions in the arguments of $G(\bullet)$ in the first and second terms, respectively. The relationship in (2.3)-(2.5) is now expressed as

$$G(z_1, z_2, \ldots, z_P) = \left\{ \sum_{p=1}^{P} \frac{1}{z_p} \left[G(0, \ldots, 0, z_p, z_{p+1}, \ldots, z_P) - G(0, \ldots, 0, 0, z_{p+1}, \ldots, z_P) \right] \right\} \bullet B^*\left[\sum_{p=1}^{P} \lambda_p (1 - z_p) \right]$$
$$+ G(0, \ldots, 0) \left[B^*(\lambda) + \sum_{p=1}^{P} \frac{1}{z_p} \left\{ B^*\left[\sum_{j=1}^{p-1} \lambda_j + \sum_{j=p}^{P} \lambda_j (1 - z_j) \right] - B^*\left[\sum_{j=1}^{p} \lambda_j + \sum_{j=p+1}^{P} \lambda_j (1 - z_j) \right] \right\} \right]. \tag{2.12}$$

This is the governing equation of our system.

Let us find the mean number N_p of class p messages in the waiting room at the beginning of service, which is given by

$$N_p = \left[\frac{dG(1, \ldots, 1, z, 1, \ldots, 1)}{dz} \right]_{z=1}, \qquad p = 1, 2, \ldots, P \tag{2.13}$$

where z appears as the p th argument of $G(\bullet)$. To do this, we first differentiate (2.12) twice with respect to z_p, and then set $z_j = 1$ for $j = 1, 2, \ldots, P$ to obtain (after some manipulation)

$$0 = (1 - \lambda_p b)\pi_p - \left[\frac{dG(0, \ldots, 0, z, 1, \ldots, 1)}{dz} \right]_{z=1} + \lambda_p bN$$
$$+ \frac{1}{2} \lambda_p^2 b^{(2)} + \pi_0 \left[B^*\left(\sum_{j=1}^{p-1} \lambda_j \right) - B^*\left(\sum_{j=1}^{p} \lambda_j \right) + \lambda_p B^{*(1)}\left(\sum_{j=1}^{p-1} \lambda_j \right) \right], \qquad p = 1, 2, \ldots, P \tag{2.14}$$

where z appears as the p th argument of $G(\bullet)$, and $B^{*(1)}(z) = dB^*(z) / dz$. In the case $p = 1$ (the highest priority class), from (2.14) we have explicitly

$$N_1 = \frac{\lambda_1{}^2 b^{(2)}}{2(1-\lambda_1 b)} + \pi_1 + \pi_0 \frac{1-\lambda_1 b - B^*(\lambda_1)}{1-\lambda_1 b}. \tag{2.15}$$

Next we differentiate (2.12) with respect to z_p and z_j $(1 \le j \le p-1)$ and then set $z_k = 1$ for $k = 1, 2, \ldots, P$ to obtain

$$0 = -\left[\frac{dG(0,\ldots,0,1,1,\ldots,1,z,1,\ldots,1)}{dz}\right]_{z=1} + \left[\frac{dG(0,\ldots,0,0,1,\ldots,1,z,1,\ldots,1)}{dz}\right]_{z=1}$$
$$+\lambda_j b(N_p - \pi_p) + \lambda_p b(N_j - \pi_j) + \lambda_p \lambda_j b^{(2)} + \lambda_p \pi_0 \left[B^{*(1)}(\sum_{k=1}^{j-1}\lambda_k) - B^{*(1)}(\sum_{k=1}^{j}\lambda_k)\right], \tag{2.16}$$
$$j = 1, 2, \ldots, p-1$$

where z appears as the p th argument of $G(\bullet)$. Also, 1 first appears in the j th and $j+1$ st positions of the arguments of $G(\bullet)$ in the first and second terms, respectively. Adding (2.16) over $j = 1, 2, \ldots, p-1$, we get

$$0 = -N_p + \left[\frac{dG(0,\ldots,0,z,1,\ldots,1)}{dz}\right]_{z=1}$$
$$+(\sum_{j=1}^{p-1}\lambda_j)b(N_p - \pi_p) + \lambda_p b\sum_{j=1}^{p-1}(N_j - \pi_j) + \lambda_p(\sum_{j=1}^{p-1}\lambda_j)b^{(2)} + \lambda_p\pi_0\left[-b - B^{*(1)}(\sum_{k=1}^{p-1}\lambda_k)\right]. \tag{2.17}$$

From (2.14) and (2.17) we obtain a recurrence relation

$$N_p = \pi_p + \frac{\lambda_p b\sum_{j=1}^{p-1}(N_j - \pi_j + \lambda_p b^{(2)}(\frac{1}{2}\lambda_p + \sum_{j=1}^{p-1}\lambda_j) + \pi_0\left[-\lambda_p b + B^*(\sum_{j=1}^{p-1}\lambda_j) - B^*(\sum_{j=1}^{p}\lambda_j)\right]}{1-(\sum_{j=1}^{p}\lambda_j)b}$$
$$p = 2, \ldots, P. \tag{2.18}$$

Thus, from (2.15) and (2.18) we can calculate N_p for $p = 1, 2, \ldots, P$ recursively.

2.2.2. Mean Message Waiting Time and Discrimination

Lastly we relate N_p and W_p ($p = 1, 2, \ldots, P$). To this end, let us define $Q_p(z)$ as the z-transform for the number of class p messages in the waiting room at the beginning of the service for a class p message. Note that the fraction that a service is given to a class p message is λ_p / λ. Therefore, collecting appropriate cases in Section 2.2.1 and dividing by λ_p / λ, we have

$$
\begin{aligned}
Q_p(z) = \left(\frac{\lambda_p}{\lambda}\right)^{-1} \lim_{i \to \infty} \Big[& \pi_p E\left[z^{n_{ip}-1+a_p} \mid case\ p \right] \\
& + \pi_0 \left\{ \frac{\lambda_p}{\lambda} B^*(\lambda) + E\left[z^{a_p - 1} \mid subcase\ 0\text{-}p\right] \bullet \text{Prob}\left[subcase\ 0\text{-}p\right] \right\} \Big] \\
= \frac{\lambda}{\lambda_p} \Big[& \frac{1}{z} \left\{ G(0, \ldots, 0, z, 1, \ldots, 1) - G(0, \ldots, 0, 0, 1, \ldots, 1) \right\} B^*[\lambda_p(1 - z)] \\
& + \pi_0 \left\{ \frac{\lambda_p}{\lambda} B^*(\lambda) + \frac{B^*\left[\sum_{j=1}^{p-1} \lambda_j + \lambda_p(1 - z)\right] - B^*\left(\sum_{j=1}^{p} \lambda_j\right)}{z} \right\} \Big].
\end{aligned}
\tag{2.19}
$$

From (2.19) we get (by making use of (2.14))

$$
Q_p^{(1)}(1) = \lambda b N_p + \frac{1}{2} \lambda_p \lambda b^{(2)}. \tag{2.20}
$$

Since the number of class p messages in the waiting room at the beginning of the service for a class p message is equal to the number of arrivals of class p messages during the waiting time of the class p message for which the service is just started, it follows that

$$
Q_p(z) = W_p^*(\lambda_p - \lambda_p z) \tag{2.21}
$$

where $W_p^*(s)$ is the LST of the distribution function for the class p message waiting time. Thus, we get

$$
W_p = Q_p^{(1)}(1) \ / \ \lambda_p = \frac{1}{2} \lambda b^{(2)} + \frac{1}{\lambda_p} \lambda b N_p, \qquad p = 1, 2, \ldots, P \tag{2.22}
$$

which is our goal.

We want to compare the mean message waiting times of our system (called *reservation priority system*) to those of the classical head-of-line priority system. The latter is given by [Cox61], [Heym69], [Klei76]:

$$W_p = \frac{\lambda b^{(2)}}{2\left(1 - \sum_{j=1}^{p-1} \lambda_j b\right)\left(1 - \sum_{j=1}^{p} \lambda_j b\right)}, \qquad p = 1, 2, \ldots, P. \tag{2.23}$$

In Figure 2.1, we compare W_p for head-of-line and reservation priority systems in the case of $P = 5$ priority classes. We assume the constant message service time $b = 1$, and the balanced traffic (same λ_p for all classes). Values of W_p ($p = 1, 2, \ldots, P$) are plotted against the total load $\rho = \lambda b$. In this figure, we see that, given ρ, W_p's for the head-of-line priority system are more widely spread than those for our reservation priority system. In other words, the reservation scheme discriminate the mean message waiting times for different priority classes less than the head-of-line scheme. Due to the conservation law in (2.2) (which reduces to the unweighted mean in the case of balanced traffic), the arithmetic means of W_p over all classes in head-of-line and reservation systems are identical.

To quantify the degree of discrimination, let us define the discrimination measure (a similar form appears in [Wong82], for example.):

$$D \triangleq \frac{\sum_{p=1}^{P} \frac{\lambda_p}{\lambda} (W_p - \overline{W})^2}{\overline{W}^2} \tag{2.24}$$

where $\overline{W}$ is defined in (2.2). Note that $D = 0$ if there is no priority structure. We assume the Erlang-m distribution for the service time ($m = 1$; exponential, $m = \infty$; constant):

$$B^*(s) = \left(\frac{m}{m + sb}\right)^m; \qquad b^{(2)} = \left(1 + \frac{1}{m}\right) b^2. \tag{2.25}$$

If we denote by $W_p^{(m)}$ and $\overline{W}^{(m)}$ the W_p and $\overline{W}$, respectively, in the case of Erlang-m distributed service

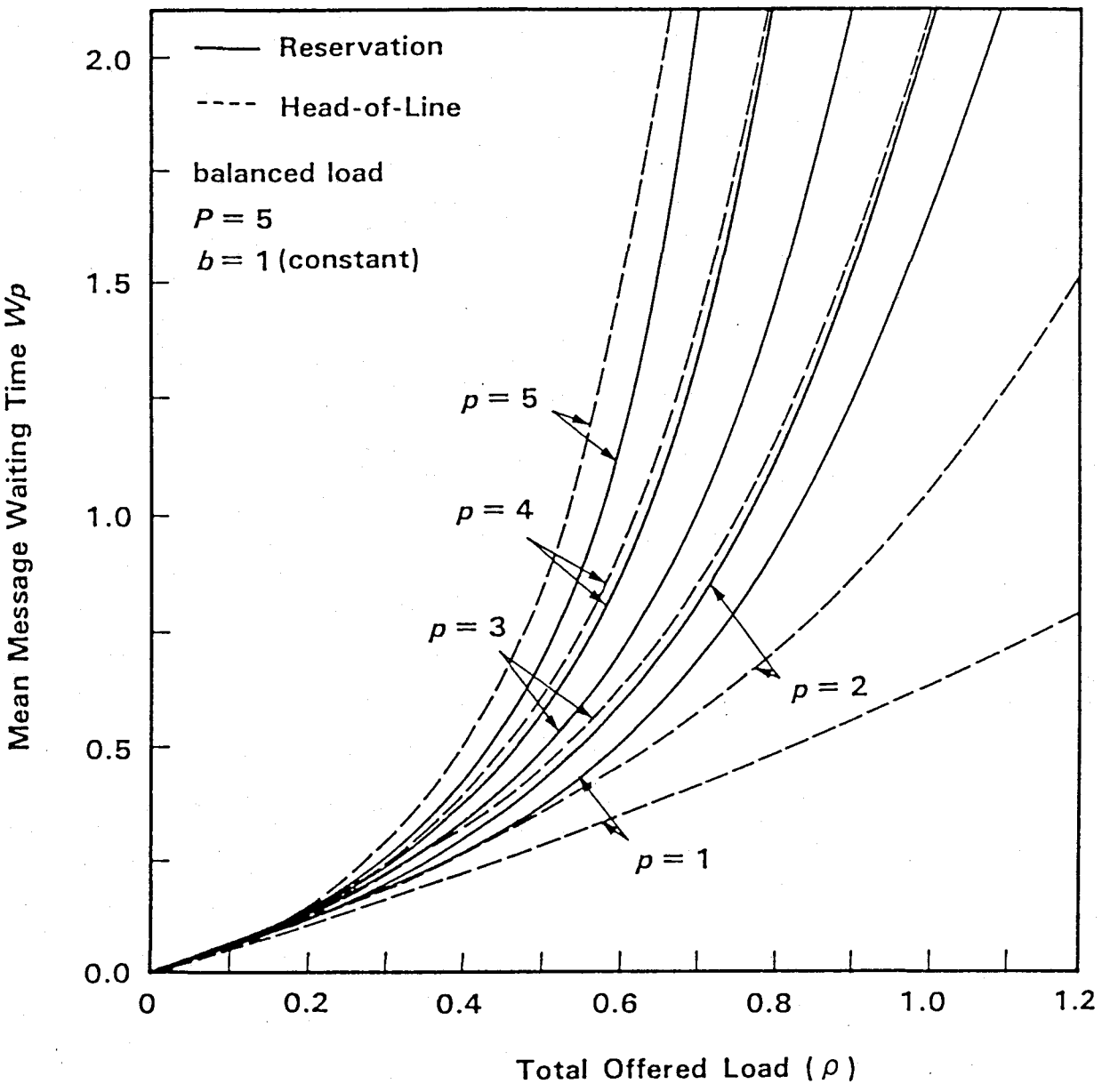


Figure 2.1. Mean Message Waiting Times for Each Priority Class in Model Without Service Switchover Times

times, we have

$$W_p^{(m)} = (1 + \frac{1}{m})\, W_p^{(\infty)}\ ; \qquad \overline{W}^{(m)} = (1 + \frac{1}{m})\, \overline{W}^{(\infty)} \tag{2.26}$$

for the head-of-line scheme (traffic can be unbalanced). Thus, D does not depend on m in head-of-line systems. In the case of reservation scheme, D does depend on m.

In Figure 2.2, we plot D against ρ for both schemes assuming again $P = 5$, $b = 1$ and balanced traffic. Here we see the difference in the degree of discrimination among priority classes.

2.3. Models with Service Switchover Times

We now proceed to study models with service switchover times. Let $R^*(s)$, r and $r^{(2)}$ be the LST of the distribution function, mean and the second moment, respectively, for the switchover time. The service disciplines are given as follows. Let us consider the successive i th and $i + 1$ st services administered to messages of all classes. The $i + 1$ st service is given to the highest priority class existing at the reservation point, namely, at the beginning of the i th service (*type I*), at the end of the i th service (*type II*), or at the end of the switchover time following the i th service (*type III*). See Figure 2.3 for reservation points in three types. If there are no messages waiting at a reservation point, no reservation is made for next service. In this case, a message from the highest priority class is chosen for the $i + 1$ st service among those existing at the moment when the server gets ready after the switchover time following the i th service. If there are no messages in the system at that time, the server experiences switchover times repeatedly until he finds at least one message in the waiting room. Then the $i + 1$ st service is given to the highest priority class of messages among them. Each service is given in a nonpreemptive fashion.

Note that, if we disregard the priority class of each message, the queueing process in all three types is equivalent to an $M/G/1$ system with the server going on vacation [Coop81] where the service time and vacation time distributions are given by $B^*(s)R^*(s)$ and $R^*(s)$, respectively. Therefore, for the three types,

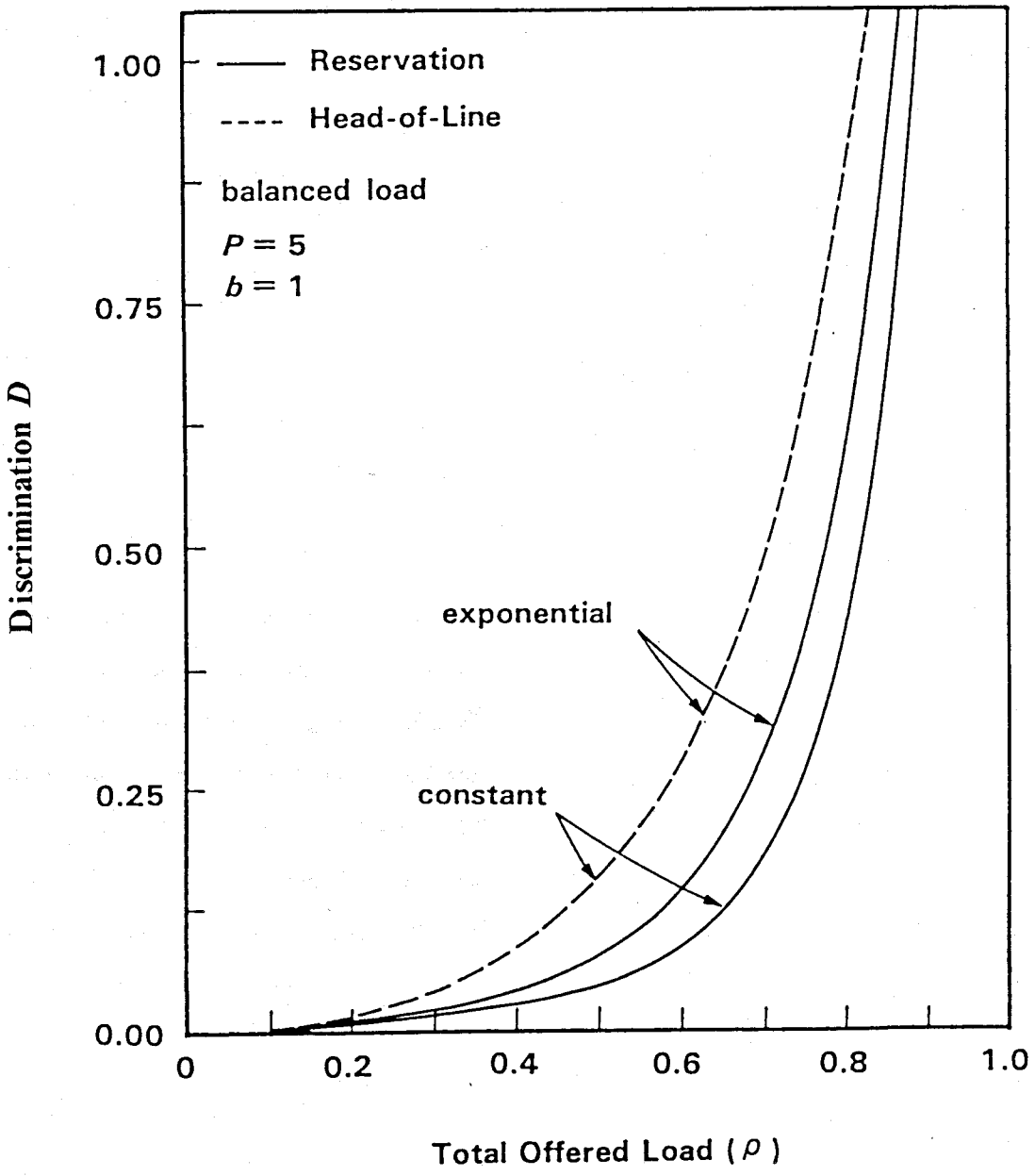


Figure 2.2. Comparison of Discrimination in Model Without Service Switchover Times

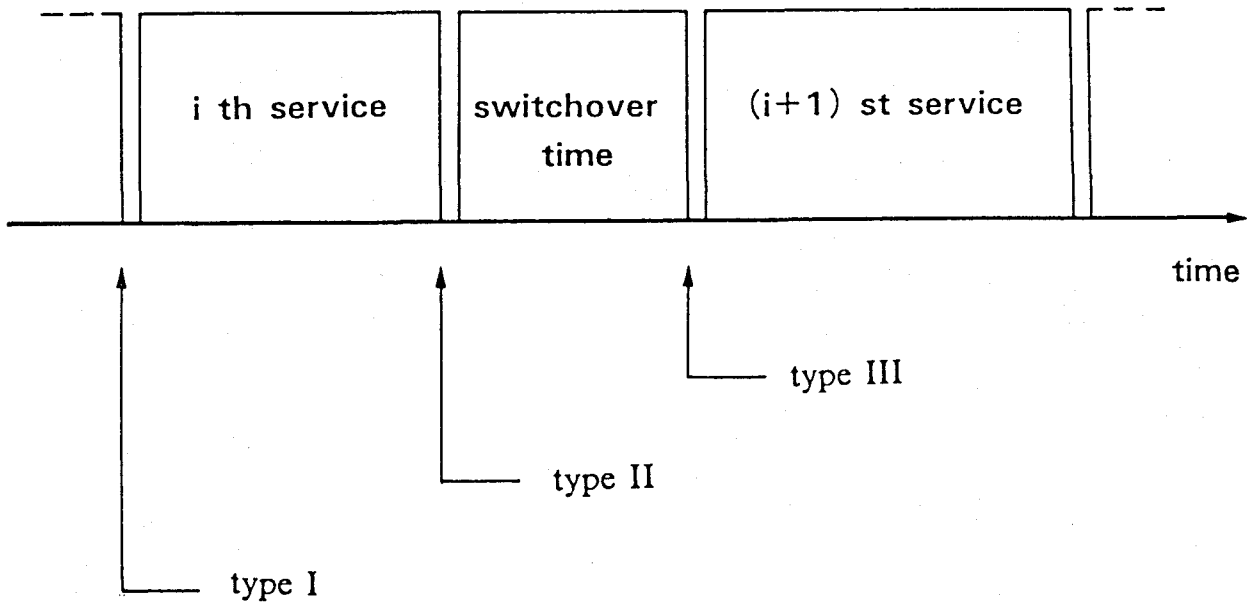


Figure 2.3. Reservation Points in Three Types With Service Switchover Times

we have the load-weighted mean of W_p's as

$$\overline{W} \triangleq \sum_{p=1}^{P} \frac{\lambda_p}{\lambda} W_p = \frac{\lambda [\, b^{(2)} + 2br + r^{(2)} \,]}{2[\, 1 - \lambda(b + r) \,]} + \frac{r^{(2)}}{2r} . \tag{2.27}$$

2.3.1. Reservation at the Beginning of Message Service

First, analysis is given to type I system.

2.3.1.1. Number of Messages at Service Start Instants

As before, let n_{ip} be the number of class p messages in the waiting room at the beginning of the i th service (excluding a message in service). Let a_p be the number of class p messages to arrive during a service time *and* the following switchover time. We now express $n_{i+1,p}$ in terms of n_{ip} and a_p in the following $P + 1$ mutually exclusive and exhaustive cases.

- *Case* p ($p = 1, 2, \ldots, P$): $n_{ij} = 0$ for $j = 1, 2, \ldots, p - 1$, and $n_{ip} > 0$.

  We have (2.3) with a different meaning of a_j.

- *Case* 0: $n_{ij} = 0$ for $j = 1, 2, \ldots, P$.

  - *Subcase* 0-p ($p = 1, 2, \ldots, P$): $a_j = 0$ for $j = 1, 2, \ldots, p - 1$, and $a_p > 0$.

    We have (2.4).

  - *Subcase* $0 - 0$: $a_j = 0$ for $j = 1, 2, \ldots, P$.

In this case, there are repeated switchover times between i th and $i+1$ st services. The $i+1$ st service is given to a message of the highest priority (say p) among messages arriving first during the successive switchover times.

Let a_j' be the number of class j messages that arrive during the last switchover time. Then we have

$$n_{i+1,j} = \begin{cases} 0 & j = 1, 2, \ldots, p-1 \\ a_j' - 1 & j = p \\ a_j' & j = p+1, p+2, \ldots, P \end{cases} \tag{2.28}$$

Now, we may equate the probability that a randomly chosen service is given to a class p message to λ_p / λ which is the long-time fraction of the number of class p messages to the number of all messages:

$$\pi_p + \pi_0 \left[\frac{R^*\left(\sum_{j=1}^{p-1} \lambda_j\right) - R^*\left(\sum_{j=1}^{p} \lambda_j\right)}{1 - R^*(\lambda)} B^*(\lambda) R^*(\lambda) \right.$$
$$\left. + B^*\left(\sum_{j=1}^{p-1} \lambda_j\right) R^*\left(\sum_{j=1}^{p-1} \lambda_j\right) - B^*\left(\sum_{j=1}^{p} \lambda_j\right) R^*\left(\sum_{j=1}^{p} \lambda_j\right) \right] = \frac{\lambda_p}{\lambda}, \qquad p = 1, 2, \ldots, P \tag{2.29}$$

where π_p $(p = 0, 1, \ldots, P)$ is the probability of *case p* as before. We need another relation among $\{\pi_p \; ; \; p = 0, 1, \ldots, P\}$ to determine them uniquely.

Using (2.11), we now express the relationship in the above cases as

$$
\begin{aligned}
G(z_1,z_2,\ldots,z_P) = {} & \left\{ \sum_{p=1}^{P} \frac{1}{z_p} \left[G(0,\ldots,0,z_p,z_{p+1},\ldots,z_P) - G(0,\ldots,0,0,z_{p+1},\ldots,z_P) \right] \right\} \\
& \bullet B^*\left[\sum_{p=1}^{P} \lambda_p (1-z_p) \right] R^*\left[\sum_{p=1}^{P} \lambda_p (1-z_p) \right] \\
& + G(0,\ldots,0) \left[\frac{B^*(\lambda) R^*(\lambda)}{1-R^*(\lambda)} \sum_{p=1}^{P} \frac{1}{z_p} \left\{ R^*\left[\sum_{j=1}^{p-1} \lambda_j + \sum_{j=p+1}^{P} \lambda_j (1-z_j) \right] \right. \right. \\
& \left. \left. - R^*\left[\sum_{j=1}^{P} \lambda_j + \sum_{j=p+1}^{P} \lambda_j (1-z_j) \right] \right\} \right. \\
& \left. + \sum_{p=1}^{P} \frac{1}{z_p} \left\{ B^*\left[\sum_{j=1}^{p-1} \lambda_j + \sum_{j=p}^{P} \lambda_j (1-z_j) \right] R^*\left[\sum_{j=1}^{p-1} \lambda_j + \sum_{j=p}^{P} \lambda_j (1-z_j) \right] \right. \right. \\
& \left. \left. - B^*\left[\sum_{j=1}^{P} \lambda_j + \sum_{j=p+1}^{P} \lambda_j (1-z_j) \right] R^*\left[\sum_{j=1}^{P} \lambda_j + \sum_{j=p+1}^{P} \lambda_j (1-z_j) \right] \right\} \right]
\end{aligned}
\tag{2.30}
$$

where $G(\bullet)$ is defined by (2.10). This is the governing equation of type I system.

By differentiating (2.30) with respect to z_p and then setting $z_j = 1$ for $j = 1, 2, \ldots, P$, we get

$$
\begin{aligned}
\pi_p = {} & \lambda_p (b+r) + \pi_0 \left[\frac{B^*(\lambda) R^*(\lambda)}{1-R^*(\lambda)} \left\{ - \left[R^*\left(\sum_{j=1}^{p-1} \lambda_j \right) - R^*\left(\sum_{j=1}^{p} \lambda_j \right) \right] + \lambda_p r \right\} \right. \\
& \left. - \left\{ B^*\left(\sum_{j=1}^{p-1} \lambda_j \right) R^*\left(\sum_{j=1}^{p-1} \lambda_j \right) - B^*\left(\sum_{j=1}^{p} \lambda_j \right) R^*\left(\sum_{j=1}^{p} \lambda_j \right) \right\} \right], \qquad p = 1,2,\ldots,P .
\end{aligned}
\tag{2.31}
$$

Solving (2.29) and (2.31), we get

$$
\pi_0 = [\, 1 - \lambda(b+r) \,] \frac{1-R^*(\lambda)}{\lambda r\, B^*(\lambda)\, R^*(\lambda)} . \tag{2.32}
$$

Thus, $\{ \pi_p \; ; \; p = 1, 2, \ldots P \}$ can be determined from (2.31).

Similar to Section 2.2, we may obtain the mean number N_P of class p messages in the waiting room at the beginning of service as

$$N_1 = \pi_1 + \frac{1}{1-\lambda_1(b+r)}\Big[\frac{1}{2}\lambda_1^2(b^{(2)}+2br+r^{(2)}) + \pi_0\Big\{\frac{B^*(\lambda)R^*(\lambda)}{1-R^*(\lambda)}\Big\{\frac{1}{2}\lambda^2 r^2 - \lambda_1 r + 1 - R^*(\lambda_1)\Big\} + 1 - B^*(\lambda_1)R^*(\lambda_1) - \lambda_1(b+r)\Big\}\Big] \tag{2.33}$$

and

$$N_p = \pi_p + \frac{1}{1-(\sum_{j=1}^{p}\lambda_j)(b+r)}\Bigg[\lambda_p(b+r)\sum_{j=1}^{p-1}(N_j-\pi_j) + (\frac{1}{2}\lambda_p^2 + \lambda_p\sum_{j=1}^{p-1}\lambda_j)(b^{(2)}+2br+r^{(2)}) + \pi_0\Bigg\{\frac{B^*(\lambda)R^*(\lambda)}{1-R^*(\lambda)}\Bigg\{\frac{1}{2}\lambda_p^2 r^2 + \lambda_p(\sum_{j=1}^{p-1}\lambda_j)r^{(2)} - r\lambda_p + R^*(\sum_{j=1}^{p-1}\lambda_j) - R^*(\sum_{j=1}^{p}\lambda_j)\Bigg\} + B^*(\sum_{j=1}^{p-1}\lambda_j)R^*(\sum_{j=1}^{p-1}\lambda_j) - B^*(\sum_{j=1}^{p}\lambda_j)R^*(\sum_{j=1}^{p}\lambda_j) - \lambda_p(b+r)\Bigg\}\Bigg]$$

$$p = 2, \ldots, P. \tag{2.34}$$

Thus, from (2.33) and (2.34) we can calculate N_p for $p = 1, 2, \ldots, P$ recursively.

2.3.1.2. Mean Message Waiting Time

We can derive $Q_p(z)$ by collecting appropriate cases in Section 2.3.1 and dividing by λ_p / λ

$$Q_p(z) = \left(\frac{\lambda_p}{\lambda}\right)^{-1} \Big[\frac{1}{z} \{ G(0, \dots, 0, z, 1, \dots, 1) - G(0, \dots, 0, 0, 1, \dots, 1) \}$$

$$\bullet B^*[\lambda_p (1 - z)]\ R^*[\lambda_p (1 - z)]$$

$$+ \pi_0 \left\{ \frac{B^*(\lambda)\ R^*(\lambda)}{1 - R^*(\lambda)} \bullet \frac{R^*[\sum_{j=1}^{p-1} \lambda_j + \lambda_p (1 - z)] - R^*(\sum_{j=1}^{p} \lambda_j)}{z} \right.$$

$$\left. + \frac{1}{z} \left(B^*[as \sum_{j=1}^{p-1} \lambda_j + \lambda_p (1 - z)]\ R^*[\sum_{j=1}^{p-1} \lambda_j + \lambda_p (1 - z)] \right.\right.$$

$$\left.\left. B^*(\sum_{j=1}^{p} \lambda_j)\ R^*(\sum_{j=1}^{p} \lambda_j) \right) \right\} \Big]$$

$$p = 1, 2, \dots, P . \qquad (2.35)$$

From (2.35) we get (by making use of (2.32)-(2.34))

$$Q_p^{(1)}(1) = \lambda(b + r)N_p + \frac{1}{2} \lambda\lambda_p (b^{(2)} + 2br + r^{(2)}) + \frac{\lambda_p r}{2}[1 - \lambda(b + r)]$$

$$p = 1, 2, \dots, P . \qquad (2.36)$$

Since the relationship (2.21) still holds for the model with service switchover times, we get the mean waiting times in type I system as

$$W_p(\text{type I}) = \frac{\lambda}{\lambda_p}(b + r)N_p + \frac{1}{2} \lambda(b^{(2)} + 2br + r^{(2)}) + \frac{r}{2}[1 - \lambda(b + r)]$$

$$p = 1, 2, \dots, P \qquad (2.37)$$

which is our goal. This result reduces to (2.22) in the limit of zero waiting time (let $r \to 0$ after setting $R^*(s) = e^{-sr}$).

2.3.2. Reservation at the End of Message Service

Next we derive the corresponding results for type II. Since the analysis is similar to Section 2.2, we only give the outline. In this case, the equation for $G(\bullet)$ defined by (2.10), where n_{ip} now denotes the number of class p messages in the waiting room at the *end* of the i th service, is given by

$$G(z_1,z_2,\dots,z_P) = \left\{ \sum_{p=1}^{P} \frac{1}{z_p}\left[G(0,\dots,0,z_p,z_{p+1},\dots,z_P) - G(0,\dots,0,0,z_{p+1},z_P)\right] \right\}$$

$$\bullet B^*\left[\sum_{p=1}^{P}\lambda_p(1-z_p)\right] R^*\left[\sum_{p=1}^{P}\lambda_p(1-z_p)\right]$$

$$+ G(0,\dots,0) \sum_{p=1}^{P} \left\{ \frac{1}{z_p} \bullet \frac{R^*\left[\sum_{j=1}^{p-1}\lambda_j + \sum_{j=p}^{P}\lambda_j(1-z_j)\right] - R^*\left[\sum_{j=1}^{p}\lambda_j + \sum_{j=p+1}^{P}\lambda_j(1-z_j)\right]}{1-R^*(\lambda)} \right. \tag{2.38}$$

$$\left. \bullet B^*\left[\sum_{j=1}^{P}\lambda_j(1-z_j)\right] \right\}.$$

From (2.38), $\{\pi_p\ ;\ p = 0, 1, \dots, P\}$ defined by (2.11) are given by

$$\pi_0 = \frac{\{1-\lambda(b+r)\}\{1-R^*(\lambda)\}}{\lambda r R^*(\lambda)} \tag{2.39}$$

and

$$\pi_p = \frac{\lambda_p}{\lambda} - \frac{R^*\left(\sum_{j=1}^{p-1}\lambda_j\right) - R^*\left(\sum_{j=1}^{p}\lambda_j\right)}{1-R^*(\lambda)}, \qquad p = 1,2,\dots,P. \tag{2.40}$$

Also, the mean number N_p of class p messages in the waiting room at the end of service can be recursively calculated by

$$N_1 = \pi_1 + \frac{1}{1-\lambda_1(b+r)}\Big[(1-\pi_0)\,\frac{1}{2}\,\lambda_1{}^2(\,b^{(2)} + 2br + r^{(2)}\,)$$

$$+\pi_0\,\frac{1}{1-R^*(\lambda)}\Big\{\frac{1}{2}\lambda_1{}^2[\,r^{(2)} + 2br + b^{(2)}(1-R^*(\lambda))\,] \qquad (2.41)$$

$$+(1-b\lambda_1(1-R^*(\lambda_1))) - \lambda_1 r\Big\}\Big]$$

and

$$N_p = \pi_p + \frac{1}{1-(\sum_{j=1}^{p}\lambda_p)(b+r)}\Big[\lambda_p\,(b+r)\sum_{j=1}^{p-1}(N_j - \pi_j)$$

$$+(1-\pi_0)\,\lambda_p\,(\sum_{j=1}^{p-1}\lambda_j + \frac{1}{2}\,\lambda_p)\,(\,b^{(2)} + 2br + r^{(2)}\,)$$

$$+\pi_0\,\frac{1}{1-R^*(\lambda)}\Big\{\lambda_p\,(\sum_{j=1}^{p-1}\lambda_j + \frac{1}{2}\,\lambda_p)\,[\,r^{(2)} + 2br + b^{(2)}(1-R^*(\lambda))\,]$$

$$+[\,1-b(\sum_{j=1}^{p}\lambda_j)]\,[R^*(\sum_{j=1}^{p-1}\lambda_j) - R^*(\sum_{j=1}^{p}\lambda_j)] - \lambda_p\,[\,r + (1-R^*(\sum_{j=1}^{p-1}\lambda_j))\,b\,]\Big\}\Big]$$

$$p = 2, \ldots, P\,. \qquad (2.42)$$

As before, it is possible to derive $Q_p(z)$, which is the z-transform for the number of class p messages in the waiting room at the beginning of the service for a class p message:

$$Q_p(z) = (\frac{\lambda_p}{\lambda})^{-1}\,[\,\frac{1}{z}\{G(0,\ldots,0,z,1,\ldots,1) - G(0,\ldots,0,0,1,\ldots,1)\}\bullet R^*[\,\lambda_p(1-z)\,]$$

$$+\pi_0\Big\{\frac{1}{1-R^*(\lambda)}\bullet\frac{R^*[\sum_{j=1}^{p-1}\lambda_j + \lambda_p(1-z)\,] - R^*(\sum_{j=1}^{p}\lambda_j)}{z}\Big\}\,] \qquad (2.43)$$

$$p = 1, 2, \ldots, P\,.$$

Then the mean message waiting times W_p(type II) can be found through (2.21) as

$$W_p(\text{ type II }) = \frac{1}{\lambda_p}\Bigg[\lambda(b+r)N_p + (1-\pi_0)\,\frac{1}{2}\,\lambda\lambda_p\,(\,b^{(2)} + 2br + r^{(2)}\,) - \pi_p\lambda b$$

$$+ \frac{\pi_0}{1-R^*(\lambda)}\Bigg\{\frac{1}{2}\,\lambda\lambda_p\Big[\,r^{(2)} + 2br + b^{(2)}(1-R^*(\lambda))\,\Big]$$

$$-\lambda b\Bigg[\,R^*\Big(\sum_{j=1}^{p-1}\lambda_j\Big) - R^*\Big(\sum_{j=1}^{p}\lambda_j\Big)\,\Bigg]\Bigg\}\Bigg] \qquad (2.44)$$

$$p = 1, 2, \ldots, P\,.$$

For type III system, the mean waiting times can be obtained by applying the analysis of the nonpreemptive priority $M/G/1$ queue with server's vacations by [Heym69]. Namely, we only replace his service time by our $B^*(\bullet)\,R^*(\bullet)$ and his vacation time by our $R^*(\bullet)$. We thus have

$$W_p(\text{ type III }) = \frac{\lambda(\,b^{(2)} + 2br\,) + (1-\lambda b)\,\dfrac{r^{(2)}}{r}}{2\Big\{\,1-\sum_{j=1}^{p}\lambda_j\,(r+b)\,\Big\}\Big\{\,1-\sum_{j=1}^{p-1}\lambda_j\,(r+b)\,\Big\}}, \quad p = 1, 2, \ldots, P\,. \qquad (2.45)$$

Both of $W_p(\text{ type II })$ and $W_p(\text{ type III })$ reduce to (2.23) in the limit of zero switchover times.

2.3.3. Numerical Comparisons

We want to compare the mean message waiting times for types I, II and III. In Figure 2.4, we compare W_p for the three types in the case of $P = 3$ priority classes. We assume the constant message service time $b = 1$, the constant switchover time $r = 0.1$, and balanced traffic. Values of W_p ($p = 1, 2, \ldots, P$) are plotted against the total load ρ. In this figure, we see that, given ρ, W_p's for type III are most widely spread among three types, and those for type I are least. In other words, the later we set the reservation point, the more are discriminated different priority classes. In Figure 2.5, we compare W_p for three types in the case of $P = 2$ priority classes with two values of constant switchover time r. We again assume

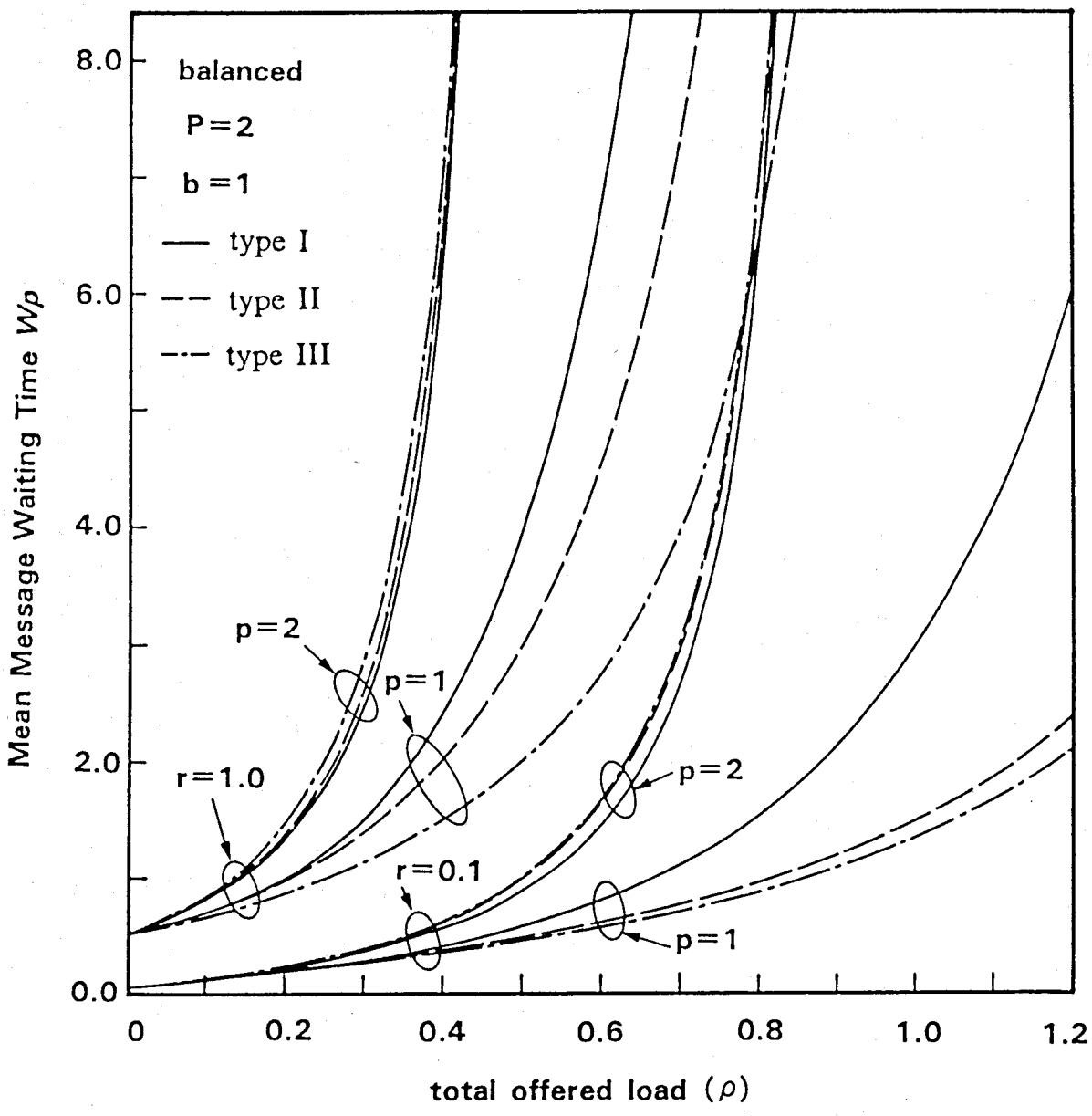


Figure 2.4. Mean Message Waiting Times for Each Priority Class in Models with Service Switchover Times

balanced traffic and the constant message service time $b = 1$. The mean waiting times for type III are always most widely spread among three types for every r.

To quantify the degree of discrimination, we again use (2.24) where $\overline{W}$ is now the load-weighted mean defined in (2.27). Note that $D = 0$ if there is no priority structure, and D for type III depends only on the means b and r (no dependence on the higher moments) even if traffic is unbalanced. In types I and II, D does depend on $b^{(2)}$ and $r^{(2)}$. In Figure 2.6, we plot D against ρ for the three types assuming again $P = 3$, the constant switchover time $r = 0.1$ and balanced traffic. We have considered the constant and exponentially distributed service times both with $b = 1$. Here we confirm the difference in the degree of discrimination among priority classes.

2.4. Conclusion

In this chapter, we have shown an exact analysis for the mean message waiting times in nonpreemptive reserved priority systems where next service class is determined by the highest priority existing at the beginning of the current service time for both models with/without service switchover times. We have compared the degree of discrimination among priority class of these systems to that of the nonpreemptive head-of-line priority systems with/without service switchover times. Assuming that message service times are identically distributed for all classes, we have shown that the mean message waiting times can be calculated recursively. When balanced traffic is assumed (the same Poisson arrival rate for all classes), we have numerically shown that, given traffic intensity, the mean waiting times are most widely spread for type III, and least for type I in the case of models with switchover times.

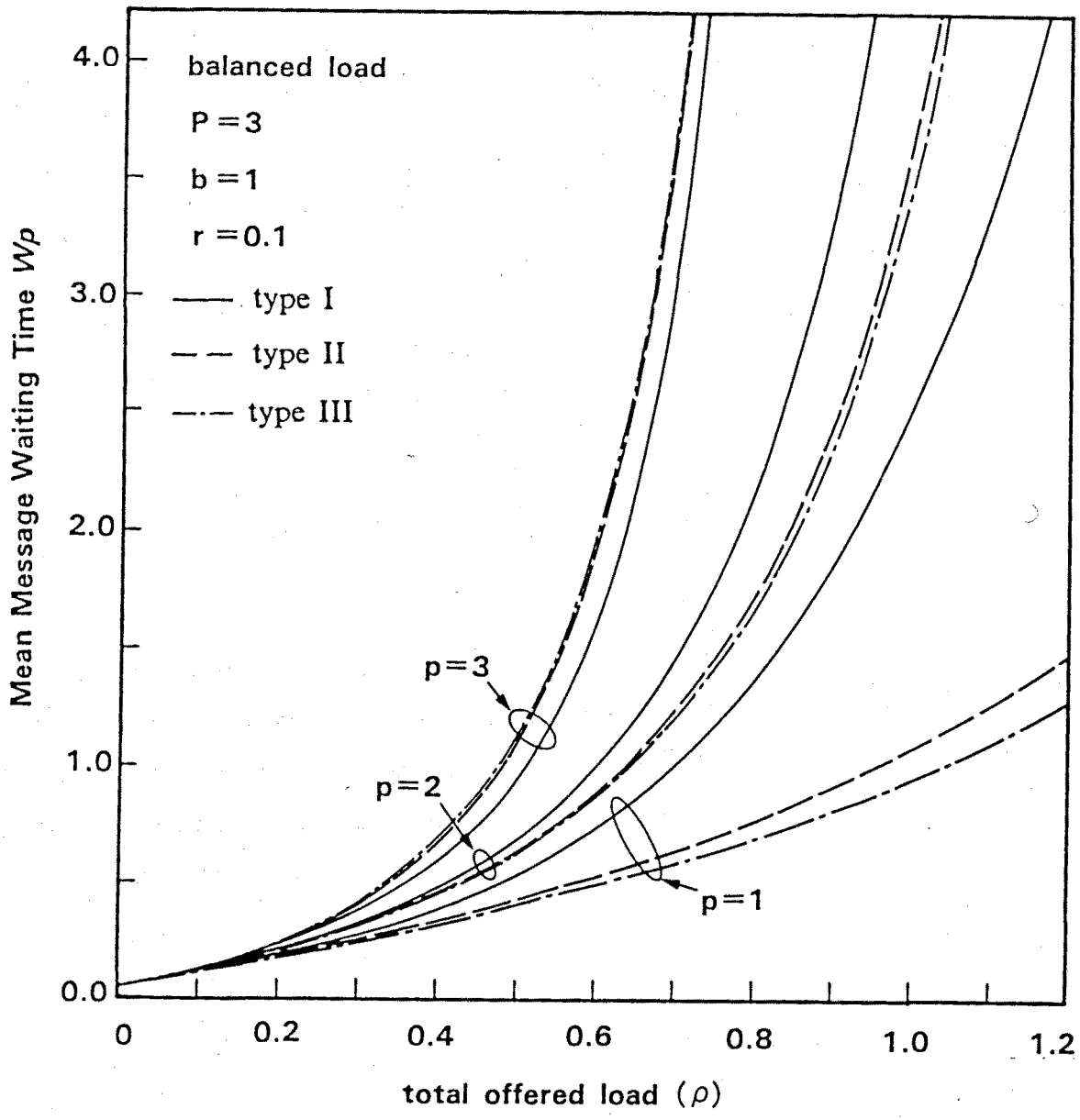


Figure 2.5. Mean Message Waiting Times for Each Priority Class with Various Switchover Times

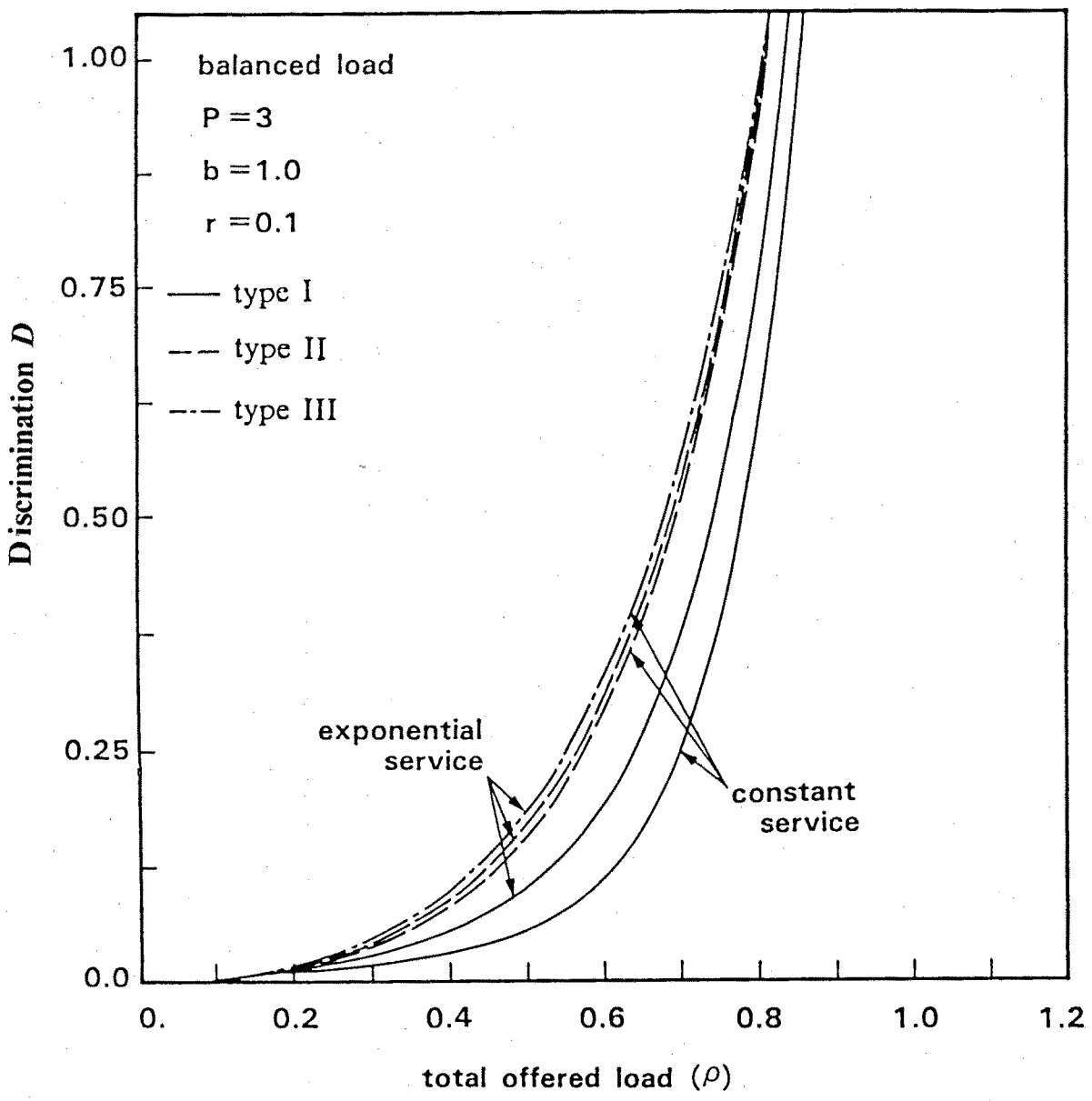


Figure 2.6. Comparison of Discrimination in Models with Service Switchover Times

Chapter 3. Two-Layer Modeling for Local Area Networks

3.1. Introduction

As described in Chapter 1, various performance studies have been made for the individual layers in the layered model of the OSI, whereby the layers above and below the one of concern have been represented simply by workload source and transmission models for analytical tractability [Reis86]. However, only few works have been published for performance study where two or more layers are combined together. Such a performance model is significant for network users to predict performance of network functions imbedded in their application programs in layer 7 in the OSI model.

In this chapter, we apply the OSI layers model to LAN. A certain aspect of this topic has been described by Gihr and Kuehn [Gihr86]. They combine the MAC (Media Access Control) layer and the LLC layer of LAN, and give a modeling concept of connection-oriented services. In contrast, we concentrate on the modeling of the MAC layer and the Transport layer in a layered protocol architecture. For the LLC layer, a connectionless service is considered in our model. In the connectionless data transmission, peer LLC entities exchange fully addressed data units (i.e., datagrams) with each other, and responsibilities of sequencing or flow control are left to the upper layers. (In the connection-oriented service, functions, such as connection set-up/termination, sequencing, flow control and error recovery, are provided by the peer LLC entities.) Connectionless services may be useful in the LAN due to its high channel reliability

and broadcast capability. See, for example, [Meis85] for comparisons between connection-oriented and connectionless services in the LAN where Meister *et al.* deal with file transfers. Recently, Mitchell and Lide [Mitc86] has offered a generic modeling framework for hierarchical structured OSI layers on LAN. They also provide a case study where a CSMA/CD protocol is used for the MAC layer. However, their analytical results are very limited and validations of their approximate method are not found. In this chapter, we adopt a token passing scheme for the MAC layer, and offer a comparative study among existing protocols to connect sessions for the Transport layer.

3.2. Modeling of Two-Layer Service in Local Area Networks

Figure 3.1 illustrates our model of the LAN in which several workstations are interconnected by a token ring. Application programs on workstations exchange data through three layers, i.e., the Transport layer, the LLC layer and the MAC layer. (In the model, the Physical layer is imbedded in the MAC layer from the above-mentioned layers' point of view.)

3.2.1. Modeling of a MAC Layer (Polling Systems)

For the MAC layer, we consider the limited (to 1) service which is the standard protocol adopted by IEEE 802.5. We utilize the analytical results for nonexhaustive service with symmetric load in [Taka85] and with asymmetric load in [Boxm86]. Both models include the following features:

1. The number of stations in the LAN is N.
2. Messages arrive at the MAC-queue of station i according to the Poisson process at rate λ_i ($i = 1, 2, \ldots, N$).

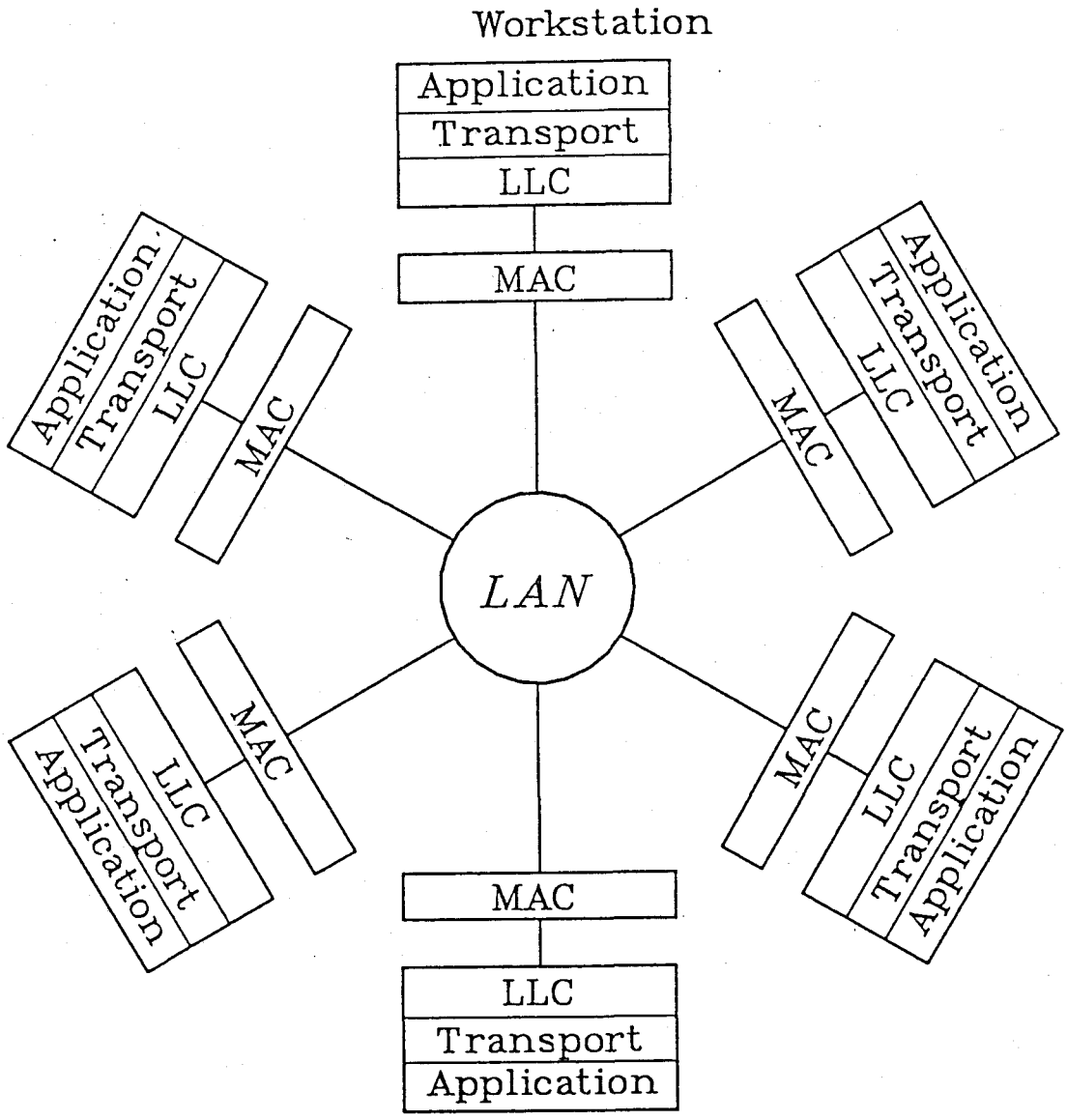


Figure 3.1. A Layered Modeling of LAN Performance

3. The service time of messages at station i is independent and identically distributed with first and second moments b_i and $b_i^{(2)}$, respectively. In the LAN, the service time of messages corresponds to the message transmission time which is given by the ratio of the message length to a channel speed.
4. The mean and variance of the switch-over time are r and δ^2, respectively. The switch-over time represents the propagation delay between two adjacent stations plus the bit latency at each station (these are assumed to be symmetric throughout the paper).
5. The server utilization due to station i is given by

$$\rho_i = \lambda_i b_i, \qquad i = 1, \ldots, N. \tag{3.1}$$

The total utilization of the server, ρ, is defined as

$$\rho = \sum_{i=1}^{N} \rho_i. \tag{3.2}$$

Nonexhaustive cyclic polling systems with symmetric load:

An exact analysis has been obtained for the symmetric case in which arrival rates and service time distribution functions are identical for all stations. The average message waiting time, w_i, for station i is found in [Taka85]:

$$w_i = \frac{\delta^2}{2r} + \frac{N[\lambda b^{(2)} + r(1 + \lambda b) + \lambda\delta^2]}{2[1 - N\lambda(r + b)]}, \qquad i = 1, \ldots, N \tag{3.3}$$

where $\lambda = \lambda_1 = \lambda_2 = \ldots = \lambda_N$, $b = b_1 = b_2 = \ldots = b_N$ and $b^{(2)} = b_1^{(2)} = b_2^{(2)} = \ldots = b_N^{(2)}$.

Nonexhaustive cyclic polling system with asymmetric load:

Approaches for asymmetric, nonexhaustive polling systems can be found in [Boxm86] and [Kueh79], for example, where approximation methods are used for deriving the mean message waiting times. We employ

an approximation in [Boxm86] where the mean message waiting times are given by

$$w_i = \frac{1-\rho-\rho_i}{1-\rho-\lambda_i r}\,\frac{1-\rho}{(1-\rho)\rho+\sum_{j=1}^{N}\rho_j^2}\left[\frac{\rho}{2(1-\rho)}\sum_{j=1}^{N}\lambda_j b_j^{(2)}+\frac{N\rho\delta^2}{2r}\right.$$

$$\left.+\frac{r}{2(1-\rho)}\sum_{j=1}^{N}\rho_j(1+\rho_j)\right], \qquad i=1,\ldots,N \tag{3.4}$$

which are shown to be in good agreement with simulation results.

The corresponding MAC-transit delays (elapsed time for messages in the MAC layer submodel) are then obtained by

$$f_i = w_i + b_i + \frac{Nr}{2}, \qquad i=1,\ldots,N\,. \tag{3.5}$$

The last term on the right-hand side of the above equation represents the mean propagation delay between sender i and a receiver station which is assumed to be uniformly distributed over the ring.

3.2.2. Modeling of a Transport Layer (Single-Chain, Closed-Queueing Network)

The Transport layer controls the messages flow between the source/sink pairs of stations, such as sequencing, error recovery and flow control. Let us follow Reiser [Reis79] to obtain a queueing model for the Transport layer. Namely, there are uni-directional virtual channels (called *chain* below) between designated pairs of stations. Each chain has a source and a sink, and messages on the chain are individually acknowledged. These assumptions lead to a model of closed queueing networks each of which connects two stations. In the model of this section, each station in the LAN is assumed to establish only a single

chain with another station. (In Section 3.3, we extend this model to the cases where each station may set up multiple chains with other stations.)

Chains interact with each other in the MAC layer submodel (i.e., contention in the channel access) as we model it with a multiple-queue single-server queueing system. To incorporate the effects of MAC layer into the queueing chain, we equate the service time of MAC-queue to f_i for station i, given by (3.5), and assume that the service discipline is IS (Infinite Servers). We also assume that the LLC layer is modelled by the simple FCFS queue because we consider connectionless-type service for the LLC layer here. Service times of LLC-queues at both sending and receiving sides correspond to the datagram processing times and are given as numerical parameters in the examples below.

Service times of a source queue correspond to the interarrival times of messages which an application program at the source station generates. On the other hand, a sink queue is considered in the following way:

- When data messages piggyback acknowledgements in the closed chain, the service times at the sink queue correspond to the interarrival times of the messages generated by the application program (abbreviated as AP in Figure 3.2) at the destination station (Figure 3.2-a).

- When acknowledgements are returned by themselves, service times at the sink queue correspond to the time to generate the acknowledgement at the Transport-queue in the destination (Figure 3.2-b). Messages will be passed to the application program separately with acknowledgements.

Our solution algorithm for the Transport layer submodel follows the MVA (Mean Value Analysis) method in [Reis79] and [Reis80]. Let us define the equilibrium quantities for queue j in a single-chain, closed network c:

$W^{(c)}$: Window size of closed network c

$\tau_j^{(c)}$: Mean service time of queue j in closed network c

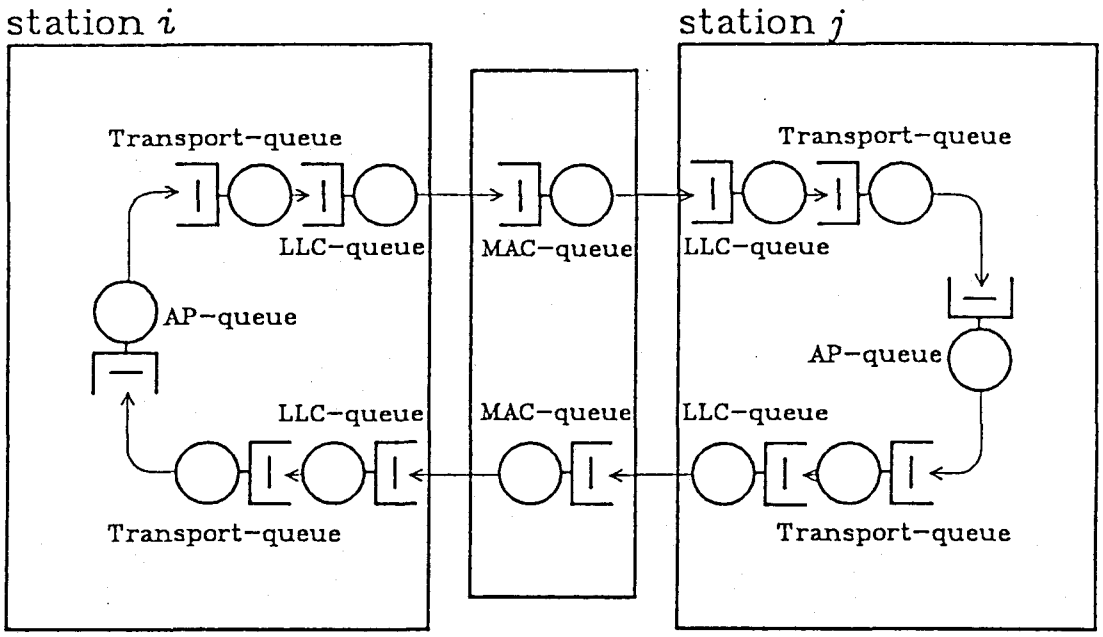


Figure 3.2-a. Transport Layer Submodel with Piggybacked Acknowledgement

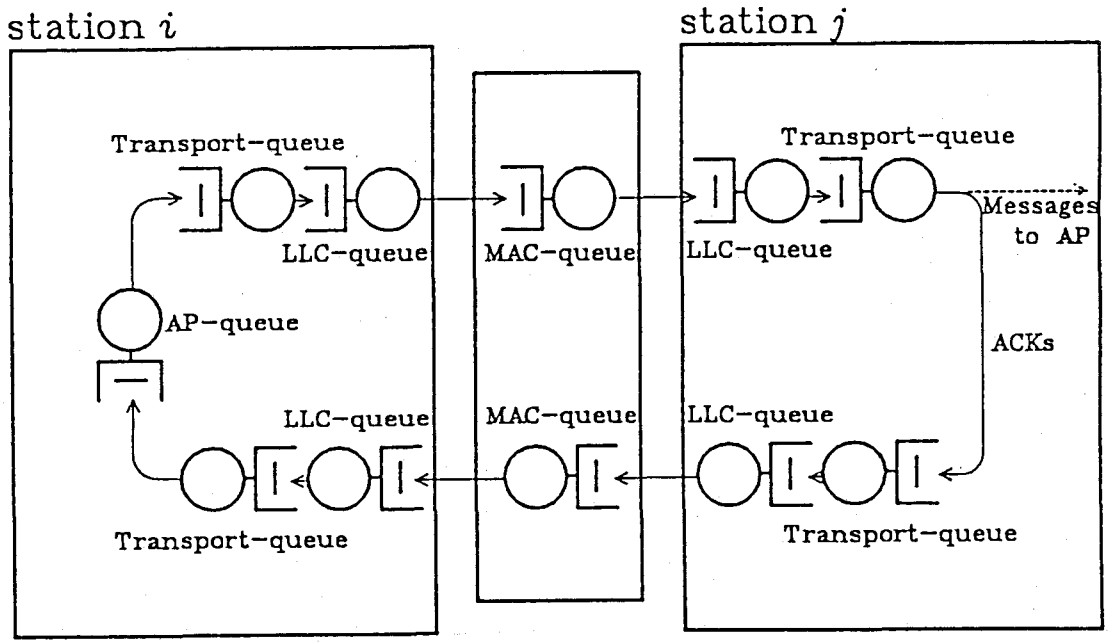


Figure 3.2-b. Transport Layer Submodel with Explicit Acknowledgement

$n_j^{(c)}(W^{(c)})$: Mean queue length of queue j in closed network c

$t_j^{(c)}(W^{(c)})$: Mean queueing time at queue j in closed network c

$\lambda^{(c)}(W^{(c)})$: Throughput of closed network c

Now we can compute the throughput of the closed network c recursively, starting with $n_j^{(c)}(W^{(c)}) = 0$, using the following relationships [Reis79]:

$$t_j^{(c)}(W^{(c)}) = \begin{cases} \tau_j^{(c)} = f_i & \text{if queue } j = \text{ MAC-queue at station } i \\ \tau_j^{(c)} & \text{if queue } j = \text{ AP-queue} \\ \tau_j^{(c)} [1 + n_j^{(c)}(W^{(c)} - 1)] & \text{if queue } j = \text{ Transport-queue or LLC-queue} \end{cases} \tag{3.6}$$

$$\lambda^{(c)}(W^{(c)}) = W^{(c)} / \sum_{k \in Q^{(c)}} t_k^{(c)}(W^{(c)}) \tag{3.7}$$

and

$$n_j(W^{(c)}) = \lambda^{(c)}(W^{(c)})\ t_j^{(c)}(W^{(c)}) \tag{3.8}$$

where $Q^{(c)}$ is a set of queues in closed network c. In (3.6), f_i is obtained from (3.5) for the MAC layer submodel. In turn we can use $\lambda^{(c)}(W^{(c)})$ as input values for the MAC layer submodel. This observation leads to an iterative solution algorithm for the combined two-layer modeling as described in the following section.

3.2.3. Solution Algorithm for Two-layer Modeling

Using the expressions derived in Sections 3.2.1 and 3.2.2 for the individual layer submodels, a numerical iterative algorithm has been developed. The outline of our algorithm is:

1. Set arrival rates to the MAC layer submodel to some initial values (e.g., set $\lambda_i = 0$ for all i).

2. Calculate mean message waiting times using (3.3) or (3.4). If the solution is infeasible, then modify arrival rates small enough to satisfy the stability condition for the polling systems [Boxm86]. (This modification is necessary because arrival rates derived from the Transport layer submodel could become too large to yield finite mean message waiting times for the MAC layer submodel.)

3. Calculate the throughput for each chain in the Transport layer submodel using the MVA solution algorithm (3.6)-(3.8); these values will be used in the next iteration cycle as arrival rates to the MAC layer submodel.

The convergence criterion for the iteration is defined by

$$\Delta_n = \sum_{i=1}^{N} \left[\lambda_i^{(n)} - \lambda_i^{(n-1)} \right] < \varepsilon \quad (\text{e.g.}, \varepsilon = 10^{-6}) \tag{3.9}$$

for the n th iteration. The detailed algorithm is given in Appendix A.

3.2.4. Numerical Results

In this section, numerical results are presented and compared to the simulation results in order to assess the accuracy of our algorithm. Computer simulation has been performed by IBM RESQ2 package [Saue84]. Throughout this chapter, the simulation results are depicted with 90 percent confidence intervals with sample mean denoted by $\bar{o}$ in figures. (For throughputs, only sample means are plotted.) Our approximation results are depicted by curves. We use the following parameters for numerical examples.

- Data-message transmission time: exponentially distributed with the mean of 1 *msec.*
- Acknowledgement transmission time: exponentially distributed with the mean of 0.1 *msec.*
- Ring latency between adjacent stations (including bit-latency): exponentially distributed with mean $r = 0.005$ *msec.* (and variance $\delta^2 = r^2$)
- Processing times for the LLC-queues at both transmitting and receiving sides: 1 *msec.*

3.2.4.1. Model 1. A Symmetric-load, Piggybacked Acknowledgement Model

The first model assumes a symmetric case with acknowledgement piggybacking for closed chains. The Transport layer submodel of each chain is shown in Figure 3.2-a. We use (3.3) for the analysis of the MAC layer submodel by the assumption that all chains are identical. Figure 3.3-a presents delays (including message service times) at MAC-queues and Transport-queues dependent on the window size of chains for three values of the processing times at the AP-queues: 25, 50 and 100 *msec.* We assume $N = 6$ terminals (three chains in the network) and 6 *msec.* of processing time for Transport-queues at both transmitting and receiving sides. In our numerical examples, we use same values of the processing times at four Transport-queues in each chain. Averages of queueing delays at four Transport-queues are depicted in figures. (More closely, queueing delays at four queues are different because arrival patterns are dominated by the output processes of previous queues. However, we could not observe significant differences among delays at four queues. Note that, in the analysis, such different queueing delays are never obtained by our approach.) The corresponding throughput in each chain is shown in Figure 3.3-b. These figures show that numerical results are in good agreement with the simulation results for Transport delays and throughput (relative errors are always below + 1%). However, MAC delays are slightly overestimated (about 5-10%) with using the exact analysis for the MAC layer. This is because we use the Poisson arrival assumption at the MAC layer and we model the MAC-queues as IS queues in the Transport Layer submodel.

Next, we illustrate delays and throughputs against the number of stations in Figure 3.4. The window size in each chain is fixed at 8. In this example, the bottleneck resides in the MAC layer in the Transport layer submodel when the number of stations is large. Our iterative algorithm requires more iteration cycles as the MAC layer becomes bottleneck. For example, we need 15 iterations to converge in the case of $N = 6$ stations and 50 *msec.* processing times at AP-queues while 1568 iterations are required in the case of $N = 40$ stations. (For the convergence criterion, $\varepsilon = 10^{-6}$ is used.) However, agreement between analytical and simulation results is good even for MAC delays.

The last example for the model shows delays and throughputs with the variable processing times at Transport-queues (2, 6 and 10 *msec.* processing times are used.) in Figure 3.5. Here we assume $N = 6$

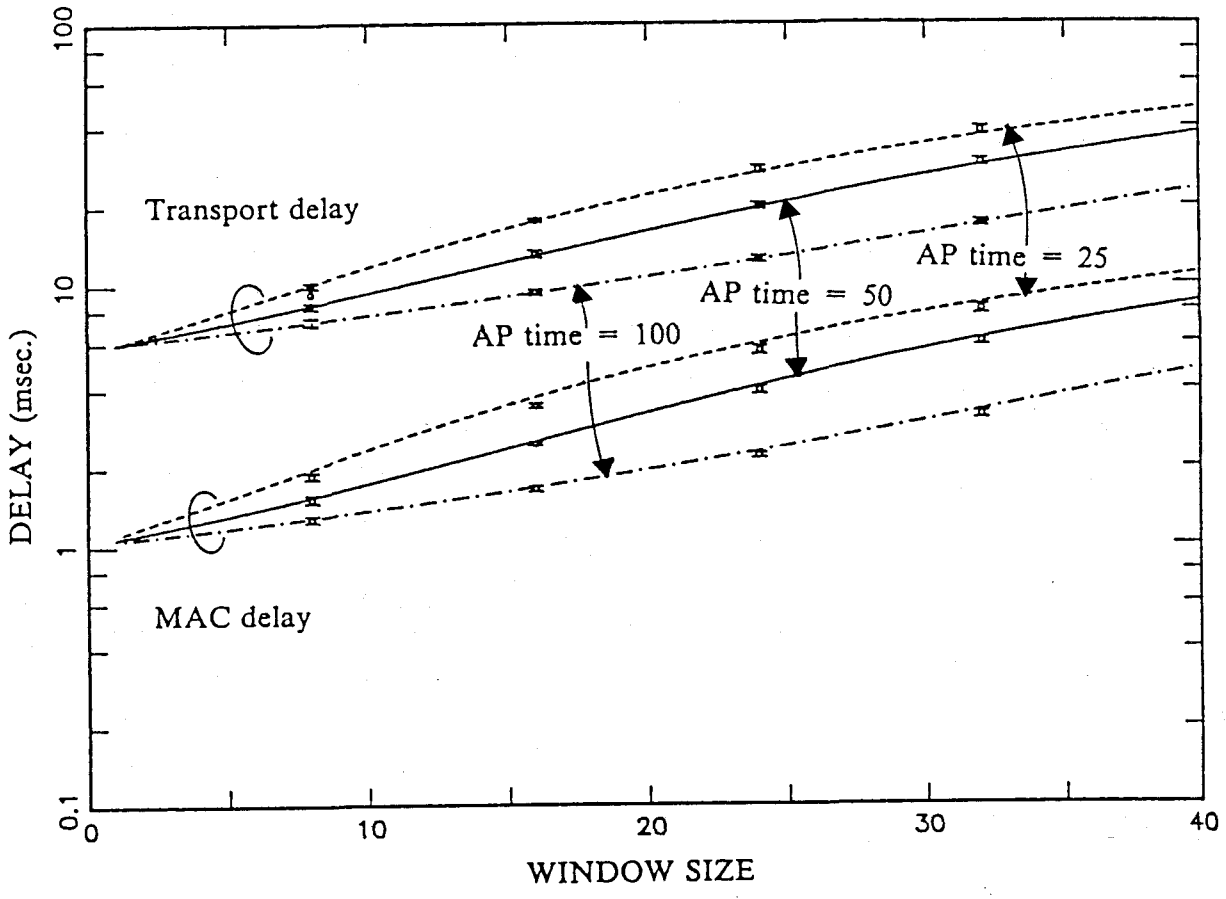


Figure 3.3-a. Delays in the Case of a Symmetric-Load and Piggyback Acknowledgement Model

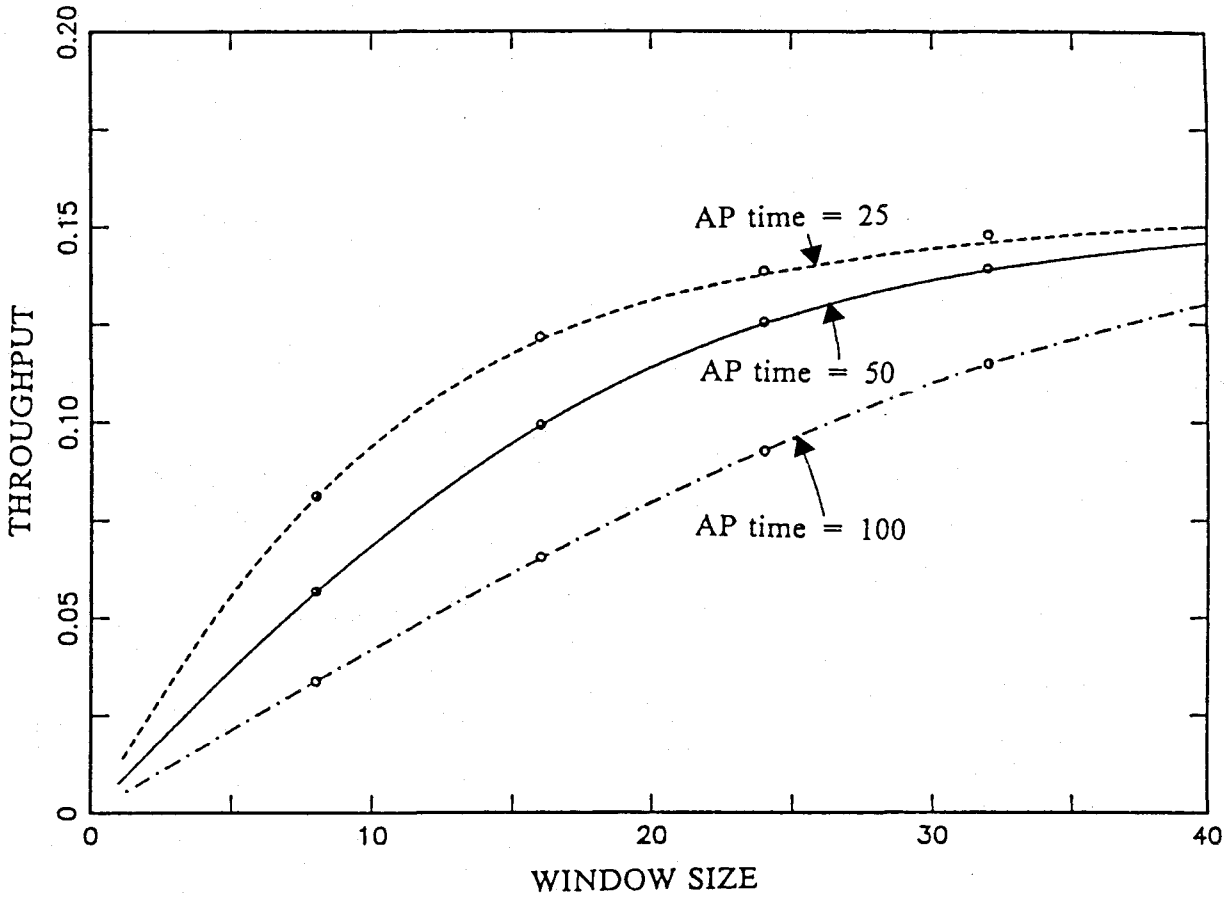


Figure 3.3-b. Throughput in the Case of a Symmetric-Load and Piggybacked Acknowledgement Model

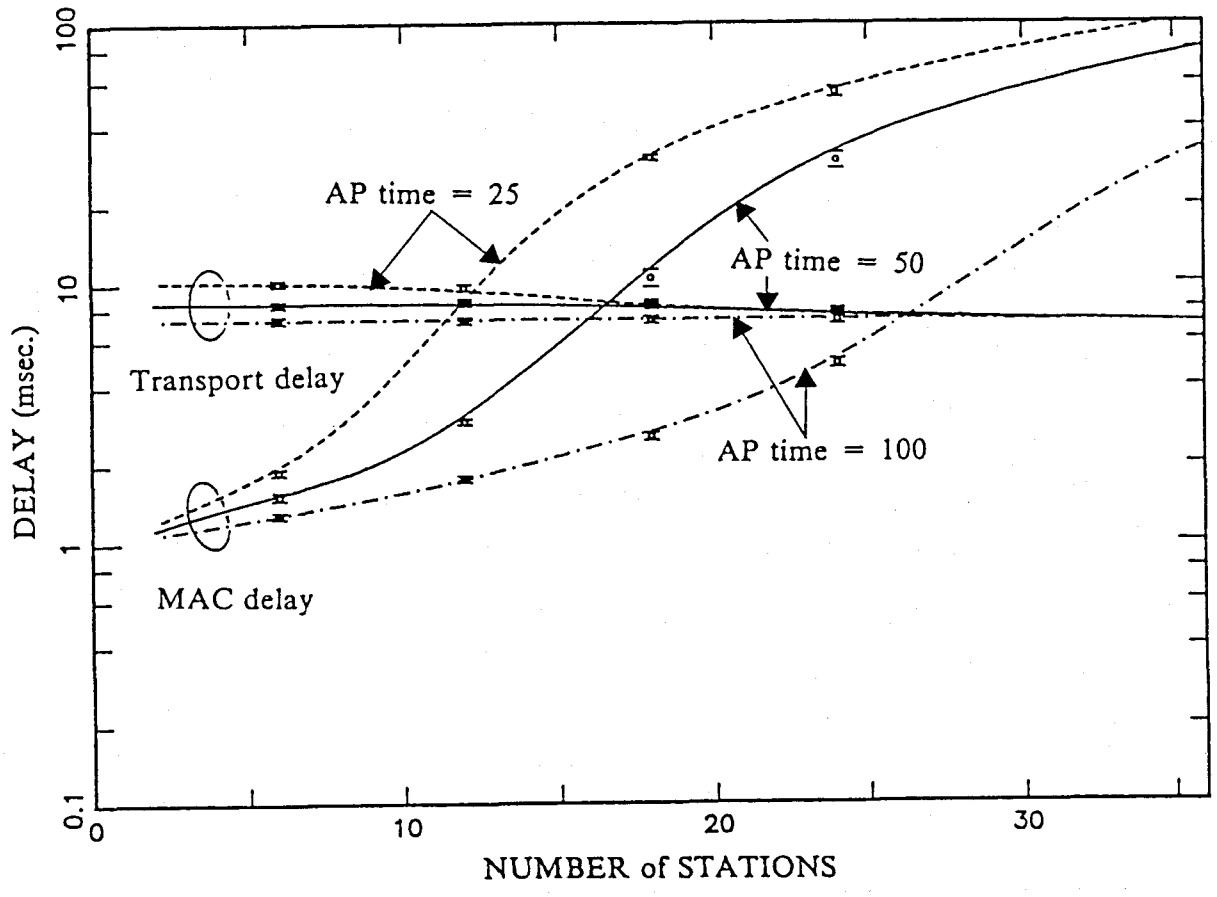


Figure 3.4-a. Delays in the Case of a Symmetric-Load and Piggybacked Acknowledgement Model (*N* is variable)

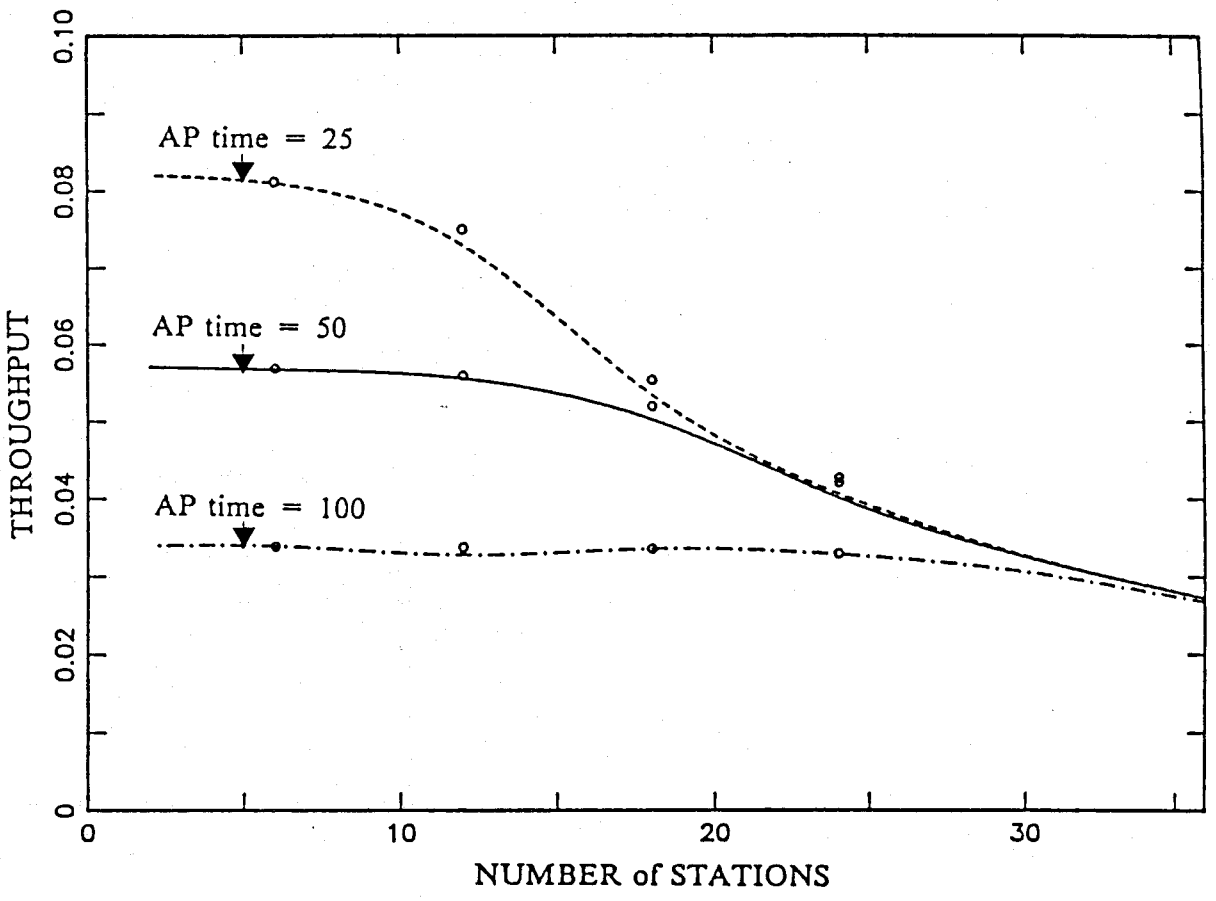


Figure 3.4-b. Throughput in the Case of a Symmetric-Load and Piggybacked Acknowledgement Model (*N* is variable)

stations and 50 *msec.* of processing time at the AP-queues. When the processing time at the Transport-queue is set to 2 *msec.*, the bottleneck is in the Transport-queues while it resides in the MAC queues in the case of 10 *msec.* processing times at Transport-queues. Here again we have excellent agreement between our computation and simulation in both cases.

3.2.4.2. Model 2. An Asymmetric-load, Explicit Acknowledgement Model

Next, we consider an asymmetric case where short-length acknowledgements are returned to the source station from the destination explicitly. See Figure 3.2-b for the Transport layer submodel of this case. We use the approximation method (3.4) because two different types of messages coexist in the MAC layer. At the MAC-queue in the source station, the data messages with 1 *msec.* message transmission times arrive while the acknowledgements with 0.1 *msec.* transmission times arrive at the MAC-queue in the destination station. Delays and throughputs are depicted against the window size and the number of stations in Figures 3.6 and 3.7, respectively. In figures, same simulation parameters are used as the previous model except that (i) the returned messages are short-length acknowledgements and (ii) the acknowledgements are not handled at the AP-queue in the destination station.

In Figure 3.6-a, MAC delays are smaller than those in Figure 3.4-a. It is because, in this model, returned messages are short-length acknowledgements, and their service times at MAC-queues are smaller than those for data-messages. So, the MAC delays tend to be small. In the current model, the acknowledgements are handled at the destination Transport-queues, not at the destination AP-queues as in the case of Model 1. This acknowledgement scheme and the smaller MAC delays lead to more throughput in the chain (Figure 3.6-b). This results in that the Transport delays are larger than those in Figure 3.4-a, and Transport-queues in the closed-chain are bottleneck in the current model. Similarly, more throughput is obtained in Figure 3.7 compared with those in Figure 3.5. Agreement between analytical and simulation results is fairly good even in the asymmetrical and heavy traffic conditions. In fact, offered traffic to the network is beyond 0.95 when $N = 24$ in Figure 3.7.

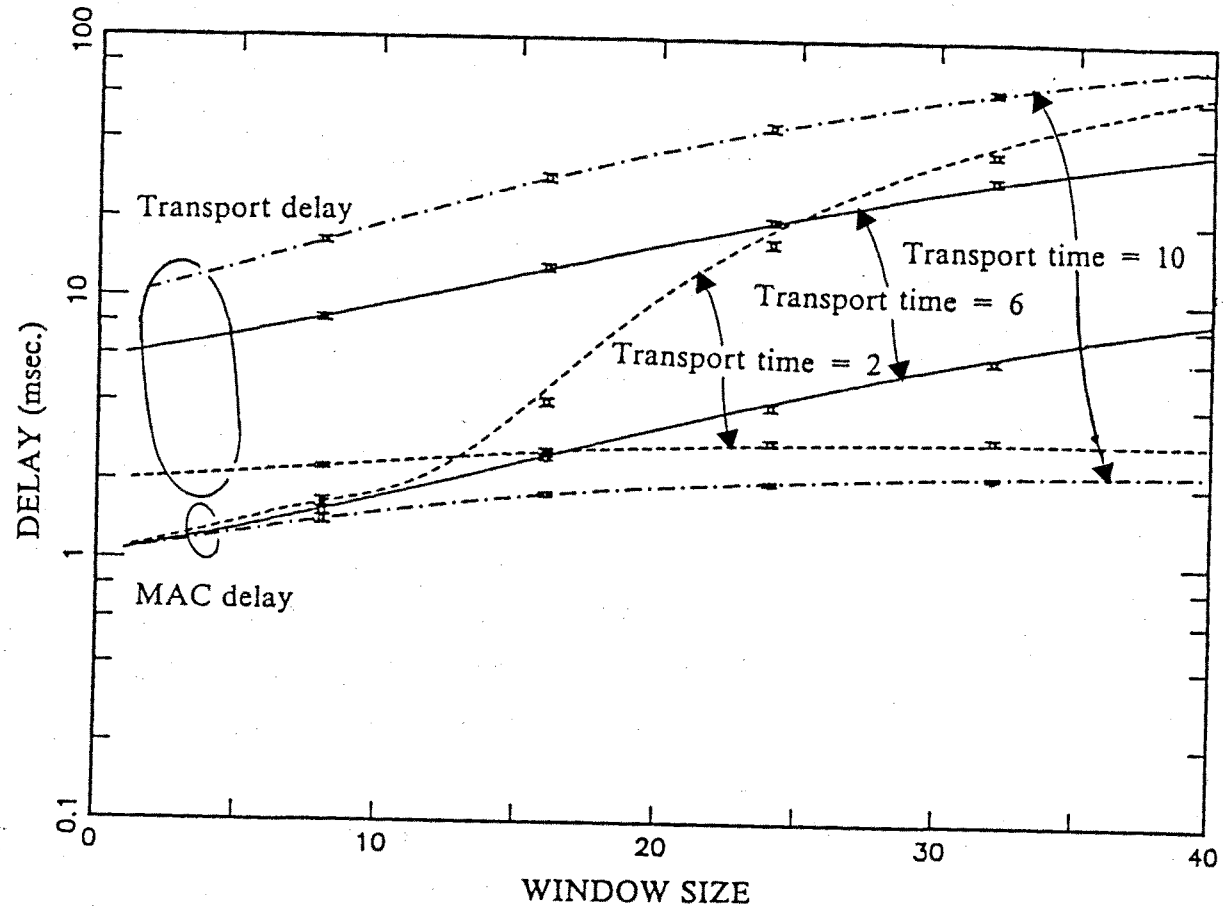


Figure 3.5-a. Delays in the Case of a Symmetric-Load and Piggybacked Acknowledgement Model (Window Sizes and Processing Times at Transport Queues are Variable.)

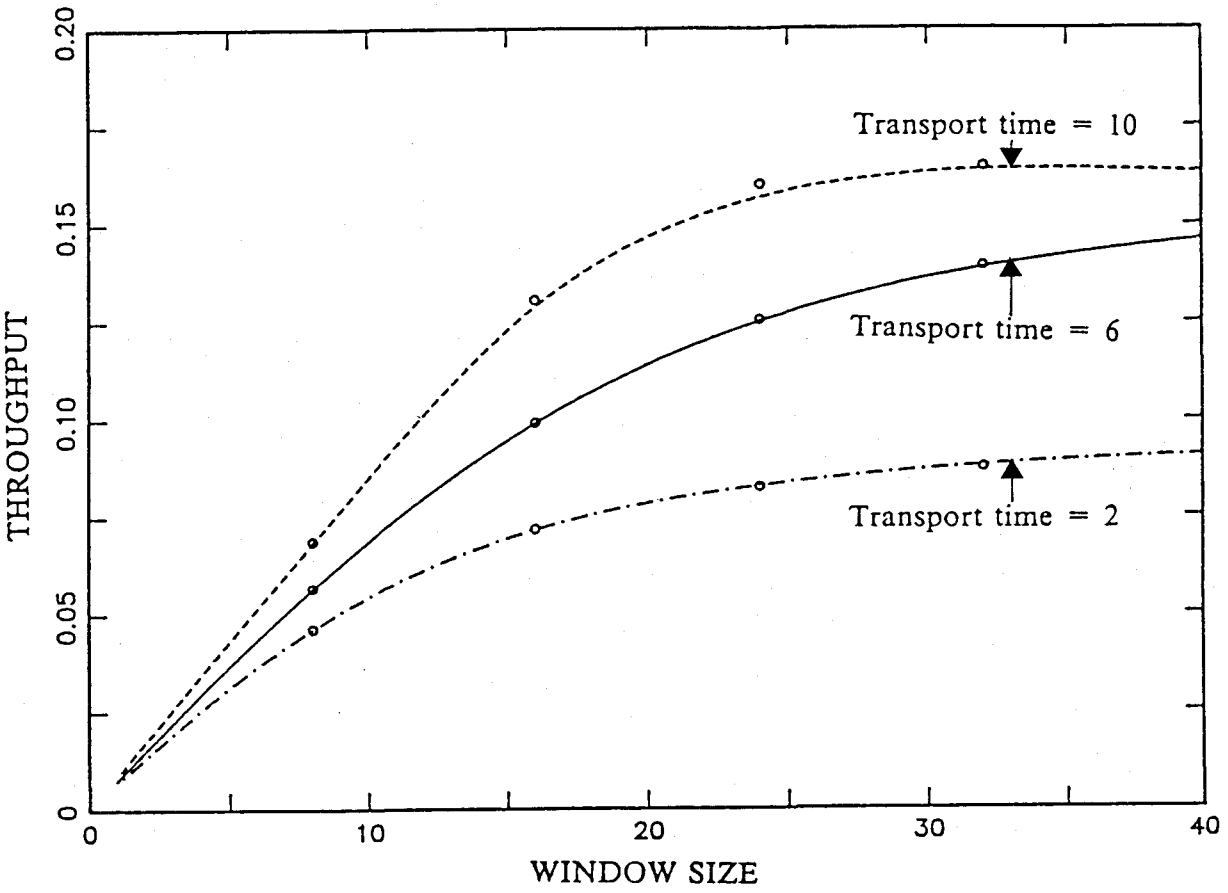


Figure 3.5-b. Throughput in the Case of a Symmetric-Load and Piggybacked Acknowledgement Model (Window Sizes and Processing Times at Transport Queues are Variable.)

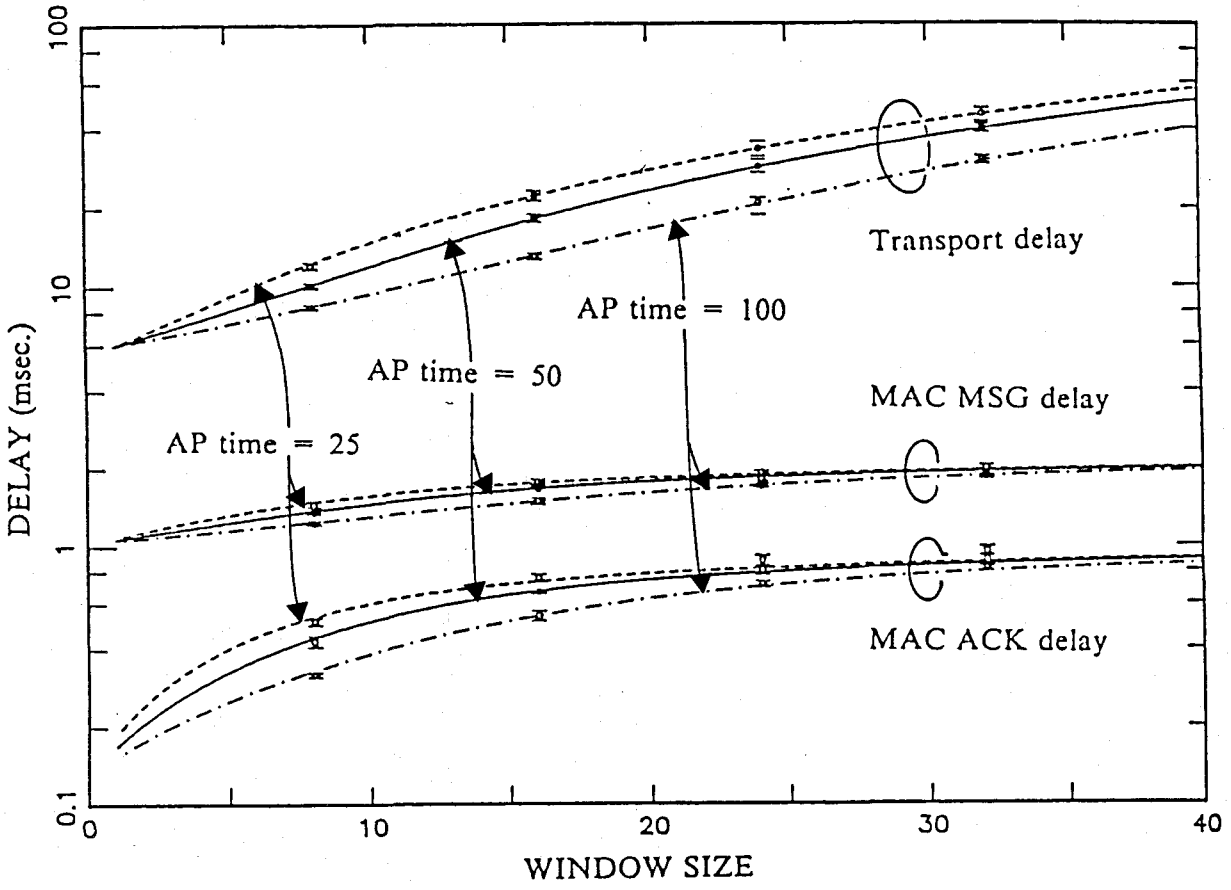


Figure 3.6-a. Delays in the Case of an Asymmetric-Load and Explicit Acknowledgement Model (Window Sizes are Variable.)

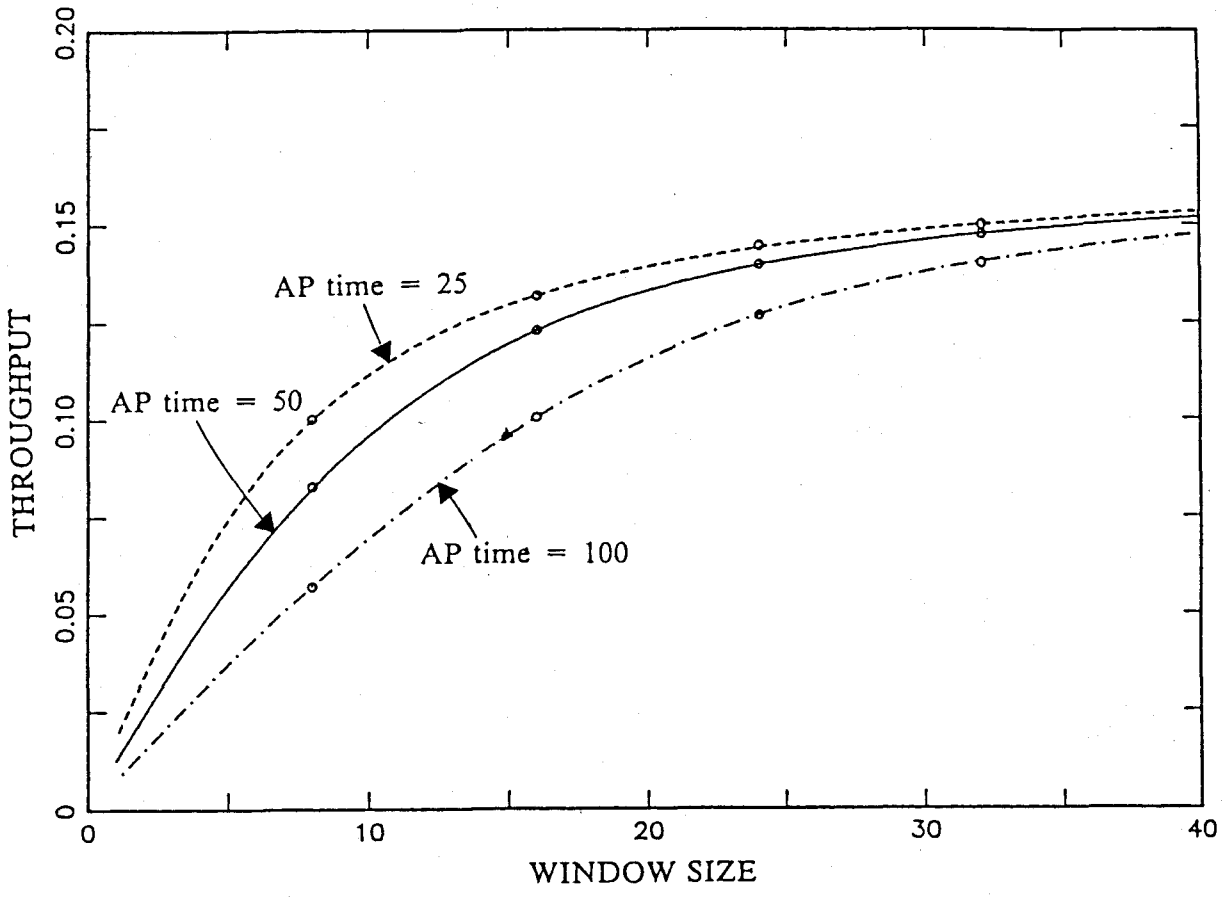


Figure 3.6-b. Throughput in the Case of an Asymmetric-Load and Explicit Acknowledgement Model (Window Sizes are Variable.)

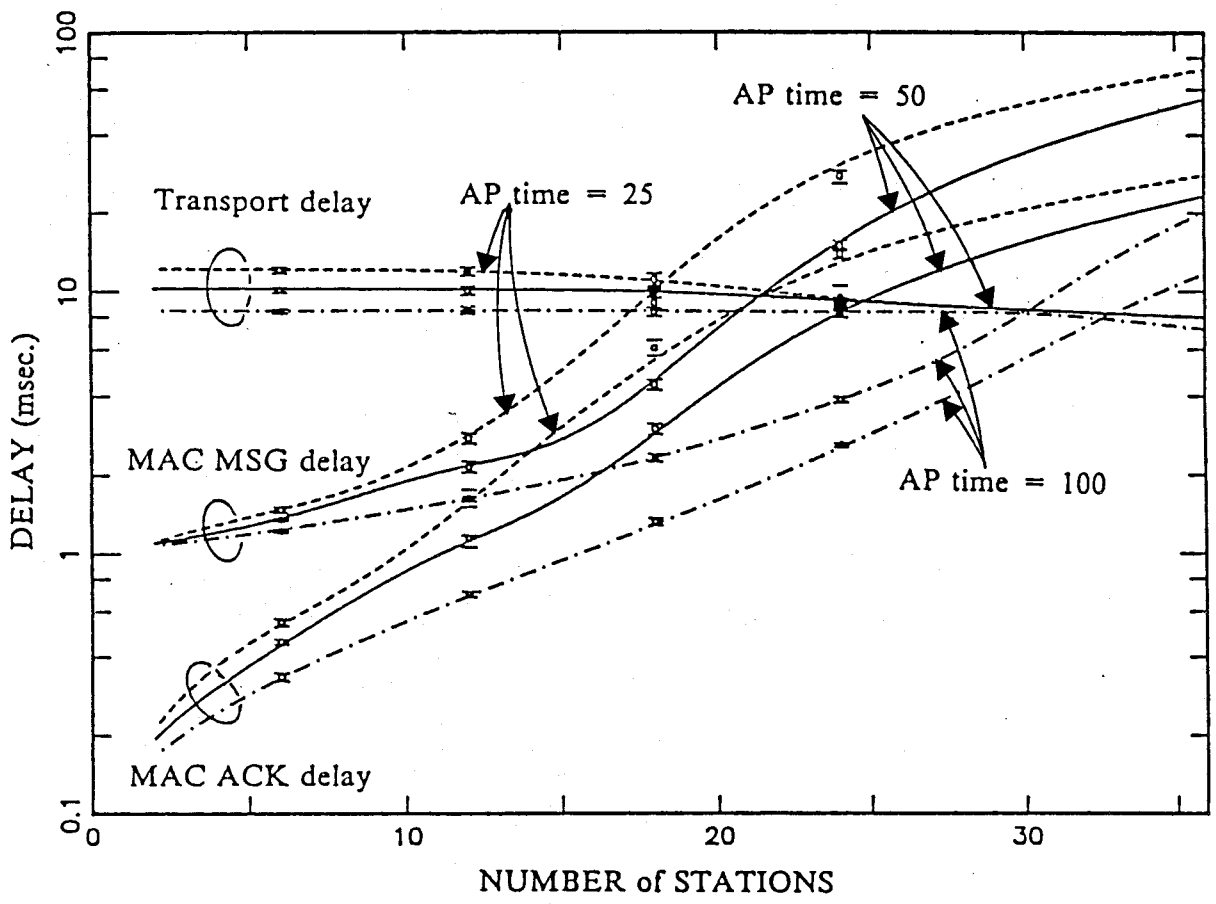


Figure 3.7-a. Delays in the Case of an Asymmetric-Load and Explicit Acknowledgement Model (*N* is Variable.)

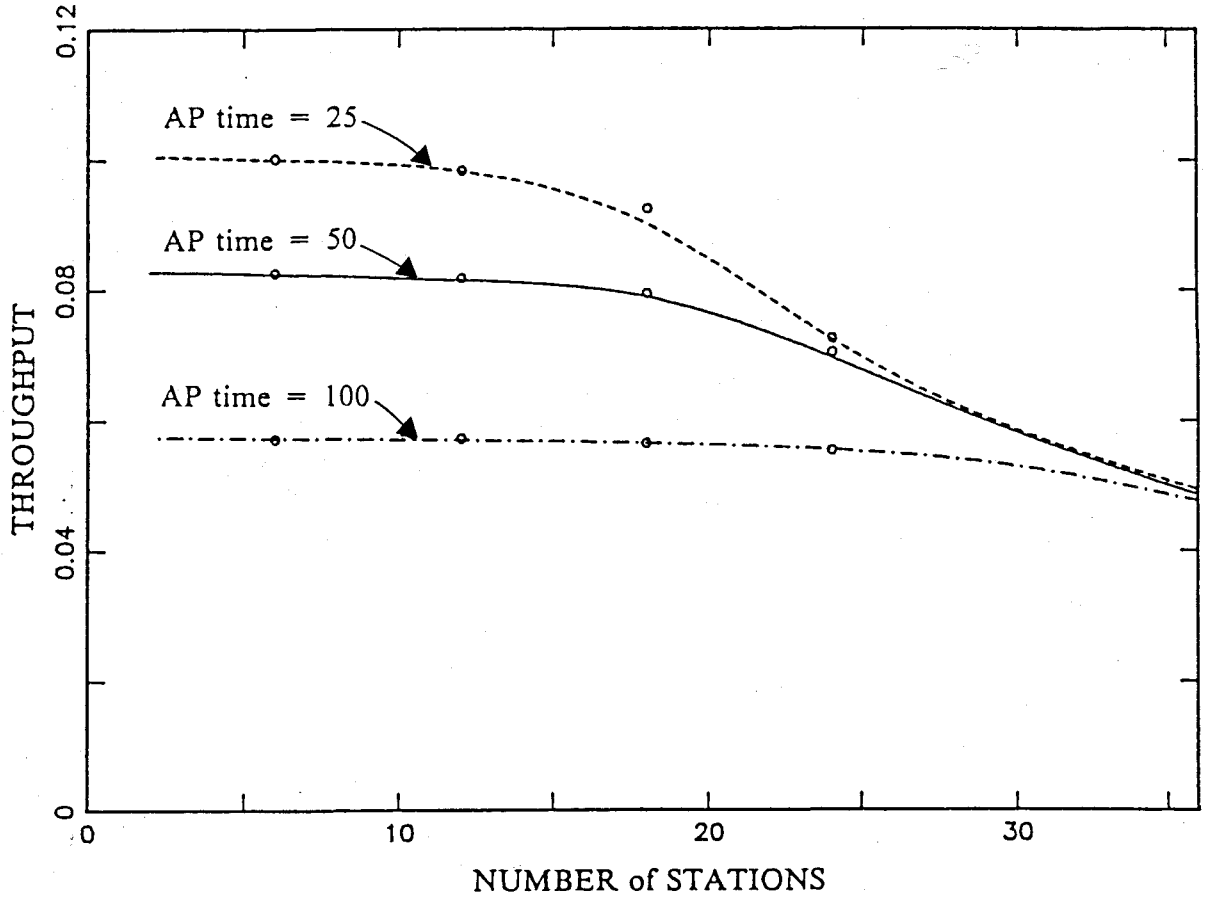


Figure 3.7-b. Throughput in the Case of an Asymmetric-Load and Explicit Acknowledgement Model (*N* is Variable.)

3.3. Multichain Closed-queueing Network for a Transport Layer

3.3.1. Solution Algorithm

In this section, we extend our solution method to investigate multichain cases where each station establishes multiple chains with several stations and it may be both a source and a sink at the same time. We need the following definitions for analyzing the extended model.

We consider the Transport layer submodel first. In the extended model, each queue may be visited by messages which belong to different chains so that we must introduce the MVA method with the multichain case [Reis79]. To start with, chains in the LAN are decomposed into several multichain closed-queueing networks so that each queue in a closed network does not have a link with the queue in another closed network. For each multichain closed-queueing network c, define the following notations:

$R^{(c)}$: Number of chains in closed network c

$W_r^{(c)}$: Window size of chain $r, r = 1, 2, \ldots, R^{(c)}$

$W^{(c)}$: Set of window sizes, which is $\left(W_1^{(c)}, W_2^{(c)}, \ldots, W_{R^{(c)}}^{(c)}\right)$

$R_j^{(c)}$: Set of chains visiting queue $j, j \in$ closed network c

$Q_r^{(c)}$: Set of queues in chain r, $r = 1, 2, \ldots, R^{(c)}$

$\tau_{rj}^{(c)}$: Mean service time of chain r message at queue $j, j \in$ closed network c ; $r \in R_j^{(c)}$

$n_{rj}^{(c)}(W^{(c)})$: Mean number of chain r messages waiting or being served at queue $j, j \in$ closed network c ; $r \in R_j^{(c)}$

$\lambda_r^{(c)}(W^{(c)})$: Throughput of chain $r, r = 1, 2, \ldots, R^{(c)}$

$t_{rj}^{(c)}(W^{(c)})$: Mean queueing time of chain r messages at queue $j, j \in$ closed network c ; $r \in R_j^{(c)}$

Thus we obtain the recursive relations for the multichain case as follows:

$$t_{rj}^{(c)}(W^{(c)}) = \begin{cases} f_i & i \text{ for given } c, r; \quad j = \text{MAC-queue} \\ \tau_{rj}^{(c)} & j = \text{AP-queue} \\ \tau_{rj}^{(c)} \left[1 + \sum_{k \in R_j^{(c)}} n_{kj}^{(c)}(W^{(c)} - e_r)\right] & \text{otherwise} \end{cases} \tag{3.10}$$

$$\lambda_r^{(c)}(W^{(c)}) = W_r^{(c)} / \sum_{j \in Q_r^{(c)}} t_{rj}^{(c)}(W^{(c)}) \tag{3.11}$$

and

$$n_{rj}^{(c)}(W^{(c)}) = \lambda_r^{(c)}(W^{(c)})\ t_{rj}^{(c)}(W^{(c)}) \tag{3.12}$$

where $W^{(c)} - e_r \triangleq (W_1^{(c)}, W_2^{(c)}, \ldots, W_{r-1}^{(c)}, W_r^{(c)} - 1, W_{r+1}^{(c)}, \ldots, W_{R^{(c)}}^{(c)})$. Obtained values $\lambda_r^{(c)}(W^{(c)})$ are used for the MAC layer submodel as described below.

For the multichain case, the arrival rate to the MAC-queue and the first and second moments of the message service time distribution function at the MAC-queue for station i are given by

$$\lambda_i = \sum_{r \in R^{(c)}(i)} \lambda_r^{(c)}(W^{(c)}), \tag{3.13}$$

$$b_i = \frac{1}{\lambda_i} \left(\sum_{r \in R^{(c)}(i)} \lambda_r^{(c)}(W^{(c)})\ b_r^{(c)} \right) \tag{3.14}$$

and

$$b_i^{(2)} = \frac{1}{\lambda_i} \left(\sum_{r \in R^{(c)}(i)} \lambda_r^{(c)}(W^{(c)})\ b_r^{(c)^{(2)}} \right) \tag{3.15}$$

where

$R^{(c)}(i)$: set of chains in closed network c visiting MAC-queue at station i (Note that station i belongs to one of a set of closed networks.),

and

$b_r^{(c)}, b_r^{(c)(2)}$: first and second moments of message transmission times in chain r of closed network c.

We can again use (3.4) with (3.13)-(3.15).

3.3.2. Numerical Results

3.3.2.1. Model 3. A Full-Duplex Communication Model

To evaluate the accuracy for the multichain cases, we first investigate the following model: every station has two sessions with another station, one is to send messages and the other is to receive messages (Figure 3.8). Acknowledgements are returned in the individual session. Within each session, when the message is received at the destination station, an acknowledgement is generated in the Transport-queue at the sending side and returned to the source station (no piggybacking). We assume that the processing times of Transport-queues equal to 6 *msec.* for this model.

The first numerical example for the model consists of six stations so that three separate closed networks exist in the LAN and each closed network has two chains (chain 1 and chain 2) between two stations. In Figure 3.9-a, results for delays are depicted dependent on the window size of chains. Window sizes for both chain 1 ($W_1^{(1)} = W_1^{(2)} = W_1^{(3)}$) and chain 2 ($W_2^{(1)} = W_2^{(2)} = W_2^{(3)}$) are kept identical to compare with Model 1 (Figure 3.3). In Figure 3.9-b, we see that throughput for each chain obtained in Model 3 is greater than that in Model 1 at a small size of windows, especially when the processing times at

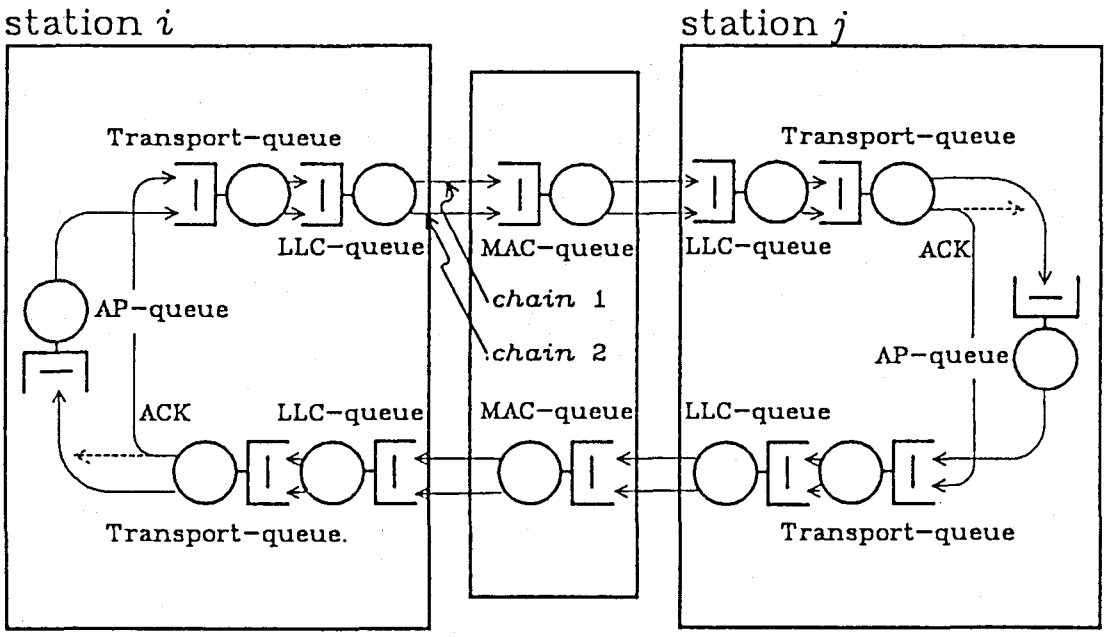


Figure 3.8. Transport Layer Submodel for a Full-Duplex Communication Model

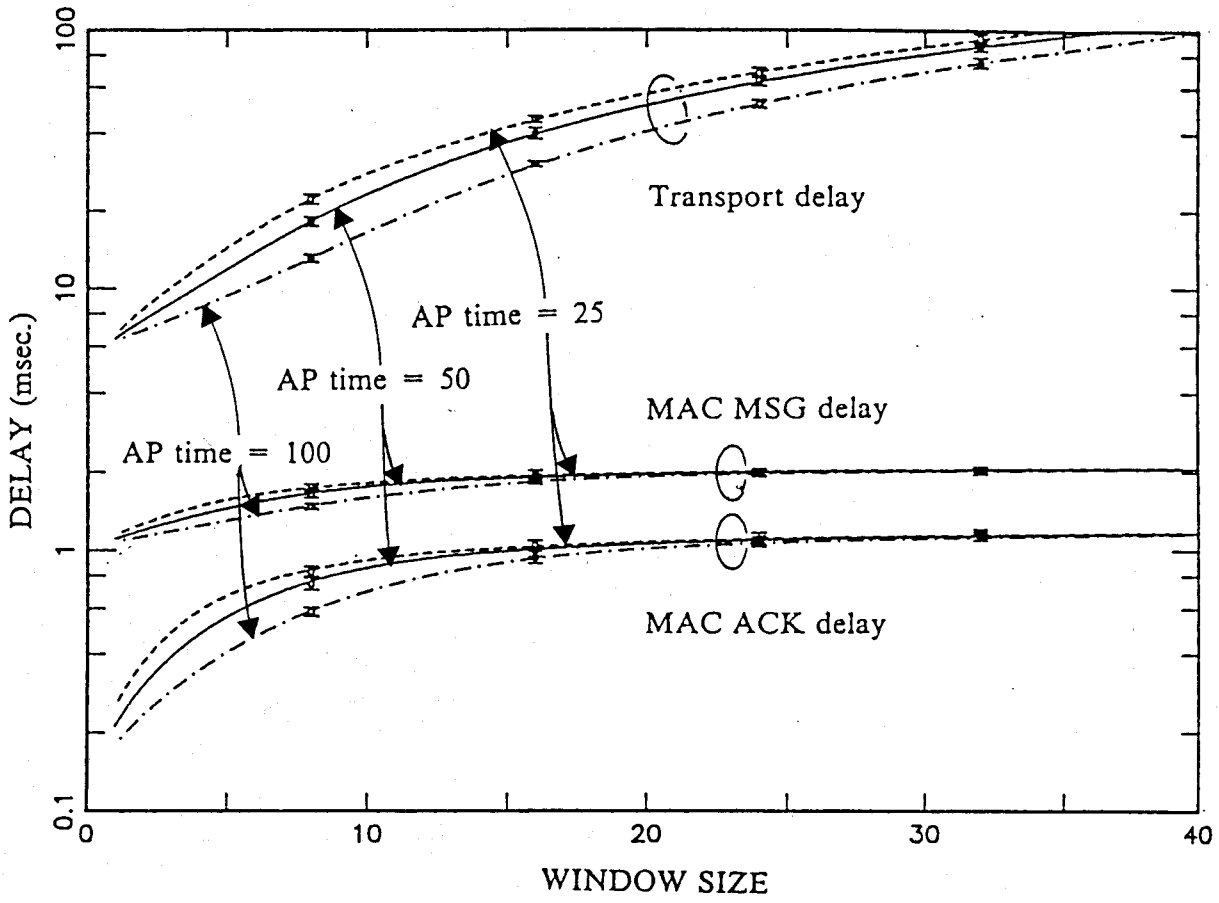


Figure 3.9-a. Delays of Chain 1 in the Case of a Full-Duplex Model (Window Sizes of Chain 1 and 2 are Variable.)

AP-queues are large. On the other hand, about two times throughput can be obtained in Model 1 when the window size becomes large. This phenomenon can be explained as follows: when the window size becomes small, it takes longer time until the next packet for piggybacking acknowledgement is generated at the destination AP-queue after the message from the source station arrives at the destination AP-queue. Such waiting times for the message at the destination tend to be large as the window size becomes small and/or the processing time of the destination AP-queue becomes large. Thus, we may conclude that the piggybacking method is more preferable for each chain when the window size is larger than 10 in the case of 50 *msec.* processing times of AP-queues and given parameters in this numerical example. However, the throughputs between a pair of two stations in Model 3 are larger than those in Model 1. This results in that the Transport delays are larger than those in Model 1 (see Figures 3.6.1-a and 3.3.1-a). On the other hand, MAC delays are smaller in spite of more offered traffic to the network. It is because, in Model 3, the returned messages are short-length acknowledgements. Another interesting point which is revealed by our numerical approach may be comparisons of response times between piggybacking and explicit acknowledgement methods. Here, we define the response time as elapsed time between the message generation time at the AP-queue in the source station and the time (i) when the piggybacked acknowledgement arrives at the source AP-queue in the case of the piggybacking method, or (ii) when the acknowledgement for the message arrives at the source AP-queue. Such performance comparisons will be reported later in Figure 3.17.

Next results show delays and throughputs against the number of stations (Figures 3.10). In these figures, the window size of chain 1 ($W_1^{(1)} = W_1^{(2)} = W_1^{(3)}$) in the closed network is fixed at 8 while three values, 4, 8 and 16, are used for the window size of chain 2 ($W_2^{(1)} = W_2^{(2)} = W_2^{(3)}$). Thus, the curves for $W_2^{(\bullet)} = 8$ are comparable with those in Figure 3.4. The analytical results overestimate the throughput by about 5% when the number of stations is large. Accordingly, both Transport delays and MAC delays are slightly underestimated. The worst estimation (-15% relative error) is obtained with $W_2 = 16$ and $N = 18$.

Next, we consider the situation with extremely unbalanced traffic condition between two chains. For this purpose, we change the window size of chain 2 from 1 to 40 while the window size of chain 1 is fixed at 8. The results are depicted in Figures 3.11-a,b with two values of the processing times at AP-queues:

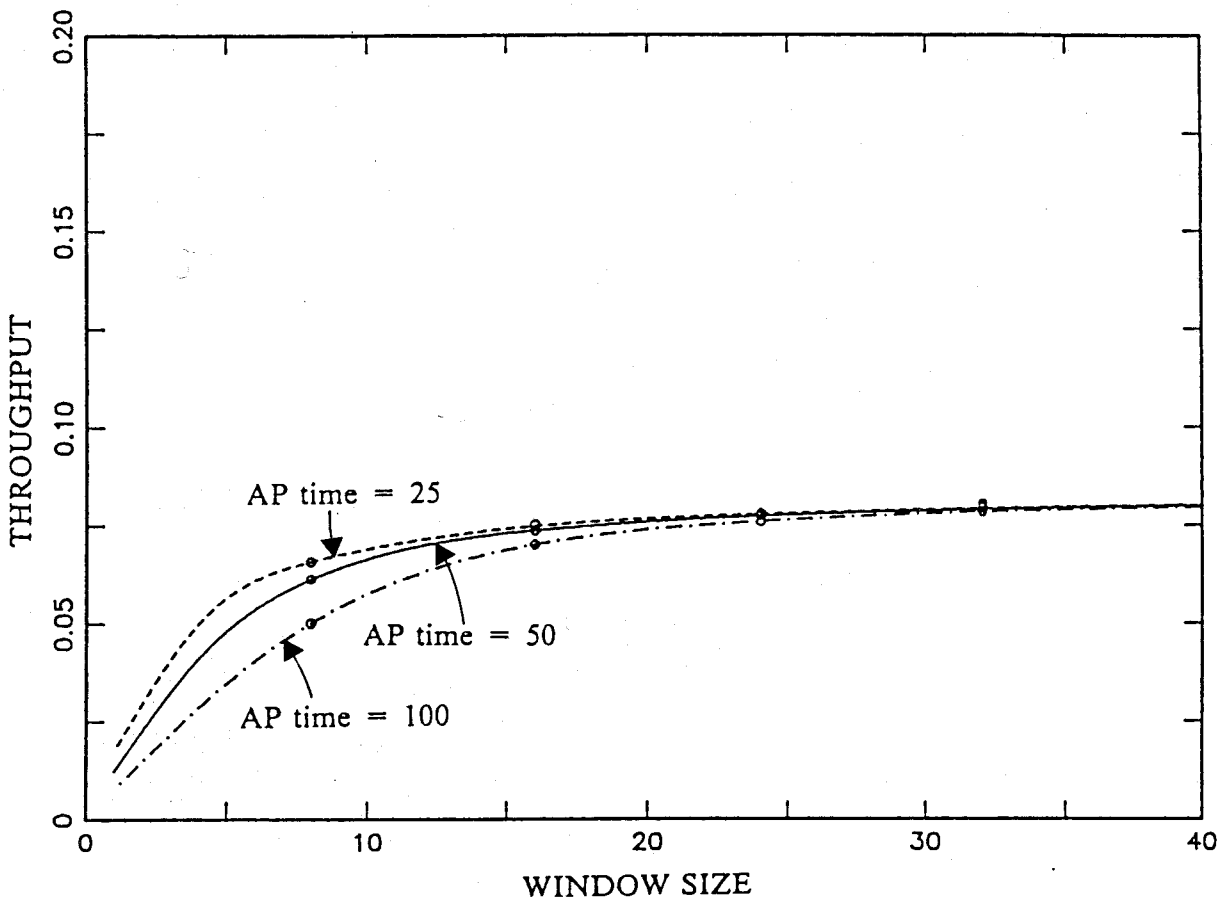


Figure 3.9-b. Throughput of Chain 1 in the Case of a Full-Duplex Model (Window Sizes of Chain 1 and 2 are Variable.)

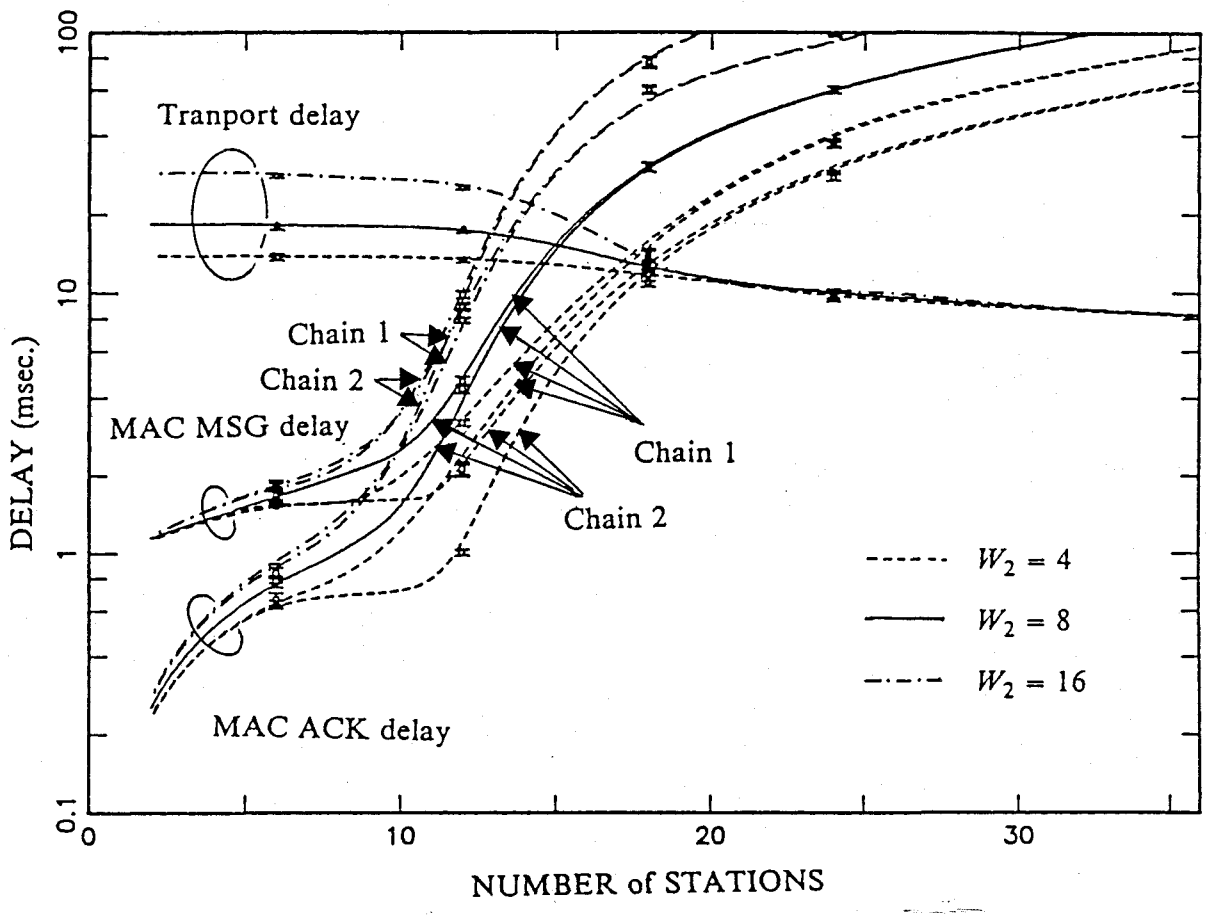


Figure 3.10-a. Delays in the Case of a Full-Duplex Model (N is Variable.)

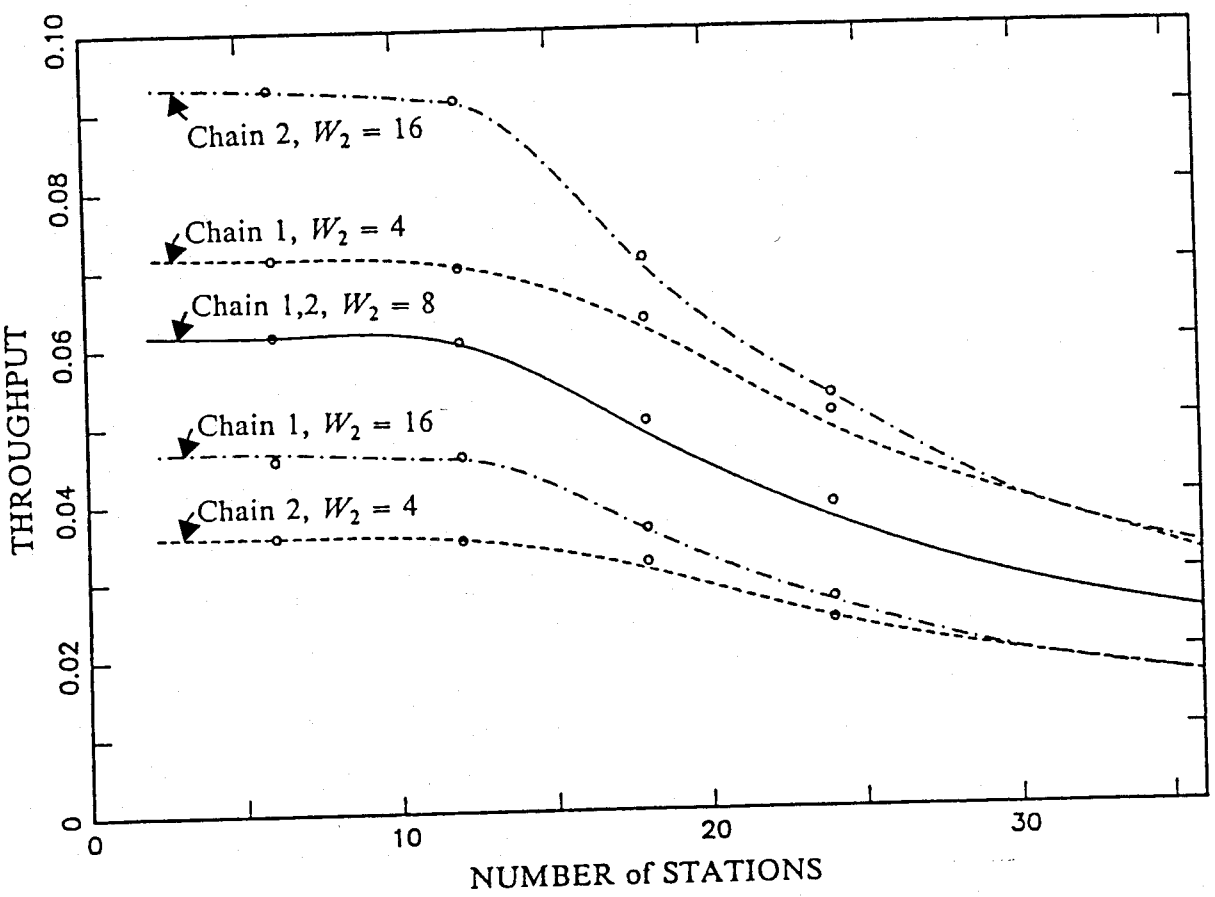


Figure 3.10-b. Throughput in the Case of a Full-Duplex Model (N is Variable.)

50 and 100 *msec.* In Figure 3.11-b, we see that the throughputs of chain 1 are degraded while the window size of chain 1 is not changed. It is because, as the window size of chain 2 becomes large, the throughput of chain 2 is also increased. Then, the queueing times at each queue tend to be large. This results in the performance degradation of chain 1.

3.3.2.2. Model 4. A Client/Server Model

Next, we investigate the Client/Server model in [Svob85] such that a server-station receives requests from client-stations, processes the requests, and returns responses to the requesters in the client-stations (Figure 3.12). In our numerical example, one server-station and three client-stations are assumed to be connected via a LAN. So, we have one closed network with three chains (chain 1, chain 2 and chain 3) between clients and the server ($N = 4, c = 1, R^{(1)} = 3$). Mean message delays and throughput for each chain are depicted in Figure 3.13. The window size of chain 1 ($W_1^{(1)}$) is changed while those of chain 2 and chain 3 ($W_2^{(1)}, W_3^{(1)}$) are fixed at 8. We assume the processing times of client-stations equal to 50 *msec.*, and that of the server station is 5 *msec.* We further assume that the processing times of Transport-queues equal to 6 *msec.*. In this model, we use an FCFS queue for the AP-queue in the server-station while the AP-queues in the client stations are modelled by IS queues. Here again we have excellent agreement between our computation and simulation.

3.4. Transport-layer Submodel with Priorities

In Model 3, we have assumed that data-messages and acknowledgements are handled with same priorities at Transport-queues. However, in real systems, acknowledgements may have a priority over data-messages. Namely, when the station receives the message, it is interrupted and generates the acknowledgement in a preemptive or nonpreemptive fashion. In the following, we take into account priority service disciplines for Transport-queues. In solving the multichain queueing network including priority service

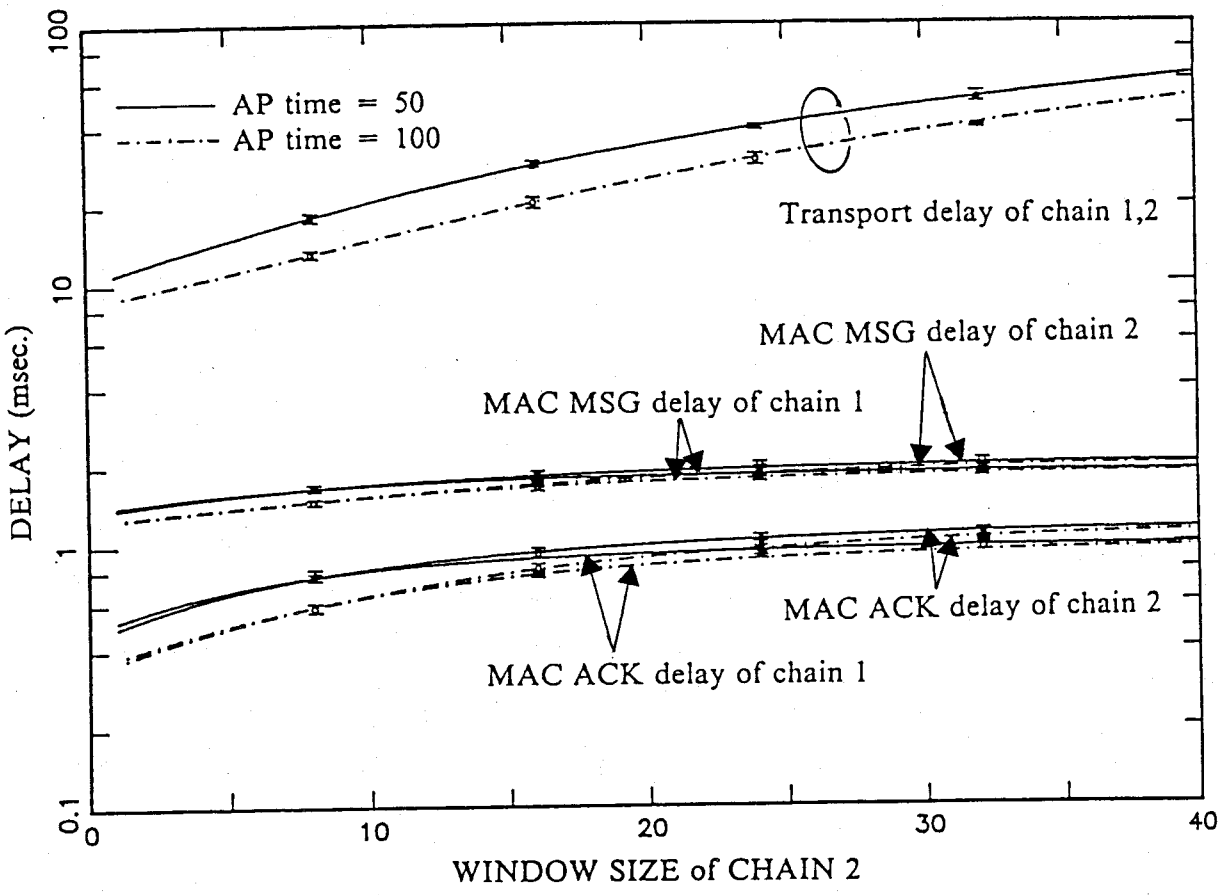


Figure 3.11-a. Delays in the Case of a Full-Duplex Model (The Window Size of Chain 2 is Variable.)

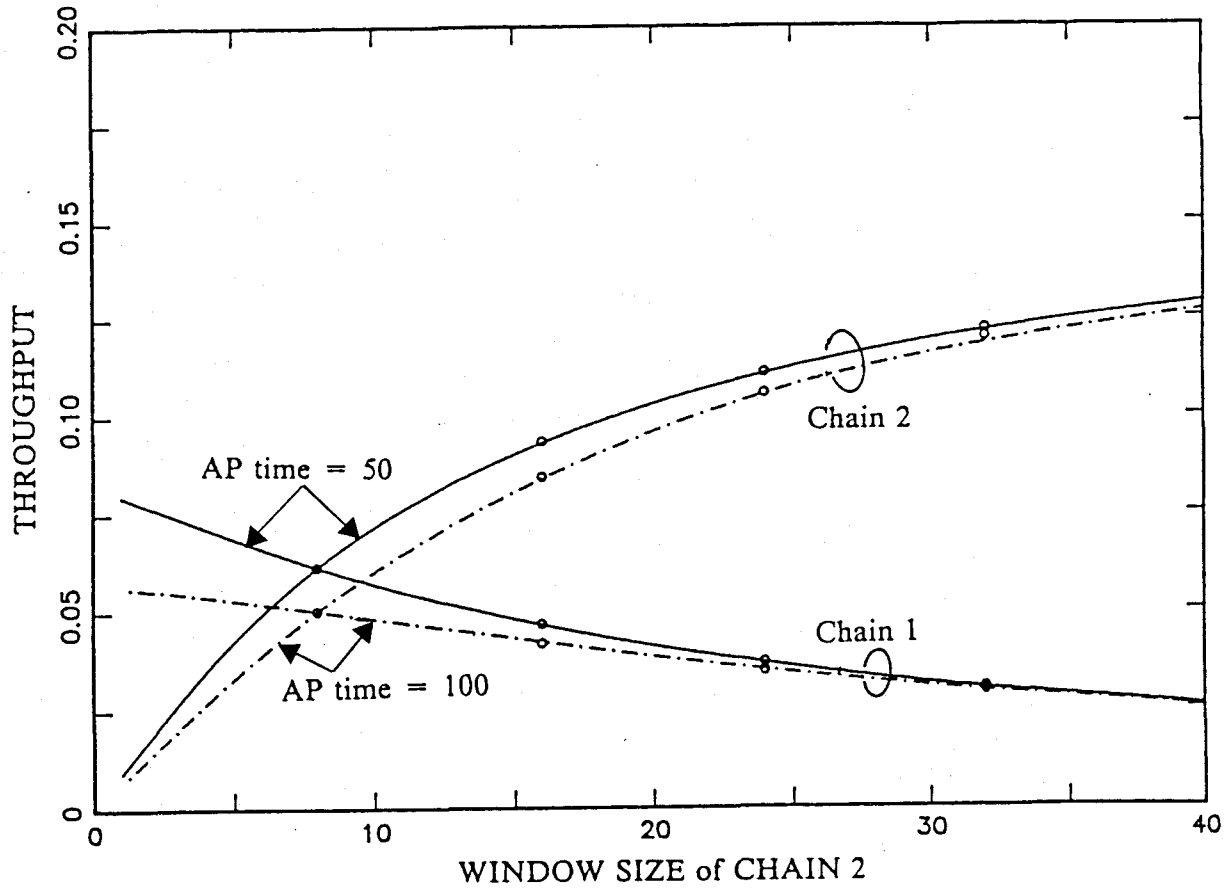


Figure 3.11-b. Throughput in the Case of a Full-Duplex Model (The Window Size of Chain 2 is Variable.)

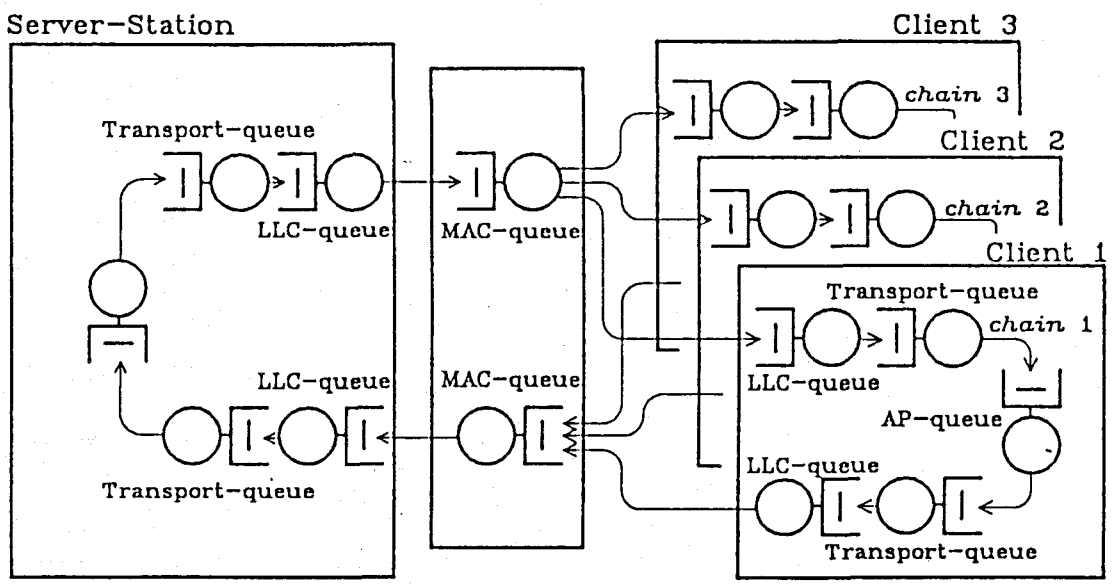


Figure 3.12. Transport Layer Submodel in the Case of a Client/Server Model

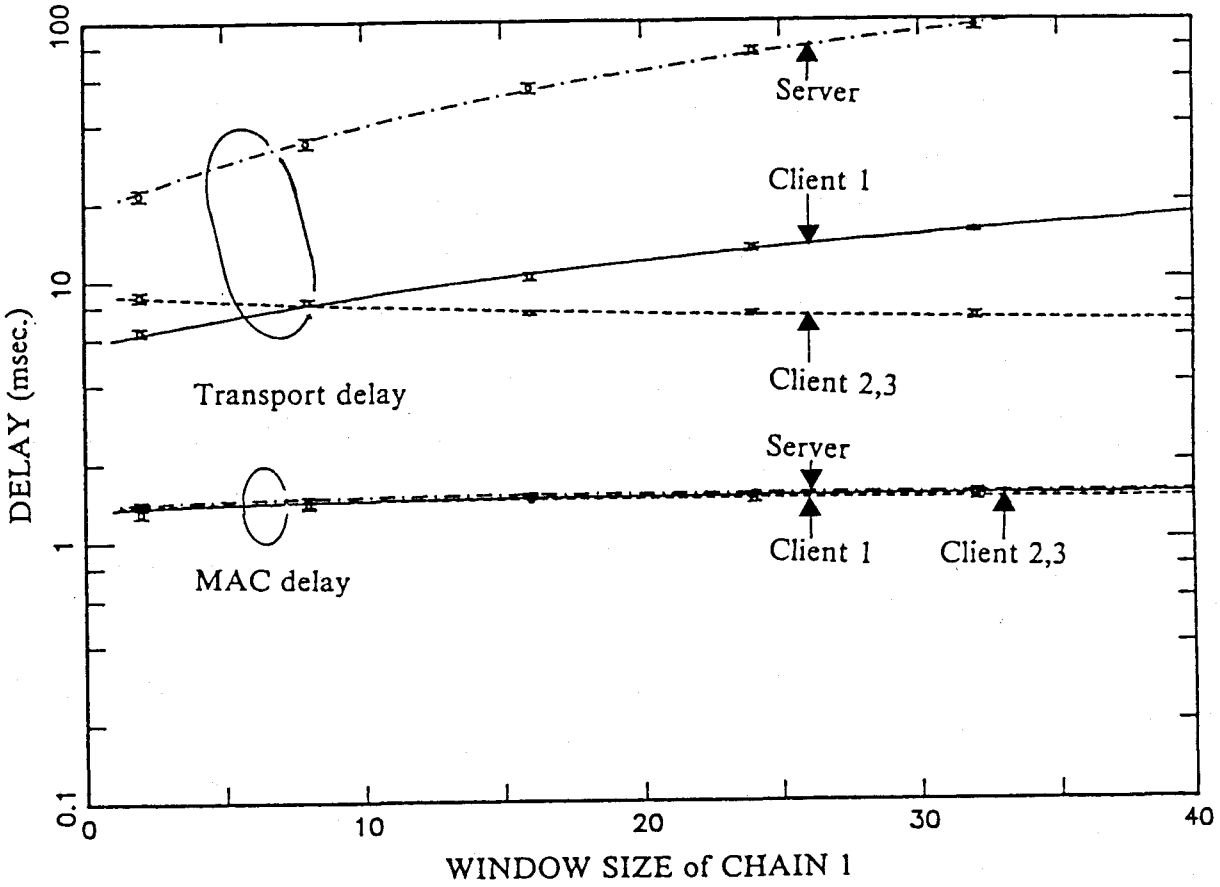


Figure 3.13-a. Delays in the Case of a Client/Server Model (The window size of chain 1 is variable)

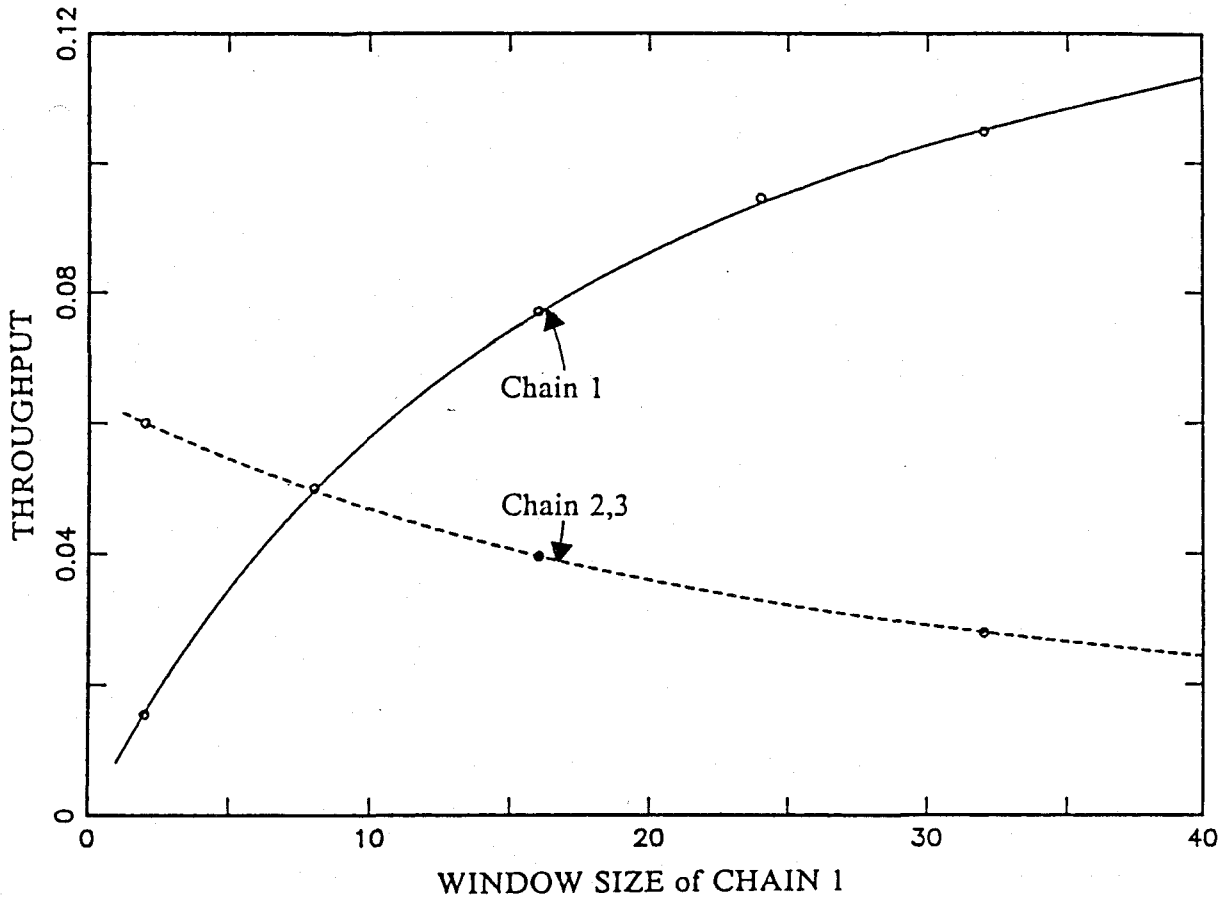


Figure 3.13-b. Throughput in the Case of a Client/Server Model (The window size of chain 1 is variable)

queues, we cannot apply the previously-mentioned approach because we do not have the exact product form solution. Thus, let us follow the MVA priority approximation method [Brya84] for the solution of Transport-layer submodel with priorities. Other approaches can be found in [Kauf84] and [Schm84] where they decompose the priority queue into separate *virtual* queues for each priority class. Although the latter approaches seem to bring good approximations, they require either class or queue aggregation and solutions to global/local balance equations to get a priori unknown service rates at separate priority queues. Therefore, their approaches produce unacceptable computational efforts to apply to our case.

On the other hand, the MVA priority approximation seems to be good in saving computational efforts. However, we find it difficult to apply it directly to our case because the class change through a chain is not considered there. Namely, each chain is assigned the priority class identical throughout the network. Thus, we need to modify the MVA priority approximation so that the priority of each chain can be chosen at each queue. Such an algorithm is newly proposed in the next section.

3.4.1. MVA Priority Approximation with Priority Changeable at Each Queue

First, we summarize the original MVA priority approximation method [Brya84]. In the MVA priority approximation, a network consists of a set of closed chains and each chain is assigned a fixed priority class throughout the network. Similar to MVA, performance quantities at each window size, W, are calculated as follows (we use the same notation as defined in Section 3.3.1 except that we omit the superscript (c) for simplicity.) :

$$t_{rj}(W) = \begin{cases} \tau_{rj}\left[1 + \sum_{k \in R_j} n_{kj}(W - e_r)\right] & \text{if } j = \text{non-priority queue} \\ \dfrac{\tau_{rj} + \sum_{k \in R_{jr}^H} n_{kj}(W - e_r)\,\tau_{kj}}{1 + \rho_{rj}' - \sum_{k \in R_{jr}^H} \rho_{kj}'} & \text{if } j = \text{preemptive priority queue} \\ \tau_{rj} + \dfrac{\sum_{k \in R_{jr}^H} n_{kj}(W - e_r)\,\tau_{kj} + \sum_{k \in R_{jr}^H} \rho_{kj}(W - e_r)\,\tau_{kj}}{1 + \rho_{rj}' - \sum_{k \in R_{jr}^H} \rho_{kj}'} & \text{if } j = \text{non-preemptive priority queue} \end{cases} \tag{3.16}$$

$$\lambda_r(W) = W_r \,/\, \sum_{j \in Q_r} t_{rj}(W) \tag{3.17}$$

and

$$n_{rj}(W) = \lambda_r(W)\, t_{rj}(W) \tag{3.18}$$

where R_{jr}^L is a set of chains with priorities lower than or equal to chain r at queue j and R_{jr}^H is a set of chains with priorities higher than chain r at queue j. For the MVA priority approximation, we must store another quantity $\rho_{rj}(W)$, which is the traffic intensity of chain r at queue j given by

$$\rho_{rj}(W) = t_{rj}(W)\,\tau_{rj} \tag{3.19}$$

and ρ_{kj}' in (3.16) is defined by

$$\rho_{kj}' = \rho_{kj}(W_1, W_2, \ldots, W_{k-1}, W_k - n_{kj}(W), W_{k+1}, \ldots, W_R) \tag{3.20}$$

which represents the traffic intensity at the window size W with chain k messages fewer by $n_{kj}(W)$. At each window size, a set of four quantities, $t_{rj}(W)$, $\lambda_r(W)$, $n_{rj}(W)$ and $\rho_{kj}(W)$, are calculated with the priority index order. Note that the above operations can be achieved because the index of chain has the same implication as the priority class throughout the network, and all R_{jr}^L and R_{jr}^H are identical independent on queue j.

However, we cannot apply this method directly to our model where class changes are allowed because, in our model, R_{jr}^{L} or R_{jr}^{H} are determined at each queue j and, for even highest priority class, the quantity ρ_{kj}' cannot be predetermined. To perform it, we introduce another iteration at each window size described below:

Step 1: Initialize $\rho_{rj}(W)$ at some value to reduce the number of iteration cycles.

Step 2: Calculate $t_{rj}(W)$, $\lambda_r(W)$ and $n_{rj}(W)$ using (3.16) through (3.18).

Step 3: Calculate $\rho_{rj}(W)$ using (3.19). If the iteration is not converged, then go to Step 2 for the next iteration cycle.

In Step 1, since $\rho_{rj}(W)$ is expected to be slightly greater than the known value of $\rho_{rj}(w - e_r)$, we may take the latter as a initial value for $\rho_{rj}(W)$ to reduce the number of iteration cycles.

3.4.2. Numerical Results

3.4.2.1. Model 5. A Full-Duplex Communication Model with Priorities

For Model 5, we consider a full-duplex communication model (Figure 3.8) where acknowledgements have a priority over messages at each Transport-queue and are processed in a preemptive fashion. For the solution, we replace (3.10) by

$$t_{rj}^{(c)}(W^{(c)}) = \begin{cases} f_i & i \text{ for given } c, r;\ j = \text{MAC-queue} \\ \tau_{rj}^{(c)} & j = \text{AP-queue} \\ \tau_{rj}^{(c)}\left[1 + \sum_{k \in R_j^{(c)}} n_{kj}^{(c)}(W^{(c)} - e_r)\right] & j = \text{LLC-queue} \\ \dfrac{\tau_{rj}^{(c)} + \sum_{k \in R_{jr}^{H(c)}} n_{kj}^{(c)}(W^{(c)} - e_r)\,\tau_{kj}}{1 + \rho_{rj}{}'^{(c)} - \sum_{k \in R_{jr}^{H(c)}} \rho_{kj}{}'^{(c)}} & j = \text{Transport-queue} \end{cases} \tag{3.21}$$

where $c = 1, 2, 3$, $r = 1, 2$, and $R_{jr}^{L(c)} = \{r\}, r = 1, 2$ when data-messages are processed at Transport-queue j in chain r in closed network c, and $R_{jr}^{L(c)} = \{1, 2\}, r = 1, 2$ when acknowledgements are processed at Transport-queue j.

To validate our approximation for Model 5, we use the same parameters as Model 3, except that two values of the processing times at Transport-queues, (i) 2 *msec.* and (ii) 6 *msec.*, are used for acknowledgements. The processing times for data-messages are fixed at 6 *msec.* (These parameters are used for all Transport-queues in both transmitting and receiving sides.) In Figure 3.14, we show the mean message delays and throughputs against the window size of chains. In Figure 3.14-a, we observe that the relative errors of the Transport delays of acknowledgements are about +20% in the case of 6 *msec.* processing times of the acknowledgements at the Transport-queues. On the other hand, the relative errors are below +5% in the case of 2 *msec.* processing times. Such a property was observed by our preliminary investigations for the accuracy of our modified MVA with priority approximation method (although they are not included in this chapter). Namely, when the processing time of the higher priority class is relatively smaller than that of the lower priority class, the accuracy of delays for both classes is good. On the other hand, the relative errors are beyond +20 or +30% when the processing times of the higher priority class become large. However, we note here that the processing time for the acknowledgements with high priority tends to be smaller than that for data-messages in actual situations. The relative errors of the Transport delays for the data-messages are always below 3% in our experiments. The MAC delays of acknowledgements

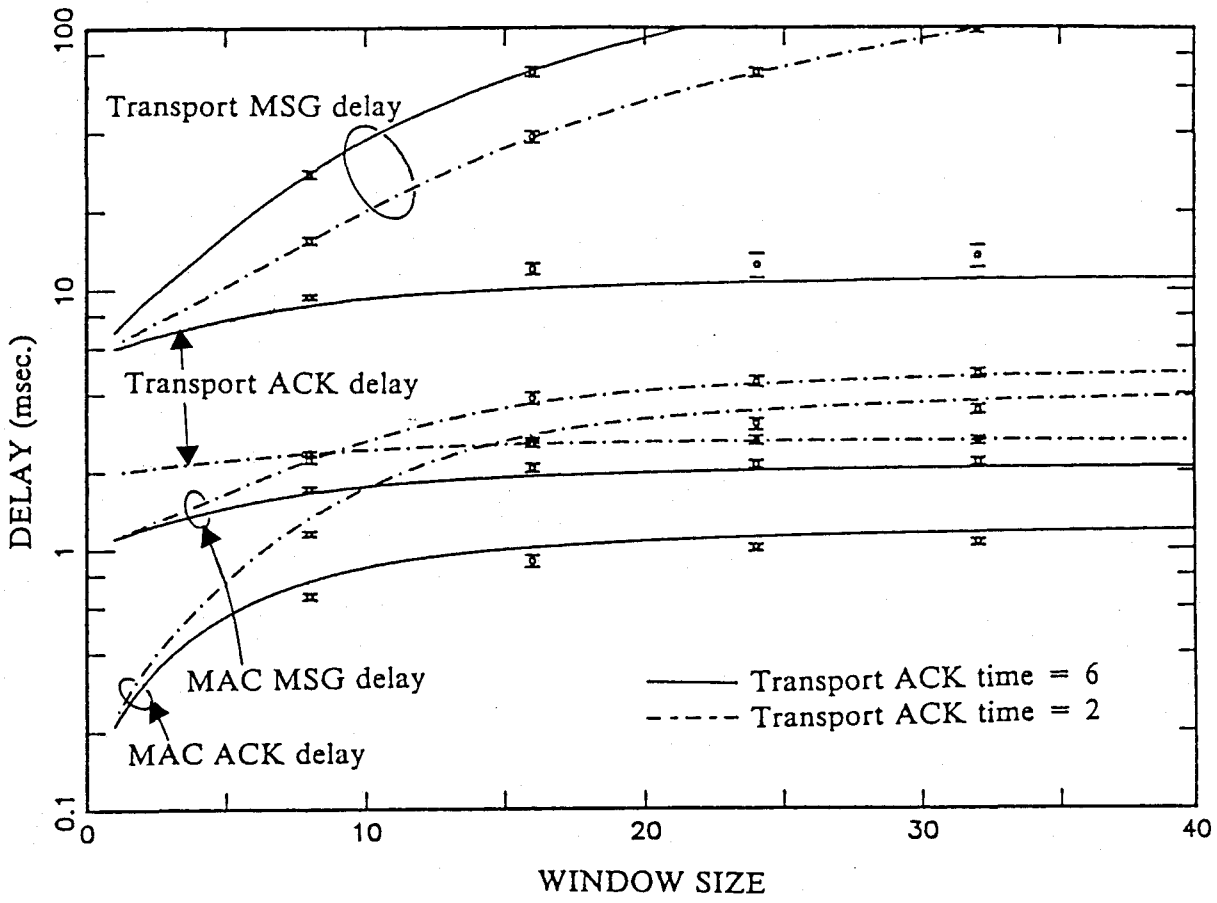


Figure 3.14-a. Delays of Chain 1 in the Case of a Full-Duplex with Priorities Model (Window Sizes of Chain 1 and 2 are Variable)

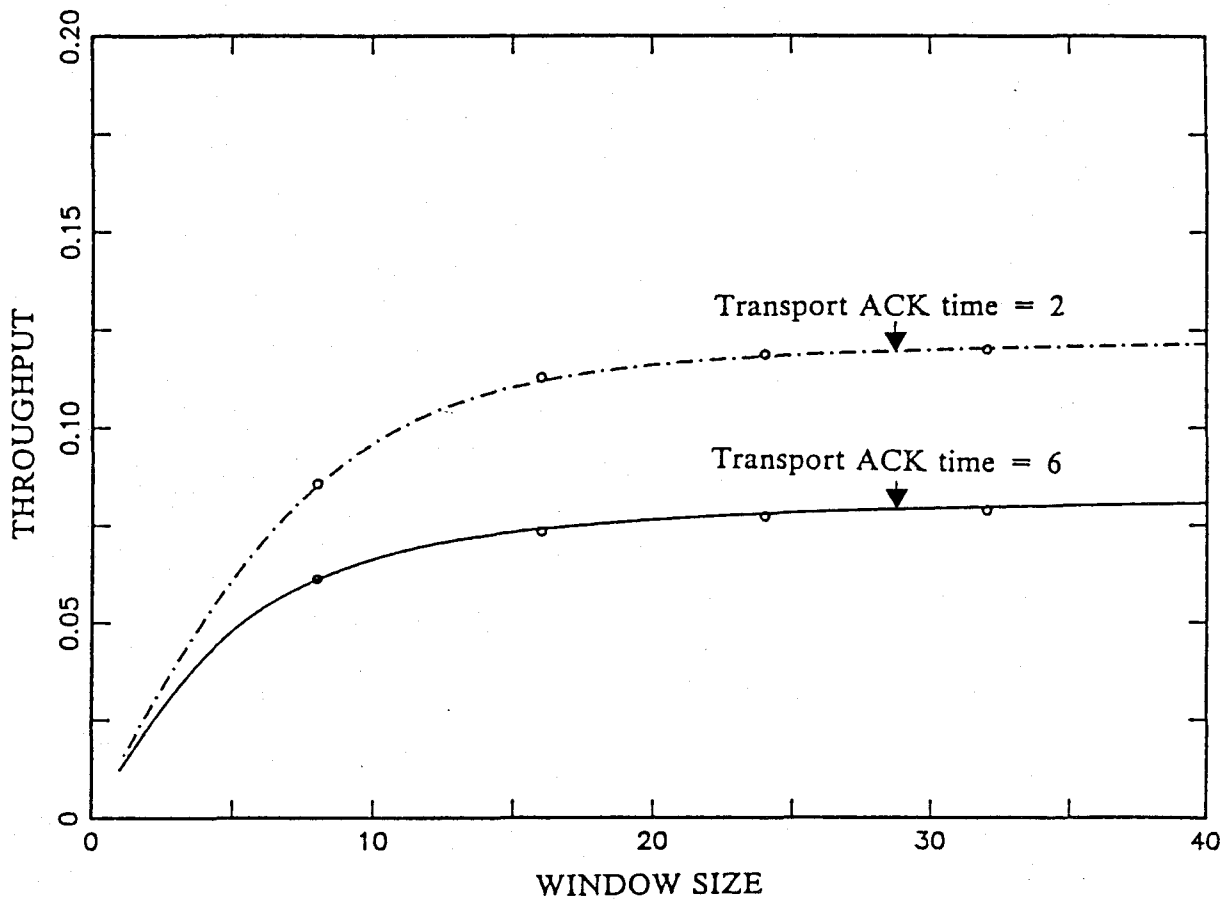


Figure 3.14-b. Throughput in the Case of a Full-Duplex with Priorities Model (Window Sizes of Chain 1 and 2 are Variable)

are always underestimated by about 10%. These errors are considered to be caused by (i) the polling approximation method and/or (ii) the Poisson arrival assumption at the MAC-queues in the analysis. In Figure 3.14-a, the Transport delays of acknowledgements are not increased by the introduction of the priority mechanism in spite of the large window sizes. Such small delays are obtained by the compensation of the large delays of the data-messages.

Next, we illustrate the delays and throughputs dependent on the number of stations in Figure 3.15. We observe that the introduction of the priority mechanism does not affect the accuracy of delays in the case where the MAC-queues are bottleneck.

For the last simulation results, we change the window sizes of chain 2 ($W_2^{(1)} = W_2^{(2)} = W_2^{(3)}$) from 1 to 40 while the window size of chain 1 ($W_2^{(1)} = W_2^{(2)} = W_2^{(3)}$) is fixed at 8. The delays of chain 1 and 2 are depicted in Figures 3.16-a and b, respectively. In Figure 3.16-c, we see that the throughputs of chain 2 are larger than t he corresponding ones in Figure 3.6-b when the window size of chain 2 is increased. On the other hand, the throughput of chain 1 is degraded compared to the full-duplex model without priority Transport-queues. It is because the acknowledgements belonging to chain 2 are processed prior to the data-messages of chain 1 at the Transport-queues in the transmitting side of chain 1. This observation can be confirmed by the fact that the Transport delays of the data-messages in chain 1 are large in Figure 3.16-a. On the other hand, when the processing times of acknowledgements at the Transport-queues are 2 *msec.*, the acknowledgements occupies the queue in less time and, therefore, the data-messages in chain 1 are served more. Thus, the throughputs of chain 1 are not so degraded, and the Transport delays for the data-messages of chain 1 are kept below 20 *msec.* We further see that the throughputs of chain 2 in the case of 6 *msec.* acknowledgement processing times at the Transport-queues are larger than in the case of 2 *msec.* processing times when the window size of chain 2 is increased. It is because the throughputs of chain 1 are severely degraded by the processing of the acknowledgements of chain 2 at the Transport-queues.

Lastly, we compare the performance of three models (Model 1, Model 3 and Model 5) to demonstrate the applicability of our modelling approach. In Figure 3.17, we show the response times and network

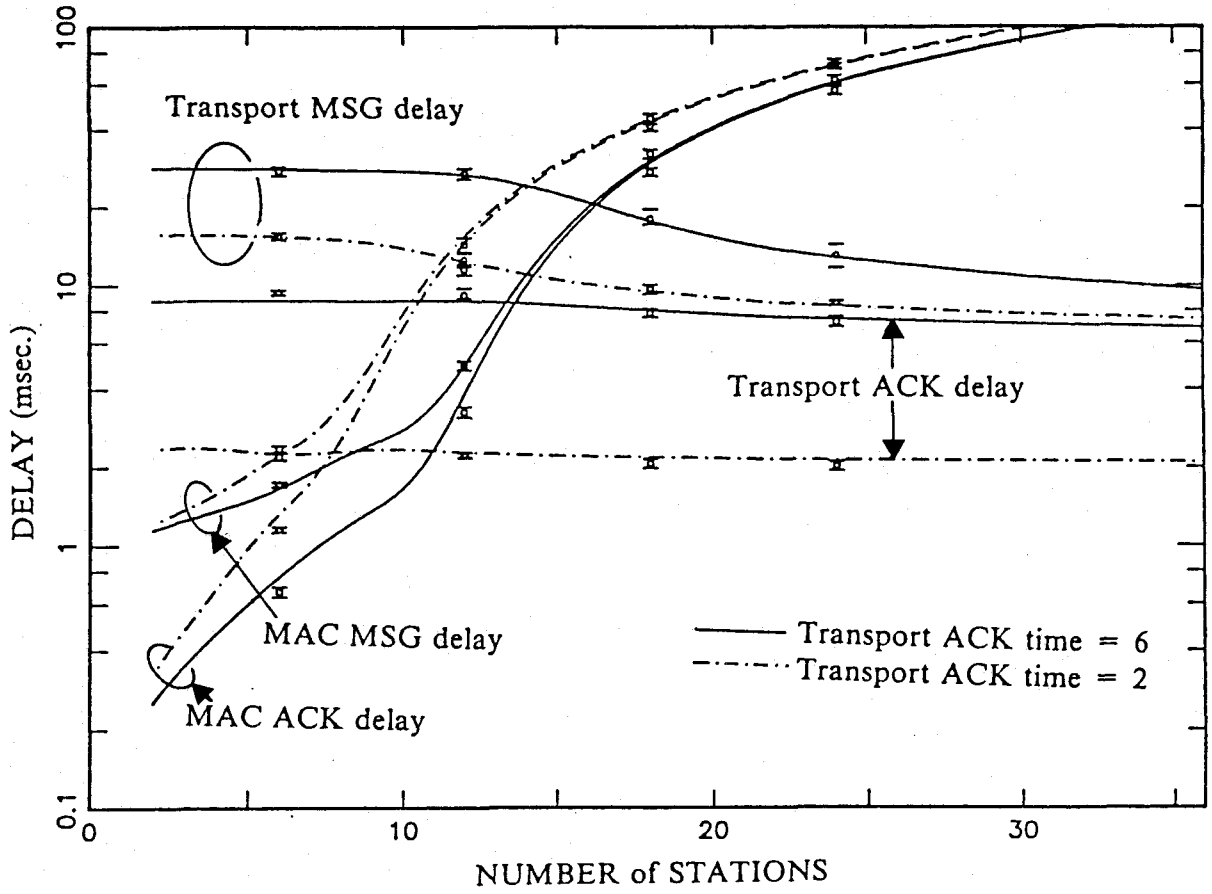


Figure 3.15-a. Delays of Chain 1 in the Case of a Full-Duplex with Priorities Model (*N* is Variable)

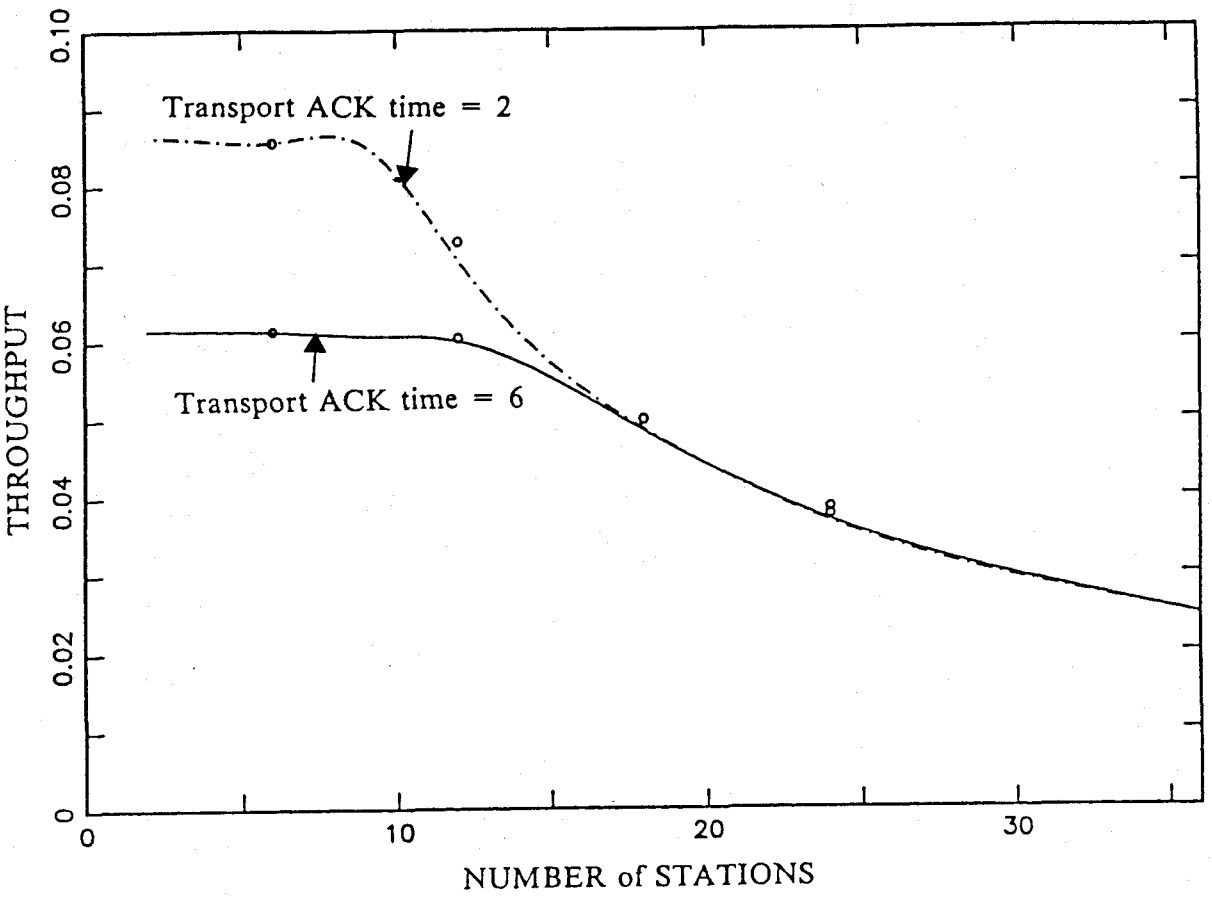


Figure 3.15-b. Throughput in the Case of a Full-Duplex with Priorities Model (*N* is Variable)

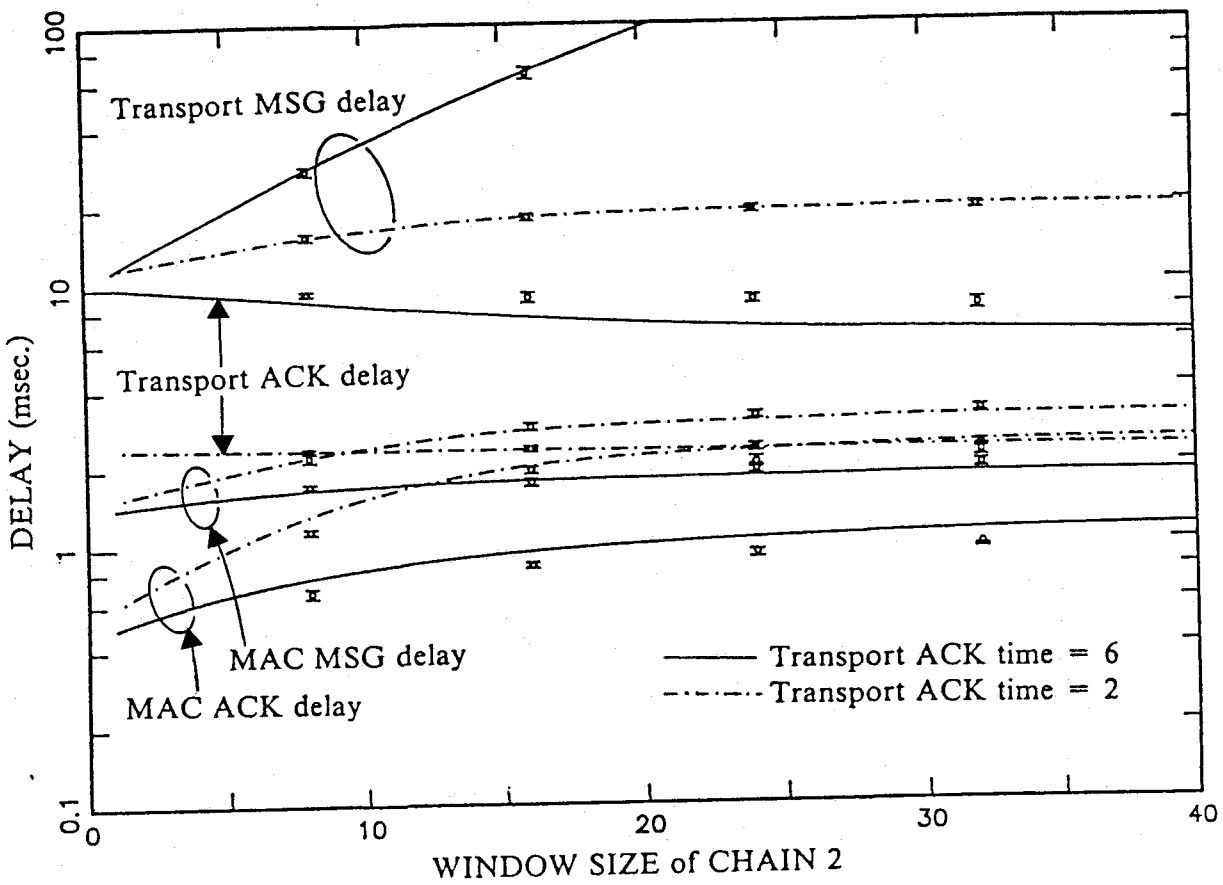


Figure 3.16-a. Delays of Chain 1 in the Case of a Full-Duplex with Priorities Model (The Window Sizes of Chain 1 are Variable)

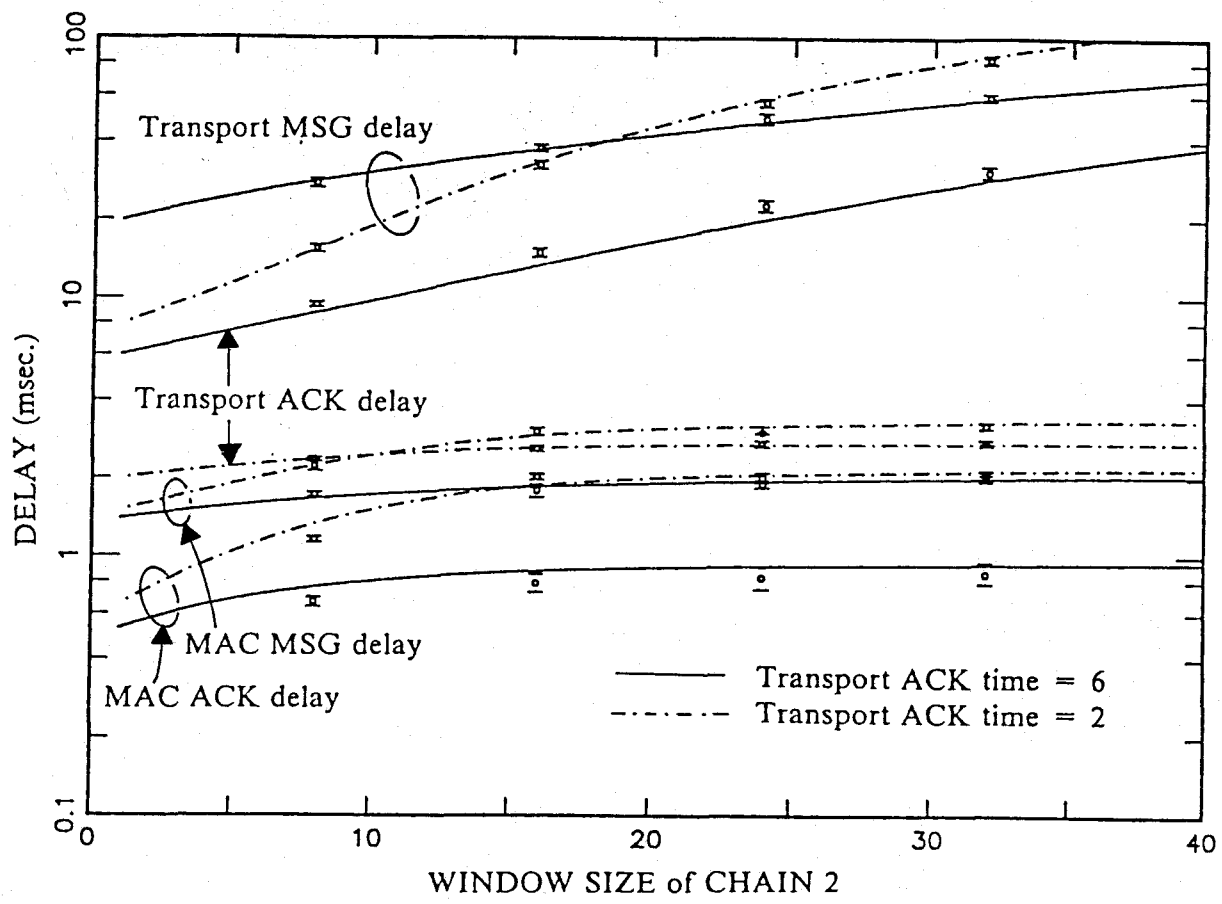


Figure 3.16-b. Delays of Chain 2 in the Case of a Full-Duplex with Priorities Model (The Window Sizes of Chain 1 are Variable)

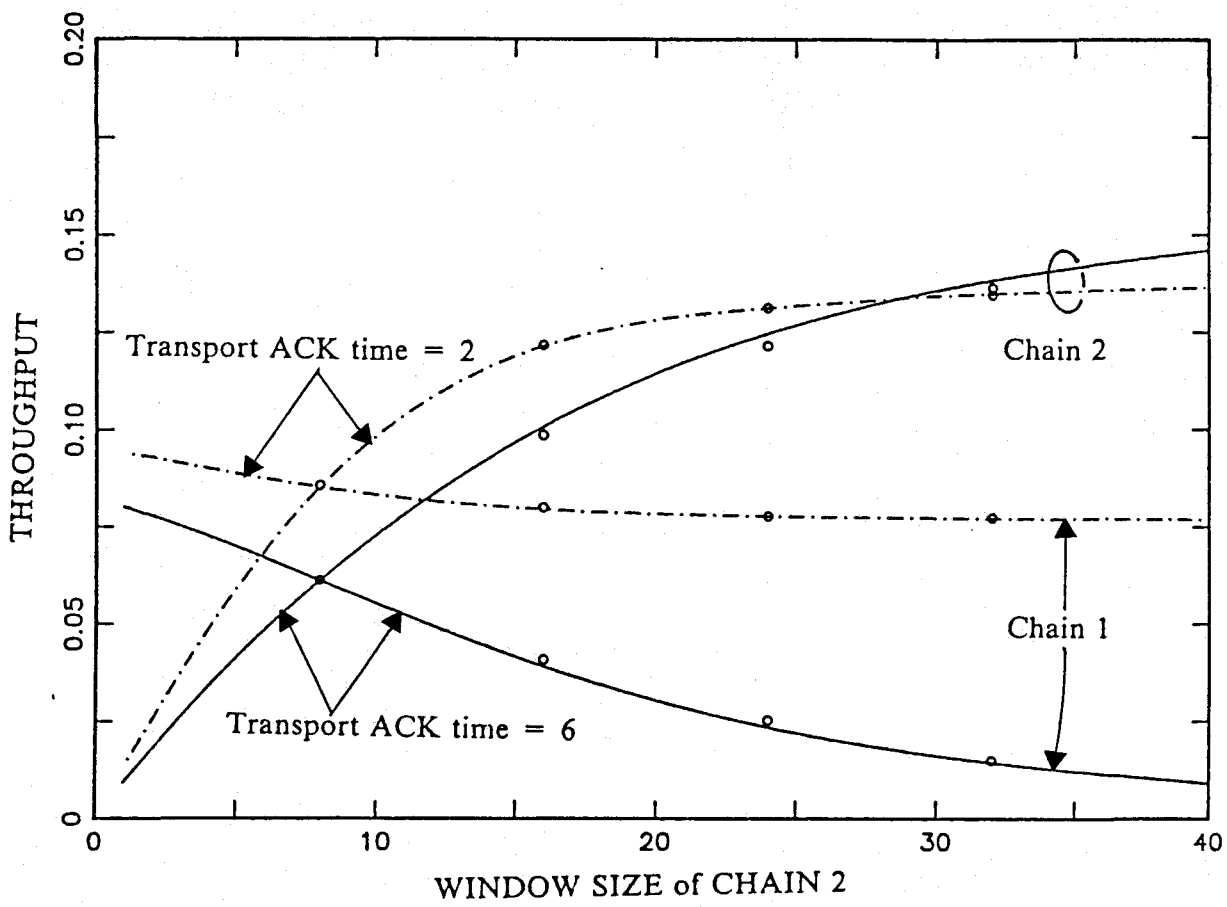


Figure 3.16-c. Throughput in the Case of a Full-Duplex with Priorities Model (The Window Sizes of Chain 1 are Variable)

throughput against the window sizes. To make it possible to compare performance among three models, we assume that the window sizes for chain 1 and chain 2 are identical. We further assume that the processing times for both data-messages and acknowledgements at Transport-queues are 6 *msec.* The processing times at the AP-queues are 50 *msec.* and 200 *msec.* Here, we define the network throughput as the total throughputs summed over all chains. In Figure 3.17-a, we see that the piggybacking scheme (Model 1) are more preferable than the explicit acknowledgements schemes (Models 3 and 5) when (i) the processing times at the AP-queues and/or (ii) the window size of the chain are small as described earlier. However, these small response times in the piggybacking scheme are obtained at the expense of low throughput as can be observed in Figure 3.17-b. In this numerical example, we cannot find any differences between explicit acknowledgement schemes without and with priority Transport-queues in both response times and network throughputs. It is because, in Model 5, while messages in chain 1 are processed with a low priority, the corresponding acknowledgements of chain 1 are processed prior to the data-messages in chain 2. Thus, the introduction of priority queues does not affect on system performance in the parameters range used in this numerical example.

3.5. Conclusion

In this chapter, we have built a two-layer performance model of LAN, which consists of a MAC layer submodel and Transport layer submodels. We have applied the polling model to the token passing MAC layer, and the closed queueing network model to the Transport layer. To deal with prioritized acknowledgement traffic, we have proposed a new MVA priority approximation. We have proposed iterative solution algorithms for the two-layer performance model, and compared system performance measures computed from our analysis to the simulation results. For almost all tried cases, our approximate analysis is shown to be within 90% confidence intervals in the mean message delay at each queue and within a few percent error in the throughput.

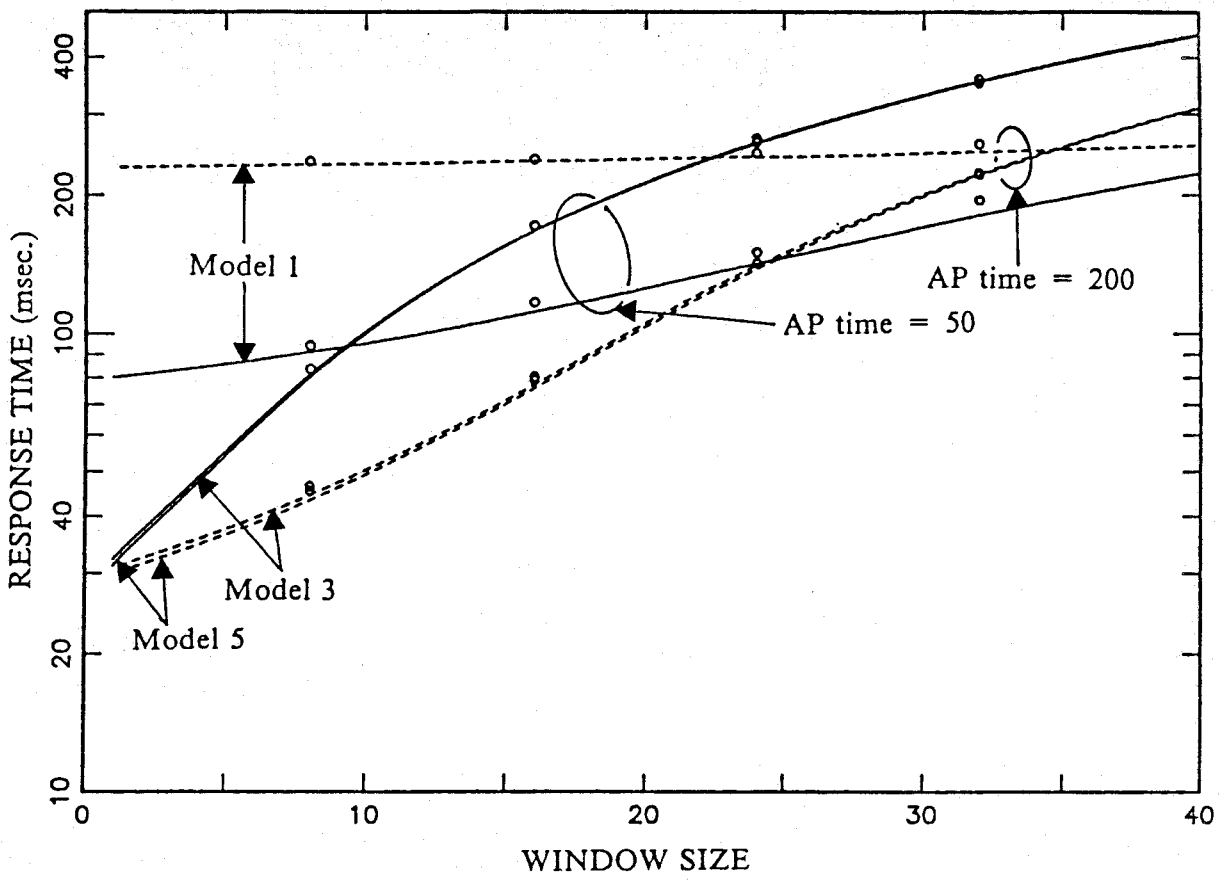


Figure 3.17-a. Response Time in Three Models 1, 3 and 5

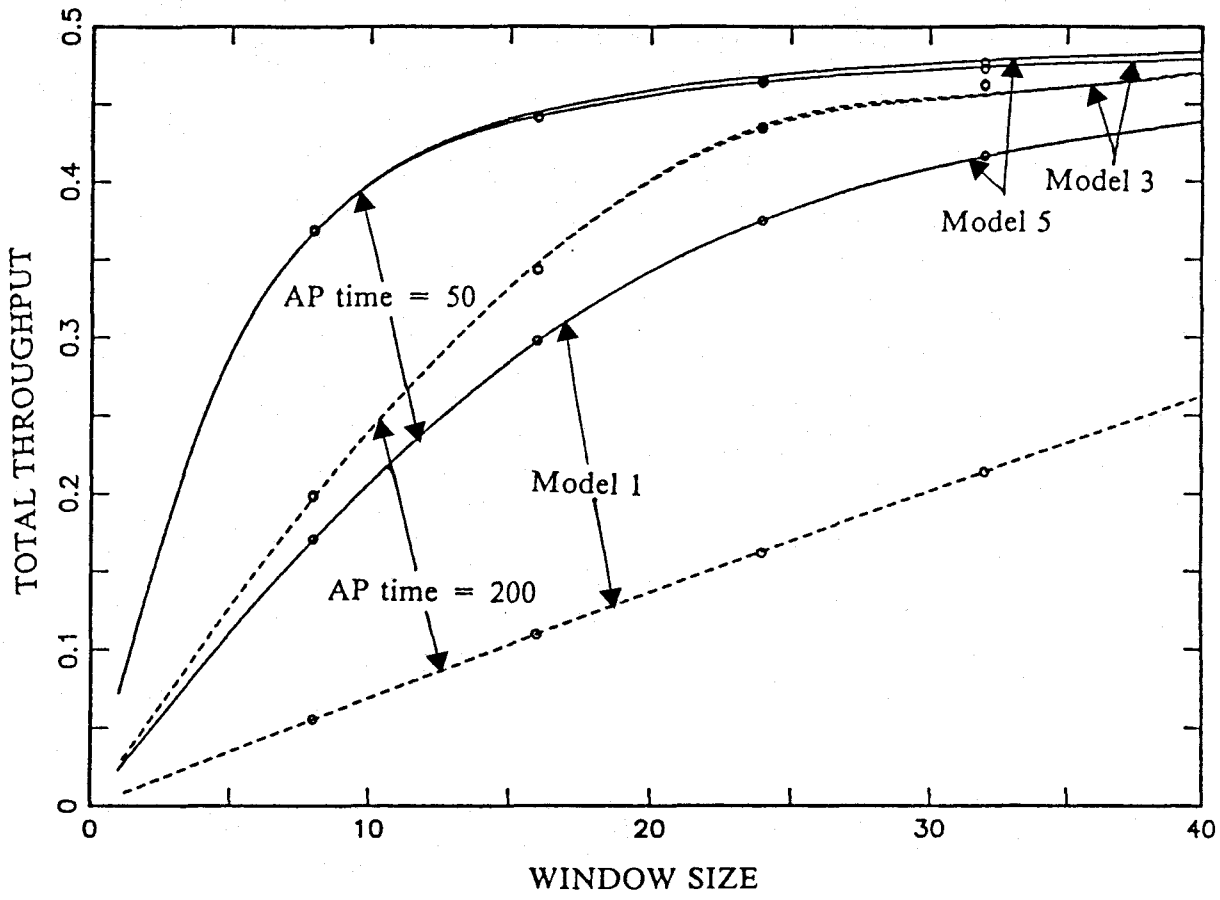


Figure 3.17-b. Network Throughput in Three Models 1, 3 and 5

Another potential application of our present approach is a flow control for interconnected LANs suggested in [Bux85] where a window size is controlled between source/destination stations via multiple interconnected LANs.

Chapter 4. Performance of Token Ring Networks with a Finite Capacity Bridge

4.1. Introduction

As pointed out in Chapter 1, there is a limit to the number of stations that can be attached to a single LAN and the distance the LAN can span. In order to increase the number of stations and/or to extend the distance covered by the network, local area networks should be interconnected by a *gateway* or *bridge*, which is a station with special functions such as routing and store-and-forward operation for internetwork messages. Performance-related aspects of this topic have been studied in [Nish86], [Bux85] and [Taki86]. Nishida *et al.* [Nish86] study the interconnection of CSMA/CD networks, while Bux and Grillo [Bux85] treat flow control for interconnected token ring networks. In [Taki86], Takine *et al.* only refer to the applicability of their result to the interconnected token ring networks. In this chapter, we will consider a system where two token ring networks are interconnected by a finite capacity bridge.

In an interconnected token ring network system where two kinds of messages (internetwork and intranetwork messages) coexist, two modes of operation are possible on each network [Bux85]. In the nonpriority-mode, the bridge and all other stations transmit at most one message when the token arrives. On the other hand, in the priority-mode, the bridge transmits internetwork messages to other stations by some priority operation. For that purpose, we have two modes of a priority operation. The one is a reservation priority operation described in Chapter 1. The other priority mode operation which we take

in this chapter is that the bridge transmits messages exhaustively when it captures the token while other stations transmit at most one message per token possession, which has been adopted by [Taki86]. While, in [Taki86], the authors assume that the bridge has an infinite capacity buffer, we consider the finite capacity buffer at the bridge. For the interconnected network model, the capacity of the buffer is a key issue for the design of a bridge because congestion of some network may produce a significant effect on other networks by message losses at the bridge. So, the performance prediction to determine the size of the bridge buffer is one of essential issues in interconnected network systems.

4.2. Token Ring Network with a Finite Capacity Bridge

4.2.1. Model Description

We first consider a single token ring network where N stations with one-message buffer and a station with a finite-buffer are connected. The last station with finite capacity is intended to represent a bridge in interconnected networks or a host computer in a single network. On the other hand, each station with a single-message buffer can have at most one outstanding message. A new message is generated only after a previous message has been completed service. Thus, such stations can be considered as terminals which are operated in an interactive mode. Below, we call the station with one-message buffer *terminal*, and the station with the finite buffer *bridge*.

A token ring network has been modelled by a polling system as described in Chapter 1. Below, we follow the terminology of polling systems for the analytical purpose. Terminal i ($1 \leq i \leq N$) takes a time exponentially distributed with mean $1 / \lambda_i$ to generate a new message only after a previous message has been completed service. The message transmission time is constant, b, being identical for all terminals. We also assume a constant walking time, r, which represents the propagation delay between two adjacent stations plus the bit latency at each station. Thus, the total walking time, R, equals to $(N + 1)r$. The

server visits the stations in cyclic order according to the station number. We consider the bridge as $N + 1$ st station. A *polling cycle* is defined as a time interval beginning at the polling instant of station 1 and ending with the completion of walking time from the bridge to station 1. If we had no bridge, the model described here would be a multi-queue, cyclic service model with a single-message buffer treated in several papers; see, for example, Chapter 2 of [Taka86a].

The bridge can have at most L messages, and those messages which arrive at the bridge to find the buffer full are lost. Messages in the buffer are served exhaustively when the server comes to the bridge. Further, we assume a Poisson arrival to the bridge at a rate λ_0, and a constant message transmission time, b_0. We model the bridge as an $M/G/1/L$ queue with independent vacation time and exhaustive service discipline, where L equals to the number of waiting places in the queue and the vacation time corresponds to the sum of the service times for terminals and walking times in our model. Such a model has been studied in [Cour80] and [Lee84]. We follow [Lee84] for the analysis of the bridge. Note that, in [Cour80] and [Lee84], the authors allow general distribution functions for the service times at terminals and the bridge. So, our assumption that the message transmission times are constant could be relaxed to general distributions. For our analysis of the combined network model, however, we pose this assumption for simplicity.

4.2.2. Analysis

We begin our analysis by defining the state of terminal i in the m th polling cycle, denoted by $u_i^{(m)}$, as

$$u_i^{(m)} = \begin{cases} 1 & \text{if the buffer at terminal } i \text{ is full} \\ 0 & \text{if the buffer at terminal } i \text{ is empty} \end{cases} \qquad i = 1, 2, ..., N, \quad m = 1, 2, ..., \infty . \tag{4.1}$$

Let $\mathbf{U}^{(m)} = (u_1{}^{(m)}, u_2{}^{(m)}, \ldots, u_N{}^{(m)})$ be the system state in the m th polling cycle. We define $P^{(m)}(\mathbf{U}^{(m)})$ as the probability that the server observes the sequence $\{u_i^{(m)}; i = 1, 2, \ldots, N\}$ for the states of the N terminals. We note that $\mathbf{U}^{(m+1)}$ only depends on $\mathbf{U}^{(m)}$ and the service time at the bridge in the m th polling cycle.

Furthermore, the service time at the bridge in the m th polling cycle only depends on $\mathbf{U}^{(m)}$ because the service to the bridge is performed exhaustively and no messages are left at the end of each service period. Therefore, the service time distribution at the bridge is determined by the length of the interval between the end of its previous service time and the beginning of the next service time, which is given by the sum of total walking times and the total service times at terminals. This time duration corresponds to the server's vacation time for the bridge. The Laplace-Stieltjes transform, $V_j^*(s)$, of this vacation time, when j messages have been transmitted from terminals in the current polling cycle, is given by

$$V_j^*(s) = e^{-(R+jb)s} \tag{4.2}$$

Let $F_j^*(s)$ be the Laplace-Stieltjes transform of the corresponding bridge busy period (server's sojourn time at the bridge), which can be obtained from the result in [Lee84] for the busy period of an $M/G/1/L$ queue with vacation model. Figure 4.1 illustrates an example of the m th polling cycle.

The process $\{\mathbf{U}^{(m)}\}$ is a Markov chain with 2^N distinct states in $[0, 1]^N$. The state transition probabilities of this Markov chain are now considered. Note that $u_i^{(m+1)}$ is 0 (no message found) if and only if there are no arrivals at terminal i after the server's visit to terminal i in the m th polling cycle. The probability of this event is given by

$$e^{-\lambda_i R_i^{(m)}} F_{|\mathbf{U}^{(m)}|}^*(\lambda_i) \tag{4.3}$$

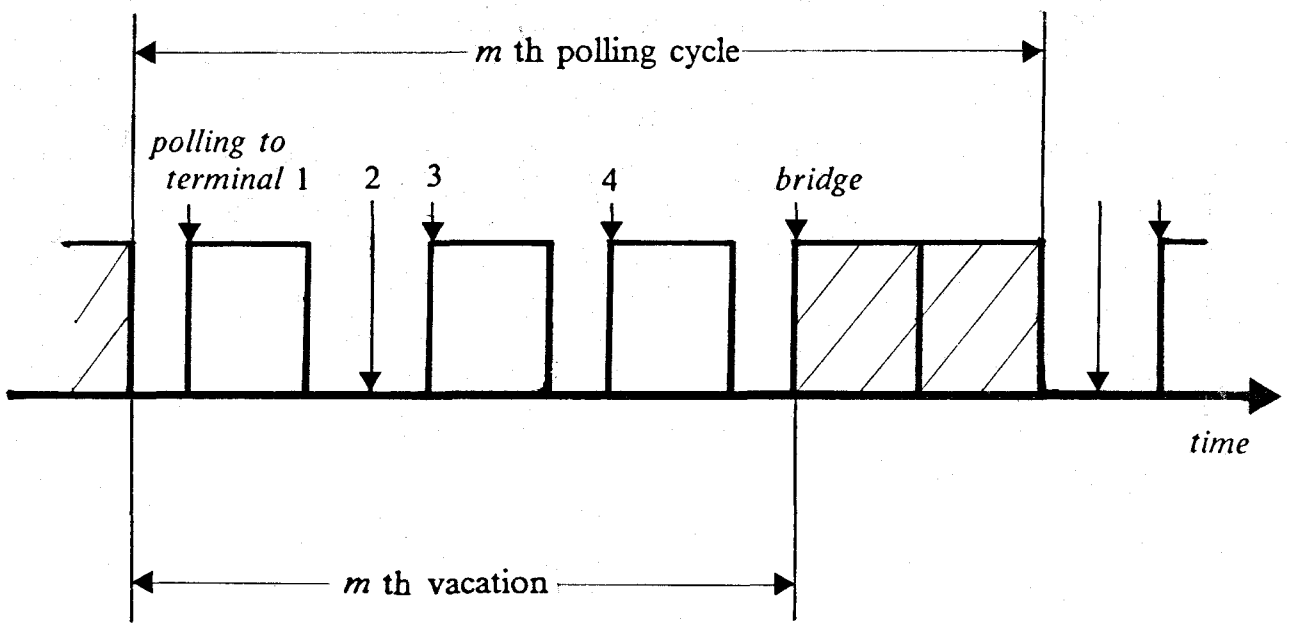
where

$$R_i^{(m)} \triangleq R + \left(\sum_{j=i+1}^{N} u_j^{(m)} + \sum_{j=1}^{i-1} u_j^{(m+1)}\right)b \tag{4.4}$$

and

$$|\mathbf{U}^{(m)}| \triangleq \sum_{j=1}^{N} u_j^{(m)}. \tag{4.5}$$

The factor $F_{|\mathbf{U}^{(m)}|}^*(\lambda_i)$ in (4.3) is the probability that no messages arrive at terminal i during the service time



The server sees the polling sequence $\mathbf{U}^{(m)} = \{1, 0, 1, 1\}$ for terminals and serves two messages for the bridge in the m th polling cycle in the case of $N = 4$ terminals.

Figure 4.1. An Example of m th Polling Cycle

at the bridge in the m th polling cycle. Since $u_i^{(m)}$ does not affect $u_i^{(m+1)}$, we have the relation for terminal i. Therefore, we have the state transition probabilities:

$$\text{Prob}\,[(u_1^{(m+1)}, \ldots, u_{i-1}^{(m+1)}, 1, u_{i+1}^{(m+1)}, \ldots, u_N^{(m+1)}) \mid (u_1^{(m)}, \ldots, u_{i-1}^{(m)}, u_i^{(m)}, u_{i+1}^{(m)}, \ldots, u_N^{(m)})]$$

$$= \left\{1 - e^{-\lambda_i R_i^{(m)}} F_{|\mathbf{U}^{(m)}|}{}^*(\lambda_i)\right\} \tag{4.6a}$$

$$\text{Prob}\,[(u_1^{(m+1)}, \ldots, u_{i-1}{}^{(m+1)}, 0, u_{i+1}^{(m+1)}, \ldots, u_N^{(m+1)}) \mid (u_1^{(m)}, \ldots, u_{i-1}^{(m)}, u_i^{(m)}, u_{i+1}^{(m)}, \ldots, u_N^{(m)})]$$

$$= e^{-\lambda_i R_i^{(m)}} F_{|\mathbf{U}^{(m)}|}{}^*(\lambda_i) \tag{4.6b}$$

$$u_j^{(m+1)} \in [0, 1] \quad j = 1, \ldots, i-1, i+1, \ldots, N; \qquad u_j^{(m)} \in [0, 1], \quad j = 1, \ldots, N$$

which are reduced to

$$\text{Prob}\,[\mathbf{U}^{(m+1)} \mid \mathbf{U}^{(m)}]$$

$$= \prod_{i=1}^{N} \left\{(1 - u_i^{(m+1)})\, e^{-\lambda_i R_i^{(m)}} F_{|\mathbf{U}^{(m)}|}{}^*(\lambda_i) + u_i^{(m+1)}\left(1 - e^{-\lambda_i R_i^{(m)}} F_{|\mathbf{U}^{(m)}|}{}^*(\lambda_i)\right)\right\}. \tag{4.7}$$

We can obtain the limiting probabiltiy

$$P(\mathbf{U}) \triangleq \lim_{m \to \infty} P^{(m)}(\mathbf{U}^{(m)}) \tag{4.8}$$

where

$$\mathbf{U} = (u_1, \ldots, u_N) \in [0, 1]^N. \tag{4.9}$$

After taking the limit $m \to \infty$, the stationary probabilities are given by (4.7) if we utilize the normalization condition:

$$\sum_{\mathbf{U} \in [0,1]^N} P(\mathbf{U}) = 1. \tag{4.10}$$

4.2.3. Performance Measures

Now we derive performance measures for both terminals and the bridge. Let us introduce

$$\pi_j \triangleq \sum_{\text{all U such that } |\mathbf{U}| = j} P(\mathbf{U}), \quad j = 0, \ldots, N \tag{4.11}$$

which represents the probability that j messages are found in the polling cycle. Thus, from (4.11), we obtain the mean polling cycle time, $E[C]$, as

$$E[C] = R + \sum_{j=0}^{N} \pi_j [jb - F_j^{*(1)}(0)] \tag{4.12}$$

where $F_j^{*(1)}$ is the first derivative of F_j^* and, therefore, $-F_j^{*(1)}(0)$ is the mean busy period of the bridge given j.

We obtain the throughput, γ_i, and the mean waiting time (including the message transmission time), $E[W_i]$, for terminal i following [Taki86]. The probability α_i that a message to transmit is found at terminal i when polled is given by

$$\alpha_i = \sum_{\text{all U such that } u_i = 1} P(\mathbf{U}) . \tag{4.13}$$

Thus, the throughput for terminal i, which is defined as the ratio of the mean message transmission time of terminal i to the mean polling cycle time, is obtained by

$$\gamma_i = \frac{\alpha_i \, b}{E[C]} . \tag{4.14}$$

We now derive the mean message waiting time for terminal i. Note that the buffer state of terminal i alternates between the *empty* state of the mean duration of $1/\lambda_i$, and the *full* state of mean duration, $E[W_i]$, which corresponds to the mean message waiting time. Therefore, the throughput of terminal i can be also defined as

$$\gamma_i = \frac{b}{E[W_i] + 1/\lambda_i} . \tag{4.15}$$

The mean message waiting time for terminal i is given by

$$E[W_i] = \frac{E[C]}{\alpha_i} - \frac{1}{\lambda_i} . \tag{4.16}$$

To apply the results obtained in [Lee84] to our analysis of the bridge, we need the Laplace-Stieltjes transforms of the message transmission time, $S^*(s)$, and the vacation time, $V^*(s)$. We assume that they are given by

$$S^*(s) = e^{-b_0 s} \tag{4.17}$$

and

$$V^*(s) = \sum_{j=0}^{N} \pi_j V_j^*(s) = \sum_{j=0}^{N} \pi_j e^{-(R+jb)s} . \tag{4.18}$$

The blocking probability, P_b, the mean message waiting time, $E[W_0]$, and the throughput, γ_0, for the bridge are then obtained using (4.17) and (4.18). Means of the service time and the vacation time are noted by $E[S]$ and $E[V]$, respectively.

The blocking probability, P_b, the mean message waiting time, $E[W_0]$, and the throughput, γ_0, for the bridge are given by the following equations:

$$P_b = \frac{\rho'}{\lambda_0 E[S]} \left[\lambda_0 E[S] - 1 + \frac{\sum_{j=1}^{L} jq_j}{1 - (p_0 + q_0)} \right] + \frac{1 - \rho'}{\lambda_0 E[V]} \left[\lambda_0 E[V] - \frac{\sum_{j=0}^{L} jq_j}{p_0 + q_0} \right], \tag{4.19}$$

$$E[W_0] = \frac{\rho}{\rho'} \left[\frac{E[V]\rho' + E[S](1 - \rho')}{\lambda_0^2 E[S] E[V]} \sum_{j=1}^{L-1} jp_j + \frac{L}{\lambda_0}(1 - \frac{\rho'}{\rho}) \right] \tag{4.20}$$

and

$$\gamma_0 = \lambda_0 b_0 (1 - P_b) \tag{4.21}$$

where ρ is traffic intensity and ρ′ is effective traffic intensity, which are given by

$$\rho = \lambda_0 b_0 , \quad \rho' = \frac{[1 - (p_0 + q_0)]\, E[S]}{(p_0 + q_0)\, E[V] + [1 - (p_0 + q_0)]\, E[S]} . \tag{4.22}$$

And, p_j and q_j are obtained by solving the following equations:

$$p_n = \sum_{k=1}^{n+1} g_{n-k+1}(p_k + q_k), \quad n = 0, \ldots, L-2 , \tag{4.23a}$$

$$p_{L-1} = \sum_{k=1}^{L-1} g^c_{n-k+1}(p_k + q_k) + q_L , \tag{4.23b}$$

$$q_n = h_n(p_0 + q_0), \quad n = 0, \ldots, L-1 , \tag{4.23c}$$

$$q_L = h^c_L(p_0 + q_0) \tag{4.23d}$$

and

$$\sum_{n=0}^{L-1} p_n + \sum_{n=0}^{L} q_n = 1 . \tag{4.23e}$$

where g_i and h_i are the probabilities that i messages arrive during the service time and the vacation time, respectively. If $S^{*(i)}$ and $V^{*(i)}$ denote the i th derivatives of $S^*(\theta)$ and $V^*(\theta)$, then

$$g_i = \left[\frac{(-\lambda_0)^i}{i!}\right] S^{*(i)}(\lambda_0) , \tag{4.24a}$$

$$h_i = \left[\frac{(-\lambda_0)^i}{i!}\right] V^{*(i)}(\lambda_0) \tag{4.24b}$$

and

$$g_k^c = \sum_{i=k}^{\infty} g_i \,, \qquad h_k^c = \sum_{i=k}^{\infty} h_i \,. \tag{4.25}$$

4.2.4. Numerical Examples

For numerical examples, we consider a token ring network with $N = 8$ terminals and a bridge. The message transmission times for both terminals and the bridge are set to 1, i.e., $b = b_0 = 1$. The walking time r is assumed to be 0.1. The arrival rates to terminals are of the same value: $\lambda_i = 0.0875$ $(i = 1, \ldots, N)$.

Figures 4.2.a-c show mean message waiting times, throughput and loss probability as functions of the arrival rate to the bridge for three buffer sizes; $L = 5, 10, 20$. In Figure 4.2-a, we show the averages of mean message waiting times over all terminals, and we illustrate the total throughput of all terminals in Figure 4.2-b. Figure 4.2-c indicates that the buffer length of 20 is sufficient to keep the loss probability below 0.01 for given parameters.

Another interesting point in this example is the difference in performance measures among terminals because we may imagine that the difference becomes remarkable as the arrival rate to the bridge becomes large. In Table 4.1, we show the throughput and the mean message waiting times for each terminal for three values of arrival rate at the bridge; $\lambda_0 = 0.1, 0.5$ and 0.9. We observe that the throughput and the mean message waiting times are almost uniformly distributed among the terminals. However, looking more carefully, we notice that the terminals are preferred in the order of indexes, i.e., the closer to the bridge in the downstream direction, the more throughput and less waiting times are granted.

4.3. Simplified Solution for a Large Number of Terminals

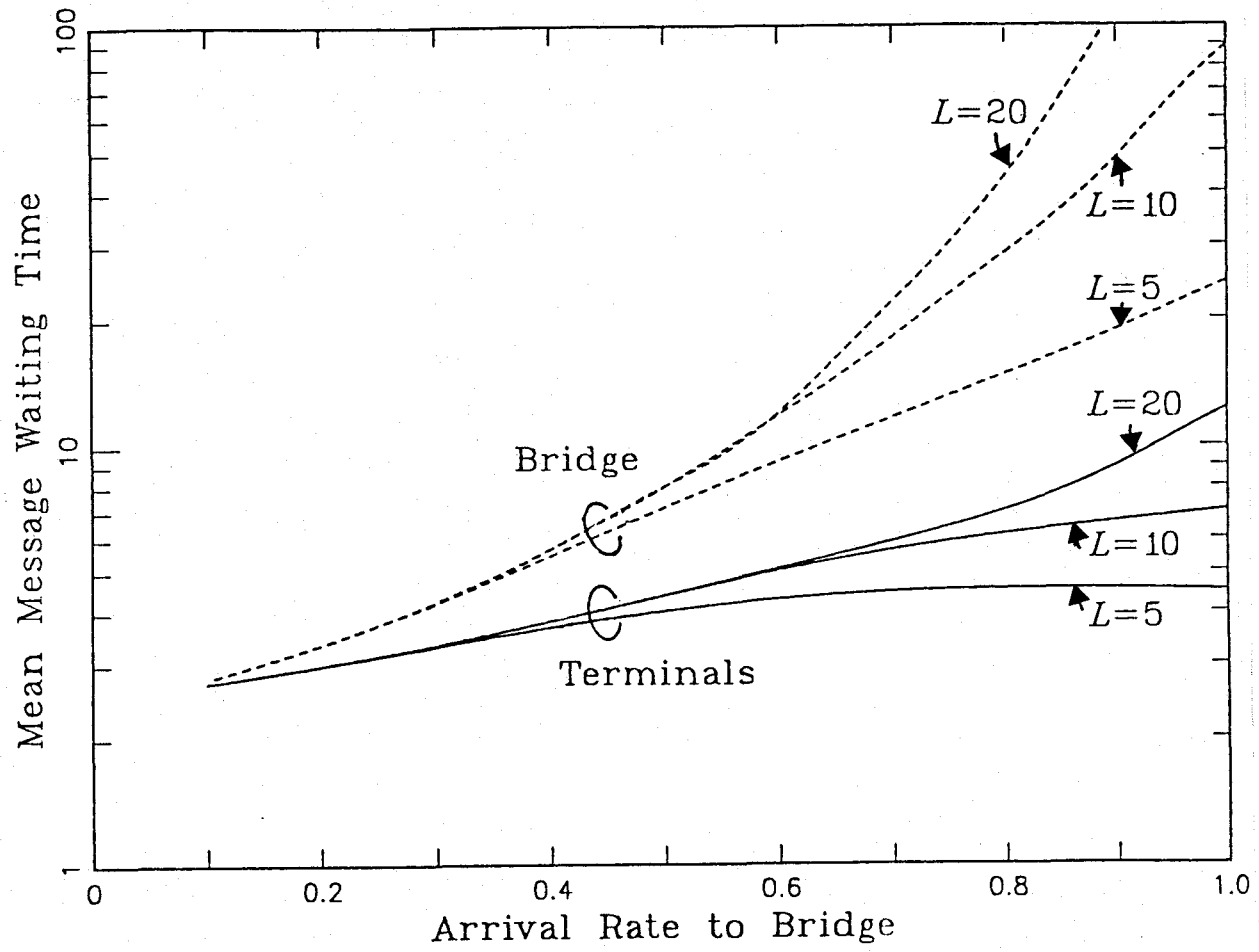


Figure 4.2-a. Mean Message Waiting Times (*L*: Buffer Size of the Bridge)

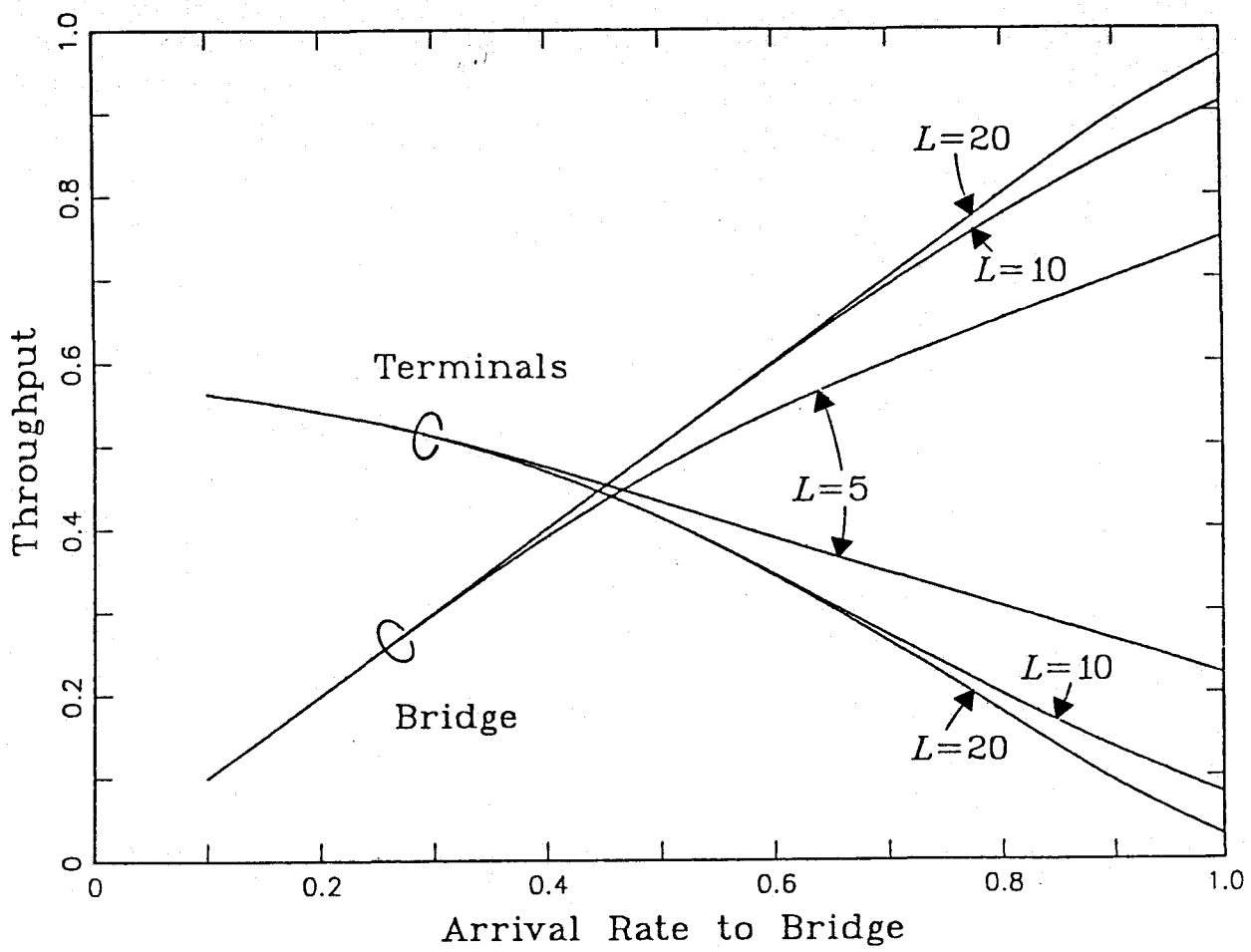


Figure 4.2-b. Throughput (*L*: Buffer Size of the Bridge)

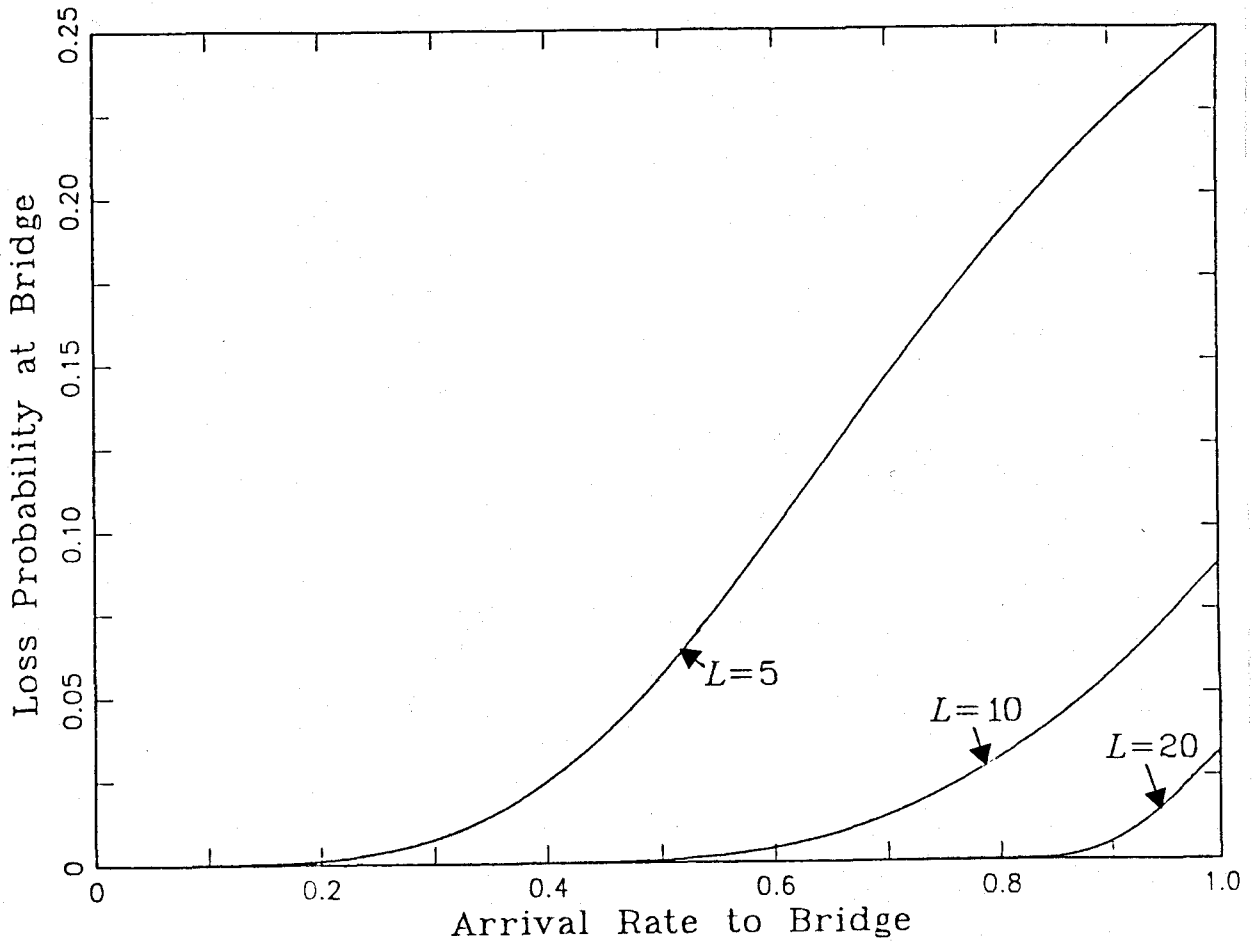


Figure 4.2-c. Loss Probability at Bridge (*L*: Buffer Size of the Bridge)

Table.4.1. Comparison of Mean Message Waiting Times and Throughput among Terminals

		$\lambda_0 = 0.1$	$\lambda_0 = 0.5$	$\lambda_0 = 0.9$
γ_i	terminal 1	0.07015	0.05368	0.03317
	terminal 2	0.07015	0.05362	0.03313
	terminal 3	0.07014	0.05357	0.03309
	terminal 4	0.07013	0.05351	0.03305
	terminal 5	0.07012	0.05344	0.03301
	terminal 6	0.07011	0.05336	0.03296
	terminal 7	0.07010	0.05328	0.03291
	terminal 8	0.07009	0.05319	0.03286
$E[W_i]$	terminal 1	2.82592	7.20143	18.72366
	terminal 2	2.82708	7.21954	18.75738
	terminal 3	2.82842	7.23929	18.79279
	terminal 4	2.82995	7.26087	18.83001
	terminal 5	2.83173	7.28455	18.86923
	terminal 6	2.83380	7.31059	18.91060
	terminal 7	2.83622	7.33936	18.95435
	terminal 8	2.83907	7.37124	19.00071

4.3.1. Recursive Algorithm

In the previous section, we have presented an analysis for a token ring network where N stations with single-message buffer and a bridge with finite capacity are interconnected. However, the analysis requires 2^N states to completely specify the system, which introduces excessive computational time and space even when N is not very large, e.g., $N = 10$. In this section, we present a simplified solution for the same model. For this purpose, we define the system state as the total number of messages transmitted from the terminals in a polling cycle. This definition reduces the number of possible states to $N + 1$. For this purpose, we consider a symmetrical case such that arrival rates to terminals are identical for all terminals, denoted by λ. Furthermore we assume that i messages transmitted in the polling cycle are distributed uniformly over N terminals. We will find this assumption is reasonable from numerical results provided shortly.

To start with, we introduce the following random variables;

$Q^{(m)}$: the number of messages transmitted in m th polling cycle.

$\underline{Q}_j^{(m)}$: the number of messages transmitted before terminal j in m th polling cycle.

$\overline{Q}_j^{(m)}$: the number of messages transmitted after terminal j in m th polling cycle

The state transition probabilities are represented by

$$p_{ij} = \text{Prob}\,[Q^{(m+1)} = j \mid Q^{(m)} = i], \quad 0 \le i,j \le N \tag{4.26}$$

Using $\underline{Q}_j^{(m)}$ and $\overline{Q}_j^{(m)}$, the transition probabilities, p_{ij}, may be written as

$$p_{ij} = \frac{\binom{N-1}{i}}{\binom{N}{i}} \sum_{r'=0}^{1} \text{Prob}\,[\underline{Q}_N^{(m+1)} = j - r', u_N^{(m+1)} = r' \mid \underline{Q}_N^{(m)} = i, u_N^{(m)} = 0]$$

$$+\frac{\binom{N-1}{i-1}}{\binom{N}{i}}\sum_{r'=0}^{1}\text{Prob}\,[\underline{Q}_N^{(m+1)}=j-r',u_N^{(m+1)}=r'\mid \underline{Q}_N^{(m)}=i-1,u_N^{(m)}=1]\,. \tag{4.27}$$

This has resulted from our assumption that i transmitting messages are uniformly distributed among N terminals. Namely, given that i messages are transmitted in the m th polling cycle, the probability that N th terminal transmits a message is given by $\binom{N-1}{i-1}/\binom{N}{i}$, and the probability that N th terminal does not transmit a message is given by $\binom{N-1}{i}/\binom{N}{i}$. We note that

$$\binom{l}{m}=0, \quad \text{if } l<m \text{ or } m<0\,. \tag{4.28}$$

In Appendix B, we show that the state transition probabilities can be computed recursively starting with (4.27).

In the simplified solution, the stationary probabilities, π_j^a, can be obtained by using $[p_{ij}]$. The mean polling cycle time, $E[C^a]$, is obtained similarly as in Section 4.2.3, i.e.,

$$E[C^a]=R+\sum_{j=0}^{N}\pi_j^a\,[jb-F_j^{*(1)}(0)]\,. \tag{4.29}$$

We obtain the probability, α^a, that a message is found at a terminal in the polling cycle as

$$\alpha^a=\sum_{j=0}^{N}\frac{\binom{N-1}{j-1}}{\binom{N}{j}}\pi_j^a=\sum_{j=0}^{N}\frac{j}{N}\pi_j^a\,. \tag{4.30}$$

Then, the throughput, γ^a, and the mean message waiting time, $E[W^a]$, for each terminal are obtained by (4.14) and (4.16).

For the bridge, we can derive γ_0^a and $E[W_0^a]$ by using the approximate vacation time given by

$$V^{a^*}(s) = \sum_{j=0}^{N} \pi_j^a \, e^{-(R+jb)s}. \tag{4.31}$$

We note that this simplified method can also be applied to the model with an infinite capacity bridge in [Taki86].

4.3.2. Validation

For validation of our simplified method, we consider the following numerical example:

message transmission time for both terminals and bridge: $b = b_0 = 1$

buffer length of bridge: $L = 5$

We first compare our method to the analysis presented in the previous section. Table 4.2 shows comparative results in the case where $N = 8, \lambda_1 = ... = \lambda_N = \lambda = 0.0875$ and $\lambda_0 = 0.1, 0.5,$ and 0.9. Another numerical example is shown in Table III. In Table 4.3, arrival rates to terminals ($\lambda_1 = ... = \lambda_N = \lambda$) are changed while the arrival rate to the bridge is fixed at 0.3. We also illustrate the comparison in the case where the walking time is changed: $r = 0.001, 0.01$ and 0.1 in Table 4.4. Here, we set $\lambda_1 = ... = \lambda_N = \lambda = 0.0875$ and $\lambda_0 = 0.3$. We observe in these tables that results are in reasonable agreement in overall range of parameters we have tried.

Next, we compare our approximate results to the simulation results in order to assess the accuracy of our algorithm when N is large. Computer simulations have been performed by IBM RESQ2. In Figures 4.3.a-c, we present the mean message waiting times, throughput and loss probability dependent on the number of terminals for three values of the total arrival rate to the terminals; $\lambda N = 0.2, 0.5$ and 0.8. We assume that R is constant, 1.0. The simulation results are depicted with 95 percent confidence intervals. Here we have excellent agreement between our computation and simulation.

Table.4.2. Comparison of Simplified Results and Detailed Results (λ_0 is Variable)

			$\lambda_0 = 0.1$	$\lambda_0 = 0.5$	$\lambda_0 = 0.9$
terminals	γ_i	$i = 1$	0.07015	0.05368	0.03317
		:	:	:	:
		$i = 8$	0.07009	0.05319	0.03286
	γ^a		0.07031	0.05391	0.03308
	$E[W_i]$	$i = 1$	2.82592	7.20143	18.72366
		:	:	:	:
		$i = 8$	2.83907	7.37124	19.00071
	$E[W^a]$		2.79379	7.12023	18.80223
bridge	γ_0		0.10000	0.47040	0.69916
	γ_0^a		0.10000	0.47210	0.69926
	$E[W_0]$		2.74392	4.06377	4.52787
	$E[W_0^a]$		2.72714	4.02680	4.53369
	P_b		0.00005	0.05920	0.22316
	P_b^a		0.00005	0.05581	0.22304

Table.4.3. Comparison of Simplified Results and Detailed Results λ is Variable)

			λ = 0.0125	λ = 0.0625	λ = 0.1125
terminals	γ_i	$i = 1$	0.01217	0.05149	0.07061
		:	:	:	:
		$i = 8$	0.01217	0.05132	0.07020
	γ^a		0.01218	0.05179	0.07096
	$E[W_i]$	$i = 1$	2.14432	3.41996	5.27249
		:	:	:	:
		$i = 8$	2.15974	3.48459	5.35581
	$E[W^a]$		2.12134	3.30982	5.20365
bridge	γ_0		0.29986	0.29883	0.29596
	γ_0^a		0.29987	0.29899	0.29621
	$E[W_0]$		1.81322	2.76166	3.94445
	$E[W_0^a]$		1.80963	2.70980	3.91138
	P_b		0.00045	0.00391	0.01345
	P_b^a		0.00044	0.00337	0.01262

Table.4.4. Comparison of Simplified Results and Detailed Results (r is Variable)

			$r = 0.001$	$r = 0.01$	$r = 0.1$
terminals	γ_i	$i = 1$	0.07012	0.06941	0.06348
		:	:	:	:
		$i = 8$	0.06971	0.06941	0.06316
	γ^a		0.07116	0.07036	0.06391
	$E[W_i]$	$i = 1$	2.83183	2.97812	4.32495
		:	:	:	:
		$i = 8$	2.91716	3.06305	4.40367
	$E[W^a]$		2.62394	2.78412	4.21839
bridge	γ_0		0.29912	0.29901	0.29756
	γ_0^a		0.29934	0.29924	0.29781
	$E[W_0]$		2.39908	2.49857	3.37461
	$E[W_0^a]$		2.28015	2.38929	3.32259
	P_b		0.00294	0.00330	0.00814
	P_b^a		0.00259	0.00295	0.00731

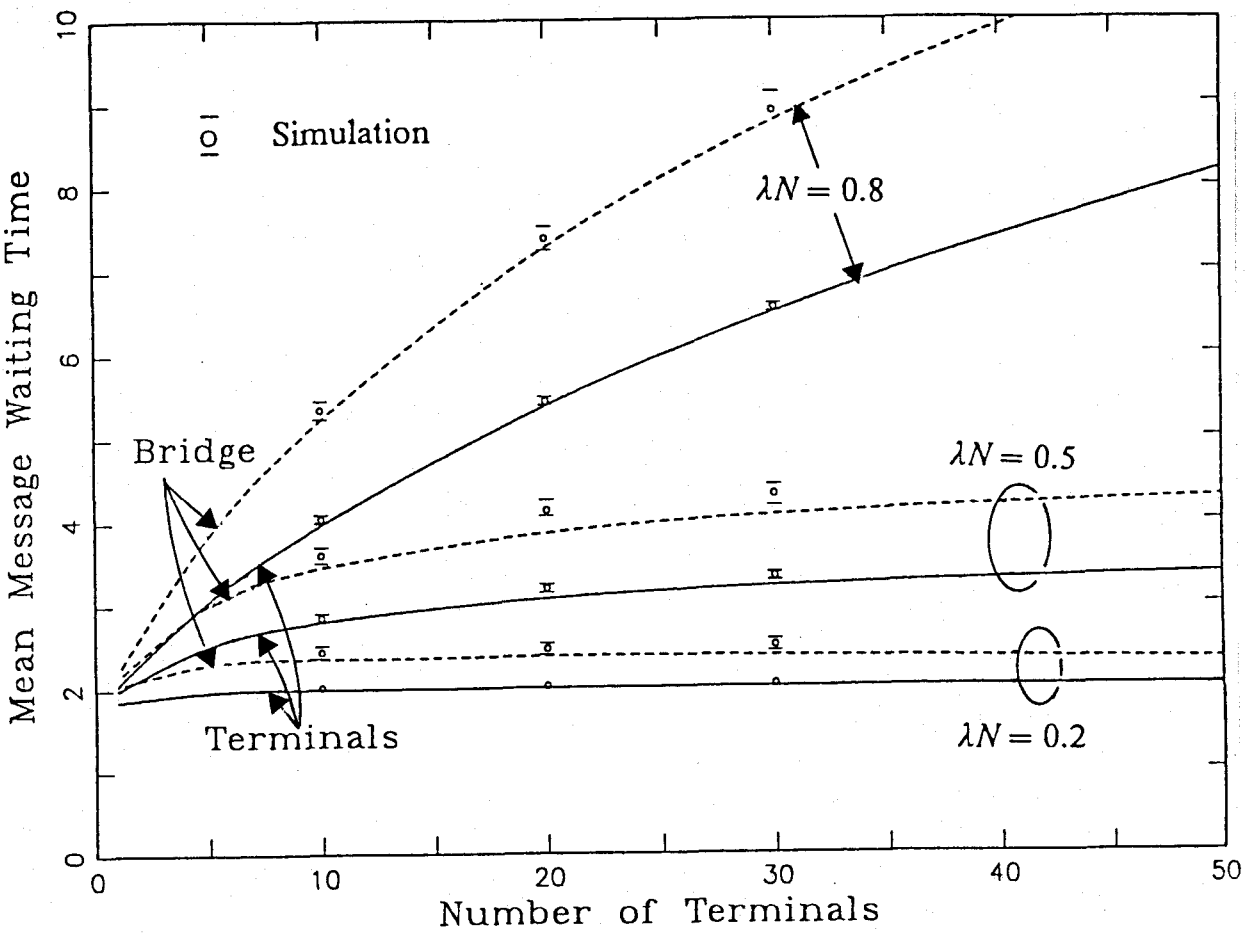


Figure 4.3-a. Comparison of Mean Message Waiting Times between Analysis and Simulation

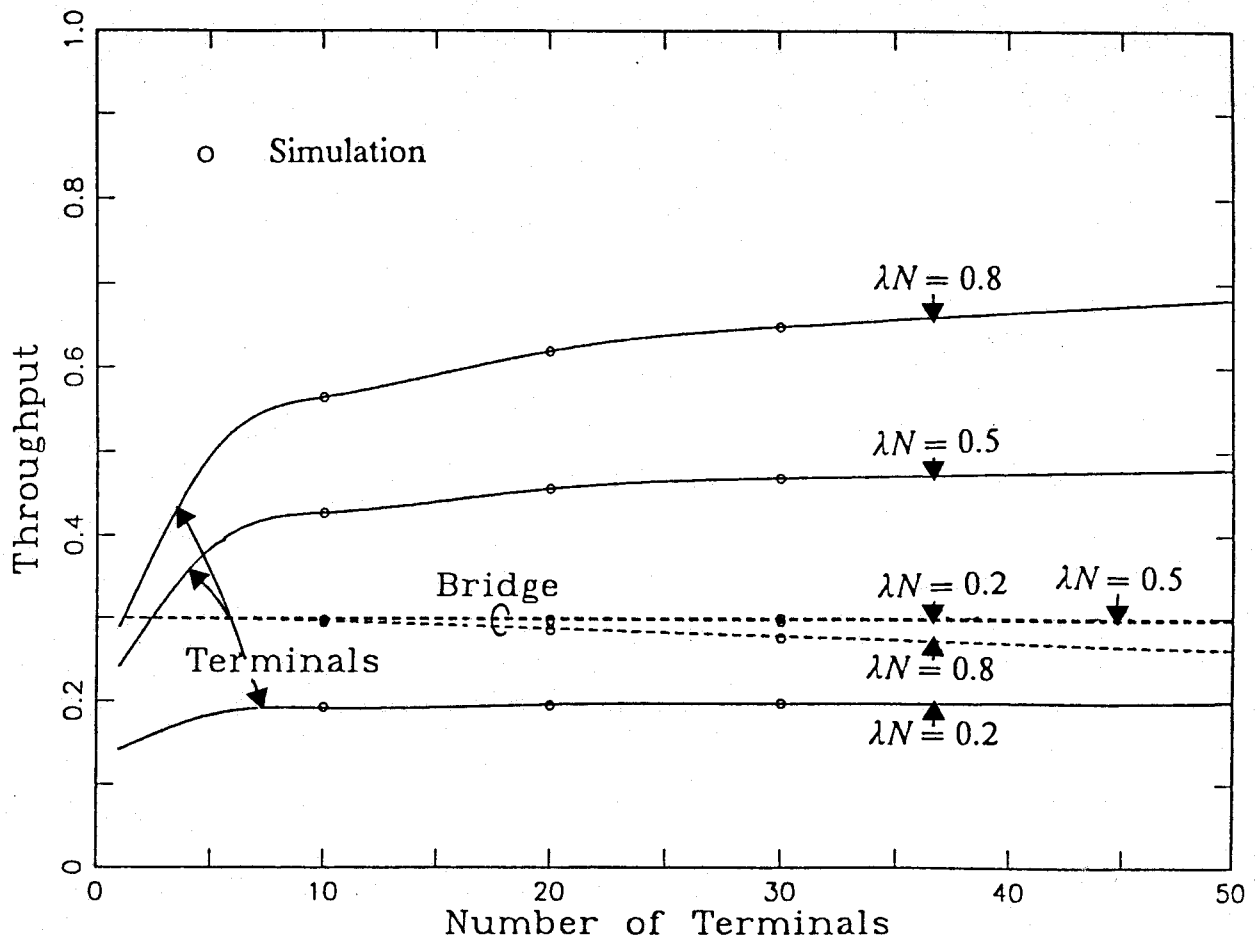


Figure 4.3-b. Comparison of Throughput between Analysis and Simulation

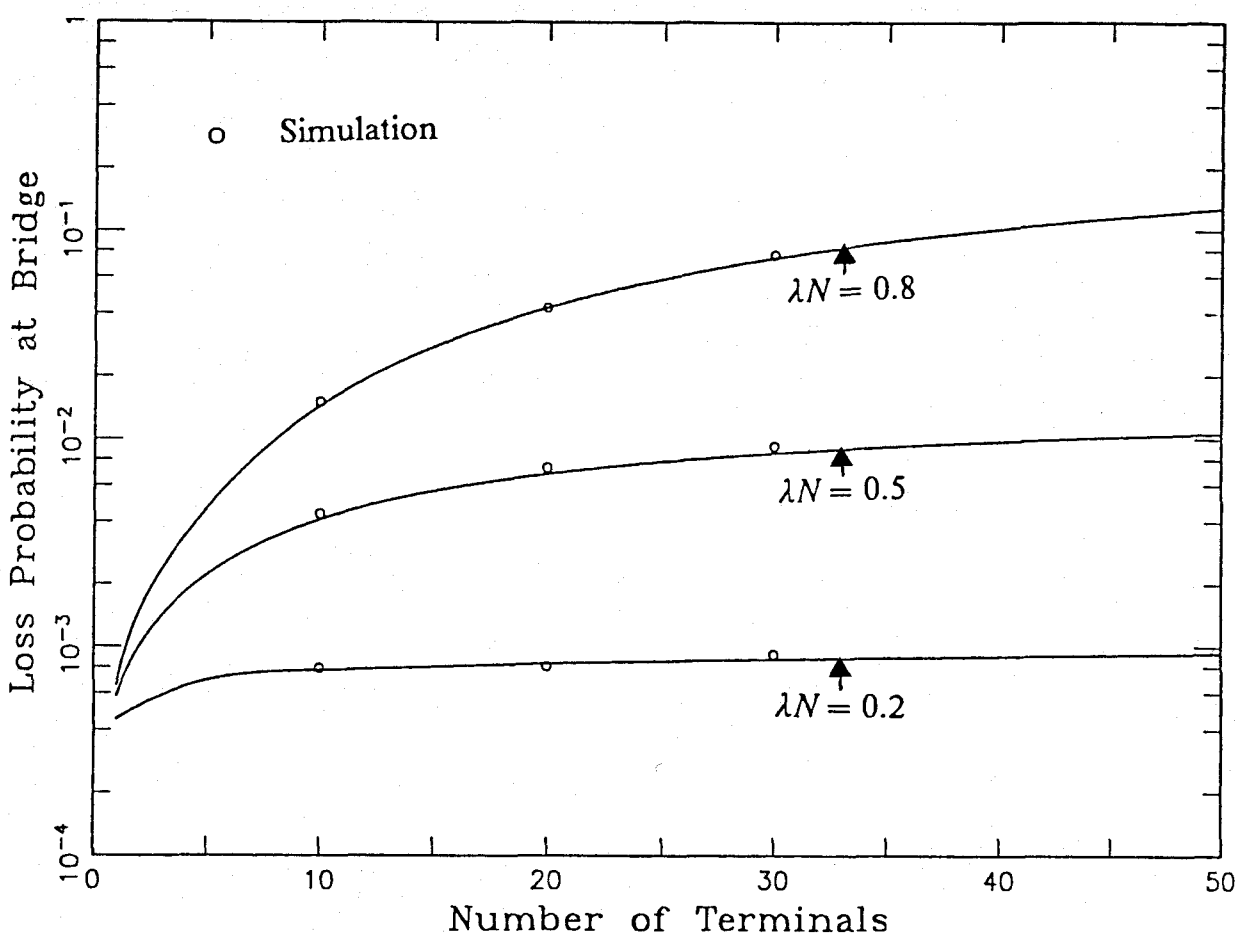


Figure 4.3-c. Comparison of Loss Probability at Bridge between Analysis and Simulation

4.4. Application to Interconnected Token Ring Networks

4.4.1. Model of the Interconnected Network

In this section, we consider an interconnected token ring network system where a token ring network is connected to another network by a bridge. The bridge has two finite-length buffers: transmit-buffer (B_1) for messages from another network, and receive-buffer (B_2) for messages destined for another network. Lengths of each buffer are L_1 and L_2, respectively (Figure 4.4).

Network operation for the interconnected network is performed as follows: when a terminal has a message to transmit destined for another network, it waits for a token. After capturing the token, the terminal transmits the message to the bridge. If the bridge buffer (B_2) is full, the message is not copied at the bridge and such an indication is returned to the terminal. (We can perform this operation by using the indicator, C indicator, which shows that the message has been copied by the destination station.

For the analysis of an interconnected network system, we decompose it into subnetworks where each subnetwork consists of single-buffer terminals and a bridge with a finite-length transmit-buffer. Each subnetwork can be analyzed by solution methods described in Section 4.2 for the model with a small number of terminals, or in Section 4.3 for the model with a middle/large number of terminals. The subnetworks are aggregated after analyzing separately. The offered traffic rate to the bridge is set equal to the sum of throughputs from other subnetworks. This approach implies an assumption that messages arrive at the bridge according to a Poisson process while, in reality, the arrival process to the bridge is more complicated.

Each terminal transmits messages destined for other terminals in the same network (*internal messages*) or destined for other networks via the bridge (*external messages*). We assume that when the terminal has a message to transmit, the destination of the message is determined independently at each transmission.

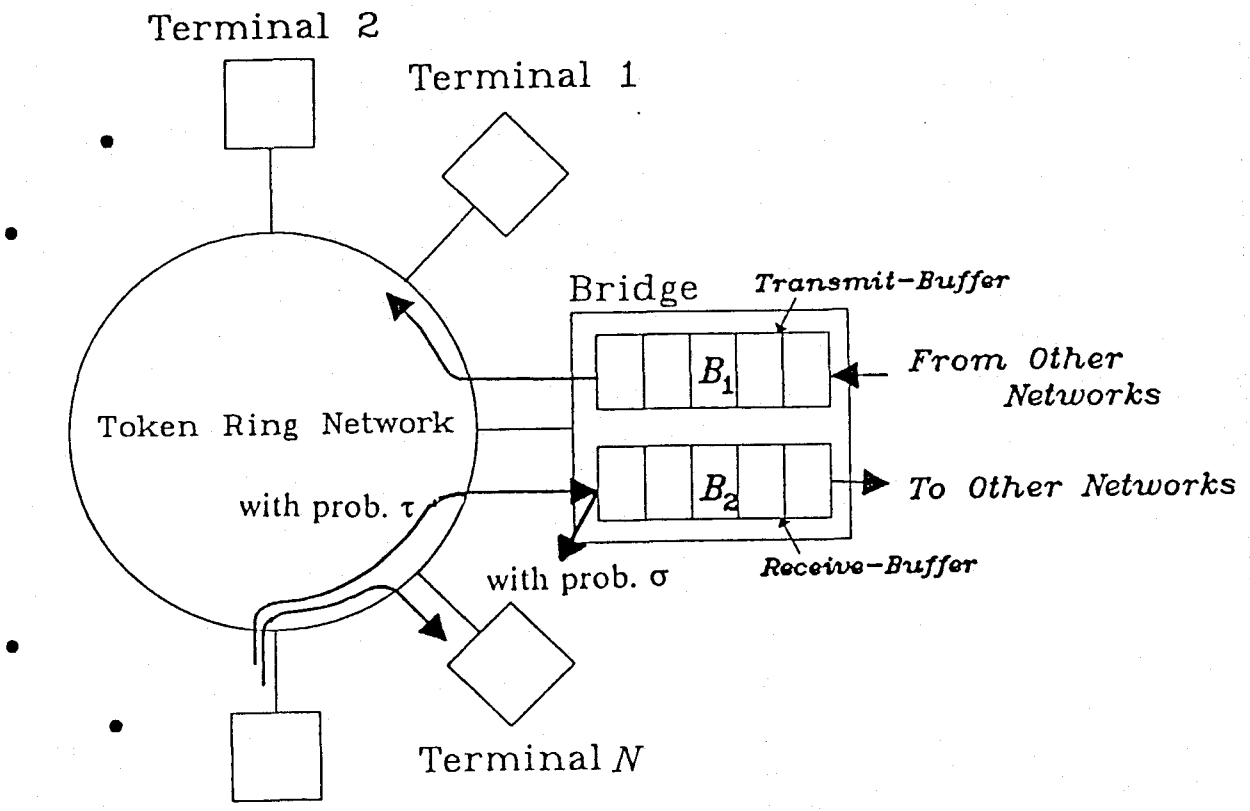


Figure 4.4. A Subnetwork Model for a Token Ring Network with a Bridge

Furthermore, the probability that the bridge buffer is full and the message is lost is assumed to be constant. We introduce the following two parameters:

- τ: the probability that a terminal generates the external message.
- σ: the probability that when the terminal transmits an external message, the receive-buffer at the bridge is full and the message is lost.

Using these parameters, the transition probabilities for terminal i are given as follows:

$$
\begin{aligned}
&u_i^{(m)} = 0 \rightarrow u_i^{(m+1)} = 1 \quad \text{with Prob.} \quad 1 - e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i) \\
&u_i^{(m)} = 0 \rightarrow u_i^{(m+1)} = 0 \quad \text{with Prob.} \quad e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i) \\
&u_i^{(m)} = 1 \rightarrow u_i^{(m+1)} = 1 \quad \text{with Prob.} \quad \tau\left[\sigma + (1-\sigma)\left(1 - e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i)\right)\right] \\
&\qquad\qquad\qquad\qquad + (1-\tau)\left(1 - e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i)\right) \\
&u_i^{(m)} = 1 \rightarrow u_i^{(m+1)} = 0 \quad \text{with Prob.} \quad \tau(1-\sigma) e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i) + (1-\tau) e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i)
\end{aligned}
\tag{4.32}
$$

where $R_i^{(m)}$ and $F_j^*(s)$ are defined in Section 4.2.2. Thus, the system state transition probabilities are now given as follows:

$$
\begin{aligned}
\text{Prob}\,[\mathbf{U}^{(m+1)} \mid \mathbf{U}^{(m)}] = \prod_{i=1}^{N} \Bigg\{ & (1 - u_i^{(m+1)}) \Big[(1 - u_i^{(m)}) e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i) \\
& + u_i^{(m+1)} \left\{ \tau(1-\sigma) e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i) + (1-\tau) e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i) \right\} \Big] \\
& + u_i^{(m+1)} \Big[(1 - u_i^{(m)}) \left(1 - e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i)\right) \\
& + u_i^{(m)} \Big\{ \tau \Big[\sigma + (1-\sigma) \left(1 - e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i)\right) \\
& + (1-\tau) \left(1 - e^{-\lambda_i R_i^{(m)}} F^*_{|\mathbf{U}^{(m)}|}(\lambda_i)\right) \Big] \Big\} \Big] \Bigg\} .
\end{aligned}
\tag{4.33}
$$

Performance measures can be obtained as

$$\gamma_i = \frac{\{(1-\tau) + \tau(1-\sigma)\}\alpha_i\, b}{E[C]}, \tag{4.34}$$

$$E[W_i] = \frac{E[C]}{\{(1-\tau) + \tau(1-\sigma)\}\alpha_i} - \frac{1}{\lambda_i} \tag{4.35}$$

where $E[C]$ is given by (4.12).

For the simplified analysis, we have

$$\begin{aligned}
&\text{Prob}\,[u_N^{(m+1)} = r' \mid \underline{Q}_N^{(m+1)} = j - r',\ \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r] \\
&= (1 - u_N^{(m+1)})\Big[(1 - u_N^{(m)})e^{-\lambda R - \lambda b(j-r')}F_i^*(\lambda) \\
&\qquad + u_N^{(m)}\left\{\tau(1-\sigma)e^{-\lambda R - \lambda b(j-r')}F_i^*(\lambda) + (1-\tau)e^{-\lambda R - \lambda b(j-r')}F_i^*(\lambda)\right\}\Big] \qquad (4.36) \\
&+ u_N^{(m+1)}\Big[(1 - u_N^{(m)})\Big(1 - e^{-\lambda R - \lambda b(j-r')}F_i^*(\lambda)\Big) \\
&+ u_N^{(m)}\left\{\tau\Big[\sigma + (1-\sigma)\Big(1 - e^{-\lambda R - \lambda b(j-r')}F_i^*(\lambda)\Big)\Big] + (1-\tau)\Big(1 - e^{-\lambda R - \lambda b(j-r')}F_i^*(\lambda)\Big)\right\}\Big]
\end{aligned}$$

and

$$\gamma^a = \frac{\{(1-\tau) + \tau(1-\sigma)\}\alpha^a b}{E[C^a]}, \tag{4.37}$$

$$E[W^a] = \frac{E[C^a]}{\{(1-\tau) + \tau(1-\sigma)\}\alpha^a} - \frac{1}{\lambda}. \tag{4.38}$$

4.4.2. Numerical Algorithm

We evaluate the interconnected network system by the model where two token ring networks are connected by the bridge as illustrated in Figure 4.5. Below, we differentiate system parameters for network 1 from those for network 2 by superscripts (1) and (2). In the evaluated model, the offered traffic load to the bridge in network 1 is equal to total throughput of terminals of network 2 destined for network 1, and vice versa. Thus, we need an iterative solution algorithm to obtain performance measures. For this purpose, let us introduce $\lambda_0^{(1)}(m)$ ($\lambda_0^{(2)}(n)$) as the arrival rate to the bridge in network 1 (network 2) which is computed in m th (n th) iteration of the analysis of network 2 (network 1). The convergence criterion for the iteration is defined by

$$\Delta_m^{(1)} = \frac{|\lambda_0^{(1)}(m) - \lambda_0^{(1)}(m+1)|}{\lambda_0^{(1)}(m)} < \varepsilon\,, \tag{4.39}$$

and

$$\Delta_n^{(2)} = \frac{|\lambda_0^{(2)}(n) - \lambda_0^{(2)}(n+1)|}{\lambda_0^{(2)}(n)} < \varepsilon \tag{4.40}$$

for a small value of ε, e.g., $\varepsilon = 10^{-6}$. An outline of the iterative algorithm is stated as follows:

1. Set arrival rates to bridges and loss probabilities at the bridges to some initial values. (e.g., set $\lambda_0^{(1)}(0) = \tau^{(2)}N^{(2)}\lambda^{(2)}$, $\lambda_0^{(2)}(0) = \tau^{(1)}N^{(1)}\lambda^{(1)}$, and $\sigma^{(1)} = \sigma^{(2)} = 0$.)

2. If the convergence criterion is satisfied, stop the iteration. Otherwise, if

$$\frac{|\lambda_0^{(1)}(m) - \lambda_0^{(1)}(m+1)|}{\lambda_0^{(1)}(m)} \geq \frac{|\lambda_0^{(2)}(n) - \lambda_0^{(2)}(n+1)|}{\lambda_0^{(2)}(n)} \tag{4.41}$$

then go to Step 3 and analyze network 1, else go to Step 4.

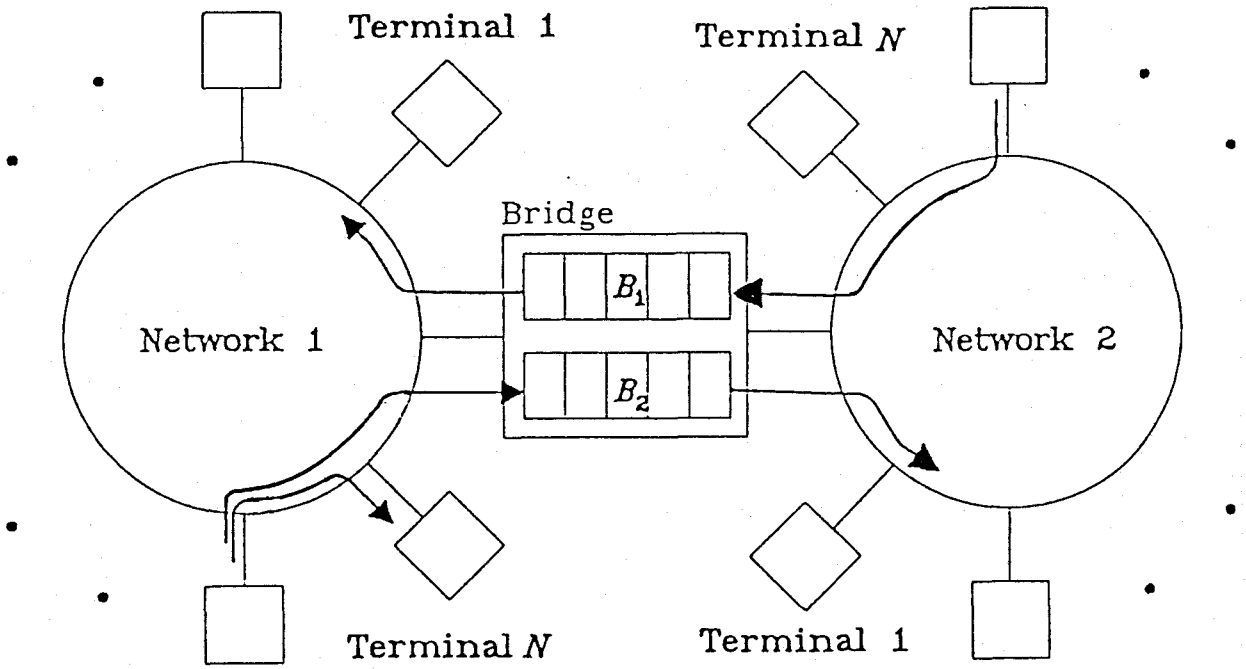


Figure 4.5. An Interconnected Token Ring Network Model

3. For the n th iteration, analyze network 1 using $\lambda_0^{(1)}(m)$ and derive performance measures, which will be used in the next iteration cycle as the arrival rates to the bridge in network 2, i.e.,

$$\lambda_0^{(2)}(n) = \tau^{(1)} \frac{\sum_{i=1}^{N^{(1)}} \alpha_i^{(1)}}{E[C^{(1)}]}, \quad \sigma^{(2)} = P_b^{(1)} \tag{4.42}$$

and go to Step 2.

4. For the m th iteration, analyze network 2 using $\lambda_0^{(2)}(n)$ and derive performance measures, which will be used in the next iteration cycle as the arrival rates to the bridge in network 1, i.e.,

$$\lambda_0^{(1)}(m) = \tau^{(2)} \frac{\sum_{i=1}^{N^{(2)}} \alpha_i^{(2)}}{E[C^{(2)}]}, \quad \sigma^{(1)} = P_b^{(2)} \tag{4.43}$$

and go to Step 2.

By using the above iterative solution algorithm, numerical results are obtained and compared to the simulation results. We consider a symmetric interconnected network model illustrated in Figure 4.5 with the following parameters:

Number of terminals:	$N^{(1)} = N^{(2)} = 8$
Message transmission time of terminals:	$b^{(1)} = b^{(2)} = 1$
Walking Times:	$r^{(1)} = r^{(2)} = 0.1$
Arrival rate to terminals:	$\lambda^{(1)} = \lambda^{(2)} = 0.0875$
Buffer length of the bridge:	$L^{(1)} = L^{(2)} = 5$
Message transmission time of the bridge:	$b_0^{(1)} = b_0^{(2)} = 1$

For the solution of each subnetwork, we employ the approximate solution presented in Section 4.3.

Figures 4.6.a-c show mean message waiting times, throughput and loss probability of network 1 (identical to those of network 2 by symmetry assumption) dependent on the external traffic ratio, $\tau^{(1)}(=\tau^{(2)})$. Agreement between analytical and simulation results is fairly good. However, we slightly underestimate the bridge loss probability overall by our assumption that messages arrive at the bridge according to Poisson. Although we do not give a proof for convergence in our iterative algorithm, all tried cases have converged throughout our numerical examples. For more precise analysis, we need a further study for an output process of the polling system and a $GI/G/1/L$ queue with vacation for the detailed analysis of the bridge.

4.5. Conclusion

We have developed an analytical method for a token ring network with single-buffer terminals and a finite capacity bridge whereby the bridge has priority over terminals. terminals transmit at most one message per token possession. Next, we have given a simplified analysis for the same model to reduce computational time and space for a symmetrical case where arrival rates are identical for all terminals. We have also developed methods applicable to interconnected networks where two token ring networks are connected by a bridge which has transmit and receive buffers. We have compared system performance measures computed from our iterative solution to the simulation results. For almost all tried cases, our analysis of the interconnected network is shown to be within 95% confidence intervals in the mean message waiting time and within a few percent error in the throughput. Our analytical method can be used for the design of the bridge to determine the size of the buffer.

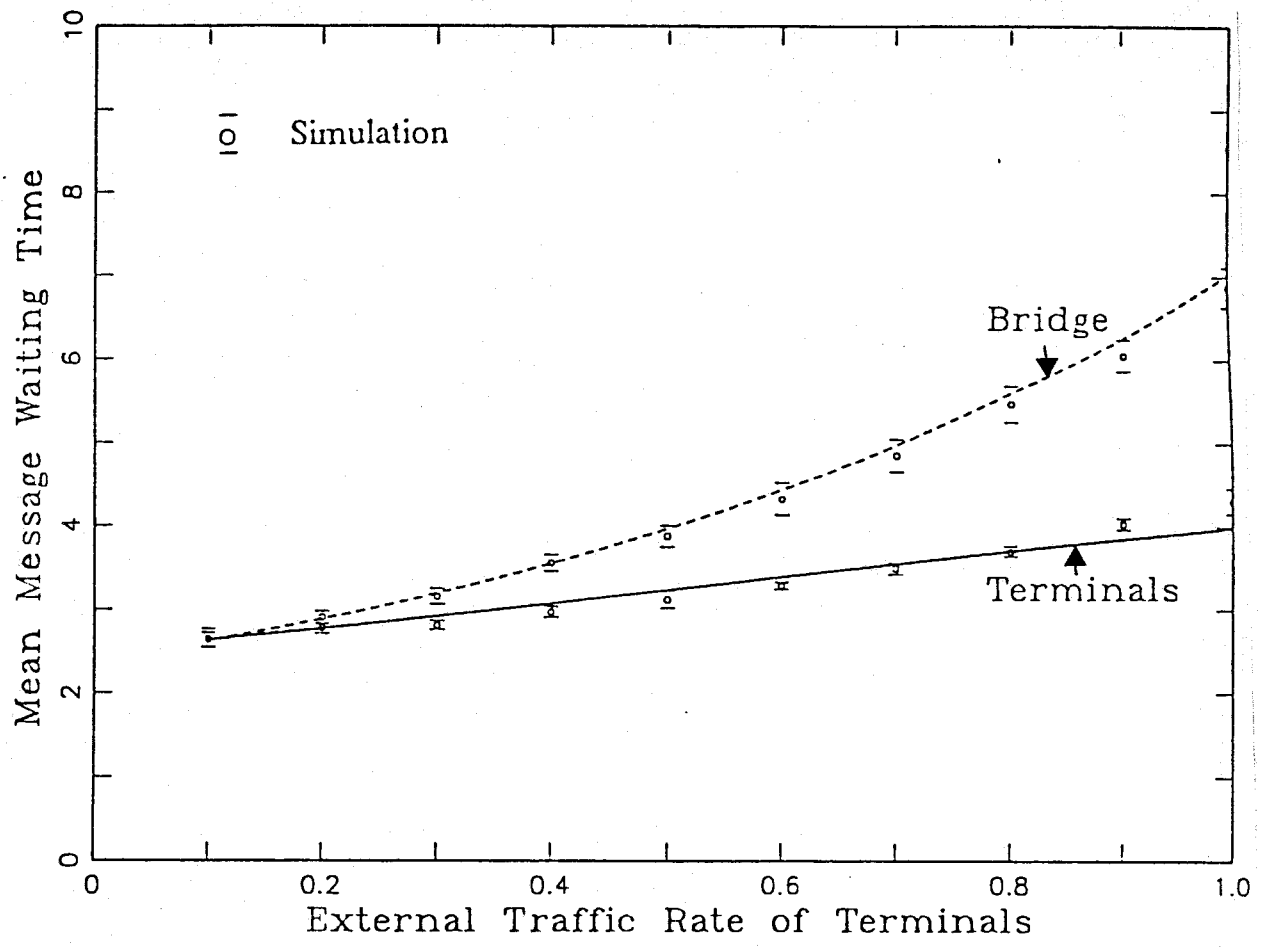


Figure 4.6-a. Mean Message Waiting Times in Interconnected Token Ring Network

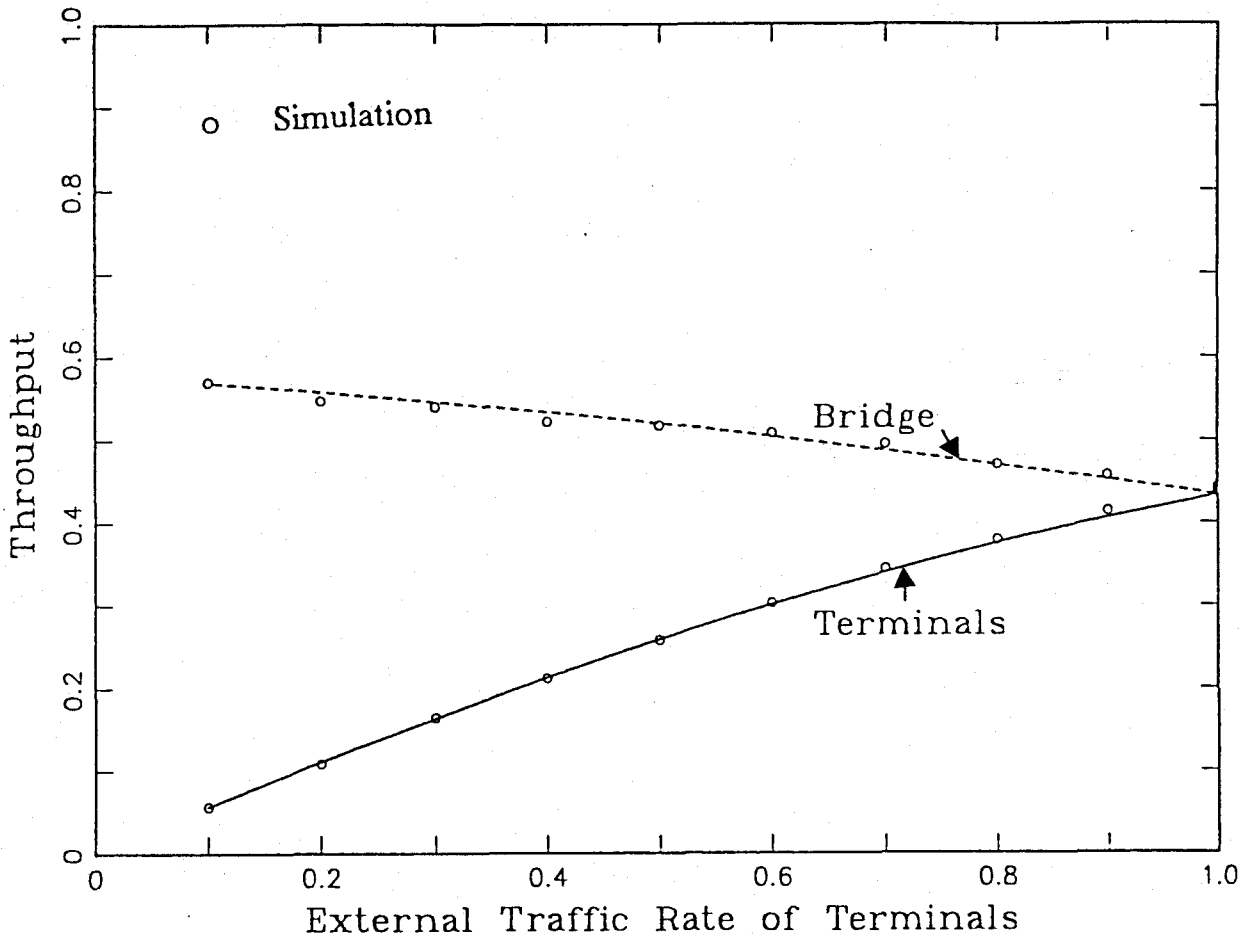


Figure 4.6-b. Throughput in Interconnected Token Ring Network

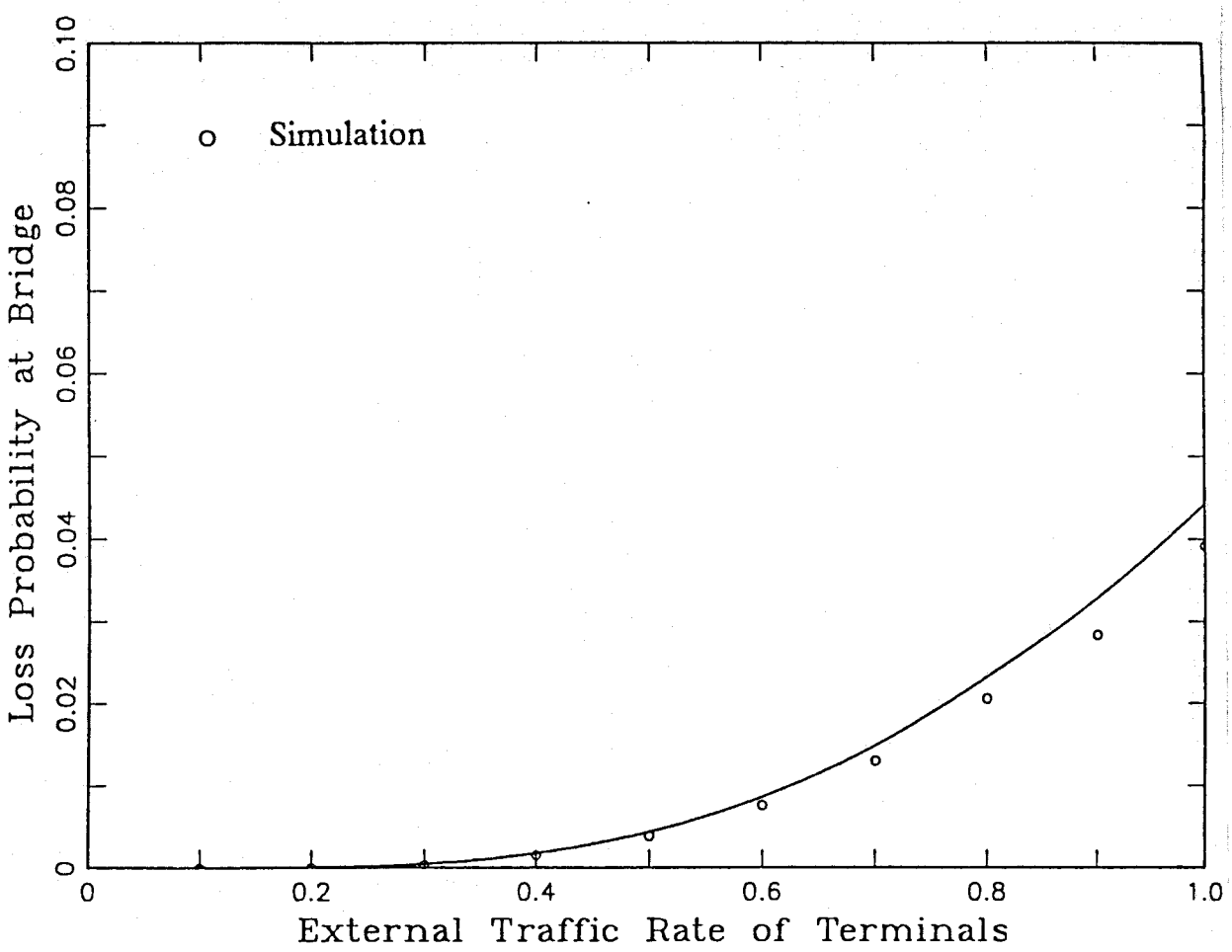


Figure 4.6-c. Loss Probability at Bridge in Interconnected Token Ring Network

Chapter 5. Concluding Remarks

5.1. Summary of Dissertation

We have studied theoretical performance-related issues in token ring networks.

In Chapter 2, we have considered the new sort of nonpreemptive priority queueing systems with/without switchover times. For each type, we have obtained the mean message waiting time for each class. We have also compared the degree of discrimination among the classes for the three types using the normalized squared deviations of the mean message waiting times.

Next, we have built a two-layer performance model of token ring networks in which a MAC layer submodel and Transport layer submodels are combined in Chapter 3. To deal with acknowledgement traffic with priority over data messages, we have revised an MVA priority approximation for the case where the priority can change at each queue. An iterative solution algorithm for this two-layer performance model is proposed and applied to the five models of communication systems: a symmetric-load, piggybacked acknowledgement model, a symmetric-load, explicit acknowledgement model, full-duplex communication models (with and without priorities), and a client/server model. System performance measures such as throughput and mean message delay computed from our analysis have been compared to the simulation results over a wide range of parameters. In most numerical examples, our approximate analysis has been

shown to be within 90% confidence intervals in the mean message delay and within a few percent error in the throughput.

In Chapter 4, we have treated a token ring network consisting of two kinds of stations: single-buffer stations and a station with finite capacity. First a detailed analysis is derived for the model. Next we have given a simplified analysis for the same model because the detailed analysis needs excessive computational time and space in the case of a middle/large number of stations in the network. Our method is applied to an interconnected token ring network model, each of which consists of single-buffer terminals and a finite capacity bridge. For the solution of the interconnected token ring networks, we have developed an iterative solution algorithm to obtain performance measures. System performance measures (mean message waiting time, throughput and loss probability at the bridge) computed from our analysis have been compared to the simulation results. For almost all tried cases, our approximate analysis have been shown to be within 95% confidence intervals in the mean message waiting time and within a few percent error in the throughput and loss probabiltiy.

5.2. Suggestions for Future Research

In Chapter 2, it is assumed that each priority class has an identical message length distribution. In actual integrated service LAN systems, however, it is highly possible that voice traffic assigned with a higher priority has a shorter length message distribution while bulk-data traffic like file transfer has a longer message length distribution. Thus, such a restriction should be loosened for this consideration.

In Chapter 4, we have treated the alternative method of the priority-mode operation to integrate the bridge and other terminals. However, the token reservation scheme may be a better candidate when we consider the environment in which various traffic other than the messages from the bridge exists. A comparative study between these two alternatives for a priority-mode operation is necessary for this purpose.

In the analysis of interconnected token ring networks in Chapter 4, the messages are assumed to arrive to the bridge according to a Poisson process. However, the actual arrival process is identical to the output process of other token ring networks. This implies that we may need more studies on (1) output processes of the token ring networks and (2) a $GI/G/1/L$ queue with vacations model for the bridge to know more precise behavior of the interconnected token ring networks.

For the total layer modeling which combines two or more layers of the OSI Reference Model, we have treated rather general communication protocols. More specific protocols such as a file transfer protocol may be a next target for a theoretical study. The total layer modeling involving priority-mode operations and/or interconnected networks is also required, which is applicable to performance evaluation of MANs.

Appendix A. Solution Algorithm for a Single-Chain Closed-Queueing Network

Our iterative algorithm to obtain the throughput and mean delays for the single-chain closed-queueing network is stated as follows:

1. Initialize arrival rates λ_i and λ_i^{base} for all i.
2. (Iteration cycle) Repeat until $\Delta_n < \varepsilon$ (e.g., $\varepsilon = 10^{-6}$).

   a. If the MAC layer submodel using (3.3) or (3.4) is infeasible, i.e.,

   $$\max(\lambda_i)Nr \geq 1 - \rho. \tag{A.1}$$

   then,

   $$\lambda_i = \frac{\lambda_i^{base} + \lambda_i}{2}. \tag{A.2}$$

   b. Calculate the mean message waiting times w_i using λ_i according to (3.3) or (3.4).

   c. For all closed queueing networks do:

      i. Solve for the throughput of chain i using MVA (3.6)-(3.8) and let it λ_i'.

ii. If the derived throughput λ_i' is greater than λ_i (this tends to happen when the MAC-queues are bottleneck), then let

$$\lambda_i^{base} = \lambda_i \,, \tag{A.3}$$

$$\lambda_i = \frac{\lambda_i + \lambda_i'}{2} \,. \tag{A.4}$$

Otherwise, let

$$\lambda_i = \frac{\lambda_i^{base} + \lambda_i}{2} \,. \tag{A.5}$$

Step 2.c.ii is required when the MAC-queues are bottleneck in the closed chain network. Figure A.1 illustrates relationships between throughputs and mean message waiting times for both MAC-delay and Transport-delay in two cases: (i) where the Transport-queues are bottleneck (Figure A.1-a), and (ii) where the MAC-queues are bottleneck (Figure A.1-b). (For the MAC delay, mean message waiting times will be derived from throughput and vice versa for the Transport delay.) When the bottleneck resides in the Transport-queue, a simple iterative algorithm can be applied because the solution always converges at the intersection as shown in Figure A.1-a. On the other hand, when the MAC-queues are bottleneck, introduction of another quantitative parameter, λ_i^{base}, has become necessary for the solution to converge, although this is a rather heuristic approach. In Step 2.c.ii, λ_i^{base} is the maximum value which guarantees that the iteration does not diverge in the current iteration. When the derived throughput λ_i' is greater than the previous one, λ_i, the current iteration is going to converge. Thus, λ_i^{base} can be increased to λ_i, and the greater value of λ_i will be tried in the next iteration cycle. On the other hand, when λ_i' is smaller than λ_i, it indicates that the iteration is going to fall into the divergence. So, λ_i^{base} cannot be changed and a smaller value of λ_i should be tried in the next cycle. In Figure A.1-b, the dotted line shows the appearance of the convergence when our modified algorithm is applied. The solid line shows the divergence in the simple solution which does not employ our algorithm.

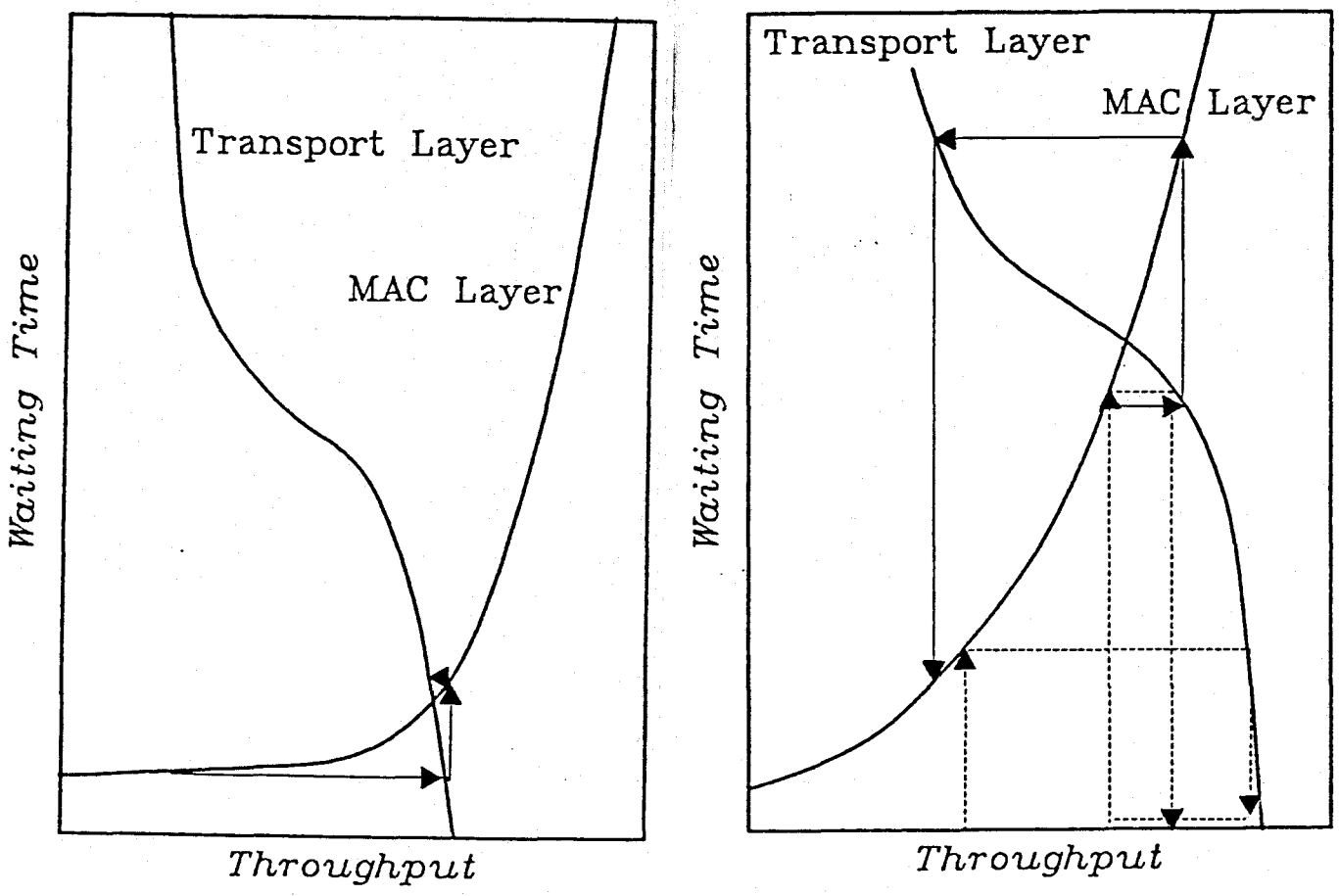


Figure A.1. Convergence Behavior of Iteration

Appendix B. Derivation of State Transition Probabilities from (4.27)

Each term in the right-hand side of (4.27) may be rewritten as

$$\begin{aligned}
&\text{Prob}\,[\underline{Q}_N^{(m+1)} = j - r', u_N^{(m+1)} = r', \mid \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r] \\
&= \text{Prob}\,[\underline{Q}_N^{(m+1)} = j - r' \mid \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r] \\
&\quad \bullet \text{Prob}\,[u_N^{(m+1)} = r' \mid \underline{Q}_N^{(m+1)} = j - r', \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad r, r' \in [0, 1]\,.
\end{aligned} \tag{B.1}$$

The probability, Prob $[u_N^{(m+1)} = r' \mid \underline{Q}_N^{(m+1)} = j - r', \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r]$, is given as

$$\begin{aligned}
&\text{Prob}\,[u_N^{(m+1)} = r' \mid \underline{Q}_N^{(m+1)} = j - r', \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r] \\
&= (1 - u_N^{(m+1)}) e^{-\lambda R - \lambda b(j-r')} F_i^*(\lambda) + u_N^{(m+1)} \left\{ 1 - e^{-\lambda R - \lambda b(j-r')} F_i^*(\lambda) \right\}
\end{aligned} \tag{B.2}$$

where

$$e^{-\lambda R - \lambda b(j-r')} F_i^*(\lambda) \tag{B.3}$$

represents the probability that N th terminal does not have a message at the $m + 1$ th polling instant. On the other hand, the probability, Prob $[\underline{Q}_N^{(m+1)} = j - r' \mid \underline{Q}_N^{(m)} = i - r, u_N^{(m)} = r]$, may be obtained in recursive form as explained below.

$$
\begin{aligned}
&\text{Prob}\,[\underline{Q}_N^{(m+1)} = k \mid \underline{Q}_N^{(m)} = l, u_N^{(m)} = r] \\
&= \frac{\binom{N-1}{l-1}}{\binom{N-1}{l}} \sum_{s'=0}^{1} \text{Prob}\,[\underline{Q}_{N-1}^{(m+1)} = k - s', u_{N-1}^{(m+1)} = s' \mid \underline{Q}_{N-1}^{(m)} = l, u_{N-1}^{(m)} = 0, \overline{Q}_{N-1}^{(m)} = r] \\
&\quad + \frac{\binom{N-2}{l-1}}{\binom{N-1}{l}} \sum_{s'=0}^{1} \text{Prob}\,[\underline{Q}_{N-1}^{(m+1)} = k - s', u_{N-1}^{(m+1)} = s' \mid \underline{Q}_{N-1}^{(m)} = l-1, u_{N-1}^{(m)} = 1, \overline{Q}_{N-1}^{(m)} = r] \\
&= \frac{\binom{N-1}{l-1}}{\binom{N-1}{l}} \sum_{s'=0}^{1} \left(\begin{array}{l} \text{Prob}\,[\underline{Q}_{N-1}^{(m+1)} = k - s' \mid \underline{Q}_{N-1}^{(m)} = l, u_{N-1}^{(m)} = 0, \overline{Q}_{N-1}^{(m)} = r] \\ \bullet \text{Prob}\,[u_{N-1}^{(m+1)} = s' \mid \underline{Q}_{N-1}^{(m+1)} = k - s', \underline{Q}_{N-1}^{(m)} = l, u_{N-1}^{(m)} = 0, \overline{Q}_{N-1}^{(m)} = r] \end{array} \right) \\
&\quad + \frac{\binom{N-2}{l-1}}{\binom{N-1}{l}} \sum_{s'=0}^{1} \left(\begin{array}{l} \text{Prob}\,[\underline{Q}_{N-1}^{(m+1)} = k - s' \mid \underline{Q}_{N-1}^{(m)} = l-1, u_{N-1}^{(m)} = 1, \overline{Q}_{N-1}^{(m)} = r] \\ \bullet \text{Prob}\,[u_{N-1}^{(m+1)} = s' \mid \underline{Q}_{N-1}^{(m+1)} = k - s', \underline{Q}_{N-1}^{(m)} = l-1, u_{N-1}^{(m)} = 1, \overline{Q}_{N-1}^{(m)} = r] \end{array} \right) \\
&\qquad\qquad r \in [0, 1], \quad k, l \in [0, 1, \ldots, N-1]. \qquad \text{(B.4)}
\end{aligned}
$$

In each step, we have the probability $\text{Prob}\,[\underline{Q}_j^{(m+1)} = k \mid \underline{Q}_j^{(m)} = l, u_j^{(m)} = r, \overline{Q}_j^{(m)} = n]$, which can be obtained recursively as follows:

$$
\begin{aligned}
&\text{Prob}\,[\underline{Q}_j^{(m+1)} = k \mid \underline{Q}_j^{(m)} = l, u_j = r, \overline{Q}_j^{(m)} = n] \\
&= \frac{\binom{N-j}{l-1}}{\binom{N-j}{l}} \sum_{s'=0}^{1} \text{Prob}\,[\underline{Q}_{j-1}^{(m+1)} = k - s', u_{j-1}^{(m+1)} = s' \mid \underline{Q}_{j-1}^{(m)} = l, u_{j-1}^{(m)} = 0, \overline{Q}_{j-1}^{(m)} = n + r] \\
&\quad + \frac{\binom{N-j-1}{l-1}}{\binom{N-j}{l}} \sum_{s'=0}^{1} \text{Prob}\,[\underline{Q}_{j-1}^{(m+1)} = k - s', u_{j-1}^{(m+1)} = s' \mid \underline{Q}_{j-1}^{(m)} = l-1, u_{j-1}^{(m)} = 1, \overline{Q}_{j-1}^{(m)} = n + r]
\end{aligned}
$$

$$= \frac{\binom{N-j}{l-1}}{\binom{N-j}{l}} \sum_{s'=0}^{1} \left(\begin{array}{l} \text{Prob}\,[\underline{Q}_{j-1}^{(m+1)} = k - s' \mid \underline{Q}_{j-1}^{(m)} = l, u_{j-1}^{(m)} = 0, \bar{Q}_{j-1}^{(m)} = r] \\ \bullet \text{Prob}\,[u_{j-1}^{(m+1)} = s' \mid \underline{Q}_{j-1}^{(m+1)} = k - s', \underline{Q}_{j-1}^{(m)} = l, u_{j-1}^{(m)} = 0, \bar{Q}_{j-1}^{(m)} = r] \end{array} \right)$$

$$+ \frac{\binom{N-j-1}{l-1}}{\binom{N-j}{l}} \sum_{s'=0}^{1} \left(\begin{array}{l} \text{Prob}\,[\underline{Q}_{j-1}^{(m+1)} = k - s' \mid \underline{Q}_{j-1}^{(m)} = l-1, u_{j-1}^{(m)} = 1, \bar{Q}_{j-1}^{(m)} = r] \\ \bullet \text{Prob}\,[u_{j-1}^{(m+1)} = s' \mid \underline{Q}_{j-1}^{(m+1)} = k - s', \underline{Q}_{j-1}^{(m)} = l-1, u_{j-1}^{(m)} = 1, \bar{Q}_{j-1}^{(m)} = r] \end{array} \right)$$

$$r \in [0, 1], \quad k, l \in [0, 1, \ldots, j-1], \quad n \in [0, 1, \ldots, N-l-1]. \tag{B.5}$$

In the above equation, $\text{Prob}\,[\underline{Q}_{j-1}^{(m+1)} = k - s' \mid \underline{Q}_{j-1}^{(m)} = l - s, u_{j-1}^{(m)} = s, \bar{Q}_{j-1}^{(m)} = r]$ has an immediate solution:

$$\begin{aligned} &\text{Prob}\,[\underline{Q}_{j-1}^{(m+1)} = k - s', \mid \underline{Q}_{j-1}^{(m)} = l - s, u_{j-1}^{(m)} = s, \bar{Q}_{j-1}^{(m)} = r] \\ &\quad = (1 - u_{j-1}^{(m+1)}) e^{-\lambda R - \lambda b(n+r+k-s')} F_{l+n+r}^{*}(\lambda) \\ &\qquad + u_{j-1}^{(m+1)} \left(1 - e^{-\lambda R - \lambda b(n+r+k-s')} F_{l+n+r}^{*}(\lambda)\right). \end{aligned} \tag{B.6}$$

On the other hand, the probability, $\text{Prob}\,[\underline{Q}_{1}^{(m+1)} = k - s', \mid \underline{Q}_{1}^{(m)} = l - 1, u_{1}^{(m)} = 1, \bar{Q}_{1}^{(m)} = r]$, needs more recursive calculations. The final step of the recursive algorithm is given as follows:

$$\begin{aligned} &\text{Prob}\,[\underline{Q}_{2}^{(m+1)} = k \mid \underline{Q}_{2}^{(m)} = l, u_{2}^{(m)} = r, \bar{Q}_{2}^{(m)} = n] \\ &\quad = \text{Prob}\,[u_{1}^{(m+1)} = k \mid u_{1}^{(m)} = l, u_{2}^{(m)} = r, \bar{Q}_{2}^{(m)} = n] \\ &\quad = (1 - u_{1}^{(m+1)}) e^{-\lambda R - \lambda b(n+r)} F_{l+n+r}^{*}(\lambda) + u_{1}^{(m+1)} \left(1 - e^{-\lambda R - \lambda b(n+r)} F_{l+n+r}^{*}(\lambda)\right) \end{aligned} \tag{B.7}$$

Our recursive algorithm for the solution may be represented by a recursive tree structure shown in Figure B.1.

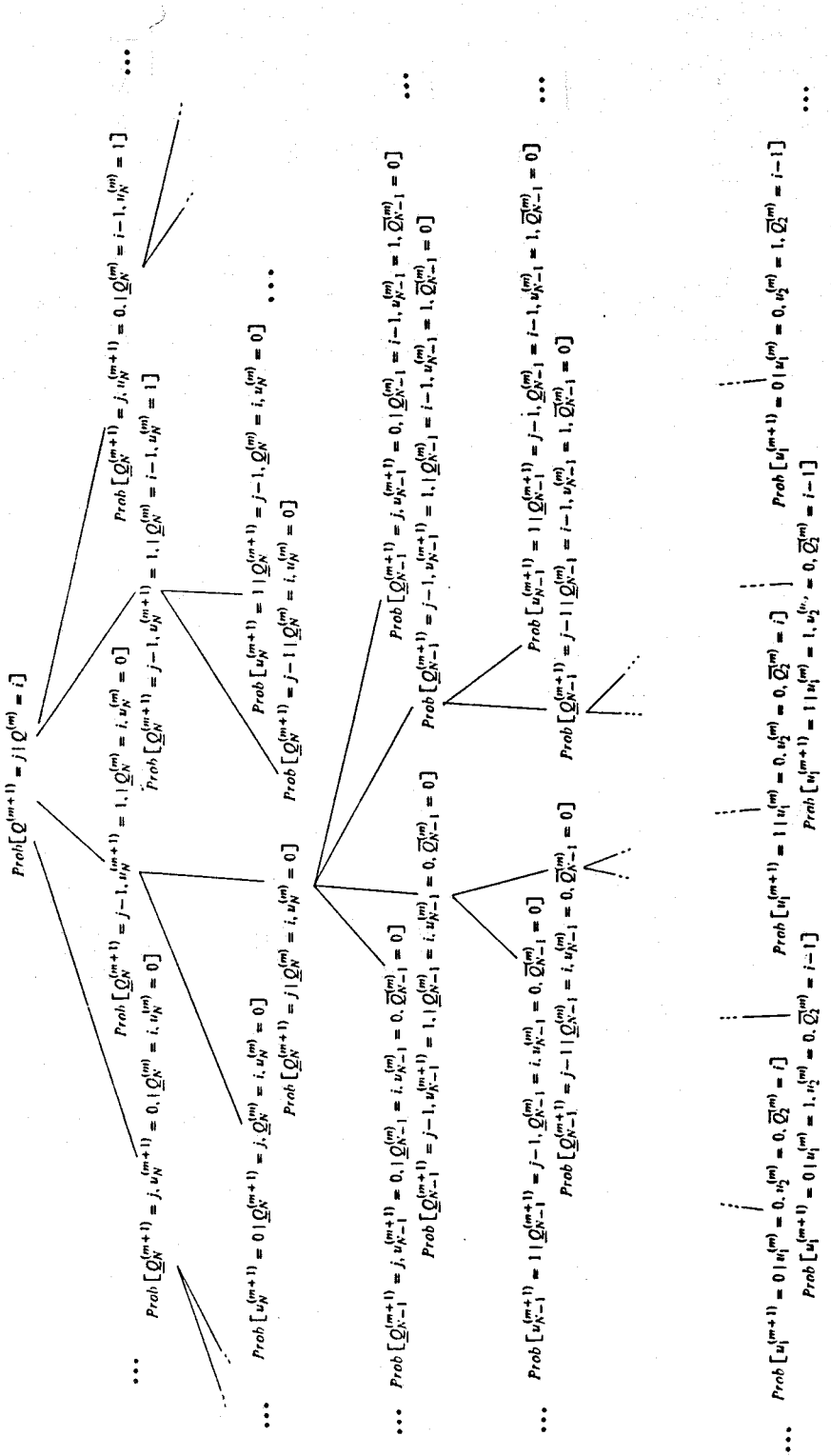


Figure B.1. Recursive Tree Structure of State Transition Probabilities

References

[Bond87] Bondi, A.B., "Decomposition Approaches to Modelling LAN Contention and Host Computer Performance," *Computer Communications*, Vol.10, No.2, pp.70-78, April 1987.

[Boxm86] Boxma, O.J. and B.Meister, "Waiting Time Approximations for Cyclic-Service Systems with Switch-over Times," *Performance Evaluation Review*, Vol.14, No.1, pp.254-262, May 1986.

[Brya84] Bryant, R.M., M.S.Lakshmi, K.M.Chandy and A.E.Krzesinski, "The MVA Priority Approximation" IBM Research Report RC 10606, Yorktown, 1984.

[Bux81a] Bux, W., F.Closs, P.A.Janson, K.Kummerle and H.R.Mueller, "A Reliable Token-Ring System for Local-area Communication," *Conference Record of NTC'81*, IEEE, Piscataway, N.J., pp.A2.2.1-A.2.2.6, 1981.

[Bux81b] Bux, W., "Local-area Subnetworks: A Performance Comparison," *IEEE Transactions on Communications*, Vol.COM-29, No.10, pp.1465-1473, February 1981.

[Bux84] Bux, W., "Performance Issues in Local-area Networks," *IBM Systems Journal*, Vol.23, No.4, pp.351-374, 1984.

[Bux85] Bux, W. and D.Grillo, "Flow Control in Local-Area Networks of Interconnected Token Rings," *IEEE Transactions on Communications*, Vol.COM-33, No.10, pp.1058-1066, October 1985.

[Cheo82] Cheong, W. and R.A.Hirschheim, *Local Area Networks*, Wiley and Sons, 1982.

[Coop81] Cooper, R.B., *Introduction to Queueing Theory, Second Edition,* Exercise 5.12., North-Holland Publishing Company, 1981.

[Cour80] Courtois, P.J., "The $M/G/1$ Finite Capacity Queue with Delays," *IEEE Transactions on Communications*, Vol.COM-28, No.2, pp.165-172, February 1980.

[Cox61] Cox, D.R., and W.L.Smith, *Queues,* Section 3.3., Methuen and Co. Ltd, 1961.

[Dixo83] Dixon R.C., N.C.Strole and J.D.Markov, "A Token-Ring Network for Local Area Communication," *IBM Systems Journal*, Vol.22, Nos.1 & 2, pp.47-62, 1983.

[Ferg85] Ferguson, M.J., and Y.Aminetzah, "Exact Results for Nonsymmetric Token Ring Systems," *IEEE Transactions on Communications*, Vol.COM-33, No.3, pp.223-231, March 1985.

[Gihr86] Gihr, H. and P.J.Kuehn, "Comparison of Communication Services with Connection-oriented and Connectionless Data Transmission," *Computer Networking and Performance Evaluation*, T.Hasegawa, H.Takagi and Y.Takahashi (editors), pp.173-186, Elsevier Science Publishers B.V. (North-Holland), 1986.

[Haye84] Hayes, J.F., *Modeling and Analysis of Computer Communications Networks,* Section 6.3., Plenum Press, 1984.

[Heym69] Heyman, D.P., "A Priority Queueing System with Server Interference," *SIAM Journal of Applied Mathematics*, Vol.17, No.1, pp.74-82, 1969.

[IEEE85a] ANSI/IEEE Standard 802.3: Carrier Sense Multiple Access with Collision Detection Access Method, IEEE Press, 1985.

[IEEE85b] ANSI/IEEE Standard 802.4: Token-Passing Bus Access Method, IEEE Press, 1985.

[IEEE85c] ANSI/IEEE Standard 802.5: Token Ring Access Method, IEEE Press, 1985.

[IEEE86] Draft IEEE Standard 802.6 Metropolitan Area Network Media Access Control, Revision G, July 1986.

[ISO83] *ISO Reference Model of Open-Systems Interconnection* ISO/TC97/Sc16, DP7498; available from the American National Standards Institute, 1430 Broadway, New York, NY 10018, January 1983.

[Kauf84] Kaufman, J.S., "Approximation Methods for Networks of Queues with Priorities," *Performance Evaluation*, Vol.4, No.3, pp.183-198, August 1984.

[Klei75] Kleinrock, L. and F.A.Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, Vol.COM-23, No.12, pp.1400-1416, 1975.

[Klei76] Kleinrock, L., *Queueing Systems, Volume 2: Computer Applications,* Sections 3.4 and 3.6., John Wiley and Sons, Inc., 1976.

[Kueh79] Kuehn, P.J., "Multiqueue Systems with Nonexhaustive Cyclic Service," *The Bell Systems Technical Journal*, Vol.58, No.3, pp.671-698, March 1979.

[Lee84] Lee, T.T., "$M/G/1/N$ Queue with Vacation Time and Exhaustive Service Discipline," *Operations Research*, Vol.32, No.4, pp.774-784, August 1984.

[Meis85] Meister B., P.Janson and L.Svobodova, "File Transfer in Local Area Networks; A Performance Study," *IEEE Transactions on Computers*, Vol.C-34, No.12, pp.1164-1173, December 1985.

[Mitc86] Mitchell, L.C. and D.A.Lide, "End to End Performance Modeling of Local Area Networks," *IEEE Journal on Selected Areas in Communication*, Vol.SAC-4, No.6, pp.975-985, September 1986.

[Mura86] Murata, M. and H.Takagi, "Mean Waiting Times in Nonpreemptive Priority $M/G/1$ Queues with Server Switchover Times,", *Teletraffic Analysis and Computer Performance Evaluation*/ O.J.Boxma, J.W.Coehn and H.C.Tijms (editors), pp.395-407, Elsevier Science Publishers B.V. (North-Holland), 1987.

[Mura87a] Murata, M. and H.Takagi, "Two-Layer Modeling for Local Area Networks," *IEEE INFOCOM '87*, pp.132-140, San Francisco, March 30 - April 2, 1987.

[Mura87b] Murata, M. and H.Takagi, "Two-Layer Modeling for Local Area Networks", submitted to *IEEE Transactions on Communications*.

[Mura87c] Murata, M. and H.Takagi, "Performance of Token Ring Networks with a Finite Capacity Bridge,", submitted to *IEEE Transactions on Communications*.

[Nish86] Nishida, T., M.Murata, H.Miyahara and K.Takashima, "Dynamic Congestion Control in Interconnected Local Area Networks," *Local Area & Multiple Access Networks*, R.L.Pickholtz (editor), Chapter 6, pp.107-136, Computer Science Press, 1986.

[Reis79] Reiser, M., "A Queueing Network Analysis of Computer Communication Networks with Window Flow Control," *IEEE Transactions on Communications*, Vol.COM-27, No.8, pp.1199-1209, August 1979.

[Reis80] Reiser, M. and S.S.Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks," *Journal of the Association for Computing Machinery*, Vol.27, No.2, pp.313-322, April 1980.

[Reis86] Reiser, M., "Communication-System Models Embedded in the OSI-Reference Model - A Survey," *Computer Networking and Performance Evaluation*, T.Hasegawa, H.Takagi and Y.Takahashi (editors), pp.85-111, Elsevier Science Publishers B.V. (North-Holland), 1986.

[Saue84] Sauer, C.H., E.A.MacNair and J.F.Kurose, "Queueing Network Simulations of Computer Communication," *IEEE Journal on Selected Areas in Communication*, Vol.SAC-2, No.1, pp.203-220, January 1984.

[Schm84] Schmitt, W., "On Decompositions of Markovian Priority Queues and Their Application to the Analysis of Closed Priority Queueing Networks," *Performance '84*, pp.393-407, December 1984.

[Stal84] Stallings, W., *Local Networks: An Introduction,* Section 5.4., Macmillan Publishing Company, 1984.

[Stro83] Strole, N.C., "A Local Communications Network Based on Interconnected Token-Access Rings: A Tutorial," *IBM Journal of Research and Development*, Vol.27, pp.481-496, 1983.

[Svob85] Svobodova, L., "Client/Server Model of Distributed Processing", IBM Research Report RZ 1350, Zurich, 1985.

[Taka85] Takagi, H., "Mean Message Waiting Times in Symmetric Multi-Queue Systems with Cyclic Service," *Performance Evaluation*, Vol.5, No.4, pp.271-277, November 1985.

[Taka86a] Takagi, H., *Analysis of Polling Systems*, The MIT Press, 1986.

[Taka86b] Takagi, H. and M.Murata, "Queueing Analysis of Nonpreemptive Reservation Priority Discipline," *Performance '86 and ACM SIGMETRICS 1986, Performance Evaluation Review*, Vol.14, No.1, pp.237-244, May 1986.

[Taki86] Takine, T., Y.Takahashi and T.Hasegawa, "Performance Analysis of a Polling System with Single Buffers and Its Applications to Interconnected Networks," *IEEE Journal on Selected Areas in Communication*, Vol.SAC-4, No.6, pp.802-812, November 1986.

[Toba77] Tobagi, F.A., and L.Kleinlock, "Packet Switching in Radio Channels: Part IV - Stability Considerations and Dynamic Control in Carrier Sence Multiple Access Modes," *IEEE Transactions on Communications*, Vol.COM-25, No.10, pp.1103-1120, 1976.

[Wong82] Wong, J.W., J.P.Sauve and J.A.Field, "A Study of Fairness in Packet Switching Networks," *IEEE Transactions on Communications*, Vol.COM-30, pp.346-353, 1982.