

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/30848849>

Analytic evaluation of contention protocols used in distributed real-time systems

Article in Real-Time Systems · July 1995

DOI: 10.1007/BF01094173 · Source: OAI

CITATIONS

7

READS

24

2 authors, including:



Kang G. Shin

University of Michigan

203 PUBLICATIONS 5,252 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



performance evaluation of MIMO multiplexing systems in HetNets [View project](#)

Analytic Evaluation of Contention Protocols Used in Distributed Real-Time Systems*

KANG SHIN

kgshin@eecs.umich.edu

*Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48105-2122*

CHAO-JU HOU

jhou@ecc.wisc.edu

*Department of Electrical and Computer Engineering
The University of Wisconsin – Madison
Madison, WI 53706-1691*

Received May 1, 1991

Editor:

Abstract. The probability of a station failing to deliver packets before their deadlines, called the *probability of dynamic failure*, P_{dyn} , is an important measure for the communication subsystem of a distributed real-time system. Another closely-related performance measure is the ε -*bounded delivery time*, T_ε , which is defined as the least time needed to deliver a packet with probability greater than $1 - \varepsilon$. Using P_{dyn} and T_ε , we comparatively evaluate four contention protocols often used in distributed real-time systems: (i) the token passing protocol and its priority-based variation (called the *token scheduling protocol*), and (ii) the P_i -persistent protocol and a priority-based variation thereof. The communication subsystem equipped with different contention protocols is modeled first as embedded Markov chains. Then, we derive the probability distributions of access delay, from which P_{dyn} and T_ε can be calculated. The blocking probability, Q_i , can also be derived from the access delay distribution. These measures are derived first under the assumption of a single buffer at each station. The single-buffer model is then extended to the multiple-buffer case. The effects of buffer size on P_{dyn} , T_ε , and Q_i , and the performance improvement with multiple buffers are analyzed over a wide range of network traffic.

Keywords: Deadlines, real-time systems, contention protocols, access delay, embedded Markov chains, queuing theory, performance metrics, performance evaluation.

1. Introduction

Due to their potential for high performance and high reliability, distributed computing systems are being used for an increasing number of real-time applications which execute computational tasks. Real-time tasks often have to communicate with one another via a common communication medium in order to collectively perform some useful function. Each real-time task must be completed before its deadline, and thus, the messages generated/needed by this task must be delivered/received in a timely manner.

* The work reported in this paper was supported in part by the ONR under Grant N00014-92-J-1080. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the ONR.

Contention protocols are used to regulate the access of all stations — where real-time tasks are executed — to the common medium as well as to support efficient sharing of the medium. Any contention protocol used for real-time applications must therefore minimize the probability of a packet missing its delivery deadline by properly arbitrating access to the communication medium.

Contention protocols have been studied extensively for general-purpose distributed systems, such as the token-passing protocol [4], [3], [15], [24], [5], [8] and the CSMA/CD protocol [26], [21], [2], [1], [28], [14]. See [22] or [23] for a good survey. The protocols treated in these papers were originally developed for data transmission without regard to real-time services. Moreover, all but the results reported in [24], [8], [14] are concerned only with the evaluation of some *average* performance measures, such as aggregate throughput, average packet delay, and channel utilization. Although these average measures provide a useful yardstick for system performance, they are inadequate in assessing real-time performances, since the timing requirement of *each* individual task/message is not considered.

Recently, considerable attention has been paid to the development/modification of protocols to handle time-constrained communications. Manfield [13] proposed and analyzed a priority-polling system where high-priority messages are assumed to arrive at a single queue, and the queue holding high-priority messages and the other queues holding low-priority messages are alternately polled. Low-priority queues are polled cyclically between high-priority polls. In [6], he also presented and analyzed another priority-polling scheme where the system is assumed to instantaneously detect the occurrence of a high-priority event. Rubin and Tsai [16] investigated a message-priority-based token scheme where different service disciplines are employed for different classes of messages. Tight upper and lower bounds on the mean message delays were obtained. In another paper [27], they presented and analyzed two related token schemes which provide priority services to high-priority messages, and restrict the channel access of low-priority messages. Although these papers aimed at the delivery of high-priority (time-constrained) packets with a prescribed access delay, their analyses focused on the mean delay experienced by messages of different priorities, rather than the delay experienced by *each* message.

Using the fraction of packets missing their deadlines as a performance measure, Zhao *et al.* developed a virtual time CSMA protocol [29] and a sliding window protocol [30], both of which are based on the minimum-laxity-first policy for scheduling time-constrained packets. Strosnider *et al.* [20], [19] made use of the IEEE 802.5 Token Ring Standard to realize the token ring scheduling protocol which can support up to 8 different priority levels and provide highly responsive and deterministic services for time-constrained messages. Kurose *et al.* [12] determined (based on a semi-Markov decision model) the various operational parameters for the time window protocol to optimize its time-constrained performance. However, the above work (perhaps except for [12]) relied solely on simulations to show the timing property of their proposed protocols. Little has been reported on the analytic evaluation of real-time contention protocols with respect to measures suitable for real-time applications, such as the probability of *each* packet missing its deadline. We will in this paper counter this problem by *analytically*

evaluating the performance of four contention protocols for real-time applications on a per-packet basis.

One important performance measure for any real-time communication subsystem is the probability of a station failing to deliver packets before their deadlines, called the *probability of dynamic failure*, P_{dyn} [11], [17], [18]. Another related performance measure is the ε -*bounded delivery time*, T_ε , which is defined as the least time needed to deliver a packet (measured since its arrival) with probability greater than $1 - \varepsilon$ [18]. These two measures are derived for both non priority-based and priority-based token-type protocols. The former is often referred to as the *token passing protocol*, and the latter as the *token scheduling protocol* [20], [19]. The P_i -persistent protocol and a priority variation thereof are also analyzed using P_{dyn} and T_ε .

A communication subsystem with different contention protocols is modeled as embedded Markov chains. The probability distribution of the *medium access delay* D_A — the time elapsed from the arrival to the beginning of transmission of a packet — is then derived. Both P_{dyn} and T_ε can be calculated from the distribution of access delay. The probability of packet rejection, Q_i , can also be derived from the access delay distribution. These measures are then used to evaluate the performance of four contention protocols. The derivation is first based on the assumption that each station has only a single buffer. This assumption greatly reduces modeling complexity, yet the results so derived can still represent the performance of contention protocols, with respect to the ability in regulating medium accesses according to the priority of the packet waiting at each station. Then, we extend the single-buffer model to the multiple-buffer case, and evaluate (i) the performance improvement over the single-buffer case, and (ii) the effects of buffer size on P_{dyn} , T_ε , and Q_i .

The remainder of this paper is organized as follows. The system model used and the contention protocols under consideration are described in Section 2. The relations among P_{dyn} , T_ε , and the distribution of access delay are also addressed there. In Section 3, we develop performance models for the contention protocols under the single-buffer assumption. The probability distribution of access delay is then derived. Section 4 extends the single-buffer model to the multiple-buffer case. Section 5 presents numerical results of protocol evaluation in terms of P_{dyn} , T_ε , and Q_i . We also simulated the four protocols in order to validate analytic models. The paper concludes with Section 6.

2. System Model, Performance Measures, and Protocols

2.1. System Model

The system under consideration is homogeneous and consists of N stations. Packet arrivals at station i are assumed to follow a Poisson process with rate λ_i , and are independent of those of other stations. Packets are assumed to be of fixed length, each requiring m units of time for transmission. There are p priority levels for packets with 1 being the highest. Packets arriving at each station are classified into different priority levels according to their laxities. (Note that the laxity, ℓ , of a packet is defined as the latest time a packet must start transmission in order to meet its deadline.) For example,

a packet with laxity ℓ is assigned priority level k when $\ell \in [\ell_{k-1}, \ell_k)$, where $\ell_{k-1} < \ell_k$, $1 \leq k \leq p$, $\ell_0 \triangleq 0$, and the values of ℓ_k 's depends on the distribution of packet laxity and the application-dependent criterion. Under the assumption that packet arrivals are independent of one another, arrivals of priority- k packets at station i form a Poisson process with rate λ_{ik} , where $\frac{\lambda_{ik}}{\lambda_i}$ is the portion of priority- k packets arriving at station i .

Under the single-buffer assumption, each station can have at most one outstanding packet at any time, so those packets arriving at an already-occupied station will be rejected. One can then concentrate on the access delay experienced by a packet due to contention with other stations in accessing the common medium.

In the multiple-buffer model, there are L buffers in each station. For simplicity, only two priority levels are assumed to exist. This is not a limiting assumption, since it can be easily extended to the case of more than two priority levels. Moreover, the results obtained for high-priority packets under the two priority-level assumption can also be used to represent the delay experienced by high-priority packets in case of p -priority levels ($p \geq 3$) by grouping low $p - 1$ levels into one level.

For non priority-based protocols, packets are queued and served on a first-come-first-served basis. For priority-based protocols, however, a buffer of size L_1 is used to store high-priority packets so that at most L_1 high-priority packets may be held at any time in each station. A high-priority packet arriving at a station whose L_1 buffers are all occupied is rejected. Another queue of size $L_2 = L - L_1$ is set aside to hold low-priority packets. Since there must be an upper bound on the number of outstanding high-priority packets in the system in order to guarantee the delivery of every high-priority packet before its deadline, the value of L_1 is usually small. Low-priority packets at a station can compete with other stations for medium access only in the absence of high-priority packets waiting at this station. At most one packet can be transmitted during each possession of the token, since packets are prioritized according to their laxities, and thus should compete for medium access on a per-packet basis. (This is often termed as a *limited service discipline*).

2.2. Performance Measures

Instead of using the usual average measures, we will use P_{dyn} and T_e to evaluate the performance of contention protocols. These measures can be readily calculated from the cumulative distribution of access delay. Specifically, given the distribution of access delay experienced by priority- k packets, $P(D_A^{<k>} \leq t)$, one can calculate the conditional probability of dynamic failure given the laxity ℓ of a priority- k packet as $P_{dyn|l}^{<k>} \equiv 1 - P(D_A^{<k>} \leq \ell)$. The calculation of the probability of dynamic failure for priority- k packets, $P_{dyn}^{<k>}$, depends on how the laxity of a packet is converted to its priority. For example, let $P_{\ell ax}(t)$, $t \geq 0$, denote the probability density of packet laxity, then $P_{dyn}^{<k>}$ can be calculated as:

$$P_{dyn}^{<k>} = \int_{\ell_{k-1}}^{\ell_k} P_{dyn|t}^{<k>} P_{\ell ax}(t) dt = \int_{\ell_{k-1}}^{\ell_k} P(D_A^{<k>} > t) P_{\ell ax}(t) dt.$$

On the other hand, the ε -bounded delivery time, $T_\varepsilon^{<k>}$, for priority- k packets is computed as the smallest t such that $P_{dym|t}^{<k>} \leq \varepsilon$. In other words, $T_\varepsilon^{<k>}$ is the upper bound of packet laxity if a priority- k packet has to be delivered before its deadline with probability $1 - \varepsilon$.

2.3. Operations of the Four Contention Protocols

In the token passing protocol, a token is passed around the stations, and any station which desires to use the medium has to capture the token. After using the medium, the station releases the token. The station in the downstream neighboring the current medium master gets the highest priority. If this station has packets to transmit, it seizes the token; otherwise, the token is passed on to the next (logically) adjacent station. Fair access to the bus is one merit of this protocol, but a packet with the highest priority (e.g., minimum laxity) may have to wait until all the packets in its upstream stations are transmitted, and thus, may get delayed and/or even miss its deadline.

The token scheduling protocol [20], [19] is designed to alleviate the above drawback by restricting fairness to packets only within the same priority level. When there are outstanding high-priority packets, low-priority packets should wait until the high-priority packets have been transmitted. A token with a reservation field is passed around the stations on the bus. Each station examines the reservation field of the token as it passes through the station and inserts the priority of its outstanding packet (if any) if and only if its priority is higher than the one currently in the reservation field. The priority recorded in the reservation field thus reflects the highest priority of those outstanding packets in the system. A station can capture the token and transmit its packets only after the token returns to the station with its registered priority. This guarantees the station with the highest-priority packet to gain medium access. We will develop analytic models to quantitatively evaluate these two protocols.

The P_i -persistent applies to *slotted-time* mediums and stations can start transmitting packets only at the beginning of each time slot. A station with an outstanding packet senses the medium. If the medium is idle, the station transmits the packet with a probability p_i . With a probability $1 - p_i$ the station defers until the next slot. The process is repeated until the packet has been transmitted. If the station initially senses the medium busy, it waits until the next slot and applies the above procedure. One simple priority-based variation of the P_i -persistent protocol is to assign distinct persistent probabilities to different priorities [25]; a station with a higher-priority packet persists to transmit this packet with a higher probability. Under light traffic, the average medium access delay of this protocol is known to be smaller than that of many other protocols. However, increasing packet collisions degrade system performance as traffic gets heavier. Moreover, a packet could get delayed indefinitely, because of the large standard deviation in delay over a wide range of traffic. We will examine the impact of the unbounded access delay of this protocol on time-critical applications.

3. Access Delay Distributions for the Single-Buffer Case

As mentioned in Section 2, the key step in deriving both P_{dyn} and T_ε is to derive the distribution of access delay. In this section, we will derive this distribution using a single-buffer system model. We will in Section 4 extend this model to the multiple-buffer case.

3.1. Token Passing Protocol

In order to derive the distribution of access delay for the token passing protocol, we must first derive another quantity of interest, called the *cycle time*. The cycle time is defined as the time required for the token to cycle through all stations once, transmitting all packets, if any, in each station buffer at the time the station captures the token.

3.1.1. Definition and Transition of States

The first step of our derivation is to define an embedded Markov chain describing the system. Define the state as $\vec{n} = (n_1, n_2, \dots, n_N)$ where n_i ($1 \leq i \leq N$) is the station buffer occupancy *seen by the token as it cycles through all stations*, or

$$n_i = \begin{cases} 0 & \text{if the buffer is empty as the token comes to station } i \\ 1 & \text{if the buffer is full as the token comes to station } i. \end{cases}$$

The token is assumed to start cycling from a particular station. For a homogeneous system, it does not matter which station is chosen as the starting point. Moreover, we are interested in the state of the system not at some particular time instant, but as the token travels through the stations. Whenever the token comes to station i , it records as the i -th component of current state depending on whether or not station i has a packet to transmit. As will become clear, with this state definition one can easily obtain the distribution of cycle time once the steady-state probabilities are computed.

State transition can be determined by examining the activities at each station in isolation, since the arrival processes at different stations are assumed to be independent. Specifically, for station i we have

$n_i \rightarrow n'_i$	transition probability
$0 \rightarrow 0$	$e^{-\lambda_i(NW+Sm)}$
$0 \rightarrow 1$	$1 - e^{-\lambda_i(NW+Sm)}$
$1 \rightarrow 0$	$e^{-\lambda_i(NW+Sm)}$
$1 \rightarrow 1$	$1 - e^{-\lambda_i(NW+Sm)}$

where n'_i is the next state of station i . W is the token walk time from station to station, which consists of two components: the propagation and transmission delay from station to station, and the bit delay (which is associated with the station's relay delay and latency)

buffer delay) for a station, and m is the time required for a packet to be transmitted. Under the assumption¹ that the stations with a full buffer are uniformly distributed on the ring over the long run, $S = \sum_{i=1}^N n_i$ represents the number of stations with outstanding packets, and $NW + Sm$ is the time the token has been absent from station i since its last visit, i.e., the cycle time. Note that the influence of other station buffer occupancies on station i is accounted for in the parameter S . The state transition probability can thus be expressed as

$$p_{\vec{n} \rightarrow \vec{n}'} = \prod_{i=1}^N (1 - e^{-\lambda_i(NW+Sm)})^{n_i} \cdot (e^{-\lambda_i(NW+Sm)})^{1-n_i}. \quad (1)$$

Note that this is not a Bernoulli trial but a Markov chain, since $S = \sum_{i=1}^N n_i$ depends on \vec{n} .

The state space of a homogeneous system can be reduced by defining S — the number of occupied stations seen by the token as it cycles through all stations — as the state. The state transition probability then becomes

$$p_{S,S'} = \binom{N}{S'} (1 - e^{-\lambda(NW+Sm)})^{S'} \cdot (e^{-\lambda(NW+Sm)})^{N-S'}, \quad (2)$$

where $S' = \sum_{i=1}^N n_i'$.

3.1.2. Cycle Time Distribution

Let $\pi_{\vec{n}}$ (π_S) be the steady-state probability that the system is in state \vec{n} (state S). Determination of the equilibrium probability distributions, $\pi_{\vec{n}}$, for all \vec{n} (π_S for $0 \leq S \leq N$), is straightforward (e.g., see [9] for a standard technique). The distribution of cycle time is

$$P(T_c = NW + Sm) = \sum_{\vec{n}: S = \sum_{j=1}^N n_j} \pi_{\vec{n}} = \pi_S. \quad (3)$$

3.1.3. Access Time Distribution

The access delay at station i , denoted by D_{A_i} , can now be derived. The following derivation draws on the approach taken in [8]. Let A_i denote the event of a packet arriving at station i in a cycle. The joint probability of events A_i and $T_c = NW + km$ (i.e., k packets in a cycle) is:

$$P(T_c = NW + km, A_i) = P(T_c = NW + km)P(A_i | T_c = NW + km). \quad (4)$$

The probability $P(T_c = NW + km)$ is given by Eq. (3). The term $P(A_i | T_c = NW + km)$ can be expressed as

$$P(A_i | T_c = NW + km) = \frac{\lambda_i \cdot \sum_{j_1, j_2, \dots, j_{k-1} \in PW_{k-1}^{N-i}} \lambda_{j_1} \lambda_{j_2} \dots \lambda_{j_{k-1}}}{\sum_{\ell_1, \ell_2, \dots, \ell_k \in PW_k^{N-0}} \lambda_{\ell_1} \lambda_{\ell_2} \dots \lambda_{\ell_k}}, \quad (5)$$

where PW_a^{b-i} is the set of a ($< b$) integers chosen from $\{\ell : 1 \leq \ell \leq b, \ell \neq i, \ell \text{ is an integer}\}$. In case of homogeneous systems (e.g., $\lambda_i = \lambda, \forall i$)

$$P(A_i | T_c = NW + km) = \frac{C_{k-1}^{N-1}}{C_k^N} = \frac{k}{N}. \quad (6)$$

The cycle time conditioned on a packet arrival can be determined by using the above two equations and the property of total probability,

$$P(T_c = NW + km | A_i) = \frac{P(T_c = NW + km, A_i)}{\sum_{k=1}^N P(T_c = NW + km, A_i)}. \quad (7)$$

By conditioning upon event A_i and the duration of a cycle $T_c = NW + km$, and by defining the time elapsed from the beginning of the cycle to the packet's arrival as $\tau = T_2 - T_1 - D_{A_i} = NW + (k-1)m - D_{A_i}$ (see Fig. 1), we now have

$$\begin{aligned} P(D_{A_i} \leq d | T_c = NW + km, A_i) &= P(\tau \geq NW + (k-1)m - d | T_c = NW + km, A_i) \\ &= 1 - P(\tau \leq NW + (k-1)m - d | T_c = NW + km, A_i). \end{aligned} \quad (8)$$

Note that the event $\{T_c = NW + km, A_i\}$ is equivalent to $\{T_c = NW + km, \tau \leq NW + (k-1)m\}$ since the event of a new packet arrival during a cycle implies that the time elapsed from the beginning of the cycle to the packet's arrival must be at least m units of time less than the cycle time. Similarly, the event $\{T_c = NW + km, A_i, \tau \leq NW + (k-1)m - d\}$ is equivalent to $\{\tau \leq NW + (k-1)m - d, T_c = NW + km\}$, where d is the access delay. Thus, we have

$$\begin{aligned} P(D_{A_i} \leq d | T_c = NW + km, A_i) &= 1 - \frac{P(\tau \leq NW + (k-1)m - d, T_c = NW + km)}{P(\tau \leq NW + (k-1)m, T_c = NW + km)} \\ &= 1 - \frac{P(\tau \leq NW + (k-1)m - d | T_c = NW + km)}{P(\tau \leq NW + (k-1)m | T_c = NW + km)} \\ &= 1 - \frac{1 - e^{-\lambda_i[NW+(k-1)m-d]}}{1 - e^{-\lambda_i[NW+(k-1)m]}} = \frac{e^{\lambda_i d} - 1}{e^{\lambda_i[NW+(k-1)m]} - 1}. \end{aligned} \quad (9)$$

The second equality is obtained by dividing both the numerator and the denominator by $P(T_c = NW + km)$. The third equality follows from the fact that the time to the first arrival is exponentially distributed and is independent of the cycle time, i.e., $P(\tau \leq NW + (k-1)m | T_c = NW + km) = 1 - e^{-\lambda[NW+(k-1)m]}$. The unconditioned distribution of D_{A_i} is then computed by averaging over k as:

$$\begin{aligned} D_{A_i}(t) &\triangleq P(D_{A_i} \leq t | A_i) \\ &= \sum_k \frac{P(D_{A_i} \leq t, T_c = NW + km, A_i)}{P(T_c = NW + km, A_i)} \cdot \frac{P(T_c = NW + km, A_i)}{P(A_i)} \\ &= \sum_k P(D_{A_i} \leq t | T_c = NW + km, A_i) \cdot P(T_c = NW + km | A_i). \end{aligned} \quad (10)$$

Note that in the above summation, one must be careful in choosing the proper value of t , since for $t \geq NW + (k-1)m$, $P(D_{A_i} \leq t | T_c = NW + km, A_i) = 1$. Let $n(t)$

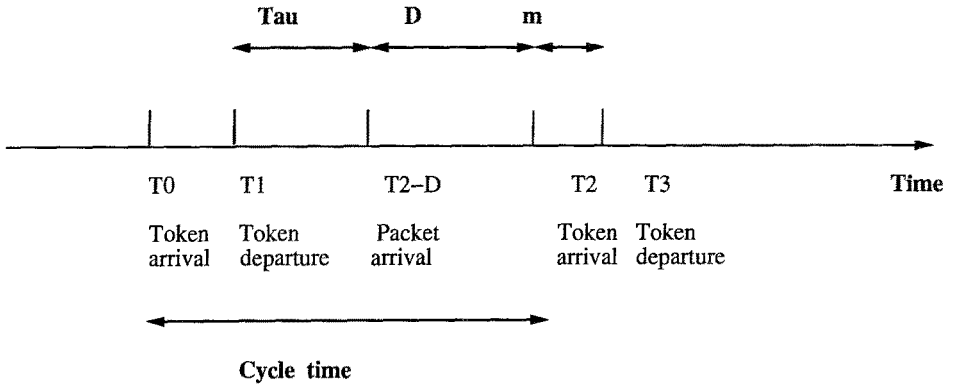


Figure 1. A timing diagram showing the access delay.

represent the maximum value of n such that $n \leq \frac{t-NW}{m} + 1$, and the above equation is reduced to

$$D_{A_i}(t) = \sum_{k=1}^{n(t)} P(T_c = NW + km | A_i) + \sum_{k=n(t)+1}^N P(D_{A_i} \leq t | T_c = NW + km, A_i) P(T_c = NW + km | A_i). \quad (11)$$

3.1.4. Probability of Packet Rejection

The probability of rejecting a packet is the same as that of a packet finding the buffer full upon its arrival. One may be tempted to express the probability of packet rejection as $Q_i^* = \sum_{\bar{n}:n_i=1} \pi_{\bar{n}}$. This is incorrect because the embedding point of the Markov chain to derive $\pi_{\bar{n}}$ is not at some arbitrary time instant, but as the token cycles through all stations. Thus, Q_i^* represents the probability of a full buffer *seen by the token as it visits station i* , rather than that *seen by a packet arriving at station i* .

To derive the probability of packet rejection, each station is considered as an M/G/1/1 queue. That is, packets arrive at station i according to a Poisson distribution with rate λ_i , and are served by a single server with the distribution of service time:

$$B_i(t) = \begin{cases} D_{A_i}(t - m) & \text{if } t \geq m \\ 0 & \text{otherwise.} \end{cases}$$

The queue-length distribution for M/G/1/1 queues has been well established (for example, see [7]), and the steady-state probability of a busy server (i.e., a full buffer in our case) at an arbitrary time instant has been shown to be $q_1^i = \frac{\lambda_i}{\lambda_i + \mu_i}$, where μ_i is the mean service rate (i.e., $\mu_i = \frac{1}{E(B_i(t))}$ in our case). Besides, since the arrival process is

Poisson, the general probability of a full buffer, q_1^i , is identical to the probability of a full buffer seen by an arriving packet at station i . Thus,

$$Q_i = q_1^i = \frac{\lambda_i E(B_i(t))}{\lambda_i E(B_i(t)) + 1}. \quad (12)$$

3.2. Token Ring Scheduling Protocol

Under this protocol, a token with a reservation field is passed around the stations on the bus. Each station examines the reservation field as the token passes through and inserts the priority of its packet (if any) if and only if its priority is higher than the one currently in the reservation field. A station is allowed to capture the token and transmit its packet only when the token returns with the station's registered priority after passing through all the other stations. Consequently, the token has to come to the station with the highest-priority packet twice before this packet can be transmitted. The overhead thus incurred in each packet transmission is the ring walk time, NW . We include this overhead in the packet transmission time, which now becomes $NW + m$.

For the purpose of modeling, one can imagine that after the token comes to the station with the highest-priority packet, rather than traveling through all other stations to find that the station does have the highest-priority packet, it sits there for NW units of time before starting transmission. Whether or not the token will stay at a station for NW units of time during its visit is characterized by the probability of a station having the highest-priority packet (to be determined in Section 3.2.5). With this characterization, the *cycle time* is defined as the time required for the token to cycle through all stations once. Note that by 'once,' we exclude the time the token visits a station for polling, since the token is now viewed as staying in the station with the highest-priority packet during polling. (In reality, however, the token may visit a station more than once in each cycle time.)

3.2.1. Definition and Transition of States

The state of the system must contain: (1) buffer occupancy at each station, (2) the priority level of an outstanding packet, if any, at each station, and (3) the number of *qualified-to-transmit* packets at a station as seen by the token. By 'qualified-to-transmit', we mean that the packet has the highest priority among all outstanding packets as the token comes to the station. The state is thus defined as $(\vec{n}, \vec{c}) = (n_1, n_2, \dots, n_N; c_1, c_2, \dots, c_N)$, where $n_i = 1(0)$ if there is one (no) qualified-to-transmit packet as seen by the token, and c_i is the priority of an outstanding packet at station i ($1 \leq c_i \leq p$) as seen by the token. $c_i \triangleq 0$ when there is no packet at station i .

The transition probabilities for station i are computed as follows:

$(n_i; c_i) \rightarrow (n'_i; c'_i)$	transition probability
$(0; 0) \rightarrow (0; 0)$	$e^{-\lambda_i C(S)}$
$(0; 0) \rightarrow (1; j)$	$\frac{\lambda_{ij}}{\lambda_i} (1 - e^{-\lambda_i C(S)}) p_{ij}$
$(0; 0) \rightarrow (0; j)$	$\frac{\lambda_{ij}}{\lambda_i} (1 - e^{-\lambda_i C(S)}) (1 - p_{ij})$
$(0; j) \rightarrow (0; j)$	$1 - p_{ij}$
$(0; j) \rightarrow (1; j)$	p_{ij}
$(1; j) \rightarrow (0; 0)$	$e^{-\lambda_i C(S)}$
$(1; j) \rightarrow (0; k)$	$\frac{\lambda_{ik}}{\lambda_i} (1 - e^{-\lambda_i C(S)}) (1 - p_{ik})$
$(1; j) \rightarrow (1; k)$	$\frac{\lambda_{ik}}{\lambda_i} (1 - e^{-\lambda_i C(S)}) p_{ik}$

where $1 \leq j, k \leq p$, $S = \sum_{i=1}^N n_i$ is the number of qualified-to-transmit packets seen by the token as it cycles through all stations, $C(S) \triangleq NW + S(m + NW)$ is the cycle time, and p_{ij} is the probability that a priority- j packet at station i has the highest priority (as the token “cycles” through the ring), which is yet to be determined (Section 3.2.3). Derivation of these transition probabilities is straightforward; e.g., $\frac{\lambda_{ik}}{\lambda_i} (1 - e^{-\lambda_i C(S)})$ is the probability that there is an arrival of priority k within the cycle time $C(S)$.

An implicit assumption used in deriving these transition probabilities is that a priority- $k (< j)$ packet arriving at a downstream station (relative to the token’s current position) during the polling time of a priority- j packet is not allowed to abort the polling. In other words, if a priority- j packet has the highest priority among all currently outstanding packets and initiates polling, it will surely gain medium access at the end of polling, regardless whether or not there will be subsequent arrivals of higher-priority packets within NW . This approximation does not sway the analytic results from reality when $NW \ll m$ in which case the probability of new higher-priority packet arrivals within NW is usually negligible. When $NW \approx m$, the results so derived for high-priority packets are worse than the simulations results, and may serve as a lower bound of performance. The state transition probabilities can be expressed in a way similar to the token passing protocol (Eq. (3.1)).

The state space can be reduced by using the homogeneity assumption. For example, if there are only two priority levels² at station i with arrival rates λ_{i1} and λ_{i2} , respectively, we may define the state as $(s_1; s_2, t_2)$, where s_1 , s_2 , and t_2 are the number of qualified-to-transmit high-priority packets, the number of qualified-to-transmit low-priority packets, and the number of outstanding low-priority packets, all of which are seen by the token as it “cycles” through all stations.³ Note that with this definition of state, we have $0 \leq s_2 \leq t_2 \leq N$, $0 \leq s_1 + t_2 \leq N$, and the size of the state space is now $\frac{(N+1)(N+2)(N+3)}{6}$.

Let $q_1(S) = \frac{\lambda_{i1}}{\lambda_i} (1 - e^{-\lambda_i C(S)})$ ($q_2(S) = \frac{\lambda_{i2}}{\lambda_i} (1 - e^{-\lambda_i C(S)})$) denote the probability that a high-priority (low-priority) packet arrives within a cycle time of $C(S)$, and $p_2 \triangleq p_{i2} \forall i$, then the state transition probability can be expressed, under the assumption of statistically identical behavior of all stations, as:

$$P_{(s_1; s_2, t_2) \rightarrow (s'_1; s'_2, t'_2)} =$$

$$\left\{ \frac{(N - (t_2 - s_2))!}{s_1'! [t_2 - (t_2 - s_2)]! (N - s_1' - t_2')!} \cdot [q_1(s_1 + s_2)]^{s_1'} \cdot [q_2(s_1 + s_2)]^{t_2' - (t_2 - s_2)} \cdot [1 - q_1(s_1 + s_2) - q_2(s_1 + s_2)]^{N - s_1' - t_2'} \right\} \cdot \binom{t_2'}{s_2'} p_2^{s_2'} (1 - p_2)^{t_2' - s_2'}.$$

Note that the term inside the braces is the probability of s_1' new arrivals of high-priority packets and $t_2' - (t_2 - s_2) \geq 0$ new arrivals of low-priority packets⁴ among $N - (t_2 - s_2)$ stations with no outstanding packets. The term after the braces is the probability that s_2' out of t_2' low-priority packets find no high-priority packets waiting in the system as the token comes to their stations.

3.2.2. Cycle Time Distribution

A standard technique can be applied to obtain equilibrium probability distributions $(\pi(\vec{n}, \vec{c}))$ or $\pi(s_1; s_2, t_2)$. The probability density function of T_c can be written as

$$P(T_c = NW + S(m + NW)) = \sum_{(\vec{n}, \vec{c}): \sum_{j=1}^N n_j = S} \pi(\vec{n}, \vec{c}), \quad (13)$$

or

$$P(T_c = NW + S(m + NW)) = \sum_{(s_1; s_2, t_2): s_1 + s_2 = S} \pi(s_1; s_2, t_2) \quad (14)$$

3.2.3. Access Delay Distribution

To derive the distribution of the access delay, we follow the same approach as in the token passing protocol except that (1) $NW + km$ has to be replaced by $NW + k(m + NW)$, (2) the counterpart of $P(A_i | T_c = NW + km)$ in Eq. (5), i.e., $P(A_i | T_c = NW + k(m + NW))$, must be approximated, since given T_c (or, equivalently k) there is no information about the number of packets that have arrived during the cycle time.⁵ Two cases are considered: (1) a packet of priority j arrives and turns out to have the highest priority among all outstanding packets as the token comes to it; (2) a packet of priority j arrives and does not have the highest priority as the token comes to it.

Let $A_i^{<j>}$ denote the event of a priority- j packet arriving at station i in a cycle. Then in the first case,

$$P(A_i^{<j>} | T_c = NW + k(m + NW)) = \frac{\lambda_{ij}}{\lambda_i} (1 - e^{-(NW + (k-1)(m + NW))\lambda_i}),$$

since given that the priority- j packet arriving during this cycle was transmitted upon the token's first visit, it must have arrived in last $NW + (k-1)(m + NW)$ units of time (see Fig. 1). Let $D_{A_{i1}}^{<j>}$ denote the time between the arrival of such a packet and the token's visit. Then, all the subsequent derivation for $P(D_{A_{i1}}^{<j>} \leq t)$ is the same as that for $P(D_{A_i} \leq t)$ in the token passing protocol.

In the second case, a priority- j packet arriving at a station in one cycle is not transmitted when the token visits the station (i.e., it does not contribute $m + NW$ units of time to the cycle time), and thus could join the system in $NW + k(m + NW)$ units of time. In other words,

$$P(A_i^{<j>} | T_c = NW + k(m + NW)) = \frac{\lambda_{ij}}{\lambda_i} (1 - e^{-(NW+k(m+NW))\lambda_i}).$$

Furthermore, for the second case the event $\{T_c = NW + k(m + NW), A_i^{<j>}\}$ is equivalent to $\{T_c = NW + k(m + NW), \tau \leq NW + k(m + NW)\}$ and the event $\{T_c = NW + k(NW + m), A_i^{<j>}, \tau \leq NW + k(m + NW) - d\}$ is equivalent to $\{\tau \leq NW + k(m + NW) - d, T_c = NW + k(m + NW)\}$. Consequently, if $D_{A_{i2}}^{<j>}$ denotes the time between the arrival of such a packet and the token's first visit since this arrival, we have

$$\begin{aligned} & P(D_{A_{i2}}^{<j>} \leq d | T_c = NW + k(m + NW), A_i) \\ &= 1 - \frac{P(\tau \leq NW + k(m + NW) - d, T_c = NW + k(m + NW))}{P(\tau \leq NW + k(m + NW), T_c = NW + k(m + NW))} \\ &= 1 - \frac{\frac{\lambda_{ij}}{\lambda_i} (1 - e^{-\lambda_i[NW+k(m+NW)-d]})}{\frac{\lambda_{ij}}{\lambda_i} (1 - e^{-\lambda_i[NW+k(m+NW)]})} = \frac{e^{\lambda_i d} - 1}{e^{\lambda_i[NW+k(m+NW)]} - 1}. \end{aligned}$$

All the subsequent derivation for $P(D_{A_{i2}}^{<j>} \leq t)$ is similar to that for $P(D_{A_i} \leq t)$ in the case of token passing protocol.

The distribution of the access delay of a priority- j packet, $P(D_{A_i}^{<j>} \leq t)$, can be approximately derived by considering whether or not the arriving packet of priority j has the highest priority among all outstanding packets (and is thus transmitted upon the token's visit). That is,

$$P(D_{A_i}^{<j>} \leq t) = p_{ij} P(D_{A_{i1}}^{<j>} \leq t) + \sum_{k=1}^{\infty} (1 - p_{ij})^k p_{ij} P(D_{A_{i2}}^{<j>} \leq t) * P^{(k)}(T_c \leq t),$$

where $P^{(k)}(T_c \leq t)$ denotes the distribution of the $(k - 1)$ -fold convolution of the density function of T_c , and the symbol $*$ denotes the convolution performed on the density functions of the two probability distributions of interest.

3.2.4. Probability of Packet Rejection

To derive the probability of packet rejection, each station is (again) modeled as an M/G/1/1 queue with the service time distribution

$$B_i(t) = \begin{cases} \sum_{j=1}^p \frac{\lambda_{ij}}{\lambda_i} D_{A_i}^{<j>}(t - (m + NW)) & \text{if } t \geq m + NW \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

Then, using the same arguments as in the token passing protocol (Section 3.1.4), Q_i can be shown to be $Q_i = \frac{\lambda_i E(B_i(t))}{\lambda_i E(B_i(t)) + 1}$.

3.2.5. *Determination of p_{ij}*

To derive p_{ij} — the probability that a priority- j packet in station i has the highest priority among all outstanding packets — we have to construct another Markov chain embedded in system evolution. Define the time interval between two successive packet transmissions as *period*, and examine the system state at the end of each period. The state is defined as $\vec{r} = (r_1, r_2, \dots, r_N)$ where r_i is the priority of the outstanding packet at station i . $1 \leq r_i \leq p$, and $r_i \triangleq 0$ if there is no outstanding packet at station i .

Note that each period consists of the following three subintervals: (1) polling interval which equals NW , and the time for the token to poll all the other stations before returning to the station with the highest-priority packet; (2) transmission interval which follows the polling interval and equals m . (3) idle interval which is the time interval between the end of the transmission interval and the beginning of the next polling interval.

There will be a nonzero idle interval if the highest-priority packet is not located at the immediate neighbor station in the downstream, and can be expressed as $K_{idle}W$, where K_{idle} is a positive integer random variable. The derivation of the exact distribution of K_{idle} depends on (i) when and where a packet will arrive if there is no packet waiting in the system after a transmission interval, or (ii) the location of the highest-priority packet if there is at least one packet waiting in the system, or (iii) whether or not a higher-priority packet will arrive during the polling initiated by a station holding a packet, and thus terminate the on-going polling. Moreover, if this occurs, the derivation also depends on the location of this newly arriving packet.

Dynamically-changing conditions with the state of the system make it very difficult to derive the exact distribution of K_{idle} . So, we approximate this distribution as follows. Except for those systems with very light traffic, it is reasonable to assume the existence of at least one outstanding packet in the system, thus excluding the possibility of condition (i). When $NW \ll m$, or, when there are not many priority levels, the effect of condition (iii) on K_{idle} is minimal, and thus we assume that a polling, once it begins, is never aborted. Hence, $P(K_{idle} \leq N) \approx 1$ will hold for all but those systems with very light traffic. Moreover, under the assumption of the statistically identical behavior of all stations, each station in the downstream (relative to the token's current position) is equally likely to have the highest-priority packet. Thus, one can approximate the distribution of K_{idle} to be uniformly distributed over $[1, N]$. (The data gathered from our simulations validate this approximate distribution.) The length of each period is thus equal to $m + NW + K_{idle}W$.

Now, define $\mu_i(\vec{r}) = 0$ (1) if $r_i > (=) \min(r_1, \dots, r_N)$ as the indicator variable that station i holds a high-priority packet, and let $\nu(\vec{r}) = \sum_{i=1}^N \mu_i(\vec{r})$ be the number of stations with currently highest-priority packets, then the transition probabilities for station i can be approximated as:

$r_i \rightarrow r'_i$	transition probability
$j \rightarrow j$	$1 - \frac{1}{\nu(\vec{r})} \mu_i(\vec{r})$
$j \rightarrow 0$	$\frac{1}{\nu(\vec{r})} \mu_i(\vec{r})$
$0 \rightarrow 0$	$E(e^{\lambda_i(m + NW + K_{idle}W)})$
$0 \rightarrow j$	$\frac{\lambda_{ij}}{\lambda_i} (1 - E(e^{\lambda_i(m + NW + K_{idle}W)}))$

where $1 \leq j \leq p$, and the expectation $E(\cdot)$ is taken with respect to K_{idle} . Note that for ease of analysis, we exclude, in the calculation of transition probability $0 \rightarrow 0$, the possibility that a packet arriving at an idle station will find no other outstanding packet with a priority higher than itself and will thus get transmitted immediately upon its arrival. This possibility is unlikely to exist for all but the case of very light traffic, and the resulting discrepancy is shown (through simulations) to be tolerably small.

The state transition probabilities can be obtained using the standard technique. Let $\phi_{\vec{r}}$ be the equilibrium probability that the system is in state \vec{r} . Then,

$$p_{ij} = \frac{\sum_{\vec{r}: r_i=j, \text{ and } r_i=\min(r_1, r_2, \dots, r_N)} \phi_{\vec{r}}}{\sum_{\vec{r}: r_i=j} \phi_{\vec{r}}}. \quad (16)$$

3.3. The P_i -Persistent Protocol

In the P_i -persistent protocol, time is slotted with a slot equal to the packet transmission time, m .⁶ Each station monitors the channel constantly, and persists to transmit a packet (if any) in each idle slot with probability p_i . If the station initially senses the medium busy, it waits until the next slot, and if the medium becomes idle in the next slot, the station transmits with a probability p_i or defers with a probability $1 - p_i$.

3.3.1. State Definition & Transition, and Parameter Derivation

A station's state at the end of a slot k depends only on its state at the end of slot $k - 1$ and the activity (packet arrival and/or transmission) during slot k . Thus, we may focus on each station's buffer occupancy at the end of a slot. That is, the state of station i , $1 \leq i \leq N$, is defined as n_i , where $n_i \in \{0, 1\}$ is the buffer occupancy at the end of each slot. The transition probabilities can be expressed as a function of $c_i \triangleq P(\text{slot seen by station } i \text{ is empty})$ (which will be determined below):

$n_i \rightarrow n'_i$	transition probability
$0 \rightarrow 0$	$e^{-\lambda_i m}$
$0 \rightarrow 1$	$1 - e^{-\lambda_i m}$
$1 \rightarrow 0$	$p_i c_i e^{-\lambda_i m}$
$1 \rightarrow 1$	$1 - p_i c_i e^{-\lambda_i m}$

Station i 's equilibrium probability distribution $\{\pi_{ij}, j = 0, 1\}$ can be determined easily: for $1 \leq i \leq N$,

$$\pi_{i1} = \frac{1 - e^{-\lambda_i m}}{1 - e^{-\lambda_i m} + p_i c_i e^{-\lambda_i m}}, \quad \pi_{i0} = \frac{p_i c_i e^{-\lambda_i m}}{1 - e^{-\lambda_i m} + p_i c_i e^{-\lambda_i m}}. \quad (17)$$

c_i can then be expressed as: for $1 \leq i \leq N$

$$c_i = P(\text{stations other than station } i \text{ do not transmit})$$

$$\begin{aligned}
&= \prod_{k=1, k \neq i}^N \pi_{k0} + \sum_{k=1, k \neq i}^N \pi_{k1}(1-p_k) \prod_{j=1, j \neq i, k}^N \pi_{j0} \\
&+ \sum_{k, \ell \in PW_2^{N-i}} \pi_{k1}(1-p_k)\pi_{\ell 1}(1-p_\ell) \prod_{j=1, j \neq i, k, \ell}^N \pi_{j0} + \dots + \prod_{k=1, k \neq i}^N \pi_{k1}(1-p_k) \quad (8)
\end{aligned}$$

where PW_2^{N-i} is the set of two integers chosen from $\{a : 1 \leq a \leq N, a \neq i, a \text{ is an integer}\}$. Eq. (18) together with Eq. (17) can be used to solve c_i for $1 \leq i \leq N$ ($2N$ equations with $2N$ unknowns). The assumption of system homogeneity, or $p_i = p$, $\lambda_i = \lambda$, and $\pi_{i1} = \pi_1$, $\forall i$, can simplify the expression for $c_i = c$ to:

$$c = \sum_{i=0}^{N-1} \binom{N-1}{i} \pi_1^i (1-p)^i \pi_0^{N-1-i} = (1 - \pi_1 p)^{N-1}. \quad (19)$$

One can solve Eq. (19) (along with Eq. (17)) numerically to get c . The access delay is then given by

$$D_{A_i}(t) = \sum_{k=0}^{\lfloor t/m \rfloor} (1 - c_i p_i)^k c_i p_i. \quad (20)$$

Note that arrival of a packet may not be synchronized to a slot boundary, and thus, an additional synchronization time $T_s \in (0, 1)$ should be added to the access delay. However, T_s has been omitted from our model for the following reasons: (1) $E(T_s)$'s are the same for all stations in a homogeneous system, (2) except for light traffic cases, the delay due to the contention and collisions is much larger than one slot period, and thus, omitting T_s may introduce an error of at most one slot, which is tolerably small.

The expression of the probability of packet rejection, Q_i , is the same as that in the token passing protocol (Section 3.1.4).

3.4. Priority-Based P_i -Persistent Protocol

In the simple priority-based variation of P_i -persistent protocol, each station persists to transmit a priority- j packet in a slot with probability p_{ij} . The higher the priority j , the larger the persistence probability p_{ij} . The approach taken to analyze the P_i -persistent protocol can be applied directly to this protocol except that the state of station i is now defined as r_i , where r_i is the priority of the outstanding packet, if any, at station i ($1 \leq r_i \leq p$); otherwise, $r_i = 0$. The transition probabilities for station i can be expressed as:

$r_i \rightarrow r'_i$	transition probability
$0 \rightarrow 0$	$e^{-\lambda_i m}$
$0 \rightarrow j$	$\frac{\lambda_{ij}}{\lambda_i} (1 - e^{-\lambda_i m})$
$j \rightarrow 0$	$p_{ij} c_i e^{-\lambda_i m}$
$j \rightarrow k$	$p_{ij} c_i \frac{\lambda_{ik}}{\lambda_i} (1 - e^{-\lambda_i m})$
$j \rightarrow j$	$1 - p_{ij} c_i + p_{ij} c_i \frac{\lambda_{ij}}{\lambda_i} (1 - e^{-\lambda_i m})$

where $1 \leq j, k \leq p$. The equilibrium probability distributions, $\{\pi_{ij}, 1 \leq j \leq p\}$, can be derived in terms of c_i , which can in turn be expressed as a function of π_{kj} 's, $1 \leq k \leq N, k \neq i, 1 \leq j \leq p$. Thus, one can, in principle, solve for both r_i 's and π_{ij} 's. By the assumption of system homogeneity ($p_{ij} = p_j, c_i = c$, and $\pi_{ij} = \pi_j$, for all i) and the assumption of two-priority levels⁷ (each having arrival rate λ_{i1} and λ_{i2} at a station), we have

$$\begin{aligned}\pi_1 &= \frac{\frac{\lambda_{i1}}{\lambda_i} c p_2 (1 - e^{-\lambda m})}{p_1 p_2 c^2 e^{-\lambda_i m} + c \left(\frac{\lambda_{i1}}{\lambda_i} p_2 + \frac{\lambda_{i2}}{\lambda_i} p_1 \right) (1 - e^{-\lambda_i m})}, \\ \pi_2 &= \frac{\frac{\lambda_{i2}}{\lambda_i} c p_1 (1 - e^{-\lambda_i m})}{p_1 p_2 c^2 e^{-\lambda_i m} + c \left(\frac{\lambda_{i1}}{\lambda_i} p_2 + \frac{\lambda_{i2}}{\lambda_i} p_1 \right) (1 - e^{-\lambda_i m})},\end{aligned}\quad (21)$$

and

$$c = (1 - \pi_1 p_1 - \pi_2 p_2)^{N-1}. \quad (22)$$

Eq. (22) together with Eq. (21) can be solved numerically to determine c . The access delay for a priority- j packet is then given by

$$D_{A_i}^{<j>}(t) = \sum_{k=0}^{\lfloor t/m \rfloor} (1 - c_i p_{ij})^k c_i p_{ij}. \quad (23)$$

The approach to derive Q_i is the same as that for the token scheduling protocol.

4. Extension to Multiple-Buffer Case

Though the single-buffer model eases analysis and can be used to evaluate the goodness of a contention protocol, we need a multiple-buffer model to study the performance differences between the single and multiple buffer cases. To meet this need, we now extend the single-buffer model to the case of L buffers in each station.

4.1. Token Passing Protocol

4.1.1. Definition and Transition of States

The embedded Markov chain constructed for the single-buffer model is also used here to describe the system except that n_i ($0 \leq n_i \leq L$) in the state $\vec{n} = (n_1, n_2, \dots, n_N)$ is now defined as the number of packets at station i seen by the token as it cycles through all stations. The station transition probabilities can be determined by examining the activities at each station i , and are given by:

$$p_{n_i \rightarrow n'_i} = \begin{cases} 0 & \text{if } n'_i \leq n_i - 2 \\ \frac{e^{-(NW+Sm)\lambda_i} [\lambda_i(NW+Sm)]^{n'_i}}{n'_i!} & \text{if } n_i = 0 \text{ and } 0 \leq n'_i \leq L - 1 \\ 1 - \sum_{k=0}^{L-1} \frac{e^{-(NW+Sm)\lambda_i} [\lambda_i(NW+Sm)]^k}{k!} & \text{if } n_i = 0 \text{ and } n'_i = L \\ \frac{e^{-(NW+Sm)\lambda_i} [\lambda_i(NW+Sm)]^{(n'_i - n_i + 1)}}{(n'_i - n_i + 1)!} & \text{if } n_i \geq 1 \text{ and } n_i - 1 \leq n'_i \leq L - 1 \\ 1 - \sum_{k=0}^{L-n_i} \frac{e^{-(NW+Sm)\lambda_i} [\lambda_i(NW+Sm)]^k}{k!} & \text{if } n_i \geq 1 \text{ and } n'_i = L. \end{cases} \quad (24)$$

Under the assumption that the stations with a full buffer are uniformly distributed on the ring over the long run, $S = \sum_{j=1}^N (1 - \delta(n_j))$ denotes the number of stations with outstanding packets, where $\delta(n_j) = 1$, for $n_j = 0$; 0, otherwise. The dependence of S on \vec{n} accounts for the influence of other stations' buffer occupancy on station i 's state transition.

Under the system homogeneity assumption, states can be aggregated, thus reducing the state space in a way similar to (but more complex than) the single-buffer model. Determination of the equilibrium probabilities, $\pi_{\vec{n}}$, is straightforward, and the distribution of cycle time can be expressed as

$$P(T_c = NW + Sm) = \sum_{\vec{n}: S = \sum_{j=1}^N (1 - \delta(n_j))} \pi_{\vec{n}}.$$

4.1.2. Queue Length Distribution

Having obtained the distribution of cycle time, we are now in a position to derive the distribution of queue length which will then be used to derive Q_i and $D_{A_i}(t)$. Each station i can be considered as an M/G/1/L queue, where the service time has the probability density function, $b_i(t) = T_c(t)$, and $T_c(t)$ is the probability density function of cycle time. The expression for $b_i(t)$ is exact when there is at least one outstanding packet at station i , but is only approximate when there is no outstanding packet at station i , since a packet arriving at an empty station i has the service time cumulative distribution $B_i(t) = D_{A_{i1}}(t - m)$, for $t \geq m$; 0, otherwise, where $D_{A_{i1}}$ is derived in a way similar to that in Section 3.2.3.

The length distribution of the M/G/1/L queue, $\{q(j), 0 \leq j \leq L\}$, can be expressed in terms of the length distribution of the corresponding M/G/1/ ∞ queue (for example, see [10]). That is, if $\{q(j), 0 \leq j \leq L\}$ and $\{q^*(j), j \geq 0\}$ represent the length distribution of the M/G/1/L queue and the corresponding M/G/1/ ∞ queue, respectively, then the following relation holds:

$$q(j) = fq^*(j), \text{ for } 0 \leq j \leq L - 1, \text{ and } q(L) = 1 - \frac{1 - fq^*(0)}{\lambda E(B(t))},$$

where $f = 1/(q^*(0) + \lambda E(B(t)) \sum_{j=0}^{L-1} q^*(j))$, $E(B(t))$ is the mean service time, and $\{q^*(j), j \geq 0\}$ can be obtained from the well-known Pollaczek-Khinchin transform equation [7], [10].

4.1.3. Probability of Packet Rejection and Distribution of Access Delay

Given the probability distribution, $\{q^i(j), 0 \leq j \leq L\}$, that there are j packets at station i , the probability of packet rejection can be expressed as $Q_i = q^i(L)$. The distribution of access delay can be approximately expressed as

$$P(D_{A_i} \leq t) = q^i(0)P(D_{A_{i1}} \leq t) + \sum_{k=1}^{L-1} q^i(k)P(D_{A_{i2}} \leq t) * P^{(k)}(T_c \leq t), \quad (25)$$

where $P^{(k)}(T_c \leq t)$ indicates the distribution of the k -fold convolution of the density function of T_c , the symbol $*$ denotes the convolution performed on the density functions of two probability distributions of interest, and $D_{A_{i1}}(t)$ and $D_{A_{i2}}(t)$ are derived in a similar way⁸ as in Section 3.2.3.

4.2. Token Scheduling Protocol

4.2.1. Definition and Transition of States

The state of station i is defined as $(n_{i1}; n_{i2}, n_{i3})$, where n_{i1} is the number of qualified-to-transmit high-priority packets, $n_{i2} = 1$ (0) indicates the event that a low-priority packet is (not) qualified to transmit, and n_{i3} is the number of outstanding low-priority packets, all of which are seen by the token as it comes to station i . Here we again adopt the imaginary scenario in Section 3.2 to characterize/define the cycle time. State transition probabilities can be determined in a way similar to (but more complex than) the single-buffer model, and are given in Table 1. Note that $S = \sum_{j=1}^N [(1 - \delta(n_{j1})) + n_{j2}]$ in Table 1 is the number of qualified-to-transmit packets seen by the token as it “cycles” through all stations. The distribution of cycle time can then be expressed in terms of the equilibrium probabilities, $\pi(\vec{n}_1; \vec{n}_2, \vec{n}_3)$:

$$P(T_c = NW + S(m + NW)) = \sum_{(\vec{n}_1; \vec{n}_2, \vec{n}_3): S = \sum_{j=1}^N [(1 - \delta(n_{j1})) + n_{j2}]} \pi(\vec{n}_1; \vec{n}_2, \vec{n}_3).$$

Table 1. State transition probabilities for the token scheduling protocol. $C(s) \triangleq NW + s(m + NW)$ is the cycle time given that there are s packets transmitted during the cycle, $p_{i2} = P(\text{no high-priority packet waiting in the system as the token comes to station } i)$, and $S = \sum_{j=1}^N [(1 - \delta(n_{j1})) + n_{j2}]$.

$(n_{i1}; n_{i2}, n_{i3})$ $\rightarrow (n'_{i1}; n'_{i2}, n'_{i3})$	Transition probability for station i
$(n_{i1}; 0, n_{i3})$ $\rightarrow (0; 1, n'_{i3})$	$e^{-\lambda_{i1}C(s)} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!} \cdot p_{i2},$ $0 \leq n_{i1} \leq 1, \max(1, n_{i3}) \leq n'_{i3} \leq L_2 - 1$
	$e^{-\lambda_{i1}C(s)} \cdot [1 - \sum_{n'_{i3}=n_{i3}}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}] p_{i2},$ $0 \leq n_{i1} \leq 1, n'_{i3} = L_2$
$(n_{i1}; 0, n_{i3})$ $\rightarrow (n'_{i1}; 0, n'_{i3})$	$e^{-\lambda_{i1}C(s)} \cdot e^{-\lambda_{i2}C(s)},$ $0 \leq n_{i1} \leq 1, n'_{i1} = n_{i3} = n'_{i3} = 0$
	$e^{-\lambda_{i1}C(s)} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!} (1 - p_{i2}),$ $0 \leq n_{i1} \leq 1, n'_{i1} = 0, \max(1, n_{i3}) \leq n'_{i3} \leq L_2 - 1$
	$e^{-\lambda_{i1}C(s)} \cdot [1 - \sum_{n'_{i3}=n_{i3}}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}] (1 - p_{i2}),$ $0 \leq n_{i1} \leq 1, n'_{i1} = 0, n'_{i3} = L_2$
	$\frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!},$ $n_{i1} = 0, 1 \leq n'_{i1} \leq L_1 - 1, n_{i3} \leq n'_{i3} \leq L_2 - 1$
	$\frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!} \cdot [1 - \sum_{n'_{i3}=n_{i3}}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}] ,$ $n_{i1} = 0, 1 \leq n'_{i1} \leq L_1 - 1, n'_{i3} = L_2$
	$[1 - \sum_{n'_{i1}=0}^{L_1-1} \frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!}] \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!},$ $n_{i1} = 0, n'_{i1} = L_1, n_{i3} \leq n'_{i3} \leq L_2 - 1$
	$[1 - \sum_{n'_{i1}=0}^{L_1-1} \frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!}] \cdot [1 - \sum_{n'_{i3}=n_{i3}}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}] ,$ $n_{i1} = 0, n'_{i1} = L_1, n'_{i3} = L_2$
	$\frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}-n_{i1}+1}}{(n'_{i1}-n_{i1}+1)!} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{n'_{i3}-n_{i3}}}{(n'_{i3}-n_{i3})!},$ $n_{i1} \geq 1, \max(n_{i1} - 1, 1) \leq n'_{i1} \leq L_1 - 1, n_{i3} \leq n'_{i3} \leq L_2 - 1$
	$\frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{(n'_{i1}-n_{i1}+1)}}{(n'_{i1}-n_{i1}+1)!} \cdot [1 - \sum_{n'_{i3}=n_{i3}}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}] ,$ $n_{i1} \geq 1, \max(n_{i1} - 1, 1) \leq n'_{i1} \leq L_1 - 1, n'_{i3} = L_2$
	$[1 - \sum_{n'_{i1}=n_{i1}-1}^{L_1-1} \frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{(n'_{i1}-n_{i1}+1)}}{(n'_{i1}-n_{i1}+1)!}] \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!},$ $n_{i1} \geq 1, n'_{i1} = L_1, n_{i3} \leq n'_{i3} \leq L_2 - 1$
	$[1 - \sum_{n'_{i1}=n_{i1}-1}^{L_1-1} \frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{(n'_{i1}-n_{i1}+1)}}{(n'_{i1}-n_{i1}+1)!}] \cdot [1 - \sum_{n'_{i3}=n_{i3}}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}] ,$ $n_{i1} \geq 1, n'_{i1} = L_1, n'_{i3} = L_2$

On the other hand, the key parameter used in the expression of state transition probabilities, $p_{i2} = P(\text{no outstanding high-priority packets in the system})$, can be derived similarly as in Section 3.2.5. That is, we define $\bar{n}_1 = (n_{11}, \dots, n_{N1})$ as the state, where

Table I. State transition probabilities for the token scheduling protocol (continued).

$(n_{i1}; n_{i2}, n_{i3})$ $\rightarrow (n'_{i1}; n'_{i2}, n'_{i3})$	Transition probability for station i
$(0; 1, n_{i3})$ $\rightarrow (0; 1, n'_{i3})$	$e^{-\lambda_{i1}C(s)} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3}+1)}}{(n'_{i3}-n_{i3}+1)!} p_{i2},$ $\max(n_{i3} - 1, 1) \leq n'_{i3} \leq L_2 - 1$
	$e^{-\lambda_{i1}C(s)} \cdot [1 - \sum_{n'_{i3}=n_{i3}-1}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3}+1)}}{(n'_{i3}-n_{i3}+1)!}] p_{i2},$ $n'_{i3} = L_2$
$(0; 1, n_{i3})$ $\rightarrow (n'_{i1}; 0, n'_{i3})$	$e^{-\lambda_{i1}C(s)} \cdot e^{-\lambda_{i2}C(s)},$ $n'_{i1} = 0, n_{i3} = 1, n'_{i3} = 0$
	$e^{-\lambda_{i1}C(s)} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3}+1)}}{(n'_{i3}-n_{i3}+1)!} (1 - p_{i2}),$ $n'_{i1} = 0, \max(n_{i3} - 1, 1) \leq n'_{i3} \leq L_2 - 1$
	$e^{-\lambda_{i1}C(s)} \cdot [1 - \sum_{n'_{i3}=n_{i3}-1}^{L_2-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3}+1)}}{(n'_{i3}-n_{i3}+1)!}] (1 - p_{i2}),$ $n'_{i1} = 0, n'_{i3} = L_2$
	$\frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!} \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!},$ $1 \leq n'_{i1} \leq L_1 - 1, n_{i3} \leq n'_{i3} \leq L_2 - 1$
	$\frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!} \cdot [1 - \sum_{n'_{i1}=0}^{L_1-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}],$ $1 \leq n'_{i1} \leq L_1 - 1, n'_{i3} = L_2$
	$[1 - \sum_{n'_{i1}=0}^{L_1-1} \frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!}] \cdot \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!},$ $n'_{i1} = L_1, n_{i3} \leq n'_{i3} \leq L_2 - 1$
	$[1 - \sum_{n'_{i1}=0}^{L_1-1} \frac{e^{-\lambda_{i1}C(s)} (\lambda_{i1}C(s))^{n'_{i1}}}{n'_{i1}!}] \cdot [1 - \sum_{n'_{i1}=0}^{L_1-1} \frac{e^{-\lambda_{i2}C(s)} (\lambda_{i2}C(s))^{(n'_{i3}-n_{i3})}}{(n'_{i3}-n_{i3})!}],$ $n'_{i1} = L_1, n'_{i3} = L_2$

n_{j1} is the number of high-priority packets at the end of each period, and the period is defined as the time interval between two successive packet transmissions.⁹ The state transition probabilities for station i can then be approximated as:

$$p_{n_{i1} \rightarrow n'_{i1}} = \begin{cases} E\left(\frac{e^{-\lambda_{i1}(m+NW+K_{idle}W)} (\lambda_{i1}(m+NW+K_{idle}W))^{n'_{i1}}}{n'_{i1}!}\right) \\ \quad n_{i1} = 0, 0 \leq n'_{i1} \leq L_1 - 1 \\ 1 - \sum_{n'_{i1}=0}^{L_1-1} E\left(\frac{e^{-\lambda_{i1}(m+NW+K_{idle}W)} (\lambda_{i1}(m+NW+K_{idle}W))^{n'_{i1}}}{n'_{i1}!}\right) \\ \quad n_{i1} = 0, n'_{i1} = L_1 \\ \frac{1}{\nu(\bar{n}_1)} E(e^{-\lambda_{i1}(m+NW+K_{idle}W)}) \\ \quad n_{i1} \geq 1, n'_{i1} = n_{i1} - 1 \\ (1 - \frac{1}{\nu(\bar{n}_1)}) \cdot E\left(\frac{e^{-\lambda_{i1}(m+NW+K_{idle}W)} (\lambda_{i1}(m+NW+K_{idle}W))^{n'_{i1}-n_{i1}}}{(n'_{i1}-n_{i1})!}\right) \\ \quad + \frac{1}{\nu(\bar{n}_1)} \cdot E\left(\frac{e^{-\lambda_{i1}(m+NW+K_{idle}W)} (\lambda_{i1}(m+NW+K_{idle}W))^{n'_{i1}-n_{i1}+1}}{(n'_{i1}-n_{i1}+1)!}\right) \\ \quad n_{i1} \geq 1, n_{i1} \leq n'_{i1} \leq L_1 - 1 \\ 1 - p_{n_{i1} \rightarrow (n_{i1}-1)} - \sum_{n'_{i1}=n_{i1}}^{L_1-1} p_{n_{i1} \rightarrow n'_{i1}} \\ \quad n_{i1} \geq 1, n'_{i1} = L_1 \end{cases}$$

where $\nu(\bar{n}_1) \triangleq \sum_{i=1}^N (1 - \delta(n_{i1}))$ is the number of stations with high-priority packets. Whether a high-priority packet at station i will be transmitted next or not depends on the location of station i in the downstream (relative to the token's current position). Due to the statistically identical behavior of all stations in a homogeneous system, the probability that a high-priority packet at a station gets transmitted in the next transmission interval is approximated as $\frac{1}{\nu(\bar{n})}$.

Let $\phi_{\bar{n}_1}$ be the equilibrium probability that the system is in state $\phi_{\bar{n}_1}$, then

$$p_{i2} = \phi_{\bar{0}} = P(\text{no outstanding high-priority packets at the end of a period}).$$

4.2.2. Queue Length Distribution

Given the cycle time distribution, the queue length distribution of high-priority packets, $\{q_1^i(j), 0 \leq j \leq L_1\}$, can be obtained by considering the queue as an M/G/1/ L_1 queue with the service time density function $b_{i1}(t) = T_c(t)$. (Again, this expression is approximate for a high-priority packet which arrives at an empty station.) The queue length distribution for low-priority packets, $\{q_2^i(j), 0 \leq j \leq L_2\}$, can be obtained with the same approach except that the service time cumulative distribution $B_{i2}(t)$ is now approximated as:

$$\begin{aligned} B_{i2}(t) = & q_1^i(0) \cdot [p_{i2} D_{A_{i1}}^{<2>}(t-m) + \sum_{k=1}^{\infty} (1-p_{i2})^k p_{i2} D_{A_{i2}}^{<2>} * CT_c^{(k)}(t-m)] \\ & + \sum_{j=1}^{L_1} q_1^i(j) \cdot [\sum_{k=0}^{\infty} (1-p_{i2})^k p_{i2} D_{A_{i2}}^{<2>} * CT_c^{(k+j)}(t-m)], \end{aligned}$$

for $t \geq m$, where $CT_c^{(k)}(t)$ is the cumulative distribution of the k -fold convolution of the density function of T_c . $D_{A_{i1}}^{<2>}$ and $D_{A_{i2}}^{<2>}$ are derived similarly as in Sect.3.2.2. Statistically, a low-priority packet has to wait for the token's $(j+1)$ -th visit¹⁰ with probability $q_1^i(j)$ (on the average) before competing with other stations for medium access. This low-priority packet will gain medium access with probability, p_{i2} , that no high-priority packets exist in other stations.

4.2.3. Probability of Packet Rejection and Distribution of Access Delay

The probability of rejecting high-priority (low-priority) packets is $Q_i^{<1>} = q_1^i(L_1)$ ($Q_i^{<2>} = q_2^i(L_2)$). The distribution of access delay experienced by high-priority packets can be expressed as:

$$P(D_{A_i}^{<1>} \leq t) = q_1^i(0)P(D_{A_{i1}}^{<1>} \leq t) + \sum_{k=1}^{L_1-1} q_1^i(k)P(D_{A_{i2}}^{<1>} \leq t) * P^{(k)}(T_c \leq t), \quad (26)$$

and the distribution of access delay experienced by low-priority packets is approximated as

$$\begin{aligned}
 P(D_{A_i}^{<2>} \leq t) = & \\
 & q_1^i(0) \cdot \{ q_2^i(0) [p_{i2} P(D_{A_{i1}}^{<2>} \leq t) + \sum_{k=1}^{\infty} (1-p_{i2})^k p_{i2} P(D_{A_{i2}}^{<2>} \leq t) * P^{(k)}(T_c \leq t)] \\
 & + \sum_{j=1}^{L_2-1} q_2^i(j) P(D_{A_{i2}}^{<2>} \leq t) * [\sum_{k=0}^{\infty} (1-p_{i2})^k p_{i2} P^{(k)}(T_c \leq t)]^{(j+1)} \} \\
 & + \sum_{j=1}^{L_1} q_1^i(j) \cdot \{ \sum_{k=0}^{L_2-1} q_2^i(k) P(D_{A_{i2}}^{<2>} \leq t) * P^{(j)}(T_c \leq t) \\
 & * [\sum_{\ell=0}^{\infty} (1-p_{i2})^\ell p_{i2} P^{(\ell)}(T_c \leq t)]^{(k+1)} \}.
 \end{aligned}$$

4.3. P_i -Persistent Protocol

4.3.1. Definition and Transition of States

The state of station i , n_i ($0 \leq n_i \leq L$), is now defined as the number of outstanding packets at station i at the end of each slot. The transition probability can then be expressed as

$$p_{n_i \rightarrow n'_i} = \begin{cases} 0 & \text{if } n'_i \leq n_i - 2 \\ \frac{e^{-\lambda_i m} (\lambda_i m)^{n'_i}}{n'_i!} & \text{if } n_i = 0 \text{ and } 0 \leq n'_i \leq L-1 \\ 1 - \sum_{k=0}^{L-1} \frac{e^{-\lambda_i m} (\lambda_i m)^k}{k!} & \text{if } n_i = 0 \text{ and } n'_i = L \\ p_i c_i e^{-\lambda_i m} & \text{if } n_i \geq 1 \text{ and } n'_i = n_i - 1 \\ p_i c_i \frac{e^{-\lambda_i m} (\lambda_i m)^{(n'_i - n_i + 1)}}{(n'_i - n_i + 1)!} + (1 - p_i c_i) & \\ \frac{e^{-\lambda_i m} (\lambda_i m)^{(n'_i - n_i)}}{(n'_i - n_i)!} & \text{if } n_i \geq 1 \text{ and } n_i \leq n'_i \leq L-1 \\ 1 - p_i c_i \sum_{k=0}^{L-n_i} \frac{e^{-\lambda_i m} (\lambda_i m)^k}{k!} - & \\ (1 - p_i c_i) \sum_{k=0}^{L-1-n_i} \frac{e^{-\lambda_i m} (\lambda_i m)^k}{k!} & \text{if } n_i \geq 1 \text{ and } n'_i = L, \end{cases} \quad (27)$$

where c_i is the probability that a slot seen by station i is empty, and can be expressed similarly as in Eq.(18) except that π_{i1} should be replaced by $1 - \pi_{i0}$. Both c_i and station i 's equilibrium probability distribution can be obtained numerically. $D_{A_{i1}}(t)$ for packets at the head of the queue is then given by

$$D_{A_{i1}}(t) = \sum_{k=0}^{\lfloor t/m \rfloor} (1 - c_i p_i)^k c_i p_i.$$

4.3.2. Queue Length Distribution, Probability of Packet Rejection, and Distribution of Access Delay

To derive the queue length distribution, $\{q^i(j), 0 \leq j \leq L\}$, each station i is modeled as an $M/G/1/L$ queue with the service time distribution $B_i(t) = D_{A_{i1}}(t - m), t \geq m; 0$, otherwise. (Note that the expression for $B_i(t)$ here is exact.) The probability of packet rejection is then given by $Q_i = q^i(L)$. The distribution of access delay can be expressed as

$$P(D_A \leq t) = \sum_{k=0}^{L-1} q^i(k) P^{(k+1)}(D_{A_{i1}} \leq t). \quad (28)$$

4.4. Priority-Based P_i -persistent Protocol

The state of station i is now defined as (n_{i1}, n_{i2}) ($0 \leq n_{i1} \leq L_1, 0 \leq n_{i2} \leq L_2$), where n_{i1} and n_{i2} are the number of high-priority and low-priority packets waiting at station i at the end of each time slot, respectively. Expressions for the state transition probabilities are given in Table 2, where c_i is the probability that a slot seen by station i is empty, and is expressed as

$$c_i = \left(1 - \sum_{n_{i2}=0}^{L_2} \sum_{n_{i1}=1}^{L_1} \pi_{(n_{i1}, n_{i2})} p_{i1} - \sum_{n_{i2}=1}^{L_2} \pi_{(0, n_{i2})} p_{i2}\right)^{N-1}.$$

Both c_i and station i 's equilibrium probability distribution can be obtained numerically.

The length distribution of the high-priority (low-priority) packet queue, $\{q_1^i(j), 0 \leq j \leq L_1\}$, ($\{q_2^i(j), 0 \leq j \leq L_2\}$), can be obtained by considering the high-priority (low-priority) queue as an $M/G/1/L_1$ ($M/G/1/L_2$) queue with the service time distribution

$$B_{i1}(t) = D_{A_{i1}}^{<1>}(t - m), \quad \text{where } D_{A_{i1}}^{<1>}(t) = \sum_{k=0}^{\lfloor t/m \rfloor} (1 - c_i p_{i1})^k c_i p_{i1},$$

or

$$B_{i2}(t) = \sum_{k=0}^{L_1} q_1^i(k) D_{A_{i1}}^{<1>(k)} * D_{A_{i1}}^{<2>}(t - m),$$

where

$$D_{A_{i1}}^{<2>}(t) = \sum_{k=0}^{\lfloor t/m \rfloor} (1 - c_i p_{i2})^k c_i p_{i2}.$$

The probability of rejecting high-priority (low-priority) packets is then given by $Q_i^{<1>} = q_1^i(L_1)$ ($Q_i^{<2>} = q_2^i(L_2)$). The distribution of access delay experienced by a high-priority packet and a low-priority packet can be expressed as

Table 2. State transition probabilities for the priority-based P_i -persistent protocol.

$P_{(n_{i1}, n_{i2}) \rightarrow (n'_{i1}, n'_{i2})}$
$e^{-\lambda_{i1}m} p_{i2} c_i e^{-\lambda_{i2}m}, \quad n_{i1} = n'_{i1} = 0, n'_{i2} = n_{i2} - 1$
$e^{-\lambda_{i1}m} \left[p_{i2} c_i e^{-\lambda_{i2}m} \frac{(\lambda_{i2}m)^{n_{i2}-n_{i2}+1}}{(n'_{i2}-n_{i2}+1)!} + (1-p_{i2}c_i) e^{-\lambda_{i2}m} \frac{(\lambda_{i2}m)^{n_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!} \right],$ $n_{i1} = n'_{i1} = 0, n_{i2} \leq n'_{i2} \leq L_2 - 1$
$e^{-\lambda_{i1}m} \left[p_{i2} c_i \left(1 - \sum_{n'_{i2}=n_{i2}-1}^{L_2-1} \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}+1}}{(n'_{i2}-n_{i2}+1)!} \right) + (1-p_{i2}c_i) \cdot \right.$ $\left. \left(1 - \sum_{n'_{i2}=n_{i2}}^{L_2-1} \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!} \right) \right], \quad n_{i1} = n'_{i1} = 0, n'_{i2} = L_2$
$p_{i1} c_i e^{-\lambda_{i1}m} \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!},$ $n_{i1} \geq 1, n'_{i1} = n_{i1} - 1, n_{i2} \leq n'_{i2} \leq L_2 - 1$
$p_{i1} c_i e^{-\lambda_{i1}m} \left(1 - \sum_{n'_{i2}=n_{i2}}^{L_2-1} \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!} \right),$ $n_{i1} \geq 1, n'_{i1} = n_{i1} - 1, n'_{i2} = L_2$
$(p_{i1} c_i e^{-\lambda_{i1}m} \frac{(\lambda_{i1}m)^{n'_{i1}-n_{i1}+1}}{(n'_{i1}-n_{i1}+1)!} + (1-p_{i1}c_i) e^{-\lambda_{i1}m} \frac{(\lambda_{i1}m)^{n'_{i1}-n_{i1}}}{(n'_{i1}-n_{i1})!}) \cdot$ $\frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!}, \quad n_{i1} \geq 1, n_{i1} \leq n'_{i1} \leq L_1 - 1, n_{i2} \leq n'_{i2} \leq L_2 - 1$
$\left[p_{i1} c_i \left(1 - \sum_{n'_{i1}=n_{i1}-1}^{L_1-1} \frac{e^{-\lambda_{i1}m} (\lambda_{i1}m)^{n'_{i1}-n_{i1}+1}}{(n'_{i1}-n_{i1}+1)!} \right) + (1-p_{i1}c_i) \cdot \right.$ $\left. \left(1 - \sum_{n'_{i1}=n_{i1}}^{L_1-1} \frac{e^{-\lambda_{i1}m} (\lambda_{i1}m)^{n'_{i1}-n_{i1}}}{(n'_{i1}-n_{i1})!} \right) \right] \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!},$ $n_{i1} \geq 1, n'_{i1} = L_1, n_{i2} \leq n'_{i2} \leq L_2 - 1$
$\left[p_{i1} c_i e^{-\lambda_{i1}m} \frac{(\lambda_{i1}m)^{n'_{i1}-n_{i1}+1}}{(n'_{i1}-n_{i1}+1)!} + (1-p_{i1}c_i) e^{-\lambda_{i1}m} \frac{(\lambda_{i1}m)^{n'_{i1}-n_{i1}}}{(n'_{i1}-n_{i1})!} \right] \cdot$ $\left(1 - \sum_{n'_{i2}=n_{i2}}^{L_2-1} \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!} \right), \quad n_{i1} \geq 1, n_{i1} \leq n'_{i1} \leq L_1 - 1, n'_{i2} = L_2$
$\left[p_{i1} c_i \left(1 - \sum_{n'_{i1}=n_{i1}-1}^{L_1-1} \frac{e^{-\lambda_{i1}m} (\lambda_{i1}m)^{n'_{i1}-n_{i1}+1}}{(n'_{i1}-n_{i1}+1)!} \right) + (1-p_{i1}c_i) \cdot \right.$ $\left. \left(1 - \sum_{n'_{i1}=n_{i1}}^{L_1-1} \frac{e^{-\lambda_{i1}m} (\lambda_{i1}m)^{n'_{i1}-n_{i1}}}{(n'_{i1}-n_{i1})!} \right) \right] \cdot \left(1 - \sum_{n'_{i2}=n_{i2}}^{L_2-1} \frac{e^{-\lambda_{i2}m} (\lambda_{i2}m)^{n'_{i2}-n_{i2}}}{(n'_{i2}-n_{i2})!} \right),$ $n_{i1} \geq 1, n'_{i1} = L_1, n'_{i2} = L_2$

$$P(D_{A_i}^{<1>} \leq t) = \sum_{k=0}^{L_1-1} q_1^i(k) P^{(k+1)}(D_{A_{i1}} \leq t),$$

and

$$P(D_{A_i}^{<2>} \leq t) = \sum_{k=0}^{L_1} q_1^i(k) [P^{(k)}(D_{A_{i1}}^{<1>} \leq t) * \sum_{\ell=0}^{L_2-1} q_2^i(\ell) P^{(\ell+1)}(D_{A_{i2}}^{<2>} \leq t)],$$

respectively.

5. Numerical Examples

We present some numerical results to illustrate the performance characteristics of the token-type and P_i -persistent-type protocols. In the single-buffer model, since the time

each packet has to wait for other stations to transmit packets determines the quality of a contention protocol, we will analyze $D_{A_i}(t)$ (or, equivalently, $P_{dyn|d}$) and T_ϵ . In case of multiple buffers at each station, we will analyze the performance trends resulting from the combined effects of contention protocols and local buffering strategies. Emphasis is placed on the effects of varying L (or L_1 and L_2) on P_{dyn} , T_ϵ , and Q_i .

Packet transmission time (m), ring walk time (NW), and the number of stations (N) are chosen to be 1.0, 0.04, and 10, respectively. All stations are assumed to have the same packet arrival rate λ_i . Two priority levels are assumed, because the performance characteristics are found to vary little even if the number of priority levels is greater than 2. Packet laxities are uniformly distributed in $[0, 2AL]$ where AL is the average laxity. Packets with laxity $< 2rAL$ are classified as high-priority, and those with laxity $\geq 2rAL$ are classified as low-priority, where $r = \frac{\lambda_{i1}}{\lambda_i}$ is the fraction of high-priority packets.

Analytic results summarized below are obtained by solving the equation $\Pi P = \Pi$ with the Gaussian elimination method, computing cumulative probability distributions from their corresponding pdfs with Simpson's integration rule, and using FFT subroutines for time-frequency domain transformations. Event-driven simulations are also carried out for a 10-station homogeneous system, and are used to validate analytic results. For each configuration the simulation was run until we obtain results with a confidence level 95% for a maximum error of 5% of the value of measures of interest. The number of simulation runs needed to achieve the above confidence interval is based on the assumption that the parameter to be measured has a normal distribution with unknown mean and variance.

5.1. Results from Single-Buffer Model

We examine, for various traffic loads, the quality of contention protocols with $D_{A_i}(t) = 1 - P_{dyn|t}$ and T_ϵ derived from the single-buffer model.

$D_{A_i}(t)$ and $D_{A_i}^{<1>}(t)$ are plotted in Figs. 2—3 for the token passing and scheduling protocols, respectively. Table 3 and 4 give some numerical examples of $P_{dyn|t}^{<1>}$, $P_{dyn|t}^{<2>}$, and $T_\epsilon^{<1>}$ for both protocols. As was expected, $P_{dyn|t}$ and T_ϵ for the token passing protocol increase as the traffic increases. However, for all possible arrival rates, $D_{A_i}(t) = 1 - P_{dyn|t}$ always approaches 1 in $t = NW + (N - 1)m$ units of time. This is because the token will, in the worst case, re-visit a station within $NW + (N - 1)m$ units of time after its last visit to the station. T_ϵ is thus bounded by the maximum cycle time, $NW + (N - 1)m$. The simulation results agree very well with the analytic results for this protocol. (For clarity of presentation, the simulation results are only plotted for $\lambda_i = 0.1$ in Fig. 2.)

The performance of the token scheduling protocol in delivering high-priority packets is significantly improved, as compared to the token passing protocol except when the traffic is extremely light. Note, however, that this improvement was made at the expense of performance in delivering low-priority packets (Table 3). The simulation results for the token scheduling protocol slightly differ from the analytic results (as compared to

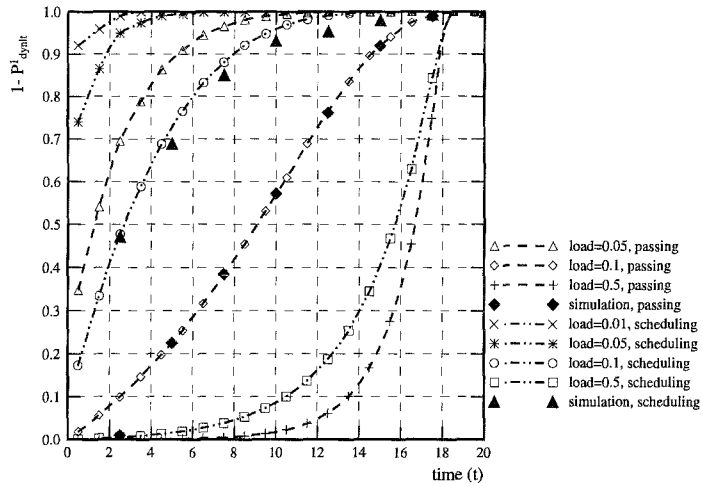


Figure 2. $1 - P_{dyn}^{<1>t}$ for token passing protocol and the token scheduling protocol. (Two packet priority levels exists with each contributing one half of the traffic.)

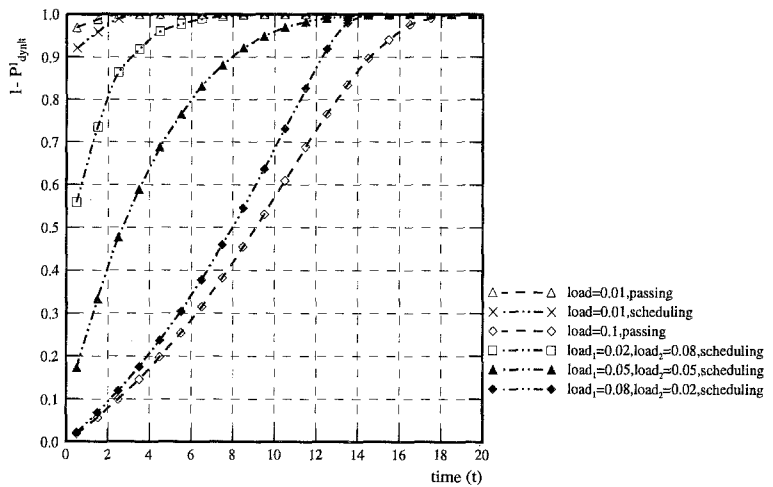


Figure 3a. Comparison of token passing protocol and token scheduling protocol with respect to P_{dyn}^1 under light and medium loads.

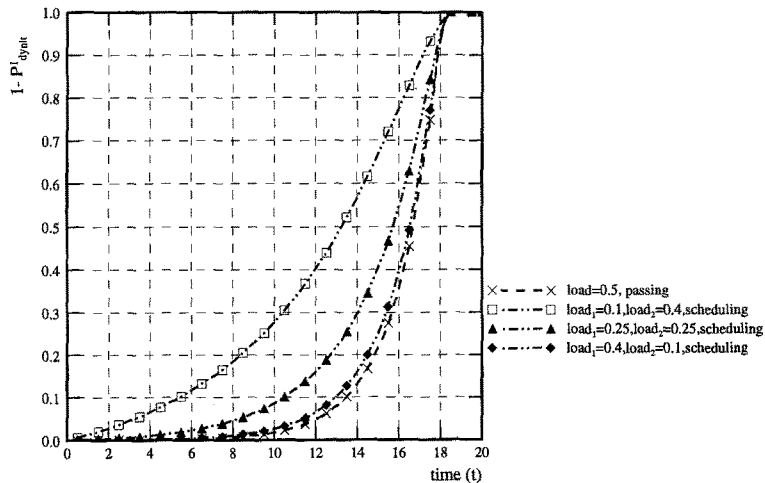


Figure 3b. Comparison of token passing protocol and token scheduling protocol with respect to $P_{dyn|t}$ under heavy loads.

Table 3. Comparison of token passing protocol and token scheduling protocol with respect to $P_{dyn|t}$.

λ_i	t	Token passing	Token scheduling	
		$P_{dyn t}$	$P_{dyn t}^{<1>}$	$P_{dyn t}^{<2>}$
0.05	1	0.5571	0.1971	0.6258
	5	0.1143	0.0076	0.5213
	9	0.0151	0.0000	0.4431
	13	0.0005	0.0000	0.4119
	17	0.0000	0.0000	0.3850
	18.5	0.0000	0.0000	0.3642
0.1	1	0.9632	0.7479	0.9721
	5	0.7749	0.2740	0.8237
	9	0.5085	0.0655	0.6436
	13	0.2008	0.0069	0.5274
	17	0.0176	0.0001	0.4898
	18.5	0.0000	0.0000	0.4277

the discrepancy for the token passing protocol) due to approximations made in their derivation.

Fig. 3 also show the effects of varying the ratio of high-priority to low-priority packets on the performance of token scheduling protocol in delivering high-priority packets. For systems with very light traffic (e.g., $\lambda_i = 0.01$), the performance is not significantly affected by this ratio, due to little contention faced by the high-priority packets in accessing

Table 4. Comparison of token passing protocol and token scheduling protocol with respect to T_ϵ . r is the ratio of high-priority to low-priority packets.

λ_i	ϵ	Token passing (T_ϵ)	Token scheduling ($T_\epsilon^{<1>}$)		
			$r = 1 : 4$	$r = 2 : 2$	$r = 4 : 1$
0.05	10^{-3}	12.37	3.89	6.20	6.27
	10^{-5}	14.05	4.45	8.00	8.53
	10^{-7}	14.18	4.79	8.23	8.76
	10^{-9}	14.21	4.98	8.59	8.92
0.1	10^{-3}	18.05	9.42	15.25	15.29
	10^{-5}	18.06	10.27	17.68	17.84
	10^{-7}	18.08	10.42	18.01	18.04
	10^{-9}	18.08	10.89	18.08	18.08
0.5	10^{-3}	18.06	17.98	18.04	18.04
	10^{-5}	18.07	18.03	18.05	18.08
	10^{-7}	18.08	18.05	18.06	18.14
	10^{-9}	18.08	18.06	18.08	18.23

the medium. (The numerical results are so close to one another for all traffic ratios that only one curve labeled as “load=0.01, scheduling” is plotted.) For systems with light to medium traffic (Fig. 3 (a), $\lambda_i = 0.1$), the performance is substantially improved as the portion of high-priority packets decreases. Not only $1 - P_{dyn|t}^{<1>}$ increases for a specific t (Fig. 3 (a)), but also $T_\epsilon^{<1>}$ becomes smaller (Table 4). For systems with heavy traffic (Fig. 3 (b), $\lambda_i = 0.5$), only $1 - P_{dyn|t}^{<1>}$ increases as the portion of high-priority traffic decreases, except when the latter is very small. This indicates that there is an upper bound on the arrival rate of high-priority packets if every high-priority packet must be delivered within its deadline which is smaller than the maximum cycle time. On the other hand, even in the worst case where every station has an outstanding high-priority packet, $T_\epsilon^{<1>}$ is bounded by the maximum cycle time, $NW + (N - 1)(m + NW)$.

Fig. 4 shows performance curves achievable with the best choice of p_i for each possible λ_i . Except under extremely light traffic (in which case $c_i p_i$ is close to 1), the distribution of access delay approaches 1 at a very slow pace (Fig. 4). This accounts for the nonzero probability of a packet being delayed indefinitely. Consequently, while this protocol is suitable for handling bursty traffic, it is not appropriate for real-time applications. The simulation results are consistent with our analytic results for this protocol.

In Fig. 5, the performance experienced by high-priority packets under the priority-based P_i -persistent protocol is compared to that under the P_i -persistent protocol for different ratios of high- and low-priority traffic. Each station persists to transmit high- and low-priority packets with the best choice of p_{i1} and p_{i2} (calculated from the analytic model) for each specified λ_i . Under very light traffic, the difference is insignificant. (The numerical results are again so close to one another for all traffic ratios that only one curve labeled as “load = 0.01” is plotted.) For systems with medium to heavy traffic (Fig. 5, $\lambda = 0.1$), the performance of the priority-based protocol is significantly better than that of the non priority-based one (labeled as “load = 0.1” in Fig. 5) if high-priority packets constitute only a small portion of the traffic. Under all circumstances, there is

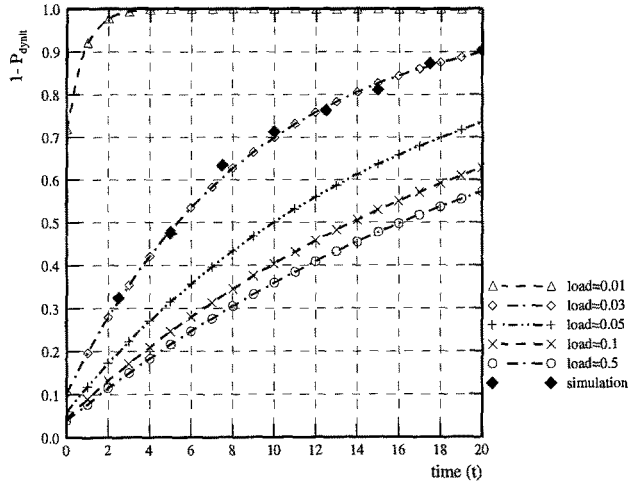


Figure 4. $1 - P_{dyn}$ for P_i -persistent protocol with the best choice of p_i for each λ_i

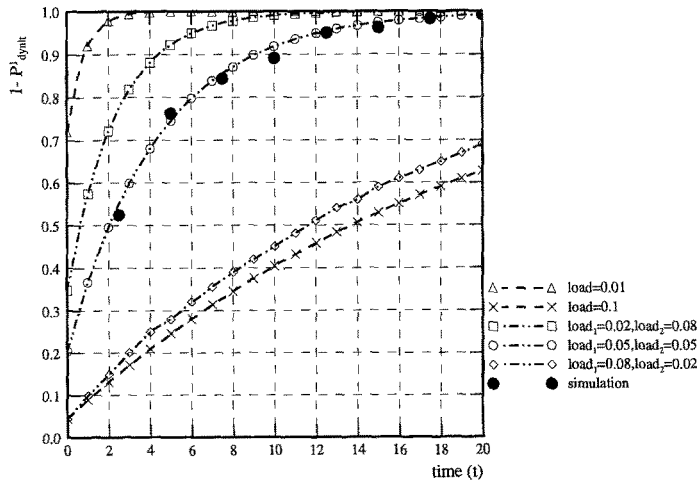


Figure 5. Comparison of P_i -persistent protocol and its priority-based variation.

a nonzero probability, $(1 - c_i p_i)^k$, that a packet will not be delivered within k units of time. When the traffic is light to medium, and high-priority packets constitute only a small portion of the traffic, this probability is negligible and $T_\epsilon^{<1>}$ with $\epsilon \leq 10^{-3}$ is acceptably small (Table 5). Consequently, the priority-based P_i -persistent protocol can be employed when high-priority packets constitute only a small portion of the traffic for *soft* real-time applications, where missing packet deadlines is undesirable but not disastrous. However, for hard real-time systems where missing deadlines is prohibited, some other mechanism which guarantees deterministic, bounded packet delivery times must be used.

Table 5. T_ϵ of P_i -persistent protocol and its priority-based variation.

λ_i	ϵ	P_i -persistent protocol (T_ϵ)	Priority P_i -persistent protocol ($T_\epsilon^{<1>}$)	
			$r = 1 : 4$	$r = 2 : 2$
0.05	10^{-3}	109.0	12.50	17.00
	10^{-5}	—	19.60	29.00
	10^{-7}	—	27.30	39.20
0.1	10^{-3}	144.2	15.50	26.70
	10^{-5}	—	25.90	49.20
	10^{-7}	—	36.10	67.00
0.5	10^{-3}	170.1	19.50	39.80
	10^{-5}	—	33.00	66.70
	10^{-7}	—	47.10	89.00

5.2. Results from Multiple-Buffer Model

To study the performance differences between the single- and multiple- buffer models, and the combined effects of contention protocols and local buffering strategies on P_{dyn} , T_ϵ and Q_i , we derived numerical examples from multiple-buffer models with $AL = 30$ and $r = 0.2$. Both the analytic and simulation results indicate that the performance for the P_i -persistent-type protocols has trends similar to that for the token-type protocols (except all the measures P_{dyn} , T_ϵ , and Q_i have larger values for the same configuration) and thus is not included.

Fig. 6 plots $P_{dyn}^{<1>}$ and $P_{dyn}^{<2>}$ for the token passing protocol. Both measures increase as L increases. This is because an arrived packet may be placed at the j -th position of the queue with probability $q^i(j-1)$ (Section 4.1.2) and has to wait for the transmission of $j-1$ packets ahead of it before competing with other stations for medium access. Besides, since time-constrained (high-priority) packets are less tolerant of the delay due to the FCFS-buffering strategy at each station, the increase in $P_{dyn}^{<1>}$ is more significant than that in $P_{dyn}^{<2>}$. T_ϵ , as shown in the first column of Table 6, increases with L for the same reason. On the other hand, Q_i decreases as L increases, but the degree of improvement becomes insignificant as $L \geq 3$ (Fig. 7). The simulation results are again in good agreement with the analytic ones. (For clarity of presentation, simulation results are shown only in Fig. 6.)

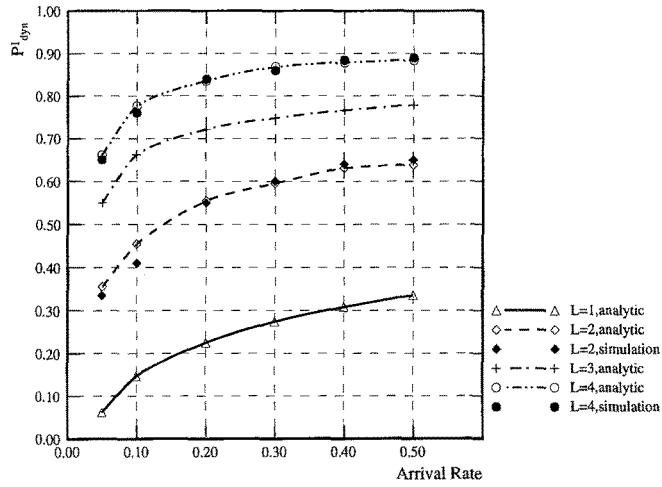


Figure 6a. $P_{dyn}^{<1>}$ for token passing protocol.

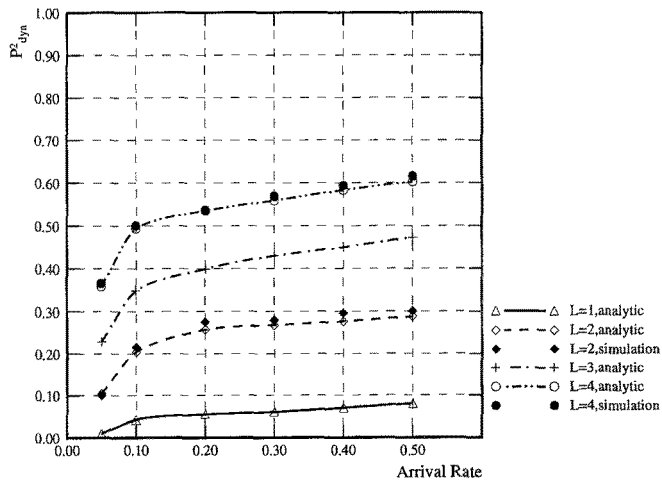


Figure 6b. $P_{dyn}^{<2>}$ for token passing protocol.

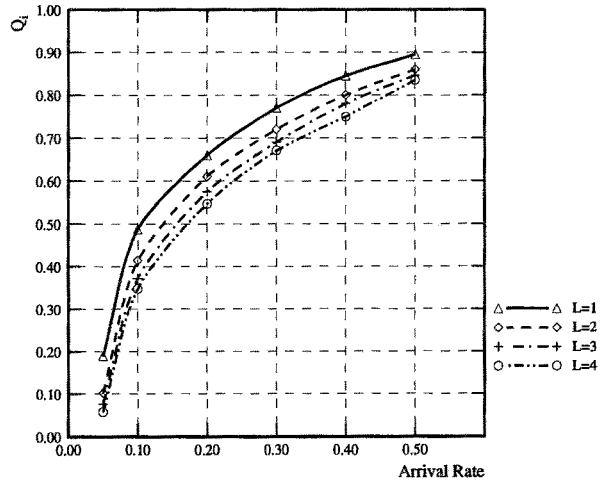


Figure 7. Q_i for token passing protocol.

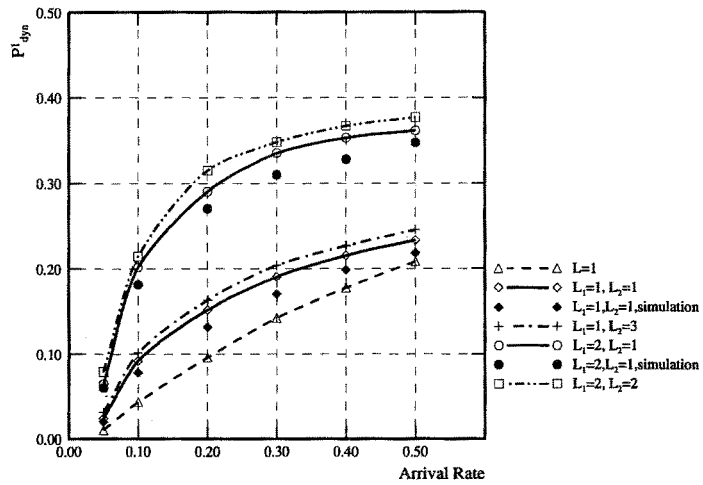


Figure 8a. $P_{dyn}^{<1>}$ for token scheduling protocol.

Table 6. Effect of buffer size on T_e .

λ_i	Buffer size	Token passing	Token scheduling	
		$T_e, \epsilon = 10^{-3}$	$T_e^{<1>}$	$T_e^{<2>}$
0.05	$L = 1$	12.37	3.89	21.76
	$L = 2 (L_1 = 1, L_2 = 1)$	22.43	4.27	32.37
	$L = 3 (L_1 = 1, L_2 = 2)$	34.26	4.32	40.28
	$L = 3 (L_1 = 2, L_2 = 1)$	34.26	10.98	46.32
	$L = 4 (L_1 = 1, L_2 = 3)$	46.38	4.36	52.39
	$L = 4 (L_1 = 2, L_2 = 2)$	46.38	11.07	57.18
0.1	$L = 1$	18.05	9.42	29.43
	$L = 2 (L_1 = 1, L_2 = 1)$	33.74	9.54	46.39
	$L = 3 (L_1 = 1, L_2 = 2)$	48.32	9.60	58.23
	$L = 3 (L_1 = 2, L_2 = 1)$	48.32	17.74	66.47
	$L = 4 (L_1 = 1, L_2 = 3)$	61.92	9.63	73.08
	$L = 4 (L_1 = 2, L_2 = 2)$	61.92	17.82	82.45
0.5	$L = 1$	18.06	17.98	41.89
	$L = 2 (L_1 = 1, L_2 = 1)$	36.12	18.04	62.43
	$L = 3 (L_1 = 1, L_2 = 2)$	54.15	18.06	89.61
	$L = 3 (L_1 = 2, L_2 = 1)$	54.15	27.76	99.75
	$L = 4 (L_1 = 1, L_2 = 3)$	71.97	18.05	119.82
	$L = 4 (L_1 = 2, L_2 = 2)$	71.97	27.97	136.38

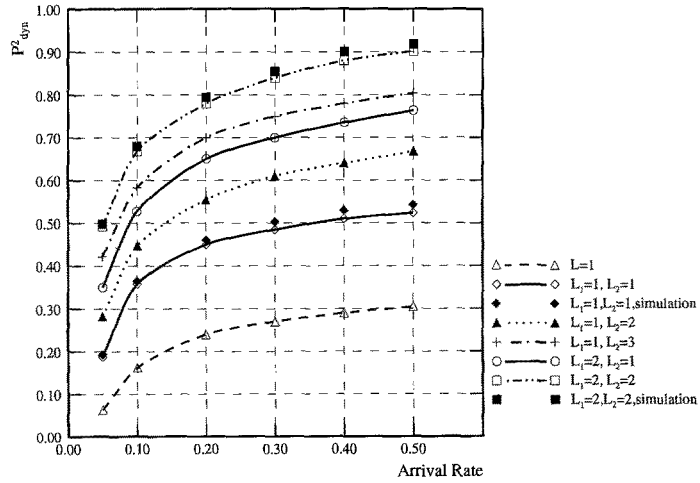


Figure 8a. $P_{dyn}^{<2>}$ for token scheduling protocol.

Figs. 8 (a) and 9 (a) plot $P_{dyn}^{<1>}$ and $Q_i^{<1>}$ of the token scheduling protocol for several combinations of L_1 and L_2 , respectively. The second column of Table 6 gives numerical

examples of $T_\epsilon^{<1>}$. Both $T_\epsilon^{<1>}$ and $P_{dyn}^{<1>}$ increase as L_1 increases, but the increase is not so pronounced as that for non priority-based protocols. This is because (1) a high-priority packet only has to wait for the transmission of high-priority packets (rather than *all* packets) arrived earlier before competing with other stations; (2) the access delay experienced by high-priority packets is much smaller than that in the non priority-based case. $Q_i^{<1>}$ decreases as L_1 increases, and the degree of improvement is more pronounced as compared to that in the non priority-based case (Fig. 9 (a)). (The effect of varying L_2 on $Q_i^{<1>}$ is minimal, and only the curves for varying L_1 are shown in Fig. 9 (a).) This results from the fact that $E(B_i(t))$ for the single-buffer case is substantially larger than $E(B_{i1}(t))$ in the two-buffer ($L_1 = 1, L_2 = 1$) case (see Eq. (12)).¹¹ The improvement on $Q_i^{<1>}$, however, levels off when $L_1 \geq 3$.

An interesting finding is that $P_{dyn}^{<1>}$, $T_\epsilon^{<1>}$, and $Q_i^{<1>}$ remain almost the same (slightly degrade) as L_2 increases. This is because the size of the buffer holding low-priority packets has minimal effects on the performance of high-priority packets. A slight performance degradation results from the non-preemptive transmission policy in which a high-priority packet arriving at a station when a low-priority packet is being transmitted must wait for the in-progress transmission to complete. This possibility increases as L_2 increases. The analytic results for high-priority packets are always slightly worse than, but still in reasonably good agreement with, the corresponding simulation results.

Figs. 8 (b) and 9 (b) show the plots of $P_{dyn}^{<2>}$ and $Q_i^{<2>}$ for the token scheduling protocol. The third column of Table 6 gives numerical examples of $T_\epsilon^{<2>}$. Both $P_{dyn}^{<2>}$ and $T_\epsilon^{<2>}$ increase as L_1 or L_2 increases, and the degradation is more pronounced as L_1 increases. Another interesting consequence of this property is that $Q_i^{<2>}$ increases with L_1 . This results from the fact that the access delay experienced by low-priority packets increases and more low-priority packets get queued in the system as more high-priority packets are accommodated.

The performance of contention protocols (in terms of P_{dyn} and T_ϵ) derived from the multiple-buffer model exhibits similar trends as that derived from the single-buffer model except that the degree of performance improvement (or degradation) is furthered by local buffering strategies. Thus, if the quality of a contention protocol in terms of minimizing P_{dyn} or T_ϵ is the main concern, a single-buffer model does provide a quantitative means of evaluating contention protocols. However, if the combined effects of contention protocols and local buffering strategies on the performance have to be considered, a multiple-buffer model must be used.

6. Conclusion

For two different types of contention protocols, we have developed both single- and multiple-² buffer models to *analytically* evaluate two important parameters for real-time performance: the probability of dynamic failure, P_{dyn} , and the ϵ -bounded delivery time, T_ϵ . Another performance measure obtained as a byproduct is the probability of packet rejection, Q_i . Our analytic results are shown to agree well with the simulation results.

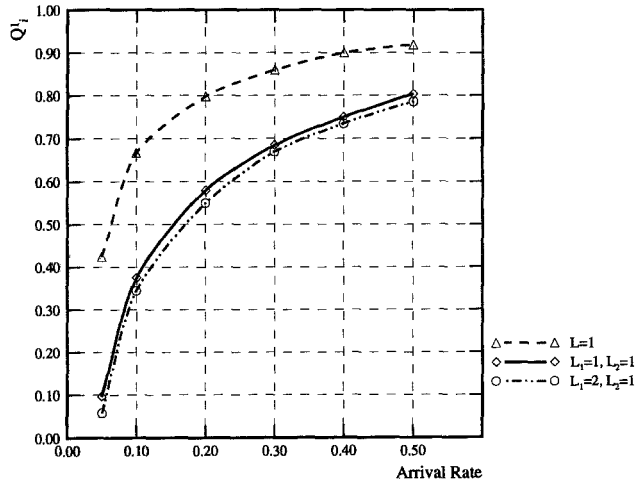


Figure 8a. $Q_i^{<1>}$ for token scheduling protocol.

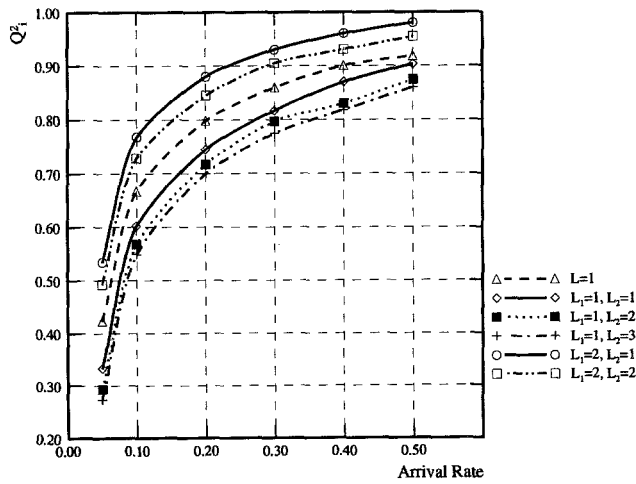


Figure 8b. $Q_i^{<2>}$ for token scheduling protocol.

Our analyses have indicated the suitability of P_{dyn} and T_ϵ for assessing communication subsystems designed for real-time applications. For example, the token scheduling protocol is shown to outperform the token passing protocol with respect to both P_{dyn} and T_ϵ over a wide range of traffic. P_{dyn} is minimized in the token scheduling protocol by giving tight-laxity packets priority over loose-laxity ones in accessing the medium, and the resulting improvement in meeting packet deadlines can be computed from the proposed analytic models. Our results also indicate the existence of a limit on the arrival rate of high-priority packets if every high-priority packet must be delivered by a specified time.

The P_i -persistent protocol is shown to be not attractive for real-time applications. The fact that there is a nonzero probability of a packet being indefinitely delayed is reflected in its unbounded T_ϵ and relatively large P_{dyn} . On the other hand, both P_{dyn} and T_ϵ of its priority-based variation are shown to be acceptably small under certain conditions.

One interesting result is that P_{dyn} and T_ϵ obtained from the multiple-buffer model exhibit similar trends as those obtained from the single-buffer model except that the degree of performance improvement/degradation is furthered by local buffering strategies. Thus, the single-buffer model suffices for the evaluation of *relative* merits of different protocols based on P_{dyn} and T_ϵ .

Analytically evaluating the performance of other real-time contention protocols, such as the virtual time CSMA/CD [29] and the window protocol implementing the minimum-laxity-first-transmission policy [30] would be an interesting extension of the work reported in this paper. Some form of state aggregation must be employed to reduce the size of the exponentially-growing state space for multiple-buffer models, which is currently under investigation.

Notes

1. This assumption is inherited from the assumption of homogeneous stations
2. One can easily extend this to the case with more than two priority levels.
3. There is no need to record the number of outstanding high-priority packets, since these packets must have the highest priority as the token comes to them, i.e., $p_{i1} = 1$.
4. $t_2 - s_2$ is the number of outstanding low-priority packets in the system that had not been transmitted by the end of the last cycle time.
5. k now represents the number of packets that have been transmitted.
6. For all stations to know the channel state (transmission, collision, or idle), the slot length must be at least $2D_p + \gamma$, where D_p is the end-to-end propagation delay, and γ the time required for the detection of a collision by a station. The packet transmission time requires to be greater than $2D_p + \gamma$, so each station can detect a collision.
7. The case in which there are more than two priority levels can be developed following the same approach.
8. Again, two cases are considered: (1) a packet arriving at a station without outstanding packets is placed at the head of the queue and gets transmitted upon the token's next visit; (2) a packet arriving at a station with outstanding packets is placed at the end of the queue, and will not be transmitted upon the token's next visit.
9. Note that the embedding point of the Markov chain is defined differently from that used in deriving the cycle time.
10. The count excludes those visits for priority polling

11. As expressed in Eq. (15), the service time distribution $B_i(t)$ in the single-buffer case is a weighted sum of $D_{A_i}^{<1>}(t)$ and $D_{A_i}^{<2>}(t)$. Since $E(D_{A_i}^{<2>}(t))$ is relatively large as a result of improving $D_{A_i}^{<1>}(t)$, $E(B_i(t))$ becomes large.

References

1. S. Akhtar and A. K. Sood. Performance evaluation of CSMA/CD networks using markovian approach. *GLOBECOM'88 IEEE Global Telecommunications Conference and Exhibition*, pages 1108–1112, 1988.
2. Theodore K. Apostolopoulos and Emmanuel N. Protonotarios. Queueing analysis of buffered CSMA/CD protocols. *IEEE Trans. on Communications*, COM-34(9):898–905, September 1987.
3. M. J. Ferguson. Computation of the variance of the waiting time for token rings. *IEEE J. Selected Areas Commun.*, SAC-4(6):775–782, September 1986.
4. M. J. Ferguson and Y. J. Aminetzah. Exact results for nonsymmetric token ring systems. *IEEE Transactions on Communications*, COM-33(3):223–231, March 1985.
5. Aura Ganz and Imrich Chlamtac. Queueing analysis of finite buffer token networks. *Proc. ACM Sigmetrics*, pages 30–36, 1988.
6. J. Gianini and D. R. Manfield. An analysis of symmetric polling systems with two priority classes. *Performance Evaluations*, 8:93–115, 1988.
7. Donald Gross and Carl Harris. *Fundamentals of Queueing Theory*. New York: Wiley, 1985.
8. Jeremiah F. Hayes. *Modeling and Analysis of Computer Communication Networks*. Plenum Press, 1984.
9. L. Kleinrock. *Queueing Systems, Vol. 1: Theory*. New York: Wiley, 1975.
10. H. Kobayashi. *Modeling and analysis: an introduction to system performance evaluation methodology*. Addison-Wesley Pub. Co., 1978.
11. C. M. Krishna and K. G. Shin. Performance measures for multiprocessor controllers. *Performance '83*, pages 229–250, 1983. North-Holland, New York.
12. James F. Kurose, Mischa Schwartz, and Yechiam Yemini. Controlling window protocols for time-constrained communication in multiple access network. *IEEE Trans. on Communications*, 36(1):41–49, January 1988.
13. D. R. Manfield. Analysis of a priority polling system for two way traffic. *IEEE Trans. Communications*, COM-33(9):1001–1006, September 1985.
14. Biswanath Mukherjee and James S. Meditch. The p_i -persistent protocol for unidirectional broadcast bus networks. *IEEE Transactions on Communications*, 36(12):1277–1286, December 1988.
15. J. W. M. Pang and R. W. Donaldson. Approximate delay analysis and results for asymmetric token passing and polling networks. *IEEE J. Selected Areas Commun.*, SAC-4(6):783–793, September 1986.
16. I. Rubin and Z. Tsai. Performance analysis of a message priority based polling scheme. *Proc. ICC*, pages 22.1.1–22.1.5, June 1988.
17. K. G. Shin, C. M. Krishna, and Y.-H. Lee. A unified method for evaluating real-time computer controllers and its application. *IEEE Trans. on Automatic Control*, AC-30(4):357–366, April 1985.
18. Kang G. Shin and C.-J. Hou. Analysis of three contention protocols in distributed real-time systems. *IEEE Proc. of 11th Real-Time Systems Symposium*, pages 136–145, December 1990.
19. J. K. Strosnider and T. E. Marchok. Responsive, deterministic IEEE 802.5 token ring scheduling. *Real-Time Systems Journal*, 1(2):133–158, September 1989.
20. J. K. Strosnider, T. E. Marchok, and J. P. Lehoczky. Advanced real-time scheduling using the IEEE 802.5 token ring. *Proceedings of IEEE Real-Time Systems Symposium*, 1988.
21. Efstathios D. Sykas, Kionysios E. Karvelas, and Emmanuel N. Protonotarios. Queueing analysis of some buffered random multiple access schemes. *IEEE Trans. on Communications*, COM-34(8):790–798, August 1986.
22. H. Takagi. *Analysis of Polling Systems*. The MIT Press, Cambridge, Massachusetts, 1986.
23. H. Takagi. Queueing analysis of polling models. *ACM Comput. Surveys*, 20:5–28, March 1988.
24. Tetsuya Takine, Yutaka Takahashi, and Toshiharu Hasegawa. Exact analysis of asymmetric polling systems with single buffers. *IEEE Transactions on Communications*, 6(10):1119–1127, October 1988. Also in *Performance '87*, P.-J. Courtois, and G. Latouche, Eds. Elsevier North-Holland, Amsterdam.
25. Andrew S. Tanenbaum. *Computer Network*. Prentice-Hall, Inc., 1989.

26. F. A. Tobagi and V. B. Hunt. Performance analysis of carrier sense multiple access with collision detection. *Computer Networks*, 4:245–259, 1980.
27. Z. Tsai and I. Rubin. Performance of token schemes supporting delay-constrained priority traffic streams. *IEEE Trans. Communications*, COM-38(11):1994–2003, November 1990.
28. Cheng-Shong Wu and Victor O. K. Li. Delay analysis of buffered CSMA/CD protocols. *GLOBECOM'88, IEEE Global Telecommunications Conference and Exhibition*, pages 694–698, 1988.
29. Wei Zhao and K. Ramamitham. Virtual time CSMA protocols for hard real-time communications. *IEEE Transactions on Software Engineering*, SE-13(8):938–952, August 1987.
30. Wei Zhao, J. Stankovic, and K. Ramamitham. A window protocol for transmission of time constrained messages. *Proceedings of the 8th International Conference on Distributed Computing Systems*, pages 384–392, 1988.