Mixed Time-Constrained and Non-Time-Constrained Communications in Local Area Networks

Kang G. Shin, *Fellow, IEEE,* and Qin Zheng, *Member, IEEE*

***Abstract*—It is well known that some time-token medium access protocols for local area networks (LANs) like the IEEE 802.4 token bus and the FDDI token ring can guarantee the medium access delay for time-constrained packets. However, a problem which has been largely overlooked is how these protocols can be made to provide a maximum throughput for nontime-constrained packets while guaranteeing the delay bound of time-constrained packets. We first show how the parameters of the IEEE 802.4 token bus and the FDDI token ring can be set to solve the above problem. Then, we design a new timer mechanism for the timed-token protocols which provides the highest guaranteed throughput of nontime-constrained packets among a set of medium access protocols called the *token passing protocol,* to which most of the existing non-contention LAN protocols belong. We present numerical examples to compare different protocols, all of which have shown the superiority of the proposed protocol to the others.**

I. Introduction

In recent years, real-time communications has been receiving considerable attention [1], [2] because of its increasing need in such applications as packetized voice/video communications, computer-integrated manufacturing, and real-time control systems. In a real-time system, each time-constrained packet generated by the source station must be received by the destination station within a specified amount of time.

According to the objectives used, research in real-time communication can be grouped into two categories: best-effort communication and hard real-time communication. The primary objective of the first category is to maximize the percentage of packets meeting their delivery deadlines. This is based on the observation that for applications like packetized voice/video communication, a certain amount of packet loss is tolerable and a packet which is not successfully delivered within a certain time limit is considered lost [2]. The best-effort communication may be acceptable for voice/video applications but may cause a serious problem is the packet to be transmitted is an actuation/control signal.

Paper approved by M. Nassehi, the Editor for local Area Networks of the IEEE Communications Society. Manuscript received January 17, 1991; revised September 28, 1991. This work was supported in part by the Office of Naval Research under Contract N00014-92-J-1080 and the National Science Foundation under Grant DMC-8721492.

K. G. Shin is with the Department of Electrical Engineering and Computer Science, Real-Time Computing Laboratory, University of Michigan, Ann Arbor, MI 48109.

Q. Zheng was with the Department of Electrical Engineering and Computer Science, Real-Time Computing Laboratory, University of Michigan, Ann Arbor, MI 48109. He is now with Mitsubishi Electric Research Labs., Inc., Cambridge, MA 02139.

IEEE Log Number 9211394.

The second category deals with the so-called hard real-time communication problem, in which *all* time-constrained packets are required to be delivered before their deadlines. This kind of service is essential for critical real-time systems, such as aircraft and nuclear power plants, and is feasible due to the predictability of most real-time traffic. For example, the packets associated with feedback signals in real-time control systems are determined by the number of sensors and the sampling periods. The traffic of packetized voice is bounded above by the number of voice channels that the network is designed to support. Similarly, the maximum traffic of alarm signals is bounded by the number of alarm sources and the maximum alarm frequencies, which are unlikely to occupy a large portion of the network bandwidth.[1] So, it is common practice to make *a priori* assumptions about the amount of the time-constrained traffic and to require that all time-constrained packets be transmitted before their deadlines.

This paper will address exclusively the problem of hard real-time communication in the presence of nontime-constrained packets. Guaranteeing service to the nontime-constrained packets is a problem which as been largely overlooked in the study of real-time communications. Different types of packets in a computer network may have different service requirements. Time-constrained (or real-time) packets, such as alarm signals, packetized voice, and the information to be used for real-time control, must reach the destination before certain deadlines. There are also other packets in the network, like those related to file transfer, e-mail, fax, and routine data collection, which can tolerate a relatively large latency but may introduce a heavy average traffic. Although nontime-constrained packets can tolerate a relatively large latency, they usually require a certain guaranteed bandwidth of the network. Because of limited station buffers, failure to do so will cause loss of packets. Thus, it is practically important to address the problem of maximizing the guaranteed throughput for nontime-constrained packets while meeting the delivery deadlines for time-constrained packets, which is the subject of this paper.

Several researchers reported different ways of setting parameters in LAN protocols to meet various service requirements [3]–[5]. Among these, the method proposed by Jayasumana *et al.* [3] is the most closely related to the problem stated above. Under a certain assumption about the traffic of time-constrained packets generated by each individual station, they showed the way of setting parameters in an IEEE 802.4 token bus such that the deadlines of time-constrained

[1] If not, it is not an adequate design.

packets can be met while guaranteeing a certain average throughput for nontime-constrained packets. They also claimed that their results can be easily extended to cover the FDDI networks. However, since there is more than one medium access protocol which can meet the requirements of time-constrained packets, one can raise the following two questions:

- Which of the various existing medium access protocols is better suited for mixed time-constrained and nontime-constrained communications?
- Is it possible to design a protocol which is best suited for mixed time-constrained and nontime-constrained communications?

By "better (best) suited" we mean that the network guarantees a higher (the highest) throughput for nontime-constrained packets while ensuring all time-constrained packets to meet their deadlines.

We shall first analyze the suitability of two adopted LAN protocols, the IEEE 802.4 token bus and the FDDI token ring, for mixed time-constrained and nontime-constrained packets. This part of the work is somewhat similar to that of [3], but possesses the following two notable differences:

- We are concerned with the problem of *maximizing* the guaranteed throughput for nontime-constrained packets while satisfying the requirements of time-constrained packets. By comparing the maximum guaranteed throughputs of different protocols, one can determine which of the protocols is best suited for mixed time-constrained and nontime-constrained packets.
- We make an assumption about the traffic of time-constrained packets of the *whole network,* rather than that of each individual station. As will be explained later, this assumption is less restrictive than the one used in [3]. Also, we do not need to assume the network to be symmetric or partially symmetric.[2]

In the second part of this paper, we shall propose a time-token medium access protocol which provides a higher guaranteed throughput for nontime-constrained packets than all the LAN protocols considered while retaining the ability of meeting the deadlines of time-constrained packets. We also show that this protocol is optimal among a set of token passing protocols in the sense that it provides the maximum guaranteed throughput for nontime-constrained packets.

This paper is organized as follows. Section II gives the problem formulation and definitions of some terms necessary for our discussion. Section III shows the ways of tuning IEEE 802.4 and FDDI protocols. A new, optimal protocol is presented and evaluated in Section IV. Performances of different protocols are compared in Section V. The paper concludes with Section VI.

[2] As was defined in [3], a network is said to be *partially symmetric* if the traffic of a given class is symmetrically distributed among the devices generating that class of traffic but need not be present at all the network nodes.

II. Problem Formulation

We shall study a class of medium access protocols called *token passing protocols* (TPP's) which have the following features.

1) A token circulates among all N stations of a bus/ring network in a round-robin fashion. Only the station possessing the token has the right to transmit packets. The token passing time from one station to the next is a constant[3], T_t.

2) Each station has timers to control how long this station and each priority class may use the channel once the station captures the token. When the timers expire or there are no packets to transmit, the token is passed over to the next station.

3) The packets transmitted by a station are seen (possibly after certain amounts of delay as in a ring network) by all other stations in the network. Thus, every station knows how long every other station takes to transmit packets.

Most of the existing token-controlled medium access protocols [6], such as the IEEE 802.4 token bus, the IEEE 802.5 reservation-based token ring, and the FDDI timer-based token ring, are TTP's. So, the definition of TPP is quite general. In the rest of this paper, packets are grouped into two classes: time-constrained (called *class A*), and nontime-constrained (called *class B*). Class A packets require a guaranteed medium access delay and class B packets require a guaranteed throughput.

The *medium access delay* of a packet is defined as the time period between its generation at a station and the beginning of its transmission. The hard real-time communication problem in a token passing network is to design a token passing protocol such that all class A packets have the medium access delay less than/equal to a given constant D_A. A necessary condition under which a token passing protocol meets this requirement is that the token must visit each station at least once every D_A units of time. Otherwise, a packet arrived at a station just after the station passed the token to its next station would miss its deadline. The above condition becomes a sufficient one if the token passing protocol is designed such that each station is given enough time to transmit all class A packets generated since the token's last visit to the station. To achieve this goal, certain assumptions must be made about the class A traffic. The authors of [3] made the following assumption.

A0) If the token visits each station at least once every D_A units of time, then the time needed for a station to transmit all its class A packets generated since the token's last visit to the station is less than/equal to a constant T_S, such that $N_A T_S \leq D_A - NT_t$ where N is the total number of stations in the network, N_A is the number of stations that have class A packets to transmit, T_t is the token passing time, and D_A is the required access delay bound of class A packets as defined above.

Note that $D_A - NT_t$ is the time available for transmitting packets over the bus/ring during one complete rotation of the token if the rotation period equals D_A. So, the maximum required transmission time for class A packets $N_A T_S$ must not exceed this value.

[3] We assume that the variations of token passing times are negligibly small relative to the average packet transmission time.

The physical meaning of A0) is that each station's average rate of generating class A packets over a time period D_A is bounded above by T_S/D_A, and the total rate of class A packets generated by the entire system does not exceed the transmission capacity of the bus. In this paper, we shall use only the latter part of the assumption as stated below.

A1) If the token visits each station at least once every D_A units of time, then the summation over N consecutive stations of the time needed for each station to transmit all its class A packets generated since the token's last visit to the station is less than/equal to a constant T_A, such that $T_A \leq D_A - NT_t$.

To clarify the meaning of A1), let $\tau_{A_k}(n)$ denote the time needed for station k to transmit all its class A packets during the token's nth visit to the station, and $\tau_{B_k}(n)$ denote the time station k uses to transmit the class B packets during the token's nth visit to the station, $k = 0, \cdots, N-1, n = 1, 2, \cdots$. For notational convenience, let $\tau_A(k + nN) = \tau_{A_k}(n)$ and $\tau_B(k + nN) = \tau_{B_k}(n)$. Also, let $\tau_A(i) = \tau_B(i) = 0$ for $i < N$. Then, A1) can be expressed as $\forall\, m \geq N$, if the time period between two consecutive visits of the token to a station $\sum_{i=m-N}^{m-1} \tau_A(i) + \sum_{i=m-N}^{m-1} \tau_B(i) + NT_t \leq D_A$, then $\sum_{i=m-N+1}^{m} \tau_A(m) \leq T_A \leq D_A - NT_t$.

Satisfaction of A0) implies that of A1). So, A1) is less restrictive than A0). There are two advantages of using A1). First, the network does not have to be symmetric, i.e., different stations may have different generating rates of class A packets. Secondly, related station, e.g., those are engaged in interactive voice communication, can be grouped together to make more efficient use of the network bandwidth. In other words, one only needs to provide an upper bound of the class A traffic of the group, which is usually smaller than the summation of the traffic bounds over all individual stations. In this paper, we assume the generation of class A packets satisfies A1) unless specified otherwise.

Most existing token passing protocols can satisfy the access delay requirement of class A packets under assumption A1), but the their capability to accommodate class B packets could be significantly different. In order to compare different protocols, we propose a measure of this capability—the *guaranteed throughput for class B packets*—which is defined as the minimum throughput[4] achievable under the condition that at least one station has an infinite length queue for class B packets, and class A packets satisfy assumption A1).

The above definition of guaranteed throughput can be elaborated as follows. If the offered load of class B packets (i.e., the time needed to transmit class B packets generated by all stations in a time period T divided by T) is low, the network can usually transmit all packets without any packet loss. As the offered load increases, the network will gradually saturate and one or more stations will have infinite length queues for class B packets. So, the throughput under the condition that at least one station has an infinite length queue actually gives a value of the offered load above which the network could saturate. This value varies with the traffic volume of class A packets and the arrival pattern of class B packets. The guaranteed throughput is therefore the minimum of this value. Based on this definition, the network will never get saturated if the offered load of class B packets is below the guaranteed throughput. The value of the guaranteed throughput for class B packets gives the network designer the useful information about how much class B packets can be safely transferred across the network.

With the above assumptions and definitions, we can now state the following two problems.

1) For the IEEE 802.4 token bus and the FDDI token ring, *set* the parameters of their timer mechanisms to maximize the guaranteed throughput for class B packets while ensuring all class A packets to meet their deadlines.

2) *Design* an optimal token passing protocol which gives the maximum guaranteed throughput for class B packets among all TPP's and simultaneously satisfies the timing requirements of class A packets.

The first problem is to show how to tune existing LAN protocols for both time-constrained and nontime-constrained communications, and the second is to show how to build a new, optimal protocol.

III. Tuning of Existing LAN Protocols

As mentioned earlier, most of the existing token-controlled medium access protocols are TPP's. So, the essential difference of the medium access control policies used by these protocols lies in the timer mechanisms employed. In this section, we shall examine and compare the timer mechanisms of the IEEE 802.4 token bus and the FDDI token ring protocols and show how their parameters can be set to meet the access delay requirements of class A packets and simultaneously maximize the guaranteed throughput for class B packets.

A. IEEE 802.4 Token Bus

The timer mechanism used in the IEEE 802.4 token bus protocol works as follows.

Rule 1) All N stations in the network are numbered as $0, 1, \cdots, N-1$. Each station has two timers: the token-holding-timer, Timer1, and the token-rotation-timer, Timer2. At the startup of the system, station 0 has the token and Timer2 at station k is set to $(N - k)T_t$ where T_t is the token passing time.[5]

Rule 2) Timer1s count down and Timer2s count up. Whenever a station gets the token, its Timer1 is set to the token-holding-time T_S. The station is allowed to transmit class A packets until all of them are transmitted or its Timer1 expires,[6] whichever occurs first.

Rule 3) After a station finishes transmitting class A packets, set its Timer1 := T_R−Timer2, and reset Timer2 := 0 where T_R is the target token-rotation-time. The station is allowed to transmit class B packets until all of them are transmitted or

[4] In this paper, by "throughput" we mean the normalized average throughput which is defined as $\lim_{T\to\infty} T(P)/T$ where $T(P)$ denotes the time used to transmit packets during a time period T.

[5] This can be implemented by letting the token start at station 0 and allowing no packets to be transmitted during the first rotation of the token.

[6] The station is allowed to finish transmitting the current packet when its Timer1 expires.

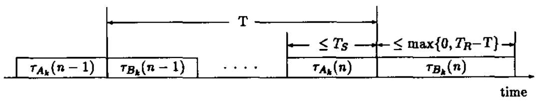


Fig. 1. Operations of IEEE 802.4 token bus.

its Timer1 expires, whichever occurs first. It then passes the token to the next station.

Recall that $\tau_{A_k}(n)(\tau_{B_k}(n))$ denotes the time station k used to transmit class A (class B) packets during the token's nth visit to the station. Fig. 1 illustrates the operation of the IEEE 802.4 token bus. A station is allowed to transmit class A packets for a time period up to T_S, and class B packets for a time period up to $\max\{0, T_R - T\}$ where T is the time period from the beginning of $\tau_{B_k}(n-1)$ to that of $\tau_{B_k}(n)$, and T_S and T_R are the token-holding-time and the target token-rotation-time which need to be set before the operation of the network.

In the IEEE 802.4 token bus protocol, a station is allowed to finish transmitting the current packet even if its Timer1 expires. In some other protocols, like the IEEE 802.5 and FDDI, a station is allowed to transmit a packet only if the transmission can be completed before Timer1 expires. Thus, the exact performance of the system depends on the distribution of packet lengths. Jayasumana *et al.* [3] attempted to analyze this exact performance. Let t'_A and t_A denote the maximum and mean values of the time needed to transmit a class A packet. They claim that if there are many class A packets to transmit, the maximum and mean times a station would use to transmit class A packets are $T_S + t'_A$ and $T_S + t_A$, respectively. However, we found that this is not always true. For example, suppose all class A packets are of fixed size which need 5 units of time to transmit and T_S is set to 12. Then, if class A queues are heavily populated, $\lceil 12/5 \rceil = 3$ class A packets are always transmitted during each token's visit to a station. Thus, the maximum and mean times a station would use to transmit class A packets are 15 units of time which is not equal to $T_S + t'_A = 17$. In order to avoid these trivial details, we assume in this paper that packet length is relatively small as compared to the token-holding-time T_S, thus making the residual effect negligible.

The following theorem shows how to set the parameters T_S and T_R in the IEEE 802.4 token bus protocol to satisfy the timing requirements of class A packets and simultaneously maximize the guaranteed throughput of class B packets.

Theorem 1:

i) For an IEEE 802.4 token bus network, all class A packets have access delays not larger than D_A if the token-holding-time T_S and the target token-rotation-time T_R are set such that $T_S \geq T_A$ and $T_R \leq D_A - T_A$.

ii) By setting $T_R = D_A - T_A$, the network achieves its maximum guaranteed throughput for class B packets,

$$\begin{aligned} U_{B_{802.4}} &= 1 - T_A/D_A \\ &\quad - (2 - T_A/D_A)NT_t/(D_A - T_A + NT_t) \end{aligned}$$

where the parameters N, T_A, and D_A are defined in Section II.

Proof: i) is a direct result of Theorem 3.

To prove ii), we first prove that for any constant $T_R, T_R \leq D_A - T_A$ is also a necessary condition to meet the access delay requirement of class A packets. Recall that a necessary condition to meet the access delay requirement of class A packets is that the token must visit each station at least once every D_A units of time. Suppose T_R is set to be larger than $D_A - T_A$. From Fig. 1, in case no class A or class B packets are transmitted during the time period T, station k could use $\tau_{B_k}(n) = T_R - T > D_A - T_A - NT_t$ units of time to transmit its class B packets. Also, station $k+1$ could use $\tau_{A_{k+1}}(n) = T_A$ units of time to transmit its class A packets. This would make the token rotation time at station $k+2$ larger than D_A.

We now compute the guaranteed throughput for class B packets. Recall that the guaranteed throughput for class B packets is defined as the minimum throughput achievable under the condition that at least one station has an infinite length of class B packets. Suppose station k has an infinite length queue of class B packets. Let T_{ave} be the average token rotation period, and $U_A(U_B)$ denote the throughput of class A (class B) packets. Then, $U_B = 1 - U_A - NT_t/T_{\text{ave}}$ where NT_t/T_{ave} is the part of the network's bandwidth used to transmit the token.

From the operation rules of the IEEE 802.4 token bus protocol, since station k has an infinite length queue of class B packets, it will keep transmitting class B packets until Timer1 expires. Thus, from Fig. 1, $\tau_{B_k}(n) = \max\{0, T_R - T\} \geq T_R - T$. Averaging both sides we get $\tau_{B_k} \geq T_R - T_{\text{ave}}$ where τ_{B_k} is the average value of $\tau_{B_k}(n)$ over n. The maximum value that τ_{B_k} can attain is $U_B T_{\text{ave}}$, which corresponds to the situation where station k has an infinite length queue of class B packets and all other stations do not have any class B packets to transmit. Thus, $T_{\text{ave}} \geq T_R/(1 + U_B)$. T_{ave} reaches its minimum value $T_R/(1 + U_B)$ under the condition that $T_R - T \geq 0$ all the time. Also, the maximum values of U_A and T_R are T_A/D_A and $D_A - T_A$, respectively. Thus, we get the minimum value of U_B satisfies

$$U_B = 1 - T_A/D_A - NT_t(1 + U_B)/(D_A - T_A).$$

Solving U_B from this equation, we get the guaranteed throughput for the IEEE 802.4 token bus

$$\begin{aligned} U_{B_{802.4}} &= 1 - T_A/D_A \\ &\quad - (2 - T_A/D_A)NT_t/(D_A - T_A + NT_t). \quad \square \end{aligned}$$

A couple of remarks on the above theorem are in order.

1) From Theorem 1, if assumption A1) holds, the token-holding-time T_S can be assigned an arbitrarily large value without affecting the network performance. This means that a station does not have to restrict the transmission of its class A packets as long as the generation of class A packets of the network abides by the discipline as stated in assumption A1). Actually, keeping a class A packet from transmission during the current token's visit will result in the missing of the packet's deadline.

For practical applications, however, some mandatory restrictions on the transmission of class A packets is necessary in

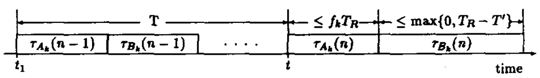


Fig. 2. Operations of the FDDI token ring.

order to prevent a malfunctioning station from holding the token indefinitely. The simplest way is to set $T_S = T_A/N_A$, i.e., to allow each station the same amount of time to transmit its class A packets. This is equivalent to using assumption A0). Since we have used a less restrictive assumption A1), more flexible ways of restricting the transmission of class A packets are also possible. As stated in Section II, we can group a set of related stations together and restrict the total transmission time of this group during each token's rotation. This results in a more flexible way to allocate the network's bandwidth to class A packets and increases the network's efficiency.

2) From the proof of Theorem 1, the guaranteed throughput for class B packets is calculated when only one station has an infinite length queue of class B packets. If the traffic pattern of the network is symmetric, i.e., all stations have many class B packets to transmit, then the guaranteed throughput can be increased to

$$U_{B_{S802.4}} = 1 - T_A/D_A - (N + 1 - T_A/D_A)T_t/(D_A - T_A + T_t).$$

This means the throughput for class B packets in the IEEE 802.4 token bus protocol is sensitive to the distribution of class B packets.

B. FDDI Token Ring

The timer mechanism used in the FDDI token ring works as follows.

Rule 1) All N stations in the network are numbered as $0, 1, \cdots, N-1$. Each station has two timers: the token-holding-timer, Timer1, and the token-rotation-timer, Timer2. At the startup of the system, Timer2 at station k is set to $(N - k)T_t$ and station 0 has the token.

Rule 2) Timer2 counts up. The token is said to be *late* at a station if the station's Timer2 counts up to T_R before the token arrives at the station. If the token is late, a station's Timer2 is reset to 0 when it reaches T_R. Otherwise, Timer1 is set to T_R-Timer2 and Timer2 is reset to 0 when the station gets the token.

Rule 3) Station k, $0 \leq k \leq N - 1$, is allowed to transmit class A packets for a time period up to f_kT_R when it gets the token where f_k is the percentage of the bandwidth station k reserved to transmit its class A packets. If the token is late, the station passes the token to the next station immediately after it finishes transmitting class A packets. Otherwise, its Timer1 starts counting down and the station is allowed to transmit class B packets until all of them are transmitted or its Timer1 expires, whichever occurs first. It then passes the token to the next station.

The operations of the FDDI Token Ring are shown in Fig. 2 where $T' = T$ is Timer2 was reset at time t_1, otherwise $T' = t - t_0$ is the last time Timer2 was reset before t_1.

Theorem 2:

i) For an FDDI token ring network, all class A packets have access delays not larger than D_A if f_k and T_R satisfy $f_kT_R \geq T_A, k = 0, 1, \cdots, N - 1$, and $T_R \leq D_A - T_A$.

ii) The guaranteed throughput is maximized by setting $T_R := D_A - T_A$, and it is bounded by $U_{B_{FDDI}} \leq U_{B_{802.4}} = 1 - T_A/D_A - NT_t(2 - T_A/D_A)/(D_A - T_A + NT_t)$.

Proof: In the same way as that of the proof of Theorem 3, it can be proved that the maximum value of the token rotation time T equals $T_R + T_A$. Thus, by setting $T_R \leq D_A - T_A$, the token rotation time $T \leq D_A$. Together with the assumption that f_k is set to be $f_kT_R \geq T_A$, the proof of i) follows.

To prove ii), we use a simplification of the FDDI protocol as proposed in [7] in which the transmission time of class B packets is restricted by $\max\{0, T_R - T\}$ instead of $\max\{0, T_R - T'\}$ (see Fig. 2). Since $T' \geq T$, with the same setting of T_R, the simplified protocol has a higher (or equal) throughput for class B packets than that of the standard FDDI.

In the same way as that of the proof of Theorem 1, it can be proved that the simplified FDDI protocol satisfies the time requirement of class A packets by setting $T_R \leq D_A - T_A$ and $f_kT_R \geq T_A$, and its guaranteed throughput for class B packets reaches its maximum $U_{B_{802.4}}$ by setting $T_R = D_A - T_A$. Thus, we proved that the guaranteed throughput for the standard FDDI is upper bounder by $U_{B_{802.4}}$. □

The simplified version of FDDI can be implemented by resetting Timer2 at the time a station captures the token no matter whether the token is late or not. In [7], it was claimed that the simplified FDDI guarantees sufficient responsiveness and capacity for the transmission of class A traffic, and it may provide improved responsiveness to class B transmissions. The proof of the above theorem supports this claim.

From Theorem 2, we also see that the timer mechanism of IEEE 802.4 is at least as good as that of the FDDI in satisfying the access delay requirement for class A packets and providing guaranteed throughput for class B packets. By comparing Figs. 1 and 2, the timer mechanism of the simplified FDDI protocol is similar to that of the IEEE 802.4. So, in general, we expect the IEEE 802.4 protocol to provide a higher throughput for class B packets than that of the standard FDDI.

It is worth stressing that we are comparing the timer mechanisms of different protocols, not the networks on which the protocols are used. By saying the IEEE 802.4 token bus in better than the FDDI token ring, we actually mean that if one uses the timer mechanism of an IEEE 802.4 token bus for an FDDI token ring, the token ring will provide a higher throughput for class B packets. There is no way one can directly compare an IEEE 802.4 token bus network with an FDDI token ring network, since the latter has a much higher bandwidth than the former.

IV. Optimal Token Passing Multiple-Access Protocol

In this section, we propose a new timer mechanism for the token passing protocols to achieve higher throughput for class B packets. First, let us consider the operations of IEEE 802.4 token bus to see how the throughput for class B packets could possible be increased. As shown in

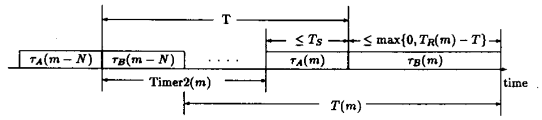


Fig. 3. IEEE 802.4 with dynamic setting of T_R.

Fig. 1, the transmission times of class B packets are controlled by the target token-rotation-time T_R. The larger T_R is, the more class B packets could be transmitted. However, T_R cannot be set too large since class A packets require the token rotation period to be no larger than D_A. Thus, the maximum constant value $T_R = D_A - T_A$ was found (Theorem 1) to ensure the required token rotation period.

However, this maximum value of T_R is very conservative in the sense that is set statically to ensure the token rotation period with the worst case traffic pattern (see the proof of Theorem 1). Actually, T_R could be set larger under other situations. An intuitive observation is that if the total time the stations used to transmit class B packets during one token's rotation is kept below $D_A - T_A - NT_t$ (T_A and NT_t units of time are reserved for the transmission of class A packets and the token, respectively), then the token rotation period will not exceed D_A. From Fig. 1, this is equivalent to increasing a station's T_R dynamically by the amount of time which has been used to transmit class B packets during the last token's visit. The idea is formalized and proved in the following theorem.

Theorem 3: For an IEEE 802.4 token bus network, all class A packets will have access delays not larger than D_A if the token-holding-time $T_S \geq T_A$ and station k's target token-rotation-time at the token's nth visit to the station is dynamically set to

$$T_R(m) \leq D_A - T_A + T_A(m) + \tau_B(m-N),$$

where $m = k + nN$, and $T_A(m) = \sum_{i=m-N+1}^{m} \tau_A(i)$.

Proof: The modified IEEE 802.4 protocol is shown in Fig. 3. As stated in Section II, all class A packets will be transmitted before their deadlines if (C1) the token visits each station at least once every D_A units of time, and (C2) a station is given enough time to transmit all class A packets generated since the token's last visit to the station.

To prove C1, let $\text{Timer2}_k(n)$ be the station k's value of Timer2 during the token's nth visit to the station. Let $\text{Timer2}(k+nN) = \text{Timer2}_k(n)$. Recall that $\tau_A(k+nN) = \tau_{A_k}(n)$, $\tau_B(k+nN) = \tau_{B_k}(n)$, and $\tau_A(i) = \tau_B(i) = 0$ for $i < N$. Then, from the timer mechanism of the IEEE 802.4 token bus protocol, we get

$$\text{Timer2}(m) = \sum_{i=m-N+1}^{m-1} \tau_A(i) + \sum_{i=m-N}^{m-1} \tau_B(i) + NT_t$$

and C1 is equivalent to

$$\text{Timer2}(m) + \tau_A(m-N) \leq D_A, \qquad m = 1, 2, \cdots.$$

We prove the above inequality by mathematical induction on m. For $m = 1$, $\text{Timer2}(1) + \tau_A(1-N) = \text{Timer2}(1) = NT_t < D_A$. Suppose $\text{Timer2}(m) + \tau_A(m-N) \leq D_A$, i.e., $\sum_{i=m-N}^{m-1} \tau_A(i) + \sum_{i=m-N}^{m-1} \tau_B(i) + NT_t \leq D_A$. We need to show $\text{Timer2}(m+1) + \tau_A(m+1-N) \leq D_A$, or equivalently,

$$\sum_{i=m+1-N}^{m} \tau_A(i) + \sum_{i=m+1-N}^{m} \tau_B(i) + NT_t \leq D_A.$$

From assumption A1) about the class A traffic, the first term of the above inequality

$$\sum_{i=m+1-N}^{m} \tau_A(i) \leq T_A.$$

Thus, we only need to show that the second term

$$\sum_{i=m-N+1}^{m} \tau_B(i) \leq D_A - T_A - NT_t.$$

If $\tau_B(i) = 0$ for all $m-N+1 \leq i \leq m$, the above inequality holds trivially. Otherwise, let m_1 be the largest integer such that $\tau_B(m_1) > 0$ and $m - N + 1 \leq m_1 \leq m$. From Rule 3 of the IEEE 802.4 protocol and since $\tau_B(m_1) > 0$,

$$\begin{aligned}\tau_B(m_1) &\leq T_R(m_1) - \text{Timer2}(m_1) - \tau_A(m_1)\\ &= T_R(m_1) - \sum_{i=m_1-N+1}^{m_1} \tau_A(i)\\ &\quad - \sum_{i=m_1-N}^{m_1-1} \tau_B(i) - NT_t.\end{aligned}$$

From the assumption about the setting of the token-rotation-time $T_R(m_1)$, we have

$$T_R(m_1) \leq D_A - T_A + \sum_{i=m_1-N+1}^{m_1} \tau_A(i) + \tau_B(m_1 - N).$$

Thus,

$$\tau_B(m_1) = D_A - T_A - \sum_{i=m_1-N+1}^{m_1-1} \tau_B(i) - NT_t.$$

Or equivalently,

$$\sum_{i=m_1-N+1}^{m_1} \tau_B(i) \leq D_A - T_A - NT_t.$$

Since $m_1 \leq m$ and $\tau_B(i) = 0$ for $m_1 \leq i \leq m$, we prove

$$\sum_{i=m-N+1}^{m} \tau_B(i) \leq \sum_{i=m_1-N+1}^{m_1} \tau_B(i) \leq D_A - T_A - NT_t.$$

Proof of C2 is trivial from C1. Since the token visits each station at least once every D_A units of time and the token-holding-time T_S is set to be no less than T_A, from assumption A1), every station has enough time to transmit its class A packets. □

By setting $T_R(m) = D_A - T_A + T_A(m) + \tau_B(m-N)$, the modified protocol allows more class B packets to be transmitted in each token's rotation than the standard IEEE 802.4. So it can provide a higher throughput for class B packets than the standard protocol. Actually, the modified protocol is not only superior to the standard IEEE 802.4 in this case, but also to any token passing protocols (TPP's) in the sense that it has the highest guaranteed throughput for class B packets. For this reason, we call the modified protocol as the "optimal" TPP when $T(m)$ is set to $D_A - T_A + T_A(m) + \tau_B(m-N)$.

To prove its optimality, we first calculate an upper bound of guaranteed throughput for token passing protocols, and then show that the optimal one reaches this upper bound.

Recall that the guaranteed throughput for class B packets is defined as the minimum throughput achievable under the condition that at least one station has an infinite queue of class B packets. Let T be the maximum average token rotation period achievable under the condition that at least one station has an infinite length queue of class B packets. From the assumption A1) about the class A traffic, the maximum class A throughput is $U_A = T_A/D_A$. Then, the guaranteed throughput for class B packets is $U_B = 1 - T_A/D_A - NT_t/T$ where NT_t/T is the portion of the network bandwidth used by the token. To satisfy the access delay requirements of class A packets, $T \leq D_A$. We get an upper bound of the guaranteed throughput for class B packets,

$$U_{B_{\text{upper}}} = 1 - T_A/D_A - NT_t/D_A .$$

Thus, for any token passing protocol, its guaranteed throughput for class B packets cannot exceed $U_{B_{\text{upper}}}$.

The physical meaning of $U_{B_{\text{upper}}}$ is given as follows. The token must visit each station at least once every D_A units of time. During each rotation of the token, NT_t units of time are used to transfer the token, and under the highest generating rate of class A packets, T_A units of time are used to transmit class A packets. Thus, stations can use up to $D_A - T_A - NT_t$ units of time to transmit class B packets, resulting in a guaranteed throughput $U_{B_{\text{upper}}} = 1 - T_A/D_A - NT_t/D_A$ for class B packets.

The following theorem shows that this upper bound is reached by the optimal TPP.

Theorem 4: The optimal TPP has a guaranteed throughput for class B packets

$$U_{B_{\text{opt}}} = U_{b_{\text{upper}}} = 1 - (T_A + NT_t)/D_A .$$

Proof: Suppose station k has as an infinite length queue of class B packets. As shown in Fig. 3, the throughput for class B packets during the time period $T(m)$ is

$$U_B(m) = (T(m) - NT_t - T_A(m))/T(m) .$$

As proved in Theorem 3, the token rotation time $T(m) \leq D_A$. Thus,

$$U_B(m) \geq (T(m) - NT_t - T_A(m))/D_A .$$

Also, $T(m) = NT_t + T_A(m) + \sum_{i=m-N+1}^{m-1} \tau_B(i) + \tau_B(m)$. Since station k has an infinite length queue of class B packets,

$$\begin{aligned}\tau_B(m) &= \max\{0, D_A - T_A + T_A(m) + \tau_B(m-N) - T\} \\ &\geq D_A - T_A + T_A(m) + \tau_B(m-N) - T .\end{aligned}$$

On the other hand,

$$\sum_{i=m-N+1}^{m-1} \tau_B(i) = T - NT_t - T_A(m) - \tau_B(m-N) .$$

Thus,

$$\sum_{i=m-N+1}^{m} \tau_B(i) \geq D_A - T_A - NT_t .$$

and

$$T(m) \geq D_A - T_A + T_A(m) .$$

We get

$$U_B(m) \geq (D_A - NT_t - T_A)/D_A = U_{B_{\text{upper}}} .$$

Averaging $U_B(m)$ over m,

$$U_B = \lim_{k\to\infty} \sum_{m=1}^{k} U_B(m)/k \geq U_{B_{\text{upper}}} .$$

On the other hand, U_B cannot be larger than $U_{B_{\text{upper}}}$ for any token passing protocol. We conclude that the guaranteed throughput for class B packets of the optimal TPP $U_{B_{\text{opt}}} = U_{B_{\text{upper}}}$. □

The optimal token passing protocol is easy to implement. One bit in the header of every packet is needed to distinguish a class A packet from a class B packet. On a standard IEEE 802.4 token bus, two modifications are needed: 1) Timer2 is reset when a station finishes transmitting its class B packets (in the standard IEEE 802.4, this is done when a station finishes transmitting class A packets). 2) When a station senses a class A packets being transmitted, it stops the counting of its Timer2. The counting of Timer2 is resumed when the station senses the transmission of class B packets or the token.

The optimal TPP can also be implemented on a standard FDDI token ring network with the same two modifications. In this case, however, "a station senses the transmission of class A packets" means that the station is forwarding or transmitting class A packets.

In addition to providing as much throughput for class B packets as possible, it is sometimes desirable that the network provides fairness to class B traffic [8]. In other words, if all stations have many class B packets to transmit, each station should have equal throughput for class B packets.

If we set $T_R(m) = D_A - T_A + T_A(m)$ in our optimal token passing protocol, i.e., resetting Timer2 at the time when a station receives the token instead of when a station passes the token, it can be proved in the same way as [8] that the protocol provides fairness to class B traffic. The modified optimal TPP continues to provide a higher throughput for class B packets than IEEE 802.4, since $T_R(m) \geq D_A - T_A$ still holds. However, the guaranteed throughput for class B packets becomes

$$\begin{aligned}U_{B_{m\ \text{opt}}} = 1 &- T_A/D_A \\ &- 2(1 - T_A/D_A)NT_t/(D_A - T_A + NT_t) .\end{aligned}$$

Clearly, $U_{B_{802.4}} \leq U_{B_{m\ \text{opt}}} < U_{B_{\text{opt}}}$. So the fairness of the modified optimal protocol is achieved at the cost of some loss in the throughput for class B packets.

TABLE I
THE GUARANTEED THROUGHPUT U_B FOR NONTIME-CONSTRAINED PACKETS UNDER DIFFERENT PROTOCOLS

U_A	IEEE 802.4	Optimal TTP	Symmetric IEEE 802.4
0.00	0.65	0.79	0.79
0.01	0.64	0.78	0.78
0.05	0.60	0.74	0.73
0.10	0.54	0.69	0.66
0.25	0.37	0.56	0.47
0.50	0.06	0.29	0.08
0.55	0.00	0.24	0.00
0.75	X	0.04	X
0.80	X	X	X

TABLE II
THE GUARANTEED THROUGHPUT U_B FOR NONTIME-CONSTRAINED PACKETS UNDER DIFFERENT PROTOCOLS

U_A	IEEE 802.4	Optimal TTP	Symmetric IEEE 802.4
0.00	0.82	0.90	0.90
0.01	0.81	0.89	0.89
0.05	0.76	0.85	0.84
0.10	0.71	0.80	0.79
0.25	0.54	0.65	0.62
0.50	0.25	0.40	0.30
0.55	0.19	0.35	0.23
0.65	0.05	0.25	0.06
0.75	X	0.15	X
0.90	X	0.00	X

The fairness of the modified protocol is achieved by restricting a station's transmission of class B packets if it had transmitted a lot of them during the previous token's visit. Actually, the protocol still provides fairness to class B traffic even is this restriction is applied once every n token visits, $n > 1$. In other words, set $T_R(m) := D_A - T_A + T_A(m) + \tau_B(m - N)$ during all but one of n consecutive token's visit to the station, and set $T_R(m) := D_A - T_A + T_A(m)$ only once every n consecutive token's visit to the station. It is expected that $U_{B_{m\ \mathrm{opt}}} \rightarrow U_{B_{\mathrm{opt}}}$ as $n \rightarrow \infty$.

V. Performance Comparison

We present in this section two numerical examples to demonstrate the performance comparison of the optimal token passing protocol and the standard IEEE 802.4 token bus (as shown in Section III, the FDDI token bus cannot be better than IEEE 802.4). The criterion used for the comparison is the guaranteed throughput for class B packets as shown in Theorem 1 and Theorem 3.

The first example uses the normal IEEE 802.4 token bus configuration [3]: bandwidth of the bus = 10 Mb/s, number of stations $N = 50$, length of the token = 184 b, one way propagation delay = 5.0 μs, token passing time = 83.5 μs. We assume the required access delay for class A packets $D_A = 20$ ms.

Table I shows the guaranteed throughput for class B packets that the above network can provide using medium access protocols and under different class A traffic conditions $U_A = T_A/D_A$. "X" means the network cannot provide any guaranteed throughput for class B packets. The last column of the table shows the throughput for class B packets of the IEEE 802.4 under symmetric class B traffic condition, i.e., when every station has an infinite length queue for class B packets.

As expected, the optimal TPP outperforms the IEEE 802.4. It uses the least amount of bandwidth $NT_t/D_A = 0.21$ to transmit the token and assigns the rest of the bandwidth to the transmission of class A and class B packets. This can be seen from the fact that $U_A + U_B$ always equals 0.79. The network cannot meet the deadline for class A packets when $U_A \geq 0.80$ since the inequality $D_A - T_A - NT_t > 0$ no longer holds. We also see that the optimal TPP outperforms the symmetric IEEE 802.4. This means that the throughput of class B packets in an optimal TPP network where only one station has class B packets to transmit is even larger than that of an IEEE 802.4 network where all stations have class B packets to transmit. This shows that our optimal TPP is quite insensitive to the distribution of class B packets.

The second example used the normal FDDI token ring configuration [6]: effective bandwidth of the ring = 100 Mb/s, number of station $N = 1000$ with a 200 km ring, length of the token = 22 symbols, token passing time = 2 μs. We assume the required access delay for class A packets is $D_A = 20$ ms. The results obtained for this example are similar to that of Example 1 and are shown in Table II.

VI. Conclusion

Addressed in this paper is the problem of handling both time-constrained (class A) and nontime-constrained (class B) packets in local area networks. For the existing LAN protocols, the IEEE 802.4 token bus and the FDDI token ring, we showed the way of setting protocol parameters to maximize the guaranteed throughput for class B packets while meeting access deadlines of class A packets. We also showed how the timer mechanisms of these two widely used protocols can be easily modified to derive an optimal medium access protocol which renders the highest guaranteed throughput for class B packets among all token passing protocols. Thus, the results in this paper are useful for either tuning existing communication protocols or building of new one. Numerical examples are also given for the purpose of performance comparison.

Our future work will focus on the generalization of the model to include different types of time-constrained packets (e.g., packets with different delay requirements) and applications to real-life problems.

References

[1] J. F. Kurose, M. Schwartz, and Y. Yemini, "Multiple-access protocols and time-constrained communication," *Comput. Sur.*, vol. 16, no. 1, pp. 43–70, 1984.
[2] ——, "Controlling window protocols for time-constrained communication in multiple access networks," *IEEE Trans. Commun.*, vol. 36, pp. 41–49, Jan. 1988.
[3] A. P. Jayasumana and G. G. Jaysumana, "On the use of the IEEE 802.4 token bus in distributed real-time control systems," *IEEE Trans. Industrial Electron.*, vol. 36, pp. 391–397, 1989.
[4] R. M. Gorur and A. C. Weaver, "Setting target rotation times in an IEEE token bus network," *IEEE Trans. Industrial Electron.*, vol. 35, pp. 366–371, 1988.
[5] D. Dykeman and W. Bux, "Analysis and tuning of the FDDI media access control protocol," *IEEE J. Select. Areas Commun.*, vol. 35, pp. 997–1010, Aug. 1988.
[6] A. C. Weaver, "Local area networks and busses—An analysis," Tech. Rep., Flight Data Syst., NASA–Johnson Space Center, 1986.

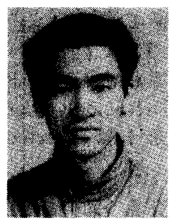
[7] K.C. Sevcik and M.J. Johnson, "Cycle time properties of the FDDI token ring protocol," Tech. Rep. CSRI-179, Comput. Sci. Res. Inst., Univ. Toronto, Toronto, Ont., Canada, 1986.
[8] M.J. Johnson, "Fairness of channel access for non-time-critical traffic using the FDDI token ring protocol," in *Proc. Seminar Real-Time Local Area Networks,* 1986, pp. 145–157.

Kang G. Shin (S'75–M'78–SM'83–F'92) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1970, and both the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 1976 and 1978, respectively.

He is Professor and Associate Chair of Electrical Engineering and Computer Science (EECS), The University of Michigan, Ann Arbor, Michigan.

He has been authored/coauthored over 200 technical papers (about 110 of these in archival journals) and several book chapters in the areas of distributed real-time computing and control, fault-tolerant computing, computer architecture, and robotics and automation. In 1987, he received the Outstanding IEEE Transactions on Automatic Control Paper Award for a paper on robot trajectory planning. In 1989, he also received the Research Excellence Award from The University of Michigan. In 1985, he founded the Real-Time Computing Laboratory, where he and his colleagues are currently building a 19-node hexagonal mesh multicomputer, called HARTS, to validate various architectures and analytic results in the area of distributed real-time computing. From 1978 to 1982 he was on the faculty of Rensselaer Polytechnic Institute, Troy, New York. He has held visiting positions at the U.S. Airforce Flight Dynamics Laboratory, AT&T Bell Laboratories, Computer Science Division within the Department of Electrical Engineering and Computer Science at UC Berkeley, and International Computer Science Institute, Berkeley, CA.

Dr. Shin was the Program Chairman of the 1986 IEEE Real-Time Systems Symposium (RTSS), the General Chairman of the 1987 RTSS, the Guest Editor of the 1987 August special issue of IEEE TRANSACTIONS ON COMPUTERS on Real-Time Systems, and was a Program Co-Chair for the 1992 *International Conference on Parallel Processing.* He chaired the IEEE Technical Committee on Real-Time Systems during 1991–1993, was a Distinguished Visitor of the Computer Society of the IEEE during 1989–1993, is an Editor of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED COMPUTING, and an Area Editor of *International Journal of Time-Critical Computing Systems.*



Qin Zheng (S'89–M'93) received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology of China in 1982 and 1985, respectively, and Ph.D. degree in electrical engineering from the University of Michigan in 1993.

He joined Mitsubishi Electric Research Labs, Inc., Cambridge Research Center in 1993 and is currently working on high-speed computer networks and distributed real-time industrial systems.