MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Synchronous Bandwidth Allocation in FDDI Networks

Qin Zheng

TR93-16 December 1993

Abstract

It is well-known that an FDDI token ring network provides a guaranteed throughput for synchronous messages and a bounded medium access delay for each node/station. However, this fact alone cannot effectively support many real-time applications that require the timely delivery of each critical message. The reason for this is that the FDDI guarantees a medium access delay bound to nodes, but not to messages themselves. The message-delivery delays may exceed the medium-access delay bound even if a node transmits synchronous messages at a rate not greater than the guaranteed throughput. We solve this problem by developing a synchronous bandwidth allocation (SBA) scheme which calculates the synchronous bandwidth necessary for each application to satisfy its message-delivery delay requirement. The result obtained in this paper is essential for effective use of the FDDI token ring networks in supporting such real-time communications as digital video/audio transmissions and distributed control/monitoring.

Mitsubishi Electric Research Laboratories
Cambridge Research Center

Technical Report 93-16 **August 20 1993**

SYNCHRONOUS BANDWIDTH ALLOCATION IN FDDI NETWORKS

by

Qin Zheng

Mitsubishi Electric Research Laboratories, Inc.
Cambridge Research Center
Cambridge, MA 02139

Abstract

It is well-known that an FDDI token ring network provides a guaranteed throughput for synchronous messages and a bounded medium access delay for each node/station. However, this fact alone cannot effectively support many real-time applications that require the timely delivery of *each* critical message. The reason for this is that the FDDI guarantees a medium access delay bound to nodes, but not to messages themselves. The message-delivery delays may exceed the medium-access delay bound even if a node transmits synchronous messages at a rate not greater than the guaranteed throughput. We solve this problem by developing a synchronous bandwidth allocation (SBA) scheme which calculates the synchronous bandwidth necessary for each application to satisfy its message-delivery delay requirement. The result obtained in this paper is essential for effective use of the FDDI token ring networks in supporting such real-time communications as digital video/audio transmissions and distributed control/monitoring.

To be submitted to *IEEE Trans. on Networking.*

201 Broadway
Cambridge Massachusetts 02139

Publication History:-

1. First printing, TR 93-06, May 28, 1993

1 Introduction

The Fiber Distributed Data Interface (FDDI) is a proposed ANSI standard for a 100 Mbps (million bits per second) token ring network using a fiber-optic medium [1, 2, 3, 4]. Thanks to its high transmission speed, the FDDI alleviates the bandwidth saturation problem of the current 10 Mbps Ethernet and the 4 or 16 Mbps IEEE 802.5 Token rings. The synchronous transmission capacity of the FDDI also makes it capable of supporting real-time applications like digital video/audio transmissions and distributed control/monitoring.

The synchronous transmission capacity of the FDDI is provided to each node/station in the form of two different guarantees: a bounded medium-access delay and a minimum throughput for synchronous traffic. Specifically, if the target token rotation time of an FDDI network is set to $TTRT$ and the synchronous transmission time of node i is set to h_i, then the time node i must wait for a chance to transmit its synchronous messages is bounded by $2 \times TTRT$, and on the average, the node is guaranteed to have a bandwidth of $h_i/TTRT \times 100$ Mbps to transmit its synchronous messages. These two guarantees make FDDI networks capable of supporting synchronous traffic, but, as discussed below, they are not sufficient for most time-critical applications.

A real-time application usually requires that each of its messages be delivered timely given a prespecified message-generation rate. But an FDDI network guarantees throughput and delay bounds individually in isolation. When a node transmits synchronous messages at the guaranteed throughput rate $h_i/TTRT \times 100$ Mbps, it is not guaranteed that all of the messages will have a delay bound $2 \times TTRT$. To understand the problem better, let's consider the following example video channel:

- The source of the channel generates a video frame every T units of time. For full motion video, $T = 33$ms.
- The time needed to transmit a maximum-size video frame at a 100 Mbps transmission rate is C units of time.
- For a smooth real-time video at the destination/receiver, each frame is required to be delivered to the destination within d units of time after its generation. We assume $d = T$ in this example.

With the parameters specified above, the maximum traffic over the video channel is $C/T \times$ 100 Mbps. Thus the throughput requirement would be satisfied if the FDDI is configured such that $h_i/TTRT = C/T$. Also the delay requirement would be satisfied if $2 \times TTRT = T$. Together we get $TTRT = T/2$ and $h_i = C/2$. From the Medium Access Control (MAC) protocol of the FDDI, h_i is the maximum time node i is allowed to transmit synchronous messages once it gets the token. Thus, $h_i = C/2$ implies that a maximum-size video frame would take two token's visits to get transmitted. Since $TTRT := T/2$, one token rotation time could be as large as T, and thus, a maximum-size frame would not be transmitted within a delay bound T in the worst case.

To solve the above problem one can either use a smaller $TTRT$ to reduce the medium access-delay bound or use a larger h_i to increase the synchronous bandwidth assigned to the station. The first method is not desirable for several reasons: (1) $TTRT$ is usually set at the ring initialization time and thus, it would be inconvenient to change $TTRT$ whenever a new application is created; (2) due to the token passing overhead and the ring latency, the overall ring efficiency would deteriorate if $TTRT$ is set too small; (3) reducing $TTRT$ also increases the synchronous bandwidth assigned to the station. This paper addresses the second approach, i.e., set h_i appropriately. Specifically, we will develop a synchronous bandwidth allocation (SBA) scheme which, given a network target token rotation time, $TTRT$, and an application traffic specification, determines the synchronous transmission time, h_i, of the node to guarantee all synchronous messages to be transmitted within the user-requested delay bound. The station management standard SMT 7.2 of the FDDI describes SBA facilities [4], suggesting *how* synchronous bandwidth may be allocated to a node, but it does not indicate *how much* of synchronous bandwidth needs to be assigned for a specific application. Clearly, the FDDI's capacity of supporting synchronous traffic cannot be effectively used without a proper SBA scheme.

Recently, the importance of SBA has been drawing considerable interest. Agrawal *et al.* [5] proposed a normalized proportional SBA scheme which was proven to be able to support any set of synchronous channels with a total peak signal rate no more than 33% of the ring bandwidth. But this scheme has the following disadvantages.

1. It can be used only for those applications where the user-requested message-delay bound d equals the message inter-generation period T. This limits the type of applications that a network can support.

2. The scheme is not optimal in the sense that it does not assign the minimum synchronous bandwidth necessary for each application, thus reducing the number of synchronous-traffic applications that a network can support.

3. It is a *global* SBA scheme in that the allocation/deallocation of synchronous bandwidth to a node would require to change the synchronous bandwidths previously assigned to all other nodes. A global SBA scheme complicates the implementation of synchronous bandwidth allocation.

To improve the scheme in [5], Chen *et al.* proposed an optimal SBA scheme [6]. However, it still suffers the limitation of $d = T$ and being a global scheme. Besides, it uses an iterative algorithm for the calculation of the optimal bandwidths which may, in theory, need an infinite number of steps to converge.

In this paper we propose an SBA scheme which does not require $d = T$ and is optimal in most cases. The calculation of the optimal bandwidths can be done in just one step. Further, allocation/deallocation of a synchronous bandwidth to a node does not require to change the synchronous bandwidths assigned to the other nodes, thus making the synchronous bandwidth allocation easy to implement.

This paper is organized as follows. Section 2 reviews the MAC protocol of the FDDI and its relevant properties. A new SBA scheme is proposed and discussed in Section 3. The paper concludes with Section 4.

2 Preliminaries

For convenience of discussions, we review the FDDI's MAC protocol and some of its properties in this section. The FDDI's MAC protocol [3] is summarized below.

Protocol 2.1 (MAC of the FDDI) .

P1: Suppose there are N active nodes in a ring which are numbered from 0 to $N-1$. As part of an FDDI ring initialization process, each node declares a Target Token Rotation

Time ($TTRT$). The smallest among them is selected as the ring's $TTRT$. Each node which supports synchronous traffic is then assigned a portion of the $TTRT$ to transmit its synchronous messages. Let h_i, called the *synchronous bandwidth allocation*, denote the portion of $TTRT$ that node i is assigned to transmit its synchronous messages.

P2: Each node has two internal timers: the token–rotation–timer (TRT) and the token–holding–timer (THT). The TRT always counts up and a node's THT counts up only when the node is transmitting asynchronous messages. If a node's TRT reaches the $TTRT$ before the token arrives at the node, the TRT is reset to 0 and the token is marked as late by incrementing the node's late count L_c by one. To initialize the timers at different nodes, no messages are allowed to be transmitted during the first token rotation after the ring initialization and L_c's are set to 0.

P3: Only the node that has the token is eligible to transmit messages. The message transmission time is controlled by the node's timers, but an in-progress message transmission will not be interrupted until its completion. When a node i receives the token, it does the following:

P3.1: If $L_c > 0$, set $L_c := L_c - 1$ and $THT := TTRT$. Otherwise, set $THT := TRT$ and $TRT := 0$.

P3.2: If node i has synchronous messages, it transmits them for a time period up to h_i or until all the synchronous messages are transmitted,[1] whichever occurs first.

P3.3: If node i has asynchronous messages, it transmits them until the THT counts up to the $TTRT$ or all of its asynchronous messages are transmitted, whichever occurs first.

P3.4: Node i passes the token to the next node $(i+1) \bmod N$. □

Let T_{ring} denote a ring's latency plus the token passing overhead, which is the time needed to circulate the token around the ring once without transmitting any message, and T_p denote the time needed to transmit a maximum-size asynchronous message. Then, the parameters of the FDDI's MAC protocol must satisfy the following protocol constraint:

$$\sum_{i=0}^{N-1} h_i \le TTRT - T_{ring} - T_p. \tag{1}$$

The physical meaning of the above inequality is that the summation of the assigned synchronous bandwidths over the nodes in the network should not exceed the effective ring bandwidth. Violation of this constraint would make the ring unstable and oscillate between "claiming" and "operational" [4]. Under this protocol constraint, a well-known fact about the FDDI is that the worst-case token rotation time is bounded by $2 \times TTRT$, and the average token rotation time is bounded by $TTRT$ [7]. A more general result was obtained by Agrawal *et al.* in a recent paper [5] as stated below.

[1]Though it is not in the standard, we assume that a node always transmits its synchronous traffic first for better synchronous performance of the network and simplicity of analysis.

Lemma 2.1 (Worst-case token rotation times) .
Under the protocol constraint (1), the time elapsed between any n consecutive token's visits to node i is bounded by $n \times TTRT - h_i$.

Once node i gets the token, it is given up to h_i units of time to transmit its synchronous messages. Let $\lfloor x \rfloor$ be the largest integer which is *equal to or smaller* than x, and $\lceil x \rceil$ be the smallest integer which is *larger* than x. The following lemma gives a lower bound of time that node i is allowed to transmit its synchronous messages during a time period t [5].

Lemma 2.2 (Synchronous transmission time) .
Under the protocol constraint (1) of the FDDI, node i has at least $\lfloor t/TTRT - 1 \rfloor \cdot h_i$ *units of time to transmit its synchronous messages during a time period t. This lower bound is reached when* $\lceil t/TTRT \rceil \cdot TTRT - t \geq h_i$.

Lemmas 2.1 and 2.2 are the best known synchronous properties of the FDDI to date. We will improve Lemma 2.2 and derive a new SBA scheme in the next section.

3 A New SBA Scheme

As discussed in the Introduction, an important feature of real-time communication is that each message must be delivered to its destination within a pre-specified delay bound. Due to the limited network transmission bandwidth, this requirement cannot be satisfied without some information on message-generation characteristics.

We use two parameters, T and C, to describe a message-generation pattern, where T is the *minimum* message inter-generation time and C is the *maximum* message-transmission time (i.e., the time needed to transmit a maximum-size message). It is reasonable to assume prior knowledge of these parameters for many real-time applications, such as interactive voice/video transmission and real-time control/monitoring. For applications where the traffic pattern is less predictable, the estimated values of T and C could be used. A process may exceed its pre-specified maximum message size and/or message generation rate at the risk that these messages may not be delivered within the pre-specified delay bound, but this particular process will not affect the guarantees of other applications.

Together with the requested message-delivery delay bound d and the address of the source node S, we use the concept of *real-time channel* [8] for real-time communication services. A real-time channel is described by a 4-tuple $\tau = (T, C, d, s)$ which guarantees each message generated at the source node s to be delivered sequentially to one or more destination nodes in a time period $\leq d$, as long as the message inter-generation time is $\geq T$ and the message transmission time is $\leq C$.

A real-time channel is a convenient way to achieve real-time communication. Users can set up channels with adequate bandwidths and delay bounds for their applications. This is in sharp contrast to the conventional circuit-switched transmission where users have few choices on the bandwidth and quality of the circuits. We will in this paper deal with the implementation of real-time channels in FDDI networks only. Readers are referred to [8, 9, 10, 11, 12, 13] for discussions on real-time channels in point–to–point networks.

A set of real-time channels is said to be *establishable* over an FDDI network if the requested message-delivery delay bound of each channel can be guaranteed by properly setting the parameters of the FDDI's MAC protocol. From Protocol 2.1, the user-adjustable parameters are

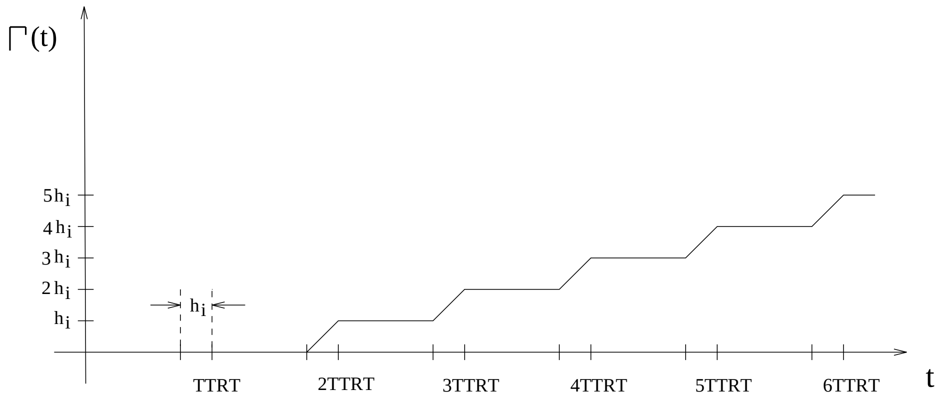

Figure 1: Worst-case synchronous transmission time , (t).

the $TTRT$ and h_i's. The $TTRT$ is usually set at the network initialization time and does not change frequently. It determines the minimum message-delay bound, $d_{min} = 2 \times TTRT$, that the network can guarantee. Any channel request with a delay bound smaller than d_{min} will be rejected. With a given $TTRT$, the synchronous bandwidth allocated to node i is determined by the value of h_i. Thus, an SBA scheme determines the values of h_i's to accommodate real-time channels. An SBA scheme is said to be *feasible* with respect to a set of real-time channels if it can guarantee the requested delay bounds of all the channels in the set. An SBA scheme is said to be *optimal* if it is always feasible whenever there exists a feasible SBA scheme. The advantage of an optimal SBA scheme is the full-utilization of the FDDI's synchronous transmission capacity since a set of real-time channels rejected by an optimal SBA scheme cannot be established with any other SBA schemes.

We derive in this section the conditions for establishing real-time channels over an FDDI network. From these conditions, a new SBA scheme will be developed which has many advantages over the SBA schemes in [5, 6].

Let , (t) denote the time that a node in the worst case is allowed to transmit its synchronous messages during a time period t. Lemma 2.2 gives a lower bound of , (t) for node i. We improve Lemma 2.2 by calculating the exact value of , (t) as follows.

Lemma 3.1 (Worst-case synchronous transmission time) *.*
Under the protocol constraint (1) of the FDDI, node i in the worst case has

$$, (t) = \lfloor t/TTRT - 1 \rfloor h_i + \delta(t)$$

units of time to transmit its synchronous messages during a time period t, where $\delta(t)$ is calculated as

$$\delta(t) = \begin{cases} 0 & if \ \lceil t/TTRT \rceil TTRT - t \geq h_i \ or \ t \leq TTRT \\ t - (\lceil t/TTRT \rceil TTRT - h_i) & otherwise. \end{cases}$$

Proof: , (t) is plotted in Fig. 1. Its correctness can be seen from Lemma 2.1. In the worst case, [2], node i would first wait $2 \times TTRT - h_i$ units of time to get the token. Once it gets the

[2]The worst case occurs when (1) all usable ring bandwidth is assigned to nodes as synchronous bandwidths, (2) no messages are transmitted during the previous token rotation, and (3) all nodes use the maximum times allowed to transmit their synchronous and asynchronous messages during the current token rotation.

token, it has h_i units of time to transmit its synchronous messages. This proves the correctness of , (t) for $t \leq 2 \times TTRT$. By Lemma 2.1, the following worst-case token inter-arrival time at node i would be $TTRT$. This proves the correctness of , (t) for $t > 2TTRT$. □

It should be noted that the synchronous transmission time in the above Lemma is calculated under the worst-case situation that all usable ring bandwidth is assigned to nodes as synchronous bandwidth. A node would be able to have more synchronous transmission time if only a part of the usable ring bandwidth is assigned as synchronous bandwidth. But using such a "better" calculation for synchronous bandwidth allocation would reduce the total amount of synchronous traffic that a network can support, thus is undesirable.

Suppose no two real-time channels have the same source node and the synchronous transmission time of a node is used for real-time channel messages only. Then from Lemma 3.1, we have the following necessary and sufficient condition for the establishment of a real-time channel over an FDDI network.

Theorem 3.1 (Channel establishment conditions over the FDDI) .
A real-time channel $\tau = (T, C, d, s)$ can be established over an FDDI network under the protocol constraint (1) if and only if

$$\forall t \geq 0, \quad \lceil (t-d)/T \rceil^{+} C \leq \, , (t), \tag{2}$$

where , (t) *is calculated from Lemma 3.1 with $i = s$, and $\lceil x \rceil^{+} = n$ if $n-1 \leq x < n$, $n = 1, 2, \cdots$, and $\lceil x \rceil^{+} = 0$ for $x < 0$.*

Proof of the necessary condition: Suppose node s does not have any message of channel τ at time $t = 0$. Then, $\forall t > 0$ a necessary condition for no messages to miss their deadlines in $[0, t]$ is that the amount of time, $\tau(t)$, needed to transmit all those messages generated during $[0, t]$ by channel τ with deadlines $\leq t$ is not greater than , (t), the time that node s in the worst-case is allowed to transmit its synchronous messages. Since the minimal message inter-generation time of channel τ is T, there are at most $\lceil (t-d)/T \rceil^{+}$ messages generated by channel τ during $[0, t]$ with deadlines $\leq t$, which take at most $\lceil (t-d)/T \rceil^{+} C$ units of time to be transmitted. Thus, the maximum value of $\tau(t)$ is $\lceil (t-d)/T \rceil^{+} C$. This proves the necessary condition.

Proof of the sufficient condition: We prove this by contradiction. Suppose a message misses its deadline at time t_1, meaning that at least one message with deadline $\leq t_1$ has *not* been transmitted by t_1. Then there must exist $t' < t_1$ such that during the time period $[t', t_1]$, node i uses *all* of its allowed synchronous transmission time for channel τ's messages. Let t_0 be the smallest such t', then there are no messages with deadlines $\leq t_1$ queued at the link at time t_0^-. Thus, in the time period $[t_0, t_1]$, node i uses all its synchronous transmission time transmitting only those messages of channel τ which are generated during $[t_0, t_1]$ with deadlines $\leq t_1$. Based on the same reasoning as the proof of the necessary condition, the maximum amount of time needed to transmit these messages is $\tau(t_1 - t_0) = \sum_{i=1}^{n} \lceil (t_1 - t_0 - d)/T \rceil^+ C$. Since one message misses its deadline at t_1, this $\tau(t_1 - t_0)$ must be larger than , $(t_1 - t_0)$, that is, $\lceil (t_1 - t_0 - d)/T \rceil^+ C >$, $(t_1 - t_0)$. By letting $t = t_1 - t_0$, the above inequality contradicts the condition that $\forall t \geq 0, \quad \lceil (t - d)/T \rceil^+ C \leq$, (t). □

It is difficult to calculate the minimum synchronous bandwidth allocation (i.e., h_s) needed for a real-time channel from Theorem 3.1 since inequality (2) must be checked in an interval of infinity length. Fortunately, with the following theorem, one can easily calculate the required minimum synchronous bandwidth allocation in most cases (i.e., when $2 \times TTRT \leq d \leq T + TTRT$, or $d \geq T + 2 \times TTRT$), and the upper bound of h_s for other cases.

Theorem 3.2 (Calculation of h_s) .
The minimum h_s required to establish a real-time channel (T, C, d, s) can be calculated as

1. *For $2 \times TTRT \leq d \leq T + TTRT$,*

$$h_s = \begin{cases} C/p & if\ q \geq C/p \\ (C+q)/(1+p) & if\ q < C/p \end{cases}$$

   where $p = \lfloor d/TTRT - 1 \rfloor$ and $q = \lceil d/TTRT \rceil TTRT - d$.

2. *For $d \geq T + 2 \times TTRT$,*

$$h_s = (TTRT/T)C$$

3. *For $T + TTRT < d < T + 2 \times TTRT$ and $T \geq TTRT$,*

$$h_s \leq \begin{cases} C/p_0 & if\ q_0 \geq C/p_0 \\ (C+q_0)/(1+p_0) & if\ q_0 < C/p_0 \end{cases}$$

   where $p_0 = \lfloor T/TTRT \rfloor$ and $q_0 = \lceil T/TTRT \rceil TTRT - T$.

4. For $2 \times TTRT < d < T + 2 \times TTRT$ *and* $T < TTRT$*,*

$$h_s \leq \lceil TTRT/T \rceil C.$$

Proof: We prove four parts of Theorem 3.2 one by one.

1. Notice that the left-hand of inequality (2) is a piecewise constant function which changes only at points $t = d + kT,\ k = 0, 1, \cdots$, with the value $(k+1)C$, and the right-hand of inequality (2) is a monotone increasing function. So we only need to check inequality (2) at these discrete points.

   For $t = d$, from Lemma 3.1, $,\ (d) = ph_s + \delta(d)$ where

   $$\delta(d) = \begin{cases} 0 & if\ q \geq h_s \\ h_s - q & if\ q < h_s. \end{cases}$$

   It is easy to check that setting h_s in the way specified by Theorem 3.2 makes $,\ (d) = C$, proving that it is the minimum h_s which satisfies inequality (2) of Theorem 3.1 at point $t = d$. From Fig. 1, we see that $\forall t \geq 0,\ ,\ (t+T) - ,\ (t) \geq ,\ (TTRT + T) - ,\ (TTRT) = ,\ (TTRT + T)$. Since $d \leq TTRT + T$, $,\ (t+T) - ,\ (t) \geq ,\ (d) = C$. Then, for any positive integer k,

   $$\begin{aligned} ,\ (d + kT) &= \sum_{i=1}^{k}(,\ (d + iT) - ,\ (d + iT - T)) + ,\ (d) \\ &\leq \sum_{i=1}^{k} C + C \\ &= (k+1)C. \end{aligned}$$

   This proves that inequality (2) of Theorem 3.1 is satisfied for all $t \geq 0$.

2. It is easy to see that the left-hand side of inequality (2) $\leq ((t-d)/T + 1)C$ and the right-hand side of inequality (2) $\geq (h_s/TTRT)(t - 2TTRT)$. For $d \geq T + 2 \times TTRT$ and $h_s = (TTRT/T)C$, we get $((t-d)/T + 1)C \leq (h_s/TTRT)(t - 2TTRT)$. Thus, inequality (2) is satisfied.

3. Since $T \geq TTRT$, $T + TTRT \geq 2 \times TTRT$. Then the first part of Theorem 3.2 can be used to calculate h_s for $d = T + TTRT$. Clearly, such calculated h_s is an upper bound for h_s when $d > T + TTRT$.

4. During any $TTRT$ units of time, the left-hand side of inequality (2) increases by at most $\lceil TTRT/T \rceil C$. Thus for $d \geq 2 \times TTRT$ and $h_s = \lceil TTRT/T \rceil C$, inequality (2) is always satisfied. $\square$

From Theorem 3.2, we have the following channel establishment algorithm.

Algorithm 3.1 (Channel establishment over FDDI) .
Suppose $n-1$ *real-time channels* $\tau_i = (T_i, C_i, d_i, s_i)$, $i = 1, \cdots, n-1$ *have already been established over an FDDI ring. Then a new channel* $\tau_n = (T_n, C_n, d_n, s_n)$ *can be established with the following steps.*

Step 1: *Calculate* h_{s_n} *from Theorem 3.2.*

Step 2: *If the protocol constraint (1) is satisfied, set the synchronous bandwidth allocation of* s_n *to* h_{s_n} *and establish channel* τ_n. *Otherwise, the channel establishment request is rejected.* $\square$

Some discussions on the above algorithm are in order.

1. For $T + TTRT < d < T + 2 \times TTRT$, Theorem 3.2 gives only an upper bound of the minimum h_s. To see how tight this upper bound is, notice that a necessary condition for the establishment of a real-time channel over an FDDI network is that the assigned synchronous bandwidth ($h_s/TTRR \times 100$ Mbps) must not be smaller than the expected signal bandwidth ($C/T \times 100$ Mbps) over the channel. This means that $h_s \geq (TTRT/T)C$ is a necessary condition and $(TTRT/T)C$ is a lower bound of the required h_s. Thus the difference between the upper bounds given in Theorem 3.2 and the minimum h_s is bounded by

$$\beta = \begin{cases} (1/\lfloor x \rfloor - 1/x)C & if\ x \geq 1 \\ (\lceil 1/x \rceil - 1/x)C & if\ x < 1 \end{cases}$$

where $x = T/TTRT$.

From this we see that the upper bound obtained in Theorem 3.2 will never exceed twice the minimum h_s. Another result is that the upper bound given in from Theorem 3.2 is actually the minimum h_s when T is a multiple of $TTRT$ or $TTRT$ is a multiple of T.

2. Algorithm 3.1 is an optimal SBA scheme when a minimum h_s can be obtained from Theorem 3.2. This includes the following four situations:

   - $2 \times TTRT \leq d \leq T + TTRT$,
   - $d \geq T + 2 \times TTRT$,
   - T is a multiple of $TTRT$,
   - $TTRT$ is a multiple of T.

   We believe that the above situations include most real-time communication applications. For example, communications in distributed control/monitoring systems usually have tight delay requirements ($d \leq T + TTRT$), and video/audio communications can often tolerate larger delays ($d \geq T + 2 \times TTRT$). Thus for most applications, the synchronous bandwidth allocation resulting from Algorithm 3.1 is optimal.

3. Comparing with the SBA schemes of [5] and [6], Algorithm 3.1 has the following advantages.

   Generality: The SBA schemes of [5, 6] can establish real-time channels with $d = T$ only, while Algorithm 3.1 can establish channels of arbitrary parameters, i.e., $d \leq T$ or $d > T$. This extension is very important in practice since for many applications, especially those in real-time control/monitoring systems, the required delay bound d is usually smaller than the message inter-generation period T. Real-time channels with $d > T$ are also useful for multimedia applications. Thus restricting $d = T$ would greatly limit a network's ability and effectiveness in supporting real-time communications.

   Optimality: The SBA scheme of [5] is not optimal, even under the restrictive assumption of $d = T$. Thus a real-time channel establishment request may be rejected even if it can be established using a different scheme. The SBA scheme of [6] is optimal under the restrictive assumption of $d = T$ and requires complex computations. By contrast, Algorithm 3.1 is optimal for a much wider range of d (which subsumes the special case of $d = T$ in [5, 6]).

   Simplicity: The SBA schemes of [5, 6] are *global* schemes in the sense that the addition/removal of a channel or change of the parameters of a channel would require adjustment of the synchronous bandwidth allocations of all nodes in the network. This requires a complex SBA implementation. By contrast, Algorithm 3.1 needs only *local* parameter adjustment, thereby making it far easier to implement than those in [5, 6].

As an application example of Algorithm 3.1, we calculate the synchronous bandwidth needed for establishing the following video channel in an FDDI network. Suppose the video frame-generation period $T = 33$ ms (30 frames/second), the transmission time of a maximum frame is 1 ms (100 Kb maximum-frame size), and the requested frame-delay bound is d ms. The maximum expected traffic of this video channel is thus $B_c = C/T \times 100 = 3$ Mbps. Suppose the network target token rotation time is set to a typical value $TTRT = 8$ ms.

Since d must be no smaller than $2 \times TTRT$, the above video channel cannot be established for $d < 16$ ms. For $16 \leq d \leq T + TTRT = 41$ ms, the minimum required h_s is calculated from Theorem 3.2 as

$$h_s = \begin{cases} 1/p & if\ q \geq 1/p \\ (1+q)/(1+p) & if\ q < 1/p \end{cases}$$

where $p = \lfloor d/8 - 1 \rfloor$ and $q = \lceil d/8 \rceil 8 - d$.

For $d \geq T + 2 \times TTRT = 49$ ms, $h_s = (TTRT/T)C = 0.24$ ms. For 41 ms $< d <$ 49 ms, we use the upper bound given in Theorem 3.2, $h_s = C/p_0 = 0.25$ ms.

Recall that the synchronous bandwidth assigned to a channel is $B_s = h_s/TTRT \times 100$ Mbps. The value of B_s as a function of d is plotted in Fig. 2 from which we have the following observations.

1. The smaller the requested delay bound d, the more synchronous bandwidth is required by the channel. For example, the video channel needs to reserve a 12.5 Mbps synchronous bandwidth, which is more than four times as much as the expected signal bandwidth over the channel, to guarantee that each video frame be delivered within a delay bound $d < 23$ ms. In general, a channel requires a synchronous bandwidth approximately $T/(\lfloor d/TTRT - 1 \rfloor TTRT)$ times as much as its expected maximum signal bandwidth to guarantee a delay bound d (from Theorem 3.2). This shows that the FDDI is not very efficient in supporting real-time communications with tight-delay requirements. The readers are referred to [14, 15] for a simple modification to the MAC protocol of the FDDI which can significantly improve FDDI's ability of supporting real-time traffic requiring small delay bounds.

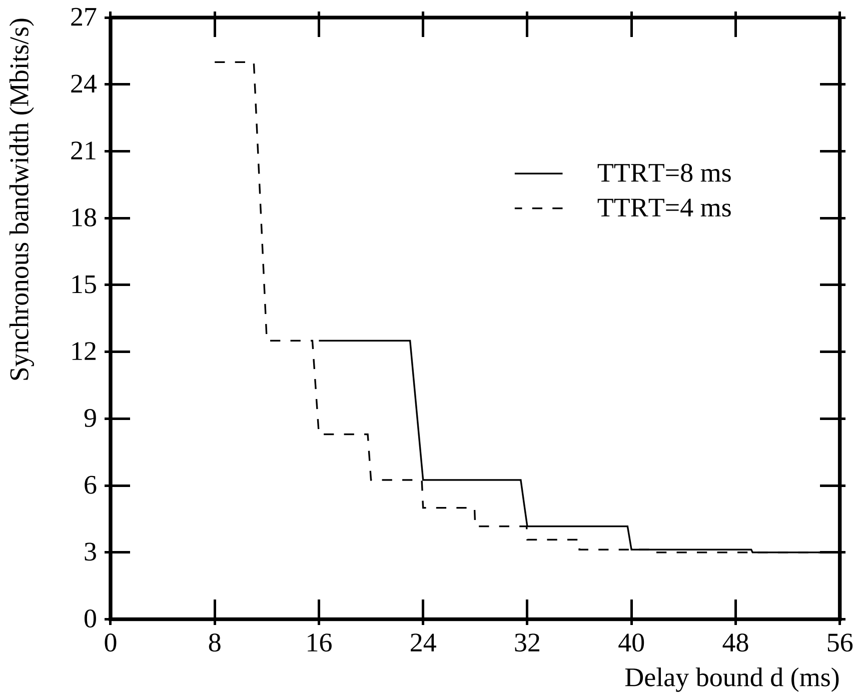

Figure 2: Synchronous bandwidth assigned to the video channel.

2. The required synchronous bandwidth reduces to the expected signal bandwidth for $d \geq T + 2 \times TTRT = 49$ ms. This fact has two implications.

   - If a channel is assigned a synchronous bandwidth equal to its expected signal bandwidth, it is guaranteed that each of its message will be transmitted with a delay no larger than $T + 2 \times TTRT$ (or $T + TTRT$ if T is a multiple of $TTRT$). This is in contrast to the common misunderstanding that the message delay bound equals the medium access delay bound $2 \times TTRT$.
   - One does not gain anything by allowing the message delay to be larger than $T + 2 \times TTRT$. In other words, a video channel which allows its frames to be delayed as large as 500 ms needs the same synchronous bandwidth as a channel requiring frame delays to be no more than 50 ms in the above example. This finding is very useful for designing distributed multimedia systems over FDDI networks.

3. The difference between the upper bound of h_s calculated from Theorem 3.2 and the actual minimum h_s is negligible if T is several times larger than $TTRT$. The difference increases

with the decrease of T. So if one has to set $T + TTRT < d < T + 2 \times TTRT$ and T is not a multiple of $TTRT$, $TTRT$ should be set as small as possible to avoid any over-reservation of synchronous bandwidth.

For the purpose of comparison, the synchronous bandwidth needed by the video channel with $TTRT = 4$ ms is also plotted as the dotted curve in Fig. 2. In general, a smaller $TTRT$ gives an FDDI network a better performance in supporting real-time communication (can provide smaller delay bounds and require less synchronous bandwidth) than a larger $TTRT$. But as discussed in [1], a small $TTRT$ reduces the overall network efficiency due to token passing overheads and ring latency. Thus, unless some applications require very tight delay bounds, a moderate $TTRT$ (around 8 ms) is appropriate.

Algorithm 3.1 can also be used for real-time channels with a common source node. Specifically, if two channels τ_1 and τ_2 have the same source node s, and τ_1 requires $h_s = t_1$ and τ_2 requires $h_s = t_2$. Then setting $h_s := t_1 + t_2$ will satisfy the requirements of both channels provided there is a mechanism at the source node to regulate the transmission times of the messages of τ_1 and τ_2 so as not to exceed t_1 and t_2 at each token's visit, respectively.

4 Conclusion

This paper has addressed the problem of allocating synchronous bandwidths in FDDI networks. We developed a general, optimal, and simple SBA scheme that can support a large variety of real-time applications, can fully utilize the network-transmission bandwidth, and is easy to implement. This paper also shows that the FDDI is capable of supporting real-time communications and is a good candidate for distributed multimedia applications.

References

[1] F. E. Ross, "An overview of FDDI: The fiber distributed data interface," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 7, pp. 1043 – 1051, September 1989.

[2] A. C. Weaver, "Local area networks and busses – an analysis," Technical report, Flight Data Systems, NASA–Johnson Space Center, 1986.

[3] *Fiber Distributed Data Interface (FDDI) – Token Ring Media Access Control (MAC).* American National Standard, ANSI X3.139, 1987.

[4] *FDDI Station Management (SMT) – draft proposed.* American National Standard, ANSI X3T9/92-067, June 25, 1992.

[5] G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing synchronous message deadlines with the timed token protocol," in *Proceedings of IEEE International Conference on Distributed Computing Systems*. IEEE, June 1992.

[6] B. Chen, G. Agrawal, and W. Zhao, "Optimal synchronous capacity allocation for hard real–time communications with the timed token protocol," in *Proc. Real-Time Systems Symposium*. IEEE, December 1992.

[7] K. C. Sevcik and M. J. Johnson, "Cycle time properties of the FDDI token ring protocol," *Technical Report CSRI-179, Computer Science Research Institute, University of Toronto*, 1986.

[8] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. SAC-8, no. 3, pp. 368–379, April 1990.

[9] D. D. Kandlur, K. G. Shin, and D. Ferrari, "Real-time communication in multi-hop networks," in *Proc. 11th Int. Conf. on Distributed Computer Systems*, pp. 300–307. IEEE, May 1991.

[10] Q. Zheng and K. G. Shin, "On the ability of establishing real–time channels in point–to–point packet–switched networks," *IEEE Transactions on Communication* (in press), 1993.

[11] Q. Zheng and K. G. Shin, "Real–time communication in local area ring networks," in *Conference on Local Computer Networks*, pp. 416–425, September 1992.

[12] Q. Zheng and K. G. Shin, "Fault–tolerant real–time communication in distributed computing systems," in *in* Proc. 22nd Annual International Symposium on Fault–tolerant Computing, pp. 86 – 93, 1992.

[13] Q. Zheng, K. G. Shin, and E. Abram-Profeta, "Transmission of compressed digital motion video over computer networks," in *Digest of COMPCON Spring'93*, pp. 37–46, February 1993.

[14] K. G. Shin and Q. Zheng, "FDDI-M: a scheme to double FDDI's ability of supporting synchronous traffic," submitted for publication, 1992.

[15] Q. Zheng, *Real-time Fault-tolerant Communication in Computer Networks*, PhD thesis, University of Michigan, 1993. PostScript version of the thesis is available via anonymous FTP from ftp.eecs.umich.edu in directory outgoing/zheng.