Synchronous Bandwidth Allocation in FDDI Networks

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

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 networks in supporting such real-time communication as digital video/audio transmissions, and distributed control/monitoring.

Index Terms—Local area networks, FDDI, synchronous bandwidth allocation, real-time communication, real-time digital video transmission.

I. 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], [5]. 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 some number, TTRT, and the synchronous bandwidth allocation of node i is set to some number, hi, (see Protocol 2.1 in Section II for the definition of TTRT and hi), then the time node i must wait for a chance to transmit its synchronous messages is bounded by 2 × TTRT, and on the average, the node is guaranteed to have a bandwidth of 100 × hi/TTRT 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 100 × h/2TTRT Mbps, it is not guaranteed that all of the messages will have a delay bound 2 × 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 msec.
- 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 100 × C/T Mbps. Thus, the throughput requirement would be satisfied if the FDDI is configured such that h/2TTRT = C/T. Also, the delay requirement would be satisfied if 2 × TTRT = T. Together we get h = 2C. From the Medium Access Control (MAC) protocol of the FDDI, hi is the maximum time node i is allowed to transmit synchronous messages once it acquires the token. Thus, hi = 2C 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 hi 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 to too small a value; 3) reducing TTRT also increases the synchronous bandwidth assigned to the station. This paper addresses the second approach, i.e., set hi appropriately. Specifically, we will develop a synchronous bandwidth allocation (SBA) scheme which, given a network target token rotation time TTRT and an application traffic specifica-
tion, 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 [5], suggesting how synchronous bandwidth may be allocated to a node, but it does not indicate how much of synchronous bandwidth needs to be allocated 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. [6] 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 intergeneration 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 the other nodes. A global SBA scheme complicates the implementation of synchronous bandwidth allocation.

To improve the scheme in [6], Chen et al. proposed an optimal SBA scheme [7]. 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. Some nonoptimal local SBA schemes were proposed in [8].

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, the scheme is local thus allocation/deallocation of a synchronous bandwidth to a node does not require to change the synchronous bandwidths previously assigned to the other nodes, thus making the synchronous bandwidth allocation easy to implement.

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

II. PRELIMINARIES

For convenience of discussion, we review the FDDI's MAC protocol and some of its properties in this section. The FDDI's MAC protocol [4] is summarized below.

**Protocol 2 (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 $THT := 0$.

**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, 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 $(i + 1) \mod N$.

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”:

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

The physical meaning of the above inequality is that the summation of the synchronous bandwidths assigned to the nodes in an FDDI ring should not exceed the effective ring bandwidth. Violation of this constraint would make the ring unstable and oscillate between “claiming” and “operational” [5]. Under this protocol constraint, there is a well-known fact about the FDDI: The worst-case token rotation time is bounded by $2 \times TTRT$, and the average token rotation time is bounded by $TTRT$ [9]. Recently, Agrawal et al. [6] obtained a more general result 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 tokens' 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 not larger than $x$, and $\lceil x \rceil$ be the smallest integer which is not smaller than $x$. We also define the following two functions, $\lfloor x \rfloor^+$ and $\lceil x \rceil^+$:

$$\lfloor x \rfloor^+ = \begin{cases} \lfloor x \rfloor & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

$$\lceil x \rceil^+ = \begin{cases} n & \text{if } n - 1 \leq x < n, n = 1, 2, \ldots, \\ 0 & \text{if } x < 0 \end{cases}$$

The following lemma [6] gives a lower bound of the time node $i$ is allowed to transmit its synchronous messages during a time period $t$.

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

III. 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 prespecified 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. An application may exceed its prespecified 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 application will not affect the guarantees to 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 [10] for real-time communication services. A real-time channel is described by a 4-tuple $\tau = (T, C, d, s)$ and 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 intergeneration time is $\geq T$ and the message transmission time is $\leq C$. For simplicity, in a shared-medium network like FDDI, a message is said to be 'delivered' when its source node completes the transmission of the message.

A real-time channel is a convenient way to achieve real-time communication. Users can set up real-time 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 to use. We will in this paper deal with the implementation of real-time channels on FDDI networks only. Readers are referred to [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] for more discussions on real-time channels and/or other real-time communication schemes 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 the TTRT and $h_i$. The TTRT is usually set at the network initialization time and does not change frequently. It determines the minimum message-delay bound, $d_{\text{min}} = 2 \times TTRT$, that the network can guarantee. Any channel request with a delay bound smaller than $d_{\text{min}}$ 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 $h_i$ 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 [6], [7].

Let $\Gamma(t)$ 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 $\Gamma(t)$ for node $i$. We improve Lemma 2.2 by calculating the exact value of $\Gamma(t)$ 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

$$\Gamma(t) = \lfloor t \div TTRT \rfloor^+ \cdot h_i + \delta(t)$$

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

$$\delta(t) = \begin{cases} 0 & \text{if } \lfloor t \div TTRT \rfloor^+ \cdot TTRT - t \geq h_i \\ 1 - \lfloor t \div TTRT \rfloor^+ \cdot TTRT - h_i & \text{otherwise}. \end{cases}$$

Proof. $\Gamma(t)$ is plotted in Fig. 1. Its correctness can be proved by observing that the bound of token rotation time stated in Lemma 2.1 is actually reached at node $i$ when 1) all usable ring bandwidth is assigned to nodes as synchronous band-
widths, 2) no messages are transmitted during the first token rotation, and 3) starting from node \(i\), all nodes use the maximum times allowed to transmit their synchronous and asynchronous messages during the subsequent token visits. Thus, if a synchronous message is generated at node \(i\) just after it starts transmitting asynchronous messages, node \(i\) would have to wait \(2 \times TTRT - h_i\) units of time for another token’s visit to transmit the synchronous message. Once it gets the token, node \(i\) has \(h_i\) units of time to transmit synchronous messages. This proves the correctness of \(\Gamma(t)\) for \(t \leq 2 \times TTRT\). The following worst-case token interarrival times at node \(i\) would be \(TTTRT\). This proves the correctness of \(\Gamma(t)\) for \(t > 2TTTRT\).

![Graph](image)

**Fig. 1. Worst-case synchronous transmission time \(\Gamma(t)\).**

It should be noted that the synchronous-transmission time in the above Lemma is calculated for the worst case where all usable ring bandwidth is assigned to nodes as the 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 the synchronous bandwidth. But using such a "better" calculation of synchronous bandwidth allocation would reduce the total amount of synchronous traffic that a network can support and hence it 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

\[
\forall t \geq 0, \left(\begin{array}{c}
\frac{(t - d)T}{T}
\end{array}\right) C \leq \Gamma(t),
\]

where \(\Gamma(t)\) is calculated from Lemma 3.1 with \(i = s\).

The proof of this theorem is almost the same as that of Theorem 1 of [12].

**PROOF OF THE NECESSARY CONDITION.** Suppose node \(s\) does not have any message of channel \(\tau\) 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 \(\tau\) with deadlines \(\leq t\) is not greater than \(\Gamma(t)\), 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 \(\tau\) is \(T\), there are at most \(\Gamma(t - dT)\) messages generated by channel \(\tau\) during \([0, t]\) with deadlines \(\leq t\), which take at most \(\Gamma(t - dT) C\) units of time to be transmitted. Thus, the maximum value of \(\tau(t)\) is \(\Gamma(t - dT) C\). 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 \([t', t_1]\), there is at least one message of channel \(\tau\) in node \(i\) and node \(i\) uses all of its allowed synchronous transmission time for channel \(\tau\)’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 \(\tau\) 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 \(\tau(t_1 - t_0) = \Gamma(t_1 - t_0 - d) C\). Since one message misses its deadline at \(t_1\), this \(\tau(t_1 - t_0)\) must be larger than \(\Gamma(t_1 - t_0)\), that is, \(\Gamma(t_1 - t_0 - d) C > \Gamma(t_1 - t_0)\). By letting \(t = t_1 - t_0\), the above inequality contradicts the condition that \(\forall t \geq 0, \left(\begin{array}{c}
\frac{(t - d)T}{T}
\end{array}\right) C \leq \Gamma(t)\).

It is difficult to calculate the minimum synchronous bandwidth allocation (i.e., \(h_i\)) needed for a real-time channel from Theorem 3.1 since inequality (2) must be checked in an interval of infinite 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 \(h_i\) for other cases.

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

1) For \(2 \times TTRT \leq d \leq T + TTRT\),

\[
h_i = \left\{ \begin{array}{ll}
C/p & \text{if } q \geq C/p \\
(C+q)/(1+p) & \text{if } q < C/p
\end{array} \right.
\]

where \(p = \sqrt{dTTTRT - 1}\) and \(q = \sqrt{dTTTRT} TTRT - d\).

2) For \(d \geq T + 2 \times TTRT\),

\[
h_i = (TTRT/T)C
\]

3) For \(T + TTRT < d < T + 2 \times TTRT\) and \(T \geq TTRT\),

\[
h_i \leq \left\{ \begin{array}{ll}
C/p_0 & \text{if } q_0 \geq C/p_0 \\
(C+q_0)/(1+p_0) & \text{if } q_0 < C/p_0
\end{array} \right.
\]

where \(p_0 = \sqrt{TTRT}\) and \(q_0 = \sqrt{TTRT} TTRT - T\).

4) For \(2 \times TTRT < d < T + 2 \times TTRT\) and \(T < TTRT\),

\[
h_i \leq \sqrt{TTRT} C
\]

**PROOF.** Notice that the minimum \(h_i\) required to set up a real-time channel is the minimum value of \(h_i\) that satisfies
inequality (2). We prove four parts of Theorem 3.2, one by one, using this fact.

1) Notice that the left-hand side of inequality (2) is a step function which changes values to \((k + 1)C\) at points \(t = d + kT, k = 0, 1, \ldots\) and the right-hand side 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, \(\Gamma(d) = ph + \delta(d)\) where

\[
\delta(d) = \begin{cases} 
0 & \text{if } q \geq h_r \\
-h_r + q & \text{if } q < h_r 
\end{cases}
\]

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

\[
\Gamma(d + kT) \geq \sum_{i=1}^{k} (\Gamma(d + iT) - \Gamma(d + iT - T) + \Gamma(d)) \\
\quad \geq \sum_{i=1}^{k} C + C \\
\quad = (k + 1)C.
\]

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

2) To provide any delay bound guarantee, the average synchronous bandwidth guaranteed to a node must be no smaller than the average signal bandwidth over a real-time channel, i.e., \(h_r/\text{TTRT}\) must be no smaller than \(\text{CIT}\). Thus \(h_r\) must be no smaller than \((\text{TTRT}/\text{T})\). To show \(h_r = (\text{TTRT}/\text{T})\) is large enough for this case, observe that the left-hand side of inequality (2) \(\leq (t - d)T/\text{T} + C\) and the right-hand side of inequality (2) \(\geq (h_r/\text{T})\) and \(h_r = (\text{TTRT}/\text{T})\), we get \((t - d)T/\text{T} + C \leq (h_r/\text{T})\) when \(T \geq \text{TTRT}, T + \text{TTRT} \geq 2 \times \text{TTRT}\). The first part of Theorem 3.2 can then be used to calculate \(h_r\) for \(d = T + \text{TTRT}\). Clearly, the \(h_r\) calculated this way is an upper bound for \(h_r\) when \(d > T + \text{TTRT}\).

3) During any \(\text{TTRT}\) units of time, the left-hand side of inequality (2) increases by at most \((\text{TTRT}/\text{T})\). Thus \(d \geq 2 \times \text{TTRT}\) and \(h_r = (\text{TTRT}/\text{T})\), inequality (2) is always satisfied.

Theorem 3.2 is a corrected and improved version of our previous results presented in [23], [24]. We have the following channel establishment algorithm.

**Algorithm 3.1.** (Channel Establishment over FDDI)

Suppose \(n - 1\) real-time channels \(x_i = (T_i, C_i, d_i, s_i), i = 1, \ldots, n - 1\) have already been established over an FDDI ring. Then a new channel \(x_n = (T_n, C_n, d_n, s_n)\) can be established with the following steps.

**Step 1.** Calculate \(h_r\) from Theorem 3.2.

**Step 2.** If the protocol constraint (1) is satisfied, set the synchronous-bandwidth allocation of \(s_n\) to \(h_r\) and establish channel \(x_n\). Otherwise, the channel establishment request is rejected.

Some discussions on the above algorithm are in order.

1) For \(T + \text{TTRT} < d < T + 2 \times \text{TTRT}\), Theorem 3.2 gives only an upper bound of the minimum \(h_r\). 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 (100 \(\times h\text{/TTRT Mbps}\)) must not be smaller than the expected signal bandwidth (100 \(\times \text{CIT Mbps}\)) over the channel. This means that \(h_r \geq (\text{TTRT}/\text{T})\) is a necessary condition and \((\text{TTRT}/\text{T})\) is a lower bound of the required \(h_r\). Thus the difference between the upper bounds given in Theorem 3.2 and the minimum \(h_r\) is bounded by

\[
\beta = \begin{cases} 
\left(1/\lfloor x \rfloor\right)^{-1} - \left(1/ x\right) & \text{if } x \geq 1 \\
\left(1/\lfloor x \rfloor\right)^{-1} - \left(1/ x\right) & \text{if } x < 1 
\end{cases}
\]

where \(x = \text{TTRT}\).

From this we see that the upper bound obtained in Theorem 3.2 will never exceed twice the minimum \(h_r\). Another result is that the upper bound given in Theorem 3.2 is actually the minimum \(h_r\) when \(T\) is a multiple of \(\text{TTRT}\) or \(\text{TTRT}\) is a multiple of \(T\).

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

- \(2 \times \text{TTRT} \leq d \leq T + \text{TTRT}\),
- \(d \geq T + 2 \times \text{TTRT}\),
- \(T\) is a multiple of \(\text{TTRT}\),
- \(\text{TTRT}\) 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 \((d \leq T + \text{TTRT})\), and video/audio communication can often tolerate larger delays \((d \geq T + 2 \times \text{TTRT})\). Thus for most applications, the synchronous-bandwidth allocation resulting from Algorithm 3.1 is optimal.

3) Comparing with previous results, the SBA scheme of [6] 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 [7] 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 [6], [7]). Also, the SBA schemes of [6], [7] 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 [6], [7]. Some nonoptimal local SBA schemes were also proposed in [8].

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 \( T = 33 \) msec (30 frames/sec), the transmission time of a maximum frame is 1 msec (100 Kb maximum-frame size), and the requested frame-delay bound is \( d \) msec. The maximum expected traffic of this video channel is thus \( B_c = CIT \times 100 = 3 \) Mbps. Suppose the network target token rotation time is set to a typical value \( TTRT = 8 \) msec.

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

\[
h_i = \begin{cases} \frac{1}{p} & \text{if } q \geq 1/p \\ \frac{(1+q)/(1+p)}{p} & \text{if } q < 1/p \end{cases}
\]

where \( p = \lfloor d/8 \rfloor \) and \( q = \lceil d/8 \rceil - 8 - d \).

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

Recall that the synchronous bandwidth assigned to a channel is \( B_v = h_i TTRT \times 100 \) Mbps. The value of \( B_v \) 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 \( d < 23 \) msec. In general, a channel requires a synchronous bandwidth approximately \( T(\lfloor d/TTRT \rfloor - 1) TTRT \) 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 communication with tight-delay requirements. The readers are referred to [24], [25] 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.

2) The required synchronous bandwidth reduces to the expected signal bandwidth for \( d \geq T + 2 \times TTRT = 49 \) msec. This fact has two important 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 msec needs the same synchronous bandwidth as a channel requiring frame delays to be no more than 50 msec 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 \( h_i \) calculated from Theorem 3.2 and the actual minimum \( h_i \) 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 \( TTRT = 4 \) msec 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 [2], 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 msec) is more appropriate.

![Fig. 2. Synchronous bandwidth assigned to the video channel.](image-url)

Algorithm 3.1 can also be used for real-time channels with a common source node. Specifically, if two channels \( r_1 \) and \( r_2 \) have the same source node \( s \), and \( r_1 \) requires \( h_i = t_1 \) and \( r_2 \) requires \( h_i = t_2 \). Then setting \( h_i := t_1 + t_2 \) 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 \( r_1 \) and \( r_2 \) so as not to exceed \( t_1 \) and \( t_2 \) at each token's visit, respectively.

### IV. CONCLUSION

In this paper we addressed the problem of allocating synchronous bandwidths in FDDI networks. Specifically, 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. We also showed that the FDDI is capable of supporting real-time communication and is a good candidate for distributed multimedia applications.
ACKNOWLEDGMENT

A portion of this paper was presented at the First ACM International Conference on Multimedia, Anaheim, California, August 1–6, 1993. The work reported in this paper was supported in part by the Office of Naval Research under Grant Nos. N00014–92–J–1080 and N00014–94–10229, and by the National Science Foundation under Grant No. MIP–9203895. Any opinions, findings, and recommendations expressed in this publication are those of the authors; they do not necessarily reflect the views of the funding agencies.

REFERENCES


Qiu Zheng (S’89–M’93) received the BS and MS degrees in electrical engineering from the University of Science and Technology of China in 1982 and 1985, respectively, and a PhD degree in electrical engineering from the University of Michigan at Ann Arbor in 1993. He joined Mitsubishi Electric Research Laboratories, Inc., Cambridge Research Center in 1993 and is currently working on high-speed computer networks and distributed real-time industrial systems.

Kang G. Shin (SM’83–F’92) received the BS degree in electronics engineering from Seoul National University, Seoul, Korea in 1970, and both the MS and PhD degrees in electrical engineering from Cornell University, Ithaca, New York in 1976 and 1978, respectively. 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. Air Force Flight Dynamics Laboratory, A&T Bell Laboratories, Computer Science Division within the Department of Electrical Engineering and Computer Science at the University of California at Berkeley, and the International Computer Science Institute, Berkeley, California. He also chaired the Computer Science and Engineering Division, EECS Department, the University of Michigan for three years beginning in January 1991.

Currently Dr. Shin is professor and director of the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, and the University of Michigan at Ann Arbor. He has authored/coauthored over 300 technical papers (more than 130 of these in archival journals) and numerous book chapters in the areas of distributed real-time computing and control, fault-tolerant computing, computer architecture, robotics and automation, and intelligent manufacturing. He is currently writing (jointly with C.M. Krishna) a textbook, Real-Time Systems, which is scheduled to be published by McGraw-Hill in the spring of 1996. In 1987, he received the Outstanding IEEE Transactions on Automatic Control Paper Award for a paper on robotic trajectory planning. In 1989, he 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.

Dr. Shin has also been applying the basic research results of real-time computing to intelligent transportation systems and manufacturing applications ranging from the control of robots and machine tools to the development of open architectures for manufacturing equipment and processes. Recently, he initiated research on the open-architecture information base for machine tool controllers.

He is an IEEE fellow; was the program chair of the 1986 IEEE Real-Time Systems Symposium (RTSS); the general chair of the 1987 RTSS; the guest editor of the August 1987 special issue of IEEE Transactions on Computers on Real-Time Systems; a program cochair for the 1992 International Conference on Parallel Processing; and has served on numerous technical program committees. He also chaired the IEEE Technical Committee on Real-Time Systems during 1991–1992; is a distinguished visitor of the IEEE Computer Society; an editor of IEEE Transactions on Parallel and Distributed Computing; and an area editor of the International Journal of Time-Critical Computing Systems.