Guaranteeing Synchronous Messages with Arbitrary Deadline Constraints in an FDDI Network*

Nicholas Malcolm and Wei Zhao
Department of Computer Science
Texas A&M University
College Station, Texas 77843-3112

Abstract

This paper addresses issues related to guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network. We show that several network parameters must be set carefully if message deadlines are to be satisfied. First, message deadlines can only be met if sufficient synchronous bandwidth is allocated to each node. Thus, proper synchronous bandwidth allocation is essential if deadlines are to be guaranteed. Second, the target token rotation time (TTRT) determines both the speed of token circulation and the network utilization available to user applications. TTRT should also be chosen carefully to ensure that the token circulates fast enough while maintaining a high available utilization. Finally, sufficient buffer space must be provided for outgoing messages, otherwise messages could be lost due to buffer overflow.

In this paper, we propose and analyze an integrated method for allocating the synchronous bandwidth and selecting TTRT so that the time constraints of synchronous messages with arbitrary deadlines are guaranteed to be met. Our method differs from previous work by taking into account both message periods and message deadlines. As a result, the network's capability for guaranteeing real-time messages is increased dramatically. Furthermore, we derive an upper bound for the maximum message waiting time. This gives an upper bound for the required buffer size. We show that this bound is independent of message deadlines — giving considerable flexibility to applications in choosing message deadlines.

1 Introduction

There is increasing use of distributed computer systems to support real-time applications. The key to successfully developing such systems is to have a communication network that supports the timely delivery of inter-task messages. The main focus of this study is thus to address issues related to guarantees of message deadlines in a communication network.

Our effort has concentrated on the Fiber Distributed Data Interface (FDDI) network. FDDI is an ANSI and ISO standard for fiber optic networks [3, 6]. The suitability of FDDI for embedded distributed real-time applications derives not only from its high bandwidth, but also from its property of a bounded access time and its dual ring architecture. The bounded access time provides a necessary condition to guarantee real-time deadlines. The dual ring architecture allows continuous real-time service even under some failure conditions. Many embedded real-time applications use FDDI as a backbone network. For example, FDDI has been selected as the backbone network for NASA's Space Station Freedom. FDDI has also been adopted by the U.S. Navy Next Generation Computer Resources Program as part of its Survivable Adaptable Fiber Optic Embedded Network (SAFENET) [19].

FDDI uses the timed token protocol at its media access control (MAC) layer. With this protocol, messages are divided into two separate classes: the *synchronous* class and the *asynchronous* class. Synchronous messages arrive in the system at regular intervals and may be associated with deadline constraints. The idea behind the timed token protocol is to control the token rotation time. At network initialization time, a protocol parameter called the *Target Token Rotation Time* (*TTRT*) is determined which indicates the expected token rotation time. Each node is assigned a fraction of the *TTRT*, known as its *synchronous bandwidth.*[1] The synchronous bandwidth of a node is the maximum time that the node is permitted to transmit synchronous messages every time it receives the token. After sending synchronous messages, a node can transmit its pending asynchronous messages. However, a node can send asynchronous messages only if the time elapsed since the previous token departure from the node is less than the value of *TTRT*, i.e., only if the token arrives earlier than expected.

In order to guarantee that the deadlines of synchronous messages are met, network parameters such

*This work was funded in part by the Australian Research Council and by the Engineering Excellence Fund of Texas A&M University.

[1] Some other synonymous terms that researchers use are *synchronous capacity* [1, 2, 4], *bandwidth allocation* [18], *synchronous allocation* [7], *synchronous bandwidth assignments* [10], and *high priority token holding time* [13]. This paper uses the term synchronous bandwidth, in accordance with the most recent version of the FDDI standard.

as the synchronous bandwidth, the target token rotation time, and the buffer size must be chosen carefully.

- The synchronous bandwidth is the most critical parameter in determining whether message deadlines will be met. If the synchronous bandwidth allocated to a node is too small, then the node may not have enough network access time to transmit messages before their deadlines. Conversely, large synchronous bandwidths can result in a long token rotation time, which can also cause message deadlines to be missed.
- Proper selection of $TTRT$ is also important. Let τ be the token walk time around the network. The proportion of time taken due to token walking is given by $\tau/TTRT$. The maximum network utilization available to user applications is then $1-\tau/TTRT$ [18]. A smaller $TTRT$ results in less available utilization and limits the network capacity. On the other hand, if $TTRT$ is too large, the token may not arrive often enough at a node in order to meet message deadlines.
- Each node has a buffer for outgoing synchronous messages. The size of this buffer also affects the real-time performance of the network.[2] A buffer that is too small can result in messages being lost due to buffer overflow. A buffer that is too large is wasteful of memory.

There is much literature addressing parameter selection for an FDDI network. However, most of this work on parameter selection is for non real-time systems. The objective in much of the previous work was to maximize the throughout and/or to minimize the delay, rather than to guarantee individual message deadlines [5, 9, 13, 14, 20]. There has recently been much progress in synchronous bandwidth allocation for real-time applications. In [1, 2, 4], a real-time system is considered where synchronous messages on different nodes can have different periods. However, the deadline of a message is assumed to be equal to its period. For this case it was shown that the worst case achievable utilization of the network can be 33%; if a message set has a utilization less than 33% of the available network utilization, then the message deadlines can always be guaranteed by a proper allocation of the synchronous bandwidth. Independently, in [17], an experiment using FDDI for transmitting multi-media data was described. In a multi-media system, message deadlines are much larger than message periods. Consequently, the synchronous bandwidth was allocated based on this assumption. Studies in [8, 16] addressed the issue of meeting synchronous message deadlines while maximizing the average asynchronous throughput. Recently, in [11], the real-time performance of FDDI and IEEE 802.5 has been compared.

Our study differs significantly from the previous work in that we seek a comprehensive solution for setting the network parameters for a general message system. In a general message system different message streams can have different periods and deadlines can be less than, equal to, or larger than periods. With our method, the network parameters — the synchronous bandwidths, $TTRT$, and the buffer size — are selected in an integrated fashion in order to guarantee the time constraints of messages. In particular, this study differs from the previous work in the following ways:

- A synchronous bandwidth allocation scheme is proposed and analyzed for a general message system. This is the first study presenting a synchronous bandwidth allocation scheme for a general message system. Our allocation scheme explicitly accounts for the periods and deadlines of individual messages — resulting in a greater capability to meet message deadlines.
- A method is developed for selecting $TTRT$ so that the worst case achievable utilization of the network is maximized. This enhances the likelihood that all of the deadlines of an arbitrary message set can be satisfied. No previous work has been reported on selection of $TTRT$ to maximize the worst case achievable utilization.
- An upper bound is derived for the queue length of synchronous messages. Consequently, the buffer size required for outgoing messages can be safely bounded. Once again, this is the first study on buffer size in an FDDI network when used for real-time applications.

2 Framework

In this section, we describe the framework of our study. We will first present the message and network models. We then discuss FDDI protocol properties that will be used in analyzing the parameter selection methods proposed in subsequent sections.

2.1 The Network and the Message Models

The network contains n nodes arranged in a ring. Message transmission is controlled by the timed token protocol, and the network is assumed to operate without any faults. To be consistent with current practice in FDDI networks, outgoing messages at a node are assumed to be queued in FIFO order.

The token walk time is denoted by τ. τ includes the node-to-node delay and the token transmission time. τ is the portion of $TTRT$ that is not available for message transmission. Let α be the ratio of τ to $TTRT$, i.e., $\alpha = \tau/TTRT$. α represents the proportion of time that is not available for transmitting messages.

There are n streams of synchronous messages, $S_1, \ldots, S_n$, with stream S_i incident on node i.[3] Each synchronous message stream S_i may be characterized as $S_i = (C_i, P_i, D_i)$:

- C_i is the maximum amount of time required to transmit a message in the stream.

[2] The size of the buffer for incoming messages also affects the real-time performance of the network. A message can be lost if the buffer for incoming messages overflows. The receiving node should be able to keep pace with incoming messages. This depends on the processor and memory speeds at the receiving node and is beyond the scope of this paper.

[3] In [2] it is noted that an arbitrary token ring network in which each node may have any number of incident synchronous message streams can be transformed into a logically equivalent network in which there is exactly one stream incident on each node.

- P_i is the interarrival period between messages in the synchronous message stream. Let the first message in stream S_i arrive at node i at time $t_{i,1}$. Then, the j-th message in stream S_i arrives at node i at time $t_{i,j} = t_{i,1} + (j-1)P_i$, where $j \geq 1$.

- D_i is the *relative deadline* of messages in the stream. The relative deadline is the maximum amount of time that may elapse between a message arrival and completion of its transmission. Thus, the transmission of the j-th message in stream S_i, which arrives at $t_{i,j}$, must be completed by $t_{i,j}+D_i$. $t_{i,j}+D_i$ is the *absolute deadline* of the message. To simplify the discussion, the terms "relative" and "absolute" will be omitted in the remainder of the paper when the meaning is clear from the context.

Before proceeding, we introduce some useful notations. For $i = 1, \ldots, n$, let

$$D_i = q_i\, TTRT + r_i \tag{1}$$

where $q_i = \lfloor \frac{D_i}{TTRT} \rfloor$ and $0 \leq r_i = D_i - q_i\, TTRT < TTRT$. Let D_{min} be a lower bound on message deadlines. Hence, for $i = 1, \ldots, n$,

$$D_{min} \leq D_i. \tag{2}$$

As with deadline D_i, D_{min} can be written in terms of its quotient and remainder when divided by $TTRT$:

$$D_{min} = q_{min}\, TTRT + r_{min} \tag{3}$$

where $q_{min} = \lfloor \frac{D_{min}}{TTRT} \rfloor$ and $0 \leq r_{min} = D_{min} - q_{min}\, TTRT < TTRT$. Finally, let P_{min} be a lower bound on message periods, i.e., for $i = 1, \ldots, n$,

$$P_{min} \leq P_i. \tag{4}$$

The *utilization*, U, of a synchronous message set is defined by

$$U = \sum_{i=1}^{n} \frac{C_i}{P_i}. \tag{5}$$

U can be regarded as the proportion of time required for synchronous traffic in the network.

2.2 Constraints

To guarantee the deadlines of synchronous messages the following constraints must be satisfied for any choice of synchronous bandwidths (H_is), the target token rotation time *(TTRT)*, and the buffer size at each node.

The Protocol Constraint. This constraint states that the synchronous bandwidths on all the nodes must sum to less than the available network bandwidth, i.e.,

$$\sum_{i=1}^{n} H_i \leq TTRT - \tau. \tag{6}$$

The Deadline Constraint. This constraint simply states that every synchronous message must be transmitted before its (absolute) deadline. Formally, let $s_{i,j}$ be the time that the transmission of the j-th message in stream S_i is completed. The deadline constraint implies that for $i = 1, \ldots, n$ and $j = 1, 2, \ldots$,

$$s_{i,j} \leq t_{i,j} + D_i \tag{7}$$

where $t_{i,j}$ is the arrival time and D_i is the (relative) deadline. Note that in the above inequality, $t_{i,j}$ and D_i are given by the application, but $s_{i,j}$ depends on the synchronous bandwidth allocation and the choice of $TTRT$.

Johnson and Sevcik [10, 15] showed that for the timed token protocol, the maximum amount of time that may pass between two consecutive token arrivals at a node can approach $2TTRT$. To satisfy the deadline constraint, it is necessary for a node to have at least one opportunity to send each synchronous message before the message deadline expires. Therefore, in order for the deadline constraint to be satisfied, it is necessary that for $i = 1, \ldots, n$,

$$D_i \geq 2TTRT. \tag{8}$$

It is important to note that (8) is only a necessary but not a sufficient condition for the deadline constraint to be satisfied. For all message deadlines to be met it is also crucial to choose the synchronous bandwidths H_i appropriately.

The Buffer Constraint. This constraint states that the size of the buffer for outgoing synchronous messages at node i must be sufficient to hold the maximum number of synchronous messages that may be awaiting transmission at node i at any time. This constraint is necessary to ensure that messages are not lost due to buffer overflow.

2.3 Timing Properties of the Timed Token Protocol

In order to guarantee the synchronous messages at a node, it is necessary to have some information regarding the frequency of token visits to that node. Fortunately, extensive studies have already been carried out on the timing properties of the timed token protocol. Some of these properties are given below. These properties will be useful later in this paper. Let $T_{i,j}$ be the time of the j-th token arrival at node i.

Theorem 1 (Generalized Johnson and Sevcik's Theorem [2]) *For $1 \leq i \leq n$ and $j \geq 1$, the maximum amount of time that may pass between the j-th token arrival at node i and the $(j+k)$-th token arrival at node i satisfies*

$$T_{i,(j+k)} - T_{i,j} \leq (k+1)\,TTRT - H_i. \tag{9}$$

In particular, the time between the j-th arrival and the $(j+1)$-th arrival satisfies

$$T_{i,(j+1)} - T_{i,j} \leq 2TTRT - H_i \leq 2TTRT. \tag{10}$$

This is the relationship between two consecutive token arrivals at a node that was derived by Johnson and Sevcik [10, 15].

The generalized Johnson and Sevcik's theorem can be used to derive the following result.

Corollary 1 *In any interval of time D, the token will visit node i at least V_i times where*

$$V_i = \lfloor \frac{D}{TTRT} - 1 \rfloor. \tag{11}$$

In each of these visits, node i can use its full synchronous bandwidth H_i to transmit its synchronous messages (if any).

This corollary will be used extensively in the following sections in order to show that the proposed synchronous bandwidth allocation scheme satisfies the deadline constraint.

2.4 Performance Metric

To gauge the performance of a scheme for choosing the synchronous bandwidths H_i and the target token rotation time $TTRT$, it is necessary to have an appropriate performance metric. A metric that has commonly been used for real-time processing and for real-time communication is the worst case achievable utilization.

A network protocol (with a given setting of its parameters) has an *achievable utilization* U' if it can meet the deadlines of *any* synchronous message set with a utilization no more than U'. For example, if a network has an achievable utilization $U' = 0.5$, then all synchronous message sets with utilization $U \leq 0.5$ will have their message deadlines satisfied. The *worst case achievable utilization* U^* of a network is the least upper bound of its achievable utilizations. Hence, the network can meet the deadlines of all synchronous message sets with utilization no more than U^*.

The importance of the worst case achievable utilization U^* of a network is that it relates to the fundamental requirements of *predictability* and *stability* in hard real-time environments. If the utilization of a message set is no more than U^*, it can be predicted that all of the messages will meet their deadlines. This is because the deadlines of all message sets with utilization no more than U^* are guaranteed to be met. U^* also provides a measure of the stability of a system. The parameters of a synchronous message set can be freely modified while still maintaining schedulability, provided that the utilization remains less than U^*. This gives a certain amount of system stability in the face of changes to the parameters of a synchronous message set.

Starting with the next section, we study guarantees of synchronous message deadlines in an FDDI network. In Section 3, we propose a synchronous bandwidth allocation scheme. The worst case achievable utilization U^* of an FDDI network with the proposed synchronous bandwidth allocation scheme is then derived. In Section 4, it is then shown how to choose $TTRT$ so that the worst case achievable utilization is maximized. For both Sections 3 and 4, it is assumed that sufficient buffer space is provided so that messages will never be lost due to buffer overflow. In Section 5, we will show that the buffer size is bounded if our methods of synchronous bandwidth allocation and $TTRT$ selection are used.

3 Synchronous Bandwidth Allocation

The selection of appropriate values for the synchronous bandwidths H_i is a crucial step in meeting message deadlines. This section proposes a new synchronous bandwidth allocation scheme for guaranteeing the time constraints of synchronous messages with arbitrary deadlines.

3.1 A Classification of Synchronous Bandwidth Allocation Schemes

Synchronous bandwidth allocation schemes may be divided into two classes: local allocation schemes and global allocation schemes. These schemes differ in the type of information they may use. A *local* synchronous bandwidth allocation scheme uses only information available locally to node i in allocating H_i. Locally available information at node i includes the parameters of stream S_i (i.e., C_i, P_i, and D_i). τ and $TTRT$ are also locally available at node i, because these values are known to all nodes. On the other hand, a *global* synchronous bandwidth allocation scheme can use global information in its allocation of synchronous bandwidth to a node. Global information includes both information locally available to nodes and information regarding the parameters of synchronous message streams incident on other nodes.

A local scheme is preferable from a network management perspective. If the parameters of stream S_i on node i change, then only the synchronous bandwidth H_i of node i need be recalculated. The synchronous bandwidths at other nodes need not change because they were calculated independently of S_i. This makes a local scheme flexible, and suitable for use in dynamic environments.

In a global scheme, if the parameters of S_i change, then it may be necessary to recompute the synchronous bandwidths at all nodes. Thus a global scheme might not be well suited to a dynamic environment. On the other hand, a global scheme may perform better than a local one because a global scheme uses extra information. However, it is known that local schemes can perform *very close* to the optimal synchronous bandwidth allocation scheme when message deadlines are equal to message periods [1, 4]. Therefore, given the previously demonstrated good performance of local schemes and their desirable network management properties, this paper concentrates on local synchronous bandwidth allocation schemes.

3.2 A New Local Synchronous Bandwidth Allocation Scheme

We begin with an informal discussion motivating our selection of a synchronous bandwidth allocation scheme. A formal proof of its correctness then follows. We will first consider the case that $D_i \geq P_i$, i.e., that deadlines are greater than periods. This assumption will be relaxed in Section 6.

How should the synchronous bandwidths H_i be allocated? A message must be sent within D_i time units of arrival if it is to meet its deadline. By Corollary 1, the number of token visits to node i in D_i time units is at least $\lfloor \frac{D_i}{TTRT} - 1 \rfloor$. This suggests that for a message at node i to meet its deadline, the synchronous bandwidth H_i must be sufficient to send the message in $\lfloor \frac{D_i}{TTRT} - 1 \rfloor$ token visits. Therefore we require

$$H_i \geq \frac{C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \quad (12)$$

On the other hand, to meet the deadline requirement, the traffic flow at a node must be statistically balanced. On average, the number of messages arriving at a node in a given time interval must be equal to the number of messages that the node can transmit in the same interval. Consider a time interval of length D_i. $\frac{D_i}{P_i}C_i$ can be loosely regarded as the average traffic demand on node i during this interval. For example, if $D_i = 2.4P_i$, then in a time interval of length D_i, 2.4 messages "arrive" at node i for transmission. For the flow to be balanced, $\frac{D_i}{P_i}C_i$ messages must also be transmitted in every interval of length D_i. Corollary 1 tells us that the minimum number of token visits to node i in D_i time units is $\lfloor \frac{D_i}{TTRT} - 1 \rfloor$, and that the full synchronous bandwidth H_i is available in each of these visits. Therefore, for the flow to be balanced, we require $\lfloor \frac{D_i}{TTRT} - 1 \rfloor H_i \geq \frac{D_i}{P_i}C_i$. That is, H_i should be allocated as

$$H_i = \frac{\frac{D_i}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \quad (13)$$

In allocating the synchronous bandwidth, however, it is preferable to make H_i as small as possible while still satisfying the deadline constraint. A smaller value of H_i is preferable for two reasons. First, it improves the response time for asynchronous messages. Second, there will be a better chance of satisfying the protocol constraint.

Upon closer inspection, H_i can be made smaller than suggested in (13) while still satisfying the deadline constraint whenever (13) can do so. Recall that $D_i = q_i TTRT + r_i$ where $0 \leq r_i < TTRT$. Consider the following two cases.

Case 1: $r_i > 0$. In this case, (13) yields

$$H_i' = \frac{\frac{D_i}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor} = \frac{\frac{q_i TTRT}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor} + \frac{\frac{r_i}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \quad (14)$$

Case 2: $r_i = 0$. In this case, (13) yields

$$H_i'' = \frac{\frac{D_i}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor} = \frac{\frac{q_i TTRT}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \quad (15)$$

Note that $H_i'' < H_i'$. However, if H_i'' is sufficient to satisfy the deadline constraint when $r_i = 0$, then it should also be sufficient when the deadline is increased and $r_i > 0$; increasing the deadline should not cause a schedulable message set to become unschedulable. This means that in Case 1, the term $(\frac{r_i}{P_i}C_i)/\lfloor \frac{D_i}{TTRT} - 1 \rfloor$ is redundant and can be ignored, resulting in a smaller value for the synchronous bandwidth. In both cases, the synchronous bandwidth can be given as

$$H_i = \frac{\frac{q_i TTRT}{P_i}C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \quad (16)$$

Combining this observation with the result in (12), we propose the following local synchronous bandwidth allocation scheme:

$$H_i = \frac{\max(\frac{q_i TTRT}{P_i}, 1)C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \quad (17)$$

In the remainder of this section, the conditions under which the protocol constraint is satisfied will be derived for the proposed local synchronous bandwidth allocation scheme. The deadline constraint will then be shown to be satisfied whenever the protocol constraint is satisfied. Following this, the worst case achievable utilization of a network with this bandwidth allocation scheme is derived. Due to space limitations, proofs of some lemmas will not be presented in this paper. The interested reader is referred to [12] for more details.

3.3 Satisfying the Protocol Constraint

The following theorem indicates that the protocol constraint will be satisfied provided that the utilization of the given set of synchronous messages falls within a certain bound.

Theorem 2 *For any synchronous message set M with utilization $U \leq \frac{q_{min}-1}{q_{min}+1}(1-\alpha)$, if the synchronous bandwidths are allocated using the scheme in (17), then the protocol constraint will be satisfied.*

3.4 Satisfying the Deadline Constraint

We now prove that the deadline constraint is always satisfied when using the local synchronous bandwidth allocation scheme defined in (17). First we introduce the notion of a busy interval.

Definition 1 *A busy interval at node i is a maximal interval of time $[t_0, t_1]$ such that at all points of time between t_0 and t_1 inclusive, the buffer for outgoing synchronous messages at node i is nonempty.*

Obviously, when we consider network behavior regarding real-time messages, we only need to consider the behavior of the network during a busy interval.

The beginning of a busy interval will always coincide with a message arrival, though every message arrival will not necessarily begin a new busy interval. Let M be the k-th message that arrives during a busy interval $[t_0, t_1]$ at node i. M arrives at time $t_0 + (k-1)P_i$ and its absolute deadline is $t_0 + (k-1)P_i + D_i$. By Corollary 1, the number of token visits between t_0 and $t_0 + (k-1)P_i + D_i$ is at least $\lfloor \frac{(k-1)P_i+D_i}{TTRT} - 1 \rfloor$. H_i units of synchronous bandwidth is available on each of these visits. This means that the amount of synchronous bandwidth available for node i between t_0 and $t_0 + (k-1)P_i + D_i$ is at least $\lfloor \frac{(k-1)P_i+D_i}{TTRT} - 1 \rfloor H_i$. The following lemma bounds how many messages can be sent with this much synchronous bandwidth.

Lemma 1 *If the synchronous bandwidth H_i at node i is allocated according to (17) and the protocol constraint is satisfied, then*

$$\lfloor \frac{(k-1)P_i + D_i}{TTRT} - 1 \rfloor H_i \geq kC_i \tag{18}$$

for all integers $k > 0$.

Intuitively, this lemma says that the amount of synchronous bandwidth available to node i before the deadline of the k-th message in a busy interval will be sufficient for the k-th message, and all those messages which preceded the k-th message in the busy interval (a total of $(k-1)+1$ messages), to be sent before their deadlines. This notion is formalized in the following theorem.

Theorem 3 *If the synchronous bandwidths are allocated using the scheme in (17) and the protocol constraint is satisfied, then the deadline constraint will be satisfied.*

Proof: First, observe that every synchronous message must arrive during a busy interval. Either a message will arrive within a busy interval, or the message arrival will coincide with the beginning of a new busy interval. Also, recall that outgoing synchronous messages are assumed to be queued in FIFO order.

Suppose, for the purpose of contradiction, that the conditions stated in this theorem are met, but that synchronous message M arriving at node i misses its deadline. Let $[t_0, t_1]$ denote the busy interval during which M arrives. Let M be the k-th message to arrive during busy interval $[t_0, t_1]$, where $k > 0$. The absolute deadline of message M is $t_0 + (k-1)P_i + D_i$. From Corollary 1, during time interval $[t_0, t_0 + (k-1)P_i + D_i]$ the token will visit node i at least $\lfloor \frac{(k-1)P_i+D_i}{TTRT} - 1 \rfloor$ times. During these visits, the total amount of synchronous messages node i can transmit is $\lfloor \frac{(k-1)P_i+D_i}{TTRT} - 1 \rfloor H_i$. Lemma 1 indicates that this should be enough to send k messages. If M misses its deadline, then the synchronous messages must be transmitted out of arrival order because the outgoing synchronous message queue is empty immediately prior to t_0. This is a contradiction because messages are queued in FIFO order. Therefore, M must meet its deadline.

3.5 Worst Case Achievable Utilization

The proofs that both the protocol and the deadline constraints can be satisfied when using the local synchronous bandwidth allocation scheme lead to the following theorem, giving the worst case achievable utilization.

Theorem 4 *If the synchronous bandwidths are allocated using the scheme in (17), then the worst case achievable utilization, U^*, of the network is given by*

$$U^* = \frac{\lfloor \frac{D_{min}}{TTRT} \rfloor - 1}{\lfloor \frac{D_{min}}{TTRT} \rfloor + 1}(1-\alpha). \tag{19}$$

Proof: From the definition of worst case achievable utilization, the theorem will be established if we prove the following statements:

1. For any synchronous message set with utilization no more than U^* as defined in (19), the protocol constraint will be always satisfied.

2. For any synchronous message set with utilization no more than U^* as defined in (19), the deadline constraint will be always satisfied.

3. For any given $\epsilon > 0$, there exists a synchronous message set with utilization U such that $U^* < U \leq U^* + \epsilon$ and the protocol constraint cannot be satisfied for this set of messages.

Statements 1 and 2 follow directly from Theorems 2 and 3. The proof of Statement 3 is left as an exercise for the reader; the solution is found in [12].

Figure 1 shows the worst case achievable utilization for several different values of $TTRT$. τ is taken to be 0.05, and all time units are normalized in terms of P_{min}. Several observations can be made from Figure 1 and formula (19).

1. For a fixed value of $TTRT$, the worst case achievable utilization increases when the D_{min} increases. In fact, from (19) it can be shown that when D_{min} approaches infinity, U^* approaches $(1-\alpha) = (1 - \tau/TTRT)$. That is, as deadlines become infinitely large, the worst case achievable utilization is the same as the available utilization of the network.

2. In [1], it was shown that for a system in which all relative deadlines are equal to message periods ($D_i = P_i$), a worst case achievable utilization of $\frac{1}{3}(1-\alpha)$ can be achieved. That result can be seen as a special case of (19): when $D_{min} = 2TTRT$, we have $q_{min} = \lfloor \frac{D_{min}}{TTRT} \rfloor = 2$ and $U^* = \frac{1}{3}(1-\alpha)$.

3. $TTRT$ clearly has an impact on the worst case achievable utilization. From Figure 1, it can be seen that when $D_{min} = 1$, $TTRT = 0.2$ gives a higher worst case achievable utilization than the other plotted values of $TTRT$. When $D_{min} = 10$, $TTRT = 0.5$ gives a higher U^* than the other plotted values of $TTRT$. This observation provides motivation for maximizing U^* by properly selecting $TTRT$ once D_{min} is given.

4 Selection of TTRT

This section considers how to choose $TTRT$ such that the worst case achievable utilization U^* is maximized. Several lemmas are necessary for deriving the main result.

Lemma 2 narrows down the range of $TTRT$ values that need to be considered.

Lemma 2 *The maximum value of the worst case achievable utilization* $U^* = \frac{\lfloor D_{min} TTRT \rfloor - 1}{\lfloor D_{min}/TTRT \rfloor + 1}(1-\alpha)$ *occurs when* $D_{min}/TTRT$ *is an integer.*

Given this lemma, we can rewrite U^* as a function of an integer argument, say m. Replacing $D_{min}/TTRT$ by m and replacing $\alpha = \tau/TTRT$ by $m\tau/D_{min}$, we have

$$U^* = f(m) = \frac{m-1}{m+1}(1 - \frac{m\tau}{D_{min}}) \tag{20}$$

where m is an integer and $m \geq 2$ (because $\frac{D_{min}}{TTRT} \geq 2$).

Theorem 5 *The worst case achievable utilization* $U^* = \frac{\lfloor D_{min}/TTRT \rfloor - 1}{\lfloor D_{min}/TTRT \rfloor + 1}(1 - \tau/TTRT)$ *for the local synchronous bandwidth allocation scheme defined in (17) is maximized if*

$$TTRT = \frac{D_{min}}{\lceil \frac{-3+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rceil}. \tag{21}$$

Outline of the Proof: By Lemma 2, we know that the maximum value of U^* occurs when $D_{min}/TTRT$ is an integer. Based on this, it can be shown [12] that the range of $D_{min}/TTRT$ that maximizes U^* is given by

$$\lceil \frac{-3+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rceil \leq \frac{D_{min}}{TTRT} \leq \lfloor \frac{-1+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rfloor. \tag{22}$$

Thus, the range of $TTRT$ that maximizes U^* is given by

$$\frac{D_{min}}{\lfloor \frac{-1+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rfloor} \leq TTRT \leq \frac{D_{min}}{\lceil \frac{-3+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rceil}. \tag{23}$$

We note that the value of expression $\lfloor \frac{-1+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rfloor - \lceil \frac{-3+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rceil$ is either zero or one. In the case that $\lfloor \frac{-1+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rfloor - \lceil \frac{-3+\sqrt{9+\frac{8D_{min}}{\tau}}}{2} \rceil = 0$, there is only one possible value for $D_{min}/TTRT$ and the value of $TTRT$ that maximizes U^* is given by (21). Otherwise, there are two choices for $TTRT$ that maximize U^*. It is better to choose a larger value for $TTRT$. This gives a larger available utilization $1 - \tau/TTRT$. Hence $TTRT$ should be chosen as in (21). For a complete proof of this theorem, see [12].

The impact of an appropriate selection of $TTRT$ is evident from Figure 2, which depicts the curve of U^* versus $TTRT$ for several different values of D_{min}. τ is taken to be 0.05, and all time units are normalized in terms of P_{min}. From Figure 2 the following observations can be made:

1. The curves in Figure 2 verify the prediction of the optimal $TTRT$ value given by (21). For example, consider the case of $D_{min} = 2.0$. By (21), the optimal value of $TTRT$ is 0.25. The curve clearly indicates that at $TTRT = 0.25$ the worst case achievable utilization is maximized. Similar observations can be made for the other cases.

2. As indicated in (21), the optimal $TTRT$ is a function of D_{min}. This coincides with the expectations from the observations of Figure 1 in Section 3.5. A general trend is that as D_{min} increases the optimal $TTRT$ increases. For example, the optimal values of $TTRT$ are approximately 0.33 for $D_{min} = 4.0$, 0.47 for $D_{min} = 8.0$, and 0.66 for $D_{min} = 16.0$.

3. The choice of $TTRT$ has a large effect on the worst case achievable utilization U^*. Consider the case of $D_{min} = 4$ shown in Figure 2. If $TTRT$ is too small (say $TTRT = 0.1$), U^* can be as low as 48%. If $TTRT$ is too large (say $TTRT = 2$), U^* can be as low as 33%. However, when the optimal value of $TTRT$ is used (i.e., $TTRT = 0.33$), U^* is 72%. This is an improvement of 24% and 39% respectively over the previous two cases.

5 Buffer Requirements

In this section, the buffer requirements for outgoing synchronous messages are derived. The previous sections have shown that as D_{min} increases, the worst case achievable utilization U^* of the network also increases. It might be expected that this improvement in U^* will result in larger synchronous message queues at node i as D_i increases. This section examines the situation in more detail.

We begin with the following lemma, which bounds the waiting time of a message and the maximum queue size at a node.

Lemma 3 *Let the synchronous bandwidth be allocated as in (17). Let* $w_{i,j}$ *be an upper bound on the waiting time of the j-th message in stream* S_i. *Then* $w_{i,j}$ *is bounded by*

$$w_{i,j} = s_{i,j} - t_{i,j} \leq \min(D_i, P_i + 2TTRT), \tag{24}$$

where $s_{i,j}$, $t_{i,j}$, *and* P_i *are the transmission completion time, the arrival time, and the period of the message, respectively.*

Further, let W_i^{max}, $0 \leq i \leq n$, *denote an upper bound on the number of outgoing synchronous messages that may be waiting on node i (including the message that is being transmitted) at any time.* W_i^{max} *is bounded by*

$$W_i^{max} \leq \begin{cases} 3 & \text{if } TTRT \leq P_i \\ \lceil \frac{2TTRT}{P_i} + 1 \rceil & \text{otherwise.} \end{cases} \tag{25}$$

Lemma 3 can be used to bound the buffer space for outgoing messages required at node i. This is because Lemma 3 bounds the number of outgoing synchronous messages that can be queued on node i, and the size of each message is known in advance as one of the parameters of stream S_i. Thus we have the following theorem.

Theorem 6 *Let b_i be the number of bytes in each synchronous message from stream S_i. Then the buffer at node i need be no more than $b_i W_i^{max}$ bytes.*

It is important to observe the significance of the results obtained here. Earlier, it was shown that with the synchronous bandwidth allocation method proposed in (17), messages will be transmitted quickly enough for the deadline constraint to be satisfied. The results in this section show a further advantage of the allocation scheme. Lemma 3 indicates that the maximum message waiting time and the maximum queue length depend on the network polling speed (i.e., $TTRT$) and the message interarrival time (i.e., period). They do not depend on message deadlines. Consequently, even when message deadlines are very large, the throughput of synchronous messages will be sufficient to prevent a large buildup of queued messages.

The independence of the maximum queue length from message deadlines has significant benefits for system design. In some situations, applications have a choice in setting message deadlines. For example, deadlines for voice transmission can range from several hundred microseconds to half a second. In general, one would prefer larger deadlines in order to improve the worst case achievable utilization. Intuitively, one would expect that this improvement must be traded off with increased requirements for buffer space; if deadlines are too large, buffers may overflow. However, Lemma 3 and Theorem 6 state that this is not true if the allocation scheme in (17) is used. The required buffer size is not changed when deadlines increase. Thus, an application designer can choose message deadlines freely without concern for potential buffer overflow.

6 Deadlines Smaller than Periods

In this section we consider synchronous message sets in which some streams can have a relative deadline smaller than their period.

6.1 Synchronous Bandwidth Allocation for $D_i < P_i$

The synchronous bandwidth must be allocated to satisfy the deadline constraint (Section 2.2). Let S_i be a stream of synchronous messages in which the relative deadline is less than the period. By Corollary 1, in the worst case the token may visit node i no more than $\lfloor \frac{D_i}{TTRT} - 1 \rfloor$ times in an interval of length D_i. The synchronous bandwidth at node i must then be at least

$$H_i = \frac{C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \tag{26}$$

This is the same as (12), derived in Section 3.2. Equation (26) is now a sufficient condition for messages of stream S_i to be transmitted before their deadlines.

Recall that the following allocation scheme was proposed in (17) for the case when $D_i \geq P_i$:

$$H_i = \frac{\max(\frac{q_i TTRT}{P_i}, 1) C_i}{\lfloor \frac{D_i}{TTRT} - 1 \rfloor}. \tag{27}$$

where $q_i = \lfloor \frac{D_i}{TTRT} \rfloor$. In the case of $D_i < P_i$, we have $\frac{q_i TTRT}{P_i} = \frac{\lfloor \frac{D_i}{TTRT} \rfloor TTRT}{P_i} \leq 1$. This means that $\max(\frac{q_i TTRT}{P_i}, 1) = 1$. Hence, (27) implies (26). Thus, (27) is still valid for use in the case when $D_i < P_i$.

Now consider the performance metric: the worst case achievable utilization. The *effective utilization*, U_e, of a set of synchronous messages is defined to be

$$U_e = \sum_{i=1}^{n} \frac{C_i}{\min(P_i, D_i)}. \tag{28}$$

If $D_i \geq P_i$ for all i, then the effective utilization, U_e, is the same as the utilization, U. If $D_i < P_i$ for some i, then the effective utilization reflects the increased demand placed on the network, and $U_e > U$. Using the notion of effective utilization, a network (with a given setting of its parameters) has an *effective* achievable utilization U_e' if it can meet the deadlines of any synchronous message streams with an effective utilization no more than U_e'. The worst case *effective* achievable utilization U_e^* of a network is the least upper bound of its effective achievable utilizations.

With these definitions, if we substitute U_e for U, then the results on the worst case utilization in the earlier sections of this paper still hold in terms of the worst case effective achievable utilization. In particular, Theorem 4 from Section 3 can be simply revised as follows:

Theorem 7 (revised Theorem 4) *If the synchronous bandwidths are allocated using the scheme in (17), then the worst case effective achievable utilization, U_e^*, of the network is given by*

$$U_e^* = \frac{\lfloor \frac{D_{min}}{TTRT} \rfloor - 1}{\lfloor \frac{D_{min}}{TTRT} \rfloor + 1}(1 - \alpha). \tag{29}$$

6.2 Selection of TTRT for $D_i < P_i$

Theorem 5 specifies in (21) the value of $TTRT$ that maximizes the worst case achievable utilization U^* in the case that $D_i \geq P_i$ for $1 \leq i \leq n$. Because the optimal value of $TTRT$ involves only D_{min} and not any particular period P_i, it is still valid for the case that $D_i < P_i$. However, (21) now gives the value of $TTRT$ that maximizes the worst case *effective* achievable utilization.

6.3 Buffer Requirement for $D_i < P_i$

Consider Lemma 3. In the case of $D_i < P_i$, we have $D_i < P_i + 2TTRT$. Hence, (24) becomes

$$w_{i,j} \leq D_i < P_i. \qquad (30)$$

This implies that there will be at most one message waiting at the outgoing synchronous message queue at a node. Therefore, when $D_i < P_i$, the buffer requirements are further reduced because of the tight deadlines.

7 Final Remarks

This paper has proposed a comprehensive solution for transmitting real-time messages from a general message system in an FDDI network. In a general message system, deadlines of messages in a stream can differ from the period of the stream. That is, deadlines can be either less than, equal to, or greater than periods.

We show that synchronous messages with arbitrary deadlines can be guaranteed by selecting the network parameters — the synchronous bandwidths, $TTRT$, and the buffer size — in an integrated fashion. The new results presented in this paper include the following:

- We propose a local synchronous bandwidth allocation scheme. The local allocation scheme utilizes information regarding the length, period, and deadline of synchronous messages at node i in calculating the synchronous bandwidth H_i. In this way, H_i will not be affected by changes in the parameters of synchronous message streams on other nodes. This makes the scheme suited for use in a dynamic environment.

  The worst case achievable utilization U^* of the network when using the proposed synchronous bandwidth allocation scheme was derived. The worst case achievable utilization U^* provides a single criteria to determine whether message deadlines will be met: a message set with utilization no more than U^* is guaranteed to be schedulable.
- A method was developed for selecting $TTRT$ to maximize the worst case achievable utilization of the network with the proposed synchronous bandwidth allocation scheme. Proper selection of $TTRT$ resulted in an improvement in the worst case achievable utilization of approximately 40% for the results reported in this paper. It is important for a network to have a high worst case achievable utilization, because message sets with a high utilization can then be guaranteed.
- An upper bound was derived for the queue length required for synchronous messages. This determines the buffer size for outgoing synchronous messages at each node. Provided that the buffer is allocated based on the maximum queue length, no outgoing synchronous messages will be lost due to buffer overflow. We found that the upper bound on the queue length at node i depends on $TTRT$ and on the message period P_i, but is independent of message deadlines D_i. Thus large deadlines do not cause buffer overflow. This gives the application considerable flexibility in choosing the message deadlines.

References

[1] G. Agrawal, B. Chen, and W. Zhao. Local synchronous capacity allocation schemes for guaranteeing message deadlines with the timed token protocol. Personal communication, also to appear in INFOCOM '93, 1993.

[2] G. Agrawal, B. Chen, W. Zhao, and S. Davari. Guaranteeing synchronous message deadlines in high speed token ring networks with timed token protocol. In *Proceedings of the 12th IEEE International Conference on Distributed Computing Systems*, Yokohama, Japan, June 1992.

[3] ANSI. *Fiber Distributed Data Interface (FDDI) — Token Ring Media Access Control (MAC)*, 1987. ANSI X3.139-1987.

[4] B. Chen, G. Agrawal, and W. Zhao. Optimal synchronous capacity allocation for hard real-time communications with the timed token protocol. In *Proceedings of the Real-Time Systems Symposium*, pages 198–207, Phoenix, Arizona, Dec. 1992.

[5] D. Dykeman and W. Bux. Analysis and tuning of the FDDI media access control protocol. *IEEE Journal on Selected Areas in Communications*, 6(6), July 1988.

[6] ISO. *ISO International Standard 9314-2, Information Processing Systems — Fibre Distributed Data Interface (FDDI) — Part 2: Token Ring Media Access Control (MAC)*, 1989.

[7] R. Jain. Performance analysis of FDDI token ring networks: Effect of parameters and guidelines for setting TTRT. *IEEE LTS*, pages 16–22, May 1991.

[8] A. P. Jayasumana and G. G. Jaysumana. On the use of the IEEE 802.4 token bus in distributed real-time control systems. *IEEE Transactions on Industrial Electronics*, 36(3), 1989.

[9] A. P. Jayasumana and P. N. Werahera. Performance of fibre distributed data interface network for multiple classes of traffic. *IEEE Proceedings*, 137(5):401–408, Sept. 1990.

[10] M. J. Johnson. Proof that timing requirements of the FDDI token ring protocols are satisfied. *IEEE Transactions on Communications*, COM-35(6):620–625, June 1987.

[11] S. Kamat and W. Zhao. A comparative study of two token ring protocols for guaranteeing synchronous real-time messages. In *Proceedings of the 13th IEEE International Conference on Distributed Computing Systems*, Pittsburgh, June 1993.

[12] N. Malcolm. *Hard Real-Time Communication in High Speed Networks.* PhD thesis, Department of Computer Science, Texas A&M University, in preparation.

[13] J. Pang and F. A. Tobagi. Throughput analysis of a timer controlled token passing protocol under heavy load. *IEEE Transactions on Communications*, 37(7):694–702, July 1989.

[14] R. Sankar and Y. Y. Yang. Performance analysis of FDDI. In *Proceedings of the IEEE Conference on Local Computer Networks*, pages 328–332, Oct. 1989.

[15] K. C. Sevcik and M. J. Johnson. Cycle time properties of the FDDI token ring protocol. *IEEE Transactions on Software Engineering*, SE-13(3):376–385, 1987.

[16] K. G. Shin and Q. Zheng. Mixed time-constrained and non-time-constrained communications in local area networks. To appear in IEEE Transactions on Communications.

[17] H. Tokuda, Y. Tobe, S. T.-C. Chou, and J. M. F. Moura. Continuous media communication with dynamic QOS control using ARTS with an FDDI network. In *SIGCOMM'92*, pages 88–98, Aug. 1992.

[18] J. N. Ulm. A timed token ring local area network and its performance characteristics. In *Proceedings of the Conference on Local Computer Networks*, pages 50–56, Feb. 1982.

[19] U.S. Department of Defense. *Survivable Adaptable Fiber Optic Embedded Network*, Sept. 1992. MIL-STD-2204.

[20] A. Valenzano, P. Montuschi, and L. Ciminiera. Some properties of timed token medium access protocols. *IEEE Transactions on Software Engineering*, 16(8), Aug. 1990.

Figure 1: U^* versus D_{min} ($\tau = 0.05$)

Figure 2: U^* versus $TTRT$ ($\tau = 0.05$)