Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Access Control Protocol

Gopal Agrawal, Biao Chen, Wei Zhao, and Sadegh Davari

Abstract—**We study the problem of guaranteeing synchronous message deadlines in token ring networks where the timed token medium access control protocol is employed. Synchronous bandwidth, defined as the maximum time for which a node can transmit its synchronous messages every time it receives the token, is a key parameter in the control of synchronous message transmission. To ensure the transmission of synchronous messages before their deadlines, synchronous capacities must be properly allocated to individual nodes. We address the issue of appropriate allocation of the synchronous capacities. Several synchronous bandwidth allocation schemes are analyzed in terms of their ability to satisfy deadline constraints of synchronous messages. We show that an inappropriate allocation of the synchronous capacities could cause message deadlines to be missed, even if the synchronous traffic is extremely low. We propose a scheme, called the *normalized proportional allocation scheme*, which can guarantee the synchronous message deadlines for synchronous traffic of up to 33% of available utilization.**

Index Terms—**Hard real-time, distributed system, FDDI, timed token medium access control protocol, synchronous messages, performance evaluation and analysis, synchronous bandwidth, worst case achievable utilization.**

NOMENCLATURE

C_i	Length (i.e., transmission time) of a message in synchronous message stream S_i .
H_i	Synchronous bandwidth allocatd to node i.
LC_i	Late counter at node i .
NA_i	Set of asynchronous messages at node i .
NS_i	Set of synchronous message streams at node i .
P_i	Period length of synchronous messages stream S_i .
S_{i_j}	jth synchronous message stream at node i .
$t_i(l)$	Time when the token makes its l th visit to node i .
THT_i	Token holding timer at node i .
TRT_i	Token rotation timer at node i .
$TTRT$	Target Token Rotation Time.
$T(N_i)$	Transformation of node i to a set of virtual nodes.
$U(M)$	Utilization factor of the synchronous messages, i.e., fraction of the time spent by the network in transmission of the synchronous messages.
U_x	Achievable utiliztion of synchronous bandwidth allocation scheme x.
U_x^*	Worst case achievable utilization of synchronous bandwidth allocation scheme x.
VN_{i_j}	jth virtual node derived from node i after its transformation.
X_i	Amount of time available to node i to transmit its synchronous messages within a given period.
m	Number of (virtual) nodes in the network.
n	Number of synchronous message streams in the network. It is assumed that $n = m$.
θ_i	Latency between node i and its upstream neighbor.
Θ	Total ring latency or token walk time.
$\triangle$	Protocol dependent overheads.
τ	Portion of the TTRT that is unavailable to transmit synchronous messages.
α	Ratio of τ to the Target Token Rotation Time (TTRT).

Manuscript received April 7, 1992; revised November 16, 1992 and June 26, 1993. This work was supported in part by AFOSR under Grant F49620-92-J-0385, NSF under Grant NCR-9210583, and Texas A&M University under an Engineering Excellence Grant.

G. Agrawal, B. Chen, and W. Zhao are with the Department of Computer Science, Texas A&M University, College Station, TX 77843.

S. Davari is with the Department of Computer Science, University of Houston-Clear Lake, Houston, TX 77058.

IEEE Log Number 9214072.

I. INTRODUCTION

IT HAS become a common practice to use digital computers for embedded real-time distributed applications such as space vehicle systems, image processing and transmission, and the integration of expert systems into avionics and industrial process control. A salient feature of these computations is that they have stringent timing requirements. A failure to meet the computational deadlines could lead to a catastrophe. Further, these systems are often distributed. This is not only because the applications themselves are often physically distributed, but also due to the potential that distributed systems have for providing good reliability, good resource sharing, and good extensibility [46], [47], [56].

The key to success in using a distributed system for these applications is the timely execution of computation tasks that usually reside on different nodes and communicate with one another to accomplish a common goal. Distributed real-time systems may be categorized as *soft* real-time systems or *hard* real-time systems. In soft real-time systems tasks are performed by the system as fast as possible but are not constrained to finish by a specific time. In hard real-time systems tasks must satisfy explicit time constraints; otherwise, grave consequences may result. Consequently, the messages transmitted in the network by the hard real-time tasks are also time constrained. End-to-end deadline guarantees are possible only if a communication network supports the timely delivery of inter-task messages. The main focus of this study is to address important issues related to guarantees of synchronous message deadlines. A guaranteed message will always be transmitted before its deadline (unless a network fault occurs).

We address the issue of guaranteeing message deadlines with the *timed token medium access control (MAC) protocol* [13]. This protocol is suitable for real-time applications because of its important property of bounded access time which is necessary for real-time communications. The timed token protocol has been incorporated in several high-bandwidth network standards, including the Fiber Distributed Data Interface (FDDI) [3], [4], IEEE 802.4 [15], the High-Speed Data Bus and the High-Speed Ring Bus (HSDB/HSRB) [8], [39], [51], and the Survivable Adaptable Fiber Optic Embedded Network (SAFENET) [12], [22], [33]. Many embedded real-time applications use them as backbone networks.

With the timed token protocol, messages are grouped 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 *Target Token Rotation Time* (TTRT) is determined, which indicates the expected token rotation time. Each station is assigned a fraction of the TTRT, known as *synchronous bandwidth*,[1] which is the maximum time for which a station is permitted to transmit its synchronous messages every time it receives the token. Once a node receives the token, it transmits its synchronous messages, if any, for a time no more than its allocated synchronous bandwidth. It can then transmit its asynchronous messages only if the time elapsed since the previous token departure from the same node is less than the value of TTRT, i.e., only if the token arrived earlier than expected.

Guaranteeing a message deadline implies transmitting the message before its deadline. With a token passing protocol, a node can transmit its message only when it captures the token. This implies that if a message deadline is to be guaranteed, the token should visit the node where the message is waiting before the expiration of the message's deadline. That is, in order to guarantee message deadlines in a token ring network, it is *necessary* to bound the time between two consecutive visits of the token to a node (called the *token rotation time* or *access time*). The timed token protocol possesses this property. In [21], [41], Johnson and Sevcik formally proved that when the network operates normally (i.e., there is no failure), the upper bound on the token rotation time (or the time elapsed between two consecutive visits to a node) is twice the expected token rotation time (i.e., $2 \cdot \mathrm{TTRT}$).

Although the prerequisite of "bounded token rotation time" is indispensable, it is insufficient for guaranteeing message deadlines. A node with inadequate synchronous bandwidth may be unable to complete the transmission of a synchronous message before its deadline. On the other hand, allocating excess amounts of synchronous bandwidth to the nodes could increase the token rotation time, which may also cause message deadlines to be missed. Thus, guaranteeing message deadlines is also dependent on the appropriate allocation of synchronous bandwidth to the nodes. As pointed out in [21], the allocation of synchronous bandwidth is an open problem. The main objective of this study is to analyze and evaluate the synchronous bandwidth allocation schemes used with the timed token protocol in a hard real-time communication system.

[1] Some other synonymous terms that researchers use are: *Bandwidth allocation* [52], *Synchronous allocation* [18], *Synchronous bandwidth assignments* [21], *High Priority token holding time* [34], *Synchronous capacity* [2].

Before discussing details of our work, we will first present an analogy between real-time communication and scheduling to motivate the readers towards the use of our methodology. For real-time systems, the basic design requirements for a communication protocol and for a centralized scheduling algorithm are similar: both are constrained by time to allocate a serially used resource to a set of processes. Liu and Layland [25] addressed the issue of guaranteeing the deadlines of synchronous (i.e., periodic) computation tasks in a single CPU environment. They analyzed a fixed priority preemptive algorithm, called the *rate monotonic algorithm*, which assigns priorities to tasks in a reverse order of the task's periods. They showed that the *worst case achievable utilization* of the algorithm is 69%. As long as the utilization of the task set is no more than 69%, task deadlines are guaranteed to be satisfied. The algorithm was also proven to be optimal among all the fixed priority scheduling algorithms in terms of achieving the highest worst case utilization. The rate monotonic scheduling algorithm has been subsequently extended by many researchers [9], [42] and is used in many hard real-time applications [10].

Intuitively, one would believe that a protocol which implements the rate monotonic transmission policy is the most desirable for a real-time environment. However, implementation of the rate monotonic policy requires global priority arbitration every time a node in the network is ready to transmit a new frame. In a high-speed network, such as the FDDI network, where the bandwidth can be as high as 100 Mbps, the overheads involved in global priority arbitration would be too prohibitive in comparison to the transmission times of the messages themselves. Consequently, it is difficult, if not impossible, to implement the rate monotonic transmission policy in such environments.

However, the methodology for analyzing this algorithm has a more profound significance than merely its relevance to the rate monotonic scheduling. The methodology stresses the fundamental requirement of *predictability* and *stability* in hard real-time environments and is therefore also befitting to other hard real-time scheduling problems. In this methodology, the worst case achievable utilization is used as a metric for evaluating the predictability of a scheduling algorithm. That is, if the CPU utilization of all tasks is within the bounds specified by the metric, all the tasks will meet their deadlines. This metric also gives a measure of the stability of the scheduling algorithm in the sense that the tasks can be freely modified as long as their total utilization is held within the limit. These advantages (of predictability and stability) have led us to adopt the same methodology in our study of guaranteeing message deadlines with the timed token protocol. We aim to analyze synchronous bandwidth allocation schemes based on the worst case achievable utilization.

In this paper, four synchronous bandwidth allocation schemes are analyzed. Our analysis reveals that an improper allocation of the synchronous capacities could lead to a worst case achievable utilization that asymptotically approaches 0%. That is, the deadlines of some messages could be missed even if the synchronous traffic is arbitrarily close to zero. On the other hand, one of the schemes proposed in the paper—the *normalized proportional allocation scheme*—has a Worst Case Achievable Utilization of 33%. That is, as long as the total synchronous traffic is no more than 33%, the synchronous messages are guaranteed to be transmitted before their deadlines (regardless of the number of stations, message lengths, periods, phases, etc.) The remaining 67% of the channel bandwidth could be used by asynchronous traffic. To the best of our knowledge, no other scheme has been reported to achieve a better utilization.

The remainder of the paper is organized as follows: Section II will review the previous relevant work. Section III will outline the characteristics of the system under consideration, i.e., the message and network models. Some properties of the timed token protocol and the synchronous bandwidth allocation schemes are introduced in Section IV. In Section V, we will study several allocation schemes and derive their worst case achievable utilizations. Section VI contains the concluding remarks and suggestions for future work.

II. Previous Relevant Work

Extensive research has been done on the timed token protocol since it was first proposed by Grow [13] and analyzed by Ulm [52] in 1982. Introductory tutorials on this protocol and its use in networking standards can be found in the papers by Ross [36]–[38], Iyer and Joshi [16], [17] and others [28], [44], [45].

The timing properties of the FDDI token ring were first formally analyzed by Johnson and Sevcik in [21], [41]. Other interesting timing properties of the FDDI were given in a study conducted by Jain [18]. He suggests that a value of 8 ms for TTRT is desirable as it can achieve 80% utilization on a wide range of configurations and results in less than 1 second maximum access delay on large rings. Further simulation studies have been carried out by Sankar and Yang [40] to study the influence of the target token rotation time (TTRT) on the performance of various FDDI ring configurations.

Ulm [52] discussed the performance characteristics of the timed token protocol with respect to parameters such as the channel bandwidth, the network cable length, and the number of stations. Dykeman and Bux [11] studied a procedure for estimating the maximum throughput of asynchronous messages when using single and multiple asynchronous priority levels. They also proposed a procedure for tuning the protocol for desired performance by setting appropriate values for the token-holding-time thresholds for each of the priority levels. Other analysis concentrating on the performance of the FDDI with respect to the throughput of asynchronous traffic has been done by Pang and Tobagi [34], Jayasumana and Werahera [19], Valenzo, Montuschi, and Ciminiera [53], etc.

Note that none of the above studies on the timed token protocol have specifically addressed its use and performance in guaranteeing hard real-time message *deadlines*. On the other hand, many studies of CSMA/CD and token ring protocols for distributed hard real-time applications have been conducted. The issues in design and analysis of deadline driven communication protocols for CSMA/CD networks were addressed in [5], [23], [26], [35], [43], [49], [57]–[60]. The real-time performance of various token ring protocols was considered in [24], [30], [43], [48], [55]. Our work reported in this paper complements the previous studies by addressing the issues pertinent to hard real-time communication in a high-speed network where the timed token medium access control protocol is utilized.

III. System Characteristics

In this section, an overview of the system under consideration is given, including the network and message models.

A. Network Model

We consider the network topology as consisting of m nodes connected by point-to-point links forming a circle i.e., a ring. A special bit pattern called the *token* circulates around the ring providing access control among the active nodes.

We denote the latency between a node i and its upstream neighbor[2] by θ_i. This delay includes the node bit delay, the node latency buffer delay, the media propagation delay, etc. The sum total of all such latencies in the ring is known as the *ring latency* Θ, i.e., $\sum_{i=1}^{m} \theta_i = \Theta$. Thus, the ring latency Θ denotes the token walk time around the ring when none of the nodes in the network disturb it.

B. Message Model

Messages generated in the system at run time may be classified as either *synchronous messages* or *asynchronous messages*. We assume that there are n streams of synchronous messages, $S_1, S_2, \cdots, S_n$ in the system which form a synchronous *message set*, M, i.e.,

$$M = \{S_1, S_2, \cdots, S_n\}. \tag{1}$$

The characteristics of messages are as follows:

1) Synchronous messages are *periodic*, i.e., messages in a synchronous message stream have a constant inter-arrival time. We denote P_i to be the period length of stream $S_i (i = 1, 2, \cdots, n)$.
2) The *deadline* of a synchronous message is the end of the period in which it arrives. That is, if a message in stream S_i arrives at time t, then its deadline is at time $t + P_i$.
3) Messages are independent in that message arrivals do not depend on the initiation or the completion of transmission requests for other messages.
4) The *length* of each message in stream S_i is C_i which is the maximum amount of time needed to transmit this message.
5) Asynchronous messages are nonperiodic and do not have a hard real-time deadline requirement.

[2]The upstream neighbor of node i is node $i - 1$ if $i > 1$ else node m if $i = 1$.

The *utilization factor* of a synchronous message set, $U(M)$, is defined as the fraction of time spent by the network in the transmission of the synchronous messages. That is,

$$U(M) = \sum_{i=1}^{n} \frac{C_i}{P_i}, \qquad (2)$$

where n is the number of synchronous message streams.

In the following discussion, we assume that there is one stream of synchronous messages on each node (i.e., $m = n$). In Appendix A, we show that an arbitrary token ring network where a node may have zero, one, or more streams of synchronous messages can be transformed into a logically equivalent network with one stream of synchronous messages per node.[3] Hence, this assumption of one stream per node simplifies the analysis without loss of generality. We also assume that the network is free from hardware or software failures.

IV. Timed Token Medium Access Control Protocol

A. Protocol Parameters

The timed token protocol uses the following parameters and variables for its operation.

1) *Target Token Rotation Time* (TTRT). When the network is initialized, the value of the TTRT is determined, which gives the expected value of the token rotation time. It is selected to be sufficiently small to support the response time requirements of the messages at all the nodes in the network. Since the time elapsed between two consecutive visits of the token at a node can be as much as $2 \cdot \mathrm{TTRT}$ [21], a node may not be able to transmit any message in this interval. Recall that the synchronous messages have their deadlines as the end of their periods. Hence, in order to meet message deadlines it is necessary to select TTRT such that, for $1 \leq i \leq n$,

$$\mathrm{TTRT} \leq \frac{P_i}{2}, \qquad (3)$$

where P_i is the period of synchronous message stream S_i. Any P_i may therefore be represented as a linear function of TTRT. That is

$$P_i = m_i \cdot \mathrm{TTRT} - \delta_i, \qquad (4)$$

where $m_i = \lceil \frac{P_i}{\mathrm{TTRT}} \rceil \geq 2$. If $m_i = 2$, then $\delta_i = 0$ and if $m_i \geq 3$ then $0 \leq \delta_i < \mathrm{TTRT}$. The above expression for P_i has been introduced as it will be useful in several proofs encountered later on. We assume that (3) holds throughout this paper.

2) *Synchronous Bandwidth of Node* $i(H_i)$. This parameter represents the maximum time for which a station is permitted to transmit synchronous messages every time the station receives the token. Note that each station can be assigned a different H_i value.[4] This paper will deal with the issue of appropriate allocation of these H_i values.

3) *Token Rotation Timer of node* $i(\mathrm{TRT}_i)$. This counter is initialized to equal TTRT, and counts down until it expires (i.e., $\mathrm{TRT}_i = 0$) or until the token is received and the time elapsed since the previous token departure is less than TTRT. In either situation, the TRT_i is reinitialized to TTRT. After being reset, it continues the subsequent counting down cycles in the same manner as above.

4) *Token Holding Timer of node* i (THT_i). This (down) counter is used to control the amount of time for which the node i can transmit asynchronous messages.

5) *Late Counter of node* i (LC_i). This counter is used to record the number of times that TRT_i has expired since the last token arrival at node i.

[3] Furthermore, if multiple message streams at a node are queued into multiple queues (one message stream per queue), then it would require several MACs to implement this. This is not practical.

[4] In FDDI stations, the assignment of H_i to station i is a function of the station management entity of the FDDI protocol.

B. Protocol Operation

At ring initialization, the following parameters are initialized at all nodes:

1) $\mathrm{THT}_i \leftarrow 0$;
2) $\mathrm{LC}_i \leftarrow 0$;
3) $\mathrm{TRT}_i \leftarrow \mathrm{TTRT}$.

TRT_i counter always counts down. When it reaches zero, the following actions take place:

1) $\mathrm{TRT}_i \leftarrow \mathrm{TTRT}$;
2) $\mathrm{LC}_i \leftarrow \mathrm{LC}_i + 1$.

TRT_i then begins the counting down process again with LC_i being incremented by one at every expiration of TRT_i. Normally, if LC_i exceeds one, the ring recovery process is initiated [20].

A token is considered to arrive *early* at node i if $\mathrm{LC}_i = 0$ and *late* at node i if $\mathrm{LC}_i > 0$ at the time of its arrival.

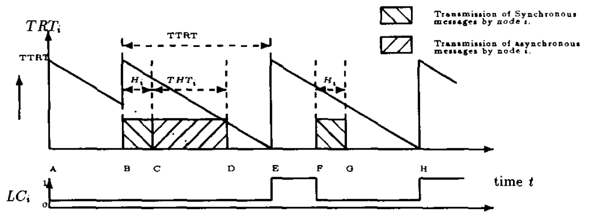
When the token arrives *early* at node i, the following actions take place:

1) $\mathrm{THT}_i \leftarrow \mathrm{TRT}_i$;
2) $\mathrm{TRT}_i \leftarrow \mathrm{TTRT}$;
3) Synchronous frames (if any) can then be transmitted for a maximum time of H_i (i.e., the synchronous bandwidth at node i);
4) After transmitting synchronous frames (if any), the station enables counter THT_i (i.e., it starts counting down). The station may then transmit asynchronous frames as long as $\mathrm{THT}_i > 0$ and $\mathrm{TRT}_i > 0$.

When the token arrives *late* at node i, the following actions take place:

1) $\mathrm{LC}_i \leftarrow 0$;
2) TRT_i continues to count down towards expiration (note that it is *not* reset to TTRT as in the case when the token is early);
3) node i can transmit synchronous frames for a maximum time of H_i;
4) no asynchronous frame will be transmitted.

Fig. 1 shows an example of how TRT_i and LC_i (at some node i) vary with time t. At point B in the figure, the node receives the token early. At point F, the token is received



A: TRT_i is reset to TTRT. $LC_i \leftarrow 0$. Token is released.

B: Token is received early. $THT_i \leftarrow TRT_i$. $TRT_i \leftarrow TTRT$. $LC_i \leftarrow 0$.

C: Synchronous messages transmitted in interval B-C. THT_i starts count down.

D: Asynchronous messages transmitted in interval C-D. Token released.

E: TRT_i expires and is reset to TTRT. $LC_i \leftarrow 1$.

F: Token received late since $LC_i = 1$. $LC_i \leftarrow 0$. Transmit synchronous frames only.

G: Allocated capacity for synchronous transmission expires. Token is released.

H: TRT_i expires again and is reset to TTRT. $LC_i \leftarrow 1$.

Fig. 1. An example of TRT_i and LC_i versus time t.

late. Synchronous messages are transmitted in both cases, but asynchronous messages are transmitted only when the token arrives early.

C. Synchronous Bandwidth Allocation Schemes

As mentioned earlier, synchronous bandwidth allocation plays an important role in guaranteeing synchronous message deadlines. In this subsection, we formally present the definition of allocation schemes and discuss their requirements and performance metrics.

Definition: The synchronous message parameters (given by the C_i's and P_i's) at the various stations and the Target Token Rotation Time (TTRT) should be the dictating factors for the allocation of the H_i's. We define a synchronous bandwidth *allocation scheme* as an algorithm which, when given as input the values of all C_i and P_i in the message set and the value of TTRT, will produce as output the values of the synchronous bandwidth H_i to be allocated to station i in the network. Formally, let function f represent an allocation scheme. Then,

$$f(C_1, C_2, \cdots C_n, P_1, P_2, \cdots P_n, \text{TTRT}) = (H_1, H_2, \cdots H_n). \tag{5}$$

Let us consider a simple example. We assume a network with three nodes. We have the following values for the message set's parameters:

$$\begin{aligned} C_1 &= 1/2, P_1 = 1, \\ C_2 &= 1/2, P_2 = 2, \\ C_3 &= 1/2, P_3 = 2. \end{aligned} \tag{6}$$

The value of TTRT is assumed to be $1/2$. Using an allocation scheme where

$$H_i = \frac{C_i}{P_i} \cdot \text{TTRT}, \tag{7}$$

we obtain the values of synchronous bandwidth as:

$$\begin{aligned} H_1 &= \frac{C_1}{P_1} \cdot \text{TTRT} = 1/4, \\ H_2 &= \frac{C_2}{P_2} \cdot \text{TTRT} = 1/8, \\ H_3 &= \frac{C_3}{P_3} \cdot \text{TTRT} = 1/8, \end{aligned} \tag{8}$$

i.e., $f(C_1, C_2, C_3, P_1, P_2, P_3, \text{TTRT}) = (\frac{1}{4}, \frac{1}{8}, \frac{1}{8})$.

In Section V, we will introduce several other allocation schemes and analyze their effect on the real-time performance of the network. Before that, we will discuss the general requirements that any allocation scheme should satisfy.

Requirements: The synchronous capacities allocated to the nodes by any scheme must satisfy the two constraints given below in order to ensure that the real-time messages can be transmitted before their deadlines and that the timed token protocol requirements are satisfied.

- *Protocol constraint*: Theoretically, the total available time to transmit synchronous messages, during one complete traversal of the token around the ring, can be as much as TTRT. However, factors such as ring latency Θ and other protocol/network dependent overheads reduce the total available time to transmit the synchronous messages. We denote the portion of TTRT unavailable for transmitting synchronous messages by τ. That is, $\tau = \Theta + \Delta$ where Δ represents the protocol dependent overheads.[5] We define the ratio of τ to the target token rotation time (TTRT) to be α. The usable ring utilization available for synchronous messages would therefore be $(1 - \alpha)$ [52].
  Thus, a protocol constraint on the allocation of synchronous bandwidth is that *the sum total of the synchronous bandwidth allocated to all nodes in the ring should not be greater than the available portion of the Target Token Rotation Time* (TTRT), i.e.,
  $$\sum_{i=1}^{n} H_i \leq \text{TTRT} - \tau. \tag{9}$$
- *Deadline constraint*: *The allocation of the synchronous bandwidth to the nodes should be such that the synchronous messages are always guaranteed to be transmitted before their deadlines, i.e., before the end of the period in which they arrived.* In other words, if X_i is the minimum amount of time available for node i to transmit its synchronous messages in a time interval $(t, t + P_i)$, then
  $$X_i \geq C_i. \tag{10}$$

Note that X_i will be a function of H_i and the number of token visits to node i in time interval $(t, t + P_i)$.

We say a message set is *guaranteed* by an allocation scheme if both the protocol and the deadline constraints are satisfied. Once a message set is guaranteed, messages will be transmitted before their deadlines, as long as the network operates normally.

[5]For example, according to the FDDI standard, the protocol dependent overheads include the token transmission time, asynchronous overrun, etc. Refer to [3] for details.

Performance Metric: Numerous synchronous bandwidth allocation schemes can be proposed. An appropriate metric is needed in order to evaluate and compare the effects of allocation schemes on the performance of the network.

As mentioned in Section I, we adopt the methodology developed in analyzing the rate monotonic scheduling algorithm. As per this methodology, the *worst case achievable utilization* will be used as the metric for evaluating and comparing the allocation schemes.

We say that U_x is an *achievable utilization* of scheme x if scheme x can guarantee every synchronous message set whose utilization factor is less than or equal to U_x. The *worst case achievable utilization* (U_x^*) of a scheme x is the least upper bound of its achievable utilizations U_x. That is, as long as the utilization factor of a synchronous message set is no more than U_x^*, the message set can be guaranteed by scheme x. In a hard real-time system, we consider one scheme to be better than another if its worst case achievable utilization is higher. When the context is clear, we may omit the index in the notations of U_x and U_x^*.

The major advantages of this metric are as follows.

- This metric evaluates the predictability of a hard real-time communication systems. If the utilization of a synchronous message set is within the bound specified by the metric, all synchronous messages in the set will meet their deadlines.
- This metric also gives a measure of the stability of the system in the sense that the parameters of synchronous messages can be freely changed as long as their total utilization is held within the limit.
- In practice, using this metric simplifies network management considerably while configuring the system, as it eliminates the problem of being encumbered with individual values of synchronous and asynchronous message lengths, inter-arrival periods, phase differences between message arrivals, relative positions of the nodes, token position at initialization, etc. As long as the network manager can ensure that the total utilization of the time-critical synchronous messages is no more than the worst case achievable utilization of the protocol, he/she can be cognizant of the fact that the message set will be transmitted with no deadlines being missed.

The objective of this paper is to derive the worst case achievable utilization for synchronous bandwidth allocation schemes.

D. Protocol Timing Properties

Analyzing an allocation scheme requires that we test if both the protocol and the deadline constraints are satisfied. Testing of the deadline constraint is especially challenging because it involves both network parameters (e.g., H_i, TTRT, and τ) and message parameters (e.g., C_i and P_i). In particular, we need to know the minimum available time (i.e., the tight lower bound) within a given time period during which a node can transmit its synchronous messages. This is directly related to the minimum number (i.e., the tight lower bound) of token visits to a node that may occur within some period.

Fortunately, extensive work has been done on the timing behavior of the timed token protocol and the minimum time available to a node for transmitting its synchronous messages has been obtained. Johnson and Sevcik showed that any two consecutive token visits to a node are bounded by $2 \cdot \mathrm{TTRT}$ which is stated in the following theorem.

Let $t_i(l)$ $(l = 1, 2, \cdots)$ denote the time when the token makes its lth visit to node i.

Theorem 4.1 (Johnson and Sevcik's Theorem [21], [41]): For any integer $l > 0$ and any node $i (1 \le i \le n)$,

$$t_i(l+1) - t_i(l) \le \mathrm{TTRT} + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau \le 2 \cdot \mathrm{TTRT}. \tag{11}$$

This theorem gives the upper bound between two consecutive token arrivals as $2 \cdot \mathrm{TTRT}$. A formal proof for the above result was first obtained by Johnson and Sevcik in [21], [41]. Using this result, we can obtain a lower bound on the minimum number of token visits to a node within the period of its synchronous message stream. However, this bound is not tight when the period is longer than $3 \cdot \mathrm{TTRT}$. The result obtained by Johnson and Sevcik has been further generalized in [7] to obtain a tight bound on the time elapsed between any v consecutive visits by the token to a particular node.

Theorem 4.2 (Generalized Johnson and Sevcik's Theorem): For any integer $l > 0, v > 0$ and any node i $(1 \le i \le n)$,

$$t_i(l+v-1) - t_i(l) \le (v-1) \cdot \mathrm{TTRT} + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau. \tag{12}$$

Refer to [7] for a proof of this theorem. This theorem indicates an upper bound on the maximum time that could possibly elapse between any v consecutive token arrivals. Johnson and Sevcik's Theorem is a special case when $v = 2$. This result has been used for the derivation of a lower bound on the time available for a node to transmit its synchronous messages within a given time period.

Corollary 4.1: If at time t a synchronous message with period P_i arrives at node $i (1 \le i \le n)$, then in time interval $(t, t + P_i)$ the minimum amount of time (X_i) available for node i to transmit its synchronous message is given by

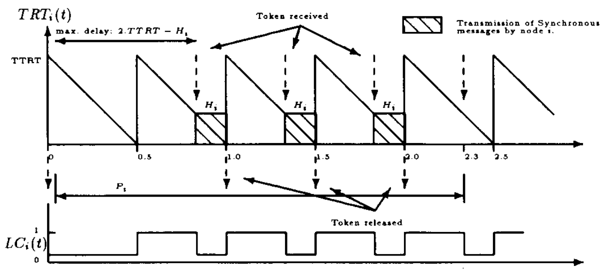
$$\begin{aligned} X_i &= \lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor \cdot H_i + \max(0, \min(\mathrm{TTRT} - (\delta_i \\ &\quad + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau), H_i)) \\ &\ge \lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor \cdot H_i. \end{aligned} \tag{13}$$

Refer to [7] for a proof of the above corollary. Note that if

$$\delta_i = \lceil \frac{P_i}{\mathrm{TTRT}} \rceil \cdot \mathrm{TTRT} - P_i \ge \mathrm{TTRT} - \left(\sum_{h=1,\cdots,n,h\neq i} H_h + \tau \right), \tag{14}$$

the second term of the right-hand side of (13) is zero. Hence, (13) becomes

$$X_i = \left\lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \right\rfloor \cdot H_i. \tag{15}$$



Synchronous capacity $H_i = 0.2$ $TTRT = 0.5$

Message length $C_i = 0.8$. Message period $P_i = 2.3$

Total time available for transmitting synchronous messages in the worst case:

$= \lfloor \frac{P_i}{TTRT} - 1 \rfloor H_i = \lfloor \frac{2.3}{0.5} - 1 \rfloor 0.2 = 0.6$ units.

Fig. 2. Token arrivals at node i in a worst case situation.

The expressions of X_i given by (13) and (15) will be used in the analysis of our synchronous bandwidth allocation schemes. Fig. 2 shows an example of a worst case scenario where the amount of time for which a node can transmit its synchronous messages is given by the lower bound of (15): In the first count down cycle of TRT_i, the node in the figure does not receive the token at all. This may happen because some other node is transmitting its asynchronous messages during this cycle. In the second, the third, and the fourth cycles, all nodes can transmit only synchronous messages (as the token will visit the nodes "late" in these time intervals). In the fifth cycle, the node i receives the token too late to transmit its remaining 0.2 units of synchronous messages before the time $t = 2.3$, which happens to be the deadline. That is, node i is able to transmit its synchronous message for 0.6 units of time only; as can be predicted by (15).

V. Analysis of Synchronous Bandwidth Allocation Schemes

In this section, we consider four synchronous bandwidth allocation schemes and derive their worst case achievable utilizations. While the worst case achievable utilization of the first two schemes is asymptotically close to 0%, the third and fourth schemes achieve a nonzero worst case utilization.

We define $P_{\min} = \min\{P_1, P_2, P_3, \ldots, P_n\}$. To simplify our analysis we assume that $P_{\min}$ is normalized to one unit of time. That is, all other time variables such as P_is, C_is, H_is, etc., are measured in this reference time unit.

The underlying principle for computing the worst case achievable utilization is simple. Given any allocation scheme, we can compute the synchronous bandwidth (H_i) available to each node i. Both protocol and deadline constraints must be satisfied by the allocation of these synchronous bandwidth. Message sets with the least possible utilization factors are then searched such that the allocation of the synchronous bandwidth does not satisfy at least one of the constraints. That gives the upper bound on the utilization factor of message sets i.e., any message set with a utilization factor below that bound will be transmitted successfully without violating either the protocol or the deadline constraints. This then represents the worst case achievable utilization of the allocation scheme.

The following lemma will be used in our analysis. Its proof is presented in Appendix B.

Lemma 5.1: For any synchronous message stream i $(1 \leq i \leq n)$ we have

$$\frac{\lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor}{P_i/\mathrm{TTRT}} \geq \frac{1}{3 - \frac{\delta_i}{\mathrm{TTRT}}} \geq \frac{1}{3}. \tag{16}$$

A. Full Length Allocation Scheme

With this scheme, the synchronous bandwidth allocated to a node is equal to its total time required for transmitting its synchronous messages, i.e.,

$$H_i = C_i. \tag{17}$$

This scheme attempts to transmit a synchronous message in a single turn *rather than* splitting it into chunks and distributing its transmission over its period P_i. Although the synchronous bandwidth allocated is sufficient, the worst case achievable utilization is zero because the protocol constraint may be violated, as shown in the next theorem.

Theorem 5.1: The worst case achievable utilization of the full length allocation scheme can asymptotically approach 0%.

Proof: We prove the theorem by showing that for any given $\epsilon > 0$, there exists a message set M such that $U(M) \leq \epsilon$ and the protocol constraint cannot be satisfied when the synchronous bandwidth of the nodes is allocated using the full length scheme.

Let $\mathrm{TTRT} = \frac{1}{k}$ where $k \geq 2$. This is because by (3), $\mathrm{TTRT} \leq P_{\min}/2 = 1/2$. Now, for any given $\epsilon > 0$ and $\tau > 0$, we construct a set of synchronous messages as follows:

$$C_1 = \left(1 - \frac{1}{k}\right)\epsilon, P_1 = 1,$$
$$C_2 = \frac{2-\epsilon}{k}, P_2 = \frac{2-\epsilon}{\epsilon}.$$

All other $C_i = 0$ for $i > 2$.

The utilization factor is

$$U = \sum_{i=1}^{n} C_i/P_i = \frac{(1 - \frac{1}{k})\epsilon}{1} + \frac{(2-\epsilon)/k}{(2-\epsilon)/\epsilon} = \epsilon. \tag{18}$$

With this set of messages, we can show that the protocol constraint is not satisfied, i.e., the total of all synchronous bandwidth exceeds $\mathrm{TTRT} - \tau$. That is,

$$\sum_{i=1}^{n} H_i = \sum_{i=1}^{n} C_i = \left(1 - \frac{1}{k}\right)\epsilon + \frac{(2-\epsilon)}{k}$$
$$= \frac{2}{k} + \epsilon\left(1 - \frac{2}{k}\right). \tag{19}$$

Since $k \geq 2$, $(1 - \frac{2}{k}) \geq 0$. Therefore,

$$\sum_{i=1}^{n} H_i \geq \frac{2}{k} \geq \mathrm{TTRT} - \tau. \tag{20}$$

We see that this scheme may over-allocate the synchronous bandwidth for a message set with utilization $U \leq \epsilon$. Protocol constraint (9) is therefore not satisfied. Since ϵ can be arbitrarily close to 0, the worst case achievable utilization of this scheme can asymptotically approach 0%. □

B. Proportional Allocation Scheme

With this scheme, the synchronous bandwidth allocated to a node is proportional to the ratio of C_i and P_i at node i, i.e.,

$$H_i = \frac{C_i}{P_i} \cdot (\mathrm{TTRT} - \tau). \tag{21}$$

Theorem 5.2: The worst case achievable utilization of the proportional scheme can asymptotically approach 0%.

Proof: We prove the theorem by showing that for any given $\epsilon > 0$, there exists a message set M such that $U(M) \leq \epsilon$ and the deadline constraint cannot be satisfied when the synchronous bandwidth of the nodes is allocated using the proportional scheme.

Let TTRT $= 1/k$ where k is an integer and $k > 2$. Given any $\epsilon > 0$, let $\epsilon' = \min(\epsilon, \frac{1}{2k})$. Consider a message set with the following parametric values:

$$C_1 = \left(1 - \frac{1}{k}\right)\epsilon', \quad P_1 = 1,$$
$$C_2 = \left(1 + \frac{1}{k} - \epsilon'\right)\frac{\epsilon'}{k}, P_2 = 1 + \frac{1}{k} - \epsilon'. \tag{22}$$

All other $C_i = 0$ for $i > 2$. We assume that $\tau = 0$.

The utilization factor is

$$U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \left(\epsilon' - \frac{\epsilon'}{k}\right) + \frac{\epsilon'}{k} = \epsilon' \leq \epsilon. \tag{23}$$

The synchronous bandwidth allocated to node 1 is given by

$$H_1 = \frac{C_1}{P_1} \cdot (\mathrm{TTRT} - \tau) = \frac{\epsilon'}{k} - \frac{\epsilon'}{k^2}. \tag{24}$$

The synchronous bandwidth allocated to node 2 is given by

$$H_2 = \frac{C_2}{P_2} \cdot (\mathrm{TTRT} - \tau) = \frac{\epsilon'}{k^2}. \tag{25}$$

From (13), the minimum amount of time (X_2) for node 2 to transmit its synchronous message in a period P_2 is given by

$$X_2 = \lfloor \frac{P_2}{\mathrm{TTRT}} \rfloor \cdot H_2 + \max(0, \min(\mathrm{TTRT} - (\delta_2 + H_1 + \tau), H_2)). \tag{26}$$

Now,

$$\begin{aligned}\delta_2 &= \lceil \frac{P_2}{\mathrm{TTRT}} \rceil \cdot \mathrm{TTRT} - P_2 \\ &= \lceil \frac{1 + 1/k - \epsilon'}{1/k} \rceil \cdot \frac{1}{k} - \left(1 + \frac{1}{k} - \epsilon'\right). \\ &= \lceil k + 1 - k\epsilon' \rceil \cdot \frac{1}{k} - \left(1 + \frac{1}{k} - \epsilon'\right) \\ &= (k+1) \cdot \frac{1}{k} - \left(1 + \frac{1}{k} - \epsilon'\right) = \epsilon' \quad (\text{since } k\epsilon' \leq \frac{1}{2}).\end{aligned} \tag{27}$$

Hence,

$$\begin{aligned}\mathrm{TTRT} - (\delta_2 + H_1 + \tau) &= \frac{1}{k} - \epsilon' - \frac{\epsilon'}{k} + \frac{\epsilon'}{k^2} \\ &= \frac{1 - (k+1)\epsilon'}{k} + \frac{\epsilon'}{k^2} > \frac{\epsilon'}{k^2} \\ &= H_2.\end{aligned} \tag{28}$$

Substituting (28) into (26), we have

$$\begin{aligned}X_2 &= \lfloor \frac{P_2}{\mathrm{TTRT}} - 1 \rfloor \cdot H_2 + H_2 = \lfloor \frac{P_2}{\mathrm{TTRT}} \rfloor \cdot H_2 \\ &= \lfloor \frac{1 + 1/k - \epsilon'}{1/k} \rfloor \cdot \frac{\epsilon'}{k^2} = \frac{\epsilon'}{k} \\ &< C_2.\end{aligned} \tag{29}$$

We see that deadline constraint (10) cannot be satisfied at node 2. Since ϵ can be arbitrarily close to 0, the worst case achievable utilization of this scheme can asymptotically approach 0%. □

Intuitively speaking, this scheme divides the transmission of its message into as many parts as the number of times the token is *expected* to arrive at node i within its period P_i. However, since the token could be late by as much as $2 \cdot$ TTRT, the number of token arrivals may be less than expected. Hence, node i may not be able to complete the transmission of some part of a message before the end of period P_i.

C. Equal Partition Allocation Scheme

In this scheme, the usable portion of TTRT is divided equally among the n nodes for allocating their synchronous capacities, i.e.,

$$H_i = \frac{\mathrm{TTRT} - \tau}{n}, \tag{30}$$

where n is the number of nodes in the system.

Theorem 5.3: The worst case achievable utilization of the equal partition synchronous bandwidth allocation scheme is $\frac{1}{3n-(1-\alpha)} \cdot (1 - \alpha)$ where $\alpha = \frac{\tau}{\mathrm{TTRT}}$ and n is the number of nodes.

This theorem can be proved by showing that the following statements are true.

1) For any message set M, the protocol constraint will be satisfied.
2) For any message set M with utilization factor $U(M) \leq \frac{1-\alpha}{3n-(1-\alpha)}$, the deadline constraint will be satisfied.
3) For any given $\epsilon > 0$, there exists a message set M with utilization factor $U(M) = \frac{1-\alpha}{3n-(1-\alpha)} + \epsilon$, so that the deadline constraint cannot be satisfied for this set of messages when the synchronous bandwidth are allocated by using the equal partition scheme.

Interested readers can refer to [2] for a detailed proof.

Note that when the number of nodes, n, becomes very large, the worst case achievable utilization of this scheme is approximately 0%. Intuitively speaking, the low worst case achievable utilization of this scheme occurs because the allocation of the synchronous bandwidth to the nodes is not proportional to the synchronous traffic load offered by the nodes (i.e., the ratio of C_i/P_i). The normalized proportional scheme discussed next attempts to overcome this problem by allocating the synchronous bandwidth to a node depending on local message parameters such as C_i/P_i and the total utilization factor of all the synchronous messages in the system.

D. Normalized Proportional Allocation Scheme

With this scheme, the synchronous bandwidth is allocated according to the normalized load of the synchronous message on a node, i.e.,

$$H_i = \frac{C_i/P_i}{U} \cdot (\mathrm{TTRT} - \tau), \tag{31}$$

where $U = \sum_{i=1}^{n} C_i/P_i$.

Theorem 5.4: The worst case achievable utilization factor of the normalized proportional allocation scheme is $\frac{1}{3}(1-\alpha)$ where $\alpha = \frac{\tau}{\mathrm{TTRT}}$.

Proof: To prove the theorem, we show that the following statements are true.

1) For any message set M, the protocol constraint will be satisfied if $\Sigma_{i=1}^{n} \frac{C_i}{P_i} = U \le 1$.
2) For any message set M with utilization factor $U(M) \le \frac{1}{3}(1-\alpha)$, the deadline constraint will always be satisfied.
3) For any given $\epsilon > 0$, there exists a message set M with utilization factor $\frac{1}{3}(1-\alpha) < U(M) \le \frac{1}{3}(1-\alpha) + \epsilon$ so that the deadline constraint cannot be satisfied for this set of messages when the synchronous capacities are allocated by the normalized proportional scheme.

Proof of Statement 1: For any message set M with $\sum_{i=1}^{n} \frac{C_i}{P_i} = U \le 1$,

$$\sum_{i=1}^{n} H_i = \sum_{i=1}^{n} \frac{C_i/P_i}{U} \cdot (\mathrm{TTRT} - \tau) = \mathrm{TTRT} - \tau. \tag{32}$$

Hence, the protocol constraint (9) is satisfied.

Proof of Statement 2: Consider a message set whose utilization factor $U(M) \le \frac{1}{3}(1-\alpha)$. From Lemma 5.1, we have

$$\begin{aligned} U &\le \frac{1}{3}(1-\alpha) \le \frac{\lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor}{P_i/\mathrm{TTRT}}(1-\alpha) \\ &\le \frac{\lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor}{P_i}(\mathrm{TTRT} - \tau). \end{aligned} \tag{33}$$

Multiplying with C_i/U on both sides, we get

$$C_i \le \frac{\frac{\lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor}{P_i} \cdot (\mathrm{TTRT} - \tau) \cdot C_i}{U}. \tag{34}$$

That is, for $1 \le i \le n$,

$$C_i \le \lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor \cdot \frac{C_i}{P_i U} \cdot (\mathrm{TTRT} - \tau). \tag{35}$$

Substituting $\frac{C_i/P_i}{U} \cdot (\mathrm{TTRT} - \tau) = H_i$, we have

$$C_i \le \lfloor \frac{P_i}{\mathrm{TTRT}} - 1 \rfloor \cdot H_i. \tag{36}$$

From (13) and (36), we see that any node i can transmit its synchronous message before the deadline.

Proof of Statement 3: For any given $\epsilon > 0$, let

$$\epsilon' = \min\left(\frac{1-\alpha}{3}, \epsilon\right), \tag{37}$$

where $\alpha = \frac{\tau}{\mathrm{TTRT}}$. Let TTRT $= \frac{1}{2}$. Consider the following message set:

$$\begin{aligned} C_1 &= \epsilon', & P_1 &= 1, \\ C_2 &= \epsilon', & P_2 &= \frac{3}{2} - \epsilon', \\ C_3 &= 1 - 3\epsilon' - \alpha, & P_3 &= 3. \end{aligned} \tag{38}$$

Note that (37) guarantees that $C_3 \ge 0$. All other $C_i = 0$ for $i > 3$.

The utilization of this message set is

$$U = \sum_{i=1}^{3} \frac{C_i}{P_i} = \frac{1}{3}(1-\alpha) + \frac{\epsilon'}{(3/2) - \epsilon'}. \tag{39}$$

Since $\frac{3}{2} - \epsilon' > 1$ and $\epsilon' \le \epsilon$, we have

$$U < \frac{1}{3}(1-\alpha) + \epsilon' \le \frac{1}{3}(1-\alpha) + \epsilon. \tag{40}$$

Consider the synchronous bandwidth allocated to node 2:

$$\begin{aligned} H_2 &= \frac{C_2}{P_2 \cdot U} \cdot (\mathrm{TTRT} - \tau) = \frac{C_2}{P_2 \cdot U} \cdot \mathrm{TTRT} \cdot (1-\alpha) \\ &= C_2 \cdot \frac{\frac{1}{2}(1-\alpha)}{\left(\frac{3}{2} - \epsilon'\right)\left(\frac{1}{3}(1-\alpha) + \frac{\epsilon'}{\frac{3}{2}-\epsilon'}\right)} \\ &= C_2 \cdot \frac{\frac{1}{2}(1-\alpha)}{\frac{1}{2}(1-\alpha) - \frac{1}{3}\epsilon'(1-\alpha) + \epsilon'} \\ &= C_2 \cdot \frac{1}{1 + \frac{2\epsilon'}{(1-\alpha)}\left(1 - \frac{1-\alpha}{3}\right)}. \end{aligned} \tag{41}$$

Since $0 < \epsilon' \le \frac{1}{3}$ and $0 \le \alpha < 1$, the denominator of Equation (41) is greater than 1. Hence,

$$H_2 < C_2. \tag{42}$$

From the proof of statement 1 of this theorem, we have

$$\sum_{i=1}^{n} H_i = \mathrm{TTRT} - \tau. \tag{43}$$

That is

$$H_2 = \mathrm{TTRT} - \left(\sum_{h=1,\cdots,n,h\ne 2} H_h + \tau\right). \tag{44}$$

We now show that $\delta_2 \ge \mathrm{TTRT} - (\sum_{h=1,\cdots,n,h\ne 2} H_h + \tau)$, i.e., $\delta_2 \ge H_2$. By (4) and (42), we have

$$\begin{aligned} \delta_2 &= \lceil \frac{P_2}{\mathrm{TTRT}} \rceil \cdot \mathrm{TTRT} - P_2 \\ &= \lceil \frac{3/2 - \epsilon'}{1/2} \rceil \cdot \frac{1}{2} - \left(\frac{3}{2} - \epsilon'\right) = \lceil 3 - 2\epsilon' \rceil \cdot \frac{1}{2} - \frac{3}{2} + \epsilon' \\ &= \frac{3}{2} - \frac{3}{2} + \epsilon' = \epsilon' = C_2 > H_2. \end{aligned} \tag{45}$$

From (44) and (45), we have $\delta_2 >$ TTRT$-$ $(\sum_{h=1,\cdots,n,h\ne 2} H_h + \tau)$. Thus, the minimum amount of time (X_2) for node 2 to

TABLE I
SUMMARY OF THE SYNCHRONOUS BANDWIDTH ALLOCATION SCHEMES

Name	Formula of H_i	$W.C.A.U.*$	Comments
Full length	$H_i = C_i$	0	Uses local information only, i.e., C_i.
Proportional	$H_i = \frac{C_i}{P_i} \cdot (\mathrm{TTRT} - \tau)$	0	Uses local information only, i.e., $\frac{C_i}{P_i}$.
Equal partition	$H_i = \frac{\mathrm{TTRT}-\tau}{n}$	$\frac{1-\alpha}{3n-(1-\alpha)}$	Uses global information only, i.e., the number of nodes n.
Normalized proportional	$H_i = \frac{C_i/P_i}{U} \cdot (\mathrm{TTRT} - \tau)$	$\frac{1-\alpha}{3}$	Uses both local and global information, i.e., load on the system (U) and the load offered by local message streams ($\frac{C_i}{P_i}$).

*W.C.A.U is the abbreviation of "worst case achievable utilization."

transmit its synchronous message in a time interval $(t, t+P_2)$ is given by (15), i.e.,

$$\begin{aligned} X_2 &= \lfloor \frac{P_2}{\mathrm{TTRT}} - 1 \rfloor \cdot H_2 \\ &= \lfloor \frac{(3/2)-\epsilon'}{1/2} - 1 \rfloor \cdot H_2 \\ &= H_2. \end{aligned} \tag{46}$$

From (42) and (46), we have

$$X_2 < C_2. \tag{47}$$

Therefore, deadline constraint (10) is violated and this set of messages cannot be guaranteed. □

In the normalized proportional allocation scheme, both local information (i.e., C_i and P_i) and global information (i.e., U and TTRT) are used. It results in a normalization of the allocated synchronous capacities, thereby achieving a worst case achievable utilization equal to 33% of the available ring utilization.

In order to verify the worst case achievable utilization derived in this section, simulation experiments were carried out. An FDDI token ring network was simulated by a program written in Simscript II on a Sun Sparc workstation. At the beginning of each simulation, the parameters of synchronous message streams (i.e., C_i and P_i) were generated randomly. In order to simulate a worst case scenario, the asynchronous message queues were assumed to be inexhaustible. For each allocation scheme, 100 000 message sets were simulated. It was found that none of the message sets, with a utilization lower than that of the worst case achievable utilization of the allocation scheme employed, missed any deadline. For a detailed description of the simulation program and the data analysis, see [1].

VI. CONCLUSION

Guaranteeing message deadlines is a key issue in distributed real-time applications. The property of the bounded token rotation time of the timed token protocol provides a necessary condition to ensure that the message deadlines are satisfied. However, the synchronous bandwidth allocated to each node in the network was also shown to be a decisive factor in guaranteeing time-critical messages. The worst case achievable utilization was used as the metric to evaluate and compare various allocation schemes. This metric is of importance to real-time applications because it is related to the predictability and the stability of the system.

Table I summarizes the four allocation schemes discussed in this paper. Their worst case achievable utilizations range from 0% to 33% of available utilization. To explore the performance differences, we categorize the allocation schemes based on the type of information they use. An allocation scheme is *local* if it computes the synchronous bandwidth of a node without using the information of messages on other nodes. Hence, the allocation function of a local scheme has the form

$$H_i = f(C_i, P_i, \mathrm{TTRT}). \tag{48}$$

On the other hand, a *global* scheme utilizes system wide information, including the message periods and lengths on different nodes, the total utilization, the total number of message streams, etc.

As the global allocation schemes use system wide information to allocate synchronous capacities, they can reasonably be expected to result in a better performance than local schemes. Indeed, the two global schemes proposed in this paper achieve better performance than the local ones as shown in Table I. In particular, the normalized proportional scheme has a high worst case achievable utilization of $\frac{1-\alpha}{3}$ which is independent of the number of the nodes in the system or the message lengths and periods.

However, it would be interesting if the 33% worst case achievable utilization is the highest. This raises the issue of the *optimality* of allocation schemes. An optimal allocation scheme should always guarantee a message set if there exists another scheme which can do so. Clearly, the optimal scheme has the highest worst case achievable utilization. Since the global allocation schemes use system wide information, it is likely that an *optimal* allocation scheme will be a global one. Some preliminary studies in the design and implementation of such an optimal synchronous bandwidth allocation scheme have been done.

However, a disadvantage of the global schemes lies in the assumption that the message parameters remain constant. A change in a message stream at a particular node may require a readjustment of synchronous bandwidth over the entire network. This may not be acceptable in some situations. Because local schemes compute the synchronous bandwidth of a node independently of the message parameters at other nodes, they can overcome the above problem. If the parameters of a message stream at a node change during run-time, a local allocation scheme needs to adjust the synchronous bandwidth of only the node involved. Other nodes are not disturbed. That is, the entire network can continue its normal operations while individual nodes change their synchronous bandwidth in response to the changing message parameters. This, of course, assumes that the total utilization factor of the message set remains within the worst case achievable utilization of the allocation scheme.

As the local allocation schemes use less information than the global ones, they may not be expected to achieve a worst case achievable utilization as high as some of the global ones. Both the local allocation schemes examined in this paper (i.e., the full length scheme using only C_i, and the proportional scheme using $\frac{C_i}{P_i}$) turned out to have a worst case achievable utilization of 0%. However, recent studies indicate that despite the fact that local schemes use less information, their performance in terms of the worst case achievable utililzation may be as high as the global schemes.

We are also working on multi-link ring networks where more than one link can connect two neighboring nodes. With this topology, we would like to study the protocol performance in the context of the worst case achievable utilization.

Appendix A
Transformation of Network Model

In this appendix, we present a transformation that converts an arbitrary network model to a logically equivalent virtual model where each node has exactly one synchronous message stream.

Let node i be denoted by N_i. Zero, one, or more synchronous message streams may be arriving at the node from the external world requesting transmission. Let the set of synchronous message streams arriving at N_i be denoted by NS_i. Hence, if node N_i has p streams of synchronous message streams arriving at it, we denote the synchronous message set as

$$NS_i = \{S_{i_1}, S_{i_2}, \cdots, S_{i_p}\}. \tag{49}$$

Similarly, the asynchronous message set at node i is denoted as NA_i. Thus, we can represent node i as:

$$N_i = (NS_i, NA_i, \theta_i) \tag{50}$$

where θ_i is the latency between node i and its upstream neighbor.

Node N_i is considered an *active* node if $NS_i \neq \phi$. That is, there is at least one stream of synchronous messages arriving at node N_i. If $NS_i = \phi$, node N_i is an *inactive* node.

The network can then be represented by the set of nodes as shown below:

$$\text{Network } = \{N_1, N_2, \cdots, N_m\}. \tag{51}$$

In order to simplify our analysis, the above network model needs to be transformed into a simpler *virtual network* model in which each virtual node will have one synchronous message stream arriving at it. The transformation, T, may be represented as follows:

For all nodes N_i $(1 \leq i \leq m)$ in the network, do:

- If N_i is an active node with p streams of synchronous messages, it is transformed into p virtual nodes as follows:

$$T(N_i) = (VN_{i_1}, VN_{i_2}, \cdots, VN_{i_p}), \tag{52}$$

where the virtual node VN_{i_j} is represented as

$$VN_{i_j} = \begin{cases} (\{S_{i_j}\}, \phi, \theta_{i_j}), & \text{if } 1 \leq j < p, \\ (\{S_{i_p}\}, NA_i, \theta_{i_p}), & \text{if } j = p, \end{cases} \tag{53}$$

where

$$\theta_{i_j} = \begin{cases} \theta_i, & \text{if } j = 1, \\ 0, & \text{if } 2 \leq j \leq p. \end{cases} \tag{54}$$

That is, an active node with p streams of synchronous messages is split into p different virtual nodes, each with one of the synchronous message streams available at it. Any asynchronous messages available at the original node should be transmitted only after the synchronous messages have been transmitted. This is because asynchronous messages are the low priority messages. Hence, the asynchronous messages at node i will be considered to be available only at the last virtual node (VN_{i_p}) in the down-link direction of the token traversal. Since the virtual nodes are derived from a single node, the transmission delay (θ_{i_j}) between such nodes is 0. However, the transmission delay between the first virtual node (VN_{i_1}) and its upstream neighbor (which is also a virtual node) is θ_i.

- If N_i is an inactive node with no synchronous messages, it is transformed into a virtual node NV_i as follows:

$$T(N_i) = VN_i = (\{S_d\}, NA_i, \theta_i), \tag{55}$$

where S_d represents a stream of dummy synchronous messages introduced into the virtual node VN_i with message length $C_d = 0$ and period $P_d = \infty$.

After transformation of the network, the virtual nodes are connected in a ring fashion.

Note that the total ring latency of the virtual network will be equal to that of the actual network from which it was derived. It is evident that the virtual network model is logically equivalent to the original network model.

Appendix B
Proof of Lemma 5.1

Lemma 5.1: For any synchronous message stream i $(1 \leq i \leq n)$ we have

$$\frac{\lfloor \frac{P_i}{\text{TTRT}} - 1 \rfloor}{P_i/\text{TTRT}} \geq \frac{1}{3 - \frac{\delta_i}{\text{TTRT}}} \geq \frac{1}{3}. \tag{56}$$

Proof: From (4), we have

$$P_i = m_i \cdot \text{TTRT} - \delta_i, \tag{57}$$

where $m_i = \lceil \frac{P_i}{\text{TTRT}} \rceil$ and $\delta_i = \lceil \frac{P_i}{\text{TTRT}} \rceil \cdot \text{TTRT} - P_i$. Depending on the value of δ_i, we have two cases to consider the following cases.

Case 1: $0 < \delta_i < \mathrm{TTRT}$. This implies $m_i \geq 3$. We have

$$\frac{\lfloor P_i/\mathrm{TTRT} - 1\rfloor}{P_i/\mathrm{TTRT}} = \frac{\lfloor \frac{(m_i \cdot \mathrm{TTRT} - \delta_i)}{\mathrm{TTRT}} - 1\rfloor}{(m_i \cdot \mathrm{TTRT} - \delta_i)/\mathrm{TTRT}} = \frac{\lfloor m_i - \frac{\delta_i}{\mathrm{TTRT}} - 1\rfloor}{m_i - \frac{\delta_i}{\mathrm{TTRT}}} = \frac{m_i - 2}{m_i - \frac{\delta_i}{\mathrm{TTRT}}}. \quad (58)$$

Note that the right-hand side of (58) is an increasing function of m_i. If f represents the right-hand expression in (58), then f is an increasing function of m_i since

$$\frac{df}{dm_i} = \frac{2 - (\delta_i/\mathrm{TTRT})}{(m_i - (\delta_i/\mathrm{TTRT}))^2} > 0 \cdot \quad (\text{since } \frac{\delta_i}{\mathrm{TTRT}} \leq 1 \text{ and } m_i \geq 2). \quad (59)$$

Therefore, the minimum value of (58) is obtained by substituting the minimum value of m_i, i.e., $m_i = 3$. Hence

$$\frac{\lfloor P_i/\mathrm{TTRT} - 1\rfloor}{P_i/\mathrm{TTRT}} = \frac{m_i - 2}{m_i - \frac{\delta_i}{\mathrm{TTRT}}} \geq \frac{1}{3 - \frac{\delta_i}{\mathrm{TTRT}}}. \quad (60)$$

Further, the right-hand side of (60) is an increasing function of δ_i. If we let $\delta_i \to 0^+$, we have

$$\frac{\lfloor P_i/\mathrm{TTRT} - 1\rfloor}{P_i/\mathrm{TTRT}} \geq \frac{1}{3 - \frac{\delta_i}{\mathrm{TTRT}}} \geq \frac{1}{3}. \quad (61)$$

Thus, the lemma holds in this case.

Case 2: $\delta_i = 0$. This implies $P_i = m_i \cdot \mathrm{TTRT}$. Since $m_i \geq 2$,

$$\frac{\lfloor P_i/\mathrm{TTRT} - 1\rfloor}{P_i/\mathrm{TTRT}} = \frac{m_i - 1}{m_i} \geq \frac{1}{2}. \quad (62)$$

From (61) and (62), we have

$$\frac{\lfloor P_i/\mathrm{TTRT} - 1\rfloor}{P_i/\mathrm{TTRT}} \geq \frac{1}{2} \geq \frac{1}{3 - \frac{\delta_i}{\mathrm{TTRT}}} \geq \frac{1}{3}. \quad (63)$$

□

References

[1] G. Agrawal, "Real-time comunication with FDDI networks," Ph.D. Dissertation, Comput. Sci. Dept., Texas A&M Univ., in preparation.

[2] G. Agrawal, B. Chen, W. Zhao and S. Davari, "Guaranteeing synchronous message deadlines with timed token medium access control protocol," Tech. Rep. 92-006, Comput. Sci. Dept., Texas A&M Univ., Apr. 1992.

[3] *FDDI Token ring media access control (MAC).* ANSI Standard X3.139, 1987.

[4] *FDDI Token ring station management*, draft proposed—ANSI Standard X3T9.5 rev. 5, May 10, 1989.

[5] L. N. Bhuyan, D. Ghosal, and Q. Yang, "Approximate analysis of single and multiple ring networks," *IEEE Trans. Comput.*, vol. 38, no. 7, pp. 1027–1040, July 1989.

[6] K. Chae and A. A. Nilsson, "Performance evaluation of FDDI network and interconnected heterogeneous networks," in *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Sept. 30–Oct. 3, 1990, pp. 75–83.

[7] B. Chen, G. Agrawal, and W. Zhao, "Properties of the timed token protocol," Tech. Rep. 92-030, Comput. Sci. Dept., Texas A&M Univ., Oct. 1992.

[8] M. D. Cohn, "A network architecture for advanced aircraft," in *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Oct. 10–12, 1989, pp. 358–364.

[9] S. Davari and S. Dhall, "An on-line algorithm for real-time tasks allocation," in *Proc. IEEE Real-Time Syst. Symp.*, Dec. 1986, pp. 194–200.

[10] S. Davari and W. Zhao, "RMS aids real-time scheduling," *RICIS Rev.*, vol. 3, no. 1, 1991.

[11] D. Dykeman and W. Bux, "Analysis and tuning of the FDDI media access control protocol," *J. Select. Areas Commun.*, vol. 6, no. 6, pp. 997–1010, July 1988.

[12] D. T. Green and D. T. Marlow, "SAFENET—A LAN for navy mission critical systems," in *Proc. 14th Conf. on Local Comput. Net.*, Minneapolis, MN, Oct. 1989, pp. 340–346.

[13] R. M. Grow, "A timed token protocol for local area networks," in *Proc. Electro/82, Token Access Protocols*, May 1982.

[14] E. Hotard, "FDDI concentrator design issues," in *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Sept. 30–Oct. 3, 1990, pp. 388–394.

[15] IEEE/ANSI Standard 802.4—1985 "Token passing bus access method and physical layer specifications," *IEEE* , New York, 1985.

[16] V. Iyer and S. P. Joshi, "New standards for local networks push upper limits for lightwave data," *Data Commun.*, pp. 127–138, July 1984.

[17] ——, "FDDI's 100 M-bps protocol improves on 802.5 Spec's 4-M-bps limit," *Elect. Design News.* pp. 151–160, May 2, 1985.

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

[19] A. P. Jayasumana and P. N. Werahera, "Performance of fibre distributed data interface network for multiple classes of traffic," *IEEE Proc.*, pt. E, vol. 137, no. 5, pp. 401–408, Sept. 1990.

[20] M. J. Johnson, "Reliability mechanisms of the FDDI high bandwidth token ring protocol," *Comput. Net. ISDN Syst.*, vol. II., no. 2, pp. 121–131, 1986.

[21] ——, "Proof that timing requirements of the FDDI token ring protocols are satisfied," *IEEE Trans. Commun.*, vol. COM-35. no. 6, pp. 620–625, June 1987.

[22] R. J. Kochanski and J. L. Paige, "SAFENET—The standard and its application," *IEEE LCS*, vol. 2, no. 1., pp. 46–51, Feb. 1991.

[23] J. F. Kurose, M. Schwartz and T. Yemini, "Controlling window protocols for time-constrained communication in a multiple access environment," in *Proc. IEEE Int. Data Commun. Symp.*, 1983, pp. 41–49.

[24] C. C. Lim, L. Yao, and W. Zhao, "A comparative study of three token ring protocols for real-time communications," *IEEE Conf. Distributed Computing Syst.*, Arlington, Texas, May 1991, pp. 308–317.

[25] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[26] N. Malcolm and W. Zhao, "Version selection schemes for hard real-time communications," in*Proc. IEEE Real-time Syst. Symp.*, San Antonio, TX, Dec. 1991, pp. 12–21.

[27] P. Martini and T. Meuser, "Service integration in FDDI," in *Proc. IEEE Conf. on Local Computer Networks*, Minneapolis MN, Sept. 30–Oct. 3, 1990, pp. 23–32.

[28] J. Mccool, "FDDI—getting to the inside of the ring," *Data Commun.*, vol. 17, pp. 185–192, Mar. 1988.

[29] A. K. Mok and M. L. Dertouzos, "Multiprocessor scheduling in a hard real-time environment," in *Proc. Texas Conf. Computing Syst.*, Nov. 1978.

[30] J. Ng and J. Liu, "Performance of local area network protocols for hard real-time applications," in *IEEE Conf. Distributed Computing Syst.s*, Arlington, TX, May 1991, pp. 318–326.

[31] K. B. Ocheltree and R. M. Montalvo, "FDDI ring management," in *Proc. IEEE Conf. Local Computer Networks*, Minneapolis MN, Oct. 10–12, 1989, pp. 18–23.

[32] K. B. Ocheltree, "Using redundancy in FDDI networks," *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Sept. 30–Oct. 3, 1990, pp. 261–267.

[33] J. L. Paige, "SAFENET—A navy approach to computer networking," in *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Sept. 30–Oct. 3, 1990, pp. 268–273.

[34] J. Pang and F. A. Tobagi, "Throughput analysis of a timer controlled token passing protocol under heavy load," *IEEE Trans. Commun.*, vol. 37, no. 7, pp. 694–702, July 1989.

[35] K. Ramamritham, "Channel characteristic in local area hard real-time system." *Comput. Net. and ISDN Syst.*, vol. 13, no. 1, Jan. 1987.

[36] F. E. Ross, "FDDI—A tutorial," *IEEE Commun. Mag.*, vol. 24, no. 5, pp. 10–17, 1986.

[37] ——, "Rings are round for good," *IEEE Net. Mag.*, Jan. 1987.

[38] ——, "An overview of FDDI: The fibre distributed data interface," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 1043–1051, Sept. 1989.

[39] SAE, Aerospace Systems Division, Committee AS-2, "Linear token-passing multiple data bus," AS4074.1, Version 4.0, Jan. 25, 1988.

[40] R. Sankar and Y. Y. Yang, "Performance analysis of FDDI," in *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Oct. 10–12, 1989, pp. 328–332.

[41] K. C. Sevcik and M. J. Johnson, "Cycle time properties of the FDDI token ring protocol," *IEEE Trans. Software Eng.*, vol. SE-13, no. 3, pp. 376–385, 1987.

[42] L. Sha and J. B. Goodenough, "Real-time scheduling theory and Ada*," *IEEE Comput.*, pp. 53–62, Apr. 1990.

[43] K. G. Shin and C. Hou, "Analytic evaluation of contention protocols used for real-time systems," in *Proc. IEEE Real-Time Syst. Symp.*, Dec. 1990, pp. 136–145.

[44] R. Southard, "Fibre optics: A winning technology for LANs," *Electronics*, vol. 61, pp. 111–114, Feb. 1988.

[45] W. Stallings, "*Computer Communication Standards, Vol 2: Local Area Network Standards*. Indianapolis, IN: Howard W. Sams & Co., 1987.

[46] J. Stankovic, "A perspective on distributed computer systems," *IEEE Trans. Comput.*, vol. C-33, no. 12, Dec. 1984.

[47] H. Stone and S. H. Bokhai, "Control of distributed processes," *IEEE Comput.*, vol. 11, no. 7, pp. 1102–1115, July 1978.

[48] J. K. Strosnider, J. Lehoczky, and L. Sha, "Advanced real-time scheduling using the IEEE 802.5 token ring," in *Proc. IEEE Real-Time Systems Symp.*, Dec 1988, pp. 42–52.

[49] F. A. Tobagi, "Multiaccess protocols and real-time communications in local area networks," presented at the *Advanced Sem. Real-time Local Area Net.*, Bandol, France, Apr. 16–18, 1986.

[50] J. F. Torgerson, "FDDI MAC services design considerations," in *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Oct. 10–12, 1989, pp. 41–48.

[51] R. W. Uhlhorn, "The fibre-optic high-speed data bus for a new generation of military aircraft," *IEEE LCS*, vol. 2, no. 1, pp. 36–45, Feb. 1991.

[52] J. N. Ulm, "A timed token ring local area network and its performance characteristics," in *Proc. Conf. Local Comput. Net.*, Feb. 1982, pp. 50–56.

[53] A. Valenzano, P. Montuschi, and L. Ciminiera, "Some properties of timed token medium access protocols," *IEEE Trans. Software Eng.*, vol. 16, no. 8, pp. 858–869, Aug. 1990.

[54] A. C. Weaver and R. Simoncic, "Communications for the NASA space station," *Proc. IEEE Conf. Local Comput. Net.*, Minneapolis MN, Oct. 10–12, 1989, pp. 333–346.

[55] L. Yao and W. Zhao, "Performance of an extended IEEE 802.5 protocol in hard real-time systems," in *Proc. IEEE Conf. Comput. Commun., INFOCOM '91*, Apr. 1991, pp. 469–478.

[56] S. Yau, Ed., Special Issue on Distributed Computer Systems, in *IEEE Trans. Comput.*, vol. 38, no. 8, Aug. 1989.

[57] W. Zhao and K. Ramamritham, "Virtual time CSMA protocols for hard real-time communications," *IEEE Trans. Software Eng.*, vol. SE-13, no. 8, pp. 938–952, Aug. 1987.

[58] W. Zhao, J. A. Stankovic, and K. Ramamritham, "A multi-access window protocol for time constrained communications," in *Proc. IEEE Conf. Distributed Computing Syst.*, June 1988.

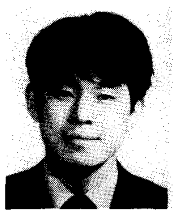
[59] W. Zhao, C. Barter, and N. Malcolm, "Virtual time CSMA protocols with two version message model for real-time communications," in *Proc. IEEE Conf. Net.*, Singapore, July 1989, pp. 295–300.

[60] W. Zhao, J. A. Stankovic, and K. Ramamritham, "A window protocol for transmission of time constrained messages," *IEEE Trans. Comput.*, vol. 39, no. 9, pp. 1186–1203, Sept. 1990.

Gopal Agrawal received his B.E. degree in computer engineering at G.B. Pant University, Uttar Pradesh, India in 1989. He is currently working towards Ph.D. degree in computer science at Texas A&M University, College Station, TX.

His research interests include real-time systems, network communication protocols, distributed systems and database systems. He is currently a student member of the ACM and IEEE computer society.

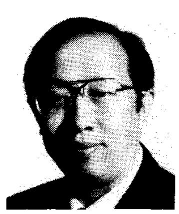
Mr. Agrawal, along with B. Chen, W. Zhao and S. Davari, received the "outstanding paper" award at the IEEE International Conference on Distrubited Computing Systems.



Biao Chen received the S.B. degree in computer science from Fudan University, Shanghai, P.R. China in 1988 and M.S. degree in Mathematics from Texas A&M University, College Station, TX in 1992. He is currently working toward the Ph.D. degree in computer science and mathematics at Texas A&M University, College Station, TX.

His research interests include real-time systems, comptuer networks, distributed systems, and comptuer architecture.

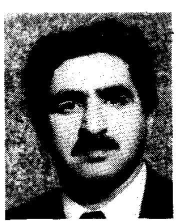
Mr. Chen received the outstanding paper award in the IEEE International Conference on Distributed Computing Systems held in 1992. He is a member of IEEE and its Computer Society and Americal Mathematics Society.



Wei Zhao received the Diploma in physics from Shaanxi Nornal University, Xian, China, in 1977, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts, Amherst, in 1983 and 1986, respectively.

He is currently an Associate Professor in the Department of Computer Science, Texas A& University. His research intersts include scheduling algorithms, communications, protocols, distributed real-time systems, concurrence control in database systems, resource management in operating systems, and their applications. He has published extensively in these areas.

Dr. Zhao received the Best Paper Award in the 12th IEEE International Conference on Distributed Computing Systems (ICDCS) for a paper on hard real-time communications. Currently, he serves as an Editor for the IEEE TRANSACTIONS ON COMPUTERS. He was a Guest Editor for a special issue of ACM Operating System Review on Real-Time Operationg Systems. He served as program/organization committee member for various conferences and workshops. He co-chaired the First IEEE Workshop on Imprecise and Approximate Computation in 1992. He is a Vice Chairman of the 14th IEEE International Conference on Distributed Computing Systems (ICDCS) to be held in 1994.



Sadegh Davari received the M. S. and the Ph.D. degrees in computer science from the University of Oklahoma, 1980 and 1985, respectively.

He is currently an associate Professor of Computer Science at the University of Houston-Clear Lake. His areas of interest include real-time systems, operating systems, and concurrent programming.

Fr. Davari participated in the preparation of the presentation material on Rate Monotonic Analysis (RMA) at the Software Engineering Institute/Carnegia Mellon Unviersity. He has given several tutodials on RMA. He has worked on several real-time projects of NASA, jointly with others from UHCL, TAMU, and SEI/CMU. He is a member of the UHCL Software Engineering Program Committee. He is a member of the ACM and IEEE Computer Society.