Article:

Burns, A. orcid.org/0000-0001-5621-8816 and Zhang, S. (1995) An optimal synchronous bandwidth allocation scheme for guaranteeing synchronous message deadlines with the timed-token MAC protocol. IEEE/ACM Transactions on Networking. pp. 729-741. ISSN 1063-6692

https://doi.org/10.1109/90.477719

An Optimal Synchronous Bandwidth Allocation Scheme for Guaranteeing Synchronous Message Deadlines with the Timed-Token MAC Protocol

Sijing Zhang and Alan Burns, *Senior Member, IEEE*

***Abstract*—This paper investigates the inherent timing properties of the timed-token medium access control (MAC) protocol necessary to guarantee synchronous message deadlines in a timed token ring network such as, fiber distributed data interface (FDDI), where the timed-token MAC protocol is employed. As a result, an exact upper bound, tighter than previously published, on the elapse time between any number of successive token arrivals at a particular node has been derived. Based on the exact protocol timing property, an optimal synchronous bandwidth allocation (SBA) scheme named enhanced MCA (EMCA) for guaranteeing synchronous messages with deadlines equal to periods in length is proposed. This scheme is an enhancement on the previously published MCA scheme.**

***Index Terms*—Real time communications, timed-token medium access control protocol, FDDI networks, synchronous messages, synchronous bandwidth, synchronous bandwidth allocation schemes.**

NOMENCLATURE

C_i	Length (i.e., the maximum transmission time) of a message in stream S_i.
D_i	(Relative) deadline of a message in stream S_i.
$EMCA$	Enhanced MCA, i.e., Enhanced minimum capacity allocation.
H_i	Synchronous bandwidth allocated to node i.
$\vec{H}$	Allocation vector, i.e., $\vec{H} = (H_1, H_2, \cdots, H_n)$.
MCA	Minimum capacity allocation. So, the MCA scheme means the minimum capacity allocation scheme.
n	Number of nodes on the token ring network.
P_i	Period length (i.e., the minimum message inter-arrival time) of synchronous messages in stream S_i.
$P_{\min}$	Tightest lower bound on message periods, i.e., the minimum of all P_i $(1 \leq i \leq n)$.
SBA	Synchronous bandwidth allocation.
S_i	Stream of synchronous messages at node i.
$TTRT$	Target token rotation time.
$t_{l,i}$	Time when the token makes its l^{th} arrival at node i.
$U(M)$	Utilisation factor of the synchronous message set M, i.e., fraction of time spent by the network in transmission of the synchronous messages.
$WCAU$	Worst Case Achievable Utilisation.
X_i	Minimum amount of time available for node i to transmit its synchronous messages within its message period P_i.
τ	Portion of $TTRT$ unavailable for transmitting messages.
α	Ratio of τ to the target token rotation time ($TTRT$), i.e., $\alpha = \tau/TTRT$.

Manuscript received October 11, 1994; revised May 22, 1995; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor B. Mukherjee.

The authors are with the Real-Time Systems Research Group, Department of Computer Science, University of York, Heslington, York YO1 5DD, UK (email: zsj@minster.york.ac.uk, burns@minster.york.ac.uk).

IEEE Log Number 9415892.

I. INTRODUCTION

IN A DISTRIBUTED system for hard real time applications, communication through message exchange between tasks residing on different nodes must happen in bounded time, in order to insure that end-to-end deadline requirements are met. This motivates the use of medium access control (MAC) communication protocols that provide a guaranteed connection and a guaranteed amount of channel bandwidth to support timely delivery of inter-task messages. With the important property of bounded time between any two consecutive visits of the token to a node, the timed token protocol becomes one of the most suitable and attractive candidates for hard real time applications. This protocol has been incorporated into many network standards including the fiber distributed data interface (FDDI), IEEE 802.4, the high speed data bus and the high speed ring bus (HSDB/HSRB), and the survivable adaptable fiber optic enbedded networks (SAFENET), used as backbone networks in many embedded real time applications [2].

FDDI uses the timed token protocol proposed by Grow [5]. With this protocol, messages are distinguished into two types: *synchronous* messages and *asynchronous* messages. Synchronous messages, such as sampled/digitised voice and video data, can be viewed as periodic messages that arrive at regular intervals and have delivery time constraints. Asynchronous messages are nonperiodic and may arrive in a random way and have no time constraints. At network initialization time, all nodes negotiate a common value for the target token rotation time ($TTRT$) since each node has different synchronous transmission requirements to be satisfied. The negotiated value for $TTRT$ should be chosen small enough to satisfy the most stringent response time requirements of all nodes. Each node is assigned a fraction of the $TTRT$, known as its *synchronous bandwidth*, which is the maximum time the node is allowed to transmit its synchronous messages each time it receives the token [2]. Whenever a node receives the token, it transmits its synchronous messages, if any, for a time no more than its allocated synchronous bandwidth. After synchronous message transmission, asynchronous messages can be sent (if there are any), but only if the time elapsed since the previous token arrival at the same node is less than $TTRT$, i.e., only if the token has arrived at the node earlier than expected. That is, synchronous traffic is assigned a guaranteed bandwidth, while the leftover bandwidth (unallocated, unused or both) is dynamically shared among all the nodes for asynchronous traffic [6].

The timed token protocol guarantees, to each node, an average bandwidth and a bounded access delay for synchronous traffic. However, this guarantee alone, although necessary, is insufficient for the timely delivery of deadline constraint messages. For guaranteeing the synchronous message deadlines with the timed token protocol, the protocol parameters ($TTRT$ and the synchronous bandwidths) have to be properly selected. A large amount of work on the selection of these parameters has been reported in the literature, with the focus on synchronous bandwidth allocation (SBA) [1], [4], [6], [9]–[11], [17]. Hamdaoui and Ramanathan [6] address the problem of setting both $TTRT$ and the synchronous bandwidth of each node so as to guarantee sets of periodic message streams. Similar work was conducted by Lim *et al.* [9] who studied the deadline guarantee of time dependent multimedia data in an FDDI network. In [2] four SBA schemes are analysed by Agrawal *et al.*, and a metric called the *worst case achievable utilization (WCAU)* is adopted as a means to compare and evaluate different schemes. The WCAU of a SBA scheme is defined as the largest utilization U such that the scheme can always guarantee a synchronous message set as long as the utilization (factor) of the message set is no more than U. Their analysis shows that the WCAU of the normalized proportional allocation scheme is 33%, the highest of the four schemes analyzed. Agrawal *et al.* [1] also developed and analyzed a local SBA scheme for guaranteeing synchronous message sets with message periods equal to deadlines. They showed that their scheme can also achieve a WCAU of 33%. Malcolm *et al.* [10] generalized the local scheme proposed by Agrawal *et al.*, and as a result, they proposed a local SBA scheme for use in a general message set where each message can have an arbitrary deadline. Another similar local SBA scheme for guaranteeing synchronous messages with arbitrary deadlines is developed by Zheng *et al.* [17]. The minimum capacity allocation (MCA) scheme, that was claimed to be optimal for guaranteeing synchronous message sets with message periods equal to deadlines, was proposed by Chen *et al.* [4].

Unfortunately, the MCA scheme is not optimal due to its failure to guarantee some schedulable synchronous message sets (with message periods equal to deadlines). The non-optimality of the MCA scheme originates from the fact that the upper bound derived by Chen *et al.* [3] on the elapse time between any number of successive token arrivals to a node is not exact and may not be tight when the number of successive token arrivals becomes large. In this paper we will develop and analyze an enhanced version of the MCA scheme, named EMCA, based on a more exact and tighter upper bound. The proposed EMCA scheme is optimal in the sense that any synchronous message set (with periods equal to deadlines) that can be guaranteed by any SBA scheme, can be guaranteed by EMCA. Our EMCA scheme also differs significantly from the MCA scheme by explicitly taking into account the synchronous bandwidth allocation for the message sets with the minimum message periods ($P_{\min}$) less than $2 \cdot TTRT$, and consequently can apply to any synchronous message set (with $P_{\min} > TTRT$).

Because the paper reports an enhanced version of the MCA scheme, for easy comparison we shall retain and use/quote most of the notations adopted by Chen *et al.* [4] in their development and analysis of the MCA scheme, and adopt the same framework as used by them. The remainder of this paper is organized as follows: In Sections II and III the framework under which this study has been conducted is presented. Specifically, we describe the network and message models in Section II and the synchronous bandwidth allocation (schemes) in Section III. We then address the timing properties of the timed token protocol in Section IV. An optimal SBA scheme named EMCA is developed and analyzed in Section V, and its superiority to any other previously published SBA schemes is shown by examples in Section VI. Finally, we conclude the paper in Section VII.

II. The Network and Message Models

A. Network Model

The network is assumed to consist of n nodes arranged to form a ring and be free from any hardware and software failures. Message transmission is controlled by the timed–token protocol. Due to inevitable overheads involved, such as ring latency and other protocol/network dependent overheads, the total bandwidth available for message transmission during one complete traversal of the token around the ring is less than $TTRT$. Let τ be the portion of $TTRT$ unavailable for transmitting messages. The ratio of τ to $TTRT$ is denoted by α. So the usable ring utilization available for message transmission, synchronous and asynchronous, would be $(1 - \alpha)$.

B. Message Model

It is assumed that there is only one stream of synchronous messages on each node.[1] That is, a total of the n synchronous message streams, denoted as $S_1, S_2, \cdots, S_n$ with S_i corresponding to node i, forms a synchronous message set, M, i.e., $M = \{S_1, S_2, \cdots, S_n\}$. Messages from a synchronous stream are assumed to have the same inter-arrival period and the same relative deadline. The period of a synchronous message stream can be thought as the minimum message inter-arrival time. The relative deadline is the maximum amount of time that may elapse between a message arrival and the completion of its transmission [10]. Let P_i be the period and D_i be the

[1] This assumption of one stream per node does not lose generality since Agrawal *et al.* [2] have shown how a token ring network with more synchronous message streams per node can be transformed into a logically equivalent network with one synchronous message stream per node.

relative deadline. That is, if a message from stream S_i arrives at time t, then its absolute deadline is at time $t + D_i$. The term *relative* will be omitted in the remainder of this paper when the context is clear. The length of each message from stream S_i, defined as the maximum amount of time needed to transmit this message, is C_i. Thus, each synchronous message stream S_i is characterized as $S_i = (C_i, P_i, D_i)$. Asynchronous messages, that are nonperiodic, do not have a hard real time deadline requirement. For the remainder of this paper (unless stated otherwise) we assume $D_i = P_i$ and therefore $S_i = (C_i, P_i)$.

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

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

III. Synchronous Bandwidth Allocation Schemes

In FDDI, the SMT (station management) standard has not specified a precise algorithm (scheme) for allocation of synchronous bandwidth [7]. It only defines facilities (parameters and frames) that can be used to support a variety of algorithms (schemes). Due to this fact, a large amount of work has been undertaken on effective allocation of the synchronous bandwidth. We use the generally adopted notion of *synchronous bandwidth allocation (SBA) scheme*. An SBA scheme can be defined as an algorithm that produces the values of the synchronous bandwidth H_i to be allocated to node i in the network given the required information for the scheme [2].

1) Classification: SBA schemes can be divided into two classes [1]: *global SBA schemes* and *local SBA schemes*. A global SBA scheme can use both global and local information in allocating synchronous bandwidth to a node. A local SBA scheme, in contrast, uses only information available locally to node i, that includes the parameters of stream S_i (i.e., C_i, P_i, and D_i), $TTRT$ and τ. Let $\vec{H} = (H_1, H_2, \cdots, H_n)$ be an allocation (vector) produced by an SBA scheme, and functions f_L and f_G be respectively a local SBA scheme and a global SBA scheme. Then, a local SBA scheme can be represented as

$$H_i = f_L(C_i, P_i, D_i, TTRT, \tau)(i = 1, 2, \cdots, n)$$

and a global SBA scheme can be represented as

$$\begin{aligned}\vec{H} &= (H_1, H_2, \cdots, H_h)\\ &= f_G(C_1, C_2, \cdots, C_n, P_1, P_2, \cdots,\\ &\quad P_n, D_1, D_2, \cdots, D_n, TTRT, \tau).\end{aligned}$$

A local scheme is usually simple, flexible, and suitable for use in dynamic environments, but it may present a weak guarantee ability due to using only locally available information. In contrast, although a global scheme might be complex and might not be well suited to a dynamic environment, it may present a strong guarantee ability and may perform better than a local one due to it using system-wide information. In this paper we study global SBA schemes.

2) Requirements: In order to guarantee message deadlines, synchronous bandwidths must be properly allocated to individual nodes such that the following two constraints are met [4]:

- *Protocol constraint*: The sum total of the synchronous bandwidths allocated to all nodes in the ring should not be greater than the available portion of the $TTRT$, i.e.,

$$\sum_{i=1}^{n} H_i \leq TTRT - \tau. \tag{2}$$

- *Deadline constraint*: Every synchronous message must be transmitted before its deadline. Let X_i be the minimum amount of time available for node i to transmit its synchronous messages during period P_i, i.e., in a time interval $(t, t + P_i)$, then for a message set with deadlines equal to periods, the deadline constraint implies that

$$X_i \geq C_i. \tag{3}$$

Note that X_i is a function of the number of token visits to node i and H_i. A synchronous message set can be guaranteed by an SBA scheme if an allocation $\vec{H}$, that satisfies both the protocol and the deadline constraints, can be produced by the scheme [4], [2]. We say an allocation $\vec{H}$ is feasible if it satisfies both the protocol and the deadline constraints. A synchronous message set is said to be schedulable if there exists at least one feasible allocation for the message set.

IV. Protocol Timing Properties

In this section, we present some results on the timing properties of the timed token protocol necessary for guaranteeing synchronous message transmission and necessary for us to develop an optimal SBA scheme. In particular, the following theorems and corollaries are of interest. Let $t_{l,i} (l = 1, 2, \cdots)$ be the time the token makes its lth arrival at node i.

Theorem 1: (Johnson and Sevcik's Theorem [8], [13]):

For any integer $l > 0$ and any node i ($1 \leq i \leq n$), under the protocol constraint (2)

$$t_{l+1,i} - t_{l,i} \leq TTRT + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau \leq 2 \cdot TTRT.$$

This theorem shows that the maximum time that could possibly elapse between any two successive token arrivals to a node is bounded by $2 \cdot TTRT$. The result given by Johnson and Sevcik can be used to obtain a lower bound on the minimum number of token visits to a node within the period of its synchronous message stream. Unfortunately, the bound is not tight when the period is longer than $3 \cdot TTRT$ [2]. Chen and Zhao [3] first extended this result, in particular, they generalized the analysis to give an upper bound on the time elapsed between any v (where v is an integer no less than two) consecutive token's arrivals at a particular node. Their generalized theorem is restated as follows:

Theorem 2: (Generalized Johnson and Sevcik's Theorem by Chen and Zhao [3]):

For any integer $l \geq 1, v \geq 2$ and any node i $(1 \leq i \leq n)$, and under the protocol constraint (2)

$$t_{l+v-1,i} - t_{l,i} \leq (v-1) \cdot TTRT + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau.$$

Theorem 2 gives an upper bound on the time possibly elapsed between any v (where v is an integer no less than two) consecutive token arrivals at a particular node. This generalized upper bound has been extensively used by many researchers [1], [4], [6], [9]–[11], [17] in their studies (analyses) of synchronous bandwidth allocation schemes However, as will be seen, the generalized upper bound may not be tight when $v \geq n+2$.

Although extensive research has been done on the timing behavior of the timed–token protocol, the results reported so far are not satisfactory enough for an *optimal* scheme to be proposed. An optimal allocation scheme should be established upon the exact timing properties of the timed token protocol. In order to develop an optimal SBA scheme, the exact timing properties of the protocol need exploring. We also investigated the inherent timing properties of the timed token protocol, and as a result, derived a new generalized version of Johnson and Sevcik's theorem (shown below), that is better than that given by Chen and Zhao [3] in the sense that our generalized upper bound is more exact and tighter.

Theorem 3: (Generalized Johnson and Sevcik's Theorem by Zhang and Burns [15]):

For any integer $l \geq 1, v \geq 2$ and any node$i$ $(1 \leq i \leq n)$, under the protocol constraint (2)

$$t_{l+v-1,i} - t_{l,i} \leq (v-1) \cdot TTRT + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau - \left\lfloor \frac{v-1}{n+1} \right\rfloor \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right).$$

Refer to [15] for a proof of above theorem. By comparing Theorem 3 and Theorem 2, we see that the upper bound derived by Chen and Zhao is tight only when either v is less than $n+2$ or the condition of $\sum_{h=1}^{n} H_h = TTRT - \tau$ holds. However, when allocating synchronous bandwidths for a given synchronous message set, full-allocation is not always best and may even result in no feasible allocations [14][2] That is, for some synchronous message sets to be guaranteed, synchronous bandwidths have to be allocated such that $\sum_{i=1}^{n} H_i < TTRT - \tau$. An example (given in Table III) in Section VI illustrates this.

As shown in its proof process [15], Theorem 3 gives an upper bound on the maximum time possibly elapsed in the worst case before node i gains permission for using the $(v-1)$th of the next $(v-1)$ turns of its allocated synchronous bandwidth (H_i). It is therefore clear that the time possibly elapsed in the worst case before node i uses up its next $(v-1)$ allocated synchronous bandwidths is bounded by the above upper bound (given in Theorem 3) plus H_i, i.e.

$$(v-1) \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor \frac{v-1}{n+1} \right\rfloor \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right).$$

Note that the above upper bound is independent of any particular node. Realizing this and considering elapse time before node i uses up its next v allocated synchronous bandwidths (H_i's) (for simplicity of presentation), we get, with Theorem 3, the following corollary:

Corollary 1: Let $I(v)$ be the tight upper bound on the (maximum) time that could possibly elapse in the worst case before any node uses up its next v (where v is a positive integer) allocated synchronous bandwidths (H_i's), then, under the protocol constraint (2)

$$I(v) = v \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor \frac{v}{n+1} \right\rfloor \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right).$$

The exact results on timing properties given in Theorem 3 and Corollary 1 are very important and can be used in the derivation of the exact lower bound on the time available for a node to transmit its synchronous messages within a given time period, necessary for us to develop an optimal SBA scheme.

Now we derive the exact expression of X_i $(1 \leq i \leq n)$, the minimum amount of time available for node i to transmit its synchronous messages during its message period P_i, given an allocation $\vec{H}$ (no matter from which scheme this allocation is produced) that satisfies the protocol constraint (2). Assume that at time t, a synchronous message with period P_i (where $P_i > TTRT$) arrives at node i.[3] Then, by Corollary 1, we have the following steps to follow, to derive X_i:

1) Choose an integer $m_i (m_i \geq 1)$ such that $I(m_i - 1) \leq P_i < I(m_i)$. Assume that $I(0) = 0$ if $m_i = 1$.
2) We know, by Corollary 1, that during the first $I(m_i - 1)$ time interval of P_i, i.e., in the time interval of $(t, t + I(m_i - 1)]$, node i can use H_i at least $(m_i - 1)$ times. Thus, $X_i \geq (m_i - 1) \cdot H_i$.
3) In the worst case, node i can get the chance of using part of H_i during the remaining time interval, i.e., $(t + I(m_i - 1), t + P_i]$, if any, only when $I(m_i) - H_i < P_i < I(m_i)$. Therefore, the minimum amount of time available for node i to do synchronous transmission

[2] By full-allocation we mean in this paper, that all the usable network bandwidth is exhaustively allocated among all synchronous nodes only, i.e., the synchronous bandwidths are allocated such that $\sum_{i=1}^{n} H_i = TTRT - \tau$.

[3] It is necessary to confine each P_i such that $P_i > TTRT$ for any synchronous message set to be guaranteed because we see by Corollary 1 (when $v = 1$) that if $P_i \leq TTRT$, node i cannot get the chance of using its allocated synchronous bandwidth H_i even once in the worst case during P_i.

during the remaining period can then be obtained by the calculation of $\max(P_i - [I(m_i) - H_i], 0)$, in particular

$$\max\left(P_i - \left\{m_i \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor\frac{m_i}{n+1}\right\rfloor \cdot\left(TTRT - \sum_{h=1}^{n} H_h - \tau\right) - H_i\right\}, 0\right).$$

4) Including the result of $\max(\ldots)$ from 3), in the X_i expression from 2), we get the (total) minimum available time (X_i) for node i to send its synchronous messages during P_i.

$$X_i(\vec{H}) = (m_i - 1) \cdot \max\left[P_i\left\{m_i \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor\frac{m_i}{n+1}\right\rfloor \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau) - H_i\right\}, 0\right].$$

It is clear from the above steps that the key problem concerned here is to find the integer m_i (for the synchronous message stream S_i) confined by 1 above. Then the minimum available time (X_i) can be in turn determined by the $X_i(\vec{H})$ expression in 4. The following theorem determines the possible value range of the integer $m_i (1 \leq i \leq n)$ for a given synchronous message set.

Theorem 4: For any given allocation $\vec{H} = (H_1, H_2, \cdots, H_h)$ that meets the protocol constraint (2), the positive integer m_i $(i = 1, 2, \cdots, n)$ that satisfies the inequality of $I(m_i - 1) \leq P_i < I(m_i)$ (where $P_i > TTRT$) must be either

$$m_i = \left\lceil \frac{(n+1)\cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil$$

or

$$m_i = \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor$$

and

$$0 \leq \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor - \left\lceil \frac{(n+1)\cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil \leq 1.$$

Refer to Appendix A for a proof of the above theorem. With Theorem 4, X_i can be formally determined by the following theorem:

Theorem 5: Assume that at time t, a synchronous message with period P_i ($P_i > TTRT$) arrives at node i $(1 \leq i \leq n)$. Then, in time interval $(t, t + P_i]$ and under the protocol constraint (2), the minimum amount of time (X_i) available for node i to transmit synchronous messages is given by

$$X_i(\vec{H}) = (m_i - 1)\cdot H_i + \max\left[P_i - \left\{m_i \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor\frac{m_i}{n+1}\right\rfloor \cdot\left(TTRT - \sum_{h=1}^{n} H_h - \tau\right) - H_i\right\}, 0\right]$$

where m_i is an integer ($m_i \geq 1$) that satisfies the inequality of $I(m_i - 1) \leq P_i < I(m_i)$, and must be either m or $m - 1$, where

$$m = \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor.$$

Proof: This theorem follows from Theorem 4 and the $X_i(\vec{H})$ expression in the above step 4 (as well as the analysis earlier).

Chen *et al.* [4] have also derived an X_i expression, as shown below

$$X_i(\vec{H}) = (q_i - 1)\cdot H_i + \max\left(0, \min\left[r_i - \left(\sum_{h=1,\cdots,n h\neq i} H_h + \tau\right), H_i\right]\right) \quad (4)$$

where $q_i = \left\lfloor \frac{P_i}{TTRT} \right\rfloor$ and $r_i = P_i - q_i \cdot TTRT$. Comparing (4) with that given in Theorem 5, it is clear that our new X_i expression is better in the sense that for any particular allocation and any given length of the message period, more available time for transmitting synchronous messages may be obtained, increasing the possibility of satisfying the deadline constraint (3). Theorem 5 is necessary for testing the deadline constraint, shown again, as follows:

$$X_i(\vec{H}) \geq C_i (\text{where } X_i(\vec{H}) \text{is determined by Theorem 5}). \quad (5)$$

Testing the deadline constraint (3) by using our exact X_i expression, as shown in (5), may now make an allocation deemed to be infeasible under (4) become feasible for the message set considered. The following example illustrates this.

Example: Considering the following simple synchronous message set with $P_i = D_i (i = 1, 2)$:

$$\text{Stream1}: C_1 = 36\ P_1 = 300$$
$$\text{Stream2}: C_2 = 24\ P_2 = 300.$$

For simplicity, we suppose that $TTRT = 50$ and $\tau = 0$. By applying the proportional allocation (PA) scheme (see Section VI for the definition of this scheme) the allocation $\vec{H} = (H_1, H_2) = (6, 4)$ is produced. This allocation $\vec{H}$ is feasible since it clearly satisfies the protocol constraint, and also meets the deadline constraint when judged by using our exact X_i expression (given in Theorem 5), that is, the given message set can be guaranteed by the PA scheme. But, the above allocation $\vec{H}$ might be wrongly supposed to be infeasible because it fails in meeting the deadline constraints (3) when X_i is calculated by (4). The rationale behind this is; when judged by the upper bound derived by Chen and Zhao [3] (see Theorem 2), each node may receive the token and then use its allocated synchronous bandwidth only five times in the worst case during its message period. Hence, the deadline constraint apparently cannot be satisfied for either of these two synchronous message streams. However, when judged by the new tighter upper bound (see Theorem 3 and Corollary 1), the token can visit each node at least seven times and at least seven times its allocated synchronous bandwidth can be used for transmitting synchronous messages during its message period, even in the worst case. Therefore, the deadline constraints are met by the same allocation $\vec{H} = (H_1, H_2) = (6, 4)$.

A. Relaxing the Restriction of $P_{\min} \leq 2 \cdot TTRT$

Due to the restriction of $P_{\min} \geq 2 \cdot TTRT$ with the MCA scheme, any synchronous message set with $P_{\min} < 2 \cdot TTRT$ is restrained from being considered, and, as a result, cannot be guaranteed by the MCA scheme although it may actually be schedulable (e.g., message set E listed in Table V in Section VI). In order to develop an optimal SBA scheme, we derive below, a new restriction necessary for satisfying the deadline constraint of a synchronous message stream with its period greater than $TTRT$ (no matter whether or not the message period is less than $2 \cdot TTRT$).

For node i with $P_i < 2 \cdot TTRT$, we see, from Corollary 1, that the node i may get the chance of using its allocated synchronous bandwidth H_i at most once during P_i in the worst case. So, in order to meet the deadline constraint of the stream S_i, P_i should be long enough to insure that node i can get the chance of using H_i once after receiving the token. Since allocating H_i more than C_i makes no sense for satisfying the deadline constraint (5) but, on the contrary, may cause the protocol constraint (2) to be violated, we assume in the following discussion that the synchronous bandwidths are allocated such that $H_i \leq C_i$ $(1 \leq i \leq n)$. From Theorem 3 (when $v = 2$) and Corollary 1 (when $v = 1$), we know that under the protocol constraint (2), the longest duration for which node i may suffer from waiting for the token in the worst case is $TTRT + \sum_{h=1,\cdots,n,h\neq i} H_h + \tau$, that is, node i may have to wait for this time in the worst case before regaining the token to start its synchronous transmission. In order to meet message deadlines, a synchronous message must be transmitted by the end of its period. This requires that during P_i, the token must visit node i at least once, and at least one H_i should be used by node i. This means (by Corollary 1) that the following inequality must hold:

$$P_i \geq I(l) = TTRT + \sum_{h=1}^{n} H_h + \tau \qquad (i = 1, 2, \cdots, n).$$

This implies that

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

where $P_{\min}$ represents the minimum of all $P_i (i = 1, 2, \cdots, n)$ and $\sum_{i=1}^{n} H_i \leq TTRT - \tau$. Inequality (6) should always hold for any feasible allocation $\vec{H}$ (no matter from which SBA scheme the $\vec{H}$ is produced), under the protocol constraint (2) and the assumption of $H_i \leq C_i$ $(1 \leq i \leq n)$. A violation of (6) under the protocol constraint (2) means that the produced allocation $\vec{H}$ at least cannot meet the deadline constraint of the synchronous message stream with its period matching $P_{\min}$ and, in turn, fails in guaranteeing the message set considered. In fact, whenever an allocation $\vec{H}$ cannot satisfy (6), it cannot satisfy the deadline constraint (5), either. This can be easily shown as follows: Assume $P_i = P_{\min}$ (where $TTRT < P_i < TTRT + \sum_{h=1}^{n} H_h + \tau$) that violates (6). It is easy to check, by Theorem 5, that the only possible value of m_i is one and that the deadline constraint of stream S_i cannot be met when $m_i = 1$.

Note that both (6) and the protocol constraint (2) are necessary for an allocation to become feasible. Combining (6) and (2) into one, we have,

$$\sum_{i=1}^{n} H_i \leq \min\left(P_{\min} - TTRT - \tau, TTRT - \tau\right). \tag{7}$$

With the analysis above, we see that the violation of (7) means that the given allocation $\vec{H}$ fails in satisfying either the protocol constraint (2), or the deadline constraint (5) (when (6) is violated under (2)). It should be noticed that (6) is a weaker restriction (for the synchronous message set to be considered) compared with that (i.e., $P_{\min} \geq 2 \cdot TTRT$) used in the MCA scheme, and allows the schedulability of message sets with $P_{\min} < 2 \cdot TTRT$ to be considered.

V. EMCA (Enhanced MCA) Scheme

In this section, we develop an optimal SBA scheme, named EMCA, that is an enhanced version of the previously published MCA scheme [4].

A good SBA scheme will allocate the smallest possible value of H_i (commensurate with the deadline constraint being satisfied). A smaller value of H_i has two advantages [12]: First, it improves the response time for asynchronous messages, and second, it gives a better chance of satisfying the protocol constraint. Chen *et al.* [4] proposed a global SBA scheme named the minimum capacity allocation (MCA), claimed to be optimal for guaranteeing synchronous message which was set with message deadlines equal to periods. The scheme is so named because Chen *et al.* claimed that their MCA scheme always allocates the minimum required synchronous capacities to the nodes.[4] However, this is not the case. In fact, the MCA scheme cannot always keep allocating the minimum required synchronous bandwidths to

[4] The term *synchronous capacity* used by Chen *et al.* [4] means synchronous bandwidth.

nodes for every synchronous message set considered (although the message set is schedulable) and it is therefore not optimal, either. An allocation is optimal if it can always guarantee a message set whenever there exists an allocation scheme that can do so [4]. In order to develop an optimal SBA scheme, one needs to explore exact timing properties of the protocol. Chen *et al.* [3], [4] made a detailed study on the protocol timing properties. Unfortunately, the results they obtained, based on which their MCA scheme was developed (though important), are not precise enough for an optimal SBA scheme to be proposed. Specifically, the upper bound (see Theorem 2) they derived may not be tight and, consequently, their X_i expression (used to calculate the minimum amount of time available for node i to do synchronous transmission during P_i) is not exact. The new exact results, presented earlier, enables an optimal SBA scheme to be developed.

A. *EMCA—The Enhanced MCA Scheme*

The basic framework for constructing the EMCA scheme is similar to that used by Chen *et al.* to construct the MCA scheme. Both aim at finding an optimal allocation $\vec{H}$ that satisfies both the protocol constraint (2) and deadline constraint (3). So, we can construct the EMCA scheme in a similar way to the MCA scheme, i.e., similar steps/methods to determine whether or not the EMCA scheme can provide a feasible allocation for a given synchronous message set, and how to find such a feasible allocation for a schedulable message set. In our EMCA scheme, however, we adopt a more exact X_i expression (given in Theorem 5) for testing the deadline constraint (3), as shown in (5).

A message set is schedulable if there exists at least one solution $\vec{H}$ that satisfies both the protocol constraint (2) and the deadline constraint (5) [4]. For a given message set, there may be more than one solution for (2) and (5). But, an optimal SBA scheme can always find a solution whenever it exists. Hence, the optimal allocation problem is equivalent to solving the system of inequalities (2) and (5). Since the minimum allocation (vector) $\vec{H}$ that satisfies the deadline constraint (5) maximize the possibility of meeting the protocol constraint (2) as well, we construct the EMCA scheme by searching the minimal $\vec{H}$ vector (if any) which satisfies both (2) and (5). Specifically, a procedure named Min_H is designed to calculate the minimal solution for the system of inequalities (2) and (5). Theorem 6 below is useful for constructing the procedure Min_H.

Theorem 6: For any schedulable synchronous message set, there must exist at least one feasible allocation $\vec{H} = (H_1, H_2, \cdots, H_h)$ in which each H_i $(i = 1, 2, \cdots, n)$ is bounded by

$$\frac{C_i}{\left\lfloor \frac{P_i\cdot(n+1)}{n\cdot TTRT} \right\rfloor + 1} \le H_i \le \frac{C_i}{\max\left[\left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1, 1\right]}.$$

Refer to Appendix B for a proof of above theorem. Let $\vec{m} = (m_1, m_2, \cdots, m_n)$ and $\vec{X} = (X_1, X_2, \cdots, X_n)$. Then, by Theorem 5, the procedure Find_X (which is called by the procedure Min_H) can be designed to calculate X_i $(i = 1, 2, \cdots, n)$ (as well as $\vec{m}$) for a given synchronous message set, given an allocation $\vec{H}$.

Procedure Find_X:

Line 1 begin;
Line 2 for $i = 1, 2, \cdots, n$;
Line 3 begin;
Line 4 $m_i := \left\lfloor \frac{P_i\cdot(n+1)+n\cdot\left(TTRT-\sum_{h=1}^{n} H_h-\tau\right)}{n\cdot TTRT+\sum_{h=1}^{n} H_h+\tau} \right\rfloor$;
Line 5 calculate $I\ (m_i - 1)$ as defined in Corollary 1;
Line 6 if $I(m_i - 1) > P_i$;
Line 7 $m_i := m_i - 1$;
Line 8 calculate X_i as defined in Theorem 5;
Line 9 end;
Line 10 return $(\vec{X}, \vec{m})$;
Line 11 end.

Procedure Min_H:

Line 1 begin;
Line 2 for $i = 1, 2, \cdots, n$;
Line 3 $H_i = \frac{C_i}{\left\lfloor \frac{P_i\cdot(n+1)}{n\cdot TTRT} \right\rfloor + 1}$;
Line 4 repeat;
Line 5 if $\sum_{i=1}^{n} H_i > \min(P_{\min} - TTRT - \tau, TTRT - \tau)$;
Line 6 return (fail, nil);
Line 7 call procedure Find_X to calculate $\vec{X}$ with return $(\vec{X}, \vec{m})$;
Line 8 for $i = 1, 2, \cdots, n$;
Line 9 begin;
Line 10 $\Delta_i := C_i - X_i$;
Line 11 if $\Delta_i > 0$;
Line 12 $H_i = H_i + \frac{\Delta_i}{m_i - 1}$;
Line 13 end;
Line 14 until none of Δ_i's are larger than zero;
Line 15 return (success, $\vec{H}$);
Line 16 end.

Now we state the rationale behind the procedure Min_H. From Theorem 6 we know that for a synchronous message set to be guaranteed, the synchronous bandwidth H_i allocated to node i $(i = 1, 2, \cdots, n)$ should be no less than $C_i / \left(\left\lfloor \frac{P_i\cdot(n+1)}{n\cdot TTRT} \right\rfloor + 1\right)$. So the procedure begins with all the H_i being initialized to this lower bound. The procedure then refines $\vec{H}$ iteratively.

From the analysis in Section IV we know that any feasible allocation $\vec{H}$ must satisfy (7), i.e., $\sum_{i=1}^{n} H_i \le \min(P_{\min} - TTRT - \tau, TTRT - \tau)$. The violation of (7) means that the given allocation $\vec{H}$ fails in meeting either the protocol constraint or the deadline constraint. Thus, at the beginning of each iteration, we first check if (7) is met (see Line 5). Because each H_i is initialized to a lower bound and then keeps either unchanged or increased in each subsequent iteration, an allocation $\vec{H}$ violating (7) in some iteration means that the allocation $\vec{H}$ refined in any subsequent iteration will definitely violate (7). Therefore, once the violation of (7) is found in some iteration, the procedure stops calculating process immediately and returns with a failure status.

In each iteration, the procedure Find_X is then called to calculate $\vec{X}$ as well as $\vec{m}$ if the refined allocation $\vec{H}$ satisfies

(7). With the returned values of both the vectors of $\vec{X}$ and $\vec{m}$, the deficiency (Δ_i), i.e., the difference between the minimum available transmission time (X_i) and the message length (C_i), is then calculated for each node. All the H_i's with a positive deficiency (i.e., $\Delta_i > 0$) need to be refined by a proper amount no more than the deficiency. Note that m_i is a decreasing function of $\sum_{h=1}^{n} H_h$ (see Theorem 5), that is, as the sum total of all the allocated synchronous bandwidths keeps increasing from one iteration to another, m_i may become smaller and smaller. However, no matter how small the m_i could be, we know by Theorem 5 that it cannot be smaller than one. Therefore, all the m_i's tend to no change as the number of iterations increase, and eventually, after a certain number of iterations all the m_is will remain unchanged.

As the number of iterations increases, m_i may decrease as a result of the increased sum total of all synchronous bandwidths allocated. So we estimate in each iteration the increment amount ΔH_i such that it is the minimum required amount for meeting the message deadline of stream S_i if all the m_is are supposed unchanged for the refined allocation $\vec{H}$, in order to make the finally produced allocation $\vec{H}$ as small as possible to maximize the possibility of satisfying the protocol constraint. One can easily check, under the assumption that all the m_i's keep unchanged, that in order to meet the deadline constraint, each insufficient H_i (with $\Delta_i > 0$) should be incremented by at least $\frac{\Delta_i}{m_i - 1}$. Hence, we choose the increment of $\frac{\Delta_i}{m_i - 1}$ to refine every insufficient H_i in the procedure Min_H. In fact, the m_i may reduce due to the increase of the total synchronous bandwidth allocated. The reduced m_i means larger H_i required to meet the deadline constraint. This could cause the refined H_i to be no longer sufficient when the m_i reduced. On the other hand, it is clear, from the $X_i(\vec{H})$ expression in Theorem 5, that the second term of $X_i\left(\vec{H}\right)$, i.e., max $(\cdots)$, is reduced when all the other insufficient H_js $(j \neq i)$, as well as the H_i itself, increase. Thus, increment $\frac{\Delta_i}{m_i - 1}$ of H_i may be also insufficient (even though all the m_js remain unchanged). So iterations continue until either $\sum_{i=1}^{n} H_i > \min(P_{\min} - TTRT - \tau, TTRT - \tau)$ or the refined allocation $\vec{H}$ is sufficient for every node.

Note that a previously sufficient H_i may become insufficient, in some iteration, as a result of all the insufficient H_j's $(j \neq i)$ being incremented in the previous iteration (which may cause a decrease or a loss of the available synchronous bandwidth from the second term of $X_i(\vec{H})$). Therefore, the previously sufficient H_i may need to be further incremented/refined so that the deadline constraint remains satisfied for the stream S_i.

The intuitive picture of the refining process of the Min_H procedure, once started, is shown as follows: In the first iterations, some s_i $(1 \leq i \leq n)$ could reduce sharply and frequently from one iteration to another. But, after a certain number of iterations, all the m_i's tend to not change, getting into a stable state. As the number of iterations increases, the general trend of each deficiency (Δ_i) is definitely toward decreasing although the occasional increasing of the Δ_i in some iterations could happen. That is, all the positive Δ_i's will eventually tend to zero as the number of iterations increase. Smaller deficiency Δ_i means smaller increment of H_i required. Hence, the general trend of the increment of H_i, if required, is decreasing, tending to zero. In each iteration, every insufficient H_i with $\Delta_i > 0$ is refined by a properly chosen increment. The iterations continue until either a feasible allocation $\vec{H}$ is eventually produced or the violation of (7) happens. Once a feasible allocation $\vec{H}$ is found, the Min_H procedure returns it with a success status.

As shown above, the Min_H procedure itself actually functions as the EMCA scheme. We name the procedure by Min_H rather than Scheme_EMCA because the main function of this procedure is to search for the minimum allocation vector, $\vec{H}^{\min}$.

B. EMCA—Optimal Synchronous Bandwidth Allocation Scheme

In order to show the optimality of the EMCA scheme, we discuss the properties of the solutions to the inequality system of the protocol constraint (2) and the deadline constraint (5). Define Π to be the set of solutions for both (2) and (5) for the synchronous message set under consideration, that is

$$\Pi = \{\vec{H} | \vec{H} \text{ satisfies (2) and (5)}\}.$$

For two given vectors $\vec{H}' = (H_1', H_2', \cdots, H_h')$ and $\vec{H}'' = (H_1'', H_2'', \cdots, H_h'')$, we say $\vec{H}' < (\leq) \vec{H}''$ if for $i = 1, \cdots, n, H_i' < (\leq) H_i''$. Similar to the MCA scheme, we have the following theorem to list some properties of Π that are of interest.

Theorem 7: If Π is not empty, i.e., the inequality system of (2) and (5) is solvable for the synchronous message set under consideration, then

1) $(\Pi, \leq)$ is a partially ordered set.
2) There is a minimal element $\vec{H}^{\min}$ in Π , i.e., for any $\vec{H}$ in Π, $\vec{H}^{\min} \leq \vec{H}$.
3) $\vec{H}^{\min}$ is bounded. In particular, for $i = 1, 2, \ldots, n$, the ith element of $\vec{H}^{\min}$ is bounded by

$$\frac{C_i}{\left\lfloor \frac{P_i \cdot (n+1)}{n \cdot TTRT} \right\rfloor + 1} \leq H_i{}^{\min} \leq \frac{C_i}{\max\left[\left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1, 1\right]}.$$

Refer to [16] for a proof of this theorem. Like the MCA scheme, the most important property of the procedure Min_H is that for a theoretically schedulable synchronous message set, it always produces an allocation $\vec{H}$, that is, the minimal in Π. For the convenience of proving this property, let $\vec{H}(k)$ be vector $\vec{H}$ at the beginning of the kth iteration. If procedure Min_H successfully exits the *repeat–until* loop at the lth iteration, then, for any $i > 0$, we define

$$\vec{H}(l+i) = \text{the } \vec{H} \text{ when this procedure normally exits the loop.}$$

The following theorem shows some properties of set $\{\vec{H}(k)\}$ produced by the Min_H procedure (i.e., the EMCA scheme) in the refining process (until exit).

TABLE I
SYNCHRONOUS BANDWIDTH ALLOCATIONS FOR MESSAGE SET A

Message Parameters			Synchronous Bandwidth (H_i) allocated by						
i	C_i	P_i	FLA	EPA	PA	NPA	LA	MCA	EMCA
1	30	100	30	25	15	32.61	30	30	30
2	20	125	20	25	8	17.39	20	20	20
Protocol constraint met ?			Yes	Yes	Yes	Yes	Yes	Yes	Yes
Deadline constraint met ?			Yes	No	No	No	Yes	Yes	Yes
Message set guaranteed ?			Yes	No	No	No	Yes	Yes	Yes

TABLE II
SYNCHRONOUS BANDWIDTH ALLOCATIONS FOR MESSAGE SET B

Message Parameters			Synchronous Bandwidth (H_i) allocated by						
i	C_i	P_i	FLA	EPA	PA	NPA	LA	MCA	EMCA
1	30	146	30	25	10.27	22.72	30	15	10
2	36	146	36	25	12.33	27.28	36	18	12
Protocol constraint met ?			No	Yes	Yes	Yes	No	Yes	Yes
Deadline constraint met ?			Yes	Yes	Yes	Yes	Yes	Yes	Yes
Message set guaranteed ?			No	Yes	Yes	Yes	No	Yes	Yes

TABLE III
SYNCHRONOUS BANDWIDTH ALLOCATIONS FOR MESSAGE SET C

Message Parameters			Synchronous Bandwidth (H_i) allocated by						
i	C_i	P_i	FLA	EPA	PA	NPA	LA	MCA	EMCA
1	57	176	57	25	16.19	25	28.5	28.5	19
2	57	176	57	25	16.19	25	28.5	28.5	19
Protocol constraint met ?			No	Yes	Yes	Yes	No	No	Yes
Deadline constraint met ?			Yes	No	No	No	Yes	Yes	Yes
Message set guaranteed ?			No	No	No	No	No	No	Yes

TABLE IV
SYNCHRONOUS BANDWIDTH ALLOCATIONS FOR MESSAGE SET D

Message Parameters			Synchronous Bandwidth (H_i) allocated by						
i	C_i	P_i	FLA	EPA	PA	NPA	LA	MCA	EMCA
1	60	240	60	16.66	12.5	15.40	20	17.8	15
2	120	435	120	16.66	13.79	16.99	17.15	17.03	15
3	180	630	180	16.66	14.29	17.61	17.37	16.37	15
Protocol constraint met ?			No	Yes	Yes	Yes	No	No	Yes
Deadline constraint met ?			Yes	No	No	No	Yes	Yes	Yes
Message set guaranteed ?			No	No	No	No	No	No	Yes

TABLE V
SYNCHRONOUS BANDWIDTH ALLOCATIONS FOR MESSAGE SET E

Message Parameters			Synchronous Bandwidth (H_i) allocated by						
i	C_i	P_i	FLA	EPA	PA	NPA	LA	MCA	EMCA
1	30	90	30	25	16.67	32.86	N/A	N/A	30
2	40	230	40	25	8.70	17.14	N/A	N/A	10
Protocol constraint met ?			No	Yes	Yes	Yes	N/A	N/A	Yes
Deadline constraint met ?			No	No	No	No	N/A	N/A	Yes
Message set guaranteed ?			No	No	No	No	No	No	Yes

TABLE VI
SYNCHRONOUS BANDWIDTH ALLOCATIONS FOR MESSAGE SET F

Message Parameters			Synchronous Bandwidth (H_i) allocated by						
i	C_i	P_i	FLA	EPA	PA	NPA	LA	MCA	EMCA
1	10	75	10	25	6.67	19.39	N/A	N/A	10
2	16	76	16	25	10.53	30.61	N/A	N/A	16
Protocol constraint met ?			Yes	Yes	Yes	Yes	N/A	N/A	Yes
Deadline constraint met ?			No	No	No	No	N/A	N/A	No
Message set guaranteed ?			No	No	No	No	No	No	No

Theorem 8: If Π is not empty, $\left\{\vec{H}(k)\right\}$ produced by the EMCA scheme has the following properties:

1) $\left\{\vec{H}(k)\right\}$ is an increasing sequence, i.e., $\vec{H}(k) \leq \vec{H}(k+1)$.
2) $\left\{\vec{H}(k)\right\}$ is never larger than any element in Π, i.e., for any $\vec{H} \in \Pi$, $\vec{H}(k) \leq \vec{H}$.
3) $\left\{\vec{H}(k)\right\}$ converges, i.e., $\vec{H}^{\lim} = \lim_{k\to\infty} \vec{H}(k)$ exists.
4) $\vec{H}^{\lim} = \lim_{k\to\infty} \vec{H}(k) \in \Pi$, i.e., $\vec{H}^{\lim}$ satisfies (2) and (5).
5) $\left\{\vec{H}(k)\right\}$ converges to $\vec{H}^{\min}$, i.e., $\vec{H}^{\lim} = \lim_{k\to\infty} \vec{H}(k) = \vec{H}^{\min}$.

Refer to [16] for a proof of this theorem. Theorem 9, below, follows directly from Theorem 7 and Theorem 8.

Theorem 9: Allocation scheme EMCA is optimal.

VI. EXAMPLES

In this section we give six synchronous message sets to show that our EMCA scheme is superior to any other SBA schemes. In order to illustrate the superiority, some other previously published SBA schemes [1], [2], [4] are considered for the purpose of comparison. Notice that all the SBA schemes considered in this section assume that deadlines are equal to periods for the message sets under consideration. Due to space limitations, we simply list all these schemes as follows:

- Full length allocation (FLA) scheme [2]: $H_i = C_i$.
- Equal partition allocation (EPA) scheme [2]: $H_i = \frac{TTRT-\tau}{n}$.
- Proportional allocation (PA) scheme [2]: $H_i = \frac{C_i}{P_i} \cdot (TTRT - \tau)$.
- Normalized proportional allocation (NPA) scheme [2]:

$$H_i = \frac{C_i/P_i}{U_s}(TTRT - \tau) \text{ where } U_s = \sum_{i=1}^{n} \frac{C_i}{P_i}.$$

- Local allocation (LA) scheme [1]:[5] $H_i = \frac{C_i}{\lfloor \frac{P_i}{TTRT} \rfloor - 1}$.
- Minimum capacity allocation (MCA) scheme (see [4]).
- Enhanced minimum capacity allocation (EMCA) scheme (see Section V).

The examples are shown by a set of tables (See [16] for more such examples). The message parameters of six synchronous message sets considered (denoted by the capitals from A to F inclusive) are respectively listed in six different tables (from Tables I to VI). For simplicity we assume that $TTRT = 50$ and $\tau = 0$, and denote, in the tables, all these considered SBA schemes by their abridged forms (shown in brackets above). The synchronous bandwidths are calculated by each of considered schemes and then listed in tables.

[5]The local schemes proposed by Malcolm *et al.* [10], [11] and Zheng *et al.* [17] differ from that proposed by Agrawal *et al.* [1] that applies only to synchronous message sets with message deadlines equal to periods, in that they are proposed for guaranteeing synchronous messages with arbitrary deadline constraints. But all these local schemes take the same form as the scheme by Agrawal *et al.* [1] when applied to synchronous message sets with message periods equal to deadlines.

An allocation $\vec{H}$ (no matter which scheme it is produced from) is said to be able to guarantee a message set if it can meet both the deadline constraints (5) and the protocol constraint (2). Generally speaking, the NPA, LA, MCA and EMCA schemes performs better than any of the FLA, EPA and PA schemes because any of the NPA, LA, MCA and EMCA schemes can achieve a relatively higher value of the WCAU, no less than $\frac{1-\alpha}{3}$ [1], [2], [4] (for synchronous message sets with the minimum message period ($P_{\min}$) no less than $2 \cdot TTRT$). Note that both the EPA and NPA schemes are full-allocation schemes, i.e., any allocation $\vec{H}$ produced by either scheme keeps the condition of $\sum_{i=1}^{n} H_i = TTRT - \tau$ true. Any produced allocation $\vec{H}$, therefore, can always satisfy the protocol constraint, and the only checking needed is whether this allocation $\vec{H}$ can also meet the deadline constraints for the message set considered. Those synchronous message sets (e.g., message set C in Table III) that cannot be guaranteed by any full allocation, will never be guaranteed by either of the EPA and NPA schemes. Unlike the EPA and NPA schemes, an allocation produced by the FLA, LA, MCA or EMCA scheme can always meet the deadline constraints for synchronous message sets with $P_{\min} \geq 2 \cdot TTRT$.[6] So for any allocation $\vec{H}$ produced (for a message set with $P_{\min} \geq 2 \cdot TTRT$) by any of these four schemes, only the protocol constraint needs to be checked.

Although both the NPA scheme and the LA scheme are both claimed to be able to guarantee any synchronous message set with its utilization factor no more than 33% [1], [2], they are not equivalent. In Table I, the NPA scheme fails in guaranteeing the message set A while the message set B in Table II cannot be guaranteed by the LA scheme. It should be noticed that a message set failing to be guaranteed by a SBA scheme with a high value of the WCAU, does not mean that this message set cannot be guaranteed by another SBA scheme with a lower value of the WCAU. The message set A shown in Table I can be guaranteed even by the FLA scheme (whose WCAU is 0% [2]) but fails to be guaranteed by the NPA scheme. Table II presents another example where the EPA scheme (whose WCAU is $\frac{1-\alpha}{3n-(1-\alpha)}$ [2]) and even the PA scheme (whose WCAU is 0% [2]) can guarantee the given message set B but the LA scheme cannot. Tables V and VI show two examples where neither the LA scheme nor the MCA scheme is applicable to the message sets given because $P_{\min} < 2 \cdot TTRT$. Tables III, IV, and V are three examples where the given message sets can only be guaranteed by the EMCA scheme. The message set F given in Table VI cannot be guaranteed by EMCA because the produced allocation $\vec{H}$ violates the deadline constraint. Since the allocation scheme EMCA is optimal, no other schemes can guarantee this message set.

VII. Conclusions

This paper has considered and addressed issues pertaining to guaranteeing deadlines of synchronous messages in a timed token ring network such as FDDI where the timed token protocol is used.

[6]Both the LA and MCA schemes can only apply to synchronous message sets with $P_{\min} \geq 2 \cdot TTRT$, due to the restriction of $P_{\min} \geq 2 \cdot TTRT$ inherent with these two schemes.

Guaranteeing message deadlines is a key issue in distributed real time applications. The timing property of bounded token rotation time of the timed token protocol provides a necessary condition to ensure the message deadlines are met. In this paper we present a generalized version of Johnson and Sevcik's theorem [8], [13] that gives the maximum time possibly elapsed in the worst case between any number of consecutive token arrivals to a particular node. Our generalized version is better than previously published [3] in the sense that the upper bound expression we derived is more exact and tighter. Our new exact upper bound expression is important because based on it:

- An optimal SBA scheme can be developed.
- An exact X_i expression (better than previously published [4]) has been derived. Testing the deadline constraint by our new X_i expression may cause some synchronous message sets previously deemed to be unable to be guaranteed by a SBA scheme when the deadline constraint is tested by using the X_i expression derived by Chen *et al.* [4], to become schedulable by the same allocation scheme.

We have proposed in this paper an optimal SBA scheme named EMCA (enhanced MCA), and have demonstrated by examples that the EMCA scheme performs better than the MCA scheme as well as any other SBA scheme. Our work enhances (in nature) the previous work conducted by Chen *et al.* [4] on the MCA scheme, the first so-called optimal SBA scheme with the timed token protocol. To the best of our knowledge, no previous work on the optimal SBA scheme has been reported except for the MCA scheme [4].

Appendix A
Proof of Theorem 4

Before we formally prove Theorem 4, we need the following lemma:

Lemma 1: If $a \geq 1$; $b \geq 0$; $1 \leq a - b < 2$ then, $0 \leq \lfloor a \rfloor - \lceil b \rceil \leq 1$.

Proof: Let $a = b + 1 + c$ (where $0 \leq c < 1$ because $1 \leq a - b < 2$), we have

$$\lfloor a \rfloor - \lceil b \rceil = \lfloor b + 1 + c \rfloor - \lceil b \rceil = 1 + \lfloor b + c \rfloor - \lceil b \rceil. \quad \text{(A1)}$$

There are the following two cases to consider:

Case 1: $\boldsymbol{b}$ is an integer In this case, $\lfloor b + c \rfloor = \lceil b \rceil$. Thus, from (A1) we have

$$\lfloor a \rfloor - \lceil b \rceil = 1 + \lfloor b + c \rfloor - \lceil b \rceil = 1. \quad \text{(A2)}$$

Case 2: $\boldsymbol{b}$ is not an integer In this case, there are two subcases to consider:

Subcase 1: $\boldsymbol{b + c \geq \lceil b \rceil}$: In this subcase, $\lfloor b + c \rfloor = \lceil b \rceil$. Thus, from (A1) we have

$$\lfloor a \rfloor - \lceil b \rceil = 1 + \lfloor b + c \rfloor - \lceil b \rceil = 1. \quad \text{(A3)}$$

Subcase 2: $\boldsymbol{b + c < \lceil b \rceil}$: In this subcase, $\lfloor b + c \rfloor = \lceil b \rceil - 1$. Thus, from (A1) we have

$$\lfloor a \rfloor - \lceil b \rceil = 1 + \lfloor b + c \rfloor - \lceil b \rceil = 0. \quad \text{(A4)}$$

Combine (A2), (A3) and (A4) into one we have $0 \leq \lfloor a \rfloor - \lceil b \rceil \leq 1$.

Theorem 4: Given an allocation $\vec{H} = (H_1, H_2, \ldots, H_h)$ that meets the protocol constraint (2), the positive integer m_i ($i = 1, 2, \cdots, n$) that can satisfy the inequality of $I(m_i - 1) \leq P_i < I(m_i)$ (where $P_i > TTRT$) must be either

$$m_i = \left\lceil \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil$$

or

$$m_i = \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor$$

and

$$0 \leq \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor - \left\lceil \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil \leq 1.$$

Proof: From $I(m_i - 1) \leq P_i$ we have the following derivations:

$$I(m_i - 1) \leq P_i$$

$$\Rightarrow (m_i - 1) \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor \frac{m_i - 1}{n+1} \right\rfloor \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right) \leq P_i$$

$$\Rightarrow \frac{(m_i - 1) \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - P_i}{TTRT - \sum_{h=1}^{n} H_h - \tau} \leq \left\lfloor \frac{m_i - 1}{n+1} \right\rfloor \leq \frac{m_i - 1}{n+1}$$

$$\Rightarrow \frac{(m_i - 1) \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - P_i}{TTRT - \sum_{h=1}^{n} H_h - \tau} \leq \frac{m_i - 1}{n+1}$$

$$\Rightarrow m_i \leq \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}.$$

Because m_i is an integer, we have

$$m_i \leq \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor. \quad \text{(A5)}$$

Similarly, from $P_i < I(m_i)$ we have the following derivations:

$$P_i < I(m_i)$$

$$\Rightarrow P_i < m_i \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor \frac{m_i}{n+1} \right\rfloor \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right)$$

$$\Rightarrow \frac{m_i}{n+1} - \frac{n}{n+1} \leq \left\lfloor \frac{m_i}{n+1} \right\rfloor < \frac{m_i \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - P_i}{TTRT - \sum_{h=1}^{n} H_h - \tau}$$

$$\Rightarrow \frac{m_i - n}{n+1} < \frac{m_i \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - P_i}{TTRT - \sum_{h=1}^{n} H_h - \tau}$$

$$\Rightarrow \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} < m_i.$$

Because m_i is an integer, we further have

$$\left\lceil \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil \leq m_i. \quad \text{(A6)}$$

From (A5) and (A6), we have,

$$\left\lceil \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil \leq m_i \leq \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor. \quad \text{(A7)}$$

Let

$$a = \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}$$

$$b = \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}.$$

Under the protocol constraint (2) (i.e., $\sum_{h=1}^{n} H_h \leq TTRT - \tau$) and the assumption of $P_i > TTRT$ (both are the pre-condition of this theorem), it is easy to check that $a > 1$ and $b > 0$, and we have

$$a - b = \frac{P_i \cdot (n+1) + n \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} - \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}$$

$$= \frac{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - n \left(\sum_{h=1}^{n} H_h + \tau \right) + n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}$$

$$= 1 + \frac{n \cdot \left(TTRT - \sum_{h=1}^{n} H_h - \tau \right)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}.$$

This implies that $1 \leq a - b < 2$. By Lemma 1 we have that $0 \leq \lfloor a \rfloor - \lceil b \rceil \leq 1$, that is

$$0 \leq \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor - \left\lceil \frac{(n+1) \cdot P_i - \sum_{h=1}^{n} H_h - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rceil \leq 1. \tag{A8}$$

Therefore, the theorem follows from (A7) and (A8).

Appendix B
Proof of Theorem 6

Theorem 6: For any schedulable synchronous message set, there must exist at least one feasible allocation $\vec{H} = (H_1, H_2, \cdots, H_h)$ where each H_i $(i = 1, 2, \cdots, n)$ is bounded by

$$\frac{C_i}{\left\lfloor \frac{P_i \cdot (n+1)}{n \cdot TTRT} \right\rfloor + 1} \leq H_i \leq \frac{C_i}{\max\left(\left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1, 1\right)}.$$

Proof: By Corollary 1 we know that

$$I(v) = v \cdot TTRT + \sum_{h=1}^{n} H_h + \tau - \left\lfloor \frac{v}{n+1} \right\rfloor \cdot \left[TTRT - \left(\sum_{h=1}^{n} H_h + \tau \right) \right] \leq v \cdot TTRT + \sum_{h=1}^{n} H_h + \tau.$$

By the above expression, we know that during P_i, node i can use H_i at least $\left\lfloor \frac{P_i - \sum_{h=1}^{n} H_h - \tau}{TTRT} \right\rfloor$ times. Since

$$\left\lfloor \frac{P_i - \sum_{h=1}^{n} H_h - \tau}{TTRT} \right\rfloor \geq \left\lfloor \frac{P_i - TTRT}{TTRT} \right\rfloor = \left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1$$

we see that during P_i, node i can use H_i at least $\left(\left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1\right)$ times. Note that node i should use H_i no less than once (in order to guarantee the message deadline of stream i).[7] Therefore, node i can use H_i, in the worst case, at least $\max\left(\left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1, 1\right)$ times during P_i. This implies that the synchronous bandwidth (H_i) allocated to node i is sufficient for the given synchronous message set to be guaranteed (if the message set is schedulable) when bounded by

$$H_i \leq \frac{C_i}{\max\left(\left\lfloor \frac{P_i}{TTRT} \right\rfloor - 1, 1\right)}. \tag{B1}$$

On the other hand, from the proof process of Theorem 4, we know that in the worst case, during P_i, node i can use H_i at most m_i times bounded by (A5). From (A5) we have the following derivations:

$$\begin{aligned} m_i &\leq \left\lfloor \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} \right\rfloor \\ &= \left\lfloor \frac{(n+1) \cdot (P_i - \sum_{h=1}^{n} H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau} + 1 \right\rfloor \\ &= \left\lfloor \frac{P_i - \sum_{h=1}^{n} H_h - \tau}{\frac{n \cdot TTRT + \sum_{h=1}^{n} H_h + \tau}{n+1}} \right\rfloor + 1 \\ &= \left\lfloor \frac{P_i - \sum_{h=1}^{n} H_h - \tau}{TTRT - \frac{TTRT - \sum_{h=1}^{n} H_h - \tau}{n+1}} \right\rfloor + 1 \\ &\leq \left\lfloor \frac{P_i - \sum_{h=1}^{n} H_h - \tau}{TTRT - \frac{TTRT}{n+1}} \right\rfloor + 1 \\ &\leq \left\lfloor \frac{P_i}{TTRT - \frac{TTRT}{n+1}} \right\rfloor + 1 \\ &= \left\lfloor \frac{(n+1) \cdot P_i}{n \cdot TTRT} \right\rfloor + 1. \end{aligned}$$

That is, $m_i \leq \left\lfloor \frac{(n+1) \cdot P_i}{n \cdot TTRT} \right\rfloor + 1$. This implies that for guaranteeing synchronous message deadlines, the synchronous bandwidth (H_i) has to be allocated such that

$$\frac{C_i}{\left\lfloor \frac{(n+1) \cdot P_i}{n \cdot TTRT} \right\rfloor + 1} \leq H_i. \tag{B2}$$

Thus, the theorem follows from (B1) and (B2).

Acknowledgments

We wish to thank the anonymous reviewers and the editor for their careful reading of the manuscript and suggestions that help to clarify and improve this paper.

[7] Here, we assume that for $i = 1, 2, \cdots, n$, $H_i \leq C_i$ because allocating H_i more than C_i does not make more sense for satisfying the deadline constraint but, on the contrary, runs the risk of violating the protocol constraint.

References

[1] A. G. Agrawal, B. Chen, and W. Zhao, "Local synchronous capacity allocation schemes for guaranteeing message deadlines with the timed token protocol," presented at INFOCOM, 1993.

[2] A. G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing synchronous message deadlines with the timed token medium access control protocol," *IEEE Trans. Comput.*, vol. 43, no. 3, pp. 327–339, Mar. 1994.

[3] B. Chen and W. Zhao, "Properties of the timed token protocol," Computer Science Dept., Texas A&M University, College Station, TX, Tech. Rep. 92-038, Oct. 1992.

[4] B. Chen, G. Agrawal, and W. Zhao, "Optimal synchronous capacity allocation for hard real time communications with the timed–token protocol," in *Proc. 13th IEEE Real Time Systems Symp.*, 1992, pp. 198–207.

[5] R. M. Grow, "A timed–token protocol for local area networks," in *Proc. ELECTRO, Token Access Protocols*, paper 17/3, 1982.

[6] M. Hamdaoui and P. Ramanathan, "Selection of timed token mac protocol parameters to guarantee message deadlines," Dept. of Electrical and Comp. Eng., Univ. of Wisconsin, Madison, WI, Tech. Rep., Nov. 1992.

[7] R. Jain, *FDDI Handbook—High Speed Networking Using Fiber and Other Media.* Reading, MA: Addison–Wesley, 1994

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

[9] C. C. Lim, L. Yao, and W. Zhao, "Transmitting time-dependent multimedia data in FDDI networks," in *Proc. SPIE Int. Symp. OE/FIBERS*, 1992, pp. 144–154.
[10] N. Malcolm and W. Zhao, "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network," in *Proc. IEEE Conf. Local Comp. Networks*, 1993, pp. 186–195.
[11] ——, "The timed-token protocol for real time communications," *Comp.* vol. 27, no. 1, pp. 35–41, Jan. 1994.
[12] ——, "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network," Dept. Comp. Science, Texas A&M University, College Station, TX, Tech. Rep., 1993.
[13] 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, Mar. 1987.
[14] S. Zhang and A. Burns, "On the schedulability of synchronous message sets with the minimum message deadline less than 2*TTRT in an FDDI network," in *Proc. IC3N Fourth Int. Conf. Comp. Commun. Networks*, Las Vegas, NV, 1995, pp. 498–501.
[15] ——, "Exact timing properties of the timed token protocol," Dept of Computer Science, University of York, UK, Tech. rep. (YCS 243) (available via ftp: //ftp.cs.york.ac.uk/pub/realtime/papers), May 1994.
[16] ——, "EMCA—an optimal synchronous bandwidth allocation scheme for guaranteeing synchronous message deadlines with the timed token protocol in an FDDI network," Dept. of Computer Science, University of York, UK, Tech. rep. YCS 244 (available via ftp: //ftp.cs.york.ac.uk/pub/realtime/papers), 1994.
[17] Q. Zheng and K.G. Shin, "Synchronous bandwidth allocation in FDDI networks," in *Proc. ACM Multimedia*, 1993, pp. 31–38.

Sijing Zhang received the B.Sc. and M.Sc. degrees in computer science from Jilin University, Changchun, Jilin, PRC, in 1982 and 1988 respectively. He is working toward the Ph.D. degree in computer science at the University of York, UK.

His research interests include real time communications, computer networks, distributed systems, and computer architecture.

Alan Burns (SM'92) received the Ph.D. degree from the University of York, UK, in 1978.

He is the Professor of Real Time Systems in the Department of Computer Science, University of York, U.K. He was promoted to a personal chair in 1994. His research interests cover a number of aspects of real time systems including; the assessment of languages for use in the real time domain, communications, distributed operating systems, the formal specification of scheduling algorithms and implementation strategies, and the design of dependable user interfaces to real time applications. Professor Burns has authored, or co-authored, more than 180 papers and reports, and eight books. Many of these are in the real time area. His teaching activities include courses in Operating Systems and Real Time Systems.