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

Gopal Agrawal
Baio Chen
Wei Zhao
Texas A&M University

Sadegh Davari
University of Houston-Clear Lake

March 22, 1992
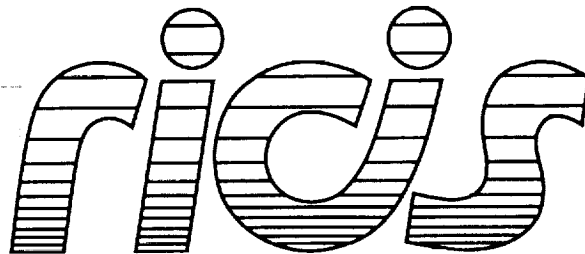
N92-32656

Unclas

G3/61   0116936

(NASA-CR-190710) GUARANTEEING
SYNCHRONOUS MESSAGE DEADLINES WITH
THE TIMED TOKEN MEDIUM ACCESS
CONTROL PROTOCOL (Research Inst.
for Computing and Information
Systems) 45 p

*JOHNSON -*
*IN-61-CR*
*116936*
*p-45*

*Research Institute for Computing and Information Systems*
*University of Houston-Clear Lake*

# TECHNICAL REPORT

# The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information Systems (RICIS) in 1986 to encourage the NASA Johnson Space Center (JSC) and local industry to actively support research in the computing and information sciences. As part of this endeavor, UHCL proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a continuing cooperative agreement with UHCL beginning in May 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The UHCL/RICIS mission is to conduct, coordinate, and disseminate research and professional level education in computing and information systems to serve the needs of the government, industry, community and academia. RICIS combines resources of UHCL and its gateway affiliates to research and develop materials, prototypes and publications on topics of mutual interest to its sponsors and researchers. Within UHCL, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business and Public Administration, Education, Human Sciences and Humanities, and Natural and Applied Sciences. RICIS also collaborates with industry in a companion program. This program is focused on serving the research and advanced development needs of industry.

Moreover, UHCL established relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research. For example, UHCL has entered into a special partnership with Texas A&M University to help oversee RICIS research and education programs, while other research organizations are involved via the "gateway" concept.

A major role of RICIS then is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. RICIS, working jointly with its sponsors, advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research and integrates technical results into the goals of UHCL, NASA/JSC and industry.

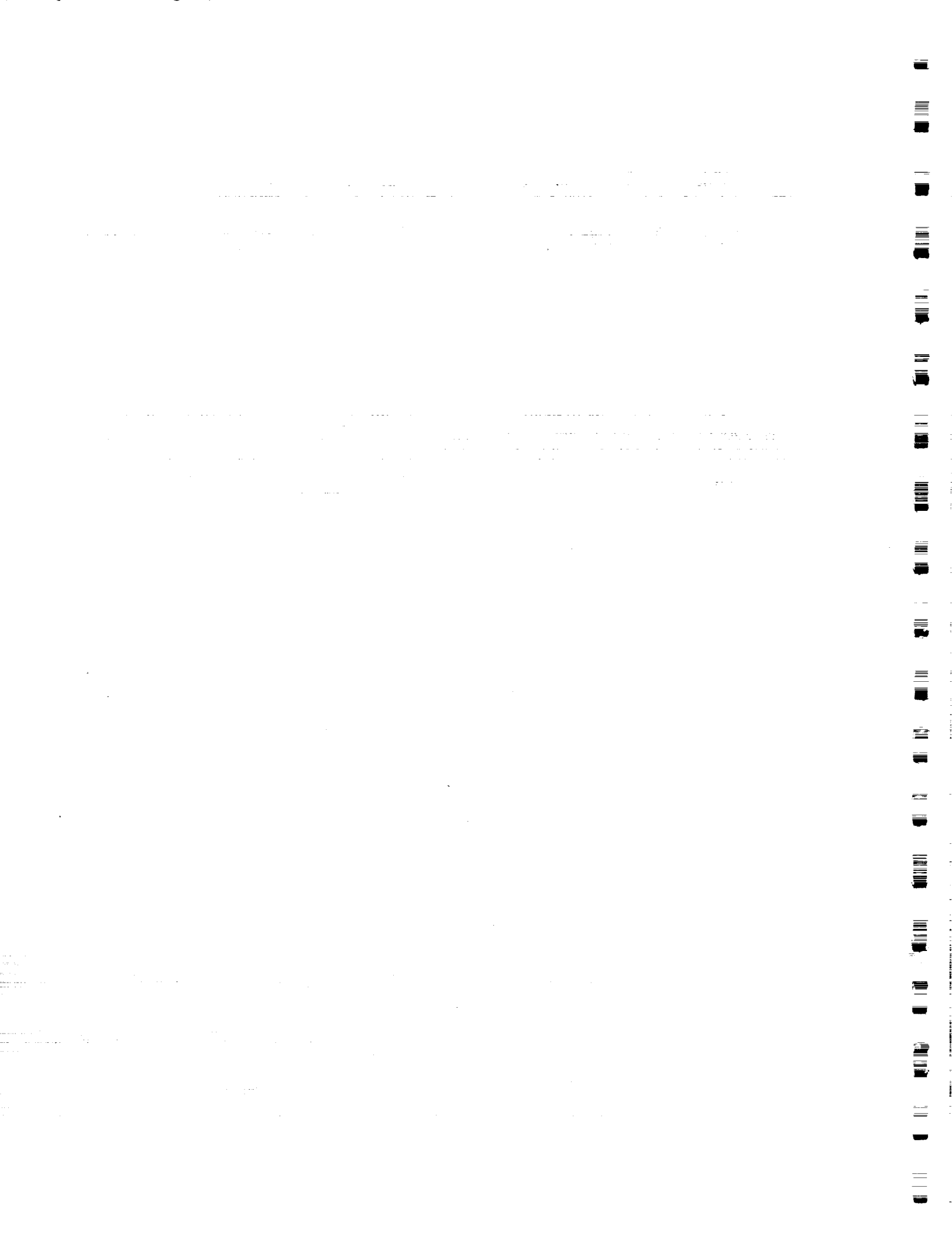# Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Access Control Protocol

## RICIS Preface

This research was conducted by Gopal Agrawal, Baio Chen and Wei Zhao of Texas A&M University and Dr. Sadegh Davari of the Department of Computer Science at the University of Houston-Clear Lake. The research was supported by the RICIS Program Office and in part by an Engineering Excellence grant from Texas A&M University.

RICIS research support funds are derived from Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake.

The views and conclusions contained in this report are those of the authors and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, Texas A&M, NASA or the United States Government.

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

Gopal Agrawal. Baio Chen. and Wei Zhao
Department of Computer Science
Texas A&M University
College Station. TX 77843

Sadegh Davari
Department of Computer Science
University of Houston – Clear Lake
Houston. TX 77058

March 22. 1992

## 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 capacity. 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 capacity 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. To date. no other synchronous capacity allocation scheme has been reported to achieve such substantial performance.

Another major contribution of this paper is an extension to the previous work on the bounded token rotation time. We prove that the time elapsed between any $v$ consecutive visits to a particular node is bounded by $v \cdot TTRT$ where $TTRT$ is the target token rotation time set up at system initialization time. The previous result by Johnson and Sevcik [26. 49] is a special case where $v = 2$. We use this result in the analysis of various synchronous allocation schemes. It can also be applied in other similar studies.

**Key Words:** Hard Real-Time. Distributed System. FDDI. Timed Token Medium Access Control Protocol. Synchronous Messages. Performance Evaluation and Analysis. Synchronous Capacity. Worst Case Achievable Utilization.

# Contents

# Glossary

| Term | Definition | Location |
|------|------------|----------|
| $C_i$ | The length (i.e.. transmission time) of a message in synchronous message stream $S_i$. | 3.2. |
| $H_i$ | The synchronous capacity allocated to node $i$. | 4.1 |
| $LC_i$ | The late counter at node $i$. | 4.1. |
| $NA_i$ | The set of asynchronous messages at node $i$. | A. |
| $NS_i$ | The set of synchronous message streams at node $i$ | A. |
| $P_i$ | The period length of synchronous messages stream $S_i$. | 3.2. |
| $R_i(l)$ | The time at which the token is expected to arrive at node $i$ after its $l^{th}$ visit at that node. | B. |
| $S_{i_j}$ | The $j^{th}$ synchronous message stream at node $i$. | A. |
| $THT_i$ | The token holding timer at node $i$. | 4.1. |
| $TRT_i$ | The token rotation timer at node $i$. | 4.1. |
| $TRT_i'$ | It is defined as being equal to $TRT_i + (1 - LC_i) \cdot TTRT$. | B. |
| $TTRT$ | The Target Token Rotation Time. | 4.1 |
| $T(N_i)$ | The transformation of node $i$ to a set of virtual nodes. | A. |
| $U(M)$ | The utilization factor of the synchronous messages. i.e.. fraction of the time spent by the network in transmission of the synchronous messages. | 3.2. |
| $U_x$ | The Achievable Utilization of synchronous capacity allocation scheme $x$. | 4.3.3. |
| $U_x^*$ | The Worst Case Achievable Utilization of synchronous capacity allocation scheme $x$. | 4.3.3. |
| $VN_{i_j}$ | The $j^{th}$ virtual node derived from node $i$ after its transformation. | A. |
| $X_i$ | The amount of time available to node $i$ to transmit its synchronous messages within a given period. | 5. |
| $m$ | The number of (virtual) nodes in the network. | 3.1. |
| $n$ | The number of synchronous message streams in the network. In this paper. it is assumed that $n = m$. | 3.2. |
| $\theta_i$ | The latency between node $i$ and its upstream neighbor. | 3.1. |
| $\Theta$ | The total ring latency or token walk time. | 3.1. |
| $\Delta$ | The protocol dependent overheads. | 4.3.2. |
| $\tau$ | The portion of the $TTRT$ that is unavailable to transmit synchronous messages. | 4.3.2. |
| $\alpha$ | The ratio of $\tau$ to the Target Token Rotation Time ($TTRT$). | 4.3.2. |
| $t_i(l)$ | The time when the token arrives at node $i$ in its $l^{th}$ visit. | 5. |

# 1 Introduction

High speed networks are vital for the support of distributed real-time applications (e.g., voice/video transmission, process control). 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. There are two common types of time constraints: *laxity*, which specifies the maximum time a message can wait before its transmission begins, and *deadline*, which defines the latest time by which the transmission of the message must finish. In this paper, we address the issue of guaranteeing synchronous message deadlines in high speed networks. By *guaranteeing*, we mean that as long as the network operates normally (i.e., no failures), synchronous messages are always transmitted before their deadlines.

We address the issue of guaranteeing message deadlines with the *timed token medium access control (MAC) protocol* [15]. This protocol is suitable for real-time applications not only because of its use in high bandwidth networks but also due to the fact that it has the important property of bounded access time which is necessary for real-time communications. The timed token protocol has been incorporated into many network standards, including the Fiber Distributed Data Interface (FDDI) [1, 2], IEEE 802.4 [19], the High-Speed Data Bus and the High-Speed Ring Bus (HSDB/HSRB) [8, 46, 47, 58], and the Survivable Adaptable Fiber Optic Embedded Network (SAFENET) [14, 28, 35, 40]. Many embedded real-time applications use them as backbone networks. For example, the FDDI has been selected as a backbone network for NASA's Space Station Freedom [4, 7, 61].

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 capacity*,[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 message, if any, for a time no more than its allocated synchronous capacity. 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 [26, 49], Johnson and Sevcik formally proved that when the network operates normally (i.e. there is no failure), the token rotation time between two consecutive visits to a node is bounded by twice the expected token rotation time

---

[1] Some other synonymous terms that researchers use are: *Bandwidth allocation* [59], *Synchronous allocation* [23], *Synchronous bandwidth assignments* [26], and *High Priority token holding time* [41].

1

(i.e., $2 \cdot TTRT$).

Although the prerequisite of 'bounded token rotation time' is indispensable, it is insufficient for guaranteeing message deadlines. A node with inadequate synchronous capacity may be unable to complete the transmission of a synchronous message before its deadline. On the other hand, allocating excess amounts of synchronous capacities 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 capacities to the nodes. As pointed out in [26], the allocation of synchronous capacities is an open problem. The main objective of this study is to analyze and evaluate the synchronous capacity allocation schemes used with the timed token protocol in a hard real-time communication system.

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 [31] 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, 50] and is used in many hard real-time applications [10].

Intuitively, one would believe that a communication protocol which implements the rate monotonic transmission policy is the most desirable for a real-time communication 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 capacity allocation schemes based on the Worst Case Achievable Utilization.

In this paper, four synchronous capacity 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 capacity could be used by asynchronous traffic. To the best of our knowledge, no other scheme has been reported to achieve a better utilization. Hence, this allocation scheme should be recommended for use in hard real-time communication networks that use the timed token MAC protocol.

Another major contribution of this paper is to extend the analysis of the bound on the token rotation time given by Johnson and Sevcik in [26, 49]. We show that the time elapsed between any $v$ consecutive arrivals of the token at a node is bounded by $v \cdot TTRT$. The previous result by Johnson and Sevcik [26, 49] is a special case where $v = 2$. Our newer bound is used in the analysis of the synchronous capacity allocation schemes and will be applicable in other similar studies.

The remainder of the paper is organized as follows: Section 2 will review the previous relevant work. Section 3 will outline the characteristics of the system under consideration. i.e., the message and network models. The timed token protocol and the synchronous capacity allocation schemes are introduced in Section 4. Section 5 discusses some timing properties of the protocol. In Section 6 we will study several allocation schemes and derive their Worst Case Achievable Utilizations. Section 7 contains the concluding remarks and suggestions for future work.

# 2   Previous Relevant Work

Extensive research has been done on the timed token protocol since it was first proposed by Grow [15] in 1982. Introductory tutorials on this protocol and its use in networking standards can be found in the papers by Ross [43, 44, 45]. Iyer and Joshi [20, 21] and others [34, 52, 53].

Some important characteristics and architectural design considerations of FDDI token ring networks are discussed in [3, 12, 13, 16, 18, 22, 27, 33, 38, 55, 57]. The various fault recovery and ring management procedures of the FDDI are outlined in the papers by Ocheltree and Montalvo [38, 39]. An overview of the FDDI MAC services is given in [57]. Design considerations and the role of concentrators are discussed in [17, 18]. Issues concerned with interoperability and interconnection of FDDI with heterogeneous networks can be found in [3, 6, 16, 33].

The timing properties of the FDDI token ring were first formally analyzed by Johnson and Sevcik in [26, 49]. Other interesting timing properties of the FDDI were given in a study conducted by Jain [23]. He suggests that a value of 8 ms for TTRT is desirable as it can achieve 80% utilization on all 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 [48] to study the influence of the target token rotation time (TTRT) on the performance of various FDDI ring configurations.

Uhm [59] discussed the performance characteristics of the timed token protocol with respect to parameters

3

such as the channel capacity, the network cable length, and the number of stations. Dykeman and Bux [11] studied and developed 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 [41], Jayasumana and Werahera [24], Valenzo, Montuschi, and Ciminiera [60], etc.

Note that none of the above studies on the timed token protocol have specifically addressed the use and performance of the protocol in hard real-time environments. 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 are addressed in [5, 29, 32, 42, 51, 56, 63, 64, 65, 66]. The real-time performance of various token ring protocols are considered in [30, 37, 51, 54, 62]. 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.

# 3 System Characteristics

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

## 3.1 Network model

We consider the network topology as consisting of $m$ nodes connected by point-to-point links forming a circle i.e., the token ring. A special bit pattern called the *token* circulates around ring (from node $i$ to nodes $i + 1$, $i + 2$, ... until node $m$, then to nodes 1, 2, ...), helping to determine which node should send a frame of message among the contending nodes.

We denote the latency between a node $i$ and its upstream neighbor[2] by $\theta_i$. 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* $\Theta$, i.e., $\sum_{i=1}^{n} \theta_i = \Theta$. Thus, the ring latency $\Theta$ denotes the token walk time around the ring when none of the nodes in the network disturb it.

## 3.2 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, $S_1, S_2, \ldots, S_n$ in the system which form a synchronous *message set*, $M$, i.e.,

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

---

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

4

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 $S_i$ ($i = 1, 2, \ldots, n$).

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 non-periodic and do not have a hard real-time deadline requirement.

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,

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

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. 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.

# 4 Timed Token Medium Access Control Protocol

## 4.1 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 $2 \cdot TTRT$ [26], 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 $1 \leq i \leq n$,

$$TTRT \leq \frac{P_i}{2} \tag{3}$$

5

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.

$$P_i = m_i \cdot TTRT - \delta_i. \tag{1}$$

where $m_i = \lceil \frac{P_i}{TTRT} \rceil \geq 2$. If $m_i = 2$. then $\delta_i = 0$ and if $m_i \geq 3$ then $0 \leq \delta_i < TTRT$. 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 capacity 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.[3] This paper will deal with the issue of appropriate allocation of these $H_i$ values.

3. *Token Rotation Timer of node $i$ ($TRT_i$).* This counter is initialized to equal TTRT. and counts down until it expires (i.e.. $TRT_i = 0$) 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 $i$.

## 4.2   Protocol operation

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

1. $THT_i \leftarrow 0$:

2. $LC_i \leftarrow 0$:

3. $TRT_i \leftarrow TTRT$.

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

1. $TRT_i \leftarrow TTRT$:

2. $LC_i \leftarrow LC_i + 1$.

The $TRT_i$ 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 [25].

A token is considered to arrive *early* at node $i$ if $LC_i = 0$ at the time of its arrival. The token is *late* if $LC_i > 0$.

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

---

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

1. $THT_i \gets TRT_i$:

2. $TRT_i \gets TTRT$:

3. Synchronous frames (if any) can then be transmitted for a maximum time of $H_i$ (i.e.. the synchronous capacity 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 $THT_i > 0$ and $TRT_i > 0$.

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

1. $LC_i \gets 0$:

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.

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

## 4.3  Synchronous capacity allocation schemes

As mentioned earlier. synchronous capacity 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.

### 4.3.1  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 capacity *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 capacities $H_i$ to be allocated to station $i$ in the network. Formally. let function $f$ represent an allocation scheme. Then.

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

Let us consider a simple example. We assume a network with only 3 nodes. We have the following values for the message set's parameters:

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

7

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

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

we obtain the values of synchronous capacities as:

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

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

In Section 6, 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.

### 4.3.2 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 $\Theta$ 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 $\tau$. That is, $\tau = \Theta + \Delta$ where $\Delta$ represents the protocol dependent overheads.[4] We define the ratio of $\tau$ to the target token rotation time (TTRT) to be $\alpha$. The usable ring utilization available for synchronous messages would therefore be $(1 - \alpha)$ [59].

  Thus, a protocol constraint on the allocation of synchronous capacities is that *the sum total of the synchronous capacities allocated to all nodes in the ring should not be greater than the available portion of the Target Token Rotation Time (TTRT)*, i.e..

$$\sum_{i=1}^{n} H_i \leq TTRT - \tau. \tag{9}$$

- *Deadline constraint: The allocation of the synchronous capacities 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 $(t, t + P_i)$.

---

[4]For example, according to the FDDI standard, the protocol dependent overheads include the token transmission time, asynchronous overrun, etc. Refer to [1] for details.

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.

### 4.3.3 Performance metric

Numerous synchronous capacity 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 1, 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 capacity allocation schemes.

## 5  Protocol Timing Properties

Although extensive work has been done on the timing behavior of the timed token protocol, we need to further explore additional timing properties of the protocol in order to carry out analysis of the allocation schemes. To analyze an allocation scheme, we should 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 $\tau$) 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 within its period.

Johnson and Sevcik showed that any two consecutive token visits to a node are bounded by $2 \cdot TTRT$. 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 messages. However, this bound is not tight when the period is longer than $3 \cdot TTRT$. Because of this, we need to generalize the analysis done by Johnson and Sevcik to obtain a tight bound on the time elapsed between any $v$ consecutive visits by the token to a particular node. This then leads us to a derivation of a tight lower bound on the time available for a node to transmit its synchronous messages within a given time period.

Let $t_i(l)$ ($l = 1, 2, \ldots$) denote the time when the token makes its $l^{th}$ visit to node $i$.

**THEOREM 5.1** *(Johnson and Sevcik's Theorem [26, 49])*

*For any integer $l > 0$ and any node $i$ ($1 \leq i \leq n$),*

$$t_i(l+1) - t_i(l) \leq 2 \cdot TTRT - H_i \leq 2 \cdot TTRT. \tag{11}$$

Refer to Appendix B for a proof of the above theorem. This theorem gives the upper bound between two consecutive token arrivals as $2 \cdot TTRT$. A formal proof for the above result was first obtained by Johnson and Sevcik in [26, 49]. The tighter upper bound of $(2 \cdot TTRT - H_i)$ will be useful in the analysis of synchronous capacity allocation schemes in Section 6. Next, we will derive a generalized version of this theorem.

**THEOREM 5.2** *(Generalized Johnson and Sevcik's Theorem)*

*For any integer $l > 0$, $v > 0$ and any node $i$ ($1 \leq i \leq n$),*

$$t_i(l+v-1) - t_i(l) \leq v \cdot TTRT - H_i. \tag{12}$$

Refer to Appendix B 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 $v = 2$. The upper bound specified by (12) is tight in the sense that the equal sign holds in the worst case situation.

**COROLLARY 5.1** *Assume that at time $t$, a synchronous message with period $P_i$ arrives at node $i$ ($1 \leq i \leq n$). Then, in time interval $(t, t + P_i)$ the total amount of time ($X_i$) available for node $i$ to transmit this synchronous message is bounded by*

$$X_i \geq \lfloor \frac{P_i}{TTRT} - 1 \rfloor \cdot H_i. \tag{13}$$

*In the worst case, the lower bound will be tight if*

$$\delta_i = \lceil \frac{P_i}{TTRT} \rceil \cdot TTRT - P_i \geq H_i. \tag{14}$$

Refer to Appendix B for a proof of the above corollary. This corollary will be used extensively in the analysis of our synchronous capacity allocation schemes. Figure 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 (13): 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 may be 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 Corollary 5.1.

# 6  Analysis of Synchronous Capacity Allocation Schemes

In this section we consider four synchronous capacity 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 non-zero 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 $P_i s, C_i s, H_i s$, 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 capacity ($H_i$) available to each node $i$. Both protocol and deadline constraints must be satisfied by the allocation of these synchronous capacities. Message sets with the least possible utilization factors are then searched such that the allocation of the synchronous capacities 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 C.

**LEMMA 6.1**   *For any synchronous message stream $i$ $(1 \leq i \leq n)$ we have*

$$\frac{\lfloor \frac{P_i}{TTRT} - 1 \rfloor}{P_i/TTRT} \geq \frac{1}{3 - \frac{s_i}{TTRT}} \geq \frac{1}{3}. \tag{15}$$

## 6.1  Full length allocation scheme

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

$$H_i = C_i. \tag{16}$$

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 capacity allocated is

11

sufficient, the Worst Case Achievable Utilization is zero because the protocol constraint may be violated, as shown in the next theorem.

**THEOREM 6.1** *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 capacity of the nodes is allocated using the full length scheme.

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

$$C_1 = (1 - \tfrac{1}{k})\epsilon, \quad P_1 = 1.$$
$$C_2 = \tfrac{2-\epsilon}{k}, \quad P_2 = \tfrac{2-\epsilon}{\epsilon}.$$

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

The utilization factor is

$$
\begin{aligned}
U &= \sum_{i=1}^{n} C_i/P_i = \frac{(1-\frac{1}{k})\epsilon}{1} + \frac{(2-\epsilon)/k}{(2-\epsilon)/\epsilon} \\
&= \epsilon - \frac{\epsilon}{k} + \frac{\epsilon}{k} = \epsilon.
\end{aligned}
\tag{17}
$$

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

$$
\begin{aligned}
\sum_{i=1}^{n} H_i &= \sum_{i=1}^{n} C_i = C_1 + C_2 \\
&= (1 - \frac{1}{k})\epsilon + \frac{(2-\epsilon)}{k} \\
&= \frac{2}{k} + \epsilon(1 - \frac{2}{k}).
\end{aligned}
\tag{18}
$$

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

$$
\begin{aligned}
\sum_{i=1}^{n} H_i &\geq \frac{2}{k} \\
&> \frac{1}{k} \geq TTRT - \tau.
\end{aligned}
\tag{19}
$$

We see that this scheme may over-allocate the synchronous capacity for a message set with utilization $U \leq \epsilon$. The protocol constraint is therefore not satisfied. Since $\epsilon$ can be arbitrarily close to 0, the Worst Case Achievable Utilization of this scheme can asymptotically approach 0%. Q.E.D.

12

## 6.2 Proportional allocation scheme

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

$$H_i = \frac{C_i}{P_i} \cdot (TTRT - \tau). \tag{20}$$

**THEOREM 6.2** *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 capacity 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:

$$
\begin{aligned}
C_1 &= (1 - \tfrac{1}{k})\epsilon', & P_1 &= 1, \\
C_2 &= (1 + \tfrac{1}{k} - \epsilon')\tfrac{\epsilon'}{k}, & P_2 &= 1 + \tfrac{1}{k} - \epsilon'.
\end{aligned}
\tag{21}
$$

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

The utilization factor is

$$
\begin{aligned}
U &= \frac{C_1}{P_1} + \frac{C_2}{P_2} \\
&= (\epsilon' - \frac{\epsilon'}{k}) + \frac{\epsilon'}{k} = \epsilon' \leq \epsilon.
\end{aligned}
\tag{22}
$$

The synchronous capacity allocated to node 2 is

$$H_2 = \frac{C_2}{P_2} \cdot (TTRT - \tau) \leq \frac{C_2}{P_2} \cdot TTRT = \frac{\epsilon'}{k^2}. \tag{23}$$

Furthermore, because

$$
\begin{aligned}
\delta_2 &= \lceil \frac{P_2}{TTRT} \rceil \cdot TTRT - P_2 = \lceil \frac{1 + 1/k - \epsilon'}{1/k} \rceil \cdot \frac{1}{k} - (1 + \frac{1}{k} - \epsilon') \\
&= \lceil k + 1 - k\epsilon' \rceil \cdot \frac{1}{k} - 1 - \frac{1}{k} + \epsilon' \qquad \text{(since } 0 < k\epsilon' \leq 1/2) \\
&= 1 + \frac{1}{k} - 1 - \frac{1}{k} + \epsilon' \\
&= \epsilon' > \frac{\epsilon'}{k^2} = H_2
\end{aligned}
\tag{24}
$$

from Corollary 5.1, in the worst case the total amount of time ($X_2$) for node 2 to transmit its synchronous message in a period of $P_2$ is given by

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

13

$$= \frac{k-1}{k^2}\epsilon' = (1 - \frac{1}{k})\frac{\epsilon'}{k}$$

$$= (1 + \frac{1}{k} - \epsilon' - \frac{2}{k} + \epsilon')\frac{\epsilon'}{k} = (1 + \frac{1}{k} - \epsilon')\frac{\epsilon'}{k} - (\frac{2}{k} - \epsilon')\frac{\epsilon'}{k}$$

$$< (1 + \frac{1}{k} - \epsilon')\frac{\epsilon'}{k} = C_2. \qquad (25)$$

We see that the deadline constraint cannot be satisfied at node 2. Since $\epsilon$ can be arbitrarily close to 0, the Worst Case Achievable Utilization of this scheme can asymptotically approach 0%. Q.E.D.

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 is 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 $P_i$.

## 6.3 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.,

$$H_i = \frac{TTRT - \tau}{n}. \qquad (26)$$

where $n$ is the number of nodes in the system.

**THEOREM 6.3** *The Worst Case Achievable Utilization of the equal partition synchronous capacity allocation scheme is $\frac{1}{3n-(1-\alpha)} \cdot (1 - \alpha)$ where $\alpha = \frac{\tau}{TTRT}$ 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 capacities are allocated by using the equal partition scheme.

A detailed proof of this theorem is presented in Appendix C.

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 Utilzation of this scheme occurs because the allocation of the synchronous capacity 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 capacity 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.

14

## 6.4 Normalized proportional allocation scheme

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

$$H_i = \frac{C_i/P_i}{U} \cdot (TTRT - \tau). \tag{27}$$

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

**THEOREM 6.4** *The Worst Case Achievable Utilization factor of the normalized proportional allocation scheme is $\frac{1}{3}(1-\alpha)$ where $\alpha = \frac{\tau}{TTRT}$.*

**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 $\sum_{i=1}^{n} \frac{C_i}{P_i} = U \le 1$.

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 using the normalized proportional scheme.

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

$$\sum_{i=1}^{n} H_i = \sum_{i=1}^{n} \frac{C_i/P_i}{U} \cdot (TTRT - \tau) = TTRT - \tau. \tag{28}$$

Hence. the protocol constraint (9) will be satisfied.

**Proof of Statement 2:** Consider a message set whose utilization factor $U(M) \le \frac{1}{3}(1-\alpha)$. From Lemma 6.1. we have

$$U \le \frac{1}{3}(1-\alpha) \ \le \ \frac{\lfloor \frac{P_i}{TTRT} - 1 \rfloor}{P_i/TTRT}(1-\alpha)$$

$$\le \ \frac{\lfloor \frac{P_i}{TTRT} - 1 \rfloor}{P_i}(TTRT - \tau). \tag{29}$$

Multiplying with $C_i/U$ on both sides. we get

$$C_i \le \frac{\lfloor \frac{P_i}{TTRT} - 1 \rfloor}{P_i} \cdot (TTRT - \tau) \cdot C_i}{U}. \tag{30}$$

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

$$C_i \ \le \ \lfloor \frac{P_i}{TTRT} - 1 \rfloor \cdot \frac{C_i}{P_i U} \cdot (TTRT - \tau). \tag{31}$$

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

$$C_i \le \lfloor \frac{P_i}{TTRT} - 1 \rfloor \cdot H_i. \tag{32}$$

15

From Corollary 5.1 and (32), we see that any node $i$ can transmit its synchronous message before the deadline.

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

$$\epsilon' = \min(\frac{1-\alpha}{3}, \epsilon). \tag{33}$$

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

$$\begin{aligned}
C_1 &= \epsilon', & P_1 &= 1. \\
C_2 &= \epsilon', & P_2 &= \frac{3}{2} - \epsilon'. \\
C_3 &= 1 - 3\epsilon' - \alpha, & P_3 &= 3.
\end{aligned} \tag{34}$$

Note that Equation (33) guarantees that $C_3 \geq 0$. All other $C_i = 0$ for $i > 3$.

The utilization of this message set is

$$\begin{aligned}
U &= \frac{C_1}{P_1} + \frac{C_2}{P_2} + \frac{C_3}{P_3} \\
&= \epsilon' + \frac{\epsilon'}{(3/2) - \epsilon'} + \frac{1}{3} - \epsilon' - \frac{1}{3}\alpha \\
&= \frac{1}{3}(1 - \alpha) + \frac{\epsilon'}{(3/2) - \epsilon'}.
\end{aligned} \tag{35}$$

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

$$U < \frac{1}{3}(1 - \alpha) + \epsilon' \leq \frac{1}{3}(1 - \alpha) + \epsilon. \tag{36}$$

Consider the synchronous capacity allocated to node 2:

$$\begin{aligned}
H_2 &= \frac{C_2}{P_2 \cdot U} \cdot (TTRT - \tau) = \frac{C_2}{P_2 \cdot U} \cdot TTRT \cdot (1 - \alpha) \\
&= C_2 \cdot \frac{\frac{1}{2}(1 - \alpha)}{(\frac{3}{2} - \epsilon')(\frac{1}{3}(1 - \alpha) + \frac{\epsilon'}{\frac{3}{2} - \epsilon'})} = C_2 \cdot \frac{\frac{1}{2}(1 - \alpha)}{\frac{1}{2}(1 - \alpha) - \frac{1}{3}\epsilon'(1 - \alpha) + \epsilon'} \\
&= C_2 \cdot \frac{1}{1 - \frac{2}{3}\epsilon' + \frac{2\epsilon'}{(1-\alpha)}} = C_2 \cdot \frac{1}{1 + \frac{2\epsilon'}{(1-\alpha)}(1 - \frac{1-\alpha}{3})}.
\end{aligned} \tag{37}$$

Since $0 < \epsilon' \leq \frac{1}{3}$ and $0 \leq \alpha < 1$, the denominator of Equation (37) is greater than 1. Hence,

$$H_2 < C_2. \tag{38}$$

We now show that $\delta_2 \geq H_2$. We have

$$\begin{aligned}
\delta_2 &= \lceil \frac{P_2}{TTRT} \rceil \cdot TTRT - P_2 \\
&= \lceil \frac{3/2 - \epsilon'}{1/2} \rceil \cdot \frac{1}{2} - (\frac{3}{2} - \epsilon') = \lceil 3 - 2\epsilon' \rceil \cdot \frac{1}{2} - \frac{3}{2} + \epsilon' \\
&= \frac{3}{2} - \frac{3}{2} + \epsilon' \\
&= \epsilon' = C_2 > H_2.
\end{aligned} \tag{39}$$

16

| 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 | $H_i = \frac{C_i}{P_i} \cdot (TTRT - \tau)$ | 0 | Uses local information only. i.e., $\frac{C_i}{P_i}$. |
| Equal partition | $H_i = \frac{TTRT - \tau}{n}$ | $\frac{1-\alpha}{3n-(1-\alpha)}$ | Uses global information only, i.e., the number of nodes $n$. |
| Normalized proportional | $H_i = \frac{C_i/P_i}{U} \cdot (TTRT - \tau)$ | $\frac{1-\alpha}{3}$ | 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'.

Table 1: Summary of the synchronous capacity allocation schemes.

From Corollary 5.1. the amount of time ($X_2$) for node 2 to transmit its synchronous message in a time interval ($t, t + P_2$) is given by

$$
\begin{aligned}
X_2 &= \lfloor \frac{P_2}{TTRT} - 1 \rfloor \cdot H_2 \\
&= \lfloor \frac{(3/2) - \epsilon'}{1/2} - 1 \rfloor \cdot H_2 \\
&= 1 \cdot H_2 < C_2.
\end{aligned}
\tag{40}
$$

Therefore, the deadline constraint (10) is violated and this set of messages cannot be guaranteed. Q.E.D.

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.

# 7 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 capacity allocated to each node in the network was also shown to be a decisive factor in guaranteeing time-critical messages. In this paper, we first derived a generalized version of Johnson and Sevcik's theorem [26, 49] which gives the maximum time that can elapse between any $r$ consecutive token arrivals at some node. We then applied this result to the analysis of synchronous capacity allocation schemes. 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 1 summarizes the four allocation schemes discussed in this paper. Their Worst Case Achievable Utilizations range from 0% to 33%. 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 capacity of a node without using the information of messages on other nodes. Hence. the allocation

function of a local scheme has the form

$$H_i = f(C_i, P_i, TTRT). \tag{11}$$

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, two global schemes proposed in this paper achieve better performance than the local ones as shown in Table 1. In particular, the normalized proportional scheme has a high Worst Case Achievable Utilization of $\frac{1-a}{3}$ which is independent of the number of the nodes in the system or the message lengths and periods. To date, no synchronous capacity allocation scheme has been reported to achieve such substantial performance.

However. it is not yet known 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. Work is underway to investigate the design and implementation of such an optimal synchronous capacity allocation scheme.

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 re-adjustment of synchronous capacities over the entire network. This may not be acceptable in some situations. Because local schemes compute the synchronous capacity 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 capacity 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 capacities 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.

However. as the local allocation schemes use less information than the global ones, they may not 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 $\frac{U_i}{P_i}$ ) turned out to have a Worst Case Achievable Utilization of 0%. The problem therefore remains either to develop a local allocation scheme with non-zero Worst Case Achievable Utilization or to formally prove that all local allocation schemes have a zero Worst Case Achievable Utilization. This issue is currently under investigation.

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.

# References

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

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

[3] D. Benson, P. Ciarfella and P. Hayden, "FDDI: Meeting the interoperability challenge," *Proc. IEEE Conf. on Local Computer Networks*, pp. 253-260, Minneapolis MN. Sept. 30 - Oct. 3, 1990.

[4] L. Bergman, "Optical protocols for advanced spacecraft networks," *Proceedings of the space station evolution symposium*, Vol. 1, part 2, League city, Texas. Aug. 6-8, 1991.

[5] L. N. Bhuyan, D. Ghosal and Q. Yang, "Approximate analysis of single and multiple ring networks," *IEEE Transactions on Computers*, 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," *Proc. IEEE Conf. on Local Computer Networks*, pp. 75-83, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

[7] E. Chevers, "Advanced DMS architectures," *Proceedings of the space station evolution symposium*, Vol. 1, part 2, League city, Texas. Aug. 6-8, 1991.

[8] M. D. Cohn, "A network architecture for advanced aircraft," *Proc. IEEE Conf. on Local Computer Networks*, pp. 358-364, Minneapolis MN, Oct. 10-12, 1989.

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

[10] S. Davari and W. Zhao, "RMS aids real-time scheduling," *RICIS Review*, Vol. 3, No. 1, 1991.

[11] D. Dykeman and W. Bux, "Analysis and tuning of the FDDI media access control protocol," *Journal of Selected Area in Communications*, Vol. 6, No. 6, July 1988.

[12] S. Ginzburg, W. Mallard and D. Newman, "FDDI over unshielded twisted pairs," *Proc. IEEE Conf. on Local Computer Networks*, pp. 395-398, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

[13] P. Gonia and M. Agrawal, "A high performance OSI implementation on FDDI," *Proc. IEEE Conf. on Local Computer Networks*, pp. 301-309, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

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

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

[16] R. M. Grow, "FDDI follow-on status," *Proc. IEEE Conf. on Local Computer Networks*, pp. 45-48, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

[17] E. Hotard, "FDDI concentrator design issues," *Proc. IEEE Conf. on Local Computer Networks*, pp. 388-394, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

[18] J. Hutchison, "The role of concentrators in FDDI rings" *Proc. IEEE Conf. on Local Computer Networks*, pp. 24-40, Minneapolis MN, Oct. 10-12, 1989.

[19] IEEE/ANSI Standard 802.4 – 1985 "Token passing bus access method and physical layer specifications," *The Institute of Electrical and Electronic Engineers, Inc.*, New York, 1985.

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

[21] V. Iyer and S. P. Joshi, "FDDI's 100 M-bps protocol improves on 802.5 Spec's 4-M-bps limit," *Electrical Design News*, pp. 151-160, May 2, 1985.

[22] R. Jain, " Error characteristics of fibre distributed data interface (FDDI)," *IEEE Trans. on Commun.*, Vol. 38, No. 8, Aug. 1990.

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

[24] A. P. Jayasumana and P. N. Werahera, "Performance of fibre distributed data interface network for multiple classes of traffic," *IEEE Proceedings. Part E*, Vol. 137, No. 5, pp. 401-408, Sept. 1990.

[25] M. J. Johnson, "Reliability mechanisms of the FDDI high bandwidth token ring protocol," *Computer Networks ISDN Systems*, Vol. II, No. 2, pp. 121-131, 1986.

[26] M. J. Johnson, "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.

[27] G. W. Kajos and D. H. Hunt, "The FDDI link error rate monitor," *Proc. IEEE Conf. on Local Computer Networks*, pp. 347-357, Minneapolis MN, Oct. 10-12, 1989.

[28] R. J. Kochanski and J. L. Paige, "SAFENET The standard and its application," *IEEE LCS*, Vol. 2, No. 1, pp. 46-51, Feb. 1991.

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

19

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

[31] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real time environment," *J. ACM*, Vol. 20, No. 1, 1973, pp. 46-61.

[32] N. Malcolm and W. Zhao, "Version selection schemes for hard real-time communications," *Proc. IEEE Real-time Systems Symposium*, San Antonio, TX, Dec., 1991.

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

[34] J. Mccool, " FDDI - getting to the inside of the ring," *Data Commun.*, pp. 185-192, March 1988.

[35] MIL-HDBK-0036, "Survivable adaptable fibre optic embedded network II—SAFENET II," Draft.

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

[37] J. Ng and J. Liu, "Performance of local area network protocols for hard real-time applications," *IEEE Conf. on Distributed Computing Systems*, pp. 318-326, Arlington, Texas, May 1991.

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

[39] K. B. Ocheltree, "Using redundancy in FDDI networks," *Proc. IEEE Conf. on Local Computer Networks*, pp. 261-267, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

[40] Lt. J. L. Paige, "SAFENET - A navy approach to computer networking," *Proc. IEEE Conf. on Local Computer Networks*, pp. 268-273, Minneapolis MN, Sept. 30 - Oct. 3, 1990.

[41] J. Pang and F. A. Tobagi, "Throughput analysis of a timer controlled token passing protocol under heavy load," *IEEE Trans. on Communications*, Vol. 37, No. 7, pp. 694-702, July 1989.

[42] K. Ramamritham, "Channel characteristic in local area hard real-time system," *Computer Networks and ISDN Syst.*, Vol. 13, No. 1, Jan. 1987.

[43] F. E. Ross, "FDDI—A tutorial," *IEEE Commun. Mag.*, Vol. 24, No. 5, pp. 10-17, 1986.

[44] F. E. Ross, " Rings are round for good," *IEEE Network Mag.*, Jan. 1987.

[45] F. E. Ross, "An overview of FDDI: The fibre distributed data interface," *IEEE Journal on Sel. Areas in Comm.*, pp.1043-1051, Vol. 7, Sept. 1989.

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

[47] SAE, Aerospace Systems Division, Committee AS-2, "High speed ring bus (HSRB)," AS4074.2, Jan. 27, 1988.

[48] R. Sankar and Y. Y. Yang, "Performance analysis of FDDI," *Proc. IEEE Conf. on Local Computer Networks*, pp. 328-332, Minneapolis MN, Oct. 10-12, 1989.

[49] 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

[50] L. Sha and J. B. Goodenough, "Real-time scheduling theory and Ada*," *IEEE Computer*, April 1990., pp. 53-62.

[51] K. G. Shin and C. Hou, "Analytic evaluation of contention protocols used for real-time systems," *Proc. IEEE Real-Time Systems Symp.*, Dec 1990.

[52] R. Southard, "Fibre optics: A winning technology for LANs," *Electronics*, pp. 111-114, Feb. 1988.

[53] W. Stallings, "*Computer Communication Standards, Vol 2: Local area Network standards*," Howard W. Sams & Co., 1987.

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

[55] M. Teener and R. Gvozdanovic, "FDDI-II operation and architectures," *Proc. IEEE Conf. on Local Computer Networks*, pp. 49-61, Minneapolis MN, Oct. 10-12, 1989.

[56] F. A. Tobagi, "Multiaccess protocols and real-time communications in local area networks," *Advanced Sem. Real-time Local Area Networks*, Bandol, France, Apr. 16-18, 1986.

[57] J. F. Torgerson, "FDDI MAC services design considerations," *Proc. IEEE Conf. on Local Computer Networks*, pp. 11-18, Minneapolis MN, Oct. 10-12, 1989.

[58] 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.

[59] J. N. Ulm, "A timed token ring local area network and its performance characteristics," *Proc. Conf. Local Computer Networks*, Feb. 1982, pp. 50-56.

[60] A. Valenzano, P. Montuschi, and L. Ciminiera, "Some properties of timed token medium access protocols," *IEEE Transactions on Software Engineering*, Vol. 16, No. 8, August 1990.

[61] A. C. Weaver and R. Simoncic, "Communications for the NASA space station," *Proc. IEEE Conf. on Local Computer Networks*, pp. 333-346. Minneapolis MN. Oct. 10-12. 1989.

[62] L. Yao and W. Zhao. "Performance of an extended IEEE 802.5 protocol in hard real-time systems." *Proc. IEEE Conf. Computer Communications. INFOCOM '91*. April 1991.

[63] W. Zhao and K. Ramamritham, "Virtual time CSMA protocols for hard real-time communications." *IEEE Transactions on Software Engineering*. SE-13(8):938-952. Aug 1987.

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

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

[66] W. Zhao, J. A. Stankovic and K. Ramamritham. "A window protocol for transmission of time constrained messages," *IEEE Trans. Computers*, 39(9):1186-1203. Sept. 1990.

## 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:

$$NS_i = \{S_{i_1}, S_{i_2}, \ldots, S_{i_p}\}. \tag{42}$$

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

$$N_i = (NS_i, NA_i, \theta_i) \tag{43}$$

where $\theta_i$ is the latency between node $i$ and its upstream neighbor.

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

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

$$\text{Network} = \{N_1, N_2, \ldots, N_m\}. \tag{44}$$

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:*

$$T(N_i) = (VN_{i_1}, VN_{i_2}, \ldots, VN_{i_p}). \tag{45}$$

*where the virtual node $VN_{i_j}$ is represented as*

$$VN_{i_j} = \begin{cases} (\{S_{i_j}\}, o, \theta_{i_j}), & \text{if } 1 \leq j < p. \\ (\{S_{i_p}\}, NA_i, \theta_{i_p}), & \text{if } j = p. \end{cases} \tag{46}$$

*where*

$$\theta_{i_j} = \begin{cases} \theta_i, & \text{if } j = 1. \\ 0, & \text{if } 2 \leq j \leq p. \end{cases} \tag{47}$$

*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 $(\theta_{i_j})$ 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 $\theta_i$.*

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

$$T(N_i) = VN_i = (\{S_d\}, NA_i, \theta_i).\tag{48}$$

*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    Proofs of Protocol Timing Properties

In this appendix, the proofs of Johnson and Sevcik's theorem and the generalized Johnson and Sevcik's theorem will be presented. These theorems will be preceded by the definitions of a few terms and lemmas to be used in the their proofs.

### B.1    Definitions of Terms

- $t_i(l)$, $(l = 1, 2, \ldots)$. It is the time when the token makes its $l^{th}$ visit to node $i$.

- $R_i(l)$, $(l = 1, 2, \ldots)$. It is defined as follows:

$$R_i(l) = \begin{cases} t_i(l) + TTRT, & \text{if the token is early on its } l^{th} \text{ visit to node } i; \\ R_i(l-1) + TTRT, & \text{otherwise.} \end{cases}\tag{49}$$

That is, $R_i(l)$ indicates the 'next expected arrival time' of the token at node $i$ after the token's $l^{th}$ visit. If the token is late on its $(l+1)^{th}$ visit to node $i$, then $R_i(l)$ will be the time at which $TRT_i$ expires and is reset to TTRT. Note that the definitions of $R_i(l)$ and $t_i(l)$ imply that

$$R_i(l) - t_i(l) \leq TTRT.\tag{50}$$

- The amount of time left, before the initiation of the ring recovery process by node $i$, can be expressed as a function of two parameters at that node — the $TRT_i$ and $LC_i$. In order to simplify our proofs, we define a single parameter $TRT_i'$, capturing the values of both $TRT_i$ and $LC_i$ within it, to indicate the amount of time left before the initiation of ring recovery process by node $i$. $TRT_i'$ is formally defined as follows:

$$TRT_i' = TRT_i + (1 - LC_i) \cdot TTRT.\tag{51}$$

23

Given the fact that $0 \leq TRT_i \leq TTRT$ and $0 \leq LC_i \leq 1$, it is clear that

$$0 \leq TRT_i' \leq 2 \cdot TTRT. \tag{52}$$

The physical meaning of $TRT_i'$ is that when $TRT_i' > TTRT$, the $TRT_i$ has not expired since the last token arrival. A token arriving at this instant would be early. When $TRT_i' \leq TTRT$, $LC_i = 1$. Hence, the $TRT_i$ has expired once since the last token arrival. In either case, the amount of time left before node $i$ initiates the ring recovery process is $TRT_i'$. In the event that $TRT_i'$ becomes zero, the ring recovery process will be initiated.

## B.1 Proofs of Theorems 5.1 and 5.2 and Corollary 5.1

In the proofs of the lemmas and theorems that follow, $TRT_i(t)$, $THT_i(t)$, $LC_i(t)$, $TRT_i'(t)$ represent the values of $TRT_i$, $THT_i$, $LC_i$ and $TRT_i'$ at time $t$.

**LEMMA B.1**      *For any integers $l > 0$, $v > 0$ and any node $i$ $(1 \leq i \leq n)$.*

$$R_i(l + v) - R_i(l) \quad \leq \quad v \cdot TTRT. \tag{53}$$

*The equality holds if the token arrival is late on its $l^{th}$, $(l+1)^{th}, \ldots,$ and $(l + v - 1)^{th}$ visits to node $i$.*

The lemma can be easily proved by an induction argument.

**LEMMA B.2** *(Johnson's Lemma [26]) After ring initialization, the $TRT'$ values of all operational stations will be monotonically increasing in the downlink direction, up to and including the station which last received the token.*[5]

The reader is referred to [26] for the proof of the above lemma.

Let us consider an example to illustrate the implication of the above lemma. Figure 3 shows the token is leaving node $A$ and is enroute to node $B$ at some time $t$. By Johnson's Lemma, the values of $TRT$'s are monotonically increasing in the downlink direction upto and including the station which last received the token (i.e., node $A$). Therefore,

$$TRT_B' < TRT_C' < TRT_D' < TRT_E' < \ldots < TRT_A'. \tag{54}$$

Now, if the $TRT_E$ expires at this moment (i.e., $TRT_E' \leq TTRT$), then the $TRT$s of nodes $B, C, D$ will have also expired. Consequently, the token will be *late* when it visits nodes $B, C, D$ and $E$ for the first time after time $t$.

**LEMMA B.3**      *For any $l > 0$ and any node $j$, if the token is late on its $(l+1)^{th}$ visit at node $j$, then*

$$t_j(l + 1) \quad \leq \quad R_j(l) + TTRT - H_j. \tag{55}$$

---

This result is known as the 'TRT alignment' in [26]. By the FDDI MAC standard, the ring initialization phase aligns the $TRT'$ values. That is, the $TRT'$ values monotonically increase in the downlink direction of the ring. The proof of Johnson's Lemma assumes that this alignment holds during normal ring operation and so do we.

**Proof:** Let us first define the phrase "token is at node $i$" to mean that the token is being held by node $i$ or is on its way to node $i$ from its upstream neighbor (node $i-1$ if $i > 1$, else node $n$ if $i = 1$).

Recall that $TRT_i'(t)$ is defined as

$$TRT_i'(t) = TRT_i(t) + (1 - LC_i(t)) \cdot TTRT. \tag{56}$$

If the token is late on its $(l+1)^{th}$ visit to node $j$, the token must be at some node $i$ at time $R_j(l)$. Assume that the token arrives at node $i$ at time $T$ ($T \le R_j(l)$). We have two cases to consider.

**Case 1: The token arrives late at node $i$.** In this case node $i$ will only transmit its synchronous messages for at most $H_i$ time. Hence the token will leave node $i$ no later than $T + H_i + \theta_i \le R_j(l) + H_i + \theta_i$.

**Case 2: The token arrives early at node $i$.** Therefore, $LC_i(T) = LC_j(T) = 0$. By Johnson's Lemma (Lemma B.2) we have

$$TRT_i'(T) - TTRT \le TRT_j'(T) - TTRT. \tag{57}$$

From (51), we get

$$TRT_i(T) \le TRT_j(T). \tag{58}$$

Thus,

$$THT_i(T) = TRT_i(T) \le TRT_j(T) = R_j(l) - T. \tag{59}$$

Hence, node $i$ can transmit asynchronous messages for at most $R_j(l) - T$ time and transmit synchronous messages for at most $H_i$ time. In this case too, the token will leave node $i$ before

$$T + (R_j(l) - T + H_i) + \theta_i = R_j(l) + H_i + \theta_i. \tag{60}$$

That is, if the token is at node $i$ when $TRT_j$ expires, then the token will leave node $i$ no later than $R_j(l) + H_i + \theta_i$.

Now consider the nodes on the way from node $i$ to node $j$. Let them be labeled as $n_1, n_2, \ldots, n_k$. According to Johnson's Lemma, the token will be late on its visit to each of these $k$ nodes on the way to node $j$. Hence, these nodes will transmit their synchronous messages only. That is, the token will arrive at node $j$ no later than

$$R_j(l) + H_i + \theta_i + \sum_{h=1}^{k} (H_{n_h} + \theta_{n_h}) + \theta_j + \Delta \le R_j(l) + \sum_{h \neq j} H_h + \sum_{h=1}^{n} \theta_h + \Delta$$
$$= R_j(l) + \sum_{h \neq j} H_h + \tau \tag{61}$$

where $\Delta$ represents the protocol dependent overheads. By (9), we have

$$t_j(l+1) \le R_j(l) + TTRT - H_j. \tag{62}$$

Q.E.D.

25

**THEOREM 5.1** *(Johnson and Sevcik's Theorem [26, 49])* For any integer $l > 0$ and any node $j$ $(1 \leq j \leq n)$.

$$t_j(l+1) - t_j(l) \leq 2 \cdot TTRT - H_j.$$ (63)

**Proof:** If the token is not late at its $(l+1)^{th}$ visit to node $j$, then

$$t_j(l+1) - t_j(l) \leq TTRT \leq 2 \cdot TTRT - H_j$$ (64)

Otherwise, from Lemma B.3 we have

$$\begin{aligned}
t_j(l+1) - t_j(l) &\leq R_j(l) + TTRT - H_j - t_j(l) \\
&\leq TTRT - H_j + (R_j(l) - t_j(l)) \\
&\leq 2 \cdot TTRT - H_j \qquad \text{(by (50)).}
\end{aligned}$$ (65)

This simply says that the maximum time that can elapse between two consecutive token arrivals at some node is bounded by $2 \cdot TTRT - H_j$. This result was first proved in [26]. Q.E.D.

**THEOREM 5.2** *(Generalized Johnson and Sevcik's Theorem)* For any integer $l > 0$, $v > 1$ and any node $j$ $(1 \leq j \leq n)$.

$$t_j(l+v-1) - t_j(l) \leq v \cdot TTRT - H_j.$$ (66)

**Proof:** We prove the theorem by induction on $v$. For $v = 2$, by Theorem 5.1 we have

$$t_j(l+1) - t_j(l) \leq 2 \cdot TTRT - H_j.$$ (67)

Hence, the theorem holds for $v = 2$.

Assume that for $v = k$, (66) holds, i.e.,

$$t_j(l+k-1) - t_j(l) \leq k \cdot TTRT - H_j.$$ (68)

Now we consider for $v = k + 1$. We have two cases:

**Case 1: The token arrives early on its v'th visit to node j $(1 < v' < v)$.** That is,

$$R_j(v'-1) \geq t_j(v').$$ (69)

Hence,

$$\begin{aligned}
t_j(v') - t_j(l) &= (t_j(v') - R_j(l)) + (R_j(l) - t_j(l)) \\
&\leq (R_j(v'-1) - R_j(l)) + (R_j(l) - t_j(l)) \qquad \text{(by (69)).}
\end{aligned}$$ (70)

By Lemma B.1 and (50), we have

$$t_j(v') - t_j(l) \leq (v'-1-l) \cdot TTRT + TTRT = (v'-l) \cdot TTRT.$$ (71)

Now

$$t_j(l+(k+1)-1) - t_j(v') = t_j(v' + (l - v' + k + 1) - 1) - t_j(v').$$ (72)

26

By the induction hypothesis (68) and (72), we have

$$t_j(l + (k+1) - 1) - t_j(v') \leq (l - v' + k + 1) \cdot TTRT - H_j. \tag{73}$$

Adding (71) and (73) on both sides, we have

$$
\begin{aligned}
t_j(l + (k+1) - 1) - t_j(l) &\leq (v' - l) \cdot TTRT - H_j + (l - v' + k + 1) \cdot TTRT \\
&= (k+1) \cdot TTRT - H_j. \tag{74}
\end{aligned}
$$

Thus, the theorem is proved.

## Case 2: The token is always late at node j between the $(l+1)^{\text{th}}$ visit and the $(l+k)^{\text{th}}$ visit inclusive.

Because the token is late, from Lemma B.3, we have

$$t_j(l + k) = t_j(l + (k+1) - 1) \leq R_j(l + k - 1) + TTRT - H_j. \tag{75}$$

Therefore, by (50) and (75),

$$
\begin{aligned}
t_j(l + (k+1) - 1) - t_j(l) &= (t_j(l + (k+1) - 1) - R_j(l)) + (R_j(l) - t_j(l)) \\
&\leq (R_j(l + k - 1) + TTRT - H_j - R_j(l)) + TTRT \\
&= R_j(l + k - 1) - R_j(l) + 2 \cdot TTRT - H_j \\
&= (k - 1) \cdot TTRT + 2 \cdot TTRT - H_j \qquad \text{(by Lemma B.1)} \\
&= (k + 1) \cdot TTRT - H_j. \tag{76}
\end{aligned}
$$

This concludes the proof of the theorem.

**COROLLARY 5.1** *Assume that at time $t$, a synchronous message with period $P_i$ arrives at node $i$ ($1 \leq i \leq n$). Then, in the time interval $(t, t + P_i)$ the total amount of time $(X_i)$ available for node $i$ to transmit this synchronous message is bounded by*

$$X_i \geq \lfloor \frac{P_i}{TTRT} - 1 \rfloor \cdot H_i. \tag{77}$$

*In the worst case, the lower bound will be tight if*

$$\delta_i \geq H_i. \tag{78}$$

*where $\delta_i = \lceil \frac{P_i}{TTRT} \rceil \cdot TTRT - P_i$.*

**Proof:** Let $t_i(l + 1)$ be the first time the token arrives after the message's arrival at time $t$. Therefore, $t_i(l) < t$. With (4), $P_i$ can be represented as

$$P_i = m_i \cdot TTRT - \delta_i. \tag{79}$$

where $m_i = \lceil \frac{P_i}{TTRT} \rceil$ and $0 \leq \delta_i < TTRT$. We have two cases to consider:

27

**Case 1:** $0 < \delta_i < TTRT$: This implies

$$
\begin{aligned}
P_i &> (m_i - 1) \cdot TTRT \\
&= (m_i - 1) \cdot TTRT - H_i + H_i \\
&\geq t_i(l + m_i - 2) - t_i(l) + H_i &&\text{(by Theorem 5.2)} \\
&\geq t_i(l + m_i - 2) - t + H_i. &&\text{(since } t_i(l) < t) \quad (80)
\end{aligned}
$$

This means

$$
t + P_i \geq t_i(l + m_i - 2) + H_i. \tag{81}
$$

Hence, by the end of the message's period (i.e., $t + P_i$) the token will have made $m_i - 2$ visits to node $i$ since time $t$. In each of these visits, node $i$ can transmit its synchronous message for the allocated synchronous capacity $H_i$. Consequently, the total amount of the time for node $i$ to transmit this synchronous message will be at least[6]

$$
(m_i - 2) \cdot H_i = \left\lfloor \frac{P_i}{TTRT} - 1 \right\rfloor \cdot H_i. \quad \text{(since } (m_i - 1) \cdot TTRT < P_i < m_i \cdot TTRT) \tag{82}
$$

**Case 2:** $\delta_i = 0$: This implies that

$$
\begin{aligned}
P_i &= m_i \cdot TTRT \\
&= m_i \cdot TTRT - H_i + H_i \\
&\geq t_i(l + m_i - 1) - t_i(l) + H_i &&\text{(by Theorem 5.2)} \\
&\geq t_i(l + m_i - 1) - t + H_i. &&\text{(since } t_i(l) < t) \quad (83)
\end{aligned}
$$

This means

$$
t + P_i \geq t_i(l + m_i - 1) + H_i. \tag{84}
$$

Hence, by the end of the message's period (i.e., $t + P_i$) the token will have made $m_i - 1$ visits to node $i$ since time $t$. In each of these visits, node $i$ can transmit its synchronous message for the allocated synchronous capacity $H_i$. Consequently, the total amount of the time for node $i$ to transmit this synchronous message will be at least

$$
\begin{aligned}
(m_i - 1) \cdot H_i &= \left\lceil \frac{P_i}{TTRT} - 1 \right\rceil \cdot H_i \\
&= \left\lfloor \frac{P_i}{TTRT} - 1 \right\rfloor \cdot H_i \quad \text{(since } \frac{P_i}{TTRT} \text{ is an integral number).} \tag{85}
\end{aligned}
$$

From Cases 1 and 2, we see that (77) holds.

Now consider the case when (78) holds, i.e., $\delta_i \geq H_i$.

$$
\delta_i = m_i \cdot TTRT - P_i \geq H_i. \tag{86}
$$

Recall that $t_i(l + 1)$ is the first time the token arrives after the message's arrival at time $t$. Let $t = t_i(l) + \epsilon$ ($\epsilon > 0$). This implies in the worst case

$$
\begin{aligned}
P_i &\leq m_i \cdot TTRT - H_i \\
&= t_i(l + m_i - 1) - t_i(l) &&\text{(by Theorem 5.2)} \\
&= t_i(l + m_i - 1) - t + \epsilon. \tag{87}
\end{aligned}
$$

---

[6]Note that since $\delta_i > 0$, $m_i \geq 3$.

28

Consequently.

$$t + P_i \leq t_i(l + m_i - 1) + \epsilon. \tag{88}$$

Because of the arbitrarity of $t$. we can let $t - t_i(l)^+$. That is. $\epsilon - 0$. Therefore,

$$t + P_i \leq t_i(l + m_i - 1). \tag{89}$$

The above inequality indicates that the $(l + m_i - 1)^{th}$ visit of the token will not be earlier than $t + P_i$. Hence. in the time interval $(t. t + P_i)$. the node will have no more than $m_i - 2 \ (= \lfloor \frac{P_i}{TTRT} - 1 \rfloor)$ visits of the token. Thus. the lower bound of $X_i$ is tight. Q.E.D.

# Appendix C  Proofs of Lemma 6.1 and Theorem 6.3

## C.1  Proof of Lemma 6.1

**LEMMA 6.1** *For any synchronous message stream $i$ $(1 \leq i \leq n)$ we have*

$$\frac{\lfloor \frac{P_i}{TTRT} - 1 \rfloor}{P_i/TTRT} \geq \frac{1}{3 - \frac{s}{TTRT}} \geq \frac{1}{3}. \tag{90}$$

**Proof:**  From (4). we have

$$P_i = m_i \cdot TTRT - \delta_i. \tag{91}$$

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

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

$$\frac{\lfloor P_i/TTRT - 1 \rfloor}{P_i/TTRT} = \frac{\lfloor \frac{(m_i \cdot TTRT - \delta_i)}{TTRT} - 1 \rfloor}{(m_i \cdot TTRT - \delta_i)/TTRT}$$

$$= \frac{\lfloor m_i - \frac{s_i}{TTRT} - 1 \rfloor}{m_i - \frac{s_i}{TTRT}} = \frac{m_i - 2}{m_i - \frac{s_i}{TTRT}}. \tag{92}$$

Note that the right hand side of (92) is an increasing function of $m_i$.[*] Therefore. the minimum value of (92) is obtained by substituting the minimum value of $m_i$. i.e.. $m_i = 3$. Hence

$$\frac{\lfloor P_i/TTRT - 1 \rfloor}{P_i/TTRT} = \frac{m_i - 2}{m_i - \frac{s_i}{TTRT}} \geq \frac{1}{3 - \frac{s_i}{TTRT}}. \tag{94}$$

Further. the right hand side of (94) is an increasing function of $\delta_i$. If we let $\delta_i - 0^+$. we have

$$\frac{\lfloor P_i/TTRT - 1 \rfloor}{P_i/TTRT} \geq \frac{1}{3 - \frac{s_i}{TTRT}} \geq \frac{1}{3}. \tag{95}$$

Thus. the lemma holds in this case.

---

[*] If $f$ represents the right hand expression in (92). then $f$ is an increasing function of $m_i$. since

$$\frac{df}{dm_i} = \frac{2 - \delta_i/TTRT}{(m_i - \delta_i/TTRT)^2} > 0. \qquad (\text{since } \frac{s_i}{TTRT} \leq 1 \text{ and } m_i \geq 2) \tag{93}$$

**Case 2:** $\delta_i = 0$. This implies $P_i = m_i \cdot TTRT$. We have

$$\frac{\lfloor P_i/TTRT - 1 \rfloor}{P_i/TTRT} = \frac{m_i - 1}{m_i} \geq \frac{1}{2}. \tag{96}$$

From (95) and (96). we get

$$\frac{\lfloor P_i/TTRT - 1 \rfloor}{P_i/TTRT} \geq \frac{1}{2} \geq \frac{1}{3 - \frac{\delta_i}{TTRT}} \geq \frac{1}{3}. \tag{97}$$

Q.E.D.

## C.2 Proof of Theorem 6.3

In this subsection, a proof of theorem 6.3 is presented. We need to prove a lemma first.

**LEMMA C.1**   *Assume that we have two message sets $M$ and $M'$ where their utilization factors are equal. i.e.. $U(M) = U(M')$. Further. assume that the synchronous capacity allocated to all the nodes is the same irrespective of the message set considered i.e.. for $i = 1 \ldots n$.*

$$H_i' = H_i. \tag{98}$$

*The first message set is arbitrary. That is.*

$$M = \{(C_1. P_1) \ldots (C_i. P_i) \ldots (C_n. P_n)\}. \tag{99}$$

*By (4). a message period $P_i$ is of the form*

$$P_i = m_i \cdot TTRT - \delta_i. \tag{100}$$

*where $m_i = \lceil \frac{P_i}{TTRT} \rceil$ and $0 \leq \delta_i < TTRT$.*

*The second message set $M'$ is of the form*

$$M' = \{(C_1'. P_1') \ldots (C_i'. P_i') \ldots (C_n'. P_n')\}. \tag{101}$$

*The parameters of the messages in $M'$ depend on those in $M$ and $H_i$ as follows:*

$$P_i' = \begin{cases} P_i. & \text{if } m_i = 2: \\ m_i \cdot TTRT - H_i. & \text{if } m_i \geq 3: \end{cases} \tag{102}$$

*and*

$$C_i' = \begin{cases} C_i. & \text{if } m_i = 2: \\ P_i' \cdot \frac{C_i}{P_i}. & \text{if } m_i \geq 3. \end{cases} \tag{103}$$

*Given the above conditions. if the deadline constraint of message set $M'$ is satisfied. then the deadline constraint of message set $M$ is also satisfied.*

**Proof:**   Based on the values of $m_i$. $P_i$. and $P_i'$. we have three cases to consider:

- $m_i = 2$.

30

- $m_i \geq 3$ and $P_i \leq P_i'$, and

- $m_i \geq 3$ and $P_i > P_i'$.

In each of these three cases we show that if there is sufficient time to successfully transmit message $(C_i'. P_i')$ in message set $M'$. then the time available is also sufficient to transmit message $(C_i'. P_i')$ in $M$.

**Case 1:** $m_i = 2$. By (102) and (103). we have

$$P_i' = P_i. \; C_i' = C_i. \; \text{and} \; H_i' = H_i. \tag{104}$$

Therefore. the lower bound on the time available to transmit both $(C_i. P_i)$ and $(C_i'. P_i')$ will be the same. Since the deadline constraint is not violated when transmitting any message in message set $M'$. the deadline constraint will not be violated when transmitting a message $(C_i. P_i)$ $(= (C_i'. P_i'))$ in message set $M$ either.

**Case 2:** $m_i \geq 3$ and $P_i \leq P_i'$. By (102). we have

$$\delta_i \geq \delta_i' = H_i' > 0. \tag{105}$$

From (103). we have

$$C_i' = P_i' \cdot \frac{C_i}{P_i} = C_i \cdot \frac{P_i'}{P_i} \geq C_i. \tag{106}$$

Using Corollary 5.1. we claim that the lower bound on the time available to transmit either message $(C_i. P_i)$ or $(C_i'. P_i')$ during their respective message periods $P_i$ and $P_i'$ is the same. This is because $X_i = X_i' = (m_i - 2) \cdot H_i = (m_i - 2) \cdot H_i'$. Since this amount of time is sufficient to transmit a message of length $C_i'$. the message with length $C_i$ can also be transmitted before the end of period $P_i$. That is. the deadline constraint of messages in this case is met.

**Case 3:** $m_i \geq 3$ and $P_i > P_i'$. Let

$$P_i = P_i' + \theta \qquad (0 < \theta < H_i'). \tag{107}$$

From (103) and (107). we have

$$C_i = C_i'(1 + \frac{\theta}{P_i'}). \tag{108}$$

Now. as seen in the proof of Corollary 5.1 the $(m-1)^{th}$ token arrival at node $i$. in the worst case. occurs at the end of the period $P_i' = m_i \cdot TTRT - H_i$. Hence. when $P_i = P_i' + \theta$. the node $i$ can transmit additional synchronous messages for a time $\theta$. That is. the maximum amount (say $X_i$) of time available to transmit synchronous messages within period $P_i$. in the worst case. is $C_i' + \theta$. Therefore.

$$
\begin{aligned}
X_i = C_i' + \theta & \geq C_i' + \frac{C_i'}{P_i'}\theta \qquad (\text{ since } \frac{C_i'}{P_i'} \leq 1) \\
& = C_i'(1 + \frac{\theta}{P_i'}) \\
& = C_i. \tag{109}
\end{aligned}
$$

From (109) we see that there will be sufficient time available for $C_i$ to be transmitted within period $P_i$. That is. the deadline constraint of the message set $M$ is satisfied in this case too.

From the results of the above cases, we see that the deadline constraint of message set $M$ is satisfied if the deadline constraint of message set $M'$ is satisfied. Q.E.D.

We are now ready to prove Theorem 6.3.

**THEOREM 6.3** *The Worst Case Achievable Utilization of the equal partition synchronous capacity allocation scheme is $\frac{1}{3n-(1-\alpha)} \cdot (1-\alpha)$ where $\alpha = \frac{\tau}{TTRT}$ and $n$ is the number of nodes.*

**Proof:** We prove the theorem by showing that the following statements are true:

1. For any message set $M$. the protocol constraint will always be satisfied.

2. For any message set $M$ with utilization factor $U(M) \leq \frac{1}{3n-(1-\alpha)} \cdot (1-\alpha)$, the deadline constraint will be satisfied.

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

**Proof of Statement 1:** From (26). we have

$$\sum_{i=1}^{n} H_i = n \cdot \frac{TTRT - \tau}{n} = TTRT - \tau. \tag{110}$$

That is. the protocol constraint is always satisfied.

**Proof of Statement 2:** Consider a message set $M = \{(C_1, P_1)\ldots(C_i, P_i)\ldots(C_n, P_n)\}$ with utilization factor

$$U \leq \frac{1}{3n - (1 - \alpha)} \cdot (1 - \alpha) \tag{111}$$

where $n$ is the number of nodes and $\alpha = \frac{\tau}{TTRT}$. Further. any period $P_i$ in message set $M$ can be expressed in the form given by (4). That is.

$$P_i = m_i \cdot TTRT - \delta_i \tag{112}$$

where $m_i = \lceil \frac{P_i}{TTRT} \rceil \geq 2$ and $0 \leq \delta_i < TTRT$. Now construct a message set $M' = \{(C'_1, P'_1)\ldots(C'_i, P'_i)\ldots(C'_n, P'_n)\}$ where

$$P'_i = \begin{cases} P_i. & \text{if } m_i = 2; \\ m_i \cdot TTRT - H'_i. & \text{if } m_i \geq 3; \end{cases} \tag{113}$$

and

$$C'_i = \begin{cases} C_i. & \text{if } m_i = 2; \\ P'_i \cdot \frac{C_i}{P_i}. & \text{if } m_i \geq 3. \end{cases} \tag{114}$$

It is easy to verify that the utilization factors are equal. i.e.. $U(M') = U(M)$. Given the equal partition scheme. for $1 \leq i \leq n$ we have $H_i = H'_i$. By Lemma C.1. if the deadline constraint of message set $M'$ can be satisfied. then the deadline constraint of message set $M$ will also be satisfied. We now show that the deadline constraint of message set $M'$ is satisfied.

32

**Case 1:** We first consider the messages whose periods are given by

$$P_i' = m_i \cdot TTRT - H_i' \tag{115}$$

where $m_i \geq 3$.

Multiplying both sides of (111) by $C_i''/U$, we have

$$
\begin{aligned}
C_i'' &\leq \frac{1}{3n - (1 - \alpha)} \cdot \frac{C_i''}{U} \cdot (1 - \alpha) \quad, \\
&\leq \frac{1}{3n - (1 - \alpha)} \cdot \frac{C_i''}{C_i''/P_i'} \cdot (1 - \alpha) \\
&= \frac{1}{3n - (1 - \alpha)} \cdot P_i' \cdot (1 - \alpha) = \frac{1}{3 - \frac{1-\alpha}{n}} \cdot \frac{1}{n} \cdot P_i' \cdot (1 - \alpha).
\end{aligned} \tag{116}
$$

Substituting $\frac{1-\alpha}{n}$ with $\frac{H_i}{TTRT}$ (since $H_i' = H_i = \frac{TTRT \cdot (1-\alpha)}{n}$), we obtain

$$C_i'' \leq \frac{1}{3 - \frac{H_i}{TTRT}} \cdot \frac{1}{n} \cdot P_i' \cdot (1 - \alpha). \tag{117}$$

From Lemma 6.1 and (113) we have

$$\frac{1}{3 - \frac{H_i}{TTRT}} \leq \frac{\lfloor P_i'/TTRT - 1 \rfloor}{P_i'/TTRT}. \tag{118}$$

Therefore, we can rewrite (117) as

$$
\begin{aligned}
C_i'' &\leq \frac{\lfloor P_i'/TTRT - 1 \rfloor}{P_i'/TTRT} \cdot \frac{1}{n} \cdot P_i' \cdot (1 - \alpha) \\
&= \lfloor \frac{P_i'}{TTRT} - 1 \rfloor \cdot \frac{TTRT}{n} \cdot (1 - \alpha) \\
&= \lfloor \frac{P_i'}{TTRT} - 1 \rfloor \cdot H_i. 
\end{aligned} \tag{119}
$$

Because of Corollary 5.1, inequality (119) shows that the deadline constraint of the messages in this case is always satisfied.

**Case 2:** We now consider those messages whose periods are given by

$$P_i' = P_i = 2 \cdot TTRT. \tag{120}$$

By (111), we have

$$U \leq \frac{1 - \alpha}{3n - (1 - \alpha)} \leq \frac{1 - \alpha}{2n} \tag{121}$$

where $n \geq 1$.

Multiplying both sides of (121) by $\frac{C_i''}{P_i' U}$, we have

$$\frac{C_i''}{P_i'} \leq \frac{1 - \alpha}{2n} \cdot \frac{C_i''/P_i'}{U} \leq \frac{1 - \alpha}{2n} \qquad \left(\text{since } \frac{C_i''}{P_i'} \leq U\right). \tag{122}$$

Thus

$$
\begin{aligned}
C_i' &\le P_i' \cdot \frac{1 - \alpha}{2n} \\
&= \frac{TTRT \cdot (1 - \alpha)}{n} \qquad \text{(by (120))} \\
&= H_i' \\
&= \lfloor \frac{2 \cdot TTRT}{TTRT} - 1 \rfloor \cdot H_i' \\
&= \lfloor \frac{P_i'}{TTRT} - 1 \rfloor \cdot H_i'.
\end{aligned} \tag{123}
$$

Because of Corollary 5.1, (123) implies that the deadline constraint of message set $M'$ is satisfied.

From Cases 1 and 2, we see that the deadline constraint of message set $M'$ is satisfied when $U(M') \le \frac{1-\alpha}{3n-(1-\alpha)}$. By Lemma C.1, the deadline constraint of message set $M$ is also satisfied.

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

Let $TTRT = P_{min}/2 = 1/2$ and $n$ be the number of nodes. For any given $\epsilon > 0$, consider the following message set:

$$
\begin{aligned}
C_1 &= \epsilon/3, & P_1 &= 1, \\
C_2 &= \frac{1-\alpha}{2n} + \frac{(\epsilon/3)(3n-(1-\alpha))}{2n}, & P_2 &= \frac{3}{2} - \frac{1-\alpha}{2n}, \\
C_i &= \epsilon/3, & P_i &= (n-2), \qquad for\ i=3...n.
\end{aligned} \tag{124}
$$

The total utilization factor of the above message set is

$$
\begin{aligned}
U &= \sum_{i=1}^{n} \frac{C_i}{P_i} = (\frac{\epsilon}{3}) + (\frac{\frac{1-\alpha}{2n} + \frac{(\epsilon/3)(3n-(1-\alpha))}{2n}}{\frac{3}{2} - \frac{1-\alpha}{2n}}) + (\sum_{i=3}^{n} \frac{\epsilon/3}{n-2}) \\
&= (\frac{\epsilon}{3}) + (\frac{1-\alpha}{3n-(1-\alpha)} + \frac{\epsilon}{3}) + (\frac{\epsilon}{3}) \\
&= \frac{1-\alpha}{3n-(1-\alpha)} + \epsilon.
\end{aligned} \tag{125}
$$

We now show that $\delta_2 \ge H_2$. We have,

$$
\begin{aligned}
\delta_2 = \lceil \frac{P_2}{TTRT} \rceil \cdot TTRT - P_2 &= \lceil \frac{3/2 - (1-\alpha)/2n}{1/2} \rceil \cdot \frac{1}{2} - (\frac{3}{2} - \frac{1-\alpha}{2n}) \\
&= \lceil 3 - (1-\alpha)/n \rceil \cdot \frac{1}{2} - \frac{3}{2} + \frac{1-\alpha}{2n} \\
&= \frac{3}{2} - \frac{3}{2} + \frac{1-\alpha}{2n} \\
&= \frac{1-\alpha}{2n} = \frac{TTRT \cdot (1-\alpha)}{n} \ge H_2.
\end{aligned} \tag{126}
$$

That is, (14) of Corollary 5.1 holds. Therefore, the amount ($X_2$) of synchronous messages that can be transmitted by node 2, in the worst case, within period $P_2$, is given by equation (13).

$$
X_2 = \lfloor \frac{P_2}{TTRT} - 1 \rfloor \cdot H_2
$$

$$= \left\lfloor \frac{(3/2) - ((1-\alpha)/2n)}{1/2} - 1 \right\rfloor \cdot \frac{1-\alpha}{2n}$$

$$= \left\lfloor 3 - \frac{1-\alpha}{n} - 1 \right\rfloor \cdot \frac{1-\alpha}{2n}$$

$$= \frac{1-\alpha}{2n} < C_2. \tag{127}$$

We see that the deadline constraint has been violated when the utilization factor of the message set is greater than $\frac{1-\alpha}{3n-(1-\alpha)}$. Q.E.D.
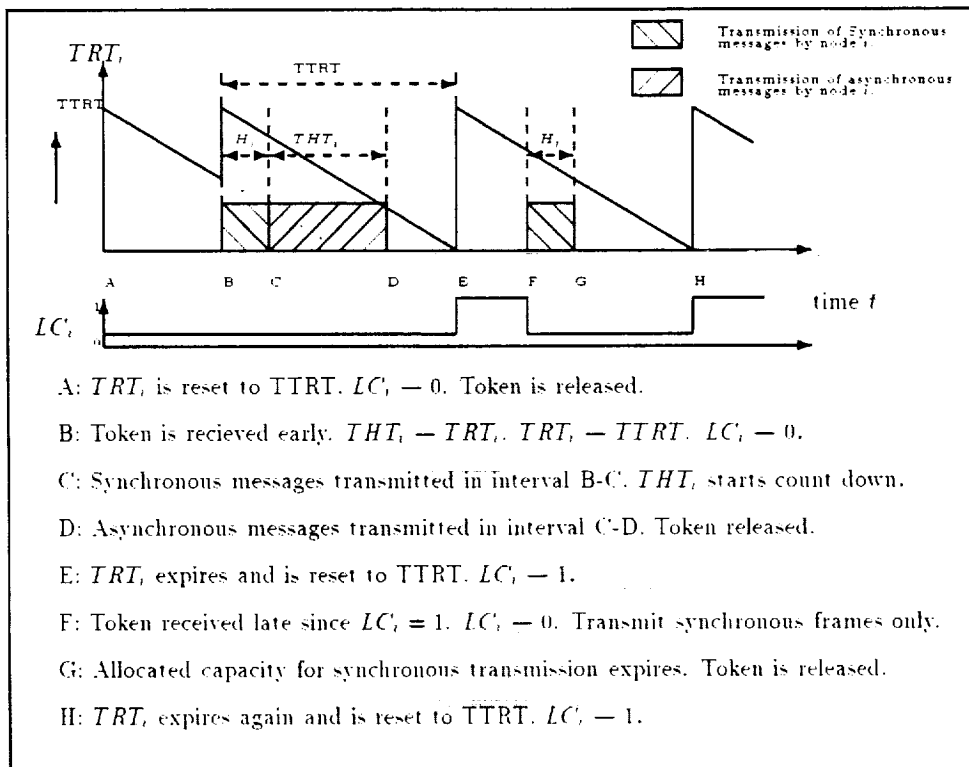
Figure 1: An Example of $TRT_i$ and $LC_i$ versus Time $t$

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

B: Token is recieved early. $THT_i - TRT_i$. $TRT_i - TTRT$. $LC_i - 0$.

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. $LC_i - 1$.

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

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

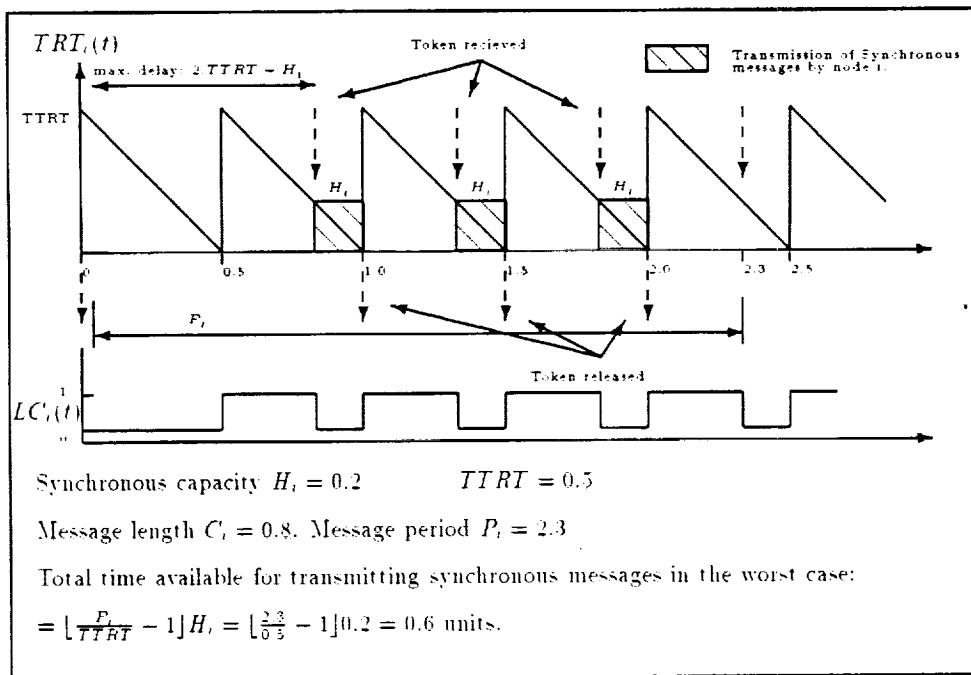H: $TRT_i$ expires again and is reset to TTRT. $LC_i - 1$.
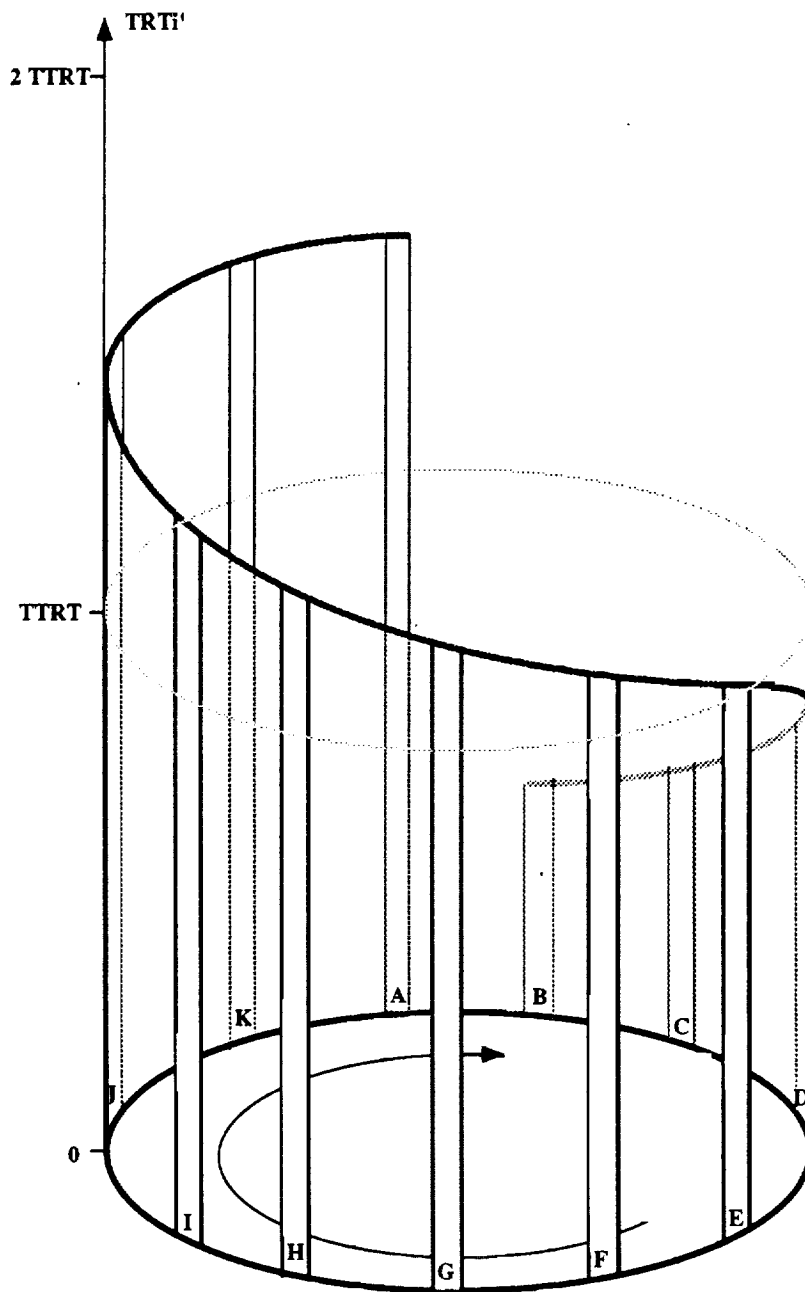
Figure 2: Token Arrivals at Node $i$ in a Worst Case Situation

Figure 3: An Illustration of TRT Alignment