GuST: Guaranteed Services Token Protocol for Real-Time Communications

Simeon Ozuomba[1], Gloria A. Chukwudebe[2]

[1]Department of Electrical/Electronic and Computer Engineering, University of Uyo, Akwa Ibom, Nigeria

[2]Department of Electrical and Electronic Engineering and Engineering Technology, Federal University of Technology Owerri, Nigeria

Corresponding email: simeonoz@yahoo.com, gloriachukwudebe@yahoo.com

Abstract: In this paper, Guaranteed Services Token (GuST) protocol for integrated services networks which can efficiently support diverse traffic consisting of hard and soft real-time traffic along with non-real-time traffic is proposed. This is to meet the increasing demand for better performance of real time communications required by distributed multimedia applications, process control, factory automation, etc.

For some time now, timed-token protocols have become the preferred Medium Access Control (MAC) protocol for supporting modern real-time systems. However, the existing timed-token protocols have been studied, and inefficiencies discovered with the way asynchronous traffic is handled. GuST employs the timed-token mechanisms in the Timely-Token protocol along with that of Budget Sharing Token (BuST) protocol. We discussed some bounds on the behavior of GuST protocol. In particular, we show that the token is never late, and the transmission of asynchronous traffic is guaranteed. We also compared GuST protocol against the Timely-Token protocol and the BuST protocol. Our comparison focuses on the ability of those protocols to support synchronous and asynchronous traffic. We demonstrated that the performance achieved by GuST is better than Timely-Toke n and BuST protocols especially for a system with light load of real-time traffic but with heavy load of non-real-time traffic. GuST protocol can be incorporated into the Ethernet network to provide real-time performance guarantees to multimedia applications. It can also be used to improve on the throughput of the Profibus which is a Fieldbus network standard.

Keywords: Timed-Token Protocol, Ethernet, Timely-Token Protocol, Budget Sharing Token Protocol, Integrated Services Networks, Real-Time Traffic, Non-Real-Time Traffic, Media Access Control (MAC), GuST: Guaranteed Services Token protocol.

1. Introduction

Nowadays, there is a rapid advent and advancements of many new and exciting applications: image processing and transmission, multimedia communications, office and factory automation, embedded real-time distributed systems, space vehicle systems, and the integration of expert systems into avionics and industrial process controls. The situation has placed an increasing demand for effective and efficient integrated services local area networks. Such networks' MAC protocols must deal with different traffic patterns and must provide not only bounded message transmission time, as required by the hard real-time tasks, but also high throughput, as demanded by soft real-time and other non-real-time tasks [1]. An attractive approach for integrating such traffic is the timed-token protocol. Consequently, the timed-token protocol has been incorporated into several high-bandwidth network standards [2], including IEEE802.4 Token Bus [3], FDDI [4][5][6][7][8], SAFENET [9], Manufacturing Automation Protocol (MAP) [10], High-Speed Ring Bus [11] and in PROFIBUS which is a Fieldbus network standard [12].

Electricity is obtained from the PV array most efficiently during sunny daytime hours [1-4]. At night or during cloudy periods, independent power systems use storage batteries to supply electricity. With grid interactive systems the grid acts as the battery, supplying electricity when the PV array cannot [5]. Energy storage devices (e.g. batteries) have been avoided in this work, to reduce capital, operation, and maintenance costs. The grid connected PV system is well known in various parts of world, and several technologies are available [6]. This research work focuses on the development of a grid connected pv system. Additionally, there have been efforts to develop the power electronics circuitry involved [7-9] and several types of inverters have been designed [10-15]. Overall, the goal is to measure the potential of a grid connected PV system in the Birbhum district of West Bengal using a solar-meter and to establish a demonstration of this type of system using existing methodologies and available equipment.

FDDI timed-token is one of the earliest timed-token passing protocols. In FDDI, the token rotation time may reach 2(TTRT) [6]. Due to this token lateness, an FDDI network can use at most half of its bandwidth to transmit synchronous traffic [5][13][14]. To alleviate this deficiency, Shin et al. proposed the FDDI-M token protocol [5]. In FDDI-M, the token is never late. This allows FDDI-M to double FDDI's ability to support synchronous traffic. However, FDDI-M has one major weakness; starvation of asynchronous traffic. This means that in some cases, FDDI- M may not be able to transmit asynchronous traffic. Budget Sharing Token (BuST) protocol [14][15] and Timely-Token [13] protocols are timed-token protocols recently introduced to improve the communication services provided by FDDI and FDDI-M networks. The BuST and Timely-Token solved the problems of token-lateness in FDDI and the starvation of asynchronous traffic in FDDI-M.

1.1 Contributions and Summary

This paper describes Guaranteed Services Token (GuST) protocol, which improves the communication services provided by the existing timed-token protocols, including the Timely-Token and BuST protocols. GuST combined the timed-token mechanisms of the Timely-Token protocol and BuST protocols. GuST differs from the existing timed-token protocols in how each node exploits the available bandwidth to deliver non real-time (asynchronous) traffic. We showed that in GuST protocol, the token is never late. We also compared GuST protocol against the Timely-Token and BuST protocols. Our comparison focuses on the ability of those protocols to support synchronous and asynchronous traffic. We demonstrated that the performance achieved by GuST is better in most cases than Timely-Token, BuST and other timed-token protocols.

1.2 Arrangement of the Paper

The paper is organized as follows. The network and message models are introduced in Section 2. Timely-Token and BuST protocols, along with their weaknesses, are described in Section 3. The GuST (Guaranteed Services Token) protocol is described in Section 4. Also, the performance bounds of the GuST protocol is presented in section 4. Section 5 compares the GuST against Timely-Token and BuST protocols. Also, in Section 5, sample numerical example and discussion of results are presented. Finally, conclusion and recommendations for further studies are given in Section 6.

2.0 Timed-Token Network and Traffic Models

- **Network Model**

The timed-token protocols in this paper operate on a token ring network consisting of N nodes. Each node has a unique number in the range 0, 1, 2…N-1. For each node i, the next node is node (i+1) or more appropriately node (i+1) mod N. The token frame circulates around the ring from node i to nodes i + 1, i + 2, … until node i + (N-1), then to nodes i , i + 1, i+2,…, etc. Let w_i denote the latency or walk-time between a node i and its upstream neighbor node (i + 1). The sum of all such latencies in the ring is known as the ring latency or the token walk-time, W, where W = w_i

- **Message Model**

Messages generated in the system at run time may be classified as either synchronous messages or asynchronous messages. Agrawal et al. [16] showed how a token-ring network having multiple synchronous streams per station could be transformed into a logically-equivalent network with one synchronous stream per station. Therefore, without loss of generality, we assume a single synchronous stream per station. The synchronous stream of station i is characterized by the triple (C_i , P_i, D_i). Message length, C_i, is the amount of time needed to transmit a maximum size message. Period length, P_i is the minimum inter-arrival period for the synchronous message stream at node i. Message deadline; D_i is the maximum amount of time that can elapse between a message arrival and the completion of its transmission. Thus, if a message stream arrives at time t, then it must be transmitted by time $t + D_i$. Similar to the Timely-Token in [13], we will assume $D_i \leq P_i$.

Furthermore, in the following discussion we assumed that the network is free from hardware or software failures.

3. Operation of the Existing Timed-Token Protocols

Generally, in the timed-token protocols, during the initialization, each node i declares a Target Token Rotation Time, TTRT. The minimum declared value is selected as the ring's TTRT. Each node i is then assigned a portion *hi* of the TTRT to transmit its synchronous traffic. When a node receives the token, it can transmit its real-time traffic for a time not greater than h_i time units. However, to initialize the timers, no packets are transmitted during the first token rotation.

The main difference among the various timed-token protocols concerns the non real-time message service. Let H be defined as, whereis the sum of the time reserved for the synchronous traffic in all the nodes in every cycle. Let T = H +W, where T is the total time allocated per cycle to the synchronous traffic and walk-time. The value of TTRT is denoted as . In the timed-token protocols, there are two categories of bandwidths that can be used by the asynchronous traffic, namely;

Category I: (**-T**) which is the total bandwidth that is not allocated to the synchronous traffic and ring latency.

-T bandwidth **(**time units) is available to the asynchronous traffic in every cycle. Let A* = **-**T

Category II: (U) which is the bandwidth that is allocated to the synchronous traffic but not used by the synchronous traffic in the previous cycle.

The different timed-token protocols differ in the way they allocate the two categories of available bandwidth to the asynchronous traffic. We now consider the asynchronous bandwidth allocation mechanisms employed in some selected timed-token protocols. Due to lack of space, we will not review the FDDI and FDDI-M protocols. It has been shown in [13] and [15] that the Timely-Token and the BuST protocols respectively perform better than the FDDI and FDDI-M. More details can be found in [13][14][15] [17]. For lack of space, we will present only those aspects of the protocols that are vital to our discussion.

3.1 Asynchronous Traffic Transmission Mechanism in the Timely-Token Protocol

In FDDI and FDDI-M protocols, problems occurred because a station cannot distinguish between unused synchronous bandwidth and unused asynchronous bandwidth. To overcome this, an integer **U** is added to the token, where **U** represents the sum of unused synchronous bandwidth of all stations during the previous cycle [13]. When the token arrives in station i, **U** should also include the unused synchronous bandwidth of station i in the previous cycle. In the Timely-Token, when the token arrives at a node, the node can transmit asynchronous traffic for a time not greater than the Token

Holding Time, THT_i where THT_i is derived from the Timely-Token algorithm [13] as;

$THT_i = TTRT - U - TRT_i$ for $TRT_i < TTRT - U$ otherwise , $THT_i = 0$, where TRT_i is the Token Rotation Time .

TRT_i measures the time between token arrivals at node i.

Drawbacks of Timely-Token Protocol

In the Timely-Token, asynchronous traffic makes use of only ***Category I*** available bandwidth. The Timely-Token does not permit the asynchronous traffic to use the spare bandwidth (i.e. U) left over by the synchronous traffic. As such, the throughput of the Timely-Token decreases when **U** > 0.

3.2 Asynchronous Traffic Transmission Mechanism in The BuST Protocol

In the BuST, a node can deliver asynchronous traffic each time it gets the token, early or not, using the spare bandwidth (i.e. U) left by the synchronous traffic. If s_i is the time units consumed by node i to deliver synchronous traffic, then it can send asynchronous traffic for a time not greater than h_i-s_i time units even if the token is not early.

Drawbacks of BuST Protocol

In BuST protocol, the asynchronous traffic makes use of only ***Category II*** available bandwidth. As such, when the load level of the synchronous traffic is heavy, $s_i = h_i$, then, no bandwidth will be left for the asynchronous traffic. In that case, asynchronous traffic will be starved. Besides, ***Category II*** bandwidth is not allocated in such a way that the unused bandwidth in a node can be used by the asynchronous traffic in another node. So, while some nodes with light load of synchronous and asynchronous traffic may have spare bandwidth left over, the other nodes with heavy load of synchronous traffic will still starve their asynchronous traffic as they cannot use the spare bandwidth from other nodes.

4.0 Outline of the GuST Protocol

- During the ring initialization phase, each node i declares a TTRT. The minimum declared value is selected as the ring's TTRT. Each node i is then assigned a portion h_i of the TTRT to transmit its synchronous traffic in every cycle. During each token rotation, station i can transmit synchronous packets for at most h_i time units.
- Each station i has a token-rotation timer, TRT_i for measuring the time between token arrivals.
- Each station i has an asynchronous-limit variable, A_i. In this variable, station i stores the amount of time it may transmit asynchronous messages. In addition, station i maintains a variable $\boldsymbol{\varphi}_i$, where it stores the portion of h_i, the reserved synchronous bandwidth it used in transmitting synchronous traffic in the previous token-rotation. We also define another variable b_i where station i stores the portion of h_i, the reserved synchronous bandwidth it used in transmitting asynchronous traffic in the previous token-rotation. Also, station i maintains a variable s_i, where it stores the total time units used out of h_i, in the previous token-rotation, where $s_i = \boldsymbol{\varphi}_i, + b_i.$
- To initialize the token-rotation timers, no packets are transmitted during the first token rotation. In addition, s_i is set to zero for all i, and U = = H. The integer U is added to the token, where U represents the sum of unused synchronous bandwidth of all stations during the previous token-rotation. When the token arrives at station i, U should also include the unused synchronous bandwidth of station i in the previous token-rotation.

When station i receives the token, it performs the following steps:

1. A_i := (TTRT - U -TRT_i)+
2. TRT_i := 0
3. U := U - (h_i - s_i)
4. If node i has synchronous packets, it transmits them until TRT_i counts up to h_i, or until all the synchronous traffic is sent, whichever comes first.
5. $\boldsymbol{\varphi}_i$ is assigned the number of time units of synchronous transmission used in step 4.
6. If TRT_i < h_i then if node i has asynchronous packets, it transmits them until TRT_i counts up to h_i, or until all the asynchronous traffic is sent, whichever comes first.
7. bi is assigned the number of time units of asynchronous transmission used in step 6.
8. s_i is assigned the total number of time units of synchronous and asynchronous transmissions used in step 4 and step 6.
9. U := U + (h_i - s_i)
10. If station i has asynchronous packets, it transmits them for a time period of up to A_i time units, or until all its asynchronous packets are transmitted, whichever occurs first.
11. Station i passes the token to station (i + 1) mod N.

4.1 Performance Bounds

In this section we show that in GuST protocol the token is never late. In principle, GuST operates like a heavily loaded Timely-Token protocol. The difference lies in how GuST and the Timely-Token handle U, the drop in load of synchronous traffic. In GuST protocol, ***Category I*** (i.e A*) available bandwidths are allocated to the asynchronous traffic just like in the Timely-Token protocol. At the same time, ***Category II*** (U) spare bandwidths left over by the synchronous traffic are allocated to the asynchronous traffic just like in the BuST protocol. Consequently, maximum throughput is maintained by GuST even in the face of drop or variation in the load level of the synchronous traffic.

Technically, the difference between GuST and Timely-Token is that in the Timely-Token s_i = $\boldsymbol{\varphi}_i$ whereas in the GuST s_i = $\boldsymbol{\varphi}_i$, + b_i . As such, analysis of the GuST protocol is simply the analysis of the heavily loaded Timely-Token system where s_i is composed of $\boldsymbol{\varphi}_i$, and b_i, the bandwidths used by the synchronous and asynchronous traffic respectively. Hence, in our analysis, we adopted the approach employed in [13] for the heavily loaded Timely-Token. There is however one slight difference in the assumption made here. In [13], the system is assumed to be heavily loaded with synchronous and asynchronous traffic. In this paper, the system is loaded with light load of synchronous traffic but with heavy load of asynchronous traffic. As such, in this paper, in every token receipt, the synchronous traffic may not use all the time units reserved for it in the node. However, the unused portions of the reserved synchronous time units are used by the asynchronous traffic in every node. In this way, the system still behaves like a heavily loaded system since all the time units for data transmission are used up in every node in every token receipt.

In order to reason about values that change over time, we enhance our notation to include rounds, that is, token rotations.

Definitions

: round m of station i, i.e., time interval [t,], where t is the time when station i receives the token for the mth time, and , is the time when station i receives the token for the (m + 1)th time.

: value assigned to A_j during . In particular, is the value assigned to A_i when the token is received at the beginning of .

: duration of asynchronous transmission of station j during . Note that ≤ [13].

h_j **:**duration of time units reserved for synchronous transmission of station j in every round.

: the portion of the h_j time units actually used for synchronous transmission in station j during .

Note that ≤ h_j [13].

: the portion of the h_j time units actually used for asynchronous transmission in station j during .

: the total of the portions of the h_j time units actually used for synchronous and asynchronous transmissions in station j during .

Note that ≤ h_j [13]. Also, = + ≤ h_j

: value of TRT_j when station j receives the token during .

In particular, is the value of TRT_i when the token is received at the beginning of.

= + + **W** [13].

Theorem 1 (The Token is never late)

For every station i, upon token arrival, $TRT_i \leq TTRT$.

The proof for **Theorem 1** is given in [13]. The same applies to GuST protocol. It was shown in [13] that for the heavily loaded Timely-Token protocol, the following expressions hold;

≤ A*; U = **-** and ≤ h_j. From our discussion in this paper, we can see that for the Timely-Token protocol [13], = whereas, for the GuST protocol,

= + ≤ .

Then, **=** + + W

= + + W

≤ + A* + W

≤ **TTRT**

So, the token is never late since the Token Rotation Time, TRT_i does not exceed TTRT.

5.0 Comparison of GuST Against The Timely-Token and BuST Protocols

In this section, we compare the GuST protocol against the Timely-Token and BuST Protocols. Our comparison focuses on the ability of these protocols to support synchronous and asynchronous traffic. We base our comparison on the expression for the upper bound on the average cycle length (**Ĉ**) for these protocols, because the expressions directly reflect the ability of the protocol to provide services to the synchronous and asynchronous traffic.

5.1 Expression For The Upper Bound On The Average Cycle Length ($\hat{C}$)

FDDI Protocol

In FDDI timed-token protocol [6][7][18][19], each node has two timers, the Token Holding Timer (THT_i) and the Token-Rotation-Timer (TRT_i). The TRT_i counter always increases, whereas the THT_i only increases when the node is delivering asynchronous traffic. When TRT_i reaches TTRT, it is reset to 0 and the token is considered as late by incrementing the node's late count ,Lc_i by one. The actual token cycle time, denoted in this paper as is given as = $TTRT_i$ + Lc_i (TTRT). The token is considered to arrive early at node i if $Lc_i = 0$ otherwise the token is late (in this case, $Lc_i \geq 1$). When the token arrives at a node, the node can transmit asynchronous traffic for a time no greater than THT_i where THT_i is given as;

$THT_i = TTRT_i$ - for < $TTRT_i$

otherwise $THT_i = 0$; (1)

where $TRT_i^{\#}$ is the time spent in the last round-trip of the token. Then, for the FDDI, $A_i = \max(0, TTRT-TRT_i^{\#})$.

Joseph and Fouad has shown in [18] that for FDDI protocol, the upper bound on the average cycle length (**$\hat{C}$**) for a heavily load system is given as

$\hat{C} \leq$(**-T**)+ **T** (2)

Then, the upper bound on the average bandwidth allocated to the asynchronous traffic (Ấ) is given as

Ấ = (**-T**) (3)

Similarly, Ozuomba and Chukwudebe showed in [19] that for FDDI protocol, **$\hat{C}$** and Ấ for a system with light load of synchronous traffic but with heavy load of asynchronous traffic, are defined as follows;

$\hat{C}$ $\leq$(**-T**) +**U** + **(H -U**) +**W** (4)

Ấ = (**-T**) + **U** (5)

where U is the unused synchronous transmission time in the last round-trip of the token. The assumption made in [19] is that U is constant for at least the N+ 1 consecutive cycle where the average is taken.

Timely-Token Protocol

The difference between the FDDI and the Timely-Token is in the use of TTRT in the FDDI and TTRT* in the Timely-Token protocol, where TTRT* = TTRT - U. For the Timely-Token, U= and s_i = $_i$ then, we can replace with **-U** in the expressions for **$\hat{C}$** in Eq 4 and Ấ in Eq 5 to obtain **$\hat{C}_T$** and **Ấ$_T$** for the Timely-Token, where

= and=

$\hat{C}_T$ $\leq$(**-T**)+ + **W** (6)

Ấ$_T$ = (**-T**) (7)

BuST Protocol

In the BuST protocol, ***Category I*** available bandwidth (i.e. (**-T**) is not used by any traffic. The asynchronous traffic makes use of only the ***Category II,*** which is the U spare bandwidth left over by the synchronous traffic. So, $THT_i = 0$ for i and $A_i = U$. Now U= and $s_i = _i$, thus, $\hat{C}_B$ and $\acute{A}_B$ for the BuST protocol are given as follows;

$$\hat{C}_B \leq \mathbf{U ++ W}$$

$$\leq \mathbf{(}H \mathbf{-) ++ W} \qquad (8)$$

$$\acute{A}_B = U = H \mathbf{-}$$

$$(9)$$

GuST Protocol

For the GuST protocol, TTRT* = TTRT-U,

U = and $s_i = _i + b_i$. Since we are considering a system with heavy load of asynchronous traffic, then,

= + .

We can replace with **-U** in the expressions for $\hat{\mathbf{C}}$ in Eq 4 and Á in Eq 5 to obtain $\hat{C}_G$ and $\acute{A}_G$ as follows;

$$\hat{C}_G \leq (\mathbf{-T})+ \mathbf{++ W}$$

$$\leq (\mathbf{-T})+ \mathbf{(}H \mathbf{-) ++ W} \qquad (10)$$

$$\acute{A}_G = (\mathbf{-T})+$$

$$= (\mathbf{-T})+\mathbf{(}H \mathbf{-)} \qquad (11)$$

5.2 Worked Example

Consider a ring network with four nodes (i.e. N = 4)

where = 100, **W** = 4 and h_i = 20 for all the nodes. We will assume that the network is heavily loaded with asynchronous traffic but with a variable load of the synchronous traffic. The synchronous traffic load, $_i$ can vary from 0 to h_i. The values of $\hat{\mathbf{C}}$ and Á for the various load levels of the synchronous traffic are computed for the Timely-Token, BuST and GuST protocols. The results are presented in Table 1 and Table 2 and also in the graphs of Fig 1 and Fig 2.

Table 1: Comparison of Average Bandwidth for Asynchronous Traffic Per Cycle, Â for BuST, Timely-Token and GuST.

		BuST	**Timely-Token**	**GuST**
i	**$\sum_i$**	**Â**	**Â**	**Â**
0	**0**	**80**	**12.8**	**92.8**
2	**8**	**72**	**12.8**	**84.8**
4	**16**	**64**	**12.8**	**76.8**
6	**24**	**56**	**12.8**	**68.8**
8	**32**	**48**	**12.8**	**60.8**

10	**40**	**40**	**12.8**	**52.8**
12	**48**	**32**	**12.8**	**44.8**
14	**56**	**24**	**12.8**	**36.8**
16	**64**	**16**	**12.8**	**28.8**
18	**72**	**8**	**12.8**	**20.8**
20	**80**	**0**	**12.8**	**12.8**
$h_i = 20$, =100, N = 4, W = 4				

Table 2: Comparison of the computed values of Average Cycle length, Ĉ for BuST, Timely-Token and GuST.

		BuST	**Timely-Token**	**GuST**
$_i$	$\sum_i$	**Ĉ**	**Ĉ**	**Ĉ**
0	**0**	**84**	**16.8**	**96.8**
2	**8**	**84**	**24.8**	**96.8**
4	**16**	**84**	**32.8**	**96.8**
6	**24**	**84**	**40.8**	**96.8**
8	**32**	**84**	**48.8**	**96.8**
10	**40**	**84**	**56.8**	**96.8**
12	**48**	**84**	**64.8**	**96.8**
14	**56**	**84**	**72.8**	**96.8**
16	**64**	**84**	**80.8**	**96.8**
18	**72**	**84**	**88.8**	**96.8**
20	**80**	**84**	**96.8**	**96.8**
$h_i = 20$, =100, N = 4, W = 4				

BuST
Timely-Token
GuST

Fig 1: Comparison of the upper bound on the bandwidth allocated to the asynchronous traffic, **Â** for the timed-token protocols studied. (Table 1 plot)

BuST
Timely-Token
GuST

Fig 2: Comparison of the upper bound on the average cycle length, $\hat{C}$ for the timed-token protocols studied. (Table 2 plot)

5.3 Discussion of results

A System With No Synchronous Traffic But With Heavy Load Of Asynchronous Traffic

When there is no synchronous traffic, that is = 0; H = 80, U = 80, Row 3 of Table 1;

- BuST will allocate all the (U) spare bandwidths to the asynchronous traffic , that is $\hat{A}$ = 80 (Table1 , row 3 , column 3) and **$\hat{C}$ = 84** (Table 2 , row 3 , column 3)
- Timely-Token will allocate the same constant average bandwidth () to the asynchronous traffic, where A* = 16, N = 4 , so = **12.8** . So $\hat{A}$ = **12.8** (Table1 , row 3 , column 4) and **$\hat{C}$ = 16.8** (Table2 , row 3 , column 3)
- GuST will allocate an average bandwidth) to the asynchronous traffic, where A* = 16, N = 4 , so, = **12.8** . Then, $\hat{A}$ = **92.8** (Table1 , row 3 , column 5) and **$\hat{C}$ = 96.8** (Table 2 , row 3 , column 5)

Thus, in the case of a system with no synchronous traffic but with heavy load of asynchronous traffic, the Timely-Token will allocate the least amount of bandwidth to the asynchronous traffic while the GuST will allocate the highest. The BuST will allocate all the spare bandwidth (U = H -= 80) left unused by the synchronous traffic to the asynchronous traffic ($\hat{A}$ = H = 80) .

A System With Heavy Load Of Synchronous and Asynchronous Traffic

When there is heavy load of synchronous traffic, that is = H = 80, U = H -= 0 then

- BuST will not allocate bandwidth to the asynchronous traffic , that is $\hat{A}$ = 0 (Table1, row 13, column 3) and **$\hat{C}$ = 84** (Table 2 , row 13, column 3)
- Timely-Token will allocate a constant average bandwidth () to the asynchronous traffic, where A* = 16, N = 4 , so = **12.8** . So $\hat{A}$ = **12.8** (Table1 , row 13, column 4) and **$\hat{C}$ = 96.8** (Table 2 , row 13, column 4)
- GuST will allocate an average bandwidth) to the asynchronous traffic, where A* = 16, N = 4 , so =**12.8**). So, $\hat{A}$ = **12.8** (Table1 , row 13, column 5) and **$\hat{C}$ = 96.8** (Table 2 , row 13, column 5)

Thus, in the case of a system with heavy load of synchronous and asynchronous traffic, the Timely-Token and the GuST have the same throughput which is higher than the BuST throughput. In particular, BuST will not allocate any bandwidth to the asynchronous traffic, in this case ($\hat{A}$ = 0).

A System With Variable Load Level of Synchronous Traffic But With Heavy Load Of Asynchronous Traffic

From the plot, Fig1 it can be seen that if there is heavy load of asynchronous traffic, then, as the load of the synchronous traffic increases from zero (no synchronous traffic) to the heavy load state,

- Timely-Token allocates the same amount of average bandwidth () to the asynchronous traffic. The graph is straight line with zero slope.
- BuST allocates all the spare bandwidths (U = H -) left over by synchronous traffic to the asynchronous traffic. From Fig 1, the decrease in bandwidth allocated to the asynchronous traffic is proportional to increase in the synchronous traffic.

- GuST allocates plus the spare bandwidth (U = H -) left over by synchronous traffic to the asynchronous traffic. From Fig 1, it can be seen that GuST has the same rate of decrease but higher throughput than BuST.

Similarly, from the plot, Fig 2, it can be seen that if there is heavy load of asynchronous traffic, then, as the load of the synchronous traffic increases from zero (no synchronous traffic) to the heavy load state,

- the average cycle length, $\hat{\mathbf{C}}_\mathbf{T}$ of Timely-Token increases as the synchronous traffic increases. Specifically, $\hat{\mathbf{C}}_\mathbf{T}$ increases from its lowest value (of () + W = **16.8** when = 0) to its maximum value (of () + H + W = **96.8)** when = 80.
- the average cycle length, $\hat{\mathbf{C}}_\mathbf{B}$ of BuST remains constant at its maximum value (of **H + W = 84**) .

- the average cycle length, $\hat{\mathbf{C}}_\mathbf{G}$ of GuST remains constant at its maximum value (of () + **H + W = 96.8**) .

Thus, in the case of a system with heavy load of asynchronous traffic but with variable load of synchronous traffic, the GuST protocol maintains higher throughput than the Timely-Token and BuST protocols as long as < H.

6.0 Conclusion and Recommendations

6.1 Conclusion

This paper presented the Guaranteed Services Token protocol (GuST) which improved the performance of existing timed-token protocols, including the Timely-Token and BuST protocols. BuST and Timely-Token protocols are timed-token protocols recently introduced to improve the communication services provided by FDDI and FDDI-M networks. However, GuST maintained higher throughput than BuST and Timely-Token protocols in the face of variations in the load level of the synchronous traffic. At the same time, GuST delivered guaranteed services as required by the hard and soft real-time applications. Consequently, GuST is more suitable for integrated services network since it can efficiently support different traffic patterns and also provide not only bounded message transmission time as required by the hard real-time tasks, but also high throughput, as demanded by soft real-time and non-real-time tasks.

6.2 Recommendations

GuST can be incorporated into Ethernet and Profibus networks to improve on the performance of those networks. The approach to be adopted and the implementation issues are areas of further research.

REFERENCES

- Regnier, P. and Lima, G. (2006). Deterministic integration of hard and soft real-time communication over Shared-Ethernet. In Proc. of Workshop of Tempo Real, Curitíba, Brazil, 2006

- Nicholas Malcolm, Wei Zhao, "The timed-token protocol for real-time communications," Computer, vol. 27, no. 1, pp. 35-41, January, 1994
- The Institute of Electrical and Electronic Engineers, “Token-passing bus access method and physical layer specifications , “ America national Standard ANSI/IEEE std. 802.4 – 1985
- Biao Chen and Wei Zhao, “Properties of the Timed Token Protocol”, Department of Computer Science Texas A&M University College Station, TX 77843-3112 Oct., 1992 Technical Report 92-038
- K. G. Shin and Q. Zheng, “FDDI-M: A scheme to double FDDI’s ability of supporting synchronous traffic”, IEEE Trans. on Parallel and Distributed Systems, Vol. 6, No. 11, pp. 1125 -1131,.Nov. 1995
- Kenneth C. Sevcik and Marjory J. Johnson, "Cycle time properties of the FDDI token ring protocol," IEEE Trans. Software. Eng., vol. SE-13, pp. 376-385, Mar. 1987
- R. Grow, “A timed token protocol for local area networks”, Proc. Electro’82, Token Access Protocols, Paper 17/3, May 1982
- E. Chan, D. Chen, J. Cao and C. H. Lee, “ The time Properties of the FDDI-M Medium Access Control Protocol” The Computer Journal, Vol. 82 No. 1 , pp. 96-102, Jan. 1999
- Survivable Adaptable Fibre-Optic Embedded Networks, MIL- STD-2004, US Dept. of Defense, Washington D.C., Sept, 1992
- L.J. Mcguffin, L.O Reid, and S.R. Sparks, “MAP/TOP in CIM Distributed Computing, “ IEEE Network, vol.2, no. 3 , May 1988, pp. 23 – 31
- R.W. Uhlhorn, “The fibre-optic high-speed data bus for a new generation of a military aircraft”, IEEE LCS, Vol.2, No.1, pp. 36 – 43, Feb. 1991
- E. Tovar, F. Vasques, “Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications”, Proc. of the 15th IFAC Workshop on Distributed Computer Control Systems, 1998
- Jorge Arturo Cobb, Miaohua Lin, ” The Timely-Token protocol,” Computer Communications Vol .27, no.7, pp. 569-580 May, 2004
- G. Franchino, G. C. Buttazzo, and T. Facchinetti, "Time Properties of the BuST Protocol under the NPA Budget Allocation Scheme”. In Proc. Of the Conference on Design, Automation and Test in Europe(Date 2008) , Munich, Germany, 10th -14th March 2008
- G. Franchino, G. C. Buttazzo, and T. Facchinetti, “BuST: Budget Sharing Protocol for Hard Real-Time Communication”. In Proc. of the 12th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2007), Sept. 2007
- Gopal Agrawal, Biao Chen, Weizhao, and Sadegh Davari (1994) “Guaranteeing synchronous message Deadlines with the Timed Token Access Control Protocol “, IEEE. Transaction on computers, Vol 43. No.3, March 1994 pp. 237-239
- J. Cobb, M. Lin, “The On-Time timed-token protocol”, Proc. IEEE GLOBECOM, 2002
- Joseph W. M. and Fouad A. T. “Throughput Analysis of a Timer-Controlled Token Passing Protocol Under Heavy Load”, IEEE trans. Comm, vol 37, No7, pp. 694 – 702, July 1989
- S. Ozuomba and G.A Chukwudebe, “An Improved Algorithm For Channel Capacity Allocation In Timer Controlled Token Passing Protocols”, A Journal Paper Published in An International Journal Of Nigerian

Computer Society (NCS), Vol. 9 No 1 , pp. 116 – 124, June 2003