PERFORMANCE ANALYSIS OF TIMELY-TOKEN PROTOCOL WITH VARIABLE LOAD OF SYNCHRONOUS TRAFFIC

By

Simeon Ozuomba
simeonoz@yahoo.com
234-7030345527
Department of Electrical/Electronic and Computer Engineering, University of Uyo, Akwa Ibom , Nigeria.

Gloria A. Chukwudebe
gloriachukwudebe@yahoo.com
234-8033279081
Department of Electrical and Electronic Engineering and Engineering Technology, Federal University of Technology Owerri, Nigeria

Abstract

Timely-Token protocol solved the problems of token-lateness in FDDI and starvation of asynchronous traffic in FDDI-M. Jorge and Miaohua proposed the Timely-Token protocol and presented its timing analysis. In this paper, we presented the performance analysis of the Timely-Token protocol with variable load of synchronous traffic but with heavy load of asynchronous traffic. The essence of such analysis is to determine how variations in the load level of the synchronous traffic affect the bandwidth allocated to the asynchronous traffic. Specifically, we derived the expressions for the Maximum Cycle Length, the Average Cycle Length and the Upper Bound on the Average Bandwidth allocated to the asynchronous traffic. Secondly, we developed mathematical expressions for relating the simulation results of the Timely-Token algorithm to the analytical results. Thirdly, we used a program written in Visual Basic for Applications (VBA) to run the simulation of the Timely-Token algorithm and also perform the computations of the performance parameters using the expressions derived from our analytical approach.

The simulation results of the Timely-Token algorithm corresponded with the analytical results for various network configurations. Finally, we believe that the ideas presented in this paper will be relevant to network designers and researchers, especially those wishing to develop improved versions of the timed-token algorithm. The expressions will enable researchers to make comparative studies of the performance of the Timely-Token protocol and other timed-token protocols. Also, the expressions for relating the simulation results to the analytical results are easy-to-use tools for validating the values obtained from the analytical computations.

Keywords: *Timed-Token, FDDI, FDDI-M, Timely Token Protocol, Synchronous traffic, Asynchronous traffic, channel capacity, Media Access Control (MAC) algorithm.*

1.0 Introduction

Computer networks are increasingly being used to support task with diversity of requirements. While the hard real-time traffic is in most cases, short , periodic and requires strict time constraint, the soft real-time traffic on the other hand deal with sporadic or aperiodic events, leading to a considerable task diversity (Regnier et al., 2006). In addition, the structure of real-time systems has been moving from centralized to distributed systems. This requires a suitable communication network capable of dealing with the new demands task diversity requires. An attractive approach for integrating such traffic is the Timed-Token passing protocol. Consequently, the Timed-Token protocol has been incorporated into several high-bandwidth network standards: FDDI, IEEE802.4 Token Bus, SAFENET , Manufacturing Automation Protocol (MAP) , High-Speed Ring Bus and in PROFIBUS [Nicholas et al., 1994], [Biao et al., Zhao 1992], [Shin et al.,1995 and Grow, 1983], [McGuffey et al., 1988], [Uhlhorn et al., 1991] and [Tovar et al., 1998].

Some of the key performance measures for the Timed-Token algorithm are the Maximum Cycle Length, the Reserved Bandwidth for the Synchronous Traffic and the Throughput for the Asynchronous Traffic. All of these performance measures are dependent on the selected Target Token Rotation Time (TTRT). Due to token lateness in the original timed-token protocol proposed in (Grow 1992) and employed in FDDI (Biao et al.,1992), the token cycle time may exceed the TTRT. In (Kenneth et al., 1987) Johnson and Sevcik formally proved that when the network operates normally, the cycle length is bounded by 2(TTRT) . This applies to the Timed-Token protocol in FDDI (Shin et al., 1995). In this case, the FDDI Timed-Token protocol can use at most half of its bandwidth to transmit synchronous traffic (Shin et al.,1995) and (Jorge et al., 2004). FDDI-M protocol (Shin et al.,1995) and Timely-Token protocol (Jorge et al., 2004) eliminated the token lateness problem and limited the upper bound on the access delay to TTRT thereby doubling the Timed-Token protocol's ability to support synchronous traffic. However, the major problem with FDDI-M is starvation of the asynchronous traffic. This problem is solved in the Timely-Token protocol except that the throughput realized for the synchronous traffic in the Timely-Token protocol in most load situations is less than the optimum throughput achievable for such traffic. To solve this problem, proper understanding of the workings of the Timely-Token protocol under various load levels of the synchronous traffic is needed. This paper presents the performance analysis of the timed-token algorithm in the Timely-Token protocol with variable load levels of the synchronous traffic but with heavy load of asynchronous traffic. The study provides analytical tool that will enable researchers to have a better understanding of how the channel capacity is allocated to the asynchronous traffic under various load levels of the synchronous traffic; from zero load level (i.e. no) of synchronous traffic to the full load level or heavy load of synchronous traffic.

1.1 Contributions and Summary

In this paper, we presented performance analysis for the timed-token algorithm in Timely-Token protocol with variable load of synchronous traffic but with heavy load of asynchronous traffic. Similar performance analysis can be found in (Joseph et al., 1989) for a heavily loaded FDDI system, in (Ozuomba et al., 2003) for FDDI systems with variable load of synchronous traffic but with heavy load of asynchronous traffic. In (Ozuomba et al., 2003), the performance parameters computed are the Maximum Cycle Length, the Average Cycle Length, and the Average Bandwidth allocated to the asynchronous traffic for the FDDI.

First, we adopted the analytical approach presented in (Joseph et al., 1989) and (Ozuomba et al., 2003) for timed-token algorithms. Specifically, we derived the expressions for the Maximum Cycle Length, the Average Cycle Length, and the Upper Bound on the Average Bandwidth allocated to the asynchronous traffic for the timed-token algorithm in the Timely-Token protocol. With the expression for the Upper Bound on the Average Bandwidth allocated to the asynchronous traffic, we can determine to what extent the Timely-Token protocol solved the problem of starvation of the asynchronous traffic in the FDDI-M. Specifically, we derived the mathematical expression that showed that for any given network configuration, with the Timely-Token protocol, the variations in the load level of the synchronous traffic do not affect the bandwidth allocated to the asynchronous traffic. This is unlike the FDDI and the FDDI-M protocols where the variations in the load level of the synchronous traffic affect the bandwidth allocated to the asynchronous traffic.

Secondly, in this paper, the results obtained from analytical computations were validated by simulation results. We developed mathematical expressions for relating the simulation results of the Timely-Token timed-token algorithm to the analytical results for the Maximum Cycle Length, Average Cycle Length and

the Upper Bound on the Average Bandwidth allocated to the asynchronous traffic. This made it possible to compare and hence, validate the analytical results with data obtained from the simulation of the Timely-Token timed-token algorithm. Also, we stated the conditions under which such expressions can be used for effective comparison of the analytical and simulation results.
Furthermore, sample numerical values were obtained with the analytical computation and simulation of the Timely-Token timed-token algorithm. The results obtained from the simulation were compared with the analytical results for various network configurations.
The rest of this paper is arranged as follows. In section 2, the characteristics of the system under consideration are presented. The Timely-Token protocol media access control algorithm along with the analysis of its timing properties are presented in section 3. Sample numerical examples, graphs and discussion of results are presented in section 4. Section 5 contains the concluding remarks and suggestions for future work.

2.0 System Characteristics for the Timely-Token Protocol

The Timely-Token network model and message model presented in this section are elaborated versions of those presented in (Joseph et al., 1989) and (Ozuomba et al., 2003) for the FDDI network. The timed-token algorithm presented in this section is elaborated versions of those presented in (Jorge et al., 2004) for the Timely-Token protocol. Specifically, more terms are defined and additional explanatory notes are added.

2.1 Network Model

We consider a token ring network consisting of N nodes. Each node has a unique number in the range 0, 1, 2…N-1. Each node is connected to two other neighboring nodes by unidirectional point-to-point media that form a single closed path. For each node i, the next node along the unidirectional medium is station (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,…, helping to determine which node should send a frame of message among the contending nodes. 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

$$W = \sum_{i}^{N-1} w_i \qquad (1)$$

2.2 Message Model

Messages generated in the system at run time may be classified as either synchronous messages or asynchronous messages. Without loss of generality, we will assume that there is one synchronous message stream at each node, i (Gopal et al., 1994). Hence, in the N-node ring, the synchronous message set, M, consist of N streams of synchronous messages, $s_0, s_1, s_2 \ldots s_{N-1}$ where, $M = \{s_0, s_1, s_2 \ldots s_{N-1}\}$ and $s_i = \{P_i, C_i, D_i\}$. 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 message stream, s_i 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 s_i arrives at time t, then it must be transmitted by time t + D_i.
In the following discussion we assumed that the network is free from hardware and software failures.

2.3 Protocol Parameters

Target Token Rotation Time, (TTRT): TTRT represents the expected time needed by the token to complete an entire round-trip of the network. When the network is initialized the value of TTRT is selected such that it will be sufficiently small to support the response time requirements of the messages at all the nodes in the network. Let τ denote the value of TTRT.
Synchronous Capacity of Node i (hi): h_i represents the maximum time for which a station, i is allowed to transmit synchronous messages every time it receives the token. Let **H** be the total time allocated to the synchronous traffic per cycle. Thus,

$$\mathbf{H} = \mathbf{h}_0 + \mathbf{h}_1 + \ldots \mathbf{h}_{N-1} = \sum_{i}^{N-1} h_i \qquad (2)$$

Let α_i be the total time allocated to the station i for its walk-time, w_i and its synchronous traffic, h_i . Thus,

$$h_i + w_i = \alpha_i \qquad (3)$$

Let T be the total allocated time units per cycle. Thus $\alpha_0 + \alpha_1 + \dots \alpha_{(N-1)} = \sum_{i}^{N-1} \alpha_i = T$ (4)

$$\therefore T = \sum_{i=0}^{i=N-1} h_i + \sum_{i=0}^{i=N-1} w_i = W + H \quad (5)$$

Let, γ_i be the used portion of the $\mathbf{h}_i$ and let, ε_i be the unused portion of the $\mathbf{h}_i$ time reserved for the synchronous traffic in node i (where $\gamma_i \le h_i$), Then, for a system that is lightly loaded with synchronous traffic, out of the h_i time units allocated to the synchronous traffic in node i, only γ_i time units are used leaving ε_i time units unused. Thus

$$\gamma_i = h_i - \varepsilon_i \quad (6)$$

Constraints:
For proper operation of the Timed-Token protocol, the choice of values for the τ and T parameters must satisfy the ***Protocol Constraint*** and the ***Deadline Constraint***.
The ***Protocol Constraint*** states that the total time allocated to the synchronous traffic in the network must not exceed the available network bandwidth, that is, $H \le \tau - W$, thus , $T \le \tau$ (7)
The ***Deadline Constraint*** states that every synchronous message must be transmitted before its deadline. For the Timed-Token protocol in FDDI, the time elapsed between two consecutive visits of the token at a node can be as much as 2TTRT (Kenneth et al., 1987). However, for the Timely-Token protocol, the time elapsed between two consecutive visits of the token at a node can be as much as TTRT (Jorge et al., 2004).Therefore, in order for the deadline constraint to be satisfied in the Timely-Token protocol, we require that for i = 0,1,...N-1,

$$\tau \le \min_{i=0,1,\dots,N-1} (D_i) \quad (8)$$

Combining the **Protocol Constraint,** Eq7 and the **Deadline Constraint,** Eq8 gives

$$T \le \tau \le \min_{i=0,1,\dots,N-1} \{D_i\} \quad (9)$$

Token Rotation Timer of Node i (TRT_i): TRT_i is the cycle length or the time between two consecutive token receipts at node i. TRT_i may be less than the actual token rotation time in node i, which we will denote as $TRT_i^{\#}$.
The Unused Synchronous Bandwidth (U_r)
In the 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_r is added to the token, where U_r represents the sum of unused synchronous bandwidth of all stations during the previous cycle. When the token arrives in station i, U_r should also include the unused synchronous bandwidth of station i in the previous cycle.

Asynchronous-Limit Variable Of Node i (THT_i):
$\mathbf{THT}_i$ is used to control the amount of time for which the node i can transmit asynchronous messages.

2.4 The Timely-Token Protocol Media Access Control (MAC) Operation

Protocol Q MAC Algorithm

Q1: During the first token rotation, to initialize timers, no station is allowed to transmit any packets. In addition, γ_i is set to zero and so $\varepsilon_i = h_i$ for all i, and

$$U_r = \sum_{i=0}^{i=N-1} (\varepsilon_i) = \sum_{i=0}^{i=N-1} h_i \text{ for } 0 \le i \le N-1, \varepsilon_i = h_i \quad (10)$$

During the ring's initialization, the following parameters are initialized at all nodes: (Jorge et al., 2004)

Q1.1. TRT_i 0; **Q1.2** $\gamma_i = 0$ **Q1.3** $\varepsilon_i = h_i$

Q1.4 $U_r = U_r = \sum_{i=0}^{i=N-1} (\varepsilon_i) = \sum_{i=0}^{i=N-1} h_i = H$ **Q1.5.** Start TRT_i; TRT_i counts up.

Q2: Under normal operating conditions of the Timely-Token Protocol, with respect to TTRT, the token can never arrive late at any node. However, similar to the FDDI-M protocol (Chan et al., 1999), the Timely-Token Protocol actually uses TTRT - U_r to determine whether the token arrives early or late at node i. Specifically, the token is early if

$$TRT_i < TTRT - U_r \quad (11a)$$

The token is late if

$$TRT_i \geq TTRT - U_r \quad (11b)$$

The token can be early or late with respect to TTRT - U_r but the token can never be late with respect to TTRT. This simply means that the cycle length or the token rotation time, TRT_i cannot exceed TTRT but it can exceed TTRT - U_r.

Q3: When the token arrives at node i, the following actions take place:

Q3.1. $THT_i \leftarrow$ ~~TTRT~~ $- U_r - TRT_i$ **Q3.2.** $TRT_i \leftarrow 0$

Q3.3 Start TRT_i; TRT_i counts up. **Q3.4** $U_r = U_r - (h_i - \gamma_i)$

Q3.5 If station i has synchronous packets, it transmits them for a time period of up to h_i time units, or until all its synchronous packets are transmitted, whichever occurs first. γ_i is assigned the number of time units of synchronous transmission used above.

Q3.6 $U_r = U_r + (h_i - \gamma_i)$

Q3.7 If station i has asynchronous packets, it transmits them for a time period of up to THT_i time units, or until all its asynchronous packets are transmitted, whichever occurs first. Let δ_i be assigned the number of time units used for the asynchronous traffic transmission .

Thus , $\delta_i = THT_i$, when $THT_i = TRT_i < TTRT - U_r$, otherwise $\delta_i = 0$ when $TRT_i \geq TTRT - U_r$. Note that $\delta_i = 0$ means that no asynchronous traffic is transmitted .

Q3.8 The token is released to the next node i+1 (or more appropriately $(i+1)$ Mod N)

3.0 Methodology

In this paper, we presented performance analysis for the timed-token algorithm in the Timely-Token protocol with variable load of synchronous traffic but with heavy load of asynchronous traffic. Similar performance analysis can be found in (Joseph et al., 1989) for a heavily loaded FDDI system, in (Ozuomba et al., 2003) for FDDI systems with variable load of synchronous traffic but with heavy load of asynchronous traffic. In (Joseph et al.,1989) and (Ozuomba et al., 2003), the performance parameters computed are the *Maximum Cycle Length*, the *Average Cycle Length*, and the *Average Bandwidth* allocated to the asynchronous traffic for the FDDI network.

In this paper, the results obtained from analytical computations were validated by simulation results. However, first, we developed the expressions for relating the simulation results of the Timely-Token timed-token algorithm to the performance parameters being computed analytically. Also, we stated the conditions under which such expressions can be used for effective comparison of the analytical and simulation results. Secondly, we developed a program in Visual Basic for Applications (VBA) for simulating the Timely-Token timed-token algorithm and for performing the analytical computations of the performance parameters. The VBA program runs in Microsoft Office Excel 2007 environment. Thirdly, we used the VBA program to both simulate the Timely-Token time-token algorithm and to compute its performance parameters analytically for some selected network configurations.

The methodology employed in this paper is chosen because it is cost effective; especially for researchers in developing countries in Africa. Generally, the analytical expressions enable researchers to easily study the performance of the Timely-Token timed-token algorithm without having to set up a physical network which is expensive. The simulation makes it possible for the researchers to easily verify or validate the analytical results of the performance parameters of the Timely-Token timed-token algorithm. Meanwhile, the simulation of the Timely-Token timed-token algorithm can now be done with any simple program once the conditions and the expressions we presented in this paper are used in conducting the simulations and in relating the simulation results to the performance parameters.

3.1 Performance Analysis of The Timely-Token Protocol MAC Algorithm

We consider the Timely-Token protocol (Protocol **Q**) with **N** active nodes in the ring and then define cycle length in terms of time t_i as follows. Let $t_0, t_1, \ldots t_{(N-1)}$ be the time at which the token reaches station 0,1,…N-1 for some given cycle. Let $t_N, t_{(N+1)}, \ldots t_{(2N-1)}$ be the time at which the token reaches station 0,1,…N-1 in the next cycle and so forth.

Thus, t_i, $i \geq 0$ is the time at which the token reaches station *(i mod N)* in the cycle $\left\lfloor \frac{i}{N} \right\rfloor$ where the given cycle is denoted as cycle 0 and where $\lfloor x \rfloor$ denotes the integer part of x. Let $t_{-N}, t_{(1-N)}, \ldots t_{(N-1)-N}$ be the time at which the token reaches station 0,1,…N-1 in the previous cycle to the given cycle. If the token reaches node i in a given cycle at time t_i, then the time at which the token had reached station i, in the previous cycle to the given cycle is t_{i-N}. It should be noted that $\mathbf{TRT_i}$ is the time measures by the token rotation timer . Let the actual cycle length (or the actual time between two consecutive token receipts at node i) be given denoted as $TRT_i^{\#}$ where

$TRT_i^{\#} = t_i - t_{i-N} = \mathbf{TRT_i}$ (12)

Note that the value of i increases by one at every token receipt, thus for any given value of i, the node denoted by j (where j = *(i mod N)* and j = 0,1,2,… N-1) and cycle denoted by k , where k = $\left\lfloor \frac{i}{N} \right\rfloor$.

From Protocol Q3.1 and Eq12, we have

$THT_i = TRT_i = t_i - t_{i-N}$ (13)

Also, from Protocol Q3.7 , Eq11a and Eq11b we have;

$$\delta_i \leq \tau - U_r - (t_i - t_{i-N}) \quad \text{for } t_i - t_{i-N} < \tau - U_r \qquad (14)$$

Otherwise

$\delta_i \leq 0$ for $t_i - t_{i-N} \geq \tau - U_r$ (15)

Combining Eq14 and Eq15 gives

$$\delta_i \leq \max(0, \tau - U_r - (t_i - t_{i-N})) \text{ for all } i \geq 0 \qquad (16)$$

To simplify the analysis without lose of generality, we will assume that ε_i is the same in every cycle (except in the initialization cycle where $\gamma_i = 0$, hence $\varepsilon_i = h_i$), and that

$$\sum_{i}^{i-N} \varepsilon_i = \epsilon = U_r \qquad (17)$$

$$\sum_{i}^{i-N} \gamma_i = \Phi \quad \text{for all } i \geq 0 \qquad (18)$$

where Φ is the total time units (or channel capacity or bandwidth) used by the synchronous traffic per cycle. ε_i is the parameter that will be used to vary the load level of the synchronous traffic.

When node i releases the token, it reaches the next node (i+1) mod N after extra propagation delay time, w_i. Let t_{i+1} denote the time the token reaches node (i+1) mod N, then we have:

$$t_{i+1} \leq t_i + \delta_i + \gamma_i + w_i \qquad (19)$$

Therefore,

$$t_{i+1} \leq t_i + \delta_i + \alpha_i - \varepsilon_i \qquad (20)$$

Then, applying Eq16 into Eq20 and we replace U_r with ϵ from Eq17 , we have

$$t_{i+1} \leq \max(t_i, t_{i-N} + \tau - \epsilon) + (\alpha_i - \varepsilon_i) \text{ for all } i \geq 0 \qquad (21)$$

where $\alpha_i - \varepsilon_i = \alpha_{(i \bmod N)} - \varepsilon_{(i \bmod N)}$

In a special case where $\gamma_i = 0$ for all $i \geq 0$; let τ' be τ and t'_i be t_i for the special case. The initial condition for Eq21 and for the special case is assumed to be $t_0 = t'_0 = 0$ (Joseph et al., 1989) and (Ozuomba et al., 2003). Then, for the special case we define τ' and t'_i as follows

$\tau' = \tau - (T - \epsilon)$ (22)

$$t'_i = t_i + W - \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \qquad (23)$$

Eq23 differs from the concept used in (Joseph et al., 1989) and (Ozuomba et al., 2003) where $t'_i = t_i - \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j)$. Recall that in the initialization cycle no data is transmitted, however, the cycle time is equal to the total walk time or propagation delay per cycle, W . The addition of W to t_i in Eq23 is to account for the propagation delay in the initialization cycle, since by (Joseph et al., 1989) and (Ozuomba et al., 2003) $t_0 = t'_0 = 0$ whereas transmission of data in the protocol begins at an assumed time, t_i, after the initialization cycle which is after, at least, W time units. From Eq23 , t_i can be expressed as

$$t_i = t'_i - W + \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \quad (24)$$

Eq21 for the special case is

$$t'_{i+1} \leq max(t'_i, t'_{i-N} + \tau' - \epsilon) \text{ where } \varepsilon_i = h_i \text{ for all } i \geq 0 \quad (25)$$

The initial condition for Eq21 and Eq25 is assumed to be 0 that is (Joseph et al., 1989) and (Ozuomba et al., 2003).

$$t_i = t'_i = 0 \quad \text{where } t_i < 0,\ t'_i < 0 \text{ for all } i < 0 \quad (26)$$

Iterating Eq21 from i=0 to i=N-1 gives

$$0 \leq t_i \leq \tau - \epsilon + \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \quad \text{for } 1 \leq i \leq N \quad (27)$$

Substituting t_i from Eq27 into Eq23 gives

$$0 \leq t'_i \leq \tau - \epsilon + W \quad \text{for } 1 \leq i \leq N \quad (28)$$

Using τ as the upper bound on t'_i and H as the upper bound on ϵ for $0 \leq i \leq N$, induction over i can be applied to Eq25 to give

$$t'_i \leq \tau - H + \left\lfloor \frac{i}{N+1} \right\rfloor (\tau' - \epsilon) \quad (29)$$

Substituting t'_i from Eq29 into Eq24 gives

$$t_i \leq \tau - H - W + \left\lfloor \frac{i}{N+1} \right\rfloor (\tau' - \epsilon) + \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \quad (30)$$

where

$$\sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) = \left\lfloor \frac{i-1}{N+1} \right\rfloor (T - \epsilon) + \sum_{j=0}^{j=(i-1) \bmod N} (\alpha_j - \varepsilon_j) \quad (31)$$

Substituting τ' from Eq22 into Eq30 and also applying Eq31 into Eq30 gives

$$t_i \leq \tau - (H + W) + \left\lfloor \frac{i}{N+1} \right\rfloor (\tau - (T - \epsilon) - \epsilon) - \left\lfloor \frac{i-1}{N+1} \right\rfloor (T - \epsilon) + \sum_{j=0}^{j=(i-1) \bmod N} (\alpha_j - \varepsilon_j) \quad (32a)$$

$$t_i \leq (\tau - T)(1 + \left\lfloor \frac{i}{N+1} \right\rfloor) + \left\lfloor \frac{i-1}{N+1} \right\rfloor (T - \epsilon) + \sum_{j=0}^{j=(i-1) \bmod N} (\alpha_j - \varepsilon_j) \quad (32b)$$

The time, t_{Nk} at which station zero receives its k[th] token (that is i = Nk in Eq32b) is given as

$$t_{Nk} \leq (\tau - T)(1 + \left\lfloor \frac{NK}{N+1} \right\rfloor) + k(T - \epsilon) \quad (33)$$

Upper Bound On Cycle Length, $max(t_i - t_{i-N})$:

If k=1, and knowing the T = W +H, Eq33 gives

$$t_N \leq \tau - \epsilon \quad (34)$$

$$max(t_i - t_{i-N}) = t_N - t_0 = \tau - \epsilon \quad (35)$$

Eq35 gives the maximum cycle length for the Timely Token protocol (Jorge et al., 2004) with variable load level of synchronous traffic (where $\epsilon \geq 0$.) but with heavy load of asynchronous traffic. When $\epsilon = 0$, that is heavy load of synchronous traffic, Eq35 gives

$$max(t_i - t_{i-N}) = t_N - t_0 = \tau \text{ when } \epsilon = 0 \quad (36)$$

Average Cycle Length ($\hat{C}$): $\hat{C}$ is given as;

$$\hat{C} \leq \lim_{k \to \infty} (t_{Nk}/k) \leq \left\lfloor \frac{N}{N+1} \right\rfloor (\tau - T) + (T - \epsilon) \quad (37)$$

Eq37 gives the average cycle length for the Timely Token protocol with variable load level of synchronous traffic (where $\epsilon \geq 0$,) but with heavy load of asynchronous traffic. When $\epsilon = 0$, that is heavy load of synchronous traffic, Eq38 gives

$$\hat{C} \leq \left\lfloor \frac{N}{N+1} \right\rfloor (\tau - T) + T \qquad (38)$$

Average Time Used By The Asynchronous Traffic Per Cycle ($\hat{A}$):

From Eq37 and Eq38 we have,

$$\hat{A} = \left\lfloor \frac{N}{N+1} \right\rfloor (\tau - T) \qquad (39)$$

Eq39 gives the average time used by the asynchronous traffic per cycle for the Timely Token protocol with variable load level of synchronous traffic (where $\epsilon \geq 0$,) but with heavy load of asynchronous traffic. Eq39 also applies to the situation where $\epsilon = 0$, that is heavy load of synchronous traffic.

3.2 Expressions For Relating Simulation Results To Performance Parameters

Simulation of the Timely Token protocol, **(Protocol Q)** MAC algorithm was carried out with a program written with Visual Basic for Applications (VBA). The program runs in Microsoft office Excel 2007 environment. Meanwhile, we derived mathematical expressions for comparing the simulation result with the results obtained from the analytical computations. The mathematical expressions for the key parameters are presented as follows.

The Average Cycle Length ($\hat{C}$):

Simulation results of **Protocol Q** showed that if the values of N, T, τ and Φ remain constant for at least M consecutive cycles where $M \geq N+1$, then $\hat{C}$ is given as the MEAN value of the actual Token Rotation Time, $TRT_i^{\#}$ in any given node i. $TRT_i^{\#}$ can also be represented as $TRT_{j,k}^{\#}$ ($TRT_{j,k}^{\#}$ stands for Token Rotation Time, of node j in cycle k) where

$$k = \left\lfloor \frac{i}{N} \right\rfloor \qquad (40a)$$

and

$$j = i \bmod N \qquad (40b)$$

Equally, for any given N, j and k, we have that i is given as $\quad i = N * k + j \qquad (40c)$

$TRT_i^{\#}$ is given as,

$$TRT_i^{\#} = TRT_{j,k}^{\#} = t_i - t_{i\text{-}N} = \mathbf{TRT_i} \qquad (41)$$

where $\mathbf{TRT_i} \leq \tau$

The MEAN is taken over the M consecutive cycles as follows:

$$MEAN(TRT_i^{\#}) = \left(\frac{1}{M+1}\right) \sum_{k}^{k+M} (TRT_{j,k}^{\#}) \quad \text{where } k \geq 1; \; M \geq N \qquad (42)$$

$MEAN(TRT_i^{\#})$ approaches $\hat{C}$ when M >>N.

Note, $MEAN(TRT_i^{\#}) = MEAN(TRT_{j,k}^{\#})$, we may choose to work in terms of i or in terms of k and j.

The Maximum Cycle Length:

$$\textit{Maximum Cycle Length} = MAX(TRT_i^{\#}) \qquad \text{for all } i \geq 0 \qquad (43)$$

Note, ***Maximum Cycle Length*** = ***max***($t_i - t_{i\text{-}N}$) when we consider all $i \geq 0$

Upper Bound On The Average Bandwidth Allocated To The Asynchronous Traffic, ($\hat{A}$):

Let the average time units used by the asynchronous traffic per cycle be denoted as MEAN(δ_i) or MEAN($\delta_{j,k}$). The value of MEAN (δ_i) is defined in terms of $\delta_{j,k}$ (where $\delta_{j,k}$ means the time units used by the asynchronous traffic in node j in cycle k.) . We have ;

$$MEAN(\delta_i) = (1/(M+1)) \{ \sum_{k}^{k+M} (\sum_{j=0}^{j=N-1} (\delta_{j,k})) \} \text{where } k \geq 1 \,;\, M \geq N; \, i = N * k + j \qquad (44)$$

MEAN(δ_i) approaches $\hat{A}$ when M >>N.

4.0 Worked Examples and Discussion of Results

The analytical results obtained for **Protocol Q** , namely; **Ĉ**, **Á**, and **max**(t_i - t_{i-N}) must correspond with those obtained from the simulation of **Protocol Q,** namely MEAN($TRT_{j,k}$), MEAN(δ_j) and MAX($TRT_{j,k}$), respectively. In this section, we will provide a worked example that provides a table for the output of the simulation result of **Protocol Q**. We will also provide the results obtained from the analytical formulas for **Ĉ**, **Á**, and **max**(t_i - t_{i-N}) . Finally, graphs of the results will be plotted to discuss the performance of the Protocol Q under various load levels of the synchronous and asynchronous traffic.

4.1 Worked Example

4.1.1 The Sample Network

Consider a ring network with four stations (i.e N = 4). The ring uses the Timely-Token protocol for its MAC where the Timed-Token parameters are : τ = 100, $\mathbf{w}_i$ =1 for all the nodes, $\mathbf{h}_i$=20 for all the nodes. We will assume that the network has variable load level of the synchronous traffic and that out of the $\mathbf{h}_i$=20 reserved for synchronous traffic in every node, a constant value of ε_i = 18 is not used by the synchronous traffic in each node in every token receipt. The simulation result for Protocol Q is shown in Table 1a and Table 1b.

Table 1a: Simulation Results of the Timely-Token Algorithm (***Protocol Q***)

Cycle k	Node 0				Node 1				Node 2				Node 3			
	TRT_i	THT_i	γ_i	ε_i	TRT_i	THT_i	γ_i	ε_i	TRT_i	THT_i	γ_i	ε_i	TRT_i	THT_i	γ_i	ε_i
0	0	0	0	20	0	0	0	20	0	0	0	20	0	0	0	20
1	4	16	2	18	22	0	2	18	24	0	2	18	26	0	2	18
2	28	0	2	18	12	16	2	18	28	0	2	18	28	0	2	18
3	28	0	2	18	28	0	2	18	12	16	2	18	28	0	2	18
4	28	0	2	18	28	0	2	18	28	0	2	18	12	16	2	18
5	28	0	2	18	28	0	2	18	28	0	2	18	28	0	2	18
6	12	16	2	18	28	0	2	18	28	0	2	18	28	0	2	18
7	28	0	2	18	12	16	2	18	28	0	2	18	28	0	2	18
8	28	0	2	18	28	0	2	18	12	16	2	18	28	0	2	18
9	28	0	2	18	28	0	2	18	28	0	2	18	12	16	2	18
10	28	0	2	18	28	0	2	18	28	0	2	18	28	0	2	18
11	12	16	2	18	28	0	2	18	28	0	2	18	28	0	2	18
12	28	0	2	18	12	16	2	18	28	0	2	18	28	0	2	18
13	28	0	2	18	28	0	2	18	12	16	2	18	28	0	2	18
14	28	0	2	18	28	0	2	18	28	0	2	18	12	16	2	18
15	28	0	2	18	28	0	2	18	28	0	2	18	28	0	2	18
N =4; w_i = 1, W =4 ; h_i = 20 , H = 80; TTRT = τ=100 ; ε_i = 18, ϵ = 72																

Table 1b: Timely-Token Performance Based on the Simulation Results of Table 1a

				Ĉ	ϵ	Φ	Á
TRT_0#	$\sum\varepsilon_i$	$\sum\gamma_i$	$\sum\delta_i$	**Mean(TRT_0#)**	**Mean($\sum\varepsilon_i$)**	**Mean($\sum\gamma_i$)**	**Mean($\sum\delta_i$)**
3	80	0	0	3.0	0.0	0.0	0.0
26	72	8	16	14.5	76.0	4.0	8.0
28	72	8	16	19.0	74.7	5.3	10.7
28	72	8	16	21.3	74.0	6.0	12.0
12	72	8	16	19.4	73.6	6.4	12.8
28	72	8	0	24.4	72.0	8.0	12.8

28	72	8	16	24.8	72.0	8.0	12.8
28	72	8	16	24.8	72.0	8.0	12.8
28	72	8	16	24.8	72.0	8.0	12.8
12	72	8	16	24.8	72.0	8.0	12.8
28	72	8	0	24.8	72.0	8.0	12.8
28	72	8	16	24.8	72.0	8.0	12.8
28	72	8	16	24.8	72.0	8.0	12.8
28	72	8	16	24.8	72.0	8.0	12.8
N =4, w_i = 1, W =4, h_i = 20 , H = 80, TTRT = τ=100 , ε_i = 18, ϵ = 72							

The items in the Table 1a and Table 1b are : $TRT_o^{\#}$ is the token rotation time of node 0. MEAN($TRT_o^{\#}$) is the mean of $\sum TRT_o^{\#}$ taken over N+1 cycles except for the first four rows in Table 1a and Table 1b. $\sum\gamma_i$ is the total time units used by the synchronous traffic per cycle . MEAN($\sum\gamma_i$) is the mean of $\sum\gamma_i$ taken over N+1 cycles except for the first four rows in Table 1a and Table 1b. $\sum\varepsilon_i$ is the total of the time units reserved for the synchronous traffic per cycle but are not used by the synchronous traffic. MEAN($\sum\varepsilon_i$) is the mean of $\sum\varepsilon_i$ taken over N+1 cycles except for the first four rows in Table 1a and Table 1b.$\sum\delta_i$ is the total time units used by the asynchronous traffic per cycle . MEAN($\sum\delta_i$) is the mean of $\sum\delta_i$ taken over N+1 cycles except for the first four rows in Table 1a and Table 1b.

Similarly, the result obtained from the analytical computation of the performance of Protocol **Q** for various network configurations are presented in Table 2 and Fig. 1.

Table 2 The Results of the Analytical Computations for the Timely-Token Algorithm (Protocol **Q**)

ε_i	γ_i	ϵ	$max(t_i - t_{i-N})$	Φ =(H-ϵ)	$\acute{A}$	$\hat{C}$
0	20	0	100	80	12.8	96.8
2	18	8	92	72	12.8	88.8
4	16	16	84	64	12.8	80.8
6	14	24	76	56	12.8	72.8
8	12	32	68	48	12.8	64.8
10	10	40	60	40	12.8	56.8
12	8	48	52	32	12.8	48.8
14	6	56	44	24	12.8	40.8
16	4	64	36	16	12.8	32.8
18	2	72	28	8	12.8	24.8
20	0	80	20	0	12.8	16.8
N =4, w_i = 1, W =4, h_i = 20 , H = 80, TTRT = τ=100						

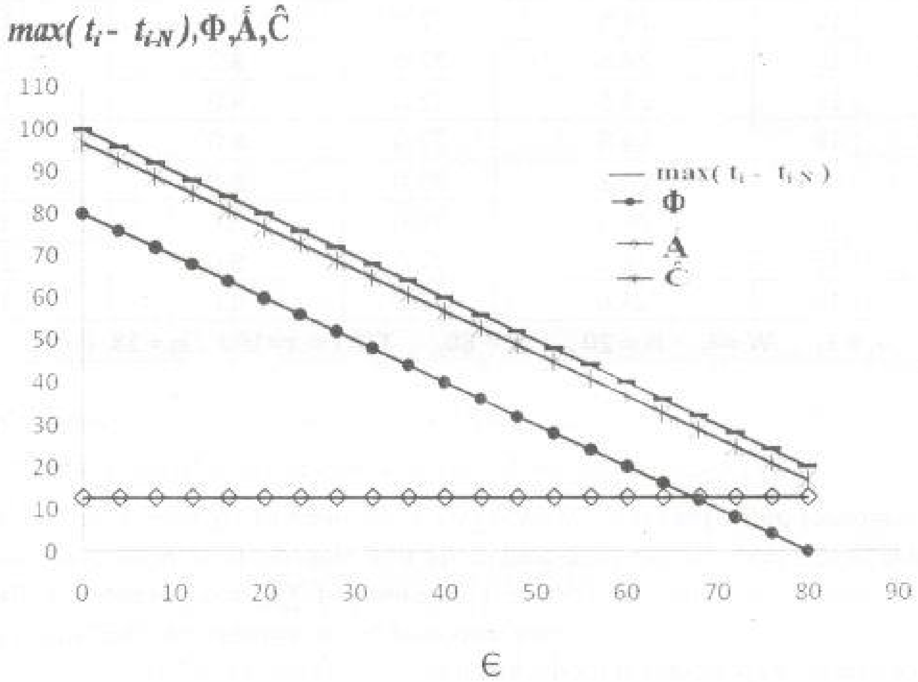

Fig 1 Graph of Å, **Φ**, Ĉ and **max(** t_i - t_{i-N}) against ϵ for Timely-Token Protocol (Table 2 plot)

4.2 Discussion of Result

First, from Table 1a and Table 1b, with ε_i = 18, ϵ =72 , then we have Mean(TRT$_0$#)= 24.8 and MEAN(δ_i) = 12.8 (when the mean is taken over N+1 cycles with constant value of ϵ=72). MAX(TRT$_0$#) = 28 (see row 11 column 2 of Table 1a, and row 11 column 1 of Table 1b) where the cycle is 2 and the node considered is node 0. The same results are obtained from the analytical result shown in Table2 (row 11 column 4 of Table 2); with ε_i = 18, ϵ =72 , then we have,
Ĉ = 24.8 = MEAN(TRT$_0$#) (row 11 column 4 of Table 1b and row 11 column 7 of Table 2);
Å = 12.8 = MEAN(δ_i) (row 11 column 8 of Table 1b and row 11 column 6 of Table 2);
max(t_i - t_{i-N}) = 28 = MAX(TRT$_0$#) (row 11 column 2 of Table 1a and row 11 column 4 of Table 2).

Secondly, at any given instance, the analytical results showed that a total of (τ - **T**) time units are available for the asynchronous traffic. However, not all of the available time is used in every cycle even when there is heavy load of asynchronous traffic. Only $\left[\frac{N}{N+1}\right](\tau - \mathbf{T})$ time units are used by the asynchronous traffic, leaving $\left[\frac{1}{N+1}\right](\tau - \mathbf{T})$ time units unused. This is captured in Table 2 (column 6 of Table 2);

Thirdly, the maximum time units that are used by the asynchronous traffic is not affected by fluctuations in the load levels of the synchronous traffic. This is captured in Table 2 (column 6 of Table 2) and Fig 1 where Å = 12.8 irrespective of the variation in the value of ϵ.

Fourthly, Table 2 and graph of Fig1 show that when $\boldsymbol{\varepsilon_i = h_i = 20}$, $\boldsymbol{\epsilon}$**= H= 80** , $\boldsymbol{\gamma_i = 0}$ and **Φ =0** thus, there is no synchronous traffic in the network. In this case, where there is only asynchronous traffic, in Table 2 (row 12 column 4 of Table 2), Maximum cycle length =20 , Ĉ = $\left[\frac{N}{N+1}\right](\tau - \mathbf{T}) + \mathbf{W}$ = 16.8, Å = $\left[\frac{N}{N+1}\right](\tau - \mathbf{T})$ = 12.8. In this case, the asynchronous traffic is (starved of or) denied access to 100 - 4 - 12.8 = 83.2 (i.e. τ- **W** - **Å**) time units available to it.

Finally, the Timely-Token protocol was able to eliminate the problem of token-lateness (i.e for all i > 0 **max(** t_i - t_{i-N}) ≤ τ) as well as doubling the available bandwidth for supporting synchronous traffic. However, channel capacity allocation mechanism employed for the asynchronous traffic does not permit the asynchronous traffic to use as much channel capacity as are available to it.

5.0 Conclusion and Recommendations

5.1 Conclusion

In this paper, we presented a performance analysis of the Timed-Token Protocol in the Timely-Token Protocol under light load of synchronous traffic and heavy load of asynchronous traffic. We derived analytical expression for the exact throughput of the Timely Token MAC algorithm for the asynchronous traffic under varying load levels of the synchronous traffic. The results give more insights into the performance of the protocol under varying load conditions of the synchronous traffic; from no (zero) load of synchronous traffic to heavy or full load of synchronous traffic. This will enable researchers to further understand how the Timely-Token Protocol supports the integrated services of synchronous and asynchronous traffic, and how each traffic class can affect the system performance. Specifically, Timely-Token Protocol was able to eliminate the problem of token-lateness as well as doubling the available bandwidth for supporting synchronous traffic. However, as the load level of the synchronous traffic drops, channel capacity allocation mechanism employed for the asynchronous traffic in the Timely Token protocol does not permit the asynchronous traffic to use as much channel capacity as are available to it. As such, the channel capacity allocation mechanism employed for the asynchronous traffic in the Timely-Token Protocol can still be improved upon for supporting asynchronous traffic as well as integrated services consisting of the synchronous and asynchronous traffic.

5.2 Recommendations

The analysis in this paper is for the Timed-Token Protocols in the Timely-Token Protocols. There is a later version of the Timed-Token protocol, the Budget Sharing Token (BuST) protocol (Franchino et al., 2008) and (Franchino et al., 2007) with claims of improved performance. This recent protocol achieved the improved performance at the expense of the channel bandwidth (capacity) allocated to the asynchronous traffic. This common problem can be solved if the analysis we have presented in this paper is employed in studying the performance of Budget Sharing Token (BuST) protocol. Furthermore, a comparative study of the performance of the various versions of the Timed-Token protocols with variable load levels of synchronous traffic will provide the much needed clues on how to develop a Timed-Token protocol with optimum bandwidth allocation to the asynchronous traffic and without the problem of token-lateness.

References

Biao Chen and Wei Zhao (1992), "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

Chan E., Chen D., Cao J. and Lee C. H.,(1999), " The time Properties of the FDDI-M Medium Access Control Protocol" The Computer Journal, Vol. 82 No. 1 , pp. 96-102, Jan. 1999.

Cobb J. and Lin M. (2002), "The On-Time timed-token protocol", Proc. IEEE GLOBECOM, 2002.

Franchino G., Buttazzo G. C., and Facchinetti T. (2007), "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.

Franchino G., Buttazzo G. C., and Facchinetti T. (2008), "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

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

Grow R.,(1982), "A timed token protocol for local area networks", Proc. Electro'82, Token Access Protocols, Paper 17/3, May 1982.

Jorge Arturo Cobb, Miaohua Lin(2004), " The Timely-Token protocol," Computer Communications Vol .27 no.7, pp. 569-580 May, 2004

Joseph W. M. and Fouad A. T.(1989), " Throughput Analysis of a Timer-Controlled Token Passing Protocol Under Heavy Load", IEEE trans. Comm, vol 37, No7 pp694 – 702, July 1989

Kenneth C. Sevcik and Marjory J. Johnson,(1987), "Cycle time properties of the FDDI token ring protocol," IEEE Trans. Software. Eng., vol. SE-13, pp. 376-385, Mar. 1987.

Mcguffin L.J., Reid L.O, and Sparks S.R.,(1988) "MAP/TOP in CIM Distributed Computing, " IEEE Network, vol.2 no. 3 May 1988, pp. 23 – 31.

Nicholas Malcolm, Wei Zhao, (1994),"The timed-token protocol for real-time communications," Computer, vol. 27, no. 1, pp. 35-41, January, 1994

Ozuomba S. and Chukwudebe G.A, (2003),"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 .

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, Curitiba, Brazil, 2006

Shin K. G. and Zheng Q.(1995),"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.

Survivable Adaptable Fibre-Optic Embedded Networks, MIL- STD-2004, US Dept. of Defense, Washington D.C., Sept, 1992

The Institute of Electrical and Electronic Engineers, (1985) " Token-passing bus access method and physical layer specifications , " America national Statndard ANSI/IEEE std. 802.4 – 1985

Tovar E., Vasques F.,(1998) "Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications", Proc. of the 15th IFAC Workshop on Distributed Computer Control Systems. 1998

Uhlhorn R.W., (1991),"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