

ANALYSIS OF TIMELY-TOKEN PROTOCOL WITH NON-UNIFORM HEAVY LOAD OF ASYNCHRONOUS TRAFFIC

¹C. Kalu, ¹S. Ozuomba,, ²G. N. Onoh

¹Dept. of Elect/Elect & Computer Engineering, University of Uyo, Akwa Ibom,

²Dept. of Elect/Elect Engineering, Enugu State University of Science and Technology

Abstract

In this paper, the timely-token protocol is studied under non-uniformly heavy load of asynchronous traffic and problems were identified. The problems are due to inappropriate definition of heavily loaded networks. Specifically, the paper presented a more comprehensive definition of a system that is heavily loaded with asynchronous traffic. Furthermore, the performance analysis is conducted to account for the two extreme cases; where every node is heavily loaded and another case, where only few nodes are heavily loaded while the rest are not loaded at all. The discovery in this paper is very essential to network designers and researcher as they strive to improve the performance of the timed token protocols under various network traffic configurations.

Keywords: *Timely-Token Protocol, Best-Effort Service, Bandwidth Reclaiming Mechanism, Asynchronous Traffic, Synchronous Traffic.*

1.0 Introduction

More accurately, it has been stated that the most promising solutions to support real-time services in communication networks are those that allow the coexistence of both real-time and non-real-time stations in the same communication domain [1], [2]. For non real-time domain, maximizing the throughput or minimizing the average message delay is the most important performance criteria. In the hard real-time domain, however, concern focuses on satisfying the time constraints of individual messages [3-5]. Hence, for hard real-time system, predictability is favoured against maximizing throughput. The situation placed a demand for effective and efficient integrated services Local Area Network (LAN). Such networks' Media Access Control (MAC) protocols must deal with different traffic patterns and must provide not only bounded message transmission time, as required by the hard real-time and soft real-time tasks, but also high throughput, as demanded by non-real-time tasks [6]. An attractive MAC approach for integrating such traffic in multi-access networks is the timed-token protocol. Consequently, the timed-token protocol has been incorporated into several

high-bandwidth network standards [7], such as, IEEE802.4 Token Bus [8], Fiber Distributed Data Interface (FDDI) [9-13], SAFENET [14], Manufacturing Automation Protocol (MAP) [15], High-Speed Ring Bus [16], and in PROFIBUS which is a Fieldbus network standard [17]. Several implementations of token passing strategies in wireless networks have also been proposed in [18-20].

Despite the wide applications of timed-token protocols, poor management of unused (spare) channel capacity during network operations continue to pose performance problems like token lateness, starvation of the asynchronous traffic, inefficient capacity utilizations [21]. The timely-token protocol is a version of the timed-token protocol developed by Cobb and Lin to improve the communication services provided by the existing timed-token protocols [21]. It solved the problems of token-lateness in FDDI [9] (Biao and Wei 1992) and starvation of asynchronous traffic in FDDI-M [10]. However, the timely-token protocol still suffers inefficient management of the available bandwidth. One of such situations where the inefficient management of the available bandwidth is

evident is the occasions where the network is non-uniformly heavily loaded. In this paper, the timely token protocol is studied under nonuniform heavy load situations and mathematical expressions were derived to determine the exact amount of drop in throughput of the system in nonuniform heavy load situations.

The rest of the paper is organized as follows. The network and message models are presented in Section 2 along with the timely-token protocol Media Access Control (MAC) algorithm and flowcharts. Analysis of the timely-token protocol MAC algorithm for a non-uniformly heavily loaded system is presented in Section 3. Section 4 contains numerical example for the performance of the timely-token protocol when it is non-uniformly heavily loaded. Finally, concluding remarks and recommendations for further studies are stated in Section 5.

2.0 Network Characteristics

2.1 Network Model

The study considered a token ring or logical ring network consisting of N nodes (or stations). Each node has a unique number in the range $0, 1, 2 \dots N-1$. In addition, each node is connected to two other neighbouring 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 $(i+1) \bmod N$.

A special bit pattern called the token circulates around the ring (logical ring or token ring) from node i to nodes $i+1, i+2, \dots$ until node $i+(N-1)$, then to nodes $i, i+1, i+2, \dots$, 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 $(i+1)$. Then ,

$$W = \sum_{j=0}^{N-1} w_j \quad \dots(1)$$

W denotes the token walk time around the ring when none of the nodes in the network disturb it [9], [22].

2.1.1 Message Model

Messages generated in the network at run time may be classified as either synchronous messages or asynchronous messages. In the following discussion it is assumed that there is one stream of synchronous messages on each node [23], [9]. It is also assumed that the network is free from hardware or software failures. Hence, in the N -node ring, the synchronous message set, M , consist of N streams of synchronous messages; $s_0, s_1, s_2 \dots s_{N-1}$ where, $M = \{s_0, s_1, s_2 \dots s_{N-1}\}$

The synchronous message stream, s_i at node i is given as $s_i = \{P_i, C_i, D_i\}$ where, c_i is the maximum amount of time needed to transmit a message in the stream at node i ; P_i is the period length of stream s_i i.e. the minimum inter-arrival period for the message stream at node i ; D_i is the relative deadline for the message stream s_i at node.

2.2 The Timed-Token Medium Access Control (MAC) Protocol Parameters

The timed-token protocol uses the following parameters for its operation [24].

- Target Token Rotation Time, (TTRT).** Let τ denote the value of TTRT [25], [24], [26].
- Synchronous Capacity of Node i (H_i).** H_i is the maximum time units allocated to node i to transmit its synchronous messages in each token receipt [25], [24], [26]. Then, the other related parameters are [25], [24], [26].

$$H = H_0 + H_1 + \dots H_{N-1} = \sum_{i=0}^{N-1} H_i \quad \dots(2)$$

$$H_i + w_i = \alpha_i \quad \dots(3)$$

$$\alpha_0 + \alpha_1 + \dots \alpha_{(N-1)} = T = W + H = \sum_{i=0}^{N-1} \alpha_i \quad \dots(4)$$

Let A be defined as the total time units available to the asynchronous traffic in every cycle, where

$$A = \tau - T \quad \dots(5)$$

Constraints:

The Protocol Constraint requires that [25], [24], [26]:

$$H + W \leq \tau \quad \dots(6)$$

The **Deadline Constraint** in the **Timely-Token** protocol requires that

$$\tau \leq \min_{i=0,1,\dots,N-1} (D_i) \quad \dots(7)$$

Combining Equations (6) and (7) gives

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

c) **Token Rotation Timer of Node i (TRT_i)** [26], [24]

d) **Token Holding Timer of Node i (THT_i)** [26].

2.3 Meaning of Heavily Loaded Nodes and Network

i. **Heavily Loaded Node:** a node is heavily loaded with asynchronous traffic, if within the period considered, in every token receipt with $THT_i > 0$, node i has as much asynchronous frames as will enable it to use up all the THT_i time units available to it to

deliver asynchronous frames. To simplify the analysis, it is assumed that a node is either heavily loaded or it is not loaded at all.

ii. **Number of Heavily Loaded Nodes (n):** n is the number of nodes that are heavily loaded with asynchronous traffic out of the N nodes in the network.

iii. **Uniform Heavy Loaded Network :** is a network in which $n = N$; that means, if all the nodes are heavily loaded with asynchronous traffic.

iv. **Non-Uniform Heavy Loaded Network:** is a network in which $1 \leq n < N$; that means, one or more of its nodes is not heavily loaded and at least one of its nodes is heavily loaded.

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

The timely-token media access control (MAC) algorithm developed by Cobb and Lin,[21] is also presented in flowchart by Ozuomba et. al,[25]. The timely-token MAC algorithm is modified here to account for the number of heavily loaded nodes in the system; the modified timely-token MAC algorithm is presented as MAC Algorithm Q.

Algorithm Q: The Modified Timely-Token MAC Algorithm: ALGORITHM Q INITIALIZATION CYCLE

Q1.1 Define TRT (that is τ) and N

Q1.3 Define H_i for $i = 0, 1, \dots, N-1$

Q1.5 Compute $\varepsilon = \left(\sum_{i=0}^{N-1} (\varepsilon_i) \right)$

Q1.7 Initialise $B_i = 1$ for $i = 0, 1, \dots, N-1$

Q1.9 Initialise Timer Q1.10.1 $i = 0$

Q1.10.3 Start TRT_i ; TRT_i Counts up

Q1.10.5 Pass the Token to Node $i + 1$

Q1.2 Define w_i for $i = 0, 1, \dots, N-1$

Q1.4 Initialise $\varepsilon_i = H_i$ for $i = 0, 1, \dots, N-1$

Q1.6 Compute $T = \left(\sum_{i=0}^{N-1} (H_i + w_i) \right)$

Q1.8 Let $n = N$

Q1.10.2 $TRT_i = 0$

Q1.10.4 $i = i + 1$

Q1.10.6 IF ($i < N$) Then Goto Step Q1.10.2

Else Goto Step Q2.1 End if.

DATA TRANSMISSION CYCLE PART I :TRANSMISSION OF SYNCHRONOUS FRAMES

Q2.1 Check Frames that arrives at Node i

Q2.2 IF (Frames is Token) Then Goto Step Q2.5 Else Goto Step Q2.3 End if
 Q2.3 Process Frame (Store, Ignore, etc.)
 Q2.5 $THT_i = TTRT - \epsilon - TRT_i$
 Q2.7.1 $TRT_i = 0$
 Q2.7.2 Start TRT_i ; TRT_i counts up
 Q2.8 IF ($TRT_i < H_i$) Then Goto Step Q2.9 Else Goto Step Q2.12 End if
 Q2.9 IF (Synchronous Frames Ready) Then Goto Step Q2.10 Else Goto Step Q2.11 End if
 Q2.10 Transmit Synchronous Frames
 Q2.11 Goto Step Q2.8
 Q2.12 $h_i = TRT_i$
 Q2.13 $\epsilon_i = H_i + h_i$
 Q2.14 $\epsilon = \epsilon + \epsilon_i$

DATA TRANSMISSION CYCLE PART II TRANSMISSION OF ASYNCHRONOUS FRAMES

Q3.1 $a_i = THT_i$
 Q3.2 Start THT_i ; THT_i counts down
 Q3.3 IF ($THT_i > 0$) Then {
 IF (Asynchronous Frames are Available) Then {IF ($B_i = 0$) Then { $B_i = 1$; $n = n + 1$ }
 Endif}
 Else { IF ($B_i = 1$) Then { $B_i = 0$; $n = n - 1$ } Endif } Endif }
 Q3.4 IF ($THT_i > 0$) Then Goto Step Q3.5 Else Goto Step Q3.6 Endif
 Q3.5 IF (Asynchronous Frames are Available) Then
 {Transmit Asynchronous Frame ; Goto Step Q3.4} Endif
 Q3.6 $a_i = a_i - THT_i$
 Q3.7 $i = i + 1$
 Q3.8 $i = (i \bmod N)$
 Q3.9 Pass the Token to Node i

3.0 Performance Analysis of the Timely-Token Protocol Media Access Control (MAC) Algorithm

Let $t_0, t_1, \dots, t_{(N-1)}$ be the time at which the token reaches station $0, 1, \dots, N-1$ for some given cycle. Also, let $t_N, t_{(N+1)}, \dots, t_{(2N-1)}$ be the time at which the token reaches station $0, 1, \dots, N-1$ in the next cycle and so forth. Let $t_N, t_{(1-N)}, \dots, t_{(N-1)-N}$ be the time at which the token reaches station $0, 1, \dots, N-1$ in the previous cycle to the given cycle. Now, 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} . Hence, the cycle length or the time between two consecutive token receipts at node i is given as;

$$\text{Cycle length for node } i = t_i - t_{i-N} \quad \dots(9)$$

For any given value of i , the node denoted by j (where $j = 0, 1, 2, \dots, N-1$) and the cycle denoted by k can be computed as follows:

$$j = (i \bmod N) \quad \dots(10)$$

$$k = \left\lfloor \frac{i}{N} \right\rfloor \quad \dots(11)$$

According to the protocol operations in Q 2.5, Q3.1 to Q3.6, when the token arrives at node i ,

$$\text{then } THT_i = TTRT - \epsilon - TRT_i \text{ for } TTRT - \epsilon > TRT_i \quad \dots(12)$$

$$\text{Otherwise, } THT_i = 0 \text{ for } TTRT - \epsilon = TRT_i \quad \dots(13)$$

$$\text{Hence, } THT_i \leq \max(0, TTRT - \epsilon - TRT_i) \text{ for all } TRT_i \quad \dots(14)$$

According to the protocol operations in Q3.1, Q3.5 and Q3.6;

$$a_i \leq THT_i \quad \dots(15)$$

For a system that is heavily loaded with asynchronous traffic, it is assumed that there is always sufficient asynchronous traffic in every cycle to use all the time available to the asynchronous traffic. Recall that TRT_i is the same thing as $t_i - t_{i-N}$, thus,

$$a_i \leq \max(0, \tau - \epsilon - (t_i - t_{i-N})) \text{ for all } (t_i - t_{i-N}) \quad \dots(16)$$

If the token reaches node i at time, t_i , then the token will reach node $i+1$ at t_{i+1} after transmitting h_i time units of synchronous traffic and a_i time units of

asynchronous traffic along with a token walk-time, w_i . Thus

$$t_{i+1} \leq t_i + a_i + h_i + w_i \quad \dots(17)$$

$$t_{i+1} \leq \max(t_i, t_{i-N} + \tau - \varepsilon) + (\sigma_i - \varepsilon_i) \quad \dots(18)$$

for all $i \geq 0$

where $\sigma_i - \varepsilon_i = \sigma_{(i \bmod N)} - \varepsilon_{(i \bmod N)}$
To properly address the peculiar nature of the initialization cycle, a special case is considered where $h_i = 0$ for all $i \geq 0$, that means $\varepsilon_i = H_i$ for all $i \geq 0$; let τ' be τ and t_i' be t_i for the special case [24], [22]. The initial condition for Eq18 and for the special case is assumed to be $t_0 = t_0' = 0$ [24], [22]. Then, for the special case τ' and t_i' are defined as follows:

$$\tau' = \tau - (T - \varepsilon) \quad \dots(19)$$

$$t_i' = t_i + W - \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) \quad \dots(20)$$

Equation (20) differs slightly from the concept used in [24], [22] where $t_i' = t_i - \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j)$. 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 which is given as $W = \sum_{j=0}^{N-1} (w_j)$. The addition of W to t_i in

Eq20 is to account for the propagation delay in the initialization cycle, since by [24], [22], $t_0 = t_0' = 0$ whereas transmission of data in the protocol begins at an assumed time, t_0 , after the initialization cycle which is after, at least, W time units. Alternatively, the initial conditions can be restated as follows $t_0 = W$, $t_0' = 0$ if the approach employed in [24], [22] is to be used. From equation (20), t_i can be expressed as

$$t_i = t_i' - W + \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) \quad \dots(21)$$

Eq18 for the special case is

$$t_{i+1}' \leq \max(t_i', t_{i-N}') + \tau' - \varepsilon \text{ where } h_i = 0 \text{ for all } i \geq 0 \quad \dots(22)$$

The initial condition for equation (18) and (22) is given as; [24], [22]

$$t_i = t_i' = 0 \text{ where } t_i < 0, \quad t_i' < 0 \text{ for all } i < 0 \quad \dots(23)$$

Iterating equation (18) from $i = 0$ to $i = N-1$ gives

$$0 \leq t_i \leq \tau - \varepsilon + \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) \text{ for } 1 \leq i \leq N \quad \dots(24)$$

Substituting t_i from equation (24) into (20) gives

$$0 \leq t_i' \leq \tau - \varepsilon + W \text{ for } 1 \leq i \leq N \quad \dots(25)$$

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 Eq22 to give

$$t_i' \leq \tau - H + \left\lfloor \frac{i}{n+1} \right\rfloor (\tau' - \varepsilon) \quad \dots(26)$$

The assumption that τ is the upper bound on t_i' in Eq(26) is valid if Eq(26), $\tau' = \tau - (T - \varepsilon) = \tau - T + \varepsilon$ and that $T = W + H$ are considered, thus $\tau' = \tau - W - H + \varepsilon$. Since $H \geq \varepsilon$ and $W \geq 0$, then $\tau' \leq \tau$.

Similarly, in the special case $h_i = 0$ for all $i \geq 0$, that means $\varepsilon_i = H_i$ us $\sum_{j=0}^{N-1} (\varepsilon_j) = \sum_{j=0}^{N-1} (H_j) = H$,

so the upper bound on ε is H . Substituting t_i' from Eq26 into Eq21 gives

$$t_i \leq \tau - H - W + \left\lfloor \frac{i}{n+1} \right\rfloor (\tau' - \varepsilon) + \left(\sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) \right) \quad \dots(27)$$

$$\text{where } \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) = \left\lfloor \frac{i-1}{n+1} \right\rfloor (T - \varepsilon) + \left(\sum_{j=0}^{(i-1) \bmod N} (\sigma_j - \varepsilon_j) \right) \quad \dots(28)$$

Substituting τ' from Eq19 into Eq27 and also applying Eq28 into Eq27 gives

$$T \leq \tau - (H + W) + \left\lfloor \frac{i}{n+1} \right\rfloor (\tau - (T - \varepsilon) - \varepsilon) - \left\lfloor \frac{i-1}{n+1} \right\rfloor (T - \varepsilon) + \left(\sum_{j=0}^{(i-1) \bmod N} (\sigma_j - \varepsilon_j) \right) \quad \dots(29)$$

If $H + W$ is replaced with T and $(\tau - (T - \varepsilon) - \varepsilon)$ with $\tau - T$, Eq29 can then give

$$t_i \leq \left(1 + \left\lfloor \frac{i}{n+1} \right\rfloor\right)(\tau - T) + \left\lfloor \frac{i-1}{n+1} \right\rfloor (T - \epsilon) + \left(\sum_{j=0}^{j=(i-1) \bmod N} (\sigma_j - \epsilon_j)\right) \quad \dots(30)$$

The time, t_{Nk} at which station zero receives its k^{th} token (that is $i = Nk$ in Eq30) is given as

$$t_{Nk} \leq \left(1 + \left\lfloor \frac{(n)K}{n+1} \right\rfloor\right)(\tau - T) + k(T - \epsilon) \quad \dots(31)$$

Note that Eq31 for t_{Nk} is different from t_{Nk} given in [24 and 22] where t_{Nk} is expressed in terms of

$$\left\lfloor \frac{NK}{N+1} \right\rfloor \cdot W + \left\lfloor \frac{NK}{n+1} \right\rfloor (T - \epsilon) + k(T - \epsilon)$$

The use of nK instead of NK is because; it is only the n heavily loaded nodes that transmit asynchronous frames whenever they receive the token. As such, nK rather than Nk is more appropriate for capturing the actual asynchronous frames that are delivered in any given cycle.

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

If $k=1$, Eq(31) gives $t_N \leq \tau - T + T - \epsilon$...(32)

Thus; $\max(t_i - t_{i-N}) = t_N - t_0 = \tau - \epsilon = \tau$ for all $i \geq 0$ and $\epsilon = 0$...(33)

Average Cycle Length (C)

Let C be the average cycle length, then from Eq31, C is given as

$$C \leq \lim_{k \rightarrow \infty} (t_{Nk}/k) \leq \left\lfloor \frac{n(\tau - T)}{n+1} \right\rfloor + (T - \epsilon) \quad \dots(34)$$

$$C \leq \left\lfloor \frac{n(\tau - T)}{n+1} \right\rfloor + (T - \epsilon) \quad \dots(35)$$

Average Time (Bandwidth) Used By The Asynchronous Traffic Per Cycle (A_v)

Let A_v denote the average (Channel Capacity) time used by the asynchronous

traffic per cycle. From Eq35, $A_v \leq \left\lfloor \frac{n(\tau - T)}{n+1} \right\rfloor$ where $1 \leq n \leq N$...(36)

4.0 Comparison Of Performance of The Timely-Token Protocols Under Nonuniform Heavy Load of Asynchronous Traffic

The Average Asynchronous Traffic Time Units Per Cycle (A_v)

On the other hand, it has been shown in [21] that for the Timely-Token protocol, the average Asynchronous Traffic (Capacity) Time Units Per Cycle for the Timely-Token protocol under uniform heavy load is given as;

$$A_v \leq \left\lfloor \frac{N(\tau - T)}{N+1} \right\rfloor; \text{ when } n = N \quad \dots(37)$$

Thus, the average Asynchronous Traffic derived by Cobb and Lin [21] is a special case of Eq30, that occurs when $n = N$.

a) The Average Cycle Length

It has been shown in [25], [21] that for the Timely-Token protocol, the **Average Cycle Length** for the Timely-Token protocol under uniform heavy load is given as;

$$C \leq \left\lfloor \frac{N(\tau - T)}{N+1} \right\rfloor + (T - \epsilon); \text{ when } n = N \quad \dots(38)$$

Thus, the **Average Cycle Length** derived by Cobb and Lin [21] is a special case of Eq35, that occurs when $n = N$.

4.1 Simulation of Protocol Q and Protocol P

The simulation of the MAC algorithms for the Timely-Token protocol was conducted with a program written with Visual Basic for Applications (VBA). The program runs in Microsoft Office Excel 2007 environment. The following mathematical expressions will be used to compare the simulation results with the results obtained from the analytical computations. For the simulation results, if the values of N , n , T , τ and ϵ remain constant for at least M consecutive cycles where $M \gg N$, then $MEAN(TRT_i)$ approaches $\hat{C}$ obtained

from the analytical computations where, $MEAN(TRT_{j,k}^*)$ is given as

$$MEAN(TRT_{j,k}^*) = \left(\frac{1}{(n+1)} \right) \left(\sum_{i=x}^{i=(x+n)} (TRT_{j,k}^*) \right) \text{ for } x > N \quad (39)$$

The average values are considered as from cycle $N + 1$ and the average values are taken for every set of $n + 1$ cycles.

Note that $TRT_{j,k}^*$ and a_i^* are values obtained from the simulation of the algorithm. Similarly,

$$MEAN(a_i^*) = \left(\frac{1}{(n+1)} \right) \left(\sum_{i=x}^{i=(x+n)} (a_i^*) \right) \text{ for } x > N \quad (40)$$

The Maximum Cycle Length is $MAX(TRT_{j,k}^*)$ for all $i \geq 0$

4.2 Numerical Examples

Consider a ring network with four stations ($N = 4$). The ring uses the Timely-Token protocol for its MAC where the timed-token parameters are given as follows: $TRT = \tau = 100$, $w_i = 1$ for all the nodes. $H = 40$, $\epsilon_i = 10$ for all the nodes.

As part of the simulation, for synchronous traffic in every node, a constant value of $\epsilon_i = 10$ is not used by the synchronous traffic in each node in every token receipt. With these given parameters $H = 4(10) = 40$. The simulation results for Protocol Q (Timely-Token protocol) for various values of n are shown in Table 1a to Table 1d

Table 1a: The simulation results of the Timely-Token protocol (Protocol Q) for $n = 1$; $\epsilon = 0$, $h = H = 40$.

	0		1		2		3			MEAN (ϵ_i)	MEAN (h_i)	MEAN (a_i)	MEAN (TRT_i)
k	TRTi	a_i	TRTi	A_i	TRTi	a_i	TRTi	a_i	n				
0	0	0	0	0	0	0	0	0	4	40.00	0.00	0.00	0.00
1	4	56	70	0	80	0	90	0	1	20.00	20.00	28.00	2.00
2	100	0	44	0	44	0	44	0	1	0.00	40.00	28.00	52.00
3	44	56	100	0	100	0	100	0	1	0.00	40.00	28.00	72.00
4	100	0	44	0	44	0	44	0	1	0.00	40.00	28.00	72.00
5	44	56	100	0	100	0	100	0	1	0.00	40.00	28.00	72.00
6	100	0	44	0	44	0	44	0	1	0.00	40.00	28.00	72.00
7	44	56	100	0	100	0	100	0	1	0.00	40.00	28.00	72.00
8	100	0	44	0	44	0	44	0	1	0.00	40.00	28.00	72.00
9	44	56	100	0	100	0	100	0	1	0.00	40.00	28.00	72.00

Table 1b: The simulation results of the Timely-Token protocol (Protocol Q) for $n = 2$; $\epsilon = 0$, $h = H = 40$.

	0		1		2		3			MEAN (ϵ_i)	MEAN (h_i)	MEAN (a_i)	MEAN (TRT_i)
k	TRTi	a_i	TRTi	a_i	TRTi	a_i	TRTi	a_i	n				
0	0	0	0	0	0	0	0	0	4	40.00	0.00	0.00	0.00
1	4	56	70	0	80	0	90	0	2	20.00	20.00	28.00	2.00
2	100	0	44	56	100	0	100	0	2	13.33	26.67	37.33	34.67
3	100	0	100	0	44	0	44	0	2	0.00	40.00	37.33	68.00
4	44	56	100	0	100	0	100	0	2	0.00	40.00	37.33	81.33
5	100	0	44	56	100	0	100	0	2	0.00	40.00	37.33	81.33
6	100	0	100	0	44	0	44	0	2	0.00	40.00	37.33	81.33
7	44	56	100	0	100	0	100	0	2	0.00	40.00	37.33	81.33

8	100	0	44	56	100	0	100	0	2	0.00	40.00	37.33	81.33
9	100	0	100	0	44	0	44	0	2	0.00	40.00	37.33	81.33

Table 1c: The simulation results of the Timely-Token protocol (**Protocol Q**) for $n = 3$; $\tau = 0$, $h = H = 40$.

k	0		1		2		3		n	MEAN (e_i)	MEAN (h_i)	MEAN (a_i)	MEAN (TRTi)
	TRTi	ai	TRTi	ai	TRTi	ai	TRTi	ai					
0	0	0	0	0	0	0	0	0	4	40.00	0.00	0.00	0.00
1	4	56	70	0	80	0	90	0	3	20.00	20.00	28.00	2.00

5	44	56	100	0	100	0	100	0	3	0.00	40.00	42.00	86.00
6	100	0	44	56	100	0	100	0	3	0.00	40.00	42.00	86.00
7	100	0	100	0	44	56	100	0	3	0.00	40.00	42.00	86.00
8	100	0	100	0	100	0	44	0	3	0.00	40.00	42.00	86.00
9	44	56	100	0	100	0	100	0	3	0.00	40.00	42.00	86.00

Table 1d: The simulation results of the Timely-Token protocol (**Protocol Q**) for $n = 4$; $\tau = 0$, $h = H = 40$.

k	0		1		2		3		n	MEAN (e_i)	MEAN (h_i)	MEAN (a_i)	MEAN (TRTi)
	TRTi	ai	TRTi	ai	TRTi	ai	TRTi	ai					
0	0	0	0	0	0	0	0	0	4	40.00	0.00	0.00	0.00
1	4	56	70	0	80	0	90	0	4	20.00	20.00	28.00	2.00
2	100	0	44	56	100	0	100	0	4	13.33	26.67	37.33	34.67
3	100	0	100	0	44	56	100	0	4	10.00	30.00	42.00	51.00
4	100	0	100	0	100	0	44	56	4	8.00	32.00	44.80	60.80
5	100	0	100	0	100	0	100	0	4	0.00	40.00	44.80	80.80
6	44	56	100	0	100	0	100	0	4	0.00	40.00	44.80	88.80
7	100	0	44	56	100	0	100	0	4	0.00	40.00	44.80	88.80
8	100	0	100	0	44	56	100	0	4	0.00	40.00	44.80	88.80
9	100	0	100	0	100	0	44	56	4	0.00	40.00	44.80	88.80

Table 2: The Result of the Analytical Computations for the Timely-Token Protocol Q for $n = 0, 1, 2, 3, 4$

MEAN (e_i)	MEAN (h_i)	N	MEAN (a_i)	Percentage increase in A_v	$\hat{C}$	Percentage increase in $\hat{c}$
0.0	40	0.0	0.0	0.0%	0.0	0.0%
0.0	40	1.0	28.0	1.0%	72.0	1.0%
0.0	40	2.0	37.3	33.3%	81.3	13.0%
0.0	40	3.0	42.0	50.0%	86.0	19.4%
0.0	40	4.0	44.8	60.0%	88.8	23.3%

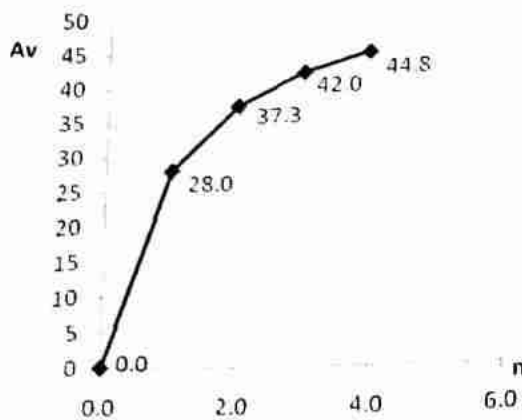


Fig 2a Graph of Average Asynchronous Traffic Time Units Per Cycle, (A_v) Versus Number of nodes with heavy load of asynchronous traffic (n)

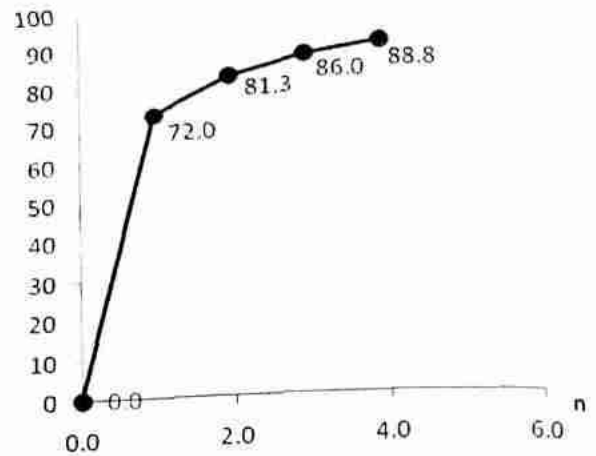


Fig 2b Graph of Average Cycle Length($\hat{C}$) Versus Number of nodes with heavy load of asynchronous traffic (n)

The items in Table 1a to Table 1d are :

TRT_i is the token rotation time of node 0.

$MEAN(TRT_i)$ is the mean of $\sum TRT_i$.

$\sum h_i$ is the total time units used by the synchronous traffic per cycle. $MEAN(\sum h_i)$ is the mean of $\sum h_i$.

$\sum \epsilon_i$ is the total of the time units reserved for the synchronous traffic per cycle but are not used by the synchronous traffic.

$MEAN(\sum \epsilon_i)$ is the mean of $\sum \epsilon_i$.

$\sum a_i$ is the total time units used by the asynchronous traffic per cycle.

$MEAN(\sum a_i)$ is the mean of $\sum a_i$.

Note that all the MEANs are taken over $n + 1$ cycle, except for row 1 to row n in Table 1a to Table 1d.

4.3 Discussion of Results
4.3.1 Validating the analytical results with the simulation results

At this point, it is worthy to note that the simulation results corresponds with the results obtained from the analytical computations based on the analytical expressions derived in this paper. Notably:

- For $n = 1$, in Table 1a row 6 to row 11; $MEAN(a_i) = A_v = 28.0$ and $MEAN(TRT_i) = \hat{C} = 72.0$. This corresponds with the analytical result for $n = 1$ in Table 2 row 3.
- For $n = 2$, in Table 1b row 6 to row 11; $MEAN(a_i) = A_v = 37.33$ and $MEAN(TRT_i) = \hat{C} = 81.33$. This

corresponds with the analytical result for $n = 2$ in Table 2 row 4.

- For $n = 3$, in Table 1c row 6 to row 11; $MEAN(a_i) = A_v = 42.0$ and $MEAN(TRT_i) = \hat{C} = 86.0$. This corresponds with the analytical result for $n = 3$ in Table 2 row 5.
- For $n = 4$, in Table 1d row 6 to row 11; $MEAN(a_i) = A_v = 44.8$ and $MEAN(TRT_i) = \hat{C} = 88.80$. This corresponds with the analytical result for $n = 4$ in Table 2 row 6.

4.3.2 Comparison Of The Performance Of The Timely-Token Protocol For Various Values Of n

From Table 2, it can be seen that, as n increases from 1 to N (where $N = 4$), the A_v and $\hat{C}$ increases. Specifically, from row 3 and row 4 of Table 2, A_v increased by 33.3% as n increase from 1 to 2. Similarly, from row 3 and row 5 of Table 2, A_v increased by 50% as n increase from 1 to 3 and from row 3 and row 6 of Table 2, A_v increased by 60% as n increase from 1 to 4.

The same results are captured in Fig 2a.

In the same way, from row 3 and row 4 of Table 2, $\hat{C}$ increased by 13.0% as n increase from 1 to 2. Similarly, from row 3 and row 5 of Table 2, $\hat{C}$ increased by 19.4% as n increase from 1 to 3 and from

row 3 and row 6 of Table 2, $\hat{e}$ increased by 23.3% as n increase from 1 to 4. The same results are captured in Fig 2b.

5.0 Conclusion and Recommendations

5.1 Conclusion

In this paper, the effect of load distribution of asynchronous traffic on the performance of the Timely-Token protocol was presented. It was discovered that even when the network is heavily loaded with asynchronous traffic, the distribution of the asynchronous frames within the network can significantly affect the performance of the timely-token protocol. Furthermore, it was discovered that a network can still be considered to be heavily loaded, even when some nodes in the network do not have asynchronous frames to transmit. In such situation, the average throughput for the synchronous traffic decreases as the number of nodes without asynchronous frames increases.

5.1 Recommendations

Given the shortcomings of the timely-token protocol, further studies are required to improve the best-effort mechanism in the timely-token protocol. Such protocol should strive to attain an average of A time units of asynchronous traffic in every cycle.

Reference

- [1] Ricardo Moraes (2010), "Survey of Real-Time Communication in CSMA-Based Networks", *Network Protocols and Algorithms*, ISSN 1943-3581 2010, Vol. 2, No. 1
- [2] White P. P., (1997), "RSVP and Integrated Services in the Internet: A Tutorial", *IEEE Communications Magazine* • May 1997
- [3] Indumathi G. and Murugesan K. (2010), "A bandwidth efficient scheduling framework for non real time applications in wireless networks", *International Journal of Distributed and Parallel systems* (IJDPS) Vol.1, No.1, September 2010
- [4] Zhang S. and Burns A., (1994 a), "Timing Properties of the Timed Token Protocol", Tech. Rept. (YCS 243), Dept of Computer Science, Univ. of York (May 1994).
- [5] Zhang S. and Burns A., (1994 b), "A Study of Timing Properties with the Timed Token Protocol", Technical Report (YCS 226), Dept. of Computer Sci., Univ. of York (March 1994).
- [6] Regnier, P. and G. Lima, (2006), "Deterministic integration of hard and soft real-time communication over shared-ethernet". In Proc. of Workshop of Tempo Real (2006), Curitiba, Brazil
- [7] Nicholas M. and Wei Z., (1994), "The timed-token protocol for real-time communications," *Computer*, vol. 27, no. 1, pp. 35-41, January, 1994.
- [8] IEEE (1995), "The Institute of Electrical and Electronic Engineers, Token-passing bus access method and physical layer specifications", *American national Standard ANSI/IEEE std. 802.4 - 1985*.
- [9] Biao C. and Wei Z., (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.
- [10] 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.
- [11] Kenneth C. S. and Marjory J. J., (1987) "Cycle time properties of the FDDI token ring protocol," *IEEE Trans. Software. Eng.*, vol. SE-13, pp. 376-385, Mar. 1987.
- [12] Grow R. (1982), "A timed token protocol for local area networks",

- Proc. Electro'82, Token Access Protocols, Paper 17/3, May 1982.
- [13] Chan E., Chen D., Cao J. and Lee C. H. (1992), "The time Properties of the FDDI-M Medium Access Control Protocol" *The Computer Journal*, Vol. 82 No. 1, pp. 96-102, Jan. 1999.
- [14] Dept. of Defense US (1992), "Survivable Adaptable Fibre-Optic Embedded Networks", MIL-STD-2004, US Dept. of Defense, Washington D.C., Sept, 1992.
- [15] McGuffin L.J., Reid L.O. and Sparks S.R. (1998), "MAP/TOP in CIM Distributed Computing," *IEEE Network*, vol.2, no. 3, May 1988, pp. 23 - 31.
- [16] 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.
- [17] Tovar E., and Vasques F. (1991), "Setting Target Rotation Time in Profibus Based Real-Time Distributed Applications", *Proc. of the 15th IFAC Workshop on Distributed Computer Control Systems*, 1998.
- [18] Lee, D, Attias, R, Puri, A, Sengupta, R, Tripakis, S and Varaiya P (2001). "A wireless token ring protocol for intelligent transportation systems," *IEEE Intelligent Transportation Systems Conference Proceedings*, Aug. 2001, pp. 1152-1157
- [19] Malpani N., Vaidya N., and Welch J (2001) "Distributed Token circulation on mobile Ad Hoc Networks," 21th International Conference on distributed computing systems (ICDCS 2001) PHOENIX, Arizona, USA, April 2001, pp.691-701.
- [20] Willig A. (2002) "Analysis of the PROFIBUS Token passing protocol over wireless links," *Proc. of IEEE International Symposium on Industrial Electronics (IEEE-ISIE 2002)*, L'Aquila, Italy, July 2002, pp. 56-61.
- [21] Cobb J. A. and Lin M., (2004), "The timely token protocol," *Computer Communications Vol .27 no.7*, pp. 569-580 May, 2004
- [22] Ozuomba S. and Chukwudebe G.A, (2003), "An 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 .
- [23] Agrawal G., Chen B., Zhao W., and Davari S. (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
- [24] 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
- [25] Ozuomba S., Chukwudebe G. A. and Obot A.B. (2011) . Statics-Threshold-Limited On-Demand Guaranteed Service For Asynchronous Traffic in Timely-token Protocol . *Nigerian Journal of Technology*, Vol 30, No 2 , June 2011, Pp 124 -142
- [26] Joseph G.A, (2003), "An 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

Note: For correspondence, contact:
conskay4@yahoo.com; 07083428736
 Department of Electrical/Electronic and
 Computer Engineering, University of Uyo,
 Akwalbom , Nigeria