PERFORMANCE ANALYSIS OF FIBER DISTRIBUTED DATA INTERFACE NETWORK MEDIA ACCESS CONTROL PROTOCOL UNDER NON-UNIFORM HEAVY LOAD OF ASYNCHRONOUS TRAFFIC

Constance Kalu
Department of
Electrical/Electronic and Computer Engineering
University of Uyo, AkwaIbom
NIGERIA

Simeon Ozuomba
Department of
Electrical/Electronic and Computer Engineering
University of Uyo, AkwaIbom
NIGERIA

Umoren Mfonobong Anthony
Department of
Electrical/Electronic and Computer Engineering
University of Uyo, AkwaIbom
NIGERIA

ABSTRACT

The classical timed token protocol employed in Fiber Distributed Data Interface (FDDI) networks for Media Access Control (MAC) has been well studied under uniform heavy load of asynchronous (non-real-time) traffic. However, in this paper, the 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. The discovery was evident from both simulation and analytical results presented in this paper. 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: Protocol, Asynchronous, Synchronous , Network, Bandwidth, FDDI, FDDI-M.

INTRODUCTION

FDDI networks are among the early multiservice networks that allow the coexistence of both real-time and non-real-time traffic in the same communication domain [1], [2], [3], [4], [5], [6]. For non real-time domain, maximizing the throughput or minimizing the average message delay is the most important performance criteria. However, in the hard real-time domain, concern focuses on satisfying the time constraints of individual messages [7], [8]. Hence, for hard real-time system, predictability is favoured against maximizing throughput. The situation placed a demand for effective and efficient integrated services (multiservices) Local Area Network (LAN). Such networks' **MAC** protocols must deal with different traffic patterns . Hence, the **MAC** protocols must provide bounded message transmission time required by the hard real-time and soft real-time tasks. At the same time, the **MAC** protocols must also provide high throughput, which is always demanded by non-real-time tasks [9]. The timed token protocol is one of such MAC protocols that can effectively meets these conflicting requirements in multiservice networks. Many versions of the timed token protocols have been developed over the years. The FDDI timed token protocol is among the earliest version of such protocols. Later versions of the timed token protocol were developed to address the problems inherent in the earlier version. However, the FDDI MAC protocol remains the basic for examining the fundamental shortcomings of the timed token MAC algorithm, as has been presented in [10] and [11].

Previous Relevant Works

The basic idea of the timed-token protocol was presented by Grow in 1982 [4], specifically; the framework of the timed-token protocol adaptable to either a physical or logical ring was described. Ulm [12] studied the performance characteristics of the protocol presented by Grow [4]. Introductory tutorials on FDDI are given in [12], [14]. The FDDI protocol timer tuning and timing properties are studied in [15], [16] . Further studies on timed-token delay bounds and mean performance figures such as throughput and mean waiting time can be found in [4], [12].

FDDI timed-token is one of the earliest timed-token passing protocol. In FDDI, the token rotation time may reach twice the Target Token Rotation Time (TTRT) , [15], [10], [1]. Due to this token lateness problem, an FDDI network can use at most half of its bandwidth to transmit synchronous traffic [3], [1], [18], [19] . To alleviate this deficiency, Shin et al. proposed the Modified Fiber Distributed Data Interface (FDDI-M) token protocol [3]. 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 [18], [19] and Timely-Token [1].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. However, each of these protocols achieved the improvements by reducing the achievable throughput for the asynchronous traffic. Further, their performance under non-uniform heavy load of asynchronous traffic has not been studied.

Contributions and Motivations

Improvements in the timed token protocols are based on the identified problems in the previous versions, starting with the classical timed token protocol in FDDI [1], [20]. Yet, there is a problem that has not been discovered and as such not yet solved in the classical timed token protocol; namely, the drop in the throughput of the asynchronous traffic in non uniform heavy loaded systems. This is due to inappropriate definition of heavily loaded network with respect to the asynchronous traffic. Consequently, the first contribution of this paper is to proffer a more comprehensive definition of heavily loaded FDDI network with respect to the asynchronous traffic, as presented in Section 2.2.1.

The second contribution of this paper is performance analysis of the FDDI timed token MAC algorithm under non-uniform heavy load of asynchronous traffic. Specifically, analytical expression for some key protocol timing and performance parameters, namely; *Maximum Cycle Length* , *Average Cycle Length (C),* and *Average Asynchronous Traffic (capacity) Time Units Per Cycle (Av)* were derived. Besides, for the various protocol parameters, the results of the analytical computations were validated with results obtained from the simulation of the FDDI timed token MAC algorithm.

NETWORK CHARACTERISTICS

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…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) mod N.

The Walk-Time and Ring Latency
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, … 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.

- **Definition:-** Let w_i denote the latency or walk-time between a node i and its upstream neighbor node (i + 1). w_i can also be defined as the time needed to transmit the token between nodes, including the overhead introduced by the protocol [2]. Then , $W = \sum_{i}^{N-1} w_i$ (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 [2], [21]. 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; $s_0, s_1, s_2 \ldots s_{N-1}$
where, $M = \{s_0, s_1, s_2 \ldots s_{N-1}\}$ (2)
The synchronous message stream, s_i at node i is given as $s_i = \{P_i , C_i , D_i\}$ (3)
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

The Timed-Token MAC Protocol Parameters

The timed-token protocol uses the following parameters for its operation [10], [17].

1. **Target Token Rotation Time, (TTRT).** Let τ denote the value of TTRT [10], [11], [17].
2. **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 [10, [11], [17]. Then, other related parameters are [10], [11], [17].

$$\mathbf{H} = \mathbf{H}_0 + \mathbf{H}_1 + \ldots \mathbf{H}_{N-1} = \sum_{i}^{N-1} H_i \quad (4)$$

$$H_i + w_i = \alpha_i \quad (5)$$

$$\alpha_0 + \alpha_1 + \ldots\alpha_{(N-1)} = T = W + H = \sum_{i}^{N-1} \alpha_i \quad (6)$$

Constraints: The *Protocol Constraint* requires that [10], [11], [17]: $T = H + W \le \tau$ (7)
The *Deadline Constraint* , in FDDI, since the time elapsed between two consecutive visits of the token at a node can be as much as 2TTRT therefore, in order for the deadline constraint to be satisfied, it is required that for i = 0,1,…N-1 [17] $\min_{i=0,1,\ldots N-1} (D_i) \le 2\tau$ (8)

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

$$\mathbf{T} \leq \tau \leq \min_{i=0,1,\dots N-1} \left\{ \frac{D_i}{2} \right\} \quad (9)$$

Token Rotation Timer of Node i (TRT$_i$) [11], [17]

Let t_0 , t_1, ...$t_{(N-1)}$ be the time at which the token reaches station 0,1,…N-1 for some given cycle. Also, let t_N , $t_{(N+1)}$, ...$t_{(2N-1)}$ be the time at which the token reaches station 0,1,…N-1 in the next cycle and so forth. Let t_{-N} , $t_{(1-N)}$, ...$t_{(N-1)-N}$ be the time at which the token reaches station 0,1,…N-1 in the previous cycle to the given cycle. 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;

$$\text{Cycle length for node i} = t_i - t_{i-N} \quad (10)$$

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

$$j = (i \bmod N) \quad (11)$$

$$k = \left\lfloor \frac{i}{\mathbf{N}} \right\rfloor \quad (12)$$

3. **Token Holding Timer of Node i (THT$_i$)** [11].
4. **Late Counter of Node i (LC$_i$).** This counter is used to record the number of times that **TRT$_i$** has expired since the last token arrival at node i.

The Timed-Token MAC Algorithm Considered Under Non-Uniform Heavy Load Of Asynchronous Traffic

The original FDDI timed token algorithm is presented in [11], [17]. In this section the original algorithm is modified to accommodate analysis of the algorithm in situations of non uniform heavy load; where only few nodes are heavily loaded at a given time. The algorithm is presented as Algorithm P in the appendix.

From Algorithm P in the appendix it will be noted that LCi is incremented by one at every expiration of TRT$_i$. The token can arrive early or late at a node. If LC$_i$ = 0 at the time the token arrives at node i then the token is considered to arrive early at node i. The token is late if LC$_i$ > 0. If LC$_i$ exceeds one (i.e. if LC$_i$ > 1), it is considered an abnormal situation, for instance , the token is considered lost ; then , the ring recovery process is initiated [2]. In essence, under normal operating conditions, the following situations can occur;

$$0 \leq LC_i \leq 1 \quad (13)$$

Meaning of Heavily Loaded Nodes and Heavily Loaded 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 : a network is said to be uniform heavy loaded when $n = N$; that means , if all the nodes are heavily loaded with asynchronous traffic.

iv. ***Non-Uniform Heavy Loaded Network :*** a network is said to be non-uniform heavy loaded when $1 \leq n \leq N$; that means, if one or more of its nodes is not heavily loaded and at least one of its nodes is heavily loaded.

Performance Analysis of the FDDI MAC Algorithm with Non-Uniform Heavy Load Of Asynchronous Traffic

According to the protocol operations in **P1, P2** and **P3**, $\mathbf{TRT_i}$ is initialized to **TTRT,** it counts down and is reset to **TTRT** every time it counts down to zero . When protocol **P** with **N** active nodes in the network is considered, the cycle length , $\mathbf{TRT_i}$ is given as ;

Thus, $$t_i - t_{i-N} \leq \mathbf{LC_i} * \tau + (\tau - \mathbf{TRT_i}) \quad (14)$$

In Protocol **P2.6** and **P3,** when the token arrives early at node i, $LC_i = 0$, then

$$t_i - t_{i-N} \leq (\tau - \mathbf{TRT_i}) \text{ where } \mathbf{LC_i} = 0 \quad (15)$$

$$\mathbf{TRT_i} \leq \tau - (t_i - t_{i-N}) \text{ where } t_i - t_{i-N} \leq \tau \quad (16)$$

$$\mathbf{THT_i} = \mathbf{TRT_i} \text{ ; thus } \mathbf{THT_i} \leq \tau - (t_i - t_{i-N}) \text{ for } t_i - t_{i-N} \leq \tau \quad (17)$$

By the protocol operations in **P3.4**, and Eq17 $a_i \leq \mathbf{THT_i}$ thus;

$$a_i \leq \tau - (t_i - t_{i-N}) \text{ for } t_i - t_{i-N} \leq \tau \text{ (i.e for } \mathbf{LC_i} = 0) \quad (18)$$

On the other hand, if $\mathbf{LC_i} \geq 1$, by protocol **P2.6** and **P3,** when the token arrives late at node i, and by Eq13, $\mathbf{LC_i} = 1$, then Eq14 gives $$t_i - t_{i-N} \leq 2\tau - \mathbf{TRT_i} \quad (19)$$

$$\mathbf{TRT_i} = 2\tau - (t_i - t_{i-N}) \quad (20)$$

When $\mathbf{LC_i} = 1$, the token is considered to arrive late. In this case, by the protocol operations in **P2.6**, no asynchronous traffic is transmitted. This means $$a_i = \mathbf{0} \text{ for } t_i - t_{i-N} \geq \tau \quad (21)$$

In any case, at node i, the amount of time units used for the transmission of asynchronous traffic, a_i is given by Eq17 and Eq21 as $$a_i \leq \max\{0, \mathbf{THT_i}\} \text{ for all } i \geq 0 \quad (22)$$

Also, a_i is given by Eq18 and Eq21 as $$a_i \leq \max\{0, \tau - (t_i - t_{i-N})\} \text{ for all } i \geq 0 \quad (23)$$

Under light load of synchronous traffic, it is not all the time units reserved for (or allocated to) the synchronous traffic that is used in every token receipt at node i. Let, $\boldsymbol{h_i}$ be the used portion and ε_i be the unused portion of the time reserved for the synchronous traffic in node i. Then, for a network that is lightly loaded with synchronous traffic, out of the $\boldsymbol{H_i}$ time units allocated to the synchronous traffic in node i, only $\boldsymbol{h_i}$ time units are used (where $\boldsymbol{h_i} \leq \boldsymbol{H_i}$), leaving ε_i time units unused. Thus

$$\boldsymbol{h_i} = \boldsymbol{H_i} - \varepsilon_i \quad (24)$$

In the analysis, ε_i is assumed to be constant in every cycle , hence $$\sum_{i}^{i-N} \varepsilon_i = \epsilon \quad (25)$$

$$\sum_{i}^{i-N} h_i = h \text{ for all } i \geq 0 \qquad (26)$$

Thus, $$\varepsilon_i = H_i - h_i \qquad (27)$$

Asynchronous traffic

For a system that is heavily loaded with asynchronous traffic, it is assumed that there is always sufficient asynchronous traffic in n nodes to use all the time available for the asynchronous traffic, where $1 \leq n \leq N$. Thus, a_i time units are used for the asynchronous traffic at node i where a_i is given from $a_i \leq \max(0, THT_i)$ hence, $a_i \leq \max\{0, \tau - (t_i - t_{i-N})\}$ for all $i \geq 0$ (28)

When node i release the token, it reaches the next node (i+1) mod N at time , t_{i+1} after extra propagation delay time, w_i, then: $$t_{i+1} \leq t_i + a_i + h_i + w_i \qquad (29)$$

Therefore, $$t_{i+1} \leq \max(t_i, t_{i-N} + \tau) + (\alpha_i - \varepsilon_i) \text{ for all } i \geq 0 \qquad (30)$$

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

In a special case 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 [10], [11], [17]. The initial condition for Eq30and for the special case is assumed to be $t_0 = t'_0 = 0$ [10], [11], [17]. Then, for the special case $\tau'$and t'_i where [10],[11], [17]

$$\tau' = \tau - (T - \epsilon) \qquad (31)$$

$$t'_i = t_i + W - \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \qquad (32)$$

$$t_i = t'_i - W + \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \qquad (33)$$

Eq30 for the special case is

$$t'_{i+1} \leq \max(t'_i, t'_{i-N} + \tau') \text{ where } h_i = 0 \text{ or } \varepsilon_i = H_i \text{ for all } i \geq 0 \qquad (34)$$

The initial condition for Eq30 and Eq34 is assumed to be 0 , that is [17] , [10] ;

$$t_i = t'_i = 0 \quad \text{where } t_i < 0,\ t'_i < 0 \text{ for all } i < 0 \qquad (35)$$

Iterating Eq30 from i=0 to i=N-1 gives

$$0 \leq t_i \leq \tau + \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \quad \text{for } 1 \leq i \leq N \qquad (36)$$

Substituting t_i from Eq36 into32 gives $$0 \leq t'_i \leq \tau + W \text{ for } 1 \leq i \leq N \qquad (37)$$

Using τ as the upper bound on t'_i for $0 \leq i \leq N$, induction over i can be applied to Eq34 to give $$t'_i \leq \tau + \left\lfloor \frac{i}{n+1} \right\rfloor \tau' \qquad (38)$$

The assumption that τ is the upper bound on t'_i in Eq38 is valid if we consider Eq31, $\tau' = \tau - (T - \epsilon) = \tau - T + \epsilon$ and that $T = W + H$ thus $\tau' = \tau - W - H + \epsilon$. Since, $H \geq \epsilon$, then $\tau' \leq \tau$.

Substituting t_i from Eq38 into Eq33 gives $$t_i \leq \tau - W + \left\lfloor \frac{i}{n+1} \right\rfloor \tau' + \sum_{j=0}^{j=i-1} (\alpha_j - \varepsilon_j) \qquad (39)$$

where $\sum_{j=0}^{j=i-1}(\alpha_j - \varepsilon_j) = \sum_{j=0}^{j=(i-1)mod\,N}(\alpha_j - \varepsilon_j) + \left\lfloor \frac{i-1}{n} \right\rfloor (\mathbf{T} - \epsilon)$

(40)

Substituting τ' from Eq31 into Eq39 and also applying Eq40 into Eq39 gives

$$t_i \le \tau - W + \left\lfloor \frac{i}{n+1} \right\rfloor (\tau - (\mathbf{T} - \epsilon)) - \left\lfloor \frac{i-1}{n} \right\rfloor (\mathbf{T} - \epsilon) + \sum_{j=0}^{j=(i-1)mod\,N}(\alpha_j - \varepsilon_j)$$

(41)

$$t_i \le \tau \left(1 + \left\lfloor \frac{i}{n+1} \right\rfloor\right) - W - \left\lfloor \frac{i}{n+1} \right\rfloor (\mathbf{T} - \epsilon) + \left\lfloor \frac{i-1}{n} \right\rfloor (\mathbf{T} - \epsilon) + \sum_{j=0}^{j=(i-1)mod\,N}(\alpha_j - \varepsilon_j) \quad (42)$$

The time, t_{Nk} at which station zero receives its k[th] token (that is i = Nk in Eq42 is given as

$$t_{Nk} \le \tau\left(1 + \left\lfloor \frac{nK}{n+1} \right\rfloor\right) - W - \left\lfloor \frac{nK}{n+1} \right\rfloor (\mathbf{T} - \epsilon) + \mathbf{k}\,(\mathbf{T} - \epsilon)$$

(43)

Note that Eq43 for t_{Nk} is different from the t_{Nk} given in (36,10] , where $t_{Nk} \le \tau\left(1 + \left\lfloor \frac{NK}{N+1} \right\rfloor\right) - W - \left\lfloor \frac{NK}{N+1} \right\rfloor (\mathbf{T}) + \mathbf{k}\,(\mathbf{T})$ and $t_{Nk} \le \tau\left(1 + \left\lfloor \frac{NK}{N+1} \right\rfloor\right) - W - \left\lfloor \frac{NK}{N+1} \right\rfloor (\mathbf{T} - \epsilon) + \mathbf{k}\,(\mathbf{T} - \epsilon)$ respectively. 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, Eq43 gives $t_N \le \tau - W + T - \epsilon$

(44a)

But T = H + W , then Eq44a becomes $t_N \le \tau + H - \epsilon$

(44b)

The cycle length is given as $t_N - t_0$

(45a)

then , substituting t_0 from Eq35 into Eq45a and t_N from Eq44b into Eq45a gives

$$\max(t_i - t_{i-N}) = t_N - t_0 = \tau + H \quad \text{for all } i \ge 0 \quad \text{and } \epsilon \ge 0 \qquad (45b)$$

Average Cycle Length ($\hat{c}$): from Eq43 $\hat{C}$ is given as

$$\hat{C} \le \lim_{k\to\infty}(t_{Nk}/k) \le \left\lfloor \frac{n}{n+1} \right\rfloor \tau - \left\lfloor \frac{n}{n+1} \right\rfloor (\mathbf{T} - \epsilon) + (\mathbf{T} - \epsilon)$$

(46)

$$\hat{C} \le \left\lfloor \frac{n}{n+1} \right\rfloor (\tau - \mathbf{T}) + \left\lfloor \frac{n}{n+1} \right\rfloor \epsilon + (\mathbf{T} - \epsilon)$$

(47)

Average Time Used By The Asynchronous Traffic Per Cycle ($\acute{A}$): from Eq47 we have,

$$\acute{A} = \left\lfloor \frac{n}{n+1} \right\rfloor (\tau - \mathbf{T}) + \left\lfloor \frac{n}{n+1} \right\rfloor \epsilon$$

(48)

Vol. 2 No. 4, 2015
ISSN 2059-3058

Simulation of Protocol Q

The simulation of the Timed-Token MAC algorithm 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, $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$ remain constant for at least M consecutive cycles where M >> N, then MEAN($TRT_{i,}$) approaches $\hat{\boldsymbol{C}}$ obtained from the analytical computations where, MEAN($RT_{j,k}{}^{\#}$) is given as

$$\text{MEAN}\left(TRT_i^{\#}\right) = \left(\frac{1}{(n+1)}\right)\left(\sum_{i=x}^{i=(x+n)} \left(TRT_i^{\#}\right)\right) \quad for\ x > N \quad (49)$$

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

<u>The Average Asynchronous Traffic Time Units Per Cycle</u> $\text{MEAN}\left(a_i^{\#}\right)$

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

$$\text{MEAN}\left(a_i^{\#}\right) = \left(\frac{1}{(n+1)}\right)\left(\sum_{i=x}^{i=(x+n)} \left(a_i^{\#}\right)\right) \quad for\ x > N \quad (50)$$

The Maximum Cycle Length is MAX($TRT_i{}^{\#}$) for all i ≥ 0

Worked Examples and Discussion of Results
Worked Example

Consider a ring network with four stations (N = 4). The ring uses the FDDI Timed Token protocol for its MAC where the timed-token parameters are given as follows: TTRT = $\boldsymbol{\tau}$ = 100, $\mathbf{w}_i$ =1 for all the nodes, $\mathbf{H}_i$ =20 for all the nodes. With these given parameters h = 4(20) = 80. The simulation results for *Protocol P* **(**FDDI Timed Token protocol**)** for various values of n are shown in Table 1a to Table 1d .

Table 1a: The simulation results of FDDI Timed Token MAC protocol (***Protocol P***) for n = 1; **Ɛ = 0, h = 80 (i.e., h = H)**

	0			1			2			3				MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
k	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	n				
0	0	0	0	0	0	0	0	0	0	0	0	0	1	80	0	0	0.00
1	4	96	96	120	0	0	140	0	0	160	0	0	1	40	40	48	2.00
2	180	0	0	84	16	0	84	16	0	84	16	0	1	0	80	48	92.00
3	84	16	16	100	0	0	100	0	0	100	0	0	1	0	80	8	132.00
4	100	0	0	84	16	0	84	16	0	84	16	0	1	0	80	8	92.00
5	84	16	16	100	0	0	100	0	0	100	0	0	1	0	80	8	92.00

6	100	0	0	84	16	0	84	16	0	84	16	0	1	0	80	8	92.00
7	84	16	16	100	0	0	100	0	0	100	0	0	1	0	80	8	92.00
8	100	0	0	84	16	0	84	16	0	84	16	0	1	0	80	8	92.00
9	84	16	16	100	0	0	100	0	0	100	0	0	1	0	80	8	92.00
10	100	0	0	84	16	0	84	16	0	84	16	0	1	0	80	8	92.00

Table 1b: Simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 2; **Ɛ = 0, h = 80**

	0			1			2			3							
k	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	N	MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.	0.0	0.0	0.0
1	4	96	96	120	0	0	140	0	0	160	0	0	2	40.0	40.0	48.0	2.0
2	180	0	0	84	16	16	100	0	0	100	0	0	2	26.7	53.3	37.3	61.3
3	100	0	0	100	0	0	84	16	0	84	16	0	2	0.0	80.0	37.3	94.7
4	84	16	16	100	0	0	100	0	0	100	0	0	2	0.0	80.0	10.7	121.3
5	100	0	0	84	16	16	100	0	0	100	0	0	2	0.0	80.0	10.7	94.7
6	100	0	0	100	0	0	84	16	0	84	16	0	2	0.0	80.0	10.7	94.7
7	84	16	16	100	0	0	100	0	0	100	0	0	2	0.0	80.0	10.7	94.7
8	100	0	0	84	16	16	100	0	0	100	0	0	2	0.0	80.0	10.7	94.7
9	100	0	0	100	0	0	84	16	0	84	16	0	2	0.0	80.0	10.7	94.7
10	84	16	16	100	0	0	100	0	0	100	0	0	2	0.0	80.0	10.7	94.7

Table 1c: The simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 3; Ɛ = 0 and h = 80

	0			**1**			**2**			**3**							
k	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	n	MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.00	0.00	0.00	0.00
1	4	96	96	120	0	0	140	0	0	160	0	0	3	40.00	40.00	48.00	2.00
2	180	0	0	84	16	16	100	0	0	100	0	0	3	26.67	53.33	37.33	61.33
3	100	0	0	100	0	0	84	16	16	100	0	0	3	20.00	60.00	32.00	71.00
4	100	0	0	100	0	0	100	0	0	84	16	0	3	0.00	80.00	32.00	96.00
5	84	16	16	100	0	0	100	0	0	100	0	0	3	0.00	80.00	12.00	116.00
6	100	0	0	84	16	16	100	0	0	100	0	0	3	0.00	80.00	12.00	96.00
7	100	0	0	100	0	0	84	16	16	100	0	0	3	0.00	80.00	12.00	96.00
8	100	0	0	100	0	0	100	0	0	84	16	0	3	0.00	80.00	12.00	96.00
9	84	16	16	100	0	0	100	0	0	100	0	0	3	0.00	80.00	12.00	96.00
10	100	0	0	84	16	16	100	0	0	100	0	0	3	0.00	80.00	12.00	96.00

Table 1d: The simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 4; **Ɛ = 0** and h = 80

	0			1			2			3							
k	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	Ai	n	MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.0	0.00	0.00	0.00
1	4	96	96	120	0	0	140	0	0	160	0	0	4	40.00	40.00	48.00	2.00
2	180	0	0	84	16	16	100	0	0	100	0	0	4	26.67	53.33	37.33	61.33
3	100	0	0	100	0	0	84	16	16	100	0	0	4	20.00	60.00	32.00	71.00
4	100	0	0	100	0	0	100	0	0	84	16	16	4	16.00	64.00	28.80	76.80
5	100	0	0	100	0	0	100	0	0	100	0	0	4	0.00	80.00	28.80	96.80
6	84	16	16	100	0	0	100	0	0	100	0	0	4	0.00	80.00	12.80	112.80
7	100	0	0	84	16	16	100	0	0	100	0	0	4	0.00	80.00	12.80	96.80
8	100	0	0	100	0	0	84	16	16	100	0	0	4	0.00	80.00	12.80	96.80
9	100	0	0	100	0	0	100	0	0	84	16	16	4	0.00	80.00	12.80	96.80
10	100	0	0	100	0	0	100	0	0	100	0	0	4	0.00	80.00	12.80	96.80

Table 2a: The simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 1; Ɛ = 40 and h = 40

	0			1			2			3							
K	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	n	MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.00	0.00	0.00	0.00
1	4	96	96	110	0	0	120	0	0	130	0	0	1	60.00	20.00	48.00	2.00
2	140	0	0	44	56	0	44	56	0	44	56	0	1	40.00	40.00	48.00	72.00
3	44	56	56	100	0	0	100	0	0	100	0	0	1	40.00	40.00	28.00	92.00
4	100	0	0	44	56	0	44	56	0	44	56	0	1	40.00	40.00	28.00	72.00
5	44	56	56	100	0	0	100	0	0	100	0	0	1	40.00	40.00	28.00	72.00
6	100	0	0	44	56	0	44	56	0	44	56	0	1	40.00	40.00	28.00	72.00
7	44	56	56	100	0	0	100	0	0	100	0	0	1	40.00	40.00	28.00	72.00
8	100	0	0	44	56	0	44	56	0	44	56	0	1	40.00	40.00	28.00	72.00
9	44	56	5	10	0	0	10	0	0	10	0	0	1	40.0	40.0	28.0	72.0

			6	0			0			0				0	0	0	0
10	100	0	0	44	56	0	44	56	0	44	56	0	1	40.00	40.00	28.00	72.00

Table 2b: The simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 2; Ɛ = 40 and h = 40

	0			1			2			3							
k	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	TRTi	THTi	ai	n	MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.00	0.00	0.00	0.00
1	4	96	96	110	0	0	120	0	0	130	0	0	2	60.00	20.00	48.00	2.00
2	140	0	0	44	56	56	100	0	0	100	0	0	2	53.33	26.67	50.67	48.00
3	100	0	0	100	0	0	44	56	0	44	56	0	2	40.00	40.00	50.67	81.33
4	44	56	56	100	0	0	100	0	0	100	0	0	2	40.00	40.00	37.33	94.67
5	100	0	0	44	56	56	100	0	0	100	0	0	2	40.00	40.00	37.33	81.33
6	100	0	0	100	0	0	44	56	0	44	56	0	2	40.00	40.00	37.33	81.33
7	44	56	56	100	0	0	100	0	0	100	0	0	2	40.00	40.00	37.33	81.33
8	100	0	0	44	56	56	100	0	0	100	0	0	2	40.00	40.00	37.33	81.33
9	100	0	0	100	0	0	44	56	0	44	56	0	2	40.00	40.00	37.33	81.33
10	44	56	56	100	0	0	100	0	0	100	0	0	2	40.00	40.00	37.33	81.33

Table 2c: The simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 3; Ɛ = 40 and h = 40

			0	1			2					3					
k	TRTi	THTi	ai	TRTi	THTi	ai					THTi	ai	n	MEAN (εi)	MEAN (hi)	MEAN (ai)	EAN (TRTi)
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.00	0.00	0.00	0.00
1	4	96	96	110	0	0	120	0	0	130	0	0	3	60.00	20.00	48.00	2.00
2	140	0	0	44	56	56	100	0	0	100	0	0	3	53.33	26.67	50.67	48.00
3	100	0	0	100	0	0	44	56	56	100	0	0	3	50.00	30.00	52.00	61.00
4	100	0	0	100	0	0	100	0	0	44	56	0	3	40.00	40.00	52.00	86.00
5	44	56	56	100	0	0	100	0	0	100	0	0	3	40.00	40.00	42.00	96.00
6	100	0	0	44	56	56	100	0	0	100	0	0	3	40.00	40.00	42.00	86.00

7	100	0	0	100	0	0	44	56	56	100	0	0	3	40.00	40.00	42.00	86.00
8	100	0	0	100	0	0	100	0	0	44	56	0	3	40.00	40.00	42.00	86.00
9	44	56	56	100	0	0	100	0	0	100	0	0	3	40.00	40.00	42.00	86.00
10	100	0	0	44	56	56	100	0	0	100	0	0	3	40.00	40.00	42.00	86.00

Table 2d: The simulation results of the FDDI Timed Token MAC protocol (***Protocol P***) for n = 4; Ɛ = 40 and h = 40

	0			1			2			3				MEAN (εi)	MEAN (hi)	MEAN (ai)	MEAN (TRTi)
k	TRTi	THTi	ai	TRTi	THTi	ai					THTi	ai	n				
0	0	0	0	0	0	0	0	0	0	0	0	0	4	80.00	0.00	0.00	0.00
1	4	96	96	110	0	0	120	0	0	130	0	0	4	60.00	20.00	48.00	2.00
2	140	0	0	44	56	56	100	0	0	100	0	0	4	53.33	26.67	50.67	48.00
3	100	0	0	100	0	0	44	56	56	100	0	0	4	50.00	30.00	52.00	61.00
4	100	0	0	100	0	0	100	0	0	44	56	56	4	48.00	32.00	52.80	68.80
5	100	0	0	100	0	0	100	0	0	100	0	0	4	40.00	40.00	52.80	88.80
6	44	56	56	100	0	0	100	0	0	100	0	0	4	40.00	40.00	44.80	96.80
7	100	0	0	44	56	56	100	0	0	100	0	0	4	40.00	40.00	44.80	88.80
8	100	0	0	100	0	0	44	56	56	100	0	0	4	40.00	40.00	44.80	88.80
9	100	0	0	100	0	0	100	0	0	44	56	56	4	40.00	40.00	44.80	88.80
10	100	0	0	100	0	0	100	0	0	100	0	0	4	40.00	40.00	44.80	88.80

The performance parameters and the average values obtained from the simulation result for *Protocol P* is shown in Table 1a to Table 1d for ε = 0 and h = 80. Similarly, the performance parameters and the average values obtained from the simulation result for *Protocol P* is shown in Table 2a to Table 2d for
ε = 40 and h = 40. The items in the tables 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 $\sum h_i$.

$\sum \varepsilon_i$ 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 $\sum \varepsilon_i$.

$\sum a_i$ 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 cycles, except for row 1 to row n in Table 2a to Table 2d.

Table 3. Graph of Average Asynchronous Traffic Time Units Per Cycle Average Cycle Length, (ĉ) Versus Number of nodes with heavy load of asynchronous traffic (n) for Ɛ = 0 and h = 80

n	A_V	% Increase in A_V	ĉ	% Increase in ĉ
0	0.0	0	0.0	0
1	8.0	0	92.0	0
2	10.7	33	94.7	3
3	12.0	50	96.0	4
4	12.8	60	96.8	5

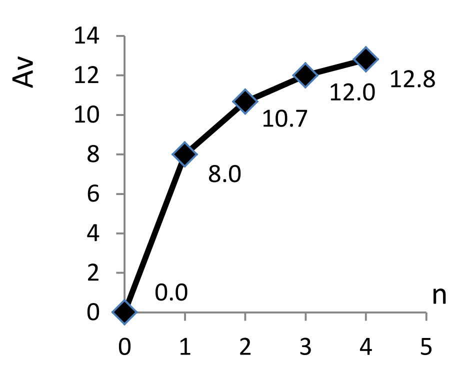

Fig 1a Graph of Average Asynchronous Traffic Time Units Per Cycle, (Av) Versus Number of nodes with heavy load of asynchronous traffic (n) for Ɛ = 0

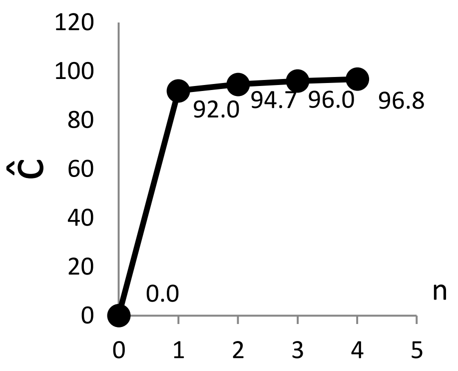

Fig 1b Graph of Average Cycle Length, (ĉ) Versus Number of nodes with heavy load of asynchronous traffic (n) for Ɛ = 0

Table 4. Graph of Average Asynchronous Traffic Time Units Per Cycle Average Cycle Length, (ĉ) Versus Number of nodes with heavy load of asynchronous traffic (n) for Ɛ = 40 and h = 40

n	A_V	% Increase in A_V	ĉ	% Increase in ĉ
0	0.0	0	0.0	0
1	28.0	0	72.0	0
2	37.3	33	81.3	13
3	42.0	50	86.0	19
4	44.8	60	88.8	23

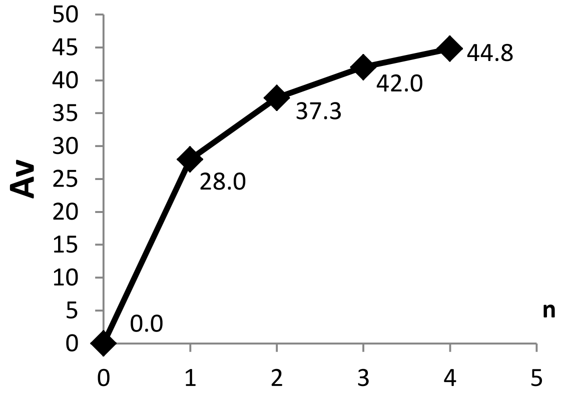

Fig 2a Graph of Average Asynchronous Traffic Time Units Per Cycle, (Av) Versus Number of nodes with heavy load of asynchronous traffic (n) for Ɛ = 40

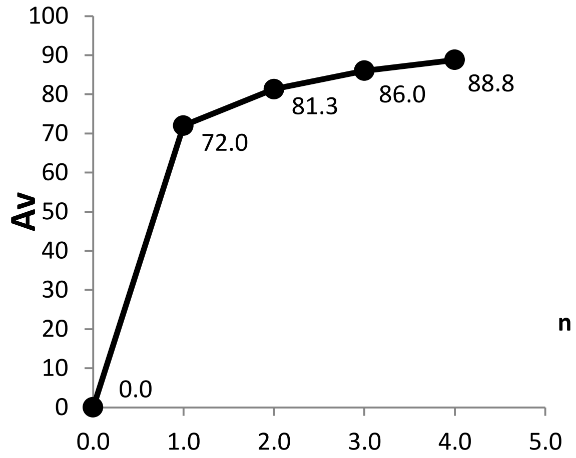

Fig 2b Graph of Average Cycle Length, (ĉ) Versus Number of nodes with heavy load of asynchronous traffic (n) for Ɛ = 40

DISCUSSION OF RESULTS

Validating the analytical results with the simulation results

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. For the simulation results , the mean values are taken over n +1 cycles ; after at least , the first n +1 cycles, the mean values from the simulation results tends to the expected mean values obtain from the analytical computations. Notably:

- For n = 1, in Table 1a row 6 to row 12; when **ε = 0**; MEAN (a_i) = Av = 8.0 and MEAN (TRT_i) = Ĉ = 92.0. This corresponds with the analytical result for n = 1in Table 3 row 3.
- For n = 1, in Table 2a row 6 to row 12; when **ε = 40**; MEAN (a_i) = Av = 28.0 and MEAN (TRT_i) = Ĉ = 72.0. This corresponds with the analytical result for n = 1in Table 4 row 3.
- For n = 2, in Table 1b row 6 to row 12; when **ε = 0;** MEAN (a_i) = Av = 10.7 and MEAN (TRT_i) = Ĉ = 94.7. This corresponds with the analytical result for n = 2 in Table 3 row 4.
- For n = 2, in Table 2b row 6 to row 12; when **ε = 40**; MEAN (a_i) = Av = 37.3 and MEAN (TRT_i) = Ĉ = 81.3. This corresponds with the analytical result for n = 2 in Table 4 row 4.
- For n = 3, in Table 1c row 7 to row 12; when **ε = 0;** MEAN (a_i) = Av = 12.0 and MEAN (TRT_i) = Ĉ = 96.0. This corresponds with the analytical result for n = 3 in Table 3 row 5.
- For n = 3, in Table 2c row 7 to row 10; when **ε = 40**; MEAN (a_i) = Av = 42.0 and MEAN (TRT_i) = Ĉ = 86.0. This corresponds with the analytical result for n = 3 in Table 4 row 5.
- For n = 4, in Table 1d row 8 to row 12; when **ε = 0;** MEAN (a_i) = Av = 12.80 and MEAN (TRT_i) = Ĉ = 96.80. This corresponds with the analytical result for n = 4 in Table 3 row 6.

- For n = 4, in Table 2d row 8 to row 12; when **ε = 40**; MEAN (a_i) = Av = 44.8 and MEAN (TRT_i) = Ĉ = 88.8. This corresponds with the analytical result for n = 4 in Table 4 row 6.

Comparison Of The Performance Of FDDI Timed Token Protocol For Various Values Of n

First, from Table 3 and Table 4, it can be seen that as **n** increases from 1 to N (where N = 4) , the A_V and ĉ increase for any given value of ε. Specifically, for **ε = 0** (in which case , the network is heavily loaded with synchronous traffic), from row 3 and row 4 of Table 3, $\mathbf{A_V}$ increased by 33% as n increased from 1 to 2. Similarly, from row 3 and row 5 of Table 3**,** $\mathbf{A_V}$ increased by 50% as n increase from 1 to 3 and from row 3 and row 6 of Table 3, $\mathbf{A_V}$ increased by 60% as n increase from 1 to 4.

In the same way, from row 3 and row 4 of Table 3, ĉ increased by 13.0% as n increase from 1 to 2. Similarly, from row 3 and row 5 of Table 3, ĉ increased by 19.4% as n increase from 1 to 3 and from row 3 and row 6 of Table 3, ĉ increased by 23.3% as n increase from 1 to 4. The same results are captured in Fig 3a and Fig 3b.

Equally, for **ε = 40,** from row 3 and row 4 of Table 4, $\mathbf{A_V}$ increased by 33.0% as n increase from 1 to 2. Similarly, from row 3 and row 5 of Table 4, $\mathbf{A_V}$ increased by 50.0% as n increase from 1 to 3 and from row 3 and row 6 of Table 4, $\mathbf{A_V}$ increased by 60.0% as n increase from 1 to 4.

Also for the Average Cycle Length **ĉ** when **ε = 40**, we obtained the following, from row 3 and row 4 of Table 3, ĉ increased by 13.0% as n increase from 1 to 2. Similarly, from row 3 and row 5 of Table 3, ĉ increased by 19% as n increase from 1 to 3 and from row 3 and row 6 of Table 3, ĉ increased by 23% as n increase from 1 to 4. The same results are captured in Fig 4a and Fig 4b.

Secondly, Table 3 shows that when ε_i = 0, **€**= 0, this is the heavy load condition for synchronous traffic. In essence, the analytical approach presented in this paper captures the general performance of the FDDI timed token protocol under varying load conditions; from no (zero) load of synchronous traffic to heavy or full load of synchronous traffic.

Thirdly, at any given instance, the analytical results showed that a total of $(\boldsymbol{\tau} - \mathbf{T}) + \boldsymbol{\epsilon}$ time units are available for the asynchronous traffic. However, not all of the available time units are used even when there is heavy load of asynchronous traffic. Rather, only $(\frac{n}{n+1})(\boldsymbol{\tau} - \mathbf{T}) + (\frac{n}{n+1})\boldsymbol{\epsilon}$ time units are used leaving $(\frac{1}{n+1})(\boldsymbol{\tau} - \mathbf{T}) + (\frac{1}{n+1})\boldsymbol{\epsilon}$ time units unused. This is captured in the graph of Fig 3a and Table 3 and Fig 4a and Table 4. When $\varepsilon_i = h_i$, € = H, thus, there is no synchronous traffic in the network, the average cycle length is $(\frac{n}{n+1})(\boldsymbol{\tau} - \mathbf{T}) + (\frac{n}{n+1})\boldsymbol{\epsilon}$ = 88.8 which is 11.2 (i.e. $(\frac{1}{n+1})(\boldsymbol{\tau} - \mathbf{T}) + (\frac{1}{n+1})\boldsymbol{\epsilon}$**)** less that the maximum value of 100 (i.e. $\boldsymbol{\tau}$).

Finally, when **τ** = 100, H= 80 and €= 0,from row 5 in Table 1a to Table 1d, (that is, for k = 2) , the Maximum cycle length is 180 which corresponds to τ + H- $\boldsymbol{\epsilon}$ specified by the analytical expression in Eq70a and Eq71a. Similarly, when **τ** = 100, H= 80,€= 40, from row 5 in Table 2a to Table 2d, (that is, for k = 2) , the Maximum cycle length is 140 which corresponds to τ + H- $\boldsymbol{\epsilon}$ specified by the analytical expression in Eq70a.

ISSN 2059-3058

CONCLUSION AND RECOMMENDATIONS

CONCLUSION

In this paper, the effect of load distribution of the asynchronous traffic on the performance of the classical timed token **MAC** protocol in FDDI network 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 FDDI timed token MAC protocol. Furthermore, it was discovered that a network can still be considered to be heavily loaded with asynchronous traffic, 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. Hence, the effect of nonuniform distribution of asynchronous traffic in the timed token protocol is that it decreases the throughput of the protocol.

Recommendations For Further Work

This paper presented a unique model and analytical approach to examine the performance of the classical timed-token MAC protocol, specifically; the FDDI timed-token MAC protocol, under non-uniform distribution of asynchronous traffic. There are later versions of the timed-token protocol with claims of improved performance. Among such protocols are, the FDDI-M , the On-time timed-token protocol , the timely-token protocol, and the recent BuST: Budget Sharing Token protocol. Further studies are required to apply the unique model and analytical approach presented in this paper to examine the effect of non-uniform distribution of asynchronous traffic in the performance of each of those improved timed-token MAC protocols. Furthermore, further studies are required to address the identified problem after examining its effect on the various versions of the timed-token MAC protocol. Such studies will lead to the development of a more robust timed-token MAC protocol that can maintain high throughput in the face of variations in the load distribution of the asynchronous traffic.

REFERENCE

1. Cobb J. A., Lin M. (2004). The Timely-Token protocol, Computer Communications Vol .27 No.7, pp. 569-580 May, 2004
2. Biao C. and Wei Zhao, (1992). Properties of the Timed Token Protocol", Department of Computer Science Texas A&M University College Station, TX 77843-3112 October, 1992 Technical Report 92-038
3. 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.
4. Grow R., (1982) . A timed token protocol for local area networks, Proc. Electro'82, Token Access Protocols, Paper 17/3, May 1982.
5. 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.
6. Ricardo M. (2010) .Survey of Real-Time Communication in CSMA-Based Networks. Network Protocols and Algorithms ISSN 1943-3581 2010, Vol. 2, No. 1
7. 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
8. 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).

9. 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), Curitíba, Brazil
10. 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
11. Ozuomba S. and Chukwudebe G.A, (2011). Static-Threshold-Limited On-Demandguaranteed Service For Asynchronous Traffic In Timely-Token Protocol. Nigerian Journal of Technology, Vol. 30, No. 2. June 2011
12. Ulm, J.M., (1982). A Timed Token Ring Local Area Network and its Performance Characteristics, Proc. 7th IEEE Conf. Local Computer Networks, pp. 50-56 (Feb. 1982)
13. Ross F.E, (1989). An Overview of FDDI: The fibre distributed data interface ", IEEE Journal on selected Areas in Communications, Vol. 7, pp. 1043 {1051, Sept. 1989
14. Weaver A.C. (1986). Local area networks and buses; an analysis, "Technical report, Flight Data Systems, NASA , Johnson Space Center, 1986.
15. Sevcik K C. and Johnson M J., (1987) . "Cycle time properties of the FDDI token ring protocol," IEEE Trans. Software. Eng., vol. SE-13, pp. 376-385, Mar. 1987.
16. Johnson M. J., (1987). Proof that timing requirements of the FDDI token ring protocol are satisfied," IEEE Trans. Commun., vol. COM-35, pp. 620-625, June 1987.
17. 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
18. 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.
19. 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
20. Cobb J. and Lin M., (2002). The On-Time timed-token protocol", Proc. IEEE GLOBECOM, 2002.
21. Agrawal G., Chen B., Weizhao Zhao, 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

Appendix

Algorithm P:

The Timed-Token MAC Algorithm Considered Under Non-Uniform Heavy Load Of Asynchronous Traffic

P1: During the ring's initialization, the following parameters are initialized at all nodes [10], [11].

P1.1 Define TTRT (that is τ) and N **P1.2** Define w_i for i = 0,1,...N-1

P1.3. Define $h_i = 0$ for i = 0, 1…N – 1 **P1.4** Define H_i for i = 0,1,...N-1

P1.5 Initialize $\varepsilon_i = H_i$ for i = 0,1,...N-1 **P1.6** Initialize $a_{(i-N)} = 0$ for i = 0,1,...N-1

P1.7 Compute $\varepsilon = \left(\sum_{i=0}^{i=N-1} (\varepsilon_i) \right) = H$ **P1.8** Compute $T = \left(\sum_{i=0}^{i=N-1} (H_i + w_i) \right)$

P1.9 Initialise Bi = 1 for i = 0,1,...N-1 **P1.10** Let n = N
P1.11 **Initialise Timer**
P1.11.1 **i = 0** **P1.11.2.** Define TRT_i = 0;
P1.11.3. Define LC_i = 0; **P1.11.4.** Define TRT_i = TTRT;
P1.11.5. Start TRT_i; TRT_i counts down **P1. 11.6** i = i +1
P1. 11.7 Pass the Token to Node i + 1
P1.11.8 IF (i < N) then Goto Q1.12 Else Goto Q2.1 Endif

P2.1: When TRT_i counts down to zero, that is TRTi = TTRT , the following actions take place:

P2.1 IF (TRTi = 0) Then LC_i = LC_i +1; TRT_i = TTRT ; Start TRT_i; TRT_i counts down End if.
P2.2 Check Frames that arrives at Node i
P2.3 IF (Frame is Token) Then Goto Step P2.6 Else Goto Step P2.4 End if.
P2.4 Process Frame (Store, Ignore, etc) **P2.5** Goto Step P2.1
P2.6 IF (LC_i $\geq$ 1) then TX = TRT_i ; THT_i = 0
Else TX = TTRT ; THT_i = TRT_i TRT_i = TTRT End if.

TRANSMIT SYNCHRONOUS TRAFFIC

P2.7 $LC_i = 0$ **P2.8** $\varepsilon' = \varepsilon - \varepsilon_I$ **P2.9.** TRT_i continues to counts down
P2.10. IF (TX - TRT_i < H_i) then Goto Step P2.11 Else Goto Step P2.14 End if.
P2.11. IF (Synchronous Frames are available) Then Goto Step P2.12 Else Goto Step P2.14 End if.
P2.12 Transmit Synchronous Frames; TRTi continues to counts down
P2.13 Goto Step P2.11 **P2.14** h_i = TX - TRT_i **P2.15** $\varepsilon_i = H_i - h_i$
P2.16 $\varepsilon = \varepsilon' + \varepsilon_i$ **P2.17** Goto P3.1

TRANSMIT ASYNCHRONOUS TRAFFIC

P3: When the token arrives early at node i, (i.e. LC_i = 0) the following actions take place:
P3.1 a_i = THTi
P3.2.1 IF (THT_i > 0) then
IF (Asynchronous Frames are Available) then IF (Bi = 0) then Bi = 1; n = n + 1; Endif
Else IF (Bi = 1) then ; Bi = 0; n = n – 1 Endif
Endif
Endif
P3.2.2. IF (THTi > 0) Then Goto Step P3.3 Else Goto Step P3.4 End if.
P3.3 IF (Asynchronous Frames are Available) then
Transmit Asynchronous Frame; THTi continues to counts down; Goto Step P3.2.2
Else Goto Step P3.4 End if.
P3.4 ai = ai – THTi **P3.5** i = i + 1 **P3.6** i = (i mod N) **P3.7** Pass the Token to Node i

NOTE: For Correspondence Contact : ***Simeon Ozuomba***
simeonoz@yahoo.com ; *08183476828*
Department of Electrical/Electronic and Computer Engineering, University of Uyo, Akwa Ibom , Nigeria.