

# STATIC-THRESHOLD-LIMITED ON-DEMAND GUARANTEED SERVICE FOR ASYNCHRONOUS TRAFFIC IN TIMELY-TOKEN PROTOCOL

S. Ozuomba<sup>a</sup>, G.A. Chukwudebe<sup>b</sup>, A.B. Obot<sup>a,c</sup>

<sup>a</sup>DEPARTMENT OF ELECTRICAL/ELECTRONIC & COMPUTER ENGINEERING, UNIVERSITY OF UYO, AKWA IBOM STATE, NIGERIA (*Email: simeonoz@yahoo.com*)

<sup>b</sup>DEPT. OF ELECTRICAL/ELECTRONIC ENGINEERING & ENGINEERING TECHNOLOGY, FEDERAL UNIVERSITY OF TECHNOLOGY OWERRI, NIGERIA (*Email: gloriachukwudebe@yahoo.com*)

<sup>c</sup>(*Email: akanobot2005@yahoo.co.uk*)

## Abstract

*In this paper, an improved Timely-Token protocol with enhanced best-effort service for improved capacity allocation to the asynchronous (that is, non real-time) traffic is proposed. Through analytical approach and the use of computer simulations, the improved Timely-Token protocol is compared with the existing Timely-Token protocol. In particular, if AT denotes a threshold value, then, when compared to the existing Timely-Token protocol, the improved protocol will allocate additional average of AT time units to the asynchronous traffic in every cycle.*

**Keywords:** On-Demand Guaranteed Service, Static Threshold Capacity, Best-Effort Service, Asynchronous Traffic, Capacity Reclaiming Mechanism, Timely-Token Protocol, Timed-Token Protocol

## 1. Introduction

In today's Local Area Networks (LANs), efficient support for both real-time and non real-time stations in the same communication network is essential [1], [2]. Such networks Media Access Control (MAC) protocols must provide not only bounded message transmission time, as required by the hard and soft real-time tasks, but also high throughput, as demanded by non real-time tasks [3], [4], [5], [6]. An attractive MAC approach for such 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 LAN [8], Fiber Distributed Data Interface (FDDI) [9], [10], [11],[12], [13], SAFENET [14], Manufactur-

ing Automation Protocol (MAP) [15], High-Speed Ring Bus [16], in PROFIBUS [17], and in wireless networks [18], [19],[20]. With the timed-token protocol, messages are grouped into two separate classes: the synchronous and the asynchronous messages. Synchronous messages arrive at regular intervals and are associated with deadline constraints. The idea behind the timed-token protocol is to control the token rotation time. At network initialization time, a protocol parameter called Target Token Rotation Time (TTRT) is determined which indicates the expected token rotation time. Each station is assigned a fraction of TTRT, known as synchronous capacity, which is the maximum time for which a station is permitted to transmit its synchronous messages in every token receipt. Once a node re-

ceives the token, it transmits its synchronous message, if any, for a time not more than its allocated synchronous capacity. It can then transmit its asynchronous messages only if the time elapsed since the previous token departure from the same node is less than the value of TTRT, i.e, only if the token arrived earlier than expected. Hence, while the synchronous messages are delivered through guaranteed service, the asynchronous messages are delivered through best-effort service. The Timely-Token protocol is a version of the timed-token protocol developed to improve the communication services provided by the existing timed-token protocols [21]. It solved the problems of token-lateness in FDDI [9] and starvation of asynchronous traffic in FDDI-M [10]. However, the Timely-Token protocol still presents a drawback for asynchronous traffic. The issue of improving the Timely-Token (MAC) protocol is address in this paper.

**1.1. Contributions and motivations**

The contributions of this work are in two parts, namely:

- i. The development of an enhanced best-effort mechanism for the Timely-Token protocol: In the Timely-Token protocol, a total of A time units are available for transmitting asynchronous messages in every cycle. With the existing Timely-Token best-effort mechanism, a node cannot deliver asynchronous messages in the current token receipt unless there are unused time units available from the previous cycle. However, the enhanced best-effort mechanism can allocate, at least, AT time units to asynchronous messages in every token receipt, even when there is no spare bandwidth. Consequently, for a system that is heavily loaded with asynchronous messages, the average time units per cycle allocated to the asynchronous messages in the existing Timely-Token is  $\left(\frac{N * A}{N + 1}\right)$  [21], whereas that of the improved Timely-Token is A [21], which is

an increase of  $\left(\frac{A}{N + 1}\right)$  time units per cycle (see Eq 47 in Section 3.3 and Discussion of Results in Section 4.3).

- ii. Performance analysis of the improved Timely-Token protocol: The performance analysis of the improved Timely-Token protocol under light load of synchronous traffic and with heavy load of asynchronous traffic was conducted. Specifically, analytical expressions for some key performance parameters (explained in section 2.4) , are derived, namely: Maximum Cycle Length , Average Cycle Length (C), and Average Asynchronous Traffic (capacity) Time Units Per Cycle (Av) . Besides, for various network configurations, the results of the analytical computations were validated with results obtained from the simulation of the protocol.

**1.2. Arrangement of the paper**

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 parameters and algorithm. The improved Timely-Token protocol is presented and analyzed in Section 3. In section 4, the improved timely-token protocol is compared against the existing Timely-Token protocol. Finally, concluding remarks and recommendations for further studies are stated in Section 5.

**2. The Timely-Token Protocol and its Parameters**

**2.1. Network Model**

The network model consists of a token ring network with N nodes as shown in Fig 1. Each node has a unique number in the range 0, 1, 2, ..., N-1. 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

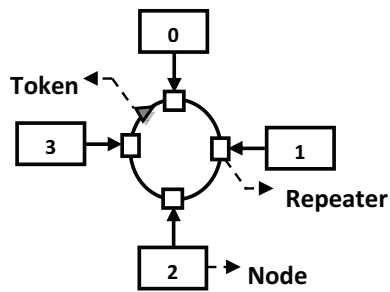


Figure 1: A 4-Station Token Ring Network.

appropriately node  $(i+1) \bmod N$ . The token frame circulates around the 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. A special bit pattern called token frame circulates around the 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,  $(i+1)$ . The sum of all such latencies in the ring is known as the ring latency or the token walk-time,  $W$ , where

$$W = \left( \sum_{i=0}^{i=N-1} (W_i) \right) \quad (1)$$

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

## 2.2. Message Model

Messages generated in the system at run time may be classified as either synchronous (real-time) messages or asynchronous (non real-time) messages. It has been demonstrated that in an arbitrary token ring network, where a node may have zero, one, or more streams of synchronous messages, can be transformed into a logically equivalent network with one stream of synchronous messages per node [23], [9]. Given this fact, in the following discussion, it is assumed that there

is one stream ( $s_i$ ) of synchronous (real-time) messages on each node  $i$ . The assumption of one stream per node simplifies the analysis without loss of generality. Also, the network is assumed to be free from hardware and software failures.

Basically, each node,  $i$  in the ring has a single stream of synchronous messages,  $s_i$ , where  $s_i$  is defined in terms of three parameters, namely, *Message Length* ( $C_i$ ), *Period Length* ( $P_i$ ), and *Message Deadline* ( $D_i$ ). Thus,  $s_i = \{C_i, P_i, D_i\}$ , where:

1. *Message Length*,  $C_i$ , is the maximum amount of time required to transmit a stream message. This includes the time required to transmit both the payload data and message headers.
2. *Period Length*,  $P_i$ , is the minimum inter-arrival period between consecutive messages in stream,  $s_i$  at node  $i$ . If the first message of node  $i$  is put in the transmission queue at time  $t_{i,1}$ , then the  $j$ -th message in stream  $s_i$  will arrive at time  $t_{i,j} = t_{i,1} + (j-1)P_i$ , where  $j > 1$ . For instance, if the first message arrive at time  $t$ , then the second message will arrive at  $t + P_i$  and the third message will arrive at  $t + 2P_i$ , as shown in Fig 2.
3. *Message Deadline*,  $D_i$ , is the relative deadline associated with messages in stream  $s_i$ , that is, the maximum amount of time that can elapse between a message arrival and the completion of its transmission. Thus, the transmission of the  $j$ -th message in stream  $s_i$  that arrives at  $t_{i,j}$  must be completed not later than  $t_{i,j} + D_i$ , which is the message's absolute deadline. Again, as an example, if the first message in the message stream,  $s_i$  arrives at time  $t$ , then it must be transmitted not later than  $t + D_i$ , as shown in Fig 2.

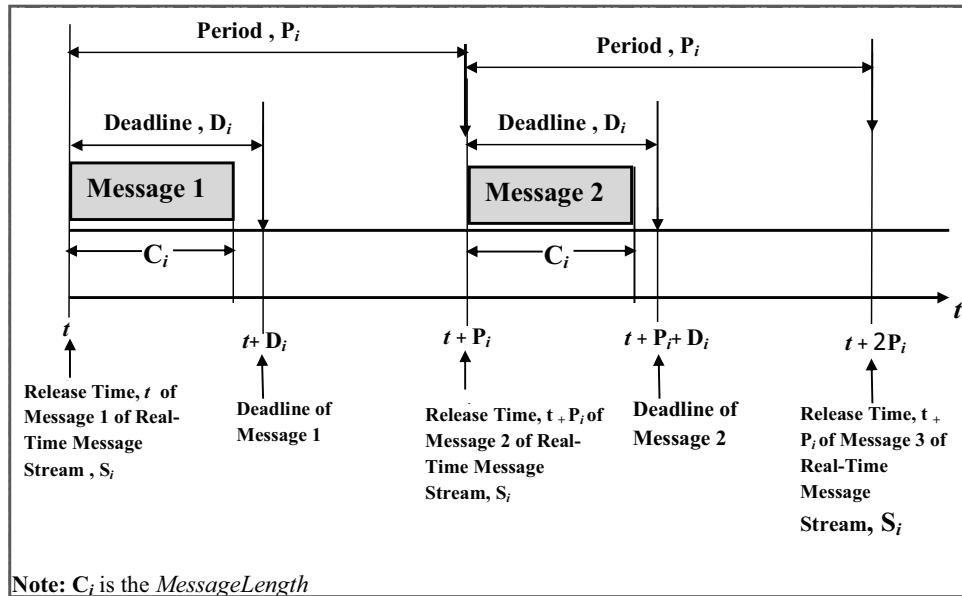


Figure 2: Model for the Synchronous (or Real-Time) Message Stream,  $S_i$  in Node  $i$ .

**2.3. The timely-token protocol parameters**

a) *Target Token Rotation Time, (TTRT):* TTRT is the time needed by the token to complete an entire round-trip of the network. The value of TTRT, denoted as  $\tau$ , is selected at network initialization such that it is sufficiently small to support the response time requirements of the real-time messages at all the nodes .

b) *Synchronous Capacity of Node  $i$  ( $H_i$ ):*  $H_i$  represents the maximum time for which a station,  $i$  is allowed to transmit synchronous messages during every token receipt. Let  $H$  be the total time units allocated to the synchronous traffic per cycle. Thus,

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

Let  $\sigma_i$  be the sum of  $w_i$  and  $H_i$ . Thus

$$H_i + w_i = \sigma_i \tag{3}$$

Let  $T$  be defined as

$$\sigma_0 + \sigma_1 + \dots + \sigma_{(N-1)} = \left( \sum_{i=0}^{i=N-1} (\sigma_i) \right) = T \tag{4}$$

$$\therefore T = \left( \sum_{i=0}^{i=N-1} (H_i) \right) + \left( \sum_{i=0}^{i=N-1} (W_i) \right) = H + W \tag{5}$$

Let,  $h_i$  be the used portion of  $H_i$  and  $\varepsilon_i$  the unused portion of  $H_i$  time reserved for the synchronous traffic in node  $i$  (where  $h_i \leq H_i$ ), Then, for a system that is lightly loaded with synchronous traffic, out of the  $H_i$  time units, only  $h_i$  time units are used leaving  $\varepsilon_i$  time units unused. Thus

$$h_i = H_i - \varepsilon_i \tag{6}$$

**Constraints:**

For proper operation of the timed-token protocol, the choice of values for the  $\tau$  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,

$$H \leq \tau - W, \text{ thus, } T \leq \tau \tag{7}$$

The *Deadline Constraint* states that every synchronous message must be transmitted be-

fore 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  $2 * TTRT$  [11] whereas, for the Timely-Token protocol, it can be as much as  $TTRT$  [21]. Therefore, in order for the deadline constraint to be satisfied in the Timely-Token protocol, it is required that for  $i = 0, 1, \dots, N - 1$ ,

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

Combining Eq 7 and Eq 8 gives

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

c) *Token Rotation Timer of Node i (TRT<sub>i</sub>):* TRT<sub>i</sub> is the cycle length or the time between two consecutive token receipts at node i.

d) *The Unused Synchronous Bandwidth, (ε):* 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,  $\varepsilon$  is added to the token, where  $\varepsilon$  represents the sum of unused synchronous bandwidth of all stations during the previous cycle. When the token arrives in station i,  $\varepsilon$  should also include the unused synchronous bandwidth of station i in the previous cycle [21].

e) *An Asynchronous-Limit Variable of Node i (THT<sub>i</sub>):* THT<sub>i</sub> is used to control the amount of time for which node i can transmit asynchronous messages.

#### 2.4. Defining the performance parameters used for the timed-token protocols

1. *Upper Bound On Cycle Length, or maximum Cycle Length,  $\max(t_i - t_{i-N})$*

The Upper Bound on Cycle Length or Maximum Cycle Length,  $\max(t_i - t_{i-N})$  is the worst-case token rotation time. It indicates the maximum delay any given node, i can experience between two consecutive token receipts at the node, i. If  $D$  is the minimum

deadline among stream deadlines, i.e.  $D = \min(D_i)$ , then according to the *Deadline Requirements*,  $\max(t_i - t_{i-N}) \leq D$  [25],[26].

Remarkably, the *Maximum Cycle Length*,  $\max(t_i - t_{i-N})$  affects the choice of TTRT and hence, the total synchronous bandwidth that can be supported. For instance, in FDDI,  $\max(t_i - t_{i-N}) = 2 * (TTRT)$  [11], hence,  $TTRT \leq D/2$  [25],[26],[27]. Thus, in FDDI, the maximum bandwidth available for synchronous messages is about half of the total bandwidth. This is the main drawback of the FDDI MAC protocol.

On the other hand, Shin and Zheng [10] proposed a modification of FDDI, called FDDI-M. In FDDI-M,  $\max(t_i - t_{i-N}) = TTRT$ , hence,  $TTRT = D$ . Consequently, by limiting the *Maximum Cycle Length* to TTRT, FDDI-M doubled the capacity to support real-time traffic when compared to FDDI. The same fact applies to the Timely-Token protocol [21], namely,  $\max(t_i - t_{i-N}) = TTRT$  and,  $TTRT = D$ .

2. *Average Time (or Bandwidth)Used By The Asynchronous Traffic Per Cycle ( $\hat{A}$ )*

The *Average Time (or Bandwidth)Used By The Asynchronous Traffic Per Cycle*,  $\hat{A}$  indicates the average bandwidth (time units) allocated to the asynchronous traffic per cycle. Higher value of  $\hat{A}$  alone does not indicate good performance when real-time messages are supported. In particular, the *Average Time (or Bandwidth)Used By The Asynchronous Traffic Per Cycle*,  $\hat{A}$  is determined by the THT<sub>i</sub> timer/counter of the best-effort capacity allocation in the timed-token MAC protocol.

3. *Average Cycle Length ( $\hat{c}$ )* The Average Cycle Length,  $\hat{c}$ , is the sum of the average bandwidth used per cycle to deliver the real-time and non real-time messages and also the token walk-time,  $W$ . Again, for any given TTRT,  $H$  and  $W$ , and *Maximum Cycle Length* =  $\tau$ , a higher value of  $\hat{c}$  indicates better performance, since this means that, on average, more messages are delivered per cycle. However, if two protocols achieve the same value of  $\hat{c}$  and *Maximum Cycle Length*

=  $TTRT$  for any given  $TTRT$ ,  $H$  and  $W$ , then, the protocol that has higher value of  $\hat{A}$  has better performance. This is because more non real-time messages will be delivered without violating the timing requirements of the real-time messages.

**2.5. The timely-token media access control (MAC) protocol**

In this section, the Timely-Token Media Access Control (MAC) algorithm developed by Cobb and Lin, [21] is presented here as *Protocol Q MAC Algorithm* while flowchart is presented in Fig 3a.

**Protocol Q MAC Algorithm**

**Q1: NETWORK INITIALISATION CYCLE**

During the first token rotation, to initialize timers, no station is allowed to transmit any packets. First,  $TTRT$ , that is  $\tau$  is defined to satisfy the deadline requirements of every synchronous message in the network. Then, the following two parameters are also defined,  $w_i$ ,  $H_i$  for  $0 \leq i \leq N-1$ . In addition,  $h_i$  is reset to zero. So, from Eq 6,  $\varepsilon_i = H_i$  for  $0 \leq i \leq N-1$ . Hence,

$$\varepsilon_i = H_i \text{ for } 0 \leq i \leq N - 1 \quad (10)$$

$$\varepsilon = \left( \sum_{i=0}^{i=N-1} \varepsilon_i \right) = \left( \sum_{i=0}^{i=N-1} H_i \right) \text{ for } 0 \leq i \leq N-1 \quad (11)$$

In summary, during the network initialization, the following parameters are defined, initialized or computed:

**Q1.1** Define  $TTRT$ , that is  $\tau$

**Q1.2** Define  $w_i$  and  $H_i$  for  $0 \leq i \leq N-1$ .

**Q1.3** Initialize  $h_i = 0$  and  $\varepsilon_i = H_i$  for  $0 \leq i \leq N-1$ , then, compute  $\varepsilon = \sum_{i=0}^{i=N-1} (\varepsilon_i) =$

$$\sum_{i=0}^{i=N-1} H_i = H.$$

**Q1.4** Compute  $T = \left( \sum_{i=0}^{i=N-1} (\varepsilon_i + w_i) \right)$ ;

**Q1.5** Initialize the timer,  $TRT_i = 0$ ; Start  $TRT_i$ ;  $TRT_i$  counts up.

**Q2: DATA TRANSMISSION CYCLES**

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

**Q2.1**  $THT_i = TTRT - \varepsilon - TRT_i$ ;

**Q2.2**  $\varepsilon' = \varepsilon - \varepsilon_i$ ,

**Q2.3**  $TRT_i = 0$ ;

**Q2.4** Start  $TRT_i$ ;  $TRT_i$  counts up.

**Q2.5** 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.  $h_i$  is assigned the number of time units used by the synchronous transmission, that is,  $h_i = TRT_i$ . Then,  $\varepsilon_i = (H_i - h_i)$ ;

**Q2.6**  $\varepsilon = \varepsilon' + \varepsilon_i$

**Q2.7** Let  $a_i$  be the total time units used for the transmission of asynchronous traffic in node  $i$  in the current cycle.

**Q2.7.1**  $a_i = THT_i$ ; Start  $THT_i$  timer,  $THT_i$  counts down.

**Q2.7.2** If  $THT_i > 0$ , then, station  $i$  transmits asynchronous packets for a period up to  $THT_i$  time units, or until all its asynchronous packets are transmitted, whichever occurs first.  $a_i$  is then assigned the number of time units used for the asynchronous traffic transmission. Thus,  $a_i = a_i - THT_i$ . Hence,  $a_i \leq \max(0, TTRT - \varepsilon - TRT_i)$ . Note that  $a_i = 0$  means that no asynchronous traffic is transmitted.

**Q2.8** The token is released to the next node  $i+1$  (or more appropriately  $(i + 1) \text{ Mod } N$ )

**3. The Proposed Improved Timely-Token Media Access Control (MAC) Protocol**

The flowchart of the improved Timely-Token protocol is presented in Fig 3b while the detailed algorithm is given here as *Protocol P MAC Algorithm*.

**3.1. Outline of the proposed improved timely-token protocol MAC algorithm**

**Protocol P (MAC Algorithm)**

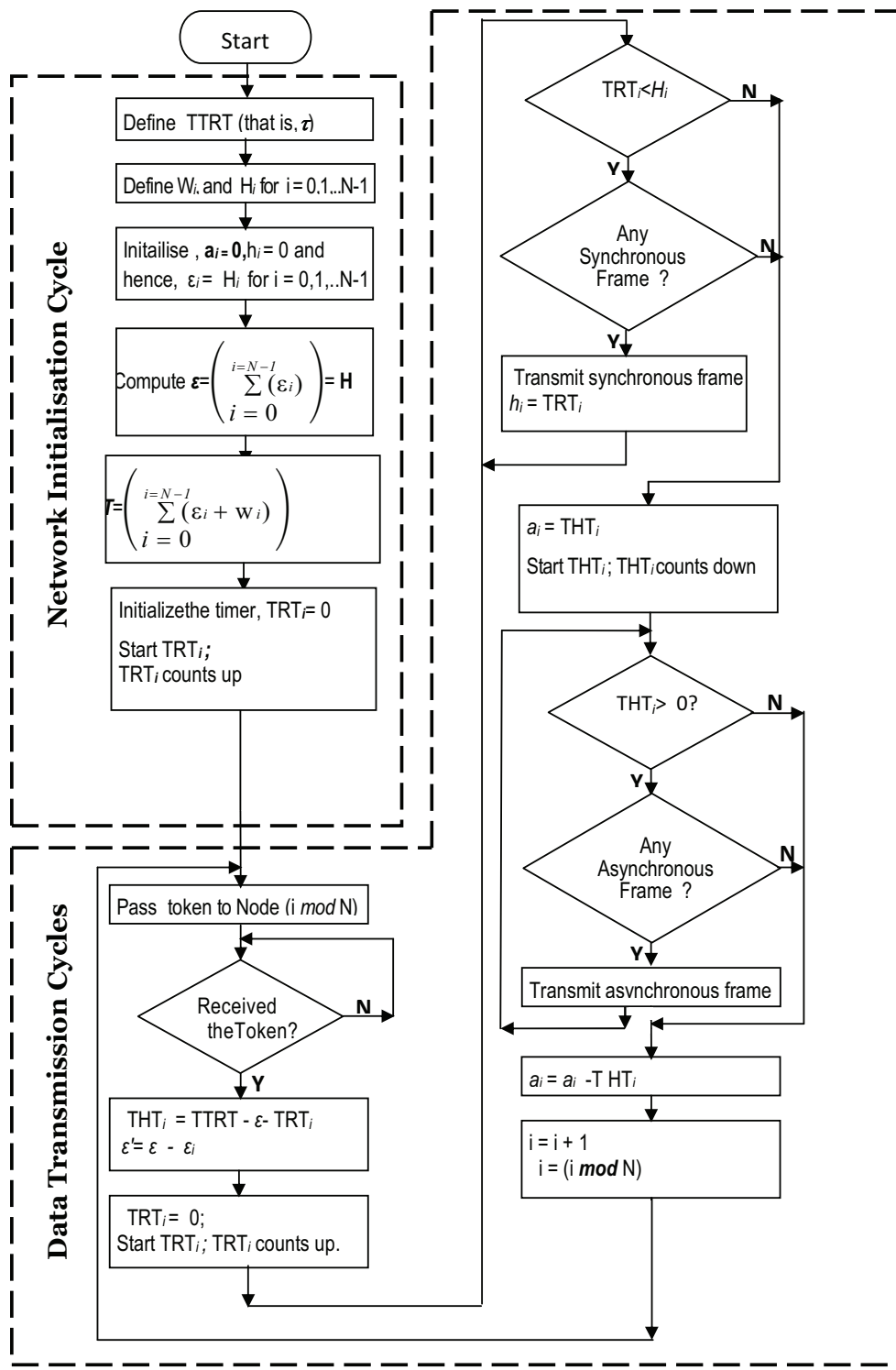


Figure 3a: Flowchart of the existing Timely-Token MAC Algorithm.

**P1: NETWORK INITIALISATION CYCLE**

During the first token rotation, to initialize timers, no station is allowed to transmit any packets. First, **TTRT**, that is  $\tau$  is defined to satisfy the deadline requirements of every synchronous message in the network. Then, the following two parameters are also defined,  $w_i, H_i$  for  $0 \leq i \leq N - 1$ . In addition,  $h_i$  is reset to zero. So,  $\varepsilon = \sum_{i=0}^{i=N-1} (H_i) = \sum_{i=0}^{N-1} H_i$  where  $\varepsilon_i = H_i$  for  $0 \leq i \leq N - 1, \varepsilon_i = H_i$

In summary, during the network initialization, the following parameters are defined, initialized or computed:

**P1.1** Define **TTRT**, that is  $\tau$ ;

**P1.2** Define  $w_i$  and  $H_i$  for  $0 \leq i \leq N - 1$ .

**P1.3** Initialize  $h_i = 0$  and  $\varepsilon_i = H_i$  for  $0 \leq i \leq N - 1$ , then, compute  $\varepsilon = \sum_{i=0}^{N-1} (\varepsilon_i) =$

$$\sum_i^{N-1} H_i = H$$

**P1.4** Compute  $T = \left( \sum_{i=0}^{i=N-1} (\varepsilon_i + w_i) \right)$  ;

**P1.5** Compute  $A_T = \frac{\tau - T}{N}$

**P1.6** Initialize the timer,  $TRT_i = 0$ ; Start  $TRT_i$ ;  $TRT_i$  counts up.

**P2: DATA TRANSMISSION CYCLES**

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

**P2.1**  $THT_i = TTRT - \varepsilon - TRT_i$ ;

**P2.2**  $TRT_i = 0$ ;

**P2.3**  $\varepsilon' = \varepsilon - \varepsilon_i$

**P2.4** Start  $TRT_i$ ;  $TRT_i$  counts up.

**P2.5** If station  $i$  has synchronous packets, it transmits them for a time period of up to  $H_i$ , or until all its synchronous packets are transmitted, whichever occurs first.  $h_i$  is assigned the number of time units used by the synchronous transmission, that is,  $h_i = TRT_i$ . Then,  $\varepsilon_i = (H_i - h_i)$ ;

**P2.6**  $\varepsilon = \varepsilon' + \varepsilon_i$

**P2.7** Let  $a_i$  be the total time units used for the transmission of asynchronous traffic in node  $i$  in the current cycle and let  $a_i - N$  be

the total time units used for the transmission of asynchronous traffic in node  $i$  in the previous cycle.

**P2.7.1**  $THT_i = THT_i + \min(A_T, a_{i-N})$ ;

**P2.7.2**  $a_i = THT_i$ ; Start  $THT_i$  timer;  $THT_i$  counts down

**P2.7.3** If station  $i$  has asynchronous packets, it transmits them for a time period of up to  $THT_i$  or until all its asynchronous packets are transmitted, whichever occurs first, where  $THT_i = \max(0, TTRT - \varepsilon - TRT_i) + \min(A_T, a_{i-N})$ . Thus,  $a_i \leq \max(0, TTRT - \varepsilon - TRT_i) + \min(A_T, a_{i-N})$ . Note that  $a_i = 0$  means that no asynchronous traffic is transmitted. Finally,  $a_{i-N} = a_i$ . Thus, in the next token receipt  $a_{i-N} = a_i$

**P2.8** The token is released to the next node  $i+1$  (or more appropriately  $(i + 1) \bmod N$ )

**The differences in the best-effort capacity allocation capabilities of the Timed-Token Protocols**

In a heavily loaded system,  $THT_i$  for FDDI, FDDI-M and Timely-Token protocol is expressed as follows;

- For FDDI [24], [22], [21],

$$\max(0, THT_i = TTRT - TRT_i) \quad (12a)$$

- For FDDI-M [25], [21],

$$\max(0, THT_i = TTRT - \sum_{i=0}^{i=N-1} (H_i) - TRT_i) \quad (12b)$$

- For Timely-Token protocol [21] were  $h_i \leq H_i$ ,

$$\max(0, THT_i = TTRT - \sum_{i=0}^{i=N-1} (H_i) + \sum_{i=0}^{i=N-1} (h_i) - TRT_i) \quad (12c)$$

- For the improved Timely-Token protocol (P2.7 of Protocol P MAC Algorithm in



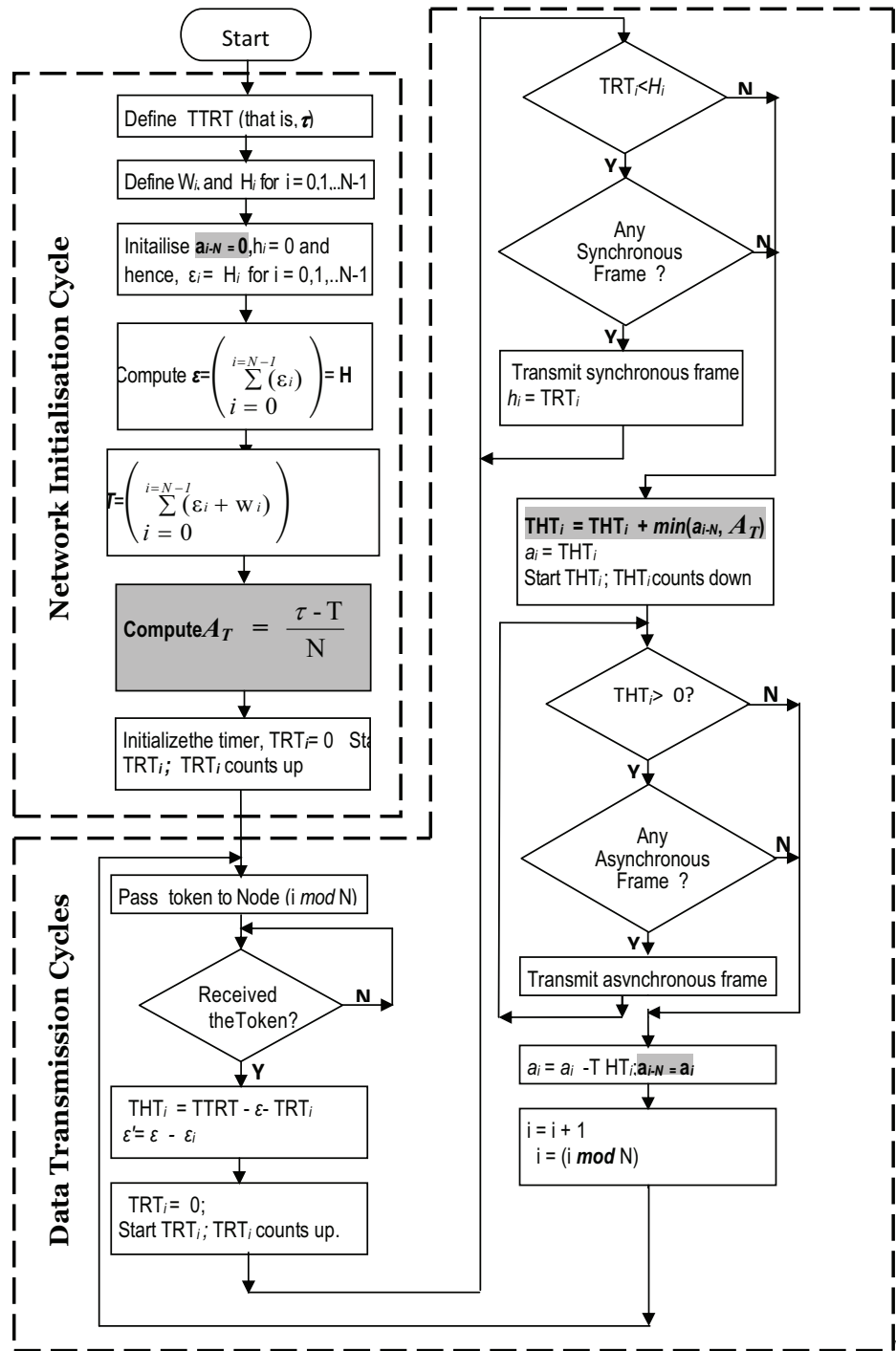


Figure 3b: Flowchart of the Improved Timely-Token MAC Algorithm. Note: In Fig 3b, the items on gray background are the distinguishing features of the Improved Timely-Token MAC Algorithm which are not in the Timely-Token MAC Algorithm of Fig 3a.

Section 3.1).

$$THT_i = \max(0, TTRT - \left(\sum_{i=0}^{i=N-1} (H_i)\right) + \left(\sum_{i=0}^{i=N-1} (h_i)\right) - TRT_i) + \min(A_T, a_{i-N}) \tag{12d}$$

where;  $A_T = \frac{A}{N}$ .  $a_{i-N}$  is the total time units used for the transmission of asynchronous traffic in node  $i$  in the previous cycle. That is,  $a_{i-N} = a_i$  in the previous token receipt in node  $i$ .

Among the three timed-token protocols listed, FDDI has the highest capacity allocation to the asynchronous traffic but it suffers from token lateness; Maximum Cycle Length exceeds  $TTRT$ . As regards FDDI-M and Timely-Token protocol, there is no token lateness; Maximum Cycle Length is equal to  $TTRT$ . Based on the expression for  $THT_i$ , the Timely-Token protocol has higher capacity allocation to the asynchronous traffic than FDDI-M. Consequently, the Timely-Token protocol is considered to perform better than FDDI-M and FDDI. Again, from Eq 10c and Eq 10d, the improved Timely-Token protocol has higher capacity allocation to the asynchronous traffic than the existing Timely-Token protocol while the two protocols have the same value for *Maximum Cycle Length*.

**3.2. Performance analysis of the improved timely-token media access control (MAC) algorithm**

For the sake of the analysis,  $TRT_i$  will be redefined in terms of time,  $t_i$  as follows;

**Definitions:-** Let  $t_0, t_1, \dots, t_{(N-1)}$  be the time at which the token reaches station 0,1, ..., N-1 for some given cycle. Also, let  $t_N, t_{(N+1)}, \dots, t_{(2N-1)}$  be the time at which the token reaches station 0, 1, ..., N-1 in the next cycle and so forth. Thus,  $t_i$  is the time at which the token reaches station  $(i \bmod N)$  in the cycle  $\lfloor \frac{i}{N} \rfloor$  where the given cycle is denoted as cycle 0 and where  $\lfloor x \rfloor$  denotes the integer

part of  $x$ . Let  $t_N, t_{(1-N)}, \dots, t_{(N-1)-N}$  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,  $TRT_i$ , the cycle length or the time between two consecutive token receipts at node  $i$  is given as;

$$TRT_i = \text{Cycle length for node } i = t_i - t_{i-N} \tag{13}$$

It is worthy to 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  $j = 0,1,2, \dots, N-1$ ) and cycle denoted by  $k$  can be computed as  $j = (i \bmod N)$  and  $k = \lfloor \frac{i}{N} \rfloor$ . According to the protocol operations in **P2.1** and **Q2.7.3** of **Protocol Q MAC Algorithm** in Section 3.1, when the token arrives at node  $i$ , then  $THT_i$  is determined as,

$$THT_i = TTRT - \varepsilon - TRT_i \text{ for } TTRT - \varepsilon > TRT_i \tag{14a}$$

Otherwise,

$$THT_i = 0 \text{ for } TTRT - \varepsilon = TRT_i \tag{14b}$$

Hence,

$$THT_i \leq \max(0, TTRT - \varepsilon - TRT_i) \text{ for all } TRT_i \tag{14c}$$

According to the protocol operations in **P2.7.1** of **Protocol Q MAC Algorithm** in Section 3.1,  $THT_i$  is updated as follows,

$$THT_i = THT_i + \min(A_T, a_{i-N}) \tag{15}$$

According to the protocol operations in **P2.7.2** and **P2.7.3**,  $a_i$  is defined as;

$$a_i = THT_i \tag{16}$$

For a system that is heavily loaded with asynchronous traffic,  $a_{i-N} \geq A_T$  for all  $i \geq N$ , thus,

$$\min(A_T, a_{i-N}) = A_T \text{ for } a_{i-N} \geq A_T \tag{17}$$

Then, Eq 15, Eq 16 and Eq 17 give

$$a_i = \max(0, TTRT - \varepsilon - TRT_i) + A_T \quad (18)$$

If  $TRT_i$  is replaced with  $t_i - t_{i-N}$ , and  $TTRT$  with  $\tau$  then, Eq 18, gives;

$$a_i = \begin{cases} \tau - \varepsilon - (t_i - t_{i-N}) + A_T \\ \text{when } \tau - \varepsilon > (t_i - t_{i-N}) \end{cases} \quad (19a)$$

or

$$a_i = A_T \text{ when } \tau - \varepsilon = (t_i - t_{i-N}) \quad (19b)$$

Again, for a heavily loaded system, Eq 19a and Eq 19b give;

$$a_i = \begin{cases} \max(0, \tau - \varepsilon - (t_i - t_{i-N})) + A_T \\ \text{for all } (t_i - t_{i-N}) \end{cases} \quad (20)$$

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

$$t_{i+1} \leq t_i + a_i + h_i + w_i \quad (21)$$

If Eq 6 and Eq 3 are applied into Eq 21, they give:

$$t_{i+1} \leq t_i + a_i + H_i - \varepsilon_i + w_i \quad (22)$$

$$t_{i+1} \leq t_i + a_i + \tau_i - \varepsilon_i \quad (23)$$

If Eq 19a and Eq 19b are applied into Eq 23 they give Eq 24a and Eq24b respectively

$$t_{i+1} \leq \begin{cases} t_{i-N} + A_T + \tau - \varepsilon + (\sigma_i - \varepsilon_i) \\ \text{for } (t_i - t_{i-N}) < \tau - \varepsilon \end{cases} \quad (24a)$$

$$t_{i+1} \leq \begin{cases} t_i + A_T + \sigma_i - \varepsilon_i \\ \text{for } \tau - \varepsilon = (t_i - t_{i-N}) \end{cases} \quad (24b)$$

Then, combining Eq 24a, Eq 24b for a system that is heavily loaded gives

$$t_{i+1} \leq \begin{cases} \max(t_i + A_T, t_{i-N} + A_T + \tau - \varepsilon_i) \\ + (\sigma_i - \varepsilon_i) \end{cases} \text{ for all } i \geq 0 \quad (25a)$$

where  $\sigma_i - \varepsilon_i = \sigma_{(i \bmod N)} - \varepsilon_{(i \bmod N)}$

Recall that during the network initialization, that is, from  $i = 0$  to  $i = N-1$ ,  $a_{i-N} = 0$  and so, by Eq 17,  $\min(A_T, a_{i-N}) = 0$ . Thus, for  $i = 0$  to  $i = N-1$ , Eq 25a becomes

$$t_{i+1} \leq \begin{cases} \max(t_i, t_{i-N} + \tau - \varepsilon) + (\sigma_i - \varepsilon_i) \\ \text{for } i = 0 \text{ to } i = N - 1 \end{cases} \quad (25b)$$

where  $\sigma_i - \varepsilon_i = \sigma_{(i \bmod N)} - \varepsilon_{(i \bmod N)}$

To properly address the peculiar nature of the *Network 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  $\tau'$  be  $\tau$  and  $t'_i$  be  $t_i$  for the special case [24], [22]. The initial condition for Eq 25a and for the special case is assumed to be  $t_0 = t'_0 = 0$  [24], [22]. Then, for the special case,  $\tau'$  and  $t'_i$  are defined as follows

$$\tau' = \tau - (T - \varepsilon) \quad (26)$$

$$t'_i = t_i + W - \sum_{j=0}^{j=i-1} (\sigma_j - \varepsilon_j) \quad (27)$$

Eq 27 differs slightly from the concept used in [24], [22], where  $t'_i = t_i - \sum_{j=0}^{j=i-1} (\sigma_j - \varepsilon_j)$ . 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  $W = \sum_{j=0}^{j=i-1} w_j$ . The addition of  $W$  to  $t_i$  in Eq 27 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 *Network 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 Eq 27,  $t_i$  can be expressed as

$$t_i = t'_i - W + \sum_{j=0}^{j=i-1} (\sigma_j - \varepsilon_j) \quad (28)$$

Eq 25a for the special case is

$$t'_i + 1 \leq \begin{cases} \max(t'_i + A_T, t'_{(i-N)} + A_T + \tau' - \varepsilon) \\ \text{where } h_i = 0 \text{ for all } i \geq 0 \end{cases} \quad (29)$$

The initial condition for Eq 25a and Eq 29 is given as; [24], [22].

$$t_i = t'_i = 0 \text{ where } ti < 0, t'_i < 0 \text{ for all } i < 0 \tag{30}$$

Iterating Eq 25a from  $i = 0$  to  $i=N-1$  (in this case Eq 25b) gives

$$0 \leq t_i \leq \tau - \varepsilon + \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) \text{ for } 1 \leq i \leq N \tag{31}$$

Substituting  $t_i$  from Eq 31 into Eq 27 gives

$$0 \leq t'_i \leq \tau - \varepsilon + W \text{ for } 1 \leq i \leq N \tag{32}$$

Using  $\tau$  as the upper bound on  $t'_i$  and  $H$  as the upper bound on  $\varepsilon$  for  $0 \leq i \leq N$ , induction over  $i$  can be applied to Eq 29 to give

$$t'_i \leq \tau - H + \lfloor \frac{i}{N+1} \rfloor (\tau' - \varepsilon + A_T) \tag{33}$$

The assumption that  $\tau$  is the upper bound on  $t'_i$  in Eq 33 is valid if Eq 26,  $\tau' = \tau - (T - \varepsilon) = \tau - T + \varepsilon$  is considered and that  $T = W + H$ . 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,  $\varepsilon_i = H_i$ , thus,  $\sum_{j=0}^{i-1} (\varepsilon_j) = \sum_{j=0}^{i-1} (H_j) = H$ . So, the upper bound on  $\varepsilon$  is  $H$ . Substituting  $t'_i$  from Eq 33 into Eq 28 gives

$$t_i \leq \tau - H - W + \lfloor \frac{i}{N+1} \rfloor (\tau' - \varepsilon + A_T) + \left( \sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) \right) \tag{34}$$

where

$$\sum_{j=0}^{i-1} (\sigma_j - \varepsilon_j) = \lfloor \frac{i-1}{N+1} \rfloor (T - \varepsilon + A_T) + \left( \sum_{j=0}^{(i-1) \bmod N} (\sigma_j - \varepsilon_j) \right) \tag{35}$$

Substituting  $\tau'$  from Eq 26 into Eq 34 and also applying Eq 35 into Eq 34 gives

$$t_i \leq \tau - (H + W) + \lfloor \frac{i}{N+1} \rfloor (\tau - (T - \varepsilon) - \varepsilon + A_T) - \lfloor \frac{i-1}{N+1} \rfloor (T - \varepsilon) + \left( \sum_{j=0}^{(i-1) \bmod N} (\sigma_j - \varepsilon_j) \right) \tag{36}$$

If  $H+W$  is replaced with  $T$  and  $(\tau - (T - \varepsilon) - \varepsilon)$  with  $\tau - T$ , Eq 36 gives

$$t_i \leq \left( 1 + \lfloor \frac{i}{N+1} \rfloor \right) (\tau - T) + \lfloor \frac{i}{N+1} \rfloor A_T + \lfloor \frac{i-1}{N+1} \rfloor (T - \varepsilon) + \left( \sum_{j=0}^{(i-1) \bmod N} (\sigma_j - \varepsilon_j) \right) \tag{37}$$

The time,  $t_{Nk}$  at which station zero receives its  $k^{th}$  token (that is  $i = Nk$  in Eq37) is given as

$$t_{Nk} \leq \left( 1 + \lfloor \frac{i}{N+1} \rfloor \right) (\tau - T) + \lfloor \frac{Nk}{N+1} \rfloor A_T + k(T - \varepsilon) \tag{38}$$

**Upper bound On Cycle Length,**

$\max(t_i - t_{i-N})$

If  $k=1$ , Eq 38 gives

$$t_N \leq \tau - T + T - \varepsilon \tag{39a}$$

$$t_N \leq \tau - \varepsilon \tag{39b}$$

$$\text{Cycle Length} = t_N - t_0 \text{ for } i = N \tag{40}$$

Substituting  $t_0$  from Eq 30 into Eq 40 and  $t_N$  from Eq 39b into Eq 40 gives

$$t_N - t_0 = \tau - \varepsilon \tag{41a}$$

Thus

$$\max(t_i - t_{i-N}) = \tau - \varepsilon \text{ for all } i \geq 0 \text{ and } \varepsilon \geq 0 \tag{41b}$$

Eq 41a and Eq 41b give the maximum cycle length for the Improved Timely-Token protocol under light load of synchronous traffic (where  $\varepsilon > 0$ ) but with heavy load of asynchronous traffic. When  $\varepsilon = 0$ , Eq 41a gives

$$t_N - t_0 = \tau \text{ when } \varepsilon = 0 \tag{42a}$$

Thus

$$\max(t_i - t_{i-N}) = \tau \text{ for all } i \geq 0 \text{ and } \varepsilon = 0 \tag{42b}$$

Eq 42a and Eq 42b give the maximum cycle length for the Improved Timely-Token protocol under heavy load of synchronous traffic where  $\varepsilon = 0$  and also with heavy load of asynchronous traffic. With  $\max(t_i - t_{i-N}) = \tau$ , it means that the token can never be late.

### Average Cycle Length, ( $\hat{c}$ )

Let ( $\hat{c}$ ) be the *Average Cycle Length*, and  $A_T = \lfloor \frac{\tau-T}{N} \rfloor$  Then, from Eq38, ( $\hat{c}$ ) is given as

$$\hat{c} \leq \lim_{k \rightarrow \infty} \left( \frac{t_{Nk}}{k} \right) \leq \lfloor \frac{N(\tau-T)}{N+1} \rfloor + \lfloor \frac{N(A_T)}{N+1} \rfloor + (T - \varepsilon) \quad (43)$$

$$\hat{c} \leq \lfloor \frac{N(\tau-T)}{N+1} \rfloor + \lfloor \frac{\tau-T}{N} \rfloor + (T - \varepsilon) \quad (44)$$

$$\hat{c} \leq (\tau - T) + (T - \varepsilon) \quad (45a)$$

$$\hat{c} \leq \tau - \varepsilon \quad (45b)$$

Eq 44, Eq 45a and Eq 45b give the average cycle length for the Improved Timely-Token protocol under light load of synchronous traffic (where  $\varepsilon > 0$ ) but with heavy load of asynchronous traffic. When  $\varepsilon = 0$ , Eq 45a and Eq 45b give

$$\hat{c} \leq (\tau - T) + T \quad (46a)$$

$$\hat{c} \leq \tau \quad (46b)$$

Eq 46a, Eq 46b and Eq 46c give the average cycle length for the Improved Timely-Token protocol under heavy load of synchronous traffic where  $\varepsilon = 0$ .

### Average (Channel Capacity) Time 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 Eq44 we have

$$A_V \leq \tau - T \quad (47)$$

Eq 47 gives the average time used by the asynchronous traffic per cycle irrespective of the load level of the synchronous traffic.

## 4. Comparison of Performance of the Two Timely-Token Protocols

### The Average Asynchronous Traffic Time Units Per Cycle ( $A_V$ )

In this paper, it has been shown that for the Improved Timely-Token protocol, the average

Asynchronous Traffic (Capacity) Time Units Per Cycle is given as  $A_{V(ITT)}$  of Eq 47, where,

$$A_{V(ITT)} \leq \tau - T \quad (48)$$

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,  $A_{V(TT)}$  is given as;

$$A_{V(TT)} \leq \lfloor \frac{N(\tau-T)}{N+1} \rfloor \quad (49)$$

For any given  $\tau$ , T and N,  $A_{V(ITT)}$  always exceeds  $A_{V(TT)}$  where,

$$A_{V(ITT)} - A_{V(TT)} = \lfloor \frac{(\tau-T)}{N+1} \rfloor \quad (50)$$

### a) The Maximum Cycle Length

There is no difference in the Maximum Cycle Length of the Improved Timely-Token protocol and the existing Timely-Token protocol. In the two protocols,  $\max(t_i - t_{i-N}) = \tau$ .

### 4.1. Simulation of protocol Q and protocol P

The simulation of the MAC algorithms for the Timely-Token protocol (Protocol Q) and the proposed Improved Timely Token protocol (**Protocol P**) was conducted with a program written with Visual Basic for Applications (VBA). The program runs in Microsoft Office Excel 2007 environment. The results obtained from the simulations are presented in tables and graph plots. In other to validate the results obtained from the analytical approach, we present mathematical expression that will relate the simulation results to the network performance parameters, namely; *Average Cycle Length ( $\hat{c}$ )*, *Average Asynchronous Traffic Time Units Per Cycle ( $A_v$ )* and *Maximum Cycle Length, ( $\max(t_i - t_{i-N})$ )*.

#### 4.1.1. The Average Cycle Length

Simulation results of **Protocol Q** showed that if the values of N, T,  $\tau$  and  $\varepsilon$  remain constant for at least a total of N(N+1)

consecutive token receipts (that is, from  $i$  to  $i + N(N+1) - 1$ ), then  $\hat{c}$  is given as the  $\text{MEAN}(TRT_i^\#)$ , (that is, the MEAN value of  $TRT_i^\#$  in any given node  $i$ ), where  $TRT_i^\#$  is given as;

$$TRT_i^\# = t_i - t_{i-N} \quad (51)$$

When  $M$  consecutive token receipts are considered, from node  $i$  to  $i+1, i+2, \dots, i+M$ , the cycles with respect to node  $i$  are;  $\lfloor \frac{i}{N} \rfloor, \lfloor \frac{i+N}{N} \rfloor, \lfloor \frac{i+2*N}{N} \rfloor, \dots, \lfloor \frac{i+M}{N} \rfloor$ . Then,

$$\text{MEAN}(TRT_i^\#) = \left( \frac{1}{(\lfloor \frac{M}{N} \rfloor + 1)} \right) \left( \sum_{x=\lfloor \frac{i}{N} \rfloor}^{\lfloor \frac{i+M}{N} \rfloor} (TRT_x^\#) \right) \quad (52)$$

where  $M \geq N(N+1) - 1$

Alternatively, we can express  $TRT_i^\#$  in terms of nodes and cycles as  $\text{MEAN}(TRT_{j,k}^\#)$  where  $\text{MEAN}(TRT_{j,k}^\#)$  stands for  $TRT_i^\#$  of node  $j$  in cycle  $k$ . Again, the cycles with respect to node  $i$  are  $\lfloor \frac{i}{N} \rfloor, \lfloor \frac{i+N}{N} \rfloor, \lfloor \frac{i+2*N}{N} \rfloor, \dots, \lfloor \frac{i+M}{N} \rfloor$ . Then,

$$\text{MEAN}_{j=(i \bmod N)}(TRT_{j,k}^\#) = \left( \frac{1}{(\lfloor \frac{M}{N} \rfloor + 1)} \right) \left( \sum_{k=\lfloor \frac{i}{N} \rfloor}^{\lfloor \frac{i+M}{N} \rfloor} (TRT_{j,k}^\#) \right) \quad (53)$$

where  $M \geq N(N+1) - 1$  and

$$\text{MEAN}(TRT_i^\#) = \text{MEAN}(TRT_{j,k}^\#) \quad (54)$$

#### 4.1.2. The Maximum Cycle Length

The maximum cycle length is  $\max(TRT_i^\#)$  for all  $i \geq 0$ .

#### 4.1.3. The Average Asynchronous Traffic Time Units Per Cycle

The simulation results of Protocol Q for the average time units used by the asynchronous traffic per cycle will be denoted as  $\text{MEAN}(a_i)$  or  $\text{MEAN}(a_{j,k})$ . The  $\text{MEAN}(a_i)$  is taken over  $M$  consecutive token receipt in all the nodes as follows:

$$\text{MEAN}(a_i) = \left( \frac{1}{(\lfloor \frac{M}{N} \rfloor + 1)} \right) \left( \sum_{x=i}^{x=i+M} (a_x) \right) \quad (55)$$

where  $M \geq N(N+1) - 1$

Then, just like  $\text{MEAN}(TRT_i^\#)$ , the value of  $\text{MEAN}(a_i)$  is defined in terms of  $a_{j,k}$  (where

$a_{j,k}$  means the time units used by the asynchronous traffic in node  $j$  in cycle  $k$ .) as follows:

$$\text{MEAN}(a_{j,k}) = \left( \frac{1}{(\lfloor \frac{M}{N} \rfloor + 1)} \right) \left( \sum_{k=\lfloor \frac{i}{N} \rfloor}^{k=\lfloor \frac{i+M}{N} \rfloor} \left( \sum_{j=\min\{(k+1)*N-1, (i+M)\}}^{j=k*N} (a_{j,k}) \right) \right) \quad (56a)$$

where  $M \geq N(N+1) - 1$

$$\text{MEAN}(a_i) = \text{MEAN}(a_{j,k}) \quad (56b)$$

#### 4.2. Numerical examples

Consider a ring network with four stations ( $N = 4$ ). The ring uses the Timely-Token protocol and the proposed Improved Timely-Token protocol for its MAC where the timed-token parameters are given as follows:  $\text{TTRT} = \tau = 100, w_i = 1$  for all the nodes,  $H_i = 20$  for all the nodes. We assume that the network is lightly loaded with synchronous traffic and that out of the  $H_i = 20$  reserved for synchronous traffic in every node, a constant value of  $\epsilon_i = 18$  is not used by the synchronous traffic in each node in every token receipt. With these given parameters, we have that  $H = 4(20) = 80$ . The simulation results for Protocol P (Timely-Token protocol) are shown in Table 1a and that of Protocol Q (Improved Timely-Token protocol) are shown in Table 1b.

The items in Table 2a and Table 2b are:  $TRT_o^\#$  is the token rotation time of node 0.  $\text{MEAN}(TRT_o^\#)$  is the mean of  $\sum TRT_o^\#$ .  $\sum h_i$  is the total time units used by the synchronous traffic per cycle.  $\text{MEAN}(\sum h_i)$  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.  $\text{MEAN}(\sum \epsilon_i)$  is the mean of  $\sum \epsilon_i$ .  $\sum a_i$  is the total time units used by the asynchronous traffic per cycle.  $\text{MEAN}(\sum a_i)$  is the mean of  $\sum a_i$ . Note that for all the MEANs are taken over  $N+1$  cycle, except for the first four rows in Tables 2a and 2b.

Table 1a: Part of the simulation results of the Timely-Token protocol (Protocol Q)

Cycle	Node 0				Node 1				Node 2				Node 3			
	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$
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

$N = 4; w_i = 1, W = 4; h_i = 20; H = 80; TTRT = \tau = 100; \varepsilon_i = 18, \epsilon = 72$

Table 1b: Part of the simulation results of the Timely-Token protocol (Protocol Q)

Cycle	Node 0				Node 1				Node 2				Node 3			
	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$	$TRT_i$	$a_i$	$h_i$	$\varepsilon_i$
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	4	2	18	16	12	2	18	28	0	2	18	28	0	2	18
3	28	4	2	18	28	4	2	18	20	8	2	18	28	0	2	18
4	28	4	2	18	28	4	2	18	28	4	2	18	24	4	2	18
5	28	4	2	18	28	4	2	18	28	4	2	18	28	4	2	18
6	28	4	2	18	28	4	2	18	28	4	2	18	28	4	2	18
7	28	4	2	18	28	4	2	18	28	4	2	18	28	4	2	18
8	28	4	2	18	28	4	2	18	28	4	2	18	28	4	2	18
9	28	4	2	18	28	4	2	18	28	4	2	18	28	4	2	18
10	28	4	2	18	28	4	2	18	28	4	2	18	28	4	2	18

$N = 4; w_i = 1, W = 4; h_i = 20; H = 80; TTRT = \tau = 100; \varepsilon_i = 18, \epsilon = 72$

Table 2a: Part of the simulation results of the Improved Timely-Token protocol (Protocol P).

$TRT_o\#$	$\sum \varepsilon_i$	$\sum h_i$	$\sum a_i$	C	$\varepsilon$	H	$A_v$
				Mean( $TRT_o\#$ )	Mean( $\sum \varepsilon_i$ )	Mean( $\sum h_i$ )	Mean( $\sum a_i$ )
0	80	0	0	0.0	0.0	0.0	0.0
4	72	8	16	2.0	76.0	1.0	8.0
28	72	8	0	10.7	74.7	3.3	5.3
28	72	8	16	15.0	74.0	4.5	8.0
28	72	8	16	17.6	73.6	5.2	9.6
28	72	8	16	23.2	72.0	6.8	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

Table 2b: Part of the simulation results of the Improved Timely-Token protocol (Protocol P).

$TRT_o\#$	$\sum \varepsilon_i$	$\sum h_i$	$\sum a_i$	C	$\varepsilon$	H	$A_v$
				Mean( $TRT_o\#$ )	Mean( $\sum \varepsilon_i$ )	Mean( $\sum h_i$ )	Mean( $\sum a_i$ )
0	80	0	0	0.0	0.0	0.0	0.0
4	72	8	16	2.0	76.0	1.0	8.0
28	72	8	4	10.7	74.7	3.3	6.7
28	72	8	16	15.0	74.0	4.5	9.0
28	72	8	16	17.6	73.6	5.2	10.4
28	72	8	16	23.2	72.0	6.8	13.6
28	72	8	16	28.0	72.0	8.0	13.6
28	72	8	16	28.0	72.0	8.0	16.0
28	72	8	16	28.0	72.0	8.0	16.0
28	72	8	16	28.0	72.0	8.0	16.0
28	72	8	16	28.0	72.0	8.0	16.0

**4.3. Discussion of result**

*4.3.1. Validating the analytical results with the simulation results*

**a) For the Timely-Token Protocol**

**Simulation results**

From Table 1a and 2a,  $N = 4, \varepsilon = 72$  and  $T = 84$ , then at steady state (cycle  $\gg N$ ) the following are observed; Mean( $TRT_o^\#$ ) =  $C = 24.8$  (row 12 column 5 of Table 2a) and Av (Timely-Token) = MEAN( $a_i$ ) = 12.8 (row 12 column 8 of Table 2a). Also, Maximum Cycle Length, max( $TRT_o^\#$ ) = 28 (see row 12 column 2 of Table 1a, and row 12 column 1 of Table 2a) where the node considered is node 0 and the cycle is 9.

**Analytical results**

Similar results were obtained from the analytical result shown in Table 3; in particular when  $N = 4, T = 84$  and  $\varepsilon = 72$  (row 11 column 1 of Table 3), Av (Timely-Token) = MEAN( $a_i$ ) = 12.8 (row 11 column 2 of Table 3). Also,  $C = \text{Mean}(TRT_o^\#) = 24.8$  (row 11 column 5 of Table 3).

**b) For the Improved Timely-Token Protocol**

**Simulation results**

From Table 1b and 2b,  $N = 4, \varepsilon = 72$  and  $T = 84$ , then at steady state (cycle  $\gg N$ ) the following are observed; Mean( $TRT_o^\#$ ) = 28 (row 12 column 5 of Table 2b) and Av (Improved Timely-Token) = MEAN( $a_i$ ) = 16 (row 12 column 8 of Table 2b). Also, Maximum Cycle Length, MAX( $TRT_o^\#$ ) = 28 (see row 12 column 2 of Table 1b, and row 12 column 1 of of Table 2b) where the cycle is 9 and the node considered is node 0.

**Analytical results**

Similar results were obtained from the analytical result shown in Table 3; in particular when  $N = 4, T = 84$  and  $\varepsilon = 72$  (row 11 column 1 of Table 3), Av (Improved Timely-Token) = MEAN( $a_i$ ) = 16 (row 11 column 3 of Table 3). Also,  $C = \text{Mean}(TRT_o^\#) = 28$  (row 11 column 6 of Table 3).

*4.3.2. Comparison of the performance of the protocols*

a) The Average Asynchronous Traffic Time Units Per Cycle for the Improved Timely-Token protocol,  $A_{v(ITT)}$  always exceed that of the Timely-Token protocol,  $A_{v(TT)}$  by  $\frac{(\tau-T)}{[N+1]}$  as shown in Table 3, (columns 2, 3 and 4) and in Fig 4 and Fig 5.

b) *The Average Cycle Length* for the Improved Timely-Token protocol,  $C_{(ITT)}$  always exceeds that of the Timely-Token protocol,  $C_{(TT)}$  by  $\frac{(\tau-T)}{[N+1]}$  as shown in Table 3 (columns 5, 6 and 7). It can also be seen from Table 3 that

$$A_{v(ITT)} - A_{v(TT)} = C_{(ITT)} - C_{(TT)} = \frac{(\tau - T)}{[N + 1]}$$

c) For any given  $\tau$  and T,  $A_{v(TT)}$  and  $A_{V(ITT)}$  vary at the same rate with respect to T, as shown in Fig 5. However,  $A_{v(TT)}$  decreases as N, (the number of nodes in the network) decreases, whereas,  $A_{V(ITT)}$  does not vary with N, as shown in Fig 4.

d) In all, the Improved Timely-Token protocol has higher throughput for the asynchronous traffic when compared with the Timely-Token protocol. The results also showed that for various network configurations (various values of T and N), the Improved Timely-Token protocol has higher overall throughput than the existing Timely-Token protocol.

**5. Conclusion and Recommendations**

**5.1. Conclusion**

In this paper, a new Timely-Token protocol was presented. The new Timely-Token protocol improved the ability of the Timely-Token protocol to support asynchronous traffic. The performance analysis of the improved Timely-Token protocol under light load of synchronous traffic but with heavy load of asynchronous traffic was also presented. The performance of the improved Timely-Token protocol and the existing Timely-Token protocol were compared. In all, in most



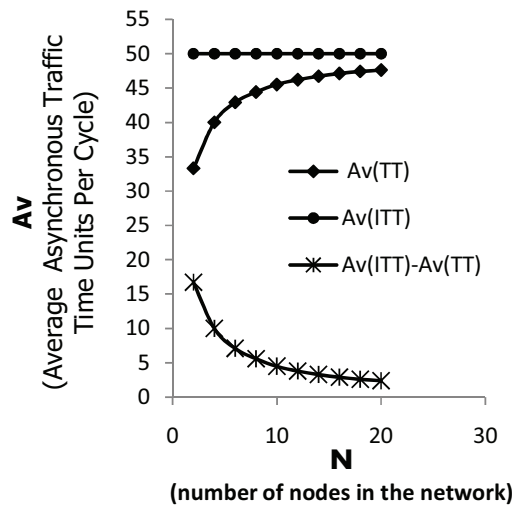


Figure 4: Graph of  $A_{V(TT)}$ ,  $A_{V(ITT)}$ , and  $A_{V(ITT)} - A_{V(TT)}$  against N (number of nodes in the network).

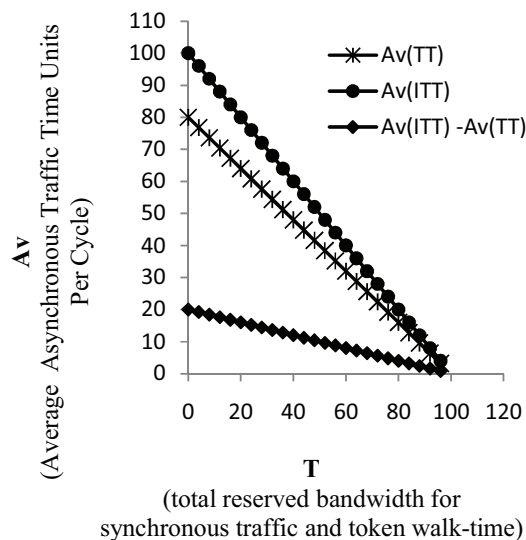


Figure 5: Graph of  $A_{V(TT)}$ ,  $A_{V(ITT)}$ , and  $A_{V(ITT)} - A_{V(TT)}$  against T (total reserved bandwidth for synchronous traffic and token walk-time).

traffic configurations, the improved Timely-Token protocol had higher throughput for the asynchronous traffic and also higher overall throughput when compared with the existing Timely-Token protocol. The improvement was achieved through an on-demand guaranteed bandwidth mechanism incorporated into the best-effort approach of the timely-token protocol.

## 5.2. Recommendations

In this paper, it is assumed that the system is heavily loaded with asynchronous traffic. In that case, every node has sufficient traffic to use all the bandwidth available to it. If however some nodes fail to use up the threshold bandwidth, AT guaranteed to them, then, they may lose the guaranteed bandwidth for some cycles. This situation will affect the performance of the new Timely-Token protocol. As such, further studies are needed to examine the effect of fluctuations in the load level of the asynchronous traffic on the Timely-Token protocol and then proffers solutions to the problems that might be discovered.

## References

1. Ricardo M. Survey of Real-Time Communication in CSMA-Based Networks. *Network Protocols and Algorithms*, Vol. 2, No. 1, 2010.
2. White P. P. RSVP and Integrated Services in the Internet: A Tutorial. *IEEE Communications Magazine* May 1997.
3. Indumathi G. and Murugesan K. 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, 2010.
4. Zhang S. and Burns 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. 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. Deterministic integration of hard and soft real-time communication over shared-ethernet. In *Proc. of Workshop of Tempo Real*, Curitiba, Brazil 2006.
  7. Nicholas M. and Wei Z. The timed-token protocol for real-time communications. *Computer*, vol. 27, no. 1, 1994, pp. 35-41.
  8. IEEE. The Institute of Electrical and Electronic Engineers, Token-passing bus access method and physical layer specifications. America national Standard ANSI/IEEE std. 802.4 - 1985.
  9. Biao C. and Wei Z. 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, 1992.
  10. Shin K. G. and Zheng Q. FDDI-M: A scheme to double FDDI's ability of supporting synchronous traffic. *IEEE Trans. on Parallel and Distributed Systems*, Vol. 6, No. 11, 1995, pp. 1125 -1131.
  11. Kenneth C. S. and Marjory J. J. Cycle time properties of the FDDI token ring protocol. *IEEE Trans. Software. Eng.*, vol. SE-13, 1987, pp. 376-385.
  12. Grow R. A timed token protocol for local area networks. *Proc. Electro82, Token Access Protocols*, Paper 17/3, May 1982.
  13. Chan E., Chen D., Cao J. and Lee C. H. The time Properties of the FDDI-M Medium Access Control Protocol. *The Computer Journal*, Vol. 82 No. 1, 1999, pp. 96-102.
  14. Dept. of Defense US. 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. MAP/TOP in CIM Distributed Computing. *IEEE Network*, vol.2, no. 3, 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, 1991, pp. 36 43.
  17. Tovar E., and Vasques F. 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. 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. 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. Analysis of the PROFIBUS Token passing protocol over wireless links. *Proc. of IEEE Int. Symposium on Industrial Electronics (IEEE-ISIE 2002)*, LAquila, Italy, July 2002, pp. 56-61.
  21. Cobb J. A. and Lin M. The timely token protocol. *Computer Communications* Vol .27 no.7, 2004, pp. 569-580.
  22. Ozuomba S. and Chukwudebe G.A. An Improved Algorithm For Channel Capacity Allocation In Timer Controlled Token Passing Protocols. *International Journal Of the Nigerian Computer Society (NCS)*, Vol. 9 No 1, 2003, pp 116 – 124.
  23. Agrawal G., Chen B., Zhao W., and Davari S. Guaranteeing synchronous message Deadlines with the Timed Token Access Control Protocol. *IEEE. Transaction on computers*, Vol 43. No.3, 1994 pp 237-239.
  24. Joseph W. M. and Fouad A. T. Throughput Analysis of a Timer-Controlled Token

- Passing Protocol Under Heavy Load. *IEEE Trans. Comm*, vol 37, No7, 1989, pp 694 – 702
25. Franchino G., Buttazzo G. C., and Facchinetti T. Token Passing Techniques for Hard Real-Time Communication. In *Factory Automation*, Javier Silvestre (Ed.), INTECH, 2010, Page 241 – 264.
  26. Franchino G., Buttazzo G. C., and Facchinetti T. 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.
  27. Franchino G., Buttazzo G. C., and Facchinetti T. Time Properties of the BuST Protocol under the NPA Budget Allocation Schem. In *Proc. Of the Conference on Design, Automation and Test in Europe(2008)*, Munich, Germany, 10th -14th March 2008.

# STATUS OF ABATTOIR WASTES RESEARCH IN NIGERIA

S.L. Ezeoha<sup>a</sup>, B.O. Ugwuishiwu

DEPARTMENT OF AGRICULTURAL AND BIORESOURCES ENGINEERING, UNIVERSITY OF NIGERIA, NSUKKA, NIGERIA

<sup>a</sup>(Email: louiezeoha@yahoo.com)

## Abstract

*Literature review was done to investigate the potential of abattoir wastes to befoul the environment, or cause hazards to human health, and harm to living resources and ecological systems. Abattoir wastes include animal blood, horns, bones, animal faeces, paunch manure, and abattoir effluent. The review result shows that abattoir wastes have the potential to pollute surface waters, underground waters, abattoir/market environment, and consumables around the abattoir, especially when abattoir wastes are not properly treated and disposed off. Abattoir wastes should be managed to achieve allowable effluent standards, odour control, or to exploit the benefits locking in the wastes before safely and economically disposing the ultimate wastes. In order to develop optimized abattoir wastes management strategies that would ensure reduction in environmental pollution in Nigeria, this paper proposes some research considerations on the pollution potential of abattoir wastes in Nigeria. The paper aims at stimulating increased research in the area of abattoir wastes management in Nigeria in order to avoid the dangerous consequences of poorly managed abattoir wastes.*

**Keywords:** pollution, abattoir, abattoir wastes, paunch manure, animal manure

## 1. Introduction

One type of wastes that is of great concern in both urban and rural areas in Nigeria is abattoir or slaughter-house wastes. Almost everyday in all the urban and rural markets in Nigeria, animals are slaughtered and the meat sold to the public for consumption. Meat wastes originate from killing; hide removal or dehairing, paunch handling, rendering, trimming, processing and clean-up operations. Therefore, abattoir wastes often contain blood, fat, organic and inorganic solids, and salts and chemicals added during processing operations [1, 2].

In ruminants, the first stomach or paunch

contains undigested materials called paunch manure, which can contain long hairs, whole grains and large plant fragments. The faeces of livestock (animal manure) consist of undigested food, mostly cellulose-fibre, undigested protein, excess Nitrogen from digested protein, residue from digested fluids, waste mineral matter, worn-out cells from intestinal linings, mucus, bacteria, and foreign matter such as dirt consumed, Calcium, Magnesium, Iron, Phosphorous, Sodium, etc. [3, 4]. Abattoir effluent (waste water) has a complex composition and can be very harmful to the environment [5]. Therefore the importance of knowing the pollution potentials of abattoir wastes