

On-Time Timed-Token Protocol

Jorge A. Cobb¹ Miaohua Lin

Center for Advanced Telecommunications Systems and Services

The University of Texas at Dallas (EC-31)

Richardson, TX 75083-0688

cobb@utdallas.edu miaohua@utdallas.edu

Abstract—The token protocols of FDDI and FDDI-M support both synchronous and asynchronous traffic. However, FDDI suffers from a token-lateness problem, and FDDI-M may starve asynchronous traffic. To remove these two weaknesses, we propose the on-time timed-token protocol. Its main feature is the addition of information to the token that allows a station to determine whether the early arrival of a token is due to unused synchronous bandwidth or unused asynchronous bandwidth. We present a synchronous bandwidth allocation scheme for this protocol, and comparison is made against FDDI and FDDI-M.

I INTRODUCTION

We present a modification to the timed-token protocol [1], which is used in FDDI networks. The purpose of the timed-token protocol is the support of synchronous traffic in a token-ring network. Messages are grouped into two separate classes: *synchronous* and *asynchronous*. Synchronous messages arrive at regular intervals and have a deadline constraint, while asynchronous messages have no deadlines. During initialization, a protocol parameter, called Target Token Rotation Time ($TTRT$), is chosen to indicate the expected token rotation time in the ring. Each station i is assigned a fraction H_i of the $TTRT$. Fraction H_i is the maximum time the station is allowed to transmit its synchronous messages upon receiving the token. The remaining fraction of the $TTRT$, A^* , where $A^* = TTRT - \sum H_i$, is the time available to transmit asynchronous messages. Since asynchronous traffic is likely to persist in the future, it is important for the timed-token protocol to support a mixture of synchronous and asynchronous traffic.

A *Synchronous Bandwidth Allocation* scheme assigns a value of H_i to each station i . Several SBA schemes for FDDI have been proposed for the support of synchronous messages [2,3,4]. However, finding the optimal SBA remains a challenge, since none of the SBA schemes proposed is optimal [5] (an SBA is optimal if it satisfies a set of synchronous traffic streams whenever there exists some SBA scheme that satisfy this set [2]).

In FDDI, the token rotation time may exceed the $TTRT$ value. Due to the lateness of the token, an FDDI ring can use at most half of its bandwidth to transmit synchronous traffic [6]. To alleviate this deficiency, Shin et al. [6] proposed the FDDI-M protocol, which performs only two small changes to the FDDI protocol. As a consequence, the token is never late, and FDDI-M can support traffic with a larger range of

deadline constraints than FDDI.

However, in some cases, FDDI-M may not be able to transmit asynchronous traffic, although intuitively it should. That is, there are scenarios where the fraction A^* of the $TTRT$ is non-zero, yet the asynchronous traffic is starved. In addition, the analysis of SBA schemes for FDDI-M is quite unintuitive and unmanageable, and does not lead to closed form expressions. Therefore, although there is a large number of SBA schemes for FDDI, such as PA [7], FLA [7], MCA [2], EMCA [4], ELA [8] and EGA [9] (readers can get a better view of the most of the above SBA schemes and even more in [10] and [11]), FDDI-M has none (only the basic SBA exists, where $H_i > C_i$, $TTRT < P_{min}$, where C_i is the maximum message transmission time, and P_{min} is the minimum synchronous period).

We propose a protocol that not only guarantees the token is never late, but also that A^* time units can be used for asynchronous traffic in every round. In addition, our algorithm is simple and yields a straightforward SBA scheme.

II TOKEN PROTOCOLS AND TRAFFIC MODEL

All token protocols below operate on a ring consisting of a number of stations (repeaters), each connected to two others by a unidirectional medium that forms a single closed path. A small frame, called *token*, circulates when all stations are idle. A station desiring to transmit waits to grasp the token before transmitting. After transmitting, the station releases the token. Without loss of generality, we assume only one synchronous message stream per station. Agrawal et al. [7] showed that a token ring network with multiple synchronous streams per station can be transformed into a logically equivalent network with one synchronous stream per (logical) station.

A *synchronous stream set* contains one synchronous message stream per station. The message stream at station i , S_i , is described by a triple $S_i = (P_i, C_i, D_i)$ where P_i is the inter-arrival time of messages in the stream, D_i is an upper bound on message delay, and C_i is the maximum time needed to transmit a message of the stream. Although most SBA schemes assume $D_i = P_i$, we do not make this assumption. Also, messages may be fragmented into smaller packets for transmission, and packet transmission cannot be pre-empted.

III PREVIOUS TOKEN PROTOCOLS

In this section, we overview the FDDI and FDDI-M

¹ Supported in part by a grant from the Texas Advanced Research Program.

protocols. In addition, we describe a weakness in each of them. These weaknesses are overcome by the On-time Timed-Token protocol, which is presented in the next section.

A. FDDI Protocol

During ring initialization, each station declares a $TTRT$ value equal to one half of the requested delay bound of its synchronous messages. The minimum declared $TTRT$ is selected as the ring's $TTRT$. Each station is then assigned a portion of the $TTRT$ to transmit synchronous packets.

Each station has a token-rotation-timer (TRT) and a token-holding-timer (THT). The TRT always increases and the THT increases only when the station transmits asynchronous packets. When $TRT > TTRT$, TRT is reset to 0 and the token is marked as "late" by incrementing the station's late count L_c by one. To initialize all timers, no packets are transmitted during the first token rotation after initialization. Also, all L_c values are set to 0.

The duration of packet transmission is controlled by the timers, but no packet transmission is preempted. When station i receives the token, it does the following:

1. If $L_c > 0$, set $L_c := L_c - 1$ and $THT := TTRT$. Otherwise, $THT := TRT$ and $TRT := 0$.
2. If station i has synchronous packets, it transmits them for a time period up to H_i or until all the synchronous packets are transmitted, whichever occurs first.
3. If station i has asynchronous packets, it transmits them until the THT counts up to $TTRT$, or all asynchronous packets are transmitted, whichever occurs first.
4. Station i passes the token to station $(i + 1) \bmod N$.

B. Late Token in FDDI

Consider a ring network with four stations, namely, stations 1, 2, 3, and 4. Suppose there is no traffic (either synchronous or asynchronous) in the previous round before the token arrives to station 1. The ring parameters are given as follows: $TTRT = 100$, $H_i = 20$, $1 \leq i \leq 4$, and $\tau = 2$, where τ is the propagation delay of the token around the ring.

When station 1 receives the token, $TRT = \tau = 2$. Suppose at that time there are only asynchronous packets to be transmitted, and just at the time when the station starts to transmit, a synchronous message arrives. Then station 1 can transmit asynchronous packets for A_1 msec. ($A_1 = TTRT - TRT = 98$). Suppose after that moment every station has unlimited synchronous messages and asynchronous messages. Then, the TRT s of stations 2, 3, and 4 will be 118, 138, 158, and all of which are greater than $TTRT$. Thus, they will receive a late token, and cannot transmit asynchronous packets. However, when the token reaches station 1 again after this round, the token rotation time of the previous round will be $A_1 + \sum H_i + \tau - H_1 = 160$, which is a lot greater than one $TTRT$. It is mentioned that $TTRT$ is the expected token rotation time, but from this example we can see that the token rotation time may significantly exceed the $TTRT$. Note that the larger the $TTRT$, the lesser is the ability to satisfy sets of synchronous streams. Thus, the direct effect of late tokens is the reduced ability to

schedule synchronous traffic. In this example, the synchronous message arriving in station 1 will experience a waiting period of 160. Thus, to be satisfied, its period must be over 180 rather than only one $TTRT$ (100).

From the example above, we find that the lateness of the token is mainly due to a station being able to use the synchronous bandwidth of other users (especially that of the previous rounds). I.e., there is no way to distinguish the reason for an early token arrival: is this because synchronous bandwidth was not used, or because asynchronous bandwidth was not used? Thus, the bandwidth consumed in the current round may exceed what is expected, resulting in a late token.

C. FDDI-M Protocol

During initialization, each station declares a $TTRT$ equal to the delay bound of its synchronous messages. Similar to FDDI, the smallest delay bound is chosen as the ring's $TTRT$. Each station is then assigned a portion of the $TTRT$ to transmit synchronous packets. However, FDDI-M additionally defines $TTRT_m := TTRT - \sum H_i - T_p$, where T_p is the transmission time of a maximum-sized packet.

Similar to FDDI, each station has TRT and THT timers. Timer THT counts up while the station transmits asynchronous packets. FDDI-M differs by having the TRT timer count up only when the station is not transmitting synchronous packets. To initialize these timers, no packets are transmitted during the first token rotation after initialization.

When station i receives the token, it does the following:

1. $THT := TRT$ and $TRT := 0$.
2. If station i has synchronous packets, it transmits them for at most H_i msec.
3. If station i has asynchronous packets, it transmits them until the THT counts up to $TTRT_m$ or all of its asynchronous packets are transmitted, whichever occurs first.
4. Station i passes the token to station $(i + 1) \bmod N$.

D. Starvation of Asynchronous Bandwidth in FDDI-M

Recall the ring example in Section III-B, where there are four stations, and $TTRT_m = 100 - 80 = 20$ (let $T_p = 0$ for convenience). Suppose there is no traffic before the token reaches station 1 in round 0. In round 1, when the token reaches station 1, suppose there is plenty of both synchronous and asynchronous traffic available at all stations. Let h_i and a_i be the time used for synchronous and asynchronous traffic at station i . Then, $TRT_1 = \tau = 2$, $h_1 = H_1 = 20$, $a_1 = TTRT_m - TRT_1 = 18$. I.e., station 1 can transmit asynchronous packets up to the total ring asynchronous allocation, and also its own synchronous allocation. For the following stations, the values of TRT are 38, 58 and 78, respectively. Since these values are all larger than $TTRT_m$, these stations cannot transmit any asynchronous packets.

In round 2, assume again there is plenty of both traffic types at all stations. Recall that in FDDI-M, TRT does not increase when the station transmits synchronous traffic. Thus, when the token comes back to station 1, $TRT_1 = a_1 + \sum H_i + \tau - H_1 = 18 + 80 + 2 - 20 = 80$, which is larger than $TTRT_m$,

and station 1 is unable to transmit asynchronous traffic. Nonetheless, station 1 does transmit its synchronous traffic. For the following stations, the values of TRT are all 62 (counting all synchronous bandwidth used by other stations), but this is larger than $TTRT_m$. Thus, no asynchronous traffic is transmitted in this round even though there are enough asynchronous packets at every station. Even worse, the situation will last if there are still both enough synchronous packets in each station during the following rounds.

Note that the token does not carry information about the bandwidth usage in the round. Thus, FDDI-M does not know whether an early token is caused by unused synchronous bandwidth or unused asynchronous bandwidth. This lack of knowledge prevents FDDI-M from preventing late tokens while at the same time ensuring the transmission of asynchronous traffic.

IV THE ON-TIME TIMED-TOKEN PROTOCOL

In this section, we present the on-time Timed-Token Protocol (On-time TTP), which overcomes both weaknesses mentioned in the previous section.

In the above FDDI and FDDI-M examples, problems occurred because a station cannot distinguish between unused synchronous bandwidth and unused asynchronous bandwidth. To overcome this, an integer u_r is added to the token, where u_r represents the sum of unused synchronous bandwidth of all stations during the previous round. When the token arrives in station i , u_r should also include the unused synchronous bandwidth of station i in the previous round.

Each station has a timer that has three types of actions: start, stop and reset. We define the token rotation time again, since it is used frequently below. The *token rotation time* in the on-time TTP refers to the duration of time from receiving the token to the next time the station receives the token.

A. Protocol Constraints

In order to guarantee message deadlines, the following three constraints must be satisfied:

$C_i \leq TTRT - \tau$. The transmission time must be no more than the available portion of the $TTRT$, where τ is the portion of $TTRT$ that is not available for transmitting packets.

$\sum H_i \leq TTRT - \tau$. The sum of the synchronous bandwidths allocated to all stations in the ring should not be greater than the available portion of the $TTRT$.

$C_i \leq D_i$: The required transmission time must be no more than the deadline of the message.

B. Protocol Outline

We next present an outline of the on-time TTP.

1. Station i receives the token, timer stops;
2. Station i computes: $A_i = TTRT - T_i - u_r$, where T_i is the timer value of station i . If $A_i > 0$, i.e., if there is asynchronous bandwidth available, station i can send asynchronous packets up to A_i secs.
3. After sending the asynchronous packets (if any), the

timer resets and starts to count. Then station i sends its synchronous packets up to H_i . After sending the synchronous packets, station i updates u_r in the token as: $u_r := u_r - u_i$; $u_i := H_i - T_i$; $u_r := u_r + u_i$, where u_i is the unused synchronous bandwidth of station i during the previous round.

4. Station i passes the token to station $(i+1) \bmod N$.

Consider the following example, using a four-station ring. Let $TTRT = 100$, $H_i = 20$, $1 \leq i \leq 4$, $\tau = 2$. Then, $A^* = 18$. Suppose there is no traffic before the token arrives at station 1. Then, $T_1 = 2$, $u_r = 80$, $A_1 = TTRT - T_1 - u_r = 18 = A^*$. Thus, station 1 can transmit 18 msec. of asynchronous packets and 20 msec. of synchronous packets. Then, T_1 reaches 20. Previously $u_1 = 20$ and current $u_1 = 0$, so u_r is updated to $80 - 20 + 0 = 60$. When the token arrives at station 2, $T_2 = 2 + 20 + 18 = 40$, and $A_2 = TTRT - T_2 - u_r = 100 - 40 - 60 = 0$. Thus, T_2 is reset, and synchronous packets are sent for 20 msec. Then, with a similar calculation to station 1, u_r is updated to $60 - 20 + 0 = 40$. When token arrives to station 3, $T_3 = 2 + 40 + 18 = 60$, $A_3 = TTRT - T_3 - u_r = 0$, then similar to the calculation of station 2, u_r is updated to 20. For station 4, same as station 3, $A_4 = 0$, and u_r is updated to 0. Thus, when the token comes back to station 1, $u_r = 0$.

By continuing this calculation, we see that the synchronous and asynchronous bandwidth can both be used if every station has enough synchronous and asynchronous packets. Also, even though the bandwidth is fully utilized, the token is never late. We formalize this with the following theorems.

Theorem 1: *The token rotation time may reach $TTRT$, but it never exceeds $TTRT$ (it is never late).*

Theorem 2: *During each round, a total of A^* time units are always available for asynchronous transmission.*

In particular, if the on-time TTP is executed in the scenarios of Sections III-B and III-D, the token is never late, and the asynchronous traffic is not starved.

V SYNCHRONOUS BANDWIDTH ALLOCATION SCHEME

A Synchronous Bandwidth Allocation (SBA) scheme guarantees that synchronous messages are transmitted before their deadline. We next present a SBA scheme for the on-time TTP. We first define X_i as follows. Given any interval of size D_i , X_i is the minimum amount of time during which station i can transmit synchronous packets in this interval.

Theorem 3: *For the on-time TTP, $X_i = m_i \cdot H_i + \max \{[R_i - (TTRT - H_i)], 0\}$, where $m_i = \lfloor D_i / TTRT \rfloor$ and $R_i = D_i - m_i \cdot TTRT$.*

Figure 1 illustrates the values of R_i , m_i and θ_i used in our SBA scheme (θ_i is defined as $TTRT - R_i$). In general we have two cases: when $D_i \geq TTRT$ for all i , and when there is an i such that $D_i < TTRT$. We first consider the former case.

If $R_i = 0$, i.e., the last $TTRT$ ends right at the edge of the deadline, then we can divide C_i equally into m_i rounds so that

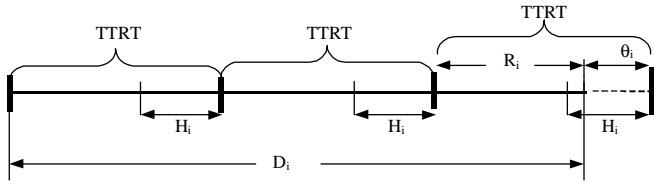


Figure 1. SBA scheme illustration

after summing the divisions, the total transmission time does not exceed the required transmission time C_i . Thus under $R_i = 0$, we can have $H_i = C_i / m_i$.

Assume $R_i \neq 0$. If $m_i \theta_i \geq C_i$, i.e., $\theta_i \geq C_i / m_i$, it means that after m_i rounds station i can transmit all the synchronous packets of the message, and need not use any synchronous bandwidth in round $(m_i + 1)$. Thus, we can also divide C_i equally by m_i rounds, and choose $H_i = C_i / m_i$. Otherwise, if $m_i \theta_i < C_i$, i.e., $\theta_i < C_i / m_i$, it means that after m_i rounds station i can not transmit all the synchronous packets of the message and needs to use some synchronous bandwidth in round $m_i + 1$. Thus, we adjust the synchronous bandwidth, so that station i can use the synchronous bandwidth equally in each round. Let $H_i = \theta_i + (C_i - m_i \theta_i) / (m_i + 1)$. Then, before the deadline, the time station i can use to transmit synchronous packets is as desired:

$$(m_i + 1) H_i - \theta_i = (m_i + 1) \theta_i + (C_i - m_i \theta_i) - \theta_i = C_i$$

Consider now the case where $D_i < TTRT$ for some i . Even if we let $\sum H_i = TTRT - \tau$, i.e., all bandwidth is given to the synchronous traffic, it is possible that the token cycle time may reach $TTRT$ (Theorem 1). Thus, the token cycle time is larger than D_i , i.e., the deadline will be missed. To avoid this situation, we only schedule a stream set if it satisfies: $\sum C_i < D_{min} - \tau$, where D_{min} is the minimum of all the message deadlines. Then, if we let $H_i = C_i$, without being affected by the asynchronous traffic, the sum of the maximum message transmission times of all the stations is $\sum C_i + \tau < D_{min} < TTRT$. To make all the stations unable to transmit asynchronous messages, we create a fake source; it does not transmit messages, but it is allocated a synchronous bandwidth $H_x = TTRT - \sum H_{real} - \tau$. Its message deadline can be set to be the maximum of all the deadlines and the maximum message transmission time can be set to 0.

The SBA scheme of the on-time TTP is thus as follows:

Case 1: If for all i , $D_i \geq TTRT$; then

If $m_i \theta_i \geq C_i$ or $R_i = 0$, let $H_i = C_i / m_i$

Otherwise, let $H_i = \theta_i + (C_i - m_i \theta_i) / (m_i + 1)$

Where $\theta_i = TTRT - R_i$

Case 2: If there is an i , s.t. $D_i < TTRT$ and $\sum C_i + \tau < D_{min}$;

Let $H_k = C_k$ for all real sources k , and let

$$H_x = TTRT - \sum H_{real} - \tau, \text{ for the fake source } x.$$

Theorem 4: Given the above SBA scheme, if the protocol constraint is satisfied, then the set of synchronous streams is

schedulable, i.e., no message deadline is violated.

VI COMPARISON WITH FDDI AND FDDI-M

In previous papers, the expression of X_i for FDDI has also been proposed. Here, X_i is the minimum available time that can be used during the period P_i by station i (because it is assumed that $P_i = D_i$). Thus, the deadline constraint is satisfied if and only if $X_i \geq C_i$. As we saw before, the most important timing property required for developing an optimal SBA scheme is the derivation of X_i . Chen et al. [2] first derived an X_i expression (on which SBA scheme MCA [2] was developed) as shown below:

$$X_i = (q_i - 1) \cdot H_i + \max \left[0, \min \left(r_i - \sum_{1 \leq h \leq n, h \neq i} H_h - \tau, H_i \right) \right] \quad (1)$$

where $q_i = \lfloor P_i / TTRT \rfloor$ and $r_i = P_i - q_i \cdot TTRT$. However, (1) is not exact enough and may give a value too low of X_i . Zhang and Burns [4] derived a tighter X_i expression as follows:

$$X_i = (m_i - 1) \cdot H_i + \max [P_i - I(m_i) + H_i, 0] \quad (2)$$

where m_i is an integer, $m_i \geq 1$, $I(m_i - 1) \leq P_i < I(m_i)$, and $m = m_i \vee m - 1 = m_i$, where

$$m = \left\lceil \frac{P_i \cdot (n+1) + n \cdot (TTRT - \sum_{h=1}^n H_h - \tau)}{n \cdot TTRT + \sum_{h=1}^n H_h + \tau} \right\rceil$$

and the function $I(v)$ is defined as

$$I(v) = v \cdot TTRT + \sum_{h=1}^n H_h + \tau - \left\lfloor \frac{v}{n+1} \right\rfloor \cdot (TTRT - \sum_{h=1}^n H_h - \tau)$$

Here $I(v)$ is the tight upper bound on the (maximum) time that could possibly elapse in the worst case before any station uses up its next v (where v is a positive integer) allocated synchronous bandwidths (H_i 's). Although (2) is a less pessimistic result than (1) for calculating the minimum available transmission time (because a larger value of X_i may be calculated), the improved X_i (upon which SBA scheme EMCA [4] was developed) is still not precise enough for an optimal SBA scheme to be developed.

In the on-time TTP, since the token is never late and the token rotation time is bounded by $TTRT$ (see Theorem 1 and 2), the maximum time that could elapse before any station uses up its next v allocated synchronous bandwidths is $v \cdot TTRT$, i.e., the $I(v)$ expression for the on-time TTP is $I(v) = v \cdot TTRT$. If we look at the $I(v)$ expression derived by Zhang et al., we may find that when the period under consideration is not very large (not more than $n \cdot TTRT$), the expression derived by Zhang et al. will become

$$I(v) = v \cdot TTRT + \sum_{h=1}^n H_h + \tau$$

Thus we can see that the $I(v)$ expression for the on-time TTP is less than that derived by Zhang et. al.. Thus, within a certain time period and with the same synchronous allocation for all stations, the on-time TTP will be able to use the token for more time than in FDDI. That is, it can send more packets than FDDI. This implies that the on-time TTP can schedule a larger range of streams sets than FDDI.

Besides the above advantage over FDDI, since Theorem 1 shows that the on-time TTP guarantees that the token is never late, the complexity of case analysis is significantly reduced compared to FDDI. This is also a reason why the on-time TTP is able to schedule a message set with larger range of deadlines.

In [6], it is mentioned that in FDDI network, at most $N = \min\{D/(2C), P/C\}$ synchronous connections can be established to ensure each station gets the token at least once every D units of time and at most $N \times C/P = \min\{1, D/(2P)\} \times 100$ Mbps of synchronous throughput can be provided (where P is the message generation period, C is the time needed to transmit a message and D is the deadline of a message). This is because we must have $TTRT \leq D/2$. However, in FDDI-M at most $\min\{1, D/P\} \times 100$ Mbps of synchronous traffic can be supported because the $TTRT$ (not $TTRT_m$) in FDDI-M can be set to be as large as D . That means FDDI-M can support double the synchronous traffic as FDDI. In the on-time TTP, similarly, the $TTRT$ can also be set to be as large as the message period, so that under the same conditions of FDDI-M, the on-time TTP can also support double the synchronous traffic transmission as compared to FDDI.

As another characteristic of the on-time TTP, it guarantees that in each round, a certain amount of asynchronous can be consumed, if desired. As shown earlier, in FDDI-M, in some cases, the total allowed asynchronous bandwidth is only consumed for a small part in a whole round.

However, there may be a tradeoff for removing late tokens. When the token arrives at a station, if there are no packets (neither synchronous nor asynchronous) in the previous round from the station's view, the station can transmit up to $TTRT - \tau$ asynchronous packets in FDDI, while in the on-time TTP the maximum amount of asynchronous packets that can be transmitted is only $TTRT - \sum H_i - \tau$. However, the token will quickly return to the station, and the station may then transmit again a total of $TTRT - \sum H_i - \tau$.

VII TOKEN IMPLEMENTATION

For an efficient implementation, a station must be able to quickly determine, with only a few bits delay, if the incoming frame is a token or not. Furthermore, it must quickly decide whether it should seize the token. One bit in the frame can be used to distinguish between data frames and token frames. Thus, the remaining issue is to quickly determine if the

station should seize the token.

A station with $H_i > 0$ should always seize the token, since it is allowed to transmit up to H_i time units of synchronous packets. Even if the station has no synchronous packets to transmit, seizing the token and then releasing it (which takes less than H_i time units) will not interfere with the guarantees of other stations.

Assume now that station i has $H_i = 0$. If the station has asynchronous packets to send, it must quickly determine if it should seize the token. From the outline of the on-time TTP we know that a station can transmit asynchronous packets for a total of $A_i = TTRT - T_i - u_r$ time units. Thus, it must look through all bits of u_r in the token and its timer T_i to decide if it should seize the token. A practical way to do this is to keep in a register the value of $TTRT - T_i$. The register has an initial value of $TTRT$ and counts down only. We compare the register value with the u_r (bit by bit) as the bits are received. We can see that a station can transmit no asynchronous packets if the register value is equal to u_r . On the other hand, if it is found that for some bit, the register is greater than u_r , then the station aborts forwarding bits of the token (and hence the next station cannot seize it since it receives a partial token) and seizes the token. Hence, the token can be seized with a delay of only a few bits per station.

REFERENCES

- [1] R.Grow, "A timed token protocol for local area networks," in Proc. Electro'82, Token Access Protocols, Paper 17/3, May 1982.
- [2] B.Chen, G.Agrawal, and W. Zhao, "Optimal synchronous capacity allocation for hard real-time communications with the timed token protocol," in Proc. IEEE RTSS'92, Dec. 1992, pp. 198-207
- [3] C.-C. Han and K. G. Sin, "A polynomial-time optimal synchronous bandwidth allocation scheme for the timed-token MAC protocol", in Proc. IEEE Infocom'95, Apr. 1995, pp. 875-882.
- [4] S. Zhang and A.Burns, "An optimal synchronous bandwidth allocation scheme for guaranteeing synchronous message deadlines with the timed-token MAC protocol", IEEE/ACM Trans. Networking vol.3, pp. 729-741, Dec. 1995.
- [5] S.Zhang and E.S.Lee, "The Nonoptimality of synchronous bandwidth allocation schemes for the timed token Protocol", IEEE Communications Letters, vol.4, NO. 3, March 2000.
- [6] K. G. Shin and Q. Zheng, "FDDI-M: A scheme to double FDDI's ability of supporting synchronous traffic", IEEE Transactions on Parallel and Distributed Systems, vol. 6, No. 11, Nov 1995
- [7] A. G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing synchronous message deadlines with the timed token medium access control protocol," IEEE Trans. on Comput., vol. 43, no. 3, pp. 327-339, Mar. 1994.
- [8] S.Zhang and A.Burns, "An efficient and practical local synchronous bandwidth allocation scheme for the timed-token mac protocol", Proc. IEEE Infocom, pp 920-927, 1996
- [9] S.Zheng and A.Burns, "Efficient global allocation of synchronous bandwidths for hard real-time communication with the timed token MAC protocol", Proc. of the Real-Time Computing Systems and Applications Conference, 1999, pp. 120-127.
- [10] Chen, D., Lee, V.C.S. and Chan, E, "On the ability of the FDDI-M protocol to support real-time traffic", Proc. of the Real-Time Computing Systems and Applications, pp. 51-57, 1998.
- [11] Chen, D., Chan, E. and Chan-Hee Lee, "Timing properties of the FDDI-M medium access protocol for a class of synchronous bandwidth allocation schemes", Proc. of the Int'l. Conference on Computer Communications and Networks, pp.825-832, 1998.