Improving the Timed Token Protocol

Yacine Bouzida and Rachid Beghdad

Laboratoire "Systèmes Informatiques", UERI
EMP (ex ENITA), BP 17, 16111 Alger,Algérie
Tel: (213) 21 86 34 69
{yacine.bouzida,rbeghdad}@eudoramail.com

Abstract. We present a critical study of "*the timed token*" real-time communication protocol. This protocol presents a drawback towards asynchronous messages. In fact, if all stations of the network have permanent synchronous and asynchronous messages, only the first station can transmit its asynchronous messages during a limited interval of time. Then, only synchronous messages will be transmitted until at least one station does not use all its synchronous capacity for the transmission of synchronous messages. The *regular timed token* protocol [7] has been developed to solve this problem. However, it still occurs in the case where the station uses all its synchronous capacity to send synchronous messages. The proposed here, called *improved timed token*, uses the main key ideas of the two previous ones and permits the transmission of synchronous messages in some critical situations where they cannot be transmitted when using either *timed* or *regular timed token* protocols.

Keywords: local networks, real-time protocol, non-real-time messages, scheduling messages, timed token, regular timed token, scheduling constraints.

1 Introduction

We address the issue of improving the *timed token medium access control (MAC) protocol*. This protocol is suitable for real-time applications not only because of its use in high bandwidth networks but also due to the fact that it has the important property of bounded access time which is necessary for real-time communications. The timed token protocol has been incorporated into many network standards, including the Fiber Distributed Data Interface (FDDI), IEEE 802.4, the High Speed Data Bus and the High Speed Ring Bus (HSDB/HSRB), and the Survivable Adaptable Fiber Optic Embedded Network (SAFENET). Many embedded real-time applications use them as backbone networks.

With the timed token protocol, messages are grouped into two separate classes : the *synchronous* class and the *asynchronous* class. Synchronous messages arrive in the system at regular intervals and may be 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 the TTRT, known as *synchronous capacity*, which is the maximum time for which a station is permitted to transmit its synchronous messages every time it

P. Lorenz (Ed.): ICN 2001, LNCS 2093, pp. 520–529, 2001
© Springer-Verlag Berlin Heidelberg 2001

receives the token. Once a node receives the token, it transmits its synchronous message, if any, for a time no 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.

The "timed token" protocol presents a drawback towards asynchronous traffic. Indeed, if all the stations of the ring have permanently real-time (synchronous) and non real-time (asynchronous) messages, the first station transmits non real-time messages during T_{TRT}. Thereafter, only the real-time messages will be transmitted until at least one of the stations does not use all its synchronous capacity for the real-time traffic.

The regular timed token protocol was developed to solve this problem. Nevertheless, the problem persists if only one station of the ring has real-time and non real-time messages, it will use all its synchronous capacity to transmit only the real-time messages.

Our contribution : the *improved timed token* protocol brings a solution to the encountered problems.

We present the "timed token" real-time MAC protocol and the regular timed token protocol in section 2. In section 3, we present our approach in details. Section 4 concludes the paper.

2 The Timed Token Protocol

This protocol uses the following parameters :

– T_{TRT} (Target Token Rotation Time) defines the target rotation time of the token.

– $H_{k,\ k=0..m-1}$ (Synchronous capacity of node i), where m is the number of the stations in the ring. This parameter represents the maximum time for which a station is permitted to transmit synchronous messages every time the station receives the token. Note that each station can be assigned a different H_i value. In this paper, we assume that $H_j=H_k,\ j\neq k\ \forall j,k \in \{0,..,m-1\}$.

– TRT_k (Token Rotation Time). It evaluates the cycle time (this counter is initialized to the T_{TRT} value and re-initialized to this value either when the token arrives early to the station or when the TRT_k is expired).

– LC_k (Late Counter of node k). This counter is used to record the number of times that TRT_k has expired since the last token arrival at node k.

– THT_k (Token Holding Time), defines the time during which the station k may transmit non real-time traffic.

Theoretically, the total available time to transmit synchronous messages, during one complete traversal of the token around the ring, can be as much as T_{TRT}. However, factors such as ring latency Θ and other protocol /network dependant overheads reduce the total available time to transmit synchronous messages. We denote the portion of T_{TRT} unavailable for transmitting synchronous messages by τ.

That is, $\tau = \Theta + \Delta$ where Δ represents the protocol dependant overheads (the token transmission time, asynchronous overrun, etc.). We define the ratio of τ to T_{TRT} to be

α. The usable ring utilization available for synchronous messages would therefore be *(1-α)*[4].

Thus, a protocol constraint on the allocation of synchronous capacities is that the sum total of the synchronous capacities allocated to all nodes in the ring should not be greater than the available portion of the Target Token Rotation Time (T_{TRT}), i.e.,

$$\sum_{k=1}^{m} H_k \leq T_{TRT} - \tau \quad \textbf{(1)}$$

In the following studied case (figure 1), we consider $\tau = 0$.

Timed Token Protocol [3]

```
For each station k, (k=0,1,2,…m-1) :
   THT_k←0 ;
   LC_k←0 ;              /*initialization procedure */
   TRT_k←T_TRT ;
   Starting the countdown of TRT_k
While the network is working :
   If TRT_k=0 then
      TRT_k←T_TRT
      LC_k←LC_k+1
   EndIf
   At the arrival of the token do :/* data transmission
*/

   0Case
        • LC_k=0 :          /* token early arrival case */
          THT_k←TRT_k
          TRT_k←T_TRT
          starting the countdown of TRT_k
          transmission of real time messages during H_k
          starting the countdown of THT_k
           While THT_k>0 and (∃ non real-time messages in
           wait): Transmission of non real-time messages
          token passing to the station (k+1)(modulo m)
        • LC_k=1 :       /* token late arrival case */
          LC_k←0
          Transmission of real-time messages during H_k
          token passing to the station (k+1)(modulo m)
        • LC_k>1 :                        /* error case */
          « error recovery» procedure

   EndCase

END
```

Critic :

Let us consider the situation where on one hand all stations have real-time traffic to transmit permanently during their synchronous capacity H_k and on the other hand, the first station uses all the time that it possess to transmit the non real-time traffic (it may transmit during an interval of time corresponding to T_{TRT}) (figure 1).

The next diagram represents the time filling of network transmissions.

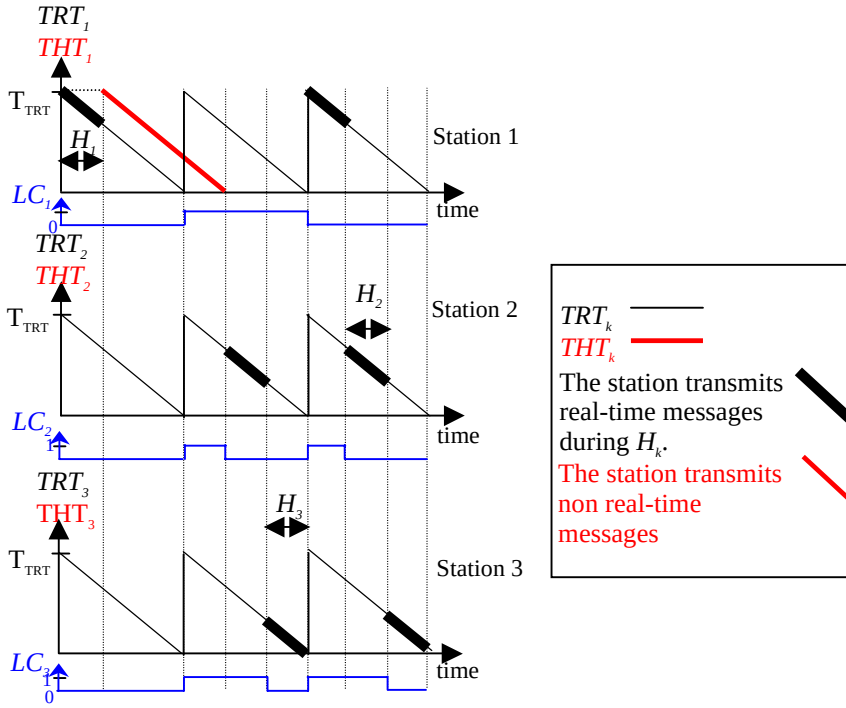

Fig. 1. Example of a critical case with the «timed token» protocol.

- We notice that in the most unfavorable case, all stations use all their synchronous capacity H_k and no longer give a chance to the non real-time traffic. This situation lasts until all stations do not use the totality of their synchronous capacity.
- If all stations respect protocol constraints, no LC_K will be able to reach a value greater than 1 (error situation).

2.1 The Regular Timed Token Protocol [7]

In this algorithm, the author considers T_{TRT} not as the token target time, but as the maximum time.

• We assign to a station *k*, a synchronous capacity H_k for the real time traffic (if any). And if the synchronous capacity H_k is not expired, the station transfers the non real-time traffic until the expiration of H_k.
• The constraint (1) is valid for this algorithm. For the example of figure 2, we assume that $\tau=0$.

Algorithm of the regular timed token protocol

```
For each station k, k=0,1,2,…,m-1 :
 THT_k←0 ; TRT_k←T_TRT ;     /* initialization procedure */
 Starting the countdown of TRT_k

For each station k, k=0,1,2,…m-1 :
 If TRT_k=0 : «ring recovery procedure»  /* error */
 At the arrival of the token, Do :/* data transmission */
      THT_k←H_k ;
      TRT_k←T_TRT ;
      Starting the countdown of TRT_k and THT_k
      while THT_k>0 and (real-time messages in wait) :
           transmission of real-time messages
      while THT_k>0 and (non real-time message in wait) :
           transmission of non real-time messages
      passing the token to the station (k+1)(modulo m)

 EndDo
End.
```

Critics :
1 – Let us consider the situation where all the stations of the ring have permanent real-time traffic. Consequently, no asynchronous messages will circulate in the network. However, the timed token algorithm allows the first station to transmit, at the beginning, the non-real-time message during T_{TRT}.
2 – When only one station of the network has, permanently, real-time and non real-time traffic, only real-time messages will be transmitted. This is the main drawback of this variant. (figure 2)

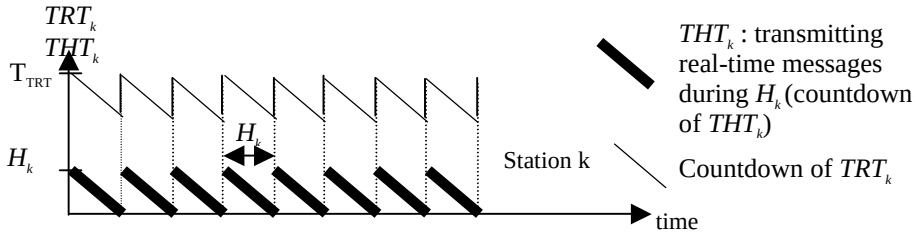

Fig. 2. An example of a critical case with the regular timed token protocol.

The previous diagram shows that no non-real-time messages will be transmitted although all the other stations remain idle.

The proposed protocol uses the same variables as those of the timed token protocol with the introduction of a new variable HR_K that denotes the remaining time from H_k after station *k* has sent all its real-time messages.

3 The Improved Timed Token Protocol

Principle :
- We assign to each station a time capacity H_k, that represents the maximum time, during which it can transmit the synchronous traffic.
- This protocol allows a station to transmit non real-time traffic whether:
- *it receives the token early, or*
- *$HR_k > 0$.*

The second principle allows the transfer of asynchronous messages (if any) of the current station, instead of sharing its remaining time HR_k with the other stations or to wait for the next reception of the token to transfer non-real-time messages.

The constraint (1) is still valid for this algorithm. For the example of figure 3, we assume that $\tau=0$.

Algorithm of the improved timed token protocol

```
For each station k, (k=0,1,2,…m-1)
   THT_k←0
   LC_k←0                  /* initialization procedure */
   TRT_k←T_TRT
   Starting the countdown of TRT_k
While the network is working, Do :
   If TRT_k=0 then
      TRT_k←T_TRT ;
      LC_k←LC_k+1
   EndIf
   At the arrival of token Do : /* data transmission */
   Case
        • LC_k=0          /* token early arrival case */
          THT_k←TRT_k
          HR_k←H_k
          TRT_k←T_TRT
          Starting the countdown of (TRT_k, HR_k)
          While HR_k>0 and (real-time messages in wait) :
                Transmission of real-time messages
          HR_k←THT_k+HR_k
          Starting the countdown of HR_k
          While HR_k>0 and(synchronous messages in wait):
            Transmission of non-real-time messages
          token passing to the station (k+1) (modulo m)
```

```
        • LC_k=1          /* token late arrival case */
          LC_k←0
          TRT_k←T_TRT
          HR_k←H_k
          Starting the countdown of (TRT_k , HR_k)
          While HR_k>0 and (real-time messages in wait) :
                Transmission of real-time messages
          While HR_k>0 and (non-real-time messages in wait) :
                Transmission of non-real-time messages
          token passing to the station (k+1) (modulo m)
        • LC_k>1                    /* error */
          « error recovery» procedure
    EndCase
END
```

Let us consider the case where we have three stations in the ring such that, at the beginning, the first station uses only a portion of its synchronous capacity to transmit real time messages. After that, all stations possess constrained messages permanently.

The next diagram represents the time filling of the network exchanges for the timed token protocol and the improved timed token protocol :

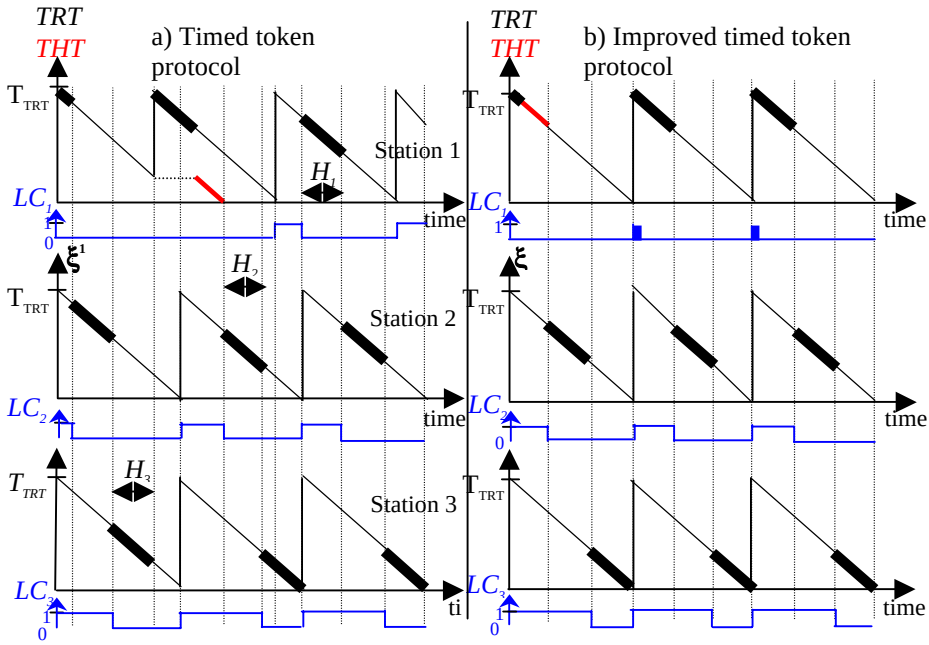

Fig. 3. First comparison between the « timed token » protocol and the improved timed token protocol. a) the first station waits for the next token arrival to transmit non real-time messages. b) In this case, it transmits them immediately. ξ^1 : in this case we assume that LC_1 was equal to 1 before the arrival of the token (at the beginning).

According to figure 3, the first station has real-time and non-real-time traffic. In the "timed token" protocol, the non-real-time messages will be transmitted at the second arrival of the token. However, the "improved timed token" protocol will make profit of the remaining synchronous capacity, used for the transmission of the real-time messages, to transmit non real-time messages at the first arrival of the token.
Another advantage of this protocol is the transmission of non-real-time messages during the first reception of the token in the case where one station has only non-real-time messages (without waiting for LC_k to be *0*). Each station that does not use all its synchronous capacity H_k uses its remaining time HR_k for the transmission of non-real-time traffic before passing the token to the following station (figure 4):

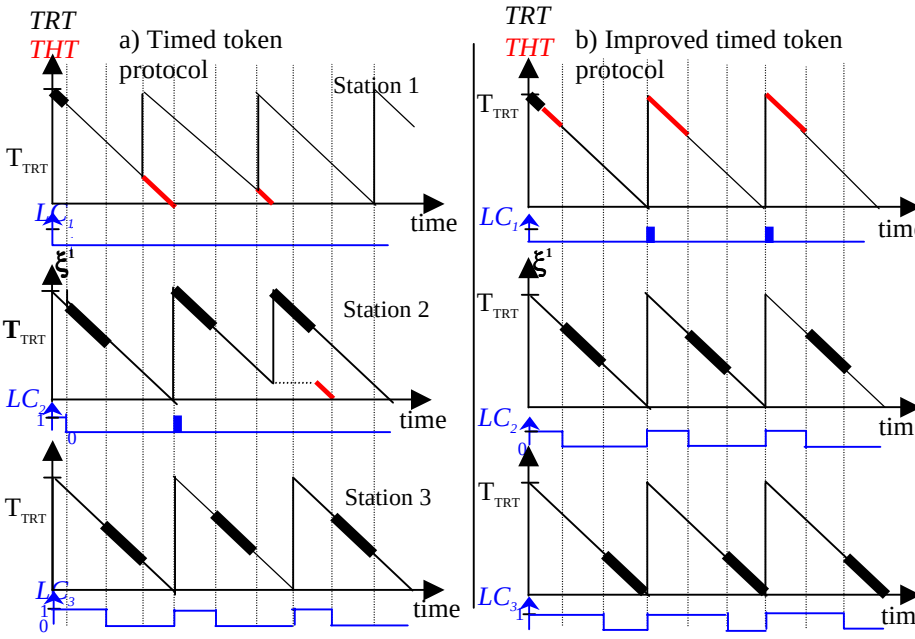

Fig. 4. Second comparison between the « timed token » protocol and the improved timed token protocol. ξ^1 : in this case, we assume that LC_1 was equal to 1 before the arrival of the token (at the beginning).

We notice that, under the « the timed token » protocol, station 2 transmits its non real-time messages by exploiting the remaining time of station 1. However, in the improved timed token protocol, the first station transmits its own asynchronous messages (if any) during HR_k before passing the token.

We give now a comparison between the regular timed token protocol and the improved timed token protocol.

Let us consider the situation where there is only one station *i* that possesses real-time and non-real-time traffic (figure 5):

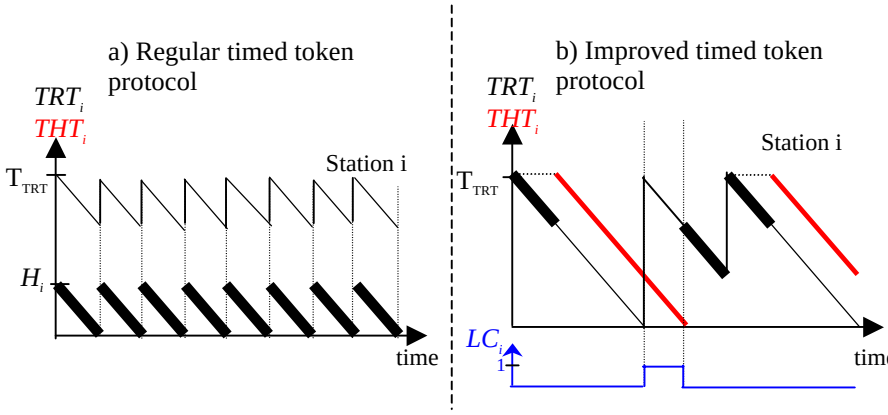

Fig. 5. A comparison between the regular timed token and the improved timed token protocols. a) non real-time messages are not transmitted. b) non real-time messages are transmitted.

Although the station may transmit non time-real messages, it will not be able to do it with the regular timed token protocol, elsewhere it will transmit only real-time messages. With the improved timed token protocol, the transmission of non-real-time messages is guaranteed.

4 Conclusion

We presented a critical study of the timed token protocol. After having given an overview of the timed token protocol and the regular timed token protocol and highlighting their drawbacks towards non real time traffic, we proposed an enhancement to the previous algorithms. The improved timed token protocol allows the transmission of asynchronous messages either without waiting the next arrival of the token or if there is a remaining time after transmitting real time messages.

References

1. G. AGRAWAL, and al. : Guaranteeing Synchronous Messages Deadlines with the Timed Token Medium Access Control Protocol , IEEE transactions on Computers, vol. 43, **3** (1994) 327-339.
2. B. Chen, S. Kamat, and W. Zhao. Fault-tolerant real-time communication in FDDI based networks. In Proceedings of the 16th RTSS, Pisa, Italy, 1995 (141-150).
3. R. GROW : A Timed Token Protocol for Local Area Networks; IEEE Electro 82', Token Access Protocols, 17/3, 1982.
4. J. N. Ulm. A timed token ring local area network and its performance characteristics. Proc.Conf. Local Computer Networks, Feb. 1982 (50-56).

5. M. Hamdaoui and P. Ramanathan : Selection of timed token parameters to guarantee message deadlines. IEEE/ACM Transactions on Networking, vol. 3, **3** (1995) 340-351.
6. N. Malcolm and W. Zhao : The timed token protocol for real-time communication. Computer, vol 27, **1** (1994):35-41.
7. F. VASQUES de CARVALHO : Sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous-couche MAC de réseaux locaux temps-réel , LAAS Phd Thesis, Toulouse University, June 1996.
8. S. Zhang and A. Burns : An optimal synchronous bandwidth allocation scheme for guaranteeing synchronous messages deadlines with the timed-token MAC protocol. IEEE/ACM Transactions on Networking, vol .3, **6** (1995) 729-741.