Buffer Insertion/Self-Token (BIST) Protocol for Multimedia LANs

Kuninobu TANNO*, Akio KOYAMA*, Toshihiro TAKETA* and Shoichi NOGUCHI**

* Faculty of Engineering, Yamagata University
**Faculty of Engineering, Nihon University
Email: tanno@eie.yz.yamagata-u.ac.jp

Abstract

This paper describes a new multimedia LAN protocol, referred to as the buffer insertion/self-token (BIST) protocol, combining the buffer insertion and the multiple-self-token ring protocols. The basic buffer insertion ring has the maximum spatial reuse on a ring by allowing stations to transmit packets concurrently. However, it is well known that the buffer insertion method suffers from the possibility of starvation. BIST prevents starvation and maintain fairness by assigning each station individual tokens, called self-tokens. Then a station is allowed to send packets only if it holds its own tokens, that is, each packet transmitted by the station is attached a self-token belonging to the station. That is why packet flow on a ring is regulated, and fairness is adjustable. Asynchronous and synchronous transmissions are supported by using two kinds of self-token, one is for asynchronous transmission and the other is for synchronous transmission. The effectiveness of the proposed protocol is demonstrated via simulation.

1. Introduction

Drastic change of LAN communication environment has demanded appearance of high throughput and low latency LANs[1]. The reasons are as follows: (1) the advent of high-speed peripheral devices, (2) resource sharing in system levels exchanges large amount of data between terminals and devices, (3) distributed processing is becoming popular, and this causes frequent intercommunication between processes, (4) multi-media communications requires not only a large amount of throughput but also synchronous transmission to include voice and image data exchanges, (5) LANs represent a key infrastructure for advanced internet community.

The Fiber Distributed Data Interface (FDDI)-I was standardized for 100 Mbps LAN protocol [2] to resolve such requirements. However, a follow-on LAN to FDDI is already anticipated for faster Mbps or Gbps LANs [3]. New gigabit LAN architectures are classified into two methodologies: improvement of conventional transmission media access architectures [4] and new development of switch-based architectures [5]. The former is of great advantage in inheriting and reusing the conventional LAN architectures [6]. But the latter takes advantage of new ATM-based switch architectures in expectation of the advent of new services [7].

Based on the former point of view [2], [8] and [9], this paper proposes and analyzes a new channel access method for ring topology, called "Buffer Insertion/Self-Token (BIST)" multimedia protocol. What the name of this protocol means is that the protocol make use of both the buffer insertion and the token ring protocols. Each station on a ring has private tokens referred to as self-token and a insertion buffer. This protocol offers a channel access method to be compatible with MAC sublayer in the data link layer [10].

With many new emerging network applications, it becomes increasingly important to support asynchronous and synchronous traffics. We enhance the basic register/self-token protocol [11] to propose a new multimedia ring protocol by preparing two kinds of self-token, one is for synchronous and the other is for asynchronous. The ackets attached synchronous tokens are transmitted under the station priority fashion, on the other hand the packets attached asynchronous tokens are transmitted under the ring priority fashion. Then these two priority control mechanisms guarantee required transmission bandwidth and transfer delay for synchronous traffic. Asynchronous traffic is transmitted by the best-effort strategy. The fairness is maintained for both traffics.

The paper is organized as follows. In Section 2, we

0-8186-7216-1/95 $04.00 © 1995 IEEE

explain the basic operation of this protocol. In Section 3, we discuss the priority transmission for synchronous traffic and the guarantee of required transfer delay. In Section 4, we explain how to assign priority for asynchronous traffic. In Section 5, we evaluate performance of this protocol via simulation. Section 6 is conclusions of this paper.

2. Basic operation

The basic buffer insertion ring [12] has the maximum spatial reuse on a ring allowing stations to transmit packets concurrently, but it needs an infinite buffer at each station or it suffered from the possibility of starvation depending on two priority transmission modes (see below) [13]. To reduce insertion buffer length or to prevent starvation and maintain fairness, private tokens belonging to individual stations are allocated to each station. These private tokens are called the self-token and held in the self-token buffer as in Fig. 1. Whenever a station send a data segment, it must attaches a self-token at the beginning of the segment. The self-token, the segment and some control information make a packet. If a self token is in the insertion buffer, a state of the self-token is free, but if it makes a packet, the state of the self-token is busy. The self-tokens are divided into for asynchronous packets and for synchronous ones.

The station priority transmission mode is adopted for asynchronous traffic, and the ring priority mode is adopted for synchronous traffic. We define two transmission modes, and explain the transmission operations in a station as shown in Fig. 1.

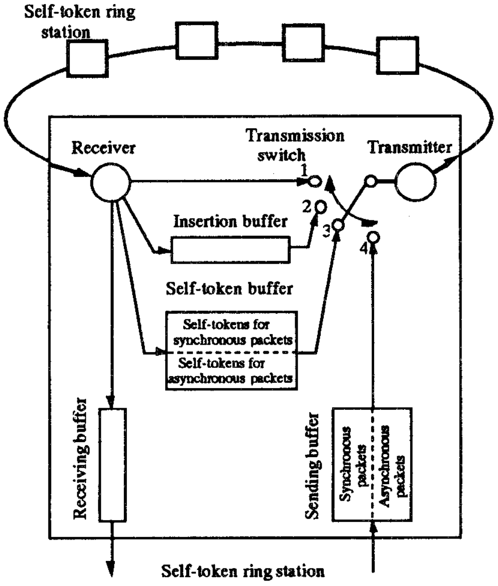

Fig. 1 Structure of station.

Station priority transmission: a station can transmit a packet in the sending buffer immediately if it holds more than one self-token and there is no transit packet just passing through the station. Otherwise, the station must defer to transmit the packet until the transit packet is transmitted. The operation of a station is explained with respect to the switching conditions of the transmission switch in Fig. 1 as follows:

Position 1 : If the insertion buffer is empty, and the sending buffer is empty or self-tokens are busy.
Position 2 : If the receiving buffer is not empty, and the sending buffer is empty or the self-token is busy.
Position 3 : If the self-token is free and the sending buffer is not empty. The switching to position 3 occurs at the end of the transit packet if it exists.
Position 4 : The switching to position 4 just follows the transmission of the self-token in the position 3. Position 3 and 4 make a new packet.

The data of the packet is copied by a destination station during the time it circulates and returns to the source station. Then, the source station removes the packet and stores the self-token in the self-token buffer. While a station is transmitting its own packet, all packets arriving from its upstream station, but not destined to the station itself, are stored in the receiving register. As long as the station is not transmitting its own packet, the content of the receiving register is transmitted to the next station downstream. Note that a station can not transmit a new packet if no self-tokens are in the self-token buffer. Then the dimension of the register is not infinite, but is equal to multiple of the maximum packet length and the number of the self-tokens. The admittance limitation of packets and the finite length of the insertion buffer are quite different from the conventional register insertion protocol. Additionally, fairness is maintained described later.

The ring priority transmission : a station can transmit a packet if it holds more than one self-token, there is no transit packet, and the insertion buffer is empty. The operation of a station is explained with respect to the switching conditions of the transmission switch in Fig. 1 as follows:

Position 1 : If the insertion buffer is empty, and the sending buffer is empty or self-tokens are busy.
Position 2 : If the receiving register is not empty.
Position 3 : If the receiving register is empty, a self-token is free and the sending register is not empty. The switching to position 3 occurs at the end of the transit packet if it exists.
Position 4 : The switching to position 4 just follows the transmission of the self-token in the

position3. Position 3 and 4 make a new packet. Starvation is prevented by introducing the self-tokens.

3. Synchronous and asynchronous transmission

3.1 Implementation of priority transmission control

A priority transmission control mechanism is prepared for synchronous traffic such as voice and video. That is, synchronous traffic is allocated the self-token for station priority, and asynchronous traffic is allocated the self-token for ring priority (see Fig. 1). Then synchronous traffic in a station can use bandwidth in higher priority than asynchronous traffic. Because if its self-tokens for synchronous are free, the station can transmit a packet at least after transmission of a transit packet, but asynchronous traffic can not transmit packets unless its self-tokens are free and the insertion buffer is empty. It means that asynchronous traffic is transmitted on bandwidth which is not occupied by synchronous traffic. However, smoother transmission of synchronous traffic leaves two problems to be solved.

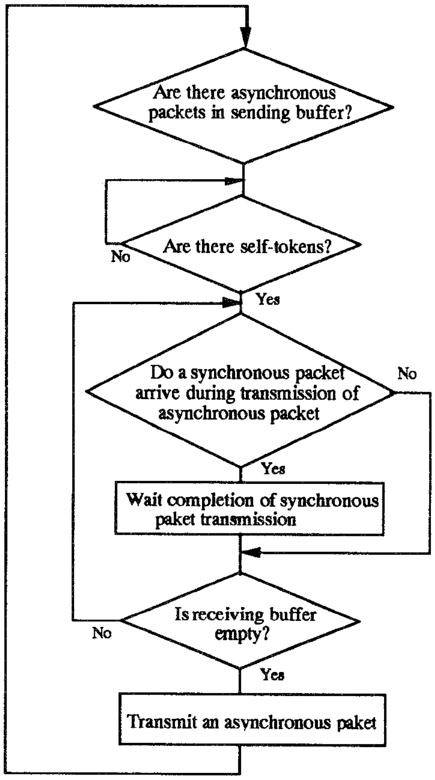

Fig. 2 Algorithm to transmit asynchronous packets

The first problem happens when synchronous traffic arrives at the sending buffer during a asynchronous packet is transmitting, that is, the asynchronous packet disturbs synchronous traffic. Unfortunately, a packet arriving at the insertion buffer during the asynchronous packet is transmitting must be waited until the asynchronous packet and a newly synchronous packet are transmitted. Then, if the packet in the insertion buffer is a synchronous packet, the packet suffers longer delay. To prevent such a situation, if arrival of synchronous traffic during transmission of a asynchronous packet is predictive (the probability of prediction seems to be high because they happen in the identical station), transmission of the asynchronous packet is postponed after transmission of synchronous packet. The transmission algorithm of such asynchronous packets is shown in Fig. 2.

The second problem happens in the insertion buffer while transmitted packets are circulating on a ring. If the insertion buffer operates in the FIFO fashion, synchronous packets can not pass asynchronous ones. If the buffer is constructed a kind of priority queue , in which synchronous packets have higher priority, synchronous packets get lower delay. We will give an example of the priority queue in the next section.

3.2 Guarantee of transfer delay for synchronous traffic

We discuss the arrival rate of synchronous traffic which is transferred to a destination within a guaranteed time. We define some symbols as follows.

N: the number of stations
H: length of header and trailer
L_{symax}: the maximum length of synchronous packets
L_{sy}: average length of synchronous packets
L_{asymax}: the maximum length of asynchronous packets
L_{asy}: average length of asynchronous packets
C: transmission rate of madia
h: delay to decode a destination address of a packet at each station
τ : propagation delay

We assume that each station is allocated one self-token for synchronous, and are allocated plural self-tokens for asynchronous transmission.

Let us consider the maximum token waiting time R_{max} for a synchronous packet underin the station priority control. Then R_{max} is given by

$$R_{max}=1.5N(H+P_{max})/C + Nh + \tau \qquad (1)$$

where $P_{max} = \max(L_{symax}, L_{asymax})$.

The first term of (1) represents the sum of the maximum repeat delay at each station except the source station and the maximum waiting time for transmission in the sending buffer at the source station, the second term is the delay for destination address decode, and the third term is propagation delay, respectively, while a packet circulates a ring.

We explain how to obtain the first term of (1). The maximum repeat delay in the insertion buffer occurs under following conditions: while an asynchronous packet is being transmitted, a repeat packet arrives at the insertion buffer. A synchronous packet is transmitted before the asynchronous packet returns to the source station, and a packet arrives at the insertion buffer during transmission of the synchronous packet. In this situation, the maximum repeat delay at each station is given by 2 P_{max}/C because the insertion buffer is always occupied by two packets. Now we consider the total repeat delay suffered from by a synchronous packet circulating a ring. The worst case is given when other synchronous packets are consecutive and just ahead of the packet. An example is the synchronous packet indicated by (4) (hereafter, for simplicity packet (4)), in Fig. 3(a). The reason why is because we assume a priority queue for the insertion buffer, if an asynchronous packet is ahead of a synchronous packet, the synchronous packet can pass the asynchronous packet as the packet (1) in Fig. 3(a) and (b). However, if a synchronous packet is ahead of another synchronous packet, the latter packet can not pass the former packet. The packet (4) in Fig. 3(a) can not pass the packets (2), (3) and (1) as shown in Fig. 3(a) to (d), but at last it pass the packet 4 in Fig. 3(e). In Fig. 3(f), the packet (4) is removed by the source station. Consequently, in general, the maximum repeat delay D_{max} is given by

$$D_{max}=NP_{max}/C+(N/2-1)P_{max} \tag{2}$$

Next, we consider the maximum waiting time for transmission W_{max}. Then W_{max} is given by

$$W_{max}=P_{max}/C \tag{3}$$

because W_{max} is equal to the maximum time to wait for the completion of repetition in the insertion buffer.

If there exists a self-token when a synchronous traffic arrived at the sending buffer, the average transfer time T_{avg} from a source to a destination is given by

$$\begin{aligned}T_{avg}=&1.5\cdot N/2\cdot(H+\max(L_{sy}, L_{asy}))/2C\\&+(H+L_{sy})/2C+(N-1)h/2+\tau/2\end{aligned} \tag{4}$$

The first term of (4) is the sum of the average repeat time and the average waiting time for transmission in the sending buffer, the second term is the transmission time for a synchronous packet, the third term is address decode time and the fourth term is the propagation delay from the source to destination.

From above, (2) shows that self-tokens for synchronous packets are guaranteed to return back to the source station within R_{max}, even in the worst case. Therefor, from (2) and (4), if the arrival duration of synchronous packets are longer than R_{max}, then they are guaranteed to be delivered to their destination stations with the average transfer time T_{avg}.

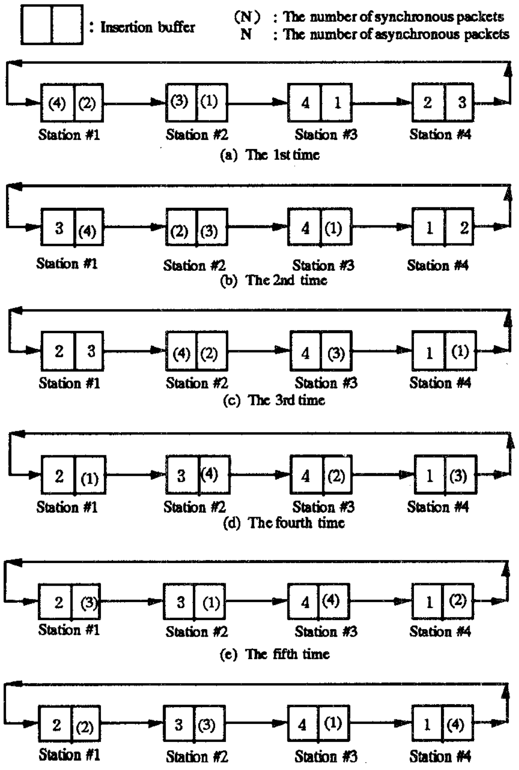

Fig. 3 The maximum repeat delay during synchronous packet transmission

4. Priority control for asynchronous traffic

In previous Section, we described how to assign higher priority to synchronous traffic than asynchronous traffic. But asynchronous traffic in a station, for example a file server, may need higher priority access than ordinary stations. In this Section, a priority transmission control for asynchronous traffic is discussed.

Priority is controlled by the number of self-tokens for asynchronous traffic assigned to stations [12]. The ordinary stations assigned the lowest priority level 0 are allowed to have one self-token. On the other hand, the stations with higher priority level p are allowed to have p+1 self-tokens, so that they can transmit their packets with less transfer delay than the ordinary ones. The protocol provides 8 priority levels.

5. Performance evaluation

We evaluate the proposed protocol via simulation, but first of all, we assume some parameters as follows.

(1)Packet length for synchronous transmission is fixed

(2)Packet length for asynchronous transmission is exponential with the maximum limitation; four times of average packet length

(3)Arrival rate of synchronous traffic is constant

(4)Arrival rate of asynchronous packets is exponential

(5)Destination stations are distributed homogeneously on a ring

(6)All stations are identical, except for asynchronous priority transmission

5.1 Guarantee for transfer delay

We verify the correctness of (1) and (4). We assume as follows: N=32, H=21[bytes], L_{symax} = 1000[bytes], L_{sy} = 250[bytes], L_{asymax} = 250[bytes], L_{asy} = 250[bytes], C = 1000[Mbps], h = 0.12[μs], τ = 50[μs]. Then,

R_{max} = 445.904 [μs]

T_{avg} = 53.960 [μs]

If synchronous duration is equal to or grater than 445.904 [μs], synchronous packets are transmitted within 53.960 [μs].

Fig. 4 shows throughput vs. average transfer delay characteristics with respect to the synchronous duration 300 [μs], 400 [μs], 800 [μs]. For synchronous traffic, the maximum transfer delay is about 39 [μs] when the duration is 400 [μs]. Even if duration is 300 [μs], the transfer delay is about 40 [μs]. These values are less than the estimated value 53.960 [μs]. It is concluded that (2) and (4) give efficient estimation of guarantee of transfer delay for synchronous traffic.

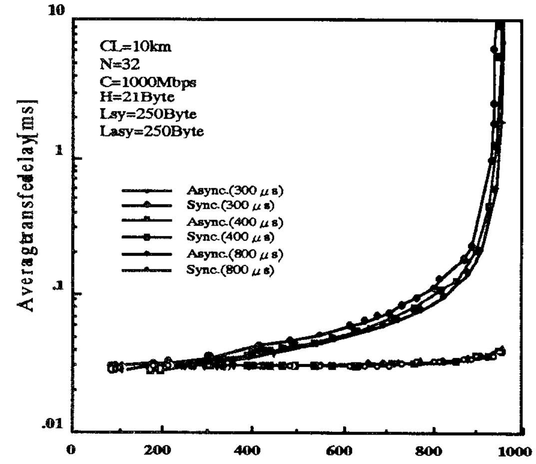

Fig. 4 Throughput vs. mean transfer delay.

5.2 Fairness and priority transmission for asynchronous traffic

Fig. 5 shows throughput vs. average transfer delay characteristics under the condition that both synchronous and asynchronous traffic are transmitted. The characteristics of 8 representative stations which located every four stations are selected. The characteristic of each station is almost equal each other. It means that each station has fair access chance to a ring.

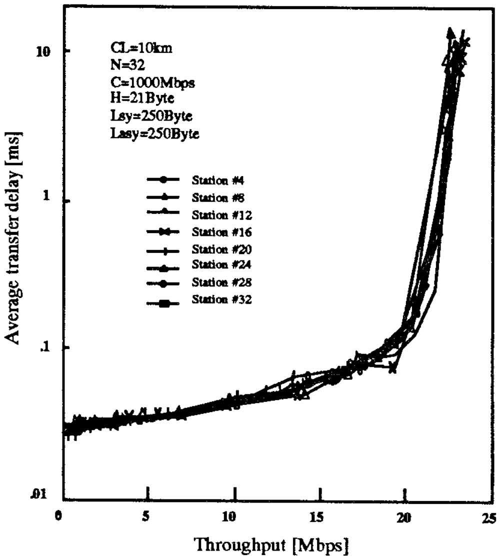

Fig. 5 Fairness for asynchronous packet transmission.

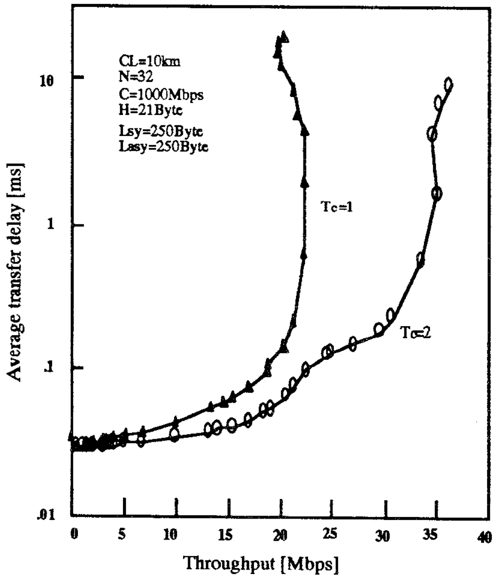

Fig. 6 Priority for asynchronous packet transmission

Fig. 6 explains priority transmission for asynchronous traffic. The 8 high priority stations which locate every four stations have two asynchronous self-token , and the 24 remainder with low priority have just one self-token. As shown in Fig. 6, the high priority stations have lower transfer delay and higher throughput than the low priority ones. Then we can assign priority for asynchronous traffic by controlling the number of self-tokens as explained in Section 4.

6. Conclusions

We have described the Register Insertion/Self-Token (BIST) protocol which support asynchronous and synchronous traffics. The protocol improved to make the conventional buffer insertion get more efficient spatial reuse by adding a new multiple token access control mechanism, called the self-token protocol.

The results which are verified via simulation show that

(1) Synchronous traffic is supported by the station priority access mechanism, and priority insertion buffers. Synchronous traffic is guaranteed to be transmitted to their destinations within the average transfer delay given by (4) if the duration of arrival rate is greater than the maximum token waiting time.

(2) Asynchronous traffic is supported by the ring priority access mechanism. The priority access among asynchronous traffic itself is given by controlling the number of the self-tokens.

(3) Fairness of asynchronous traffic as well as synchronous traffic is maintained among them.

Comparisons of this protocol with the timed token protocol is going on.

References

[1]B. W. Abeysundara and A. E. Kamal, High-Speed Local Area Networks and Their Performance: A Survey," ACM Computing Surveys, Vol. 23, No. 2, Jun. 1991.

[2]R.Jain, "Introduction to FDDI," 15th Conf on Local Computer Networks Tutorial, Oct. 1990.

[3]R. L. Fink and F. E. Ross, "Following the Fiber Distributed Data Interface," IEEE Network, Vol. 6, No. 6, pp.50-55, Mar. 1992.

[4]"Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)", IEEE Standard 802.6, Dec. 1990.

[5]Y. Takitasu, E. Amada, Y. Hatakeyama and Y. Yamaguchi, "High-Speed Multimedia Backbone LAN Architecture Based on ATM Technology", Proc. of IEEE LCN'90, pp.207-212, Mar. 1990.

[6]M.M. Nassehi, "Cyclic-Reservation Multiple-Access Scheme for Gbit/s LANs and MANs Based on Dual-Bus Configuration," IBM Res. Rep. R2, 1969 (#69551), Jun. 1990.

[7]H. T. Kung, "Gigabit Local Area Networks: A System Perspective," IEEE Commun. Mag., Vol. 30, No.4, pp.79-89, Apr. 1992.

[8] C. B.isdikian and A. N. Tantawy, "A Mechanism for Implementing Preemptive Priorities in DQDB Subnetworks," IEEE Trans. Communi., Vol. 42, No. 2/3/4, pp.834-839, Jun. 1994.

[9]I. Cidon and Y. Ofek, "MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse," IEEE Trans. Commni., Vol. 41, No.1, pp.110-120, Jan. 1993.

[10] A. S. Tanenbaum, "Computer Networks, 2nd Edition," Printice-Hall, 1989.

[11] T. Takeda and K. Tanno,"Register Insertion/Self-Token Protocol for High Speed Ring LANs," Proc. of IPCCC'90, pp.60-pp.64, 1990.

[12]D. E. Huber, W. Steinlin and P. J. Wild, "SILK: An Implementation of a Buffer Insertion Ring," IEEE J. Sel. Areas Communi., Vol. SAC-1, No. 5, pp.766-774, Nov. 1983.

[13] W. Bux and M. Schlatter, "An Approximate Method for the Performance Analysis of Buffer Insertion Rings," IEEE Trans. Commun., Vol. COM-31, No.1, pp. 50-55, Jan. 1983.