FDDI-M: A SCHEME TO DOUBLE FDDI'S ABILITY OF SUPPORTING SYNCHRONOUS TRAFFIC

Kang G. Shin

Real-time Computing Laboratory
EECS Department
The University of Michigan
Ann Arbor, Michigan 48109

Qin Zheng

Mitsubishi Electric Research Labs., Inc.
Cambridge Research Center
201 Broadway
Cambridge, MA 02178

Abstract

Synchronous messages are usually generated periodically and each of them is required to be transmitted before the generation of the next message. Due to the inherent deficiency in its Medium Access Control (MAC) protocol, an FDDI token ring can use at most one half of its ring bandwidth to transmit such synchronous traffic. This deficiency greatly reduces the FDDI's capability of supporting multimedia applications like real-time voice/video transmissions. In this paper, we show how a few simple modifications to the FDDI's MAC protocol can remove this deficiency and double a ring's ability of supporting synchronous traffic. The modified protocol, called FDDI-M, preserves all other good features of an FDDI network and can also achieve a higher throughput for asynchronous traffic than the standard FDDI and the FDDI-II, thus making it useful even for those networks without heavy synchronous traffic.

1 Introduction

The Fiber Distributed Data Interface (FDDI) is an ANSI standard for a 100 Mbps token ring network using a fiber-optic medium [1,2]. Due to its high transmission speed, the FDDI alleviates the bandwidth saturation problem of the current 10 Mbps Ethernets and the 4 Mbps or 16 Mbps IBM Token Rings. The synchronous traffic supporting capacity of the FDDI also enables it to support multimedia applications which require the transmission of both ordinary data and digital voice/video.

Using synchronous channels, an FDDI network provides both *bounded transmission delays* and a *guaranteed bandwidth* for synchronous traffic. The transmission delay is controlled by a ring's Target Token Rotation Time (TTRT), which guarantees each node to have a chance to transmit its synchronous messages at least once every 2×TTRT units of time. It also ensures the average time between two consecutive token's visits to a node not to exceed the TTRT [3]. The bandwidth of synchronous traffic that a node i can transmit is guaranteed by assigning the node a portion of the ring's TTRT, called the *high-priority token holding time*, h_i. Specifically, once node i gets the token, it is allowed to transmit its synchronous messages for a time period up to h_i. Since the average token rotation time does not exceed the TTRT, node i has a guaranteed bandwidth of $(h_i/TTRT) \times 100$ Mbps to transmit its synchronous messages.

However, a node usually cannot fully utilize the guaranteed bandwidth due to an inherent deficiency in FDDI's token passing protocol. Synchronous messages are usually generated periodically and each of them is required to be transmitted before the generation of the next one. Suppose the message generation period is T and the time needed to transmit a message is C. Since the maximum token rotation time is 2×TTRT, the TTRT must be set to be no larger than $T/2$ in order to ensure that a node will get the token at least once every T units of time. To give a node enough time to transmit at least one message after getting the token, the high-priority token holding time, h_i, of the node must be set to be no smaller than C. Since the average token rotation time does not exceed $TTRT = T/2$, a node is guaranteed to have an average synchronous

---

*The work reported in this paper was supported in part by the Office of Naval Research under Grant No. N00014-92-J-1080 and the National Science Foundation under Grant No. MIP-9203895. Any opinions, findings, and recommendations expressed in this publication are those of the authors, and do not necessarily reflect the views of the funding agencies.

0-8186-6555-6/94 $04.00 © 1994 IEEE

transmission bandwidth of $2C/T \times 100$ Mbps while it actually uses at most one half of the guaranteed bandwidth, $C/T \times 100$ Mbps, to transmit its synchronous messages. Thus, the difference between the maximum token rotation time and the achievable average token rotation time halves an FDDI token ring's ability of handling synchronous traffic.

To accommodate real–time communication more efficiently, a hybrid protocol, called the FDDI-II, has been proposed by adding time-division circuit switching to the existing packet-switched FDDI token ring [5]. However, the time-division circuit-switched protocol of the FDDI-II reserves a bandwidth exclusively for the transmission of synchronous messages. To ensure the smooth transmission of synchronous messages, the circuit bandwidth must be set to handle the peak signal rate. When a node has a message smaller than the maximum size or does not have a synchronous message to transmit (e.g., during the silent period in a voice conversation), the guaranteed bandwidth is wasted [6].

In this paper, we show that a few simple modifications to the FDDI's MAC protocol can remove the deficiency mentioned above and double a ring's capacity of supporting synchronous traffic. The modified protocol also preserves all the advantages of the original FDDI protocol like the high transmission efficiency (compared to FDDI-II) and fairness in transmitting asynchronous messages at different nodes. More importantly, the proposed modifications are easy to implement and limited to the MAC layer only. No modifications to the upper layer software are required. So, an existing FDDI network can be easily upgraded transparently to the users.

The modified protocol is similar to the token passing protocol proposed in our earlier paper [7]. The purpose of that protocol was to maximize the guaranteed throughput for asynchronous traffic, while the one proposed in this paper is to increase the network's ability of supporting synchronous traffic.

This paper is organized as follows. Section 2 reviews the FDDI's MAC protocol and presents the modifications needed to double the ring's ability of accommodating synchronous traffic. Simulation results are given in Section 3 to verify the claimed advantages of the modified protocol over the FDDI and FDDI-II. The paper concludes with Section 4.

2 A Modified FDDI MAC Protocol

For completeness, we first review the FDDI's MAC protocol. Suppose there are N nodes in a ring which are numbered from 0 to $N-1$. The nodes' access to the ring is then controlled by the following protocol.

Protocol 2.1 (FDDI) .

P1: As part of an FDDI ring initialization process, each node declares a Target Token Rotation Time (TTRT) which equals one half of the requested transmission delay bound of its synchronous messages. The smallest among them is selected as the ring's TTRT. Each node which supports synchronous traffic is then assigned a portion of TTRT to transmit its synchronous messages. Let $h_i \geq 0$ denote the portion of TTRT node i is assigned to transmit its synchronous messages.

P2: Each node has two timers: the token–rotation–timer (TRT) and the token-holding-timer (THT). The TRT always counts up and a node's THT counts up only when the node is transmitting asynchronous packets. If a node's TRT reaches the TTRT before the token arrives at the node, TRT is reset to 0 and the token is marked as "late" by incrementing the node's late count L_c by one. To initialize the timers at different nodes, no packets are allowed to be transmitted during the first token rotation after the ring initialization and L_c's are set to 0.

P3: Only the node which has the token is eligible to transmit packets. The packet transmission time is controlled by the timers, but an in-progress packet transmission will not be interrupted until its completion. When a node i receives the token, it does the following:

- **P3.1:** If $L_c > 0$, set $L_c := L_c - 1$ and $THT :=$ TTRT. Otherwise, $THT := TRT$ and $TRT := 0$.
- **P3.2:** If node 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.
- **P3.3:** If node i has asynchronous packets, it transmits them until the THT counts up to the TTRT or all of its asynchronous packets are transmitted, whichever occurs first.
- **P3.4:** Node i passes the token to the next node $(i+1) \bmod N$.

Let T_{ring} denote a ring's latency which is the time needed to circulate the token around the ring once without transmitting any packet at all, and T_p

the time needed to transmit a maximum-size packet. Then, under the condition $\sum_{i=0}^{N-1} h_i \leq TTRT - T_{ring} - T_p$, the worst-case and average token rotation times will not exceed $2\times TTRT$ and $TTRT$, respectively [3]. As discussed in Section 1, this difference between the worst-case and average token rotation times halves the FDDI's capacity of accommodating synchronous traffic.

The idea of our modifications to the FDDI's MAC protocol is to control the worst-case token rotation time not to exceed the TTRT. If this can be realized, then the TTRT can be set to the message generation period T instead of $T/2$, and twice as many of synchronous connections (or equivalently, synchronous traffic) can be established in an FDDI network.

To realize this, notice that the token rotation time is composed of three components of time: for the transmission of synchronous packets, asynchronous packets, and the ring latency T_{ring}. During one token's rotation, the time used for the transmission of synchronous packets is bounded by $T_S = \sum_{i=0}^{N-1} h_i$ (we assume that h_i is set such that no synchronous packets will be late for transmission; we will call these *packet overruns*). So, if we can control the time used for the transmission of asynchronous packets not to exceed $T_A = TTRT - T_S - T_{ring}$, the maximum token rotation time will never exceed the TTRT. This can be realized by making the following two modifications to the standard FDDI MAC protocol.

M1: Use a modified token rotation time $TTRT_m := TTRT - T_S - T_p$ instead of the TTRT.

M2: Stop the counting of a node's token rotation timer (TRT) when a synchronous packet is being transmitted/forwarded by the node.

With these modifications, the token holding timer (THT) at each node ensures the time used by the node for transmitting asynchronous packets plus the time used for transmitting asynchronous packets during the previous token's rotation not to exceed $TTRT_m + T_p - T_{ring} = TTRT - T_S - T_{ring} = T_A$, as long as $T_S = \sum_{i=0}^{N-1} h_i \leq TTRT - T_{ring}$. Thus the maximum token rotation time will never exceed $T_A + T_S + T_{ring} = TTRT$. It is easy to see that the average token rotation time can reach the TTRT under a heavy load condition. So, the ring's capacity of supporting synchronous traffic is doubled.

The above modifications are easy to implement. We only need to add one AND gate to the timer circuit as shown in Fig. 1 such that the counting of a node's TRT is stopped when the node is transmitting/forwarding a synchronous packet and resumed when the node starts transmitting/forwarding an asynchronous packet or the token. Notice that a packet's type is identified by the first bit of the Frame Control (FC) field in the packet's header [2], and hence, there is no need to change the packet format. The use of the late counter L_c can also be eliminated since the token will never be late if the modified token passing protocol is used. Making the above modifications to the standard FDDI MAC protocol leads to a new protocol, called the *FDDI-M*, as follows.

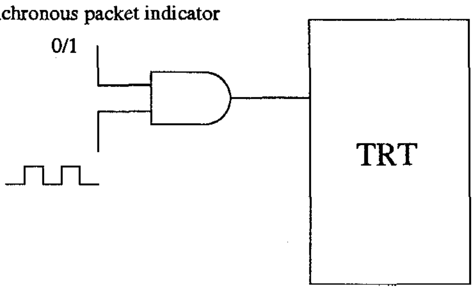

Figure 1: Modification to the timer circuit.

Protocol 2.2 (FDDI-M) .

P1: As part of an FDDI-M ring initialization process, each node declares a TTRT which equals the requested transmission delay bound of its synchronous messages. The smallest among all TTRTs is selected as the ring's TTRT. Each node i which supports synchronous traffic is then assigned a portion, $h_i \geq 0$, of TTRT to transmit its synchronous packets. Update $TTRT := TTRT - \sum_{i=0}^{N-1} h_i - T_p$, where T_p is the time needed to transmit a maximum-size packet.

P2: Each node has two timers: the token-rotation-timer (TRT) and the token-holding-timer (THT). A node's THT counts up when the node is transmitting an asynchronous packet and the TRT counts up when the node is not transmitting/forwarding a synchronous packet. To initialize the timers at different nodes, no packets are allowed to be transmitted during the first token rotation after the ring initialization.

P3: Only the node which has the token is eligible to transmit packets. The packet transmission time is controlled by the timers, but an in-progress packet transmission will not be interrupted. When node i receives the token, it does the following.

P3.1: $THT := TRT$ and $TRT := 0$.

P3.2: If node 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.

P3.3: If node i has asynchronous packets, it transmits them until the THT counts up to the TTRT or all the asynchronous packets are transmitted, whichever occurs first.

P3.4: Node i passes the token to the next node $(i+1) \bmod N$.

With the enhanced capacity of supporting synchronous traffic, a natural question is then what is the FDDI-M's ability of supporting asynchronous traffic? Surprisingly, the modified protocol also improves a ring's ability of supporting asynchronous traffic.

To see this clearly, notice that the 100 Mbps transmission bandwidth of an FDDI ring is always partitioned into three parts: the transmission of synchronous messages, asynchronous messages, and token passing. During one rotation of the token, the time used for token passing is a constant, T_{ring}. Thus, the faster a token rotates, the more ring bandwidth is wasted for passing the token. In other words, for a given synchronous throughput, the larger an average token rotation time that a protocol can achieve, the more asynchronous traffic it can support. Suppose the synchronous message generation period is T, then from the discussion in Section 1, the largest average token rotation time that a standard FDDI protocol can achieve is $TTRT = T/2$. Using the FDDI-M, on the other hand, the nodes always have $TTRT_m - T_{ring} - T_p$ units of time to transmit asynchronous messages. When a ring is heavily loaded with asynchronous traffic, the average token rotation time is at least $TTRT_m = T - T_S - T_p$. Ignoring T_p and since T_S should always be controlled to be $\leq TTRT = T/2$ in an FDDI network, the FDDI-M can always achieve a higher asynchronous traffic throughput than the standard FDDI.

Another salient feature of an FDDI network is its fairness in transmitting asynchronous messages [8]. If a number of nodes have a large amount of asynchronous traffic to transmit, then all nodes will achieve approximately an identical average asynchronous throughput. FDDI-M preserves this feature. This can be seen by observing that for every node with many asynchronous messages to transmit, the average time it uses to transmit asynchronous messages during each token's visit is $TTRT_m$ minus the average time used to transmit asynchronous messages during one token's rotation (ignoring packet overruns). Thus, on average, each node has approximately the same amount of time to transmit its asynchronous messages upon each token's visit.

3 Simulation Results

We carried out extensive simulations to verify the advantages of the FDDI-M over FDDI and FDDI-II. The network simulated is an FDDI single ring of 50 nodes and 92-kilometer ring length. Assuming a node latency of 0.6 microsecond and a propagation delay of 5.085 microseconds per kilometer [2], the ring has a latency $T_{ring} = 0.5$ ms. According to the FDDI standard, the maximum-packet size is 36 Kbits, and thus, the maximum packet transmission time $T_p = 0.36$ ms.

The synchronous traffic under consideration is the digital motion video signals compressed with the MPEG [4]. Each MPEG video channel has an average signal rate of approximately 1.5 Mbps and the video frames are transmitted at the rate of 30 frames/second. So, the frame generation period $T = 33$ ms. Each video frame is required to be transmitted before the generation of the next frame. The MPEG compression algorithm generates a large frame, called a *prime frame*, for every eight frames. The prime-frame size could be three times as large as the average frame size. In our simulation, the prime frames are generated randomly with their sizes uniformly distributed in the range of 100 Kbits ~ 150 Kbits, and other frames are generated with their sizes uniformly distributed in the range of 25 Kbits ~ 75 Kbits.

Asynchronous messages are generated randomly with their sizes uniformly distributed in the range of 0 ~ 1.5 Mbits by those 10 nodes which are not transmitting video signals. To test a ring's ability of accommodating asynchronous traffic, the asynchronous traffic is generated with a total rate equal to the ring's bandwidth of 100 Mbps. So, the actual throughput represents a ring's ability of accommodating asynchronous traffic. The fairness of a protocol can also be verified by checking each node's asynchronous traffic throughput.

Three protocols, FDDI, FDDI-II, and FDDI-M, are compared for their ability to support video channels and asynchronous traffic. The configurations of these protocols are described below.

FDDI: We simulated two configurations: (1) $TTRT = T/2 = 16.5$ ms, and (2) $TTRT = T = 33$ ms of the FDDI. The reason to use the second configuration is to check if it is really necessary to set the TTRT to one half of the requested delay bound. For both configurations, the high-priority token holding time of a node that generates video signals is set to the time needed to transmit a maximum-size frame h_i = 150 Kbits / 100 Mbps = 1.5 ms. Theoretically, no more than $n = (TTRT - T_{ring} - T_p)/h_i$ video channels should be established to ensure $\sum_{i=0}^{n-1} h_i \leq TTRT - T_{ring} - T_p$ ($n = 10$ for $TTRT = 16.5$ ms, and $n = 21$ for $TTRT = 33$ ms). In our simulation, when more than n video channels are requested, we ignore this constraint to see how many frames will not be transmitted before the generation of the next frame.

FDDI-II: In order to ensure each video frame to be transmitted before the generation of next frame, we assign a transmission bandwidth of 150 Kbits / 33 ms = 4.5 Mbps to each video channel. Thus, 98/4.5 = 21 video channels can be established. When $n > 21$ video channels are requested, we divide the 98 Mbps bandwidth equally among the channels. In this case, some frames may need longer than 33 milliseconds for their transmission.

FDDI-M: The configurations of FDDI-M are the same as those of FDDI except that the ring's TTRT is set to $33 - 1.5n$ ms, where n is the number of video channels established. Since the TTRT should not be smaller than $T_{ring} = 0.5$ millisecond, up to $n = (33 - 0.5)/1.5 = 21$ video channels can be established with the FDDI-M. When $n > 21$ video channels are requested, we set $TTRT := 2$ ms to guarantee a certain level of asynchronous throughput.

In our simulation, we use node $0 \cdots$ node $(n-1)$ as the source nodes of the n requesting video channels. Asynchronous messages are generated by node $35 \cdots$ node 44. For each $n = 0, 1, \cdots, 30$, we used a simulation period of 100 seconds during which each video channel generates 3300 frames. To simulate the worst case, all video channels start transmitting frames at time 0. For each n, the percentages of channel $(n-1)$'s prime frames missing their deadlines under different protocols are plotted in Fig. 2. We considered the prime frames only since they are more important than other frames (decompression of other frames needs information from the prime frames). Also, the prime frames are more likely to miss their deadlines because of their large size.

From Fig. 2, up to 13 video channels can be established in an FDDI ring with $TTRT = 16.5$ ms without any late frames. Setting up any more channels would cause the loss of prime frames. Setting $TTRT := 33$ ms for the FDDI does not work at all. The prime frames missed their deadlines even in case of a small number of video channels. This means that the token rotation time could exceed the TTRT even if the asynchronous traffic is light. Thus it is necessary to set the TTRT to one half of the requested delay bound in an FDDI network.

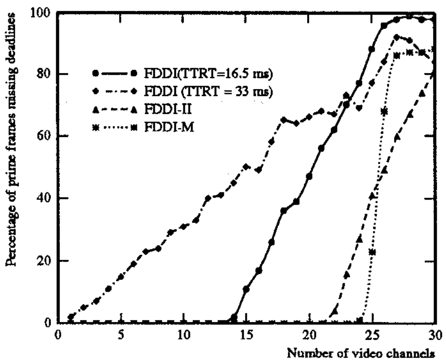

Figure 2: Percentages of prime frames missed deadline.

As expected, the FDDI-M outperforms the FDDI by far. It can accommodate 24 video channels which is 11 channels more than the FDDI ($TTRT = 16.5$ ms). The FDDI-II behaves exactly as calculated. Up to 21 channels can be established. The reason why the FDDI-M performs a little better than the FDDI-II is that the former shares the ring bandwidth among different video channels, while with the latter, one channel cannot use the reserved bandwidth of another, even when that channel is not transmitting anything.

The throughput of asynchronous messages are plotted in Fig. 3. The FDDI-M has a higher asynchronous throughput when the number of channels is less than, or equal to, 13. This can be seen more clearly in Fig. 4 which shows the average token rotation times. As discussed earlier, the smaller the average token rotation time, the more transmission bandwidth is used for the token passing, thus leaving less bandwidth for asynchronous messages. The difference between asynchronous throughput becomes more pronounced with the increase of ring latency (i.e., increase of ring size).

When $n \geq 14$ video channels are established, the FDDI has a higher asynchronous throughput than the FDDI-M. However, this does *not* mean that the FDDI is superior to the FDDI-M. The FDDI's gain in asynchronous throughput is achieved at the cost

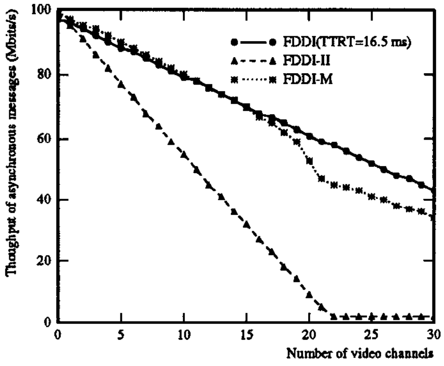

Figure 3: Throughput of asynchronous traffic.

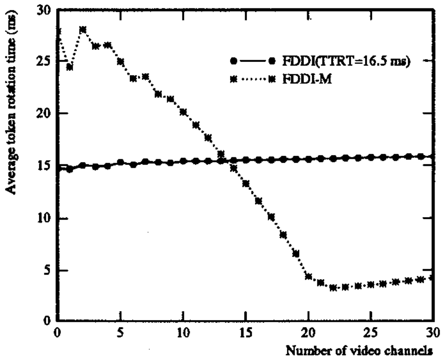

Figure 4: Average token rotation times.

of missing synchronous messages' deadlines. This is something similar to setting a very large TTRT to have the maximum bandwidth without considering synchronous messages at all. The FDDI-M reduces the average token rotation time to accommodate more video channels. When the number of video channels reaches 21, the FDDI-M's TTRT is fixed at 2 ms. Thus, the average token rotation time no longer decreases, and synchronous messages start missing their deadlines.

The FDDI-II provides the lowest throughput for asynchronous messages. Establishment of each video channel deprives 4.5 Mbps bandwidth from the network. So, after establishing 22 video channels, less than 2 Mbps transmission bandwidth is left for asynchronous traffic. This indicates one of the disadvantages of the FDDI-II when a network is heavily loaded with synchronous traffic.

4 Conclusions

We have proposed in this paper a modified MAC protocol of the FDDI, called FDDI-M, which can significantly improve an FDDI network's ability of supporting synchronous and asynchronous traffic. Our simulation results have shown that an FDDI-M ring can accommodate nearly twice as many MPEG video channels as a standard FDDI ring. Due to the simplicity of the proposed modifications and the increasing networking demands for multimedia applications, the FDDI-M has great potential use for these applications.

References

[1] Floyd E. Ross, "An overview of FDDI: The fiber distributed data interface", *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 1043 - 1051, September 1989.

[2] Alfred C. Weaver, "Local area networks and busses - an analysis", Technical report, Flight Data Systems, NASA–Johnson Space Center, 1986.

[3] K. C. Sevcik and M. J. Johnson, "Cycle time properties of the FDDI token ring protocol", *Technical Report CSRI-179, Computer Science Research Institute, University of Toronto*, 1986.

[4] D. Gall, "MPEG: a video compression standard for multimedia applications", *Communications of the ACM*, pp. 46–58, 1991.

[5] M. Teener and R. Gvozdanovic, "FDDI-II operation and architectures", *in proceedings of the 14th conference on local computer networks*, pp. 49–61, 1989.

[6] Michael Liebhold and Eric M. Hoffert, "Toward an open enviroment for digital video", *Communication of ACM*, vol. 34, pp. 104 - 112, April 1991.

[7] Kang G. Shin and Qin Zheng, "Mixed time-constrained and non-time-constrained communications in local area networks", *IEEE Trans. Communications*, pp. 1668–1676, November 1993.

[8] M. J. Johnson, "Fairness of channel access for non–time–critical traffic using the FDDI token ring protocol", *Proc. seminar real–time local area networks*, pp. 145–157, 1986.