NCC2-387
161942
P-21

Analysis of FDDI Synchronous Traffic Delays

Marjory J. Johnson

AMES GRANT
IN-61-CR

January, 1988

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS TR 88.3

(NASA-CR-183223) ANALYSIS OF FDDI SYNCHRONOUS TRAFFIC DELAYS (NASA) 21 p CSCL 09B N89-14695

Unclas
G3/60 0161942

Research Institute for Advanced Computer Science

Analysis of FDDI Synchronous Traffic Delays

Marjory J. Johnson

January, 1988

Research Institute for Advanced Computer Science
NASA Ames Research Center

The Fiber Distributed Data Interface (FDDI) high-speed token-ring protocol provides support for two classes of service: synchronous service, to support applications which require deterministic access to the channel, and asynchronous service, to support applications which do not have such stringent response-time requirements. The purpose of this paper is to determine how to set ring parameters to support synchronous traffic most efficiently. Both theoretical results and results obtained from a simulation study are presented.

This work was supported by the National Aeronautics and Space Administration under NASA Contract NAS2-387 to the Universities Space Research Association (USRA).

NCC2-387

Analysis of FDDI Synchronous Traffic Delays

Marjory J. Johnson
Research Institute for Advanced Computer Science
NASA Ames Research Center
Moffett Field, California 94035

1. Introduction

The Fiber Distributed Data Interface (FDDI) is an emerging ANSI standard for a 100 megabit-per-second fiber-optic token ring. FDDI promises to have a substantial impact on networking products and services of the future. A unique feature of FDDI is its ability to support applications, such as packet voice or real-time control, which require deterministic access to the communications channel, but which can tolerate some jitter. FDDI provides two classes of service: synchronous service, to support the types of applications described above, and asynchronous service, to support applications which do not have such stringent channel-access requirements.

NASA is studying FDDI's suitability for use on the Space Station. A possible Space Station application for which the synchronous service class might be appropriate is the transmission of periodic samples from an instrument or laboratory experiment on board the Space Station. Samples would need to be transmitted regularly, but a reasonable amount of jitter would be tolerable. Based on samples of this type, a scientist could remotely control his experiment in real time. The study reported herein was motivated by the need to determine how to set ring parameters to support this type of traffic most efficiently. To date no studies that address this issue have been reported in the literature.

FDDI is able to provide synchronous service only because token-rotation time is bounded. This bound is a function of a ring parameter, called T_Opr, which specifies the expected token-rotation time. Each node requires a specified frequency of access to the channel to support its synchronous traffic. T_Opr is negotiated by the nodes during ring initialization to ensure that the most stringent synchronous channel-access requirements of all the nodes will be satisfied. It can be proved that the maximum token-rotation time for any ring is $2 \times T\_Opr$ [3,5]. This would imply that during the negotiation process for T_Opr, each node should request a value that is half the token-rotation time it requires to support its synchronous needs. However, it can also be proved that average token-rotation time is less than or equal to T_Opr [5]. Since ring operation is more efficient for larger values of T_Opr, it is desirable, for each particular ring configuration, to determine the largest value that can be assigned to T_Opr such that the desired frequency of channel access can be guaranteed.

In this paper we discuss factors which influence token-rotation time and hence which influence channel-access time for synchronous traffic. We derive a formula for the maximum token-rotation time for a particular ring configuration, based on the total percentage of ring bandwidth allocated for synchronous traffic. Then we derive a formula for an optimal value for T_Opr, based on this maximum token-rotation time, which guarantees the desired frequency of channel access for the nodes on that ring. In practice, even the value for

T_Opr prescribed by this formula is more restrictive than necessary. Results from a simulation study suggest that under some relatively non-restrictive assumptions regarding the regularity of synchronous traffic, setting T_Opr equal to the desired maximum token-rotation time can provide satisfactory performance.

2. FDDI Access Protocol

In this section we present a brief description of the FDDI media-access-control protocol. For a more detailed discussion see [4].

FDDI is a timed-token-rotation protocol; timers within each node cooperatively attempt to maintain a specified token-rotation time by using the observed network load to regulate the amount of time that an individual node may transmit. Each node has a Token-Rotation Timer (TRT) and a Token-Holding Timer (THT), which control that node's access to the network. A node's TRT provides a mechanism of accounting for the amount of time that has passed since that node last received the token. The THT assures that initiation of transmission of an asynchronous frame is only allowed if the immediately preceding cycle (i.e., rotation) of the token was less than T_Opr .

Each node is assigned an amount of time, called its *synchronous bandwidth allocation*, for synchronous transmission each time it receives the token. The total of all synchronous assignments is not to exceed 100 percent of T_Opr . Since the average token-rotation time is less than or equal to T_Opr [5], then synchronous bandwidth assignments are actually a percentage of the total bandwidth of the ring. Whereas a node may transmit synchronous frames for its allotted time whenever it receives the token, asynchronous transmission is allowed only if the load on the ring is light enough to support it. All bandwidth that is not used for synchronous transmission is available for asynchronous transmission. Since a node need not use its entire synchronous bandwidth allocation each time it receives the token, the amount of bandwidth available for asynchronous transmission actually varies from cycle to cycle.

3. Theoretical Results

We wish to select an optimal value for *T_Opr* that will guarantee the desired frequency of channel access for a particular ring configuration.* Clearly this optimal value for *T_Opr* must be a function of maximum token-rotation time, which in turn is a function of the maximum time that can be spent transmitting both synchronous and asynchronous frames. For simplicity, we assume that overhead is negligible, and so we disregard it in the derivation of our theoretical results.

3.1. Token-Cycle Length

Average cycle time is less than or equal to *T_Opr* [5]. If the token does not return to a particular node within a *T_Opr* time period, then we say that the cycle is a *long cycle*. There are two possible causes of long cycles: asynchronous overrun and irregular synchronous bandwidth usage. Each of these phenomena is discussed below.

---

*Note that we are interested in channel-access delays only. Translating synchronous delay requirements at the application level to channel-access requirements at the media-access-control layer is beyond the scope of this paper.

Asynchronous overrun occurs when a node's THT expires during transmission of an asynchronous frame. According to the FDDI media-access-control protocol [1], transmission of this frame will be completed before the token is forwarded to the downstream node. The upper bound on asynchronous overrun by a single node is the time it takes to transmit a frame of maximum size, denoted by $Frame\_time$, since the THT might expire right after transmission of the frame begins. From [3] this overrun detracts from the amount of time available for asynchronous transmission by downstream nodes during the remainder of a particular token rotation. Hence, $Frame\_time$ is the maximum amount of time that can be attributable to asynchronous overrun during a single token cycle. Since transmission of asynchronous frames may be initiated for a T_Opr time period, the maximum amount of time that asynchronous frames may be transmitted during one complete token rotation is $T\_Opr + Frame\_time$. Hence, asynchronous overrun may make a cycle long by $Frame\_time$ amount of time.

Now consider the effect of utilization of synchronous bandwidth allocation on cycle length. Suppose synchronous bandwidth allocation is $Sync$ time units. As we noted earlier, the amount of bandwidth available for asynchronous transmission varies from cycle to cycle, depending on the amount of synchronous transmission that occurred in the preceding cycle. If each node uses its full synchronous allocation during each token cycle, then the amount of bandwidth available for asynchronous transmission is effectively $T\_Opr - Sync$ [2]. However, if some portion, say B time units, of the synchronous bandwidth is not used during a given token rotation, then the amount of bandwidth available for asynchronous transmission increases accordingly. In addition all nodes may still use their full synchronous bandwidth allocation the next time they receive the token. Depending on the particular transmission sequence, such a situation may result in a token cycle that is long by up to B time units, with asynchronous frames being transmitted for up to $T\_Opr - Sync + B$ time units and synchronous frames being transmitted for $Sync$ time units.

This phenomenon can occur only if one or more nodes uses its synchronous bandwidth allocation irregularly. For example, suppose we have a ring with four nodes, 1, 2, 3, and 4. Suppose nodes 1, 2, and 4 transmit only asynchronous frames and that node 3 transmits only synchronous frames. Suppose node 3's synchronous bandwidth allocation is B time units. Consider the following scenario. Suppose during a particular token cycle beginning with node 1 that node 3 uses none of its synchronous bandwidth allocation. During the token cycle beginning with node 1's next turn, suppose that nodes 1 and 2 transmit asynchronous frames for a period of time $\geqslant T\_Opr$ and that node 3 uses its full synchronous bandwidth allocation during this cycle. Then this latter token cycle is long by at least these B time units.

Theorem 1: The upper bound on token-cycle time is $T\_Opr + Frame\_time + Sync$, where the total synchronous bandwidth allocation for the ring is $Sync$ time units.

Proof: The effects of the two phenomena of asynchronous overrun and irregular synchronous bandwidth usage are additive. For suppose that no synchronous frames are transmitted during a particular token rotation beginning with node

x. Then during the token rotation beginning with node x's next turn, it is possible for the maximum asynchronous transmission, $T\_Opr + Frame\_time$, to occur before the token reaches any of the nodes having a nonzero synchronous bandwidth allocation. If all the nodes with a nonzero synchronous bandwidth allocation use their total allocation when they receive the token, then the length of this cycle will be $T\_Opr + Frame\_time + Sync$.

3.2. Computation of T_Opr

As a direct consequence of Theorem 1, we can compute an optimal (i.e., maximal) value for T_Opr that guarantees the desired frequency of channel access.

Theorem 2: The optimal setting for T_Opr that guarantees the desired frequency of channel access, say X time units, for a particular ring configuration is $T\_Opr = X - Frame\_time - Sync$.

Proof: By Theorem 1, if $T\_Opr = X - Frame\_time - Sync$, then the maximum token-rotation time is X. Thus, the token is guaranteed to return to a given node within X time units.

If $Sync$ and $Frame\_time$ are both relatively small, then Theorem 2 provides a more efficient setting of T_Opr than that specified by the FDDI standards documents. For example, suppose $X = 100$ time units, $Sync = 20$ time units, and $Frame\_time = 10$ time units. According to Theorem 2, the maximal setting for T_Opr to guarantee the desired frequency of channel access is $X - Frame\_time - Sync = 70$, whereas the FDDI standards documents suggest that T_Opr should be set to $X/2 = 50$. Note that if $X - Frame\_time - Sync < Sync$, then it is impossible to assign a value to T_Opr that will guarantee the desired frequency of channel access, for there is too much demand for synchronous bandwidth relative to the channel-access requirements. Setting T_Opr so as to guarantee channel access within the desired time interval would result in insufficient channel capacity to support the amount of synchronous traffic.

4. Simulation Results

Irregular synchronous bandwidth usage is a major cause of long token cycles; if synchronous traffic is generated at constant intervals, as in our proposed Space Station application, then token-cycle time will seldom reach the theoretical upper bound. We wish to determine a more efficient setting of T_Opr for this type of situation than that provided by Theorem 2. It is clear that assigning larger values to T_Opr would increase efficiency of the ring, because overhead would be reduced. A natural value to consider for T_Opr is the desired frequency of channel access. In fact, this is the upper bound of possible values you would want to assign to T_Opr, for if T_Opr were greater than the desired frequency of channel access, then the average cycle time could also be greater. As a result, the channel-access time for synchronous traffic might often exceed the desired upper bound.

We conducted a simulation study of the delays experienced by synchronous frames when T_Opr is set equal to the desired frequency of channel access, to determine whether this configuration would provide satisfactory support (both in terms of mean delay and in terms of number of delays that exceed *T_Opr*) for synchronous traffic.* The delay measured in the simulation is the time from generation of a frame at the source node to receipt of the frame at the destination node. This includes queueing delay while the frame waits in the transmission queue at the source, transmission time, and the time required for the frame to propagate from the source node to the destination node. While there is not a direct correlation between synchronous delay and the time between consecutive channel accesses, as long as synchronous delay does not exceed *T_Opr* , then the ring is performing as desired.

We modeled a ring with sixteen nodes, as illustrated in figure 1. Nodes 1-3, 6-11, and 14-16 are synchronous nodes, i.e., they transmit only synchronous frames, and nodes 4, 5, 12, and 13 are asynchronous nodes, i.e., they transmit only asynchronous frames. We modeled the network under several different traffic loads by varying the number of active synchronous nodes (i.e., those nodes actually generating and transmitting synchronous frames during the run) and by varying the interarrival rate of asynchronous frames at the asynchronous nodes. Asynchronous traffic is evenly distributed among the four asynchronous nodes for each simulation run, and the interarrival times of the asynchronous frames at each of the nodes are exponentially distributed. Each active synchronous node generates synchronous frames at a constant rate of one frame every

*This study was conducted using LANES (Local Area Network Extensible Simulator), which was developed by the Data Networks Group at NASA Ames Research Center, under the direction of Terry Grant. William Kessinger, an Industrial Engineering student at Stanford University, made the simulation runs and produced all the figures presented herein while he was working as a summer employee at RIACS.

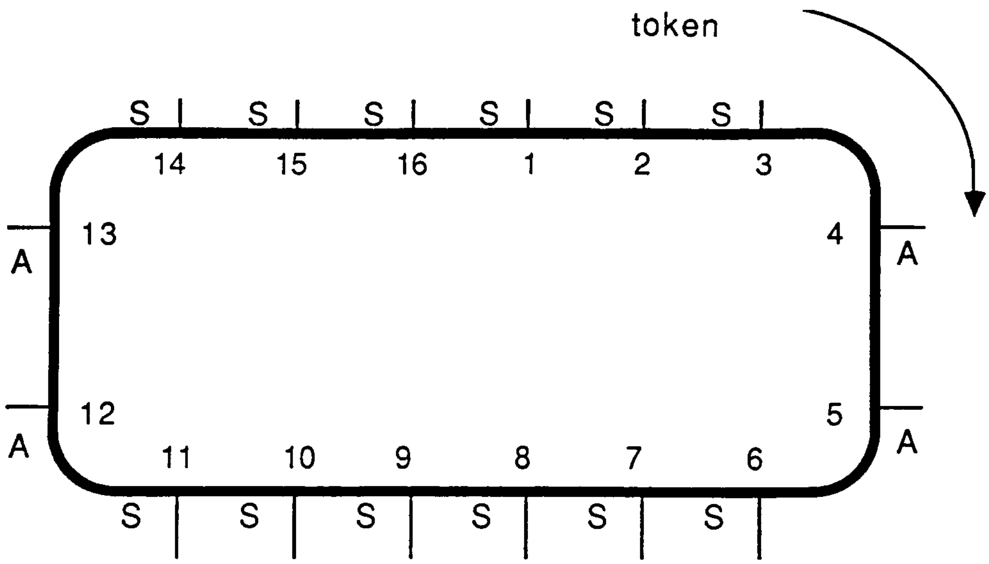

Figure 1. Ring Configuration

8000 microseconds. While the interval between any two consecutive synchronous frames generated at any individual node is constant, the generation of synchronous frames at different synchronous nodes is staggered. For each synchronous node, it is desired that any given frame at that node should be transmitted before the next one at that same node is queued for transmission, i.e., the desired frequency of channel access is 8000 microseconds. Since the purpose of our study is to examine synchronous delays when *T_Opr* is set equal to the desired frequency of channel access, then *T_Opr* is set to 8000 microseconds also.

Frame size for both synchronous and asynchronous frames is 4040 bytes, including the header. Hence, approximately 24 frames can be transmitted during each 8000 microsecond *T_Opr* time period. Each active synchronous node is allocated synchronous bandwidth to transmit exactly one synchronous frame each time it receives the token.

We modeled the network under three different levels of loading, called levels 1, 2, and 3. For level-1 loading, only 6 synchronous nodes are active and the average number of asynchronous frames generated per *T_Opr* time period is 12. Thus, the offered load is approximately 75% of ring capacity. Under level-2 and level-3 loading, all 12 of the synchronous nodes generate one frame per *T_Opr* time period. That is, synchronous traffic accounts for half the capacity of the ring. The average number of asynchronous frames generated per *T_Opr* time period for level-2 and level-3 loading are 12 and 16 frames, respectively. Under level-1 loading, the bulk of the traffic on the network is asynchronous; under level-2 loading, the offered load is approximately equal to the capacity of the ring; under level-3 loading, the ring is overloaded, so that queues at the asynchronous nodes will become infinite.

Under level-1 loading, even though ring utilization is approximately 75%, both mean and maximum synchronous delays are considerably less than *T_Opr* . Figure 2 presents a histogram of the delays experienced by a representative node during a single simulation run with level-1 loading.

Under level-2 loading, the mean delays experienced by the synchronous nodes are all less than 3000 microseconds. Figure 3 presents the 95% confidence intervals for these mean delays. Figures 4.a and 4.b are histograms of delays experienced by representative nodes during a single simulation run with level-2 loading. For this particular run, fewer than 1% of all synchronous frames experience a delay that exceeds 8000 microseconds.

Under level-3 loading, the mean delay experienced by each synchronous

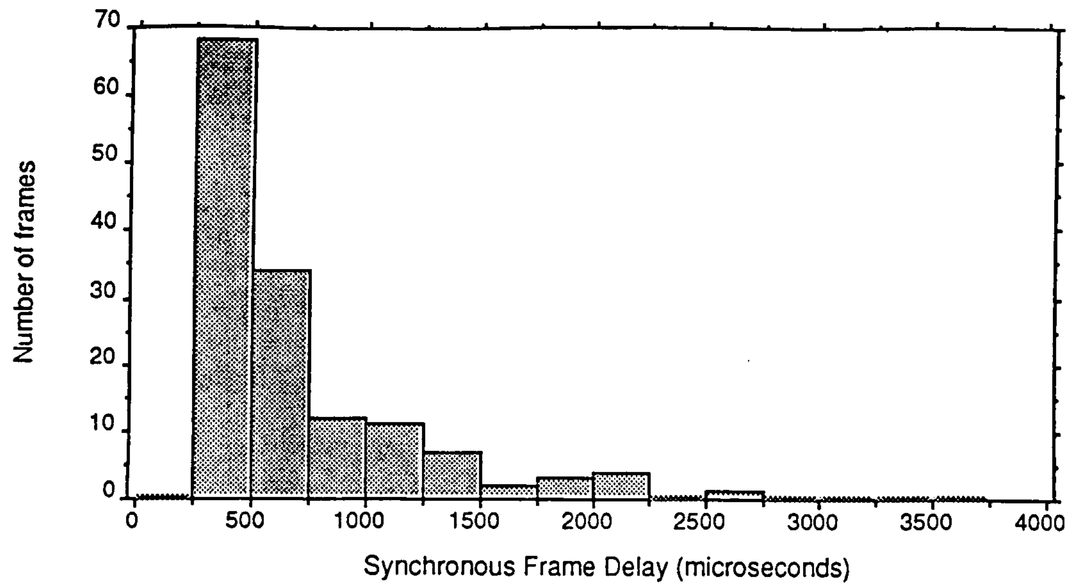

Figure 2. Frequency Distribution of Synchronous Frame Delays for Node 15 under Level-1 Loading (72.6% utilization)

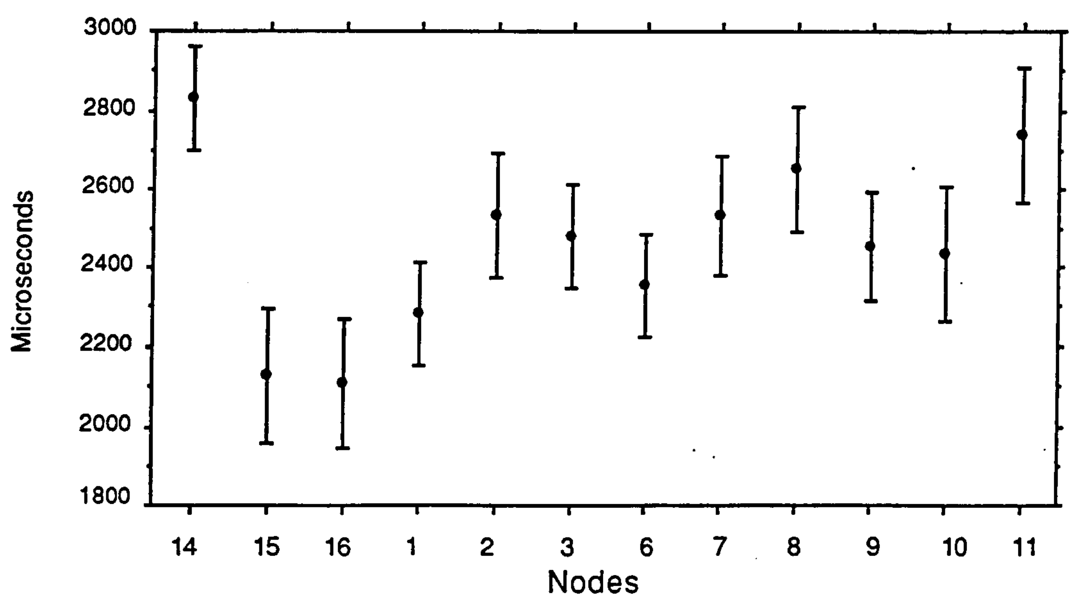

Figure 3. 95% Confidence Intervals for Mean Delays under Level-2 Loading (approx. 96% utilization)

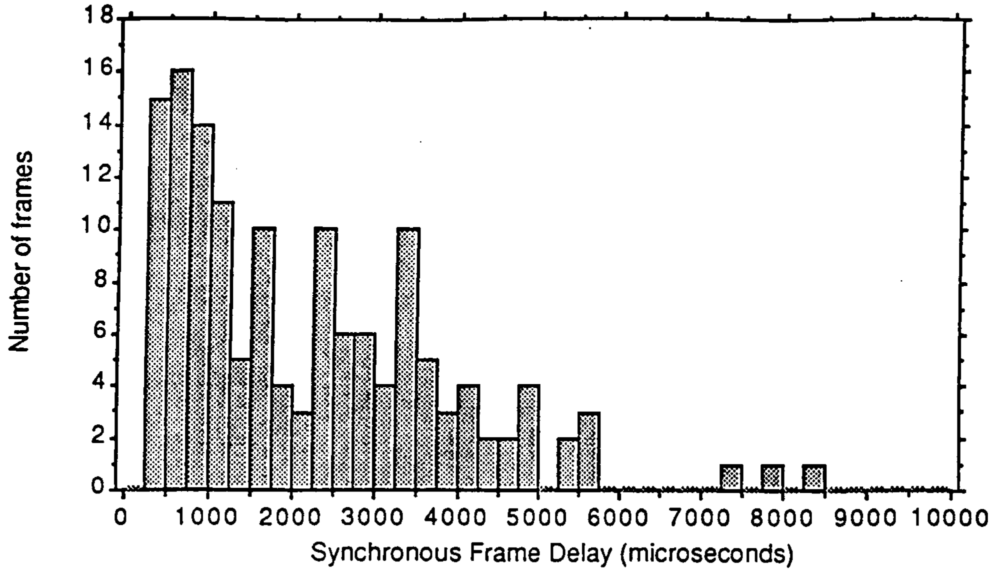

Figure 4.a. Frequency Distribution of Synchronous Frame Delays for Node 9 under Level-2 Loading (96.8% utilization)

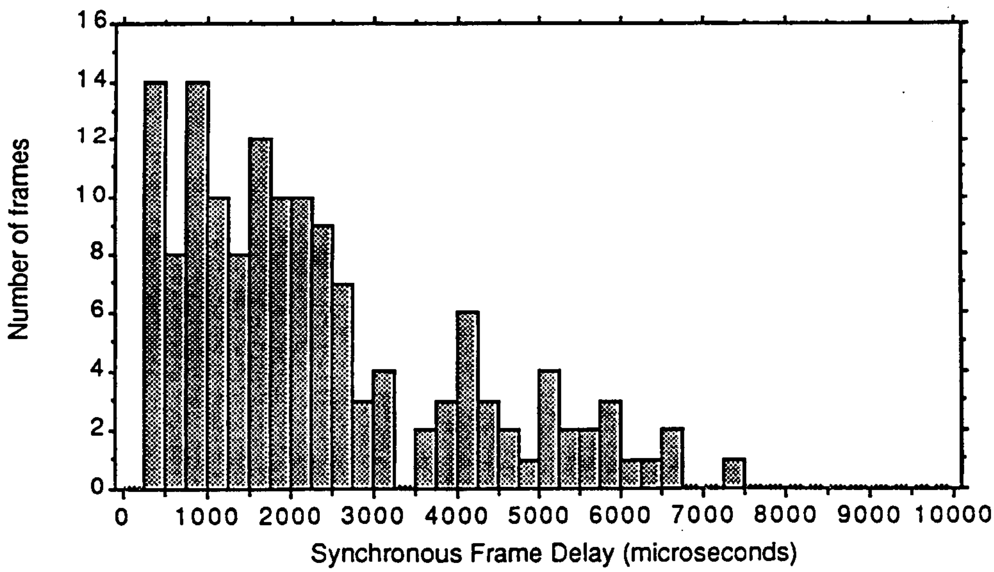

Figure 4.b. Frequency Distribution of Synchronous Frame Delays for Node 14 under Level-2 Loading (96.8% utilization)

node is still much less than 8000 microseconds. Figure 5 presents the 95% confidence intervals for the mean delays experienced by each of the synchronous nodes. Note that these mean delays are larger than the mean delays under level-2 loading, as we would expect, since the average token cycle will be longer in the more heavily loaded situation. Figures 6.a, 6.b, and 6.c are histograms of synchronous delays experienced by representative nodes during a single simulation run with level-3 loading.

For this particular run, 80 out of a total of 1718 synchronous frames (less than 5%) experience delays that exceed 8000 microseconds. The most instances of excessive delay occurred for node 2, with 16 of 144 frames, or 11%, experiencing delays greater then 8000 microseconds. Examination of the individual synchronous delays for this node reveals that these 16 excessive delays occur in 6 clusters of consecutive frames, with some clusters containing as many as 4 frames. This clustering phenomenon occurs because the token rotates relatively slowly in such a heavily overloaded ring. (The average token-cycle time is greater than 7000 microseconds for this run.) When one synchronous frame experiences a delay greater than 8000 microseconds, then a second synchronous frame will become queued for transmission before the first one is transmitted. Since this second frame must wait for two token arrivals before it will be transmitted and because the cycle time is so large, this second frame may also experience an excessive (i.e., > 8000 microseconds) delay. This pattern may continue for several consecutive frames, even though the token visits the node with acceptable frequency.

For this reason, it might be beneficial to institute a procedure to purge a

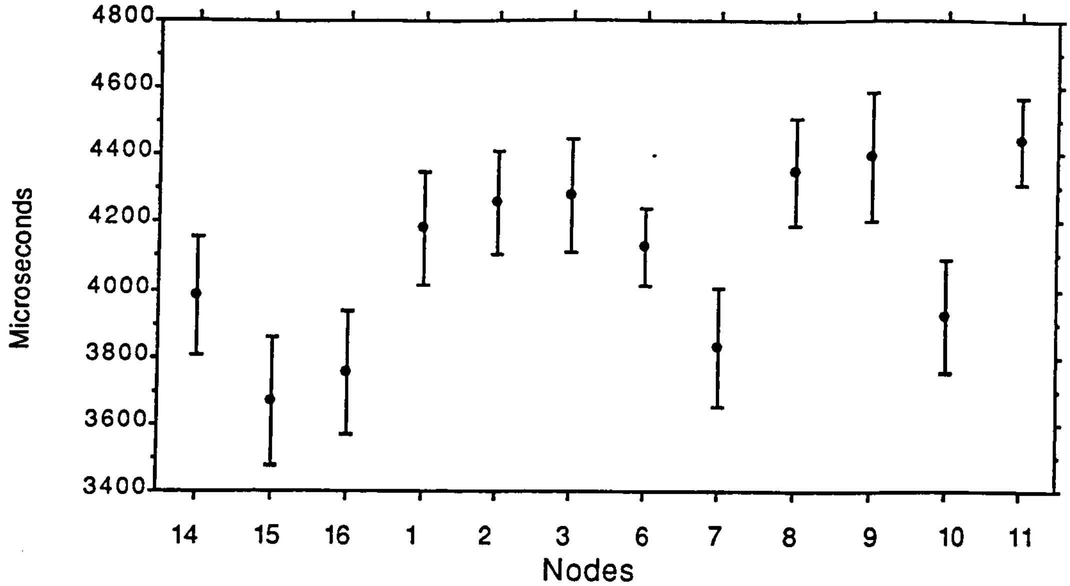

Figure 5. 95% Confidence Intervals for Mean Delays under Level-3 Loading (approx. 99% utilization)

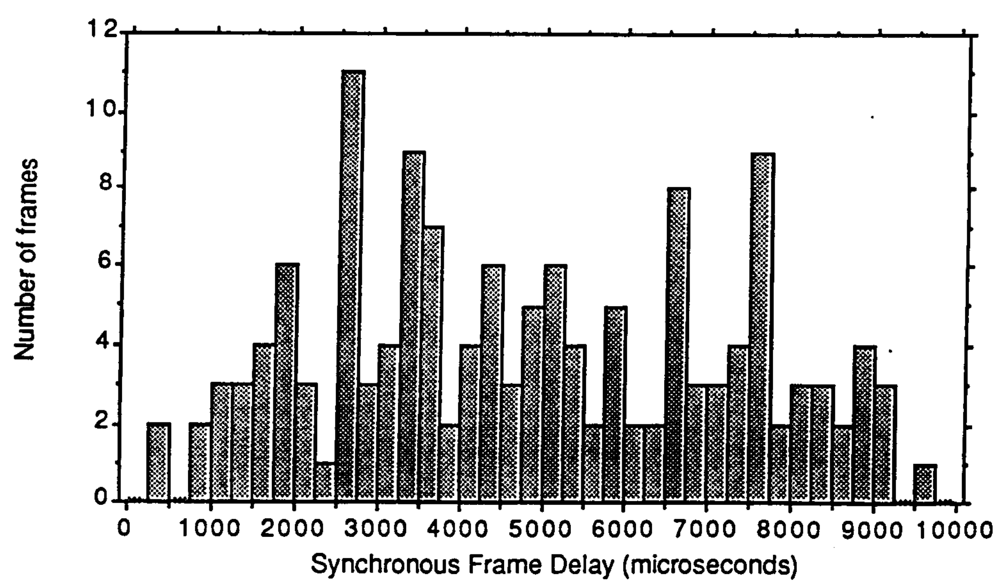

Figure 6.a. Frequency Distribution of Synchronous Frame Delays for Node 2 under Level-3 Loading (99.6% utilization)

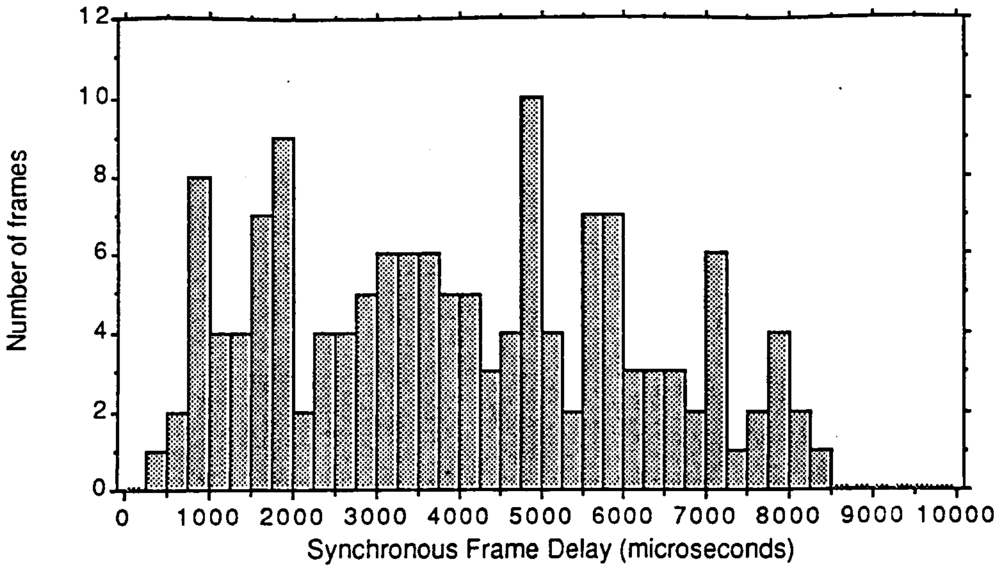

Figure 6.b. Frequency Distribution of Synchronous Frame Delays for Node 6 under Level-3 Loading (99.6% utilization)

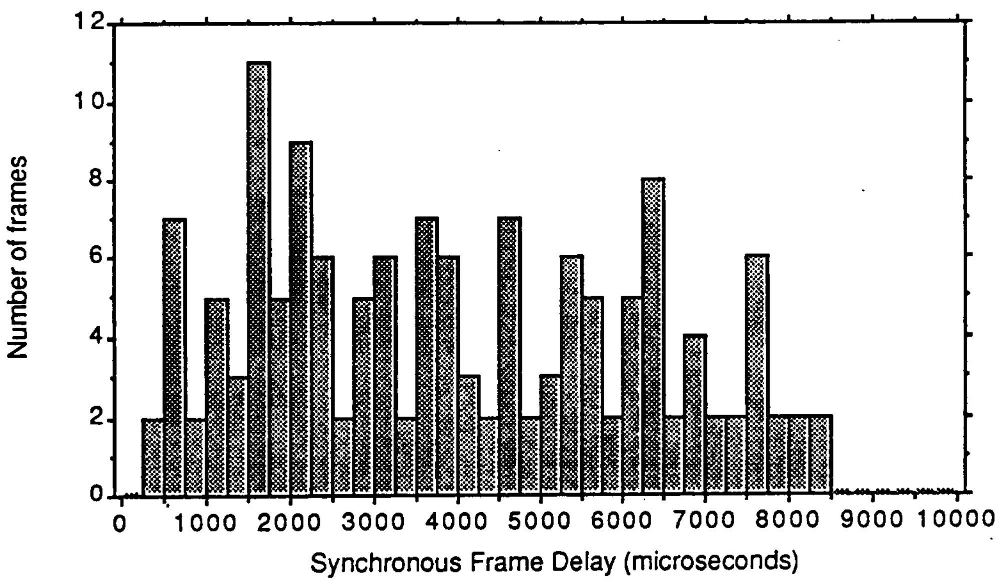

Figure 6.c. Frequency Distribution of Synchronous Frame Delays for Node 15 under Level-3 Loading (99.6% utilization)

synchronous frame which is pending transmission whenever a new synchronous frame becomes queued for transmission at the same node. This would result in an occasional lost frame, but it would prevent a series of consecutive frames from being delivered outside of the time constraints. If this type of purging had been used for the simulation run described above, then only 6 synchronous frames from node 2 (less than 5%) would have been lost. Moreover, less than 5% of the synchronous frames generated by any single node would have been lost.

Of course, it would be unreasonable purposely to overload a ring over a long period of time. However, level-3 loading might represent a burst of activity that temporarily overloads the network. Our simulation results indicate that even when the ring is heavily overloaded, if synchronous traffic is generated at constant intervals, then setting T_Opr equal to the desired frequency of channel access yields synchronous delays that are acceptable approximately 95% of the time.

It is interesting to note that it is theoretically impossible to set T_Opr so as to guarantee the desired frequency of channel access for either level-2 or level-3 loading, because in each situation there is too much demand for synchronous bandwidth relative to the channel-access requirements. That is, using the terminology of Section 3, $X - Frame\_time - Sync < Sync$. Nevertheless, we obtained satisfactory performance in both these situations by setting T_Opr equal to the desired frequency of channel access.

5. Conclusions

The FDDI standards document [1] suggests that *T_Opr* should be set to a value equal to half the desired frequency of channel access in order to support synchronous traffic. This rule for setting *T_Opr* may be suboptimal for a given ring, since it is valid for any configuration. In this paper we derive a formula to compute the optimal *T_Opr* for a particular ring configuration as a function of the synchronous requirements of that ring. If *T_Opr* is set according to this formula, then the desired frequency of channel access for that particular ring configuration is guaranteed.

Simulation results demonstrate that if synchronous traffic is generated at constant intervals and if it is acceptable that a small percentage of synchronous frames experience delays that exceed the desired upper bound, then setting *T_Opr* equal to the desired frequency of channel access produces satisfactory results and is more efficient than setting *T_Opr* according to our theoretical formula.

References

1. "FDDI Token Ring Media Access Control," American National Standard, X3.139-1987.

2. M. J. Johnson, "Fairness of Channel Access for Non-Time-Critical Traffic Using the FDDI Token Ring Protocol," *Proceedings of Seminar on Real-Time Local Area Networks*, Bandol, France, pp. 145-157, April, 1986.

3. M. J. Johnson, "Proof that Timing Requirements of the FDDI Token Ring Protocol are Satisfied," *IEEE Transactions on Communications*, Vol. COM-35, pp. 620-625, 1987.

4. F. E. Ross, "FDDI - a Tutorial," *IEEE Communications Magazine*, Vol. 24, No. 5, pp. 10-17, 1986.

5. K. C. Sevcik and M. J. Johnson, "Cycle Time Properties of the FDDI Token Ring Protocol," *IEEE Transactions on Software Engineering*, Vol. SE-13, pp. 376-385, 1987.