A High Speed Access Protocol for Distributed Systems

F. Schaffa, M. Willebeek-LeMair, B. Patel
schaffa@watson.ibm.com
IBM T.J. Watson Research Center
P.O.Box 704
Yorktown Heights, NY 10598

Abstract

A demand driven access (ADDA) protocol for a dual ring architecture with spatial reuse is presented. The proposed protocol is simple and distributed. Under light load conditions, the arbitration mechanism is dormant and allows for efficient use of the available bandwidth since no overhead is incurred. In effect, the protocol is demand driven since access arbitration is only activated to guarantee bounded access time for all nodes under heavy load. The protocol is designed for a counter-rotating dual-ring topology. Counter-rotating rings are required to carry back-pressure *control information in the opposite direction of the data flow. Since the protocol does not use a token or a SAT [1], it does not require complex initialization, monitoring and error recovery mechanisms. The spatial reuse of the ring considerably increases the potential throughput of the network. Simulation results demonstrate that for a slotted ring of N nodes, given a uniform destination distribution, a maximum throughput of close to $\frac{N}{4}$ packets per slot time is achieved under heavy load conditions. Under these load conditions, the average access delay per packet is also on the order of $\frac{N}{4}$ slot times. The ADDA protocol is shown to outperform FDDI and MetaRing [1] both in terms of throughput and access delay.*

Key Words: Communication Architecture, Access Protocol, Distributed Systems

1 Introduction

As media speeds increase, the importance of streamlined protocols has become crucial to the efficient utilization of the bandwidth. With the advent of fiber-optics, transmission rates

have dramatically increased. The physical distances over which LANs extend also continue to increase. However, since the speed at which the data can move through the fiber is bounded by the speed of light, in order to make efficient use of the available bandwidth provided by these networks, more data must be placed on the medium at a time (see Fig. 1). This may be accomplished by sending larger packets or batches of packets per transmission opportunity, and/or by allowing multiple simultaneous transmissions. Table 1 shows the number of frames that theoretically could exist simultaneously in the media as a function of network speed.

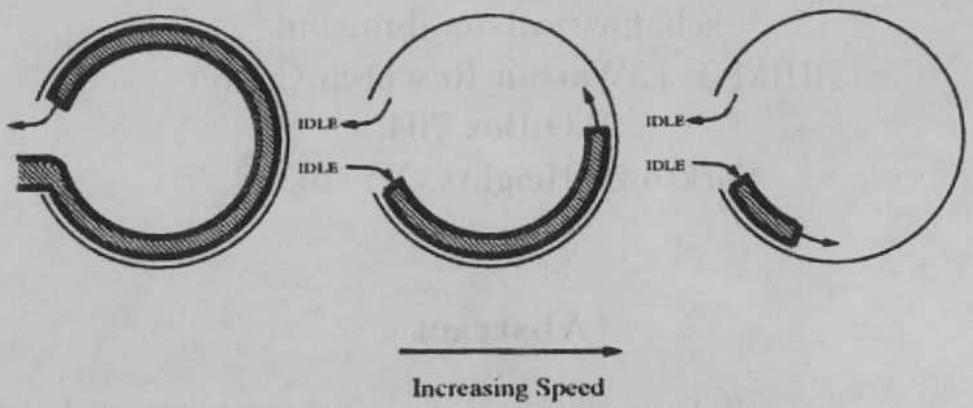

Figure 1: Effect of an increase in speed on the available bandwidth occupied by a frame.

Table 1: Number of possible frames (400byte) in a 10Km network

Speed	Number of frames
10Mbs	0.1
100Mbs	1
1000Mbs	10

Sending larger packets or batch transfers per transmission opportunity may improve the network utilization, however, this will also increase the access delay. This is particularly undesirable for real-time applications. When a packet lifetime in terms of distance on the medium is small due to the capacity/speed of the media, an alternative approach is to allow simultaneous transmission of multiple sources. Allowing multiple tokens on the ring can improve the ring performance at light load, but degrades the performance at high load [2]. Improved throughput as well as smaller access delays can be achieved using a slotted or a buffer insertion ring.

A demand driven access (ADDA) protocol for a counter rotating dual ring architecture is presented in this paper. This protocol evolved from an intrachip communication network protocol [3]. The ADDA protocol is simple and distributed. Under light load conditions the protocol allows for efficient use of the available bandwidth, since multiple stations can

transmit simultaneously. In effect, the protocol is demand driven since access arbitration is only activated as the load increases in order to guarantee all nodes a maximum access delay.

Counter-rotating rings are required to carry *back-pressure* control information in the opposite direction of the data flow. This is similar to the MetaRing design [1]. This information is minimal (one code word), however, and a dual ring topology can be more efficiently utilized by allowing both rings to carry data as well as control information. The control information for one ring is carried in the opposite direction by the adjacent ring, and vice-versa. Furthermore, packets (frames) are removed at the destination node (as opposed to the source node), thereby reducing the lifetime of a packet on the ring and considerably increasing the potential throughput of the network. Frames may also be removed, when removal at the destination fails. A hop counter is also provided in the frame header and can be used to detect stray frames for removal. Figure 2 exemplifies a possible layout of a frame.

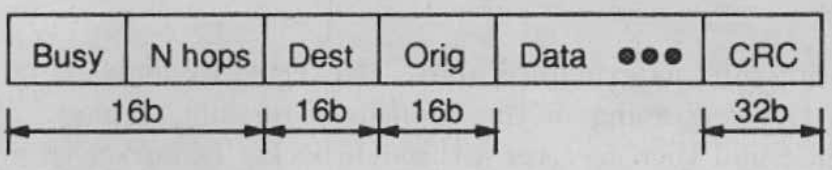

Figure 2: Frame layout.

The protocol is described in Section 2. Section 3 contains a performance evaluation of the protocol using simulation results and a comparison to FDDI and MetaRing is presented in Section 4. Concluding remarks are made in Section 5.

2 The ADDA Protocol

The ADDA protocol is a tokenless protocol, where the access control is based on a node's internal state, rather than on control information that is conveyed by the network (e.g., via tokens in token based protocols). The motivation for an alternative to the token scheme is the high access delay and low throughput of such a scheme, particularly for high speed/capacity media. Moreover, it does not require complex initialization, monitoring and error recovery mechanisms, since the protocol does not use a token or a SAT (the control information for the MetaRing).

The ADDA protocol uses a distributed mechanism to provide guaranteed minimum service for all nodes and to efficiently utilize the available bandwidth. The basic idea behind the ADDA protocol is to allow nodes early access to the ring. This is achieved by permiting a node to send a frame in the first available free slot. This strategy increases the throughput significantly, but can potentially starve nodes which are downstream of a node that is transmitting a large number of frames. To prevent starvation and guarantee a minimum access

delay, a back-pressure signal is sent to upstream nodes to signal that there is a downstream node wanting to transmit.

The distributed access control is implemented using a timer or a counter (C_f) in each node. The counter, started whenever a node has a packet to send, counts the number of successive busy frames. The counter is initialized to a value K, which can be adjusted to allocate disproportionate amounts of the available bandwidth to different nodes (the choice of K is discussed in a later section). If a transmission opportunity appears before the counter expires, the packet is placed in the on the ring and the counter is reset. However, if a packet has waited for more than K slots, the counter will expire, $C_f = 0$, and the node will send a signal to its upstream neighbor requesting an opportunity to transmit (see Fig. 3). The function of the counter is to balance ("spread") the transmission opportunities in such a way that greedy upstream nodes will not starve downstream nodes. The upstream frame request will force greedy nodes to defer their own transmissions and pass free frames downstream to waiting nodes.

The back-pressure, as explained above, is exerted by node N_i in the form of a code word inserted in the ring going in the opposite direction. If node N_{i-1} has received a back-pressure signal and then receives a through packet (a packet coming from N_{i-2}), the through packet (packet in transit) is stored in the through-packet buffer to give a downstream neighbor an opportunity to transmit. Node N_{i-1} will, in turn, send a back-pressure signal to its upstream neighbor to get an opportunity to transmit, and so forth. Thus, the free frame request propagates upstream, until it reaches a node with an empty frame.

A back-pressure request (for a transmission opportunity) is given the highest service priority. The service priorities are ranked as follows: (1) free frame request, (2) through traffic, and (3) local traffic.

Back-pressure (i.e., a request for frame) is applied when a passing frame is busy and either or both of the following events have occurred:

- $C_f = 0$: a local packet is waiting and the frame counter, C_f has expired.
- $C_b > 0$: one or more through packets are queued at this node; C_b is the number of through packets buffered.

The C_f counter is not decremented if a free-frame request arrives from a downstream neighbor. This causes stations to reduce their pressure on the network when the load gets heavy. As a consequence, a node may have to wait up to $N + K$ time slots before making a request. This is the worst case scenario, however, and will rarely occur (see simulation results in Section 3).

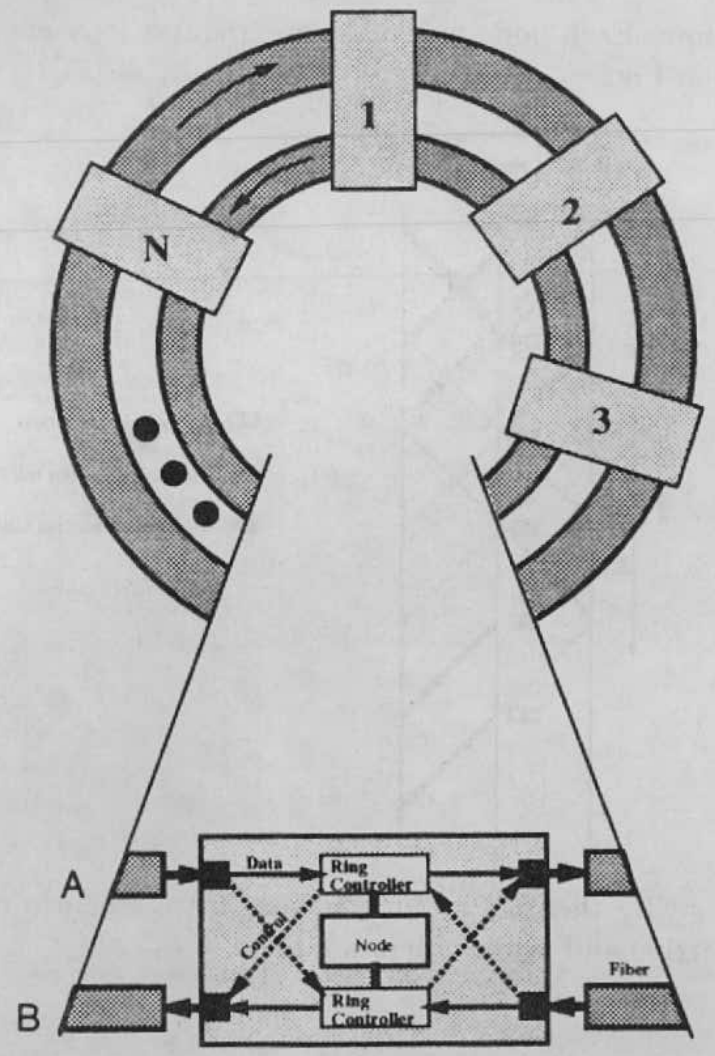

Figure 3: Data and Control Flow on the Dual-Ring. Data arriving on ring A (B) is forwarded to ring A's (B's) controller and control words are redirected to ring B's (A's) controller. Control words originating at A's (B's) ring controller are sent out on ring B (A).

2.1 Buffering Requirements

At each node, for access control purposes, in addition to the frame counter C_f, there is a buffer counter C_b, which is used to keep track of the number of through packets being buffered. It is incremented whenever a through packet is buffered in order to satisfy a downstream neighbor's request. The maximum number of through packets that may need to be stored in a node is equal to the maximum number of frames that could pass through the node within a round-trip delay between any adjacent nodes, t_{RT},

$$C_b = \left\lceil \frac{t_{RT}}{t_f} \right\rceil,$$

where t_f is a frame time. Each node will make one request for each frame to be sent. The concept is illustrated in Fig. 4.

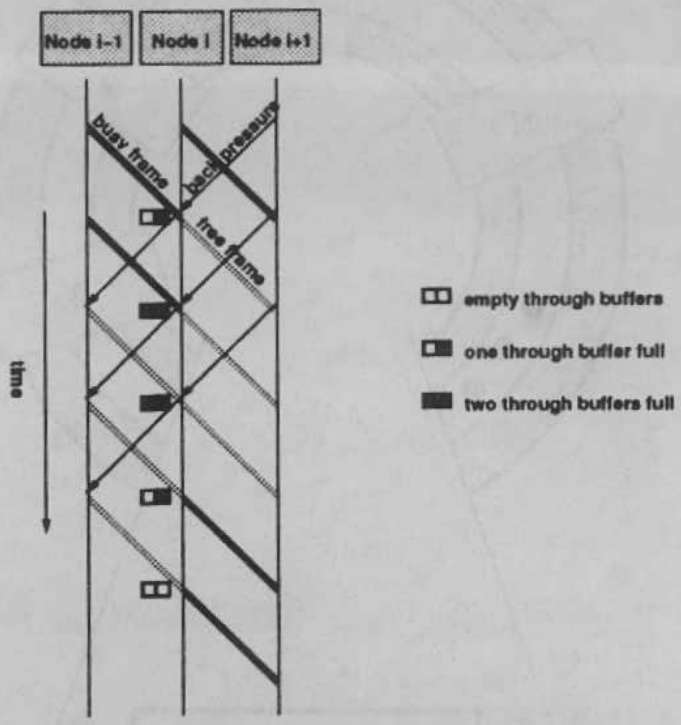

Figure 4: When $t_{RT} = 2t_f$ then $C_b = 2$. The worst case scenario occurs when 2 or more subsequent requests arrive and both buffers are filled.

2.2 The Access Control Parameter K

The access control parameter K governs the behavior of the ADDA protocol. The choice of K determines the trade-off between the access delay and the bandwidth utilization. Larger values of K increase the ring utilization as well as the maximum access delay. The value of K can be varied between nodes to achieve disproportionate bandwidth allocations. Finally, the selection of K is important to regulate the flow of data through the shared medium in order to prevent the possibility of deadlock.

The access delay, determined by the proposed protocol, is bounded by $(K + N)\,t_f$, where t_f is a frame time. However, this upper bound is more theoretical than practical since the necessary conditions for this to occur are very improbable. Under light load conditions, the access delay is less than t_f, Under heavy load conditions, for a ring of N stations, the average access delay will equal $\frac{N}{2}$ (for a uniform destination distribution). However, the dual ring configuration can reduce the distance between source and destination to $\leq \frac{N}{2}$ by sending messages in the direction which minimizes the communication distance. This will reduce the average access delay to $\frac{N}{4}$ and double the average throughput.

Non-uniform bandwidth allocation can be achieved by assigning different values of K_i to the individual nodes. The smaller a node's K_i value is, the more pressure it can exert on the network, and the greater portion of the bandwidth it is able to claim. Therefore, it may be desirable to choose small values of K for servers, bridges, and routers.

The input requests cannot exceed the capacity of the network. For every New Packet Request (NPR), a free slot must be realized somewhere in the network. The number of requests is regulated by the access control parameter K. If the number of requests exceeds the number of available slots, intermediate nodes will be forced to buffer through packets to grant the requests and generate Through Packet Requests (TPRs) to upstream neighbors. The TPRs will, in effect be searching for more free slots than are available. In the worst case, this can result in deadlock, with all nodes forwarding free slots to their downstream neighbors and sending TPR's upstream.

In order to insure that the input requests will not exceed the ring capacity a lower bound must be assigned to K. Given S slots in the ring, the ring capacity, over an interval K, is KS. The maximum allowable input rate during the interval K, is NS (when all N nodes wish to transmit to the furthest destination on the ring—S slots away). Consequently, $KS \geq NS$ or $K \geq N$, i.e., K should not be smaller than the number of nodes in the ring.

3 ADDA Protocol for Slotted Rings

In a slotted ring design, messages are transmitted in fixed size frames. Each slot starts with a tag symbol that indicates whether it is a free or a busy slot. When a free slot arrives at a node, it may place data in the slot and mark it as busy. When a frame from a busy slot is received at the destination node, it will mark the slot free to make it available. This enables slots to be reused at the receiver and potentially multiple times in a single rotation. Control sequence delimiters are used to indicate the start and end of frames. The number of slots in the ring is determined by the overall latency of the network and the chosen frame size. This can be computed as a function of the number of nodes in the network as well as the total distance between nodes. For example, the FDDI ANSI Standard [4] specifies that the latency contribution per node attachment is not to exceed 0.756×10^{-6} seconds and that the latency contribution per kilometer is not to exceed 5.085×10^{-6} seconds. Hence, for a network of 200 stations extending over a distance of 50 km, the total latency would be on the order of 400 microseconds. Given a frame size of 1K bytes at a transmission rate of 100 Mbits/sec, the ring could be divided into 5 slots. A ring with ATM sized slots (53bytes + control information) would have approximately 800 slots. Flexibility in the slot size can be achieved by buffering in the nodes.

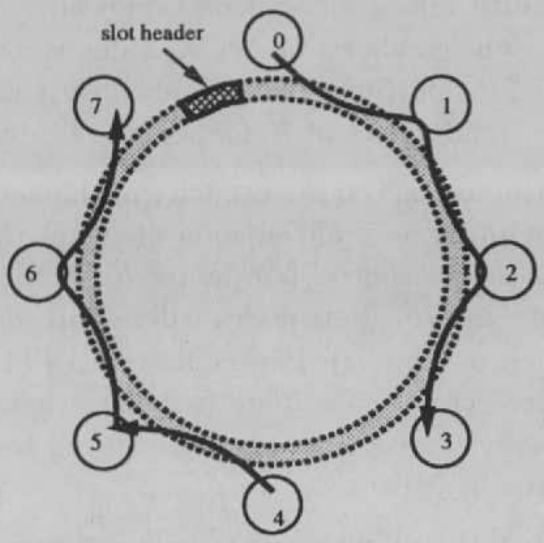

Figure 5: Multiple simultaneous transmissions between stations using a single slot.

3.1 ADDA Performance Analysis

In order to evaluate the performance of the ADDA protocol a simulation model was designed using RESQ (Research Queueing Package) [5]. All results were computed with a confidence interval of 95% and all points are within 10% of the half width of the confidence interval. The achieved throughput, utilization, access delay, and scalability of the ADDA protocol are examined. All performance measures assume a dual-ring topology, however, the results presented pertain to only one of the rings. Hence, although the delays would be the same, the throughput for a dual-ring would be double that shown for the single ring. The results presented in Figs. 7,8,9 assume a ring with $N = 16$ and $K = 16$.

3.2 Throughput

The ADDA protocol delivers higher throughput than token-based protocols. This is due primarily to low protocol overhead and the capability to release slots at the destination. For example, Fig. 5 illustrates a scenario where a single slot exists on a ring of eight stations and three transmissions are occurring simultaneously. Station 0 is sending to station 3, 4 is sending to 5, and 5 to 7. In the worst case scenario, all stations want to send to their immediate upstream neighbors and the ring throughput is $\frac{N}{N-1}$. In the best case, all stations send to their immediate downstream neighbor and the throughput is N.

The communication pattern has a significant impact on the network throughput. In order to examine this effect several communication patterns for a 16 node system were modeled and tested. These patterns are described in Fig. 6 and the simulation results[1] are presented in Fig. 7. The *communication window* shown in Fig. 6 represents the subset of

[1]The average queueing time for a packet is kept within a practical limit of 12 t_f.

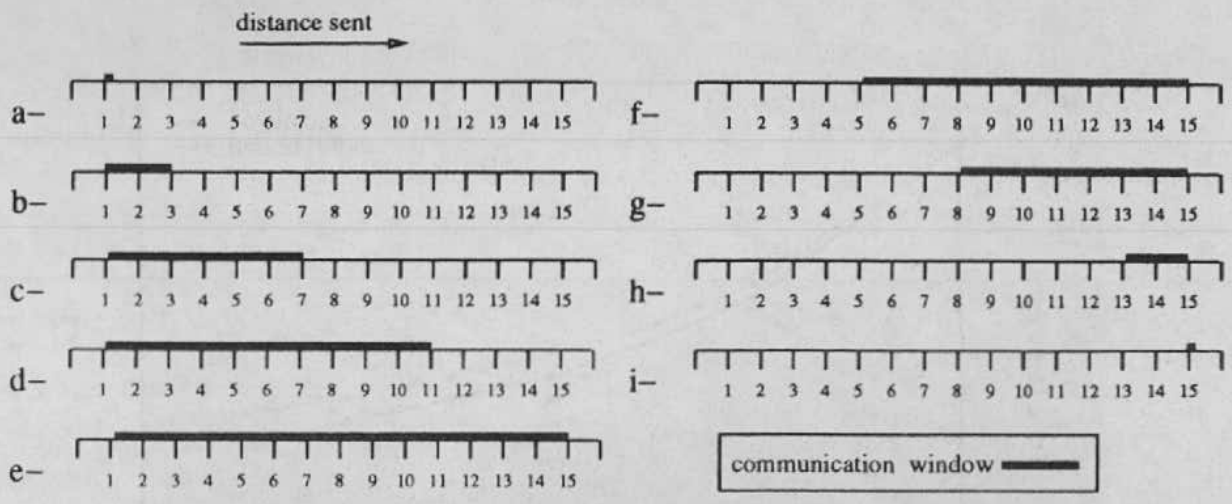

Figure 6: Different communication patterns used to test the effect on the achievable throughput.

destinations (defined in terms of hops from the source) to which each node is sending. Each destination in the window is sent to with equal probability. For example, communication pattern (a) signifies that all communication is between a node and its immediate downstream neighbor (1 hop away). This results in the highest achievable throughput equal to the number of nodes in the system N, since all nodes can transmit simultaneously. Pattern (b) signifies that communication is between a node and any one of its three nearest downstream neighbors. As the window gets larger, the communication pattern is distributed over a larger number of possible destinations (pattern (e) represents a uniform destination distribution). As the window moves to the right, communication is made over a larger distance and slots remain full for more hops. This decreases the availability of slots and consequently increases the protocol overhead to satisfy free-frame requests. The worst case scenario is depicted by pattern (i) where all nodes are sending to their immediate upstream neighbor and slots are occupied for $N-1$ hops. Here, the achieved results are lower than the optimal throughput due to the protocol overhead.

3.3 Utilization

The ADDA protocol achieves very efficient utilization of the available bandwidth for different load conditions. The protocol incurs some overhead every time a free-frame request is made and a node is forced to forward a free frame by withholding data that could be placed in the slot. This overhead will only exist when the load is very high and many nodes are contending for slots in the ring.

A simulation was performed to evaluate the effect of the protocol overhead on the ring utilization for different load conditions. The model consisted of a 16 node ring with 4 slots with uniform destination distribution and exponential interarrival times at each node.

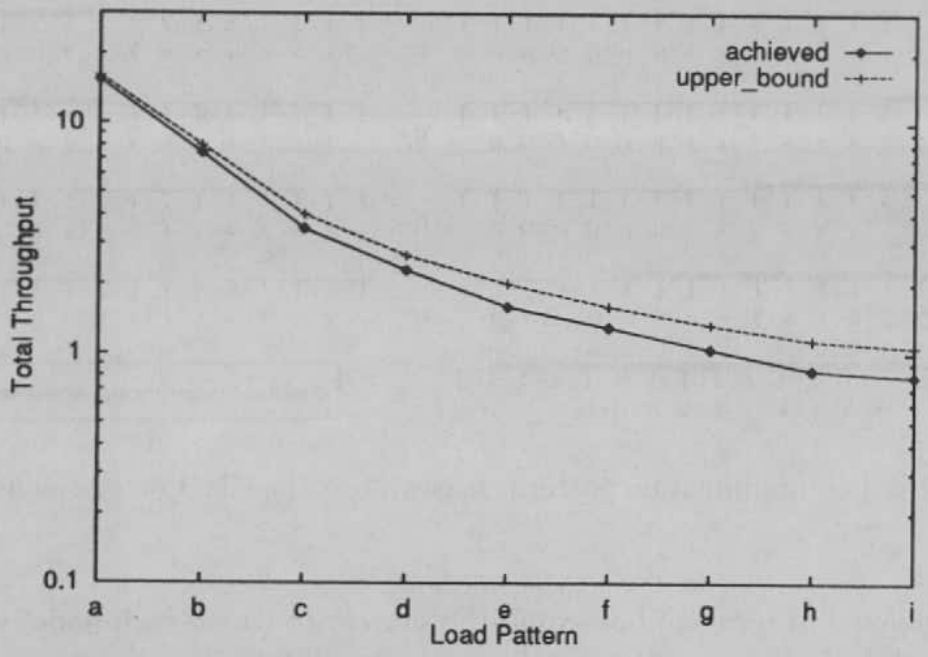

Figure 7: Throughput for the various communication patterns described in Fig. 6.

The results are summarized in Fig. 8. The *free frame request* line in the graph shows the percentage of frames that are forwarded to satisfy free-frame requests for different load conditions. As expected, under light load conditions, the protocol overhead is insignificant since very few frame requests are generated. As the load increases, the protocol overhead shows only a slight increase, but remains a small percentage of the ring bandwidth. What is most apparent from the graph is the high level of utilization that is achieved with the ADDA protocol ($> 95\%$).

The effective utilization depends on the ring slot size and how well the slots themselves are utilized. The choice of slot size depends on the ring latency as well as the communication packet sizes. The ring latency, t_l, determines the total available slot time. As shown in Fig. 5, high throughput can be achieved with even a single slot. However, a very large slot size can lead to underutilization of the bandwidth when large fractions of the slot are unused. Very small slot sizes incur overheads for slot headers and packet segmentation. Hence, a suitable slot size must be selected to most efficiently accommodate the particular network traffic requirements.

3.4 Access Delay

An upper bound for the access delay is guaranteed by the ADDA protocol. This upper bound of $(K + N)\, t_f$ is only reached under very rare load conditions. In general, the average access delay is much lower. This is illustrated with an example of a 16 node ring of 4 slots under

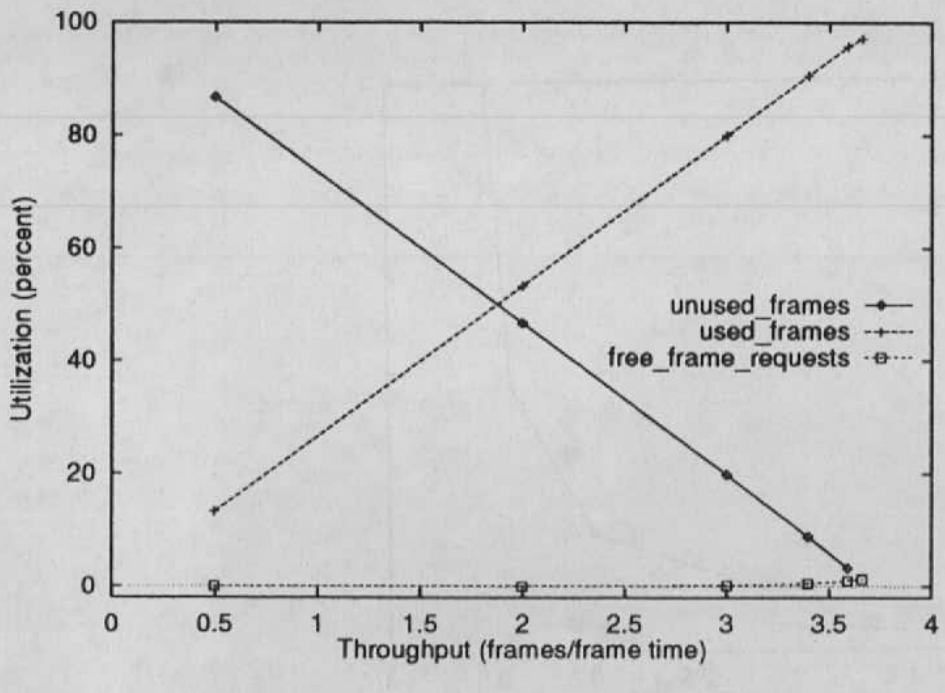

Figure 8: The percentage of free frames, busy frames, and free-frame request frames (protocol overhead) for different load conditions are shown. The results are for a 16 node ring with 4 slots.

various load conditions in Fig. 9. The average total delay, also shown in the figure, includes the time that a packet spends in the source node's queue, the access delay, and the time spent in the network to arrive at the destination. Although there is a slight increment in the access delay as the load increases, the total delay is dominated by the queueing delay.

3.5 Scalability

In order to determine the effect of an increase in the number of nodes (N) on the performance of the network, the average queue length for different size networks was measured. This was done by keeping a constant number of slots on the ring and setting K to N for the different size networks. The results are shown in Fig. 10. In order to maintain the same network throughput, the average node queue length increases as the network size decreases.

3.6 Comparison to FDDI and MetaRing

The Fiber Distributed Data Interface (FDDI)[6] is a 100 Megabit/second Local Area Network (LAN) standard nearing completion. FDDI uses a token ring architecture with optical fiber as the standardized transmission medium. It allows up to 1000 physical connections over a distance of 200 kilometers.

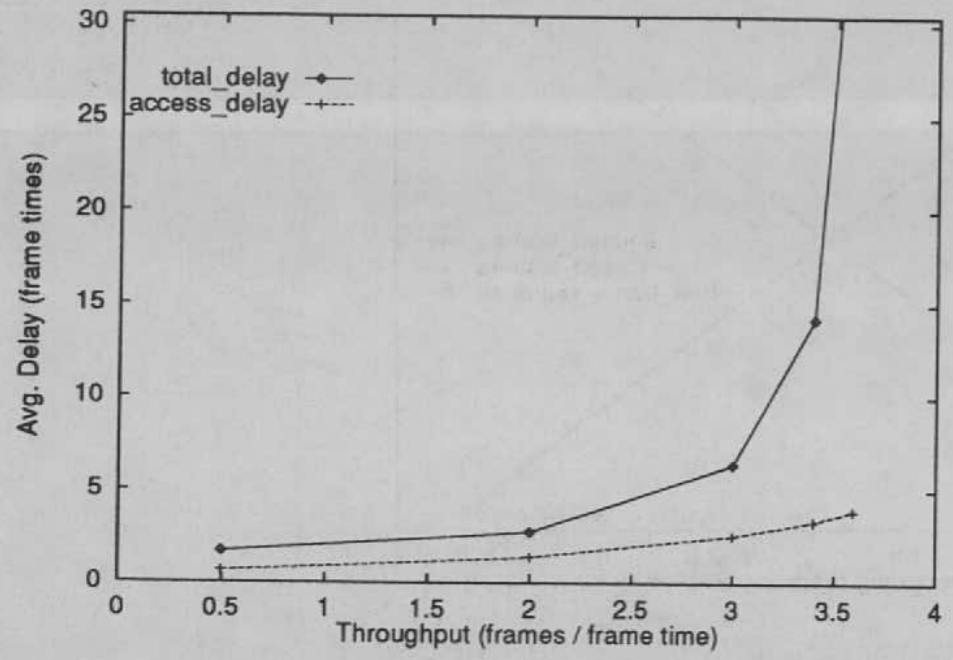

Figure 9: Average total delay versus throughput for various size networks.

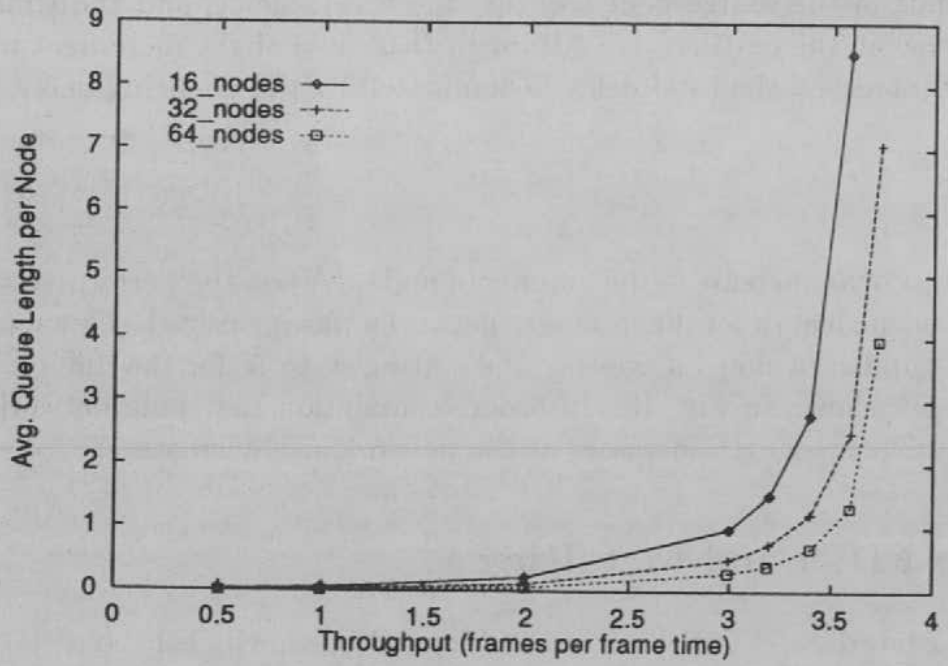

Figure 10: Average node queue length versus network throughput for various size networks.

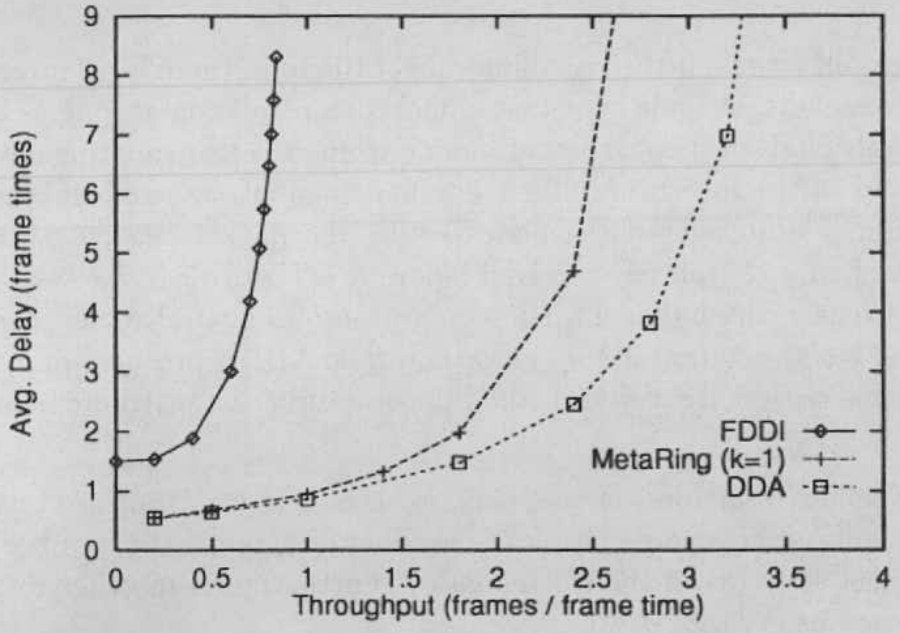

Figure 11: Comparison of ADDA with FDDI and MetaRing.

Metaring [1] is a dual ring architecture that allows concurrent access and spatial reuse with a reliable fairness mechanism. It has two basic modes of operation: one mode with buffer insertion for variable size packets and a slotted for mode for fixed size packets. In our MetaRing model, we use the slotted mode with parameters $k = 1$ and $l = 1$ to ensure fairness. This means that each node send nomore than one packet between SATs.

A performance comparison between the ADDA protocol, FDDI, and MetaRing for a 24 node unidirectional ring was made. New packet destinations are chosen uniformly and independently, and have exponential interarrival time with waiting room equal to one. Furthermore, all traffic is asynchronous and a frame size of 4K bits is assumed. There are three slots in each ring.

Simulation results for throughput versus access delay for the ADDA, FDDI, and MetaRing protocols are plotted in Fig. 11. For a fair comparison, the ADDA and MetaRing throughput were evaluated for only one ring of the dual-ring topology (it would double if both rings were taken into account). In FDDI the throughput can never exceed unity. For light loads, the ADDA yields an average throughput 5 to 7 times greater than FDDI with the same access delay. For heavy loads the ADDA outperforms FDDI by a factor of 3 or 4 with the same access delay. The ADDA yields an average throughput of between 2 and 3 for the same access delays as FDDI.

4 ADDA Protocol for Buffer Insertion Rings

In buffer insertion rings, on the receiving side of the link, there is an insertion buffer used to store through packets. A node can start a packet transmission as long as the insertion buffer is empty. If a through packet arrives at a node while it is transmitting a packet, the through packet is stored in the insertion buffer. The node cannot transmit another packet until the insertion buffer is emptied. If the node is idle, the packet can cut through the insertion buffer without being completely received before it is forwarded. Packets are removed at the destination. Clearly, the buffer insertion ring promotes spatial reuse (similar to the slotted ring), and also has the potential for starvation. The ADDA protocol prevents this by having nodes send a back-pressure signal in the opposite direction of traffic, similar to the slotted case.

In the buffer insertion scheme, packets sizes may vary, but are bounded by P_{max}. In this case, the counter C_f counts either the number of bytes or the number of packets instead of the number of slots (as in the slotted case). Furthermore, in order to avoid deadlock, K should be chosen as NP_{max}.

5 Conclusion

A demand driven access (ADDA) protocol for slotted and buffer insertion rings is presented. This protocol offers greater throughput and smaller access delays than single access token-based protocols. The improved throughput is due primarily to the low protocol overhead and the capability to release slots at the destination. Smaller access delays are achieved since multiple entry points exist on the ring. On average (under uniform destination distribution), the achievable throughput approaches $\frac{N}{4}$ frames per frame-time for a system of N nodes. In general, the throughput depends on the communication pattern. High ring utilization is achieved at all load levels since very little overhead is incurred by the protocol and nodes may continue to send data while slots are available and no other nodes are requesting transmission. Finally, the ADDA protocol is shown to outperform FDDI both in terms of throughput and access delay.

Several research issues concerning the ADDA protocol are being examined. These include, the design of a distributed algorithm for selecting and assigning K values for non-uniform bandwidth allocation.

Acknowledgment. Special thanks to Richard Lamaire for providing the FDDI performance data.

Appendix Description of the Slotted ADDA

Algorithmically, the DDA protocol can be described as:

```
initially, counter ← K, lrf ← FALSE

at frame tick do
    if packet arrived then
        buffer it local_in buffer
        mark frame as free
    fi
    if counter ≠ 0 and local packet waiting
            and no frame_requested then
        counter ← counter−1
    else if counter = 0 and lrf = FALSE then
        lrf ← TRUE
    fi
    if frame_requested then
        buffer incoming packet (if any)
        release free frame
    else if a through packet is present then
            (* either buffered or from the incoming frame *)
        send it
    else if local packet waiting then
        send it
        counter ← K
    fi
    if through_buffer > 0
            and frame_requested then
        request frame from upstream node
    else if lrf = TRUE then
        lrf ← FALSE
        request frame from upstream node
    fi
od
```

References

[1] Israel Cidon and Yoram Ofek. Metaring - a full-duplex ring with fairness and spatial reuse. In *Proceedings of the 1990 IEEE INFOCOM*, 1990.

[2] Ahmed E. Kamal. On the use fo multiple tokens on ring networks. In *Proceedings of the 1990 IEEE INFOCOM*, 1990.

[3] Frank Schaffa. *Communications on VLSI.* PhD thesis, University of California at Los Angeles, CA 90024, 1989.

[4] Preliminary Draft proposed American National Standard. *FDDI Station Management (SMT)*, 1990. ANSI X3T9.5/84-49.

[5] C. H. Sauer, A. Blum, P. Loewner, E. MacNair, and J. Kurose. The Research Queueing Package Version 2. Technical Report RA 139, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1986.

[6] F.E. Ross. FDDI - A Tutorial. *IEEE Communications Magazine*, 24(5), May 1986.