An Efficient Medium Access Control Protocol For High-Speed Networks

Mounir Hamdi†, K. B. Letaief‡, and Jun Gu†
†Department of Computer Science
‡Department of Electrical and Electronic Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
Email: hamdi@cs.ust.hk

Abstract

In this paper, we propose a new medium access control (MAC) layer protocol, called *Optical Reservation Multiple Access* (ORMA), which is suitable for data traffic on a folded bus high-speed MANs and LANs. Its main feature is an optical reservation technique, to allow stations to reserve transmission slots, which is attractive for use in high-speed networks. Unlike traditional protocols where reservation is mostly done using software, ORMA performs its reservation function using simple and efficient optical hardware circuits. A complete description of the ORMA protocol is given. Then, ORMA is shown to achieve high throughput and small transmission delays while preserving the fairness of the whole network.

1 Introduction

Medium Access Control, or MAC, protocols have been the subject of a vast amount of research over the past two decades. One reason for this research is that all higher layer services are built on the fundamental packet transfer service which is provided by the MAC sub-layer, and it is the MAC protocol which determines the characteristics of this service. Hence, improvements to MAC services result in improved system performance, while the provision of new MAC services means that new applications can be developed. Our aim in this paper is to present and evaluate the design of new MAC protocol, termed *Optical Reservation Multiple Access* (ORMA), which can satisfy the performance requirements of optical high-speed networks. The ORMA protocol is based on an efficient explicit reservation mechanism. The originality of the protocol lies on the decoupling of the reservation cycles and the transmission cycles to achieve high performance and simplicity in the protocol.

This paper is organized as follows. In Section 2, we overview state-of-the-art MAC protocols for high-speed networks. Section 3 introduces the architecture of the ORMA network. Section 4 describes the ORMA MAC protocol and its implementation. Section 5 describes the results of our discrete-event simulation. In Section 6, we give some concluding remarks.

2 Alternative Approaches

Recently, several new MAC protocols for high-speed LAN's and MAN's have been proposed including *Distributed Queue Dual Bus* (DQDB) [8], *P_i-Persistent* [13], *Load-Controlled Scheduling of Traffic* (LOCOST) [7], and *Cyclic Reservation Multiple Access* (CRMA) [10]. Most of these protocols are based on either a folded or dual unidirectional bus topology, primarily due to the unidirectional nature of the fiber optic medium. Further, they use fixed-size transmission slots generated by head-end nodes for data transmission. The DQDB MAC level protocol has been accepted as the IEEE 802.6 standard for high-speed MAN networks [8]. DQDB can achieve full channel utilization. However, some problems with unfairness in the protocol has been identified [3, 5, 6].

Under the P_i-Persistent protocol, a ready station persists with its attempts to transmit its packet in an empty slot with probability P_i until the transmission is complete. In order to increase the channel utilization and to be fair to all stations each individual station needs to modify its P_i based on the channel activities, the estimated number of active stations, and their traffic loads [11, 12]. Using, the LOCOST protocol, every station measures the traffic intensity of the channels and then, based on this measurement, it determines its transmission rate until the next measurement is made [7]. The main idea in the last two approaches is requiring each station to monitor the traffic on the channels and, based on the statistics observed, to throttle its transmission rate accordingly. They can indeed improve the fairness of the protocol. However, they have two potential problems associated with these schemes. First, each station adjusts its transmission rate according to the estimated traffic load. It is very likely the estimated load is different from the real load. Therefore, a high channel utilization may not be achieved. Second, it may take a longer period of time for the system to reach a completely fair state. It is also possible, especially when the traffic fluctuates dynamically, that a complete fair state may never be reached.

The cyclic reservation MAC protocols attempt to solve the unfairness of the DQDB-type protocol and the instability of the statistics-based protocols through explicit reservation mechanisms whereby stations have to reserve transmission slots in advance in order to get access to the channel [1, 10]. Among the many proposed cyclic reservation protocols, the CRMA has received the most attention [1, 10] because it can achieve a high channel utilization while guaranteeing fairness among the stations. However, the CRMA protocol has some problems of its own. First, CRMA requires more complex structures both at the nodes and the head-end station. As a result, the cost of an interface board between a station and the network can be almost as costly as the stations themselves as was shown in [10]. Second, the reservations made by the nodes are not certain and need to be confirmed by the head-end to be valid. They can be rejected and, in such a case, a retry is necessary by the nodes.

Our aim in this paper is to design a new MAC protocol for

0-7803-2509-5/95 US$4.00 © 1995 IEEE

high-speed LAN's and MAN's that can retain the nice characteristics of the CRMA protocol while solving all of its problems. In addition, the associated hardware cost of this protocol should be minimized to justify its cost effective implementation. Towards achieving this goal, we propose a new MAC protocol termed *Optical Reservation Multiple Access* (ORMA). The ORMA protocol uses novel and simple hardware solutions for cycle reservation which can be overlapped with the data transmission by employing separate reservation channels.

3 ORMA Network Architecture

The proposed ORMA protocol is suitable for high-speed data transmission on a folded bus; like the CRMA protocol. However, both the CRMA and ORMA can be easily extended to operate on a dual unidirectional bus architecture [11]. The architecture for an ORMA network is depicted in Figure 1. Similar to most folded bus networks, there are two special nodes in the network, namely a *head-end* node and a *fold* node. The fold node divides the bus into an *out-bound* segment and an *in-bound* segment. An attached node (station) uses the out-bound segment for transmitting messages to destination nodes, and uses the in-bound segment for receiving messages from the various stations in the network. The message sizes are of fixed length and the transmission and reception of messages are performed using fixed size slots. These slots are generated by the head-end node.

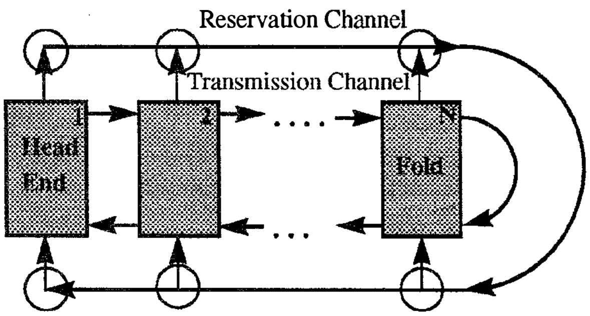

Figure 1. The ORMA network architecture.

In an ORMA network, there are three separate channels. The first one is the data channel which is used by the attached nodes for exclusively sending and receiving messages. The second channel, termed *reference* channel, and the third channel, termed *select* channel, are employed as *reservation channels* for the nodes to request transmission slots for their ready messages in the coming network cycles. The three channels may use three physically separate channels or they can employ *Wave Division Multiplexing* (WDM) to partition the high bandwidth of the optical fiber into three sub-channels.

As mentioned before, the main feature of the ORMA protocol is the employment of separate channels to perform reservation, and the fact that these reservation are done in hardware rather than the slow traditional software. The ORMA network uses conditional delays to perform the reservation. Specifically, at the beginning of each reservation cycle, each attached station injects a binary 1 into the network if it wants to reserve a transmission slot for its packet. Consequently, the sum of the binary 1's is exactly equal to the total number of reserved slots by all stations in the networks. Hence, the main thrust of this reservation scheme is to perform optically (with no electronic intervention) and as fast as possible the addition of binary numbers to determine the total number of slots reserved. The main purpose of adopting this idea to perform reservation is to avoid forcing the station to read and write onto the reservation slots; a process which can delay the transmission rate of the reservation slots at each node [1]. The operation of the optical binary adder is illustrated by Figure 2 which consists of a *reference* channel, a *select* channel, and 2×2 optical switches.

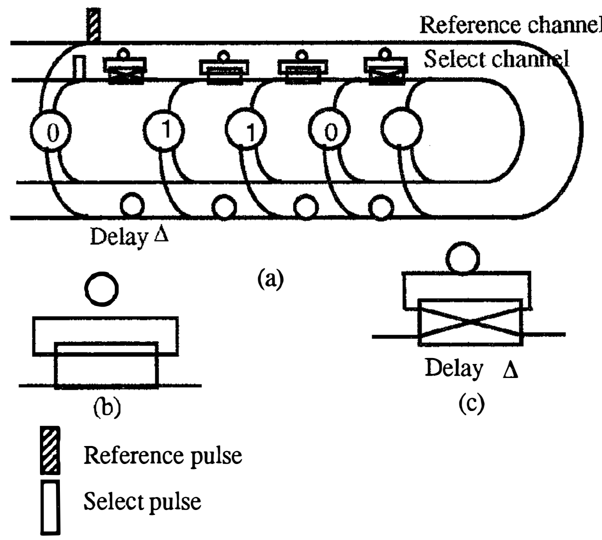

Figure 2. The ORMA network reservation scheme architecture.

The hardware summing algorithm uses *coincident pulse addressing* on folded optical networks with conditional delays. The conditional delays can be implemented using 2×2 optical switches, as shown in Figure 2(a). In this network, each switch $S(i)$ is controlled by Station $P(i)$. If all the switches are set to ***straight***, then an optical signal incurs the same propagation delay on both the *reference* channel and the *select* channel between any two stations i and j in the ORMA network. However, an additional time delay equal to Δ can be introduced on the *select* channel by setting switch $S(i)$ to *cross*, as shown in Figure 2(c). In other words, when $S(i)$ is set to ***straight***, it takes a time τ for an optical signal to propagate from $P(i)$ to $P(i+1)$ on the channel, while when $S(i)$ is set to *cross*, such propagation will take a time $\tau + \Delta$.

In the following, the algorithm for computing the binary sum on the ORMA network of Figure 2(a) is presented. In this algorithm, it is assumed that the binary data are initially stored in the stations with station $P(i)$ holding a_i, and all the optical switches are set to *cross*. Then a *reference* pulse and *select* pulse are inserted simultaneously on the *reference* channel and *select* channel, respectively, by station $P(0)$ (head-end node). If

$\sigma = \sum_{i=0}^{n-1} a_i$, then σ delays (each delay equal to Δ) will be removed from the *select* channel such that the reference pulse and the select pulse coincide at Station $P(\sigma)$.

Algorithm BINARY-SUM

Input: a binary sequence $a_i = 0$ or 1. Initially a_i is stored at $P(i)$.

$$\text{Output:} \quad \sigma = \sum_{i=0}^{n-1} a_i \quad .$$

1) $P(i)$ sets $S(i)$ to *straight* if $a_i = 1$, *cross* if $a_i = 0$.
2) At time 0, $P(0)$ injects a reference and a select pulse signals. If $P(j)$ is selected, then the sum is $\sigma = j$. That is, the index j of the station that sees the coincidence of the reference pulse and the select pulse gives the sum σ.

Note that to compute the sum of a binary sequence of length n, n + 1 stations labeled from 0 to n are needed in the ORMA network since there may be from 0 to n binary 1's in the sequence.

Proposition: In the algorithm *BINARY-SUM*, the index j of the station which sees the coincidence of the reference pulse and the select pulse is equal to the sum $\sigma = \sum_{i=0}^{n-1} a_i$.

Proof: Since both the reference pulse and the select pulse are injected simultaneously on the *reference* channel and the *select* channel, respectively, at time $t = 0$ (the beginning of a reservation cycle), then the time at which the reference pulse arrives at station j (on the lower part of the *reference* channel) is $t_{r,j} = (n+1)\tau + (n-j)(\tau + \Delta)$. Let σ be the number the number of 1's in the binary sequence a_i. Then σ switches will be set to *straight*. The time at which the select pulse arrives at Station j (on the lower part of the select channel) is $t_{s,j} = (n+1)\tau + (n-\sigma)\Delta + (n-j)\tau$. Let $t_{r,j} = t_{s,j}$, one obtains $\sigma = j$. ■

With the above hardware algorithms for computing the sum of binary numbers, the ORMA network can perform its reservation fast and independent of the data transmission channel. At the beginning of each reservation cycle, each station will set to *cross* its attached switch if it desires to reserve a slot for its packet. Otherwise, it will leave its switch in the *straight* position. At the same time, the head-end node will simultaneously inject two optical pulses into the *reference* channel and the *select* channel respectively. Consequently, a coincident pulse will occur at the station whose index equals the sum of reservations made by all station in the ORMA network. Thereafter, this station will send its index to the head-end node to inform it about the number of reservations made and hence the length of the transmission cycle.

Thus, the index of the station where the coincident pulse occurs corresponds to exactly the number of transmission requests made by all stations in the ORMA network. As a result, we have a very efficient hardware scheme that would give us the exact number of transmission requests during a transmission cycle. Hence, the nodes access the channel according to cycles. A cycle consists of a variable number of slots of fixed size. Cycles represent the payload capacity reserved in previous reservation cycles. A low level of reservations result in short cycles, whereas a high level of reservations results in accordingly long cycles.

This reservation method has many significant characteristics.

1) The ORMA reservation scheme gives us the exact number of reservation requests during a transmission cycle; not an estimation such as those adopted by many recently proposed protocols [4, 12].
2) Because of the separation of the reservation channels and the data channel, coupled with the fact that reservation can be done quite fast, any node can have very fast access to the ORMA network whenever they desire to transmit. As a result, the ORMA protocol can have a high performance under all types of traffic loads.
3) The ORMA protocol achieves total fairness among all the stations regardless of their positions on the folded bus. Each station gets an opportunity to perform reservation on any transmission cycle. Hence, its network access delay can be easily bounded. This is an important characteristic for real-time communications such as in multimedia applications.

4 ORMA Protocol

The ORMA manipulates two asynchronous transmissions: message transmission and reservation transmission. The message transmission is performed using fixed transmission slots similar to those employed by MAC protocols such as DQDB and CRMA. Each slot consists of two fields. One field is a 2-bits flag; the most significant bit indicates whether the slot is empty or full (E/F bit) and the other bit, which is a train-tail (TT), indicates if the given slot is the last slot in a transmission cycle. If the E/F bit is equal to 1 it indicates that the slot is full and if it is equal to 0 it indicates that the slot is empty. If the TT bit is equal to 1 it indicates that the given slot is the last slot in the transmission cycle and if it is equal to 0 it indicates that the slot is not the last slot in the transmission cycle. The format of the slot is shown in Figure 4. The remaining field is the data field where the message is to be loaded which includes the source and destination addresses.

We denote a *reservation cycle* to be the period of time between the initiation of a reference and a select pulse by the head-end node until the time it receives the index of the station where the coincidence pulse occurred. A *transmission cycle* denotes the time needed to produce a sequence of slots equal to the number of reservation requests. The sequence of slots which are generated during one transmission cycle forms a *train*. Each train has a format similar to that of Figure 4. All the slots in a train have the TT bit set to 0 except the last one where TT = 1 to indicate to the nodes the "train-tail".

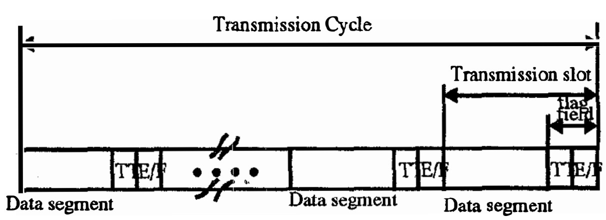

Figure 4. The transmission cycle and slot format of ORMA protocol.

4.1 Reservation cycle

The head-end node could initiate a *reference* and a *select* pulse once the index of the station where the coincidence pulse has occurred has been received from the in-bound segment of the bus. However, in order to decrease the reservation cycle time, the select pulse and the reference pulse could be generated in a pipeline fashion without having the head-node to wait for a reservation cycle to complete. When the head-end receives the index of the station where coincidence pulse has occurred from the in-bound segment of the bus, it directly determines the exact number of slots that should be produced for the corresponding transmission cycle. If the index received is 0, then, no slots should be generated. if the index received is 1, then, a single slot should be generated, and so on. The exact number of slots is equal to the index received by the head-end node.

The reservation cycle is obviously much shorter than the transmission cycle. Consequently, we have to have a way of buffering requests by the stations, and we also should have the capability of buffering the results of the reservation cycles in the head-end node. Hence, in an ORMA network, we should have two types of queues. The first type is the *Local Request Queue* (LRQ) on each node, which is used to buffer the transmission requests generated by the corresponding node. The other type is the *Global Request Queue* (GRQ) on the head-end node where the reservation results of each cycle are contained.

4.2 Data transmission cycle

During each transmission cycle, the head-end node generates a number of transmission slots equal to the number obtained during the reservation cycle, and which is stored in the GRQ. Further, the head-end node removes that value from its GRQ. When a slot passes by a node, the node checks the flag bit if the head of its LRQ is a YES flag. If the slot is empty, the E/F bit = 0, then the node inserts its message into the data segment field of the slot. If the head of the node's LRQ has a NO flag, then the slot passes to the next node with no modification.

When the *train-tail* (determined by checking the TT bit of each slot) slot passes by, the item in the head of the LRQ is dequeued. The dequeuing operation is carried out by all nodes including the nodes having items in the head of their LRQ equal to a NO. As a result, all the slots generated by the head-end node will be utilized by the stations. That is why the performance of ORMA can be close to the theoretical limit.

5 Performance Evaluation

The performance of the ORMA is being analyzed in terms of throughput, average packet delay, and fairness using discrete-event simulation employing SimPack [9].

5.1 Network Throughput

Network (or system) throughput is defined as the number of packets that are transmitted per slot time across the network. Since we are considering only one data channel, the maximum network throughput is 1 (100%). Time is slotted with unity duration so that the channel utilization is the same as the network throughput. By using separate sub-channels for reservation, we eliminate any additional overhead due to reservation slots in the transmission sub-channel, unlike various proposed protocols such as CRMA protocol, S++ protocol, and CBRMA++ protocol [1, 2, 14]. Hence, all available bandwidth of the transmission channel is used for data transmission except the unavoidable two-bit flag field and node addresses in each slot. Further, since the generation of reservation bits and their transmission is much faster than that of a data slot, the ORMA protocol can satisfy the reservation requests of the active nodes almost instantaneously. Moreover, our ORMA reservation scheme works well for uniform traffic as well as bursty traffic, and under any network load conditions. This results in almost 100% bandwidth utilization in the transmission sub-channel.

Figure 5 shows the plots for the throughput of ORMA for different network sizes and different traffic loads using our discrete-event simulation. By examining the throughput curves, we can make the following observations. First, the throughput is independent of the number of stations attached to the network as expected. For a given network load, the network utilization is the same no matter how many stations are active. Second, the throughput increases linearly with the traffic load. When the traffic load reaches 1, then the ORMA network throughput reaches its maximum value of almost 100% system utilization.

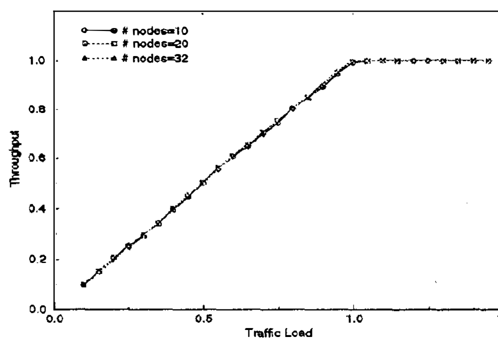

Figure 5. ORMA network throughput.

5.2 Average Packet Delay

The packet delay is the time taken from the instant a packet is generated at the source node to the instant it is received at the destination node. This time includes the waiting time in the queue, transmission time, and cycle synchronization time. Figure 6 shows the average delay as a function of the traffic load. The average delay becomes excessively large only when the system load becomes larger than 1. That is, we are in a situation where the stations are transmitting messages more than the network can handle. However, when the load of the system is less than 1, then the average delay is almost constant. This, in turn, can have good implications for real-time applications, where the delay can be bounded. Further, increasing the network size does not increase the average delay tremendously.

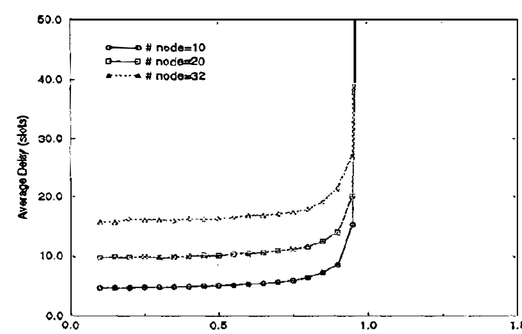

Figure 6. ORMA network average delay.

5.3 Fairness

The ORMA protocol is a fair protocol, that is, each station has equal chance to transmit a message regardless of its position in the folded bus. This can be seen from the fact that during each reservation cycle, each station has the chance to reserve one slot, which would be available to it in the corresponding transmission cycle no matter how many other stations are making reservations. Moreover, the ORMA protocol does not sacrifice any degradation in the throughput to have a fair network. In DQDB or FQFB, the fairness is based on some prefixed estimation strategy, or fair ware [3]. When the estimation is away from the actual request distribution, some of the slots may be wasted. While in ORMA, even in the case that there is only a single node having a lot of packets to send, the head-end node can still produce successive slots to meet the needs of that node. Figure 7 illustrates the fairness in throughput by showing the fraction of throughput allocated to each station as a function of its index.

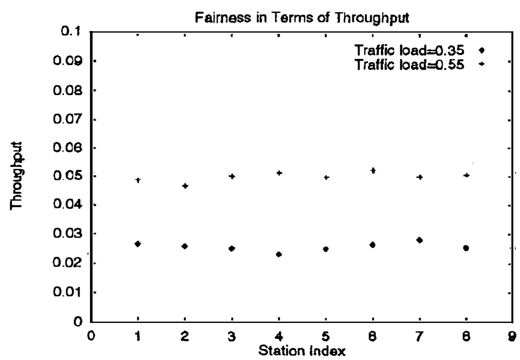

Figure 7. Fairness in terms of throughput of the ORMA protocol.

As can be seen, for different values of traffic load, the throughput of all stations are almost the same. This is unlike the DQDB, where the throughput of the stations in the middle of the network are much higher than the ones at either end of the network. However, the fairness of ORMA in terms of average delay is not as good as that in terms of throughput. This is illustrated in Figure 8. The stations which are closer to the head-end tend to have a slight advantage over stations which are further from the head-end since they get to use the reserved slots first. That is, if during a reservation cycle station 1 and station 2 each reserved one slot. Then, station 1 will use the first slot and station 2 will use the second slot. This, in turn, would give advantage to station 1 as far as average delay is concerned.

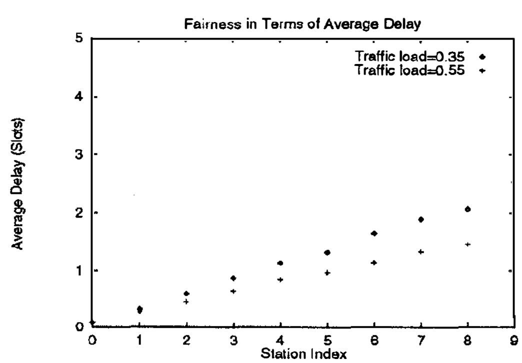

Figure 8. Fairness in terms of average delay of the ORMA protocol.

6 Conclusion

In this paper, we propose a new MAC protocol for LANs and MANs denoted ORMA. The main feature of this protocol is a simple and fast reservation scheme which is attractive for use in high-speed networks. The reservation is performed using a simple hardware circuit unlike traditional protocol where arithmetic processing is generally needed. We evaluated the performance of ORMA using a discrete event simulation. We find that the network throughput approaches the behavior of a theoretical model where it increases linearly as a function of the network load when it is less than 1. Then, it approaches 100% utilization when the network load reaches 1. Further, the average delay is found to be small and almost constant which makes our protocol suitable for real-time applications. Finally, it was shown that ORMA is a fair protocol especially as a function of the throughput allocated to each station regardless of the network load.

Acknowledgment: This research is supported in part by the Hong Kong Research Grant Council under the Grant RGC/HKUST 619/94E.

References

[1] G. Anastasi, M. Conti, E. Gregori, and L. Lenzini, "Service integration in CRMA: A simulative analysis," in *INFOCOM '93*, pp. 715-721, 1993.

[2] C. Baransel and W. Dobosiewicz, "CBRMA++/SR: On the design of a MAN/WAN MAC protocol for high-speed networks," *IEEE JSAC*, pp. 1268-1277, 1993.

[3] F. Borgonovo, A. Lombardo, S. Palazzo, and D. Panno, "FQDB: A fair multisegment MAC protocol for dual bus networks," *IEEE JSAC*, pp. 1240-1248, 1993.

[4] K. W. Cheung, L. K. Chen, C. Su. C.T. Yeung, and P. T. To, "TCMA networks: A new class of high-speed network suitable for multimedia integrated networking," in *Proceeding of the SPIE*, Vo. 2024, pp. 122-133, 1993.

[5] M. Conti, E. Gregori, and L. Lenzini, "DQDB/FBS: a fair MAC protocol stemming from DQDB fairness analysis," in *2nd IEEE Workshop on Future Trends of Distributed Computing Systems*, pp. 152-159, 1990.

[6] M. Conti, E. Gregori, and L. Lenzini, "A methodological approach to an extensive analysis of DQDB performance and fairness," *IEEE JSAC*, pp. 76-87, 1991.

[7] M. Fine and F. A. Tobagu, "Demand assignment multiple access schemes in broadcast bus local area networks," *IEEE Trans. on Communications*, pp. 1130-1159, 1984.

[8] IEEE. *Distributed Queue Dual Bus (DQDB) ANSI/IEEE* 802.6, 1990.

[9] Information Science, University of Florida. *SimPack: Getting Started with Simulation Programming in C and C++*, 1992.

[10] M. M. Nassehi, "CRMA: An access scheme for high-speed LAN and MANs," in *IEEE International Conference on Communications*, pp. 115-121, 1990.

[11] C. Partridge. *Gigabit Networking*. Addison Wesley, 1993.

[12] W. Stallings, "Local network performance," *IEEE Network Magazine*, pp. 27-36, 1984.

[13] F. Tobagi, "Multiaccess protocols in packet communications systems," *IEEE Trans. on Commun.*, pp. 468-488, 1980.

[14] G. C. Watsonm and S. Tohme, "S++ a new MAC protocol for Gb/s local area networks," *IEEE JSAC*, 1993.