MetaRing—A Full-Duplex Ring with Fairness and Spatial Reuse

Israel Cidon, *Senior Member, IEEE,* and Yoram Ofek

Abstract—**We describe the design principles of a ring network with spatial bandwidth reuse. Our goal is to provide the same functions of existing LAN/MAN designs that do not permit spatial reuse and concurrent transmission. A distributed fairness mechanism for this architecture, which uses low latency hardware control signals, is presented. The basic fairness mechanism can be extended for implementing multiple priority levels and integration of asynchronous with synchronous traffic.**

The ring is full-duplex and has two basic modes of operation: buffer insertion mode for variable size packets and slotted mode for fixed size packets or cells. As a result, this architecture is suitable for a wide range of applications and environments.

Concurrent access and spatial reuse enable the simultaneous transmissions over disjoint segments of a bidirectional ring, and therefore, can increase the effective throughput, by a factor of four or more. The efficiency of this architecture does not degrade as the bandwidth and physical size of the system increases.

The combination of a full-duplex ring, spatial reuse, reliable fairness mechanism and the exploitation of the recent advent in fiber-optic technology are the basis for the MetaRing network architecture. This network has been prototyped at the IBM T. J. Watson Research Center, and will also be deployed within the AURORA Testbed that is part of the NSF/DARPA Gigabit Networking program.

I. Introduction

LOCAL area networks (LAN's) form an efficient and cost-effective solution for the in-site data communication problem. They are simple to control, manage and access, and make use of a low-cost hardware that occupies a small space.

In order to reduce the complexity of the LAN, most designs focus on simple topology structures in the form of a bus, star, or ring. In order to further simplify the architecture, most current local area networks do not permit concurrent access with spatial bandwidth reuse of the LAN ([1]–[7]). In some of the LAN's, this restriction is inherent due to the passive nature of the transmission media (e.g., Ethernet, passive optical star). Others, like the dual token ring networks (FDDI—[3], [4]) or dual bus networks (DQDB or QPSX—[5]–[7]), have adopted such a design point in order to achieve simplicity and fairness. Note that the early token release scheme of FDDI does not imply spatial reuse, since at most one node (the one that holds the token) can transmit into the ring at any given time.

The new trend in LAN's architecture is to allow for more users, higher traffic rates, and larger area coverage. The emerging fiber-optic technology makes such designs feasible. However, for cost effectiveness, the introduction of spatial bandwidth reuse can further increase the effective capacity for the same technology generation and cost.

In this work we show how to increase the throughput of a ring-based local area network far beyond its single link capacity and still to conserve its basic *simplicity and fairness* properties [8], [9]. The proposed network is a bidirectional buffer insertion or slotted ring, that is constructed of full-duplex serial links. Furthermore, we show how the basic fairness mechanism can be extended, in a distributed manner, for supporting multiple priority levels and bandwidth reservation for synchronous traffic. As a result, this architecture is functionally equivalent to existing ring and bus-based LAN architectures, with the advantage of higher throughput.

Spatial reuse, or the ability to provide concurrent transmission over distinct segments of the ring, can significantly increase the effective ring throughput ([10]–[13]). By a simple observation one may realize that when the traffic pattern is homogeneous, a factor of 2 can be gained in a unidirectional ring structure by introducing spatial reuse. When a bidirectional ring structure is used, with a shortest path routing rule, the average distance becomes only 1/4 of the ring circumference, and the average spatial reuse is of four nodes transmitting at the same time (on each direction).

Spatial reuse may cause *starvation,* which happens if some node is constantly being "covered" by an up-stream ring traffic and thus is not able to access the ring for a very long period of time. This work introduces fairness mechanisms that regulate the access to the spatial reuse ring, for solving the starvation problem with minimal impact on the network throughput and delay. Fairness mechanisms for unidirectional slotted ring with spatial reuse were introduced in MAGNET [10], Orwell [11], ATMR [14] and BCMA [15]. The fairness algorithms of these architectures operate using network-wide fairness cycles and allow all nodes to enter the new cycle after the previous one is completed at all nodes. This operation may result with an idle time between successive fairness cycles, while the termination detection of a previous cycle takes place. This idle time is sensitive to the ring propagation delay. The fairness algorithm presented in this work operates continuously and follows the natural ordering along the ring. Therefore, it is less sensitive to the ring propagation delay. It is also more versatile since the

Paper approved by the Editor for Voice/Data Networks of the IEEE Communications Society. Manuscript received October 26, 1989; revised September 25, 1991. This paper was presented at the IEEE INFOCOM '90, San Francisco, CA, June 1990.

I. Cidon is with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598. He is also with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 3200, Israel.

Y. Ofek is with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 9203586.

basic mechanism can be used to implement several different fairness algorithms and can be used for the transmission of variable size packets.

A global fairness mechanism views the whole ring as a single resource and gives all nodes equal transmission opportunity. We present a very simple and efficient global fairness mechanism, which is based on a single control signal. In the case of a full-duplex ring, the control signal is rotating in the opposite direction to the data traffic that it regulates and has a preemptive resume priority over the regular data packets (i.e., data packets are kept intact). The global fairness mechanism can tolerate signal loss or duplication. This mechanism can also be extended to provide functions like asynchronous priority handling (as in IBM token-ring), and the integration of synchronous and asynchronous traffic (as in FDDI [3], [4]).

In a related paper [16] we have introduced the notion of local fairness where fairness is defined only among interfering nodes. A simple local fairness algorithm is presented that is usually restricted only to segments of interfering nodes. In another related work [17] a variation of starvation-free local mechanism is presented for a fixed size packet network.

The combination of buffer insertion ring, spatial reuse, built in reliable fairness mechanism and exploitation of the recent advent in fiber-optic technology are the basis of the MetaRing network. This network has been prototyped at our IBM T. J. Watson research lab and has been operational since fall 1989. The current prototype supports the transmission of variable size packets at 100 Mb/s link speed and with aggregate throughput of 700 Mb/s. A gigabit version of the MetaRing is currently being implemented as part of the IBM participation in the Aurora Testbed which is a part of NSF/DARPA Gigabit Networking Program [18].

The paper is structured as follows. The basic mechanisms of MetaRing and the underlying assumptions are described in Section II. The global fairness algorithm is described in Section III. Mechanisms for multiple priority levels and the integration of synchronous and asynchronous traffic are presented in Sections IV and V, respectively.

II. Basic Mechanisms

This section describes three basic aspects of the MetaRing principles of operation.

1) Physical access control—the proposed solution is a full-duplex ring with two access modes slotted and buffer insertion. Some additional discussions on buffer insertion and slotted rings can be found in [10], [11], [13], [19]–[22]. It is assumed that packets are transmitted via the shortest path and removed by their destinations.

2) Physical access name or address label—this is the "lowest level" address of a node, which is used for copying and removing packets from the ring.

3) Hardware mechanism for exchanging control messages or signals—these signals are exchanged between neighboring nodes, and each type of signal has a specific control function.

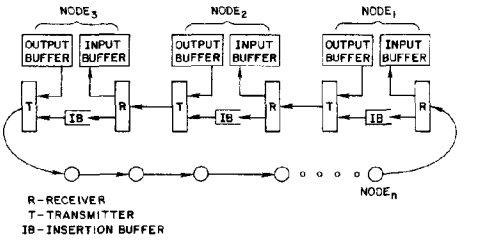

Fig. 1. Buffer insertion ring.

A. Access Control Modes

1) Buffer Insertion Mode: Buffer insertion is a random and distributed access technique to a unidirectional ring network. On the receiving side of each link, there is an insertion buffer (IB), which can store one maximal size packet, as shown in Fig. 1. A node may starts a packet transmission at any time as long as its insertion buffer is empty. If the ring traffic arrives when the node is in the middle of a packet transmission, then this traffic will be stored in the insertion buffer, until this packet transmission is completed. The node cannot transmit anymore until the insertion buffer becomes idle again, i.e., a nonpreemptive priority is given to the ring traffic. If the node is idle, the ring traffic will *cut-through* the insertion buffer. (This means that a packet does not have to be completely received before it is started to be forwarded.)

Clearly, the buffer insertion access control may enable the concurrent access or spatial reuse of the ring by more than a single node. Since the buffer insertion ring access is always permitted, unless there is ring traffic, there is no degradation in its efficiency as the bandwidth or physical size increases. All links can be kept at full utilization at all times provided nodes have enough data to transmit.

2) Slotted Mode: The motivation for a slotted mode is to minimize the insertion buffer delay. The "price" of this change is that the packet size should be fixed and some slot synchronization should be maintained. The hardware interface for the two modes is basically the same.

Initially, the ring operates in the buffer insertion mode; the nodes communicate asynchronously among themselves and perform a leader election procedure. After a leader is elected, the access mode may be changed to slotted ring. The leader generates slots which are basically empty packets of fixed size.

At the beginning of a slot, there is a *busy-bit,* if this bit is 0 the slot is empty, and if it is 1, the slot is full. A node can transmit a packet only if it receives an empty slot. The slot size and the packet size are the same. The packet is removed by the destination node, which also marks the slot as empty.

As shown in Fig. 2, fixed number of slots, say r, around the ring are maintained by the leader. The insertion buffer, in the slotted mode, functions as an elastic buffer for synchronization purposes.

3) Throughput of Ring with Spatial Reuse: We assume that the network has n nodes and is fully loaded (i.e., at all times all nodes have packets to send). Clearly, under uniform destination distribution, the maximum distance is $n/2$, and the average distance is $n/4$. Therefore, the spatial reuse is of four

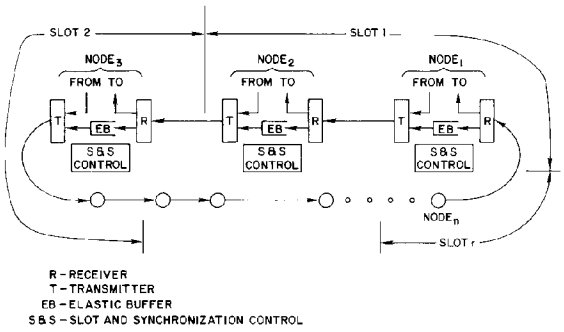

Fig. 2. Slotted ring with elastic buffer.

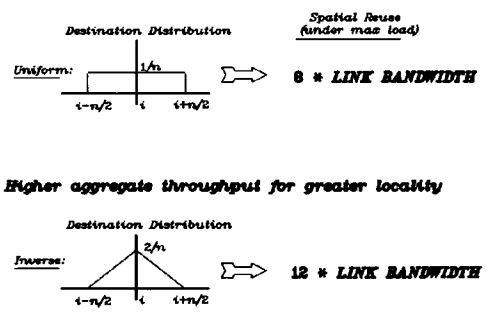

Fig. 3. Full-duplex ring throughput.

nodes transmitting at the same time, in each direction, on the average, as shown in Fig. 3. As a result, the capacity of the full-duplex buffer insertion ring is eight times a single link transmission rate, which is *4 times* more than a *dual token-ring*. If the destination distribution is inversely proportional to the distance, as shown in Fig. 3, then the average distance is $n/6$ (spatial reuse factor of 6 in each direction).

4) Problems of Access Control with Spatial Reuse: The following problems or issues are traditionally associated with buffer insertion ring.

- Starvation. The buffer insertion access control gives advantage to up-stream nodes, which can cause starvation. In Fig. 4, for example, if node 2 will transmit continuously to node 10 and if node 9 will transmit continuously to node 12, then node 11 will not be able to transmit. In Sections III, we present a solution to this problem.
- Bandwidth reservation. The problem in this case, is how to provide low delay and a guaranteed fraction of the bandwidth to some users, while still allowing *fair asynchronous distributed access* to the ring for others. This problem is treated and solved in Section V.
- Priority. The distributed nature of the basic buffer insertion access does not support multiple access priority levels. This problem is addressed in Section IV.
- Large ring transmission delay bound. Associated with buffer insertion architecture as packets may be stored and forwarded at the insertion buffers. This delay becomes negligible due to the high transmission rate. With current fiber optic speeds (100 Mb/s and above) it is possible to

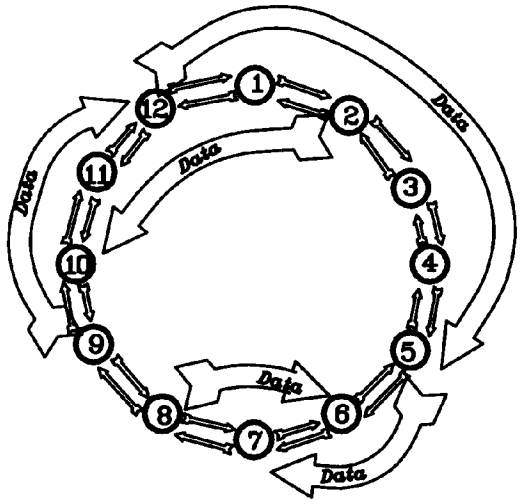

Fig. 4. Starvation on full-duplex ring.

achieve a worst case packet delay bound of less than a millisecond, which is also the propagation delay through 100 Km of fiber. At higher rates (Gb/s and up) the propagation delay dominates the insertion queueing delay even at worst case scenarios.

- "Garbage collection"—this is the problem of infinite packet looping as a result of erroneous header information. In token rings, like FDDI, this problem is solved by the node that holds the token, which removes the erroneous frames from the ring. This problem is difficult to handle in rings with spatial bandwidth reuse. Various methods can be applied to this problem. 1) A hop-count field at the header which counts the number of hops each packet has traversed. When this field reaches a limit the packet is removed from the ring. 2) The use of a monitor station (such as the leader station previously described in the slotted mode) which stamps each packet that traverses it and removes it if it encounters the packet twice. 3) The proper validation of the source and the destination fields of packets at every node before the packet is allowed to be transferred. This option requires the knowledge and maintenance of all stations' labels at every station (topology maintenance). 4) Another simple scheme, for ring and multiring networks, is based on dividing each ring into two or more disjoint physical access address or label subspaces, which breaks the ring circular symmmetry, and thus, ensures that packets will not circulate forever in the ring (see [23] for more details).

B. Physical Addresses and Routing Modes

The physical addresses of the nodes are based on short labels, e.g., 8 b, which enables fast address recognition using simple random access tables. Each node has its own unique label, but there are also group labels for multicast purposes. Some of the recovery mechanisms, which are described later, may become more efficient if the nodal labels are arranged in an increasing or decreasing order. The label assignment is part of the ring initialization and recovery procedure.

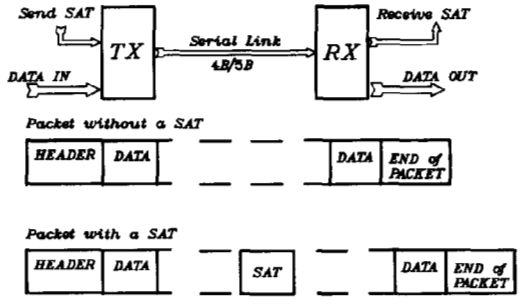

Fig. 5. The control signal transfer.

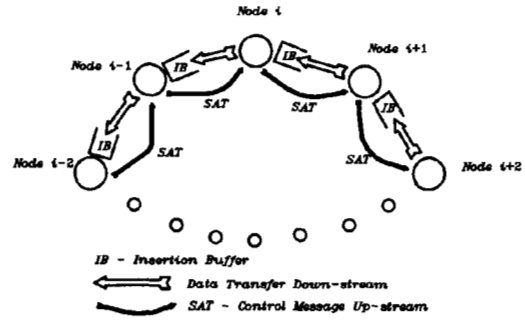

Fig. 6. The ring basic mechanism (one direction).

In our prototype we have used the short labels for routing in five different modes:

- Neighbor mode. In this mode the packet is received and removed from the ring by the first downstream node, regardless of its label. This mode can be useful for initialization and fault tolerance. It enables communication between neighbors even when nodal ID's are not known.
- Simple mode. Used to send a packet to a single destination.
- Copy mode. Used for transmitting a single packet to all the nodes between the source and the destination.
- Group mode. Used for transmitting a packet to several nodes which share the same group label. In this case, the packet is removed from the ring by its source.
- Selective-Copy or Point-to-List mode. Used for transmitting a single packet to several nodes whose labels are specified successively in the packet header. In this case, each node checks the first destination label at the packet header, and if it matches its own unique label, then the node will 1) copy the packet, and 2) remove this label from the header, and the next label becomes the first routing label in the header. In this mode, the node that detects the header delimiter will remove the packet from the ring.

C. Hardware Control Signals

The hardware control signals are used to implement critical control functions that must operate fast. The following characteristics ensure the small delay for the control signals.

- Short–few characters (possibly one).
- Preemptive resume priority–can be sent in the middle of a data packet in a way that does not damage the data packets which they preempt, as illustrated in Fig. 5. In this figure the control signal is called SAT (from the word SATisfied), and it will be used in the fairness algorithm described in the following section.

In the MetaRing, the control signals are realized by using the redundant or unused codewords in the serial transmission line. Each control signal can be followed by a predefined number of parameters, which are data words that have some special meaning and can be read and modified by each receiving node. The control signals are used for forming control channels over the same serial links. Thus, over the full-duplex ring two data channels (one for each direction) and two control channels (one for each direction) are virtually constructed. Each control channel is associated with one data channel. There are two cases as follows.

1) A control channel associated with a data channel *in the opposite direction.* In this case, the data is sent downstream and control signals are sent upstream.

2) A control channel associated with a data channel *in the same direction.*

III. Global Fairness on a Full-duplex Ring

The access to each direction of the ring is regulated by a hardware control signal, SAT, which circulates in the opposite direction to the data traffic it regulates. Circulating the SAT control signal in the opposite direction better conserves the potential spatial reuse of the full-duplex ring, as shown in Section III-G. Fig. 6 describes the basic ring mechanism for one direction. Note that the global fairness algorithm is the same for the two access modes. In the following discussion, we only describe the buffer insertion mode.

A. Informal Description of the Global Fairness

In principle, the node forwards the SAT signal upstream with no delay, unless it is not SATisfied or "starved." By "starved" we mean that the node could not send the permitted number of data units since the last time it has forwarded the SAT signal.

The node is SATisfied if between two SAT signals the node has sent at least l packets or if all packets presented in its output buffer when the previous SAT was sent upstream, were transmitted. When the node receives a SAT and it is SATisfied, it will forward the SAT upstream. If the node is not SATisfied, it will hold the SAT until it is SATisfied and then forward the SAT upstream. After a node forwards the SAT, it can send k more packets or data units, $k \geq l$ (a simple case $k = l = 1$).

B. The Global SAT Algorithm

Fig. 7 is a flow chart which describes the algorithm, and is divided into two parts, send packet and forward SAT.

1) Send Packet Algorithm: The node can transmit a packet from its output buffer when it is not empty, only if the following two conditions hold: 1) the variable COUNT is

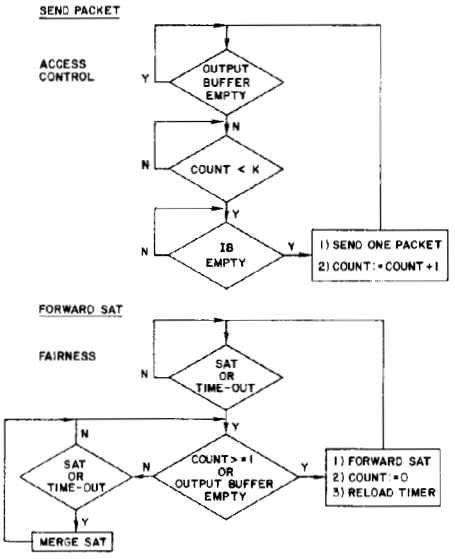

Fig. 7. The global fairness algorithm.

smaller than k, and 2) the insertion buffer is empty. After the node transmits the packet, COUNT is incremented by one.

2) Forward SAT Algorithm: This algorithm determines the actions of a node either after it receives the SAT signal, or if the SAT signal does not arrive after some maximum possible time has passed (time-out has been expired). The node will forward the SAT if the variable COUNT is greater than $l - 1$ or if its output buffer is empty. The node will hold the SAT if the variable COUNT is smaller than l and the output buffer is not empty. The node will hold the SAT until COUNT becomes l (after l packets has been transmitted). If during the time in which the node holds the SAT, another SAT arrives, the second SAT will be discarded, and if time-out occurs it will be ignored. After the node forwards the SAT, it will set the COUNT to zero and will reload the timer.

Fig. 8 illustrates the situation of having more than one SAT signal rotating in the same direction in the ring. When two SAT signals meet at the same node, the second SAT is discarded, i.e., the two SAT signals are merged. The time-out mechanism, the ability to generate and merge multiple SAT signals, are enhancements to the fairness algorithms, so they can tolerate SAT loss and duplication. This fault tolerant mechanism operates independently at any node.

C. Variations on Global Fairness

We describe two possible variations of the global fairness algorithm.

1) Average Global Fairness Algorithm: This algorithm is a generalization of the simple algorithm. In this case the number of packets the node can transmit depends on how many it already sent in previous rounds. Each time the node receives the SAT, it increments COUNT by m, as long as its value does not exceed $+k_{\max}$. During each SAT round the node can transmit if its COUNT is not less than $-k_{\min}$. A node will hold the SAT signal if its COUNT is greater than zero; otherwise, it will forward it immediately.

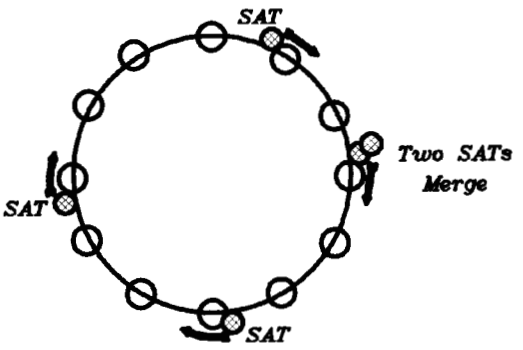

Fig. 8. Multiple SAT's merge.

2) Adaptive Fairness Algorithm: This algorithm is a more specific example of the general fairness algorithm. In this case, the values of the first and second predetermined numbers, k and l, are computed according to the current value of the parameter SAT-ROTATION-TIME. The idea is that when the SAT-ROTATION-TIME decreases l and k may be increased and vice versa. The actual function should be determined by analysis, simulation and/or experiments.

D. Properties of the Global Fairness

The following summarizes the properties of the global fairness algorithm with its fault tolerance enhancement.

Fairness Property—For a ring with a single SAT and given k and l, after each round of the SAT signal the subset of nodes with at least l packets in their output buffer will transmit at least l packets and at most k packets.

Proof: If at some round node i could not transmit l packets, then it will hold the following SAT signal until upstream becomes idle, and it is able to transmit its quota. Since the upstream nodes are not allowed to transmit more than k packets before they see the SAT again, the upstream of node i will eventually become idle.

The fairness property guarantees that all nodes have equal shares of the bandwidth, i.e., all nodes have equal rights and equal opportunities. However, the global fairness mechanism can also be implemented in an asymmetric manner, such that each node will have different quota, node i will have different l_i and k_i. This way nodes with higher traffic requirements (e.g., file servers, bridges) can get larger shares of the bandwidth.

Liveness Corollary—The SAT can not be held indefinitely by a node, i.e., the global fairness algorithm is deadlock free.

Multiple SAT Property—The buffer insertion ring will operate correctly with no starvation with multiple SAT signals.

Proof: Multiple SAT signals cause no problem to the algorithm, since a starved node that is holding a SAT signal will hold this signal as long as it cannot transmit. If another copy of a SAT signal reaches this node, it will be ignored, and by that action, it is actually merged with the first SAT signal that is being held. This mechanism will gradually merge more and more SAT signals, until there is only a single copy of the SAT signal, as was shown in Fig. 8. Since nodes do not starve, the operation is clearly fair and correct, and each time a SAT traverses a node it gives it the same quota.

Lost SAT Corollary—When a SAT is lost, after a time-out, one or more SAT signals are generated, and the SAT algorithm stabilizes itself, by merging multiple SAT signals to a single SAT signal.

E. Minimizing the Time-out Delay

The multiple SAT property ensures that the system will operate correctly in the presence of several SAT's in the ring. However, it is shown in [9] that the existence of multiple SAT's can lead to a significant increase in the necessary time-out interval, i.e., the number of SAT's can quadratically increase the time-out interval compared with the situation of having a single SAT. Although the above problem can occur only under extreme conditions, and the probability for its occurrence can be significantly reduced using randomly chosen time-out intervals (within some range), we presented in [9] a deterministic algorithm for recovery from a lost SAT, that is based on a simple election algorithm such as the one in [24].

The algorithm uses a different control signal, SAT–REC with the node's ID as a parameter. The idea is that after the time-out expired, node i sends SAT-REC(ID: = i) to its upstream neighbor. The SAT–REC(ID) only renews the nodes' quota and cannot be held by any node. A node can get only one quota even if it sees multiple SAT–REC's.

The convergence from multiple SAT–REC's to one is done by a simple leader election, only the SAT–REC with the highest ID will "survive." It is shown that this algorithm convergence time is one roundtrip propagation delay.

F. Degradation to Bus Segments Operation

In this section, the global fairness algorithm is extended in order to tolerate link failures. It is assumed that when one direction of the full-duplex link is faulty, both directions are brought down. When the ring is disconnected it is still important that the fairness mechanism will continue to operate correctly on each connected full-duplex segment.

On a full-duplex bus, the SAT signal cannot go around in circles. Therefore, in order to avoid SAT loss, when a SAT signal arrives to an edge node it will be sent back (in the opposite direction) as a different control signal, SAT′. When an edge node receives a SAT′ it will send back (in the opposite direction) a regular SAT control signal, as shown in Fig. 9. The SAT–SAT′ mechanism forms a ring on the connected bus segment, so that the fairness algorithm can continue to operate correctly.

The SAT–SAT′ mechanism is performed dynamically, i.e., the network changes from a ring to a bus and back during normal operation, whenever a link or a node fail or recover. As a result, it might happen that a SAT′ will rotate in an infinite loop over a ring. In order to prevent infinite rotation of SAT′, each node will have to detect this abnormal phenomenon. A node can detect this when it sees two successive SAT's with no SAT signal in between. This will eventually happen since SAT′ is transferred unconditionally with no delay. In this case, the SAT′ signal is discarded. A formal description of the SAT–SAT′ algorithm can be found in [9].

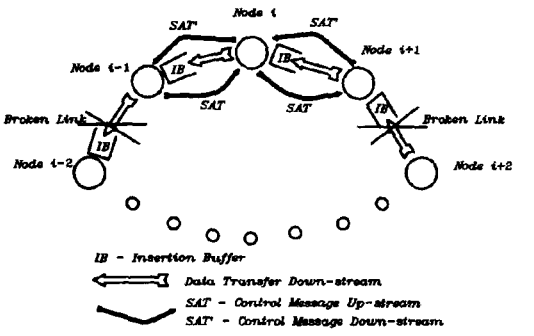

Fig. 9. SAT-SAT′ mechanism.

G. Performance Results

In this section, we present the results of a simulation model for a slotted ring operating under various modes of fairness and spatial reuse. We demonstrate the throughput advantage of using spatial reuse in the ring over ring protocols without spatial reuse. We show the advantage of operating the SAT algorithm in the opposite direction to the data it regulates versus operating it in the same direction. Finally, we show how one may improve the ring throughput by changing some parameters of the SAT algorithm.

1) Simulation Model: The ring consists of 24 nodes with equal distances between them. There are three transmission slots rotating in the ring, and we assume that at any point of time each slot covers exactly 1/3 of the ring. Nodes can begin transmission of packets only at the slot boundaries. Each node maintains an infinite queue for arriving packets. Packets are transmitted at slot boundaries and according to the SAT algorithm, if employed. Each packet is destined according to a uniform distribution (we assume shortest path routing, so the maximal distance is at most half of the ring). Each point in the curve was obtained from 100 000 simulation steps.

For delay analysis, we assume a uniform independent Bernoulli arrival process into each nodal queue, which means that any node may create at most one packet (whose length is equivalent to the slot size) at each time step. For analyzing the delay and queue sizes, we have measured the delay from the packet arrival time until the beginning of its transmission. The basic unit of delay in the system is the time it takes a transmission slot to complete a full rotation. In the delay analysis, we did not take into account the transmission and propagation delay. The average propagation delay is an additional 1/4 delay unit (6 time steps) and the transmission delay is 1/3 delay unit (8 time steps).

It is hard to obtain the exact maximum throughput through the delay analysis. Alternatively, we have operated the various systems with the assumption that each node has an unbounded number of packets to transmit. We have calculated the effective total throughput of the system under such an assumption. In order to derive numbers which are not sensitive to the slot (packet) length, we have normalized our results to the number of slots in the ring.

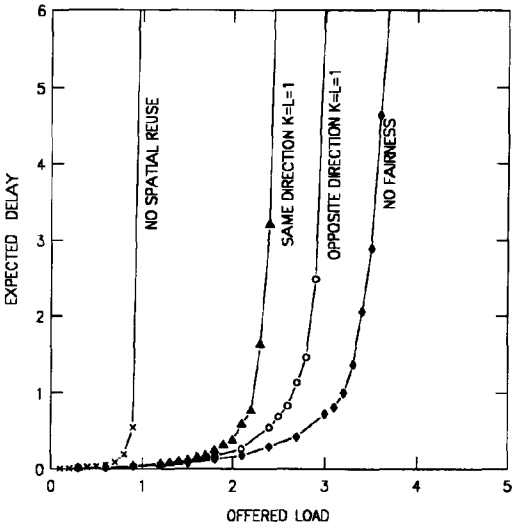

Fig. 10. Delay-throughput under different control modes.

2) Results: Fig. 10 shows a comparison of throughput delay curves between the following systems: Slotted ring with no spatial reuse (maximal throughput of 1), slotted ring with fairness algorithm ($k = l = 1$) when the SAT signal is rotating in the same direction as the data (throughput of 2.56), as before when the SAT is rotating in the opposite direction (throughput of 3.16), and a pure slotted ring with no fairness (throughput of 4). Here we clearly demonstrate the advantage of sending the SAT signal in the opposite direction to the data. The throughput gains for doing that is more than 20%. The advantage of forwarding the SAT in the opposite direction can be explained, intuitively, by observing the fact that a node is starved because of an up-stream traffic (assuming shortest path routing on a full-duplex ring). When the SAT is transferred in the same direction, the upstream nodes have just renewed their quotas, while in the opposite direction, it takes longer for the SAT to reach an unsatisfied node since the time the up-stream nodes have renewed their quotas. Therefore, when the SAT is transferred in the opposite direction, then it is more likely that it will be held for a shorter period of time, than if it is transferred in the same direction. It is also clear that if the total time the SAT is being held in every rotation gets longer the throughput will become smaller. Thus, rotation of the SAT in the opposite direction yields better performance.

Fig. 11 demonstrates the impact of the SAT algorithm parameters compared to the slotted ring with no fairness.

SLOTTED RING TYPE	THROUGHPUT
No spatial reuse	1.0
Same direction($k = l = 1$)	2.56
Opposite($k = l = 1$)	3.16
Opposite($k = l = 3$)	3.46
Opposite($k = 3, l = 1$)	3.65
Opposite($k = 5, l = 1$)	3.75
No fairness	4.0

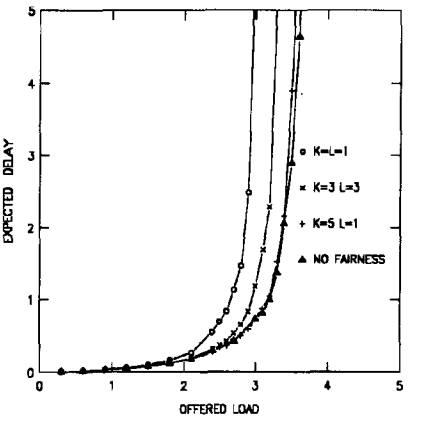

Fig. 11. Delay-throughput with different parameters.

We have omitted the simulation results for the nonuniform case due to lack of space. Recent study of a Gb/s MetaRing, in the buffer insertion mode and with variable size packets, has yielded similar performance results to those presented in this section [25].

IV. Priority for the Asynchronous Traffic

The basic fairness protocol prevents starvation of the nodal asynchronous traffic. However, the basic scheme treats all traffic at the same priority level. The implementation of priority levels among various types of asynchronous traffic is the issue of this section. The solution preserves the fairness among users which operates under the same priority level while restricting the access of lower priority traffic to the network when congestion of a higher priority traffic occurs.

A. The Algorithm Description

The basic idea is to assign a priority level to the SAT signal such that only packets with priority equal or higher to this level are allowed to be transmitted. A node can increase the priority level of the SAT if it has packets of higher priority than the current priority level of the SAT. The basic problems to be solved are how and when to decrease the value of the SAT signal to some lower priority level and to which level.

The algorithm is designed such that the last node to increase the priority level of the SAT signal remembers this fact. Thus, it can "understand" if there are no higher priority level packets in the system by getting back an SAT signal stamped with its original priority level. At this point, if the node has no more packets with the same or higher level no further increase of the SAT priority level should take place. Ideally, the level should now reflect the highest priority level that exists in the system. If no other node in the system has packets of the same level, a decrease of the SAT priority level should take place. If the node would just drop the level to some low value (e.g., zero), it might result in a temporary violation of the priority property until the SAT priority level is increased back by some other nodes. On the other hand, if the node would only slightly decrease the level (e.g., by one), it may take a long time until low priority level packets can be transmitted in the case that

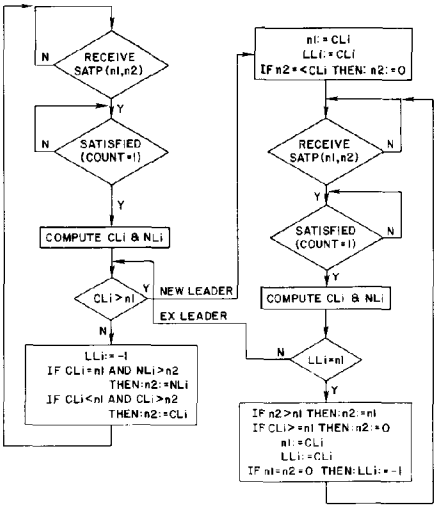

Fig. 12. Asynchronous priority flow chart.

there are no higher priority level packets in the system. In order to allow prompt priority service and fast adaptation of the levels, we use a slightly more complex algorithm.

The algorithm is based on a modified SAT signal with two parameters—SATP(n_1, n_2). The parameter n_1 reflects the highest priority level packets that are queued at some node. The parameter n_2 is generally the second highest priority (including the case $n_2 = n_1$), and is the candidate for the next level in which the system should operate.

In order to facilitate the presentation, we assume that, a single SATP(n_1, n_2) signal exists in the system and that the fairness algorithm used is the global fairness with $k = l = 1$.

Variables at node i:

CL_i—the current priority level of node i as it is calculated by the function LEVEL;

NL_i—the next priority level of node i as it is calculated by the function LEVEL;

LL_i—the leadership level of node i (the default is $LL_i = -1$).

When a node receives the SATP(n_1, n_2) signal and it is in the normal mode, it considers only packets with priority level equal or higher than max $\{n_1, n_2\}$. If it has such a packet that it was not able transmit since the last visit of the SATP signal (meaning that the node has a starvation situation of this level), it will hold the SATP until the packet is released. Then, the node will calculate its highest packet's priority level (CL_i) and it's second highest packet's priority level (NL_i) in order to modify the SATP(n_1, n_2) signal, if necessary.

A node increases n_1 if its CL_i is higher than the received n_1. At this point, the node is becoming the leader of this priority level and set n_2 to zero. The leader will set its leader level; LL_i, to CL_i. If this is not the case (i.e., $CL_i \leq n_1$) but the node anticipates (through the value of CL_i and NL_i) that it has requirements higher than n_2 for the next round, it will modify n_2 to that value. We distinguish between two cases.

1) If $CL_i = n_1$ and $NL_i > n_2$ then: n_2 is set to NL_i, since it is clear that the packet with priority level CL_i can be transmitted at the current SATP round.

2) If $n_2 < CL_i < n_1$, since the node is not allowed to transmit this packet at this round, it sets $n_2 = CL_i$.

The introduction of n_2 allows the priority level leader to decrement n_1 to the next highest priority level, even if this priority level is existing at some other node. (The exact procedure of updating these numbers (n_1 and n_2) is formally described in Fig. 12.) Only the leader can decrement both n_1 and n_2. To prevent a false decrement, the leader will not decrement n_2 below the value of the received n_1. This assures that other nodes with higher priority level will get a chance to capture the leadership before the actual priority level is further decreased. Fig. 12 is the flow chart of the priority fairness algorithm with $k = l = 1$.

Node i can transmit its quota, under the buffer insertion access control, if $CL_i \geq \max\{n_1, n_2\}$. After the node transmits its quota it sets the COUNT variable to one, i.e., the node is satisfied.

The node computes the CL_i and NL_i by the following function level:

- Start;
- CL_i: = priority level of highest priority packet in the queue;
- If no packets in the queue then: $CL_i := 0$;
- NL_i: = priority level of thesecond highest priority packet in the queue;
- If no other packets in queue then: $NL_i := 0$;
- End;

An execution example for the priority algorithm can be found in [9].

B. Discussion and Possible Extensions

The priority scheme presented in this section is distributed and asynchronous. The priority algorithm can be extended in several directions.

- The priority signal can be decoupled from the SAT and may rotate just to mark the priority level of the operation.
- In order to detect that some leader has failed a protection mechanism like the flipping of a bit in each round in some predefined pattern, can be part of the leader algorithm. The detection of violation in the leader algorithm at a node will force the SATP to be marked as (0, 0). This approach can also protect the system against multiple SATP signals, along with the merge of such duplicates at a node.
- When SATP is lost and time-out has been occurred, it is possible to use the same recovery algorithm as described in Section III-E for SAT–REC. In this case, after a leader is elected it will generate a new SATP$(n1, n2)$ and forward it around the ring.

V. Integration of Synchronous and Asynchronous Traffic

This section describes a mechanism for integrating two types of traffic over the full-duplex ring: 1) synchronous or real-time traffic which is periodic and requires a guaranteed bandwidth, and 2) asynchronous traffic with no real-time constraints that can use the remainder of the bandwidth in a bursty manner.

The following integration mechanism is functionally equivalent to the TIMED–TOKEN protocol in FDDI [3], [4]. This protocol together with the asynchronous fairness, which was previously described, still maintains fairness with spatial reuse for the asynchronous traffic at each round where a round is determined as a single rotation of a control signal around the ring. Note that the asynchronous fairness in FDDI is achieved only after many rounds of the TIMED–TOKEN, since in each rotation of the token only subset of the nodes can transmit asynchronously.

The mechanism is based on two control signals SAT and ASYNC–EN which circulate in the opposite direction to the traffic they regulate. The ASYNC–EN is used for enabling the integration of the asynchronous traffic and the SAT is used for ensuring fairness of the asynchronous traffic, as discussed in Section III.

A. Distributed Reservation and Synchronous Access

The objective of the distributed reservation is to guarantee bandwidth and bounded maximum delay for transferring packets over the ring. For the reservation mechanism we assume the following.

1) T_c—is the periodic time cycle of synchronous data transfers (in seconds).

2) BW—the data transmission rate (in bits per second).

3) p—the basic data units (in bits); in the slotted mode this is the slot duration in bit periods. The size of each data packet is dp bits where $d \geq 1$.

4) c—is the number of data units that can be transmitted over each serial link in every time cycle, where $c = (T_c BW/p)$.

5) ρ—the maximum fraction of synchronous traffic ($0 \leq \rho < 1$).

When a node tries to reserve bandwidth for real-time transmission, it performs the following protocol.

1) It computes how many data units it needs, say l.

2) It computes the route or the transmission direction; the route determines the reservation path.

3) It sends a reservation request for l data units to all nodes along its reservation path. This accelerates the reservation part and reduces the probability of conflicts.

4) If affirmative acknowledgments are received from all the nodes along the reservation path, this path becomes effective, else, the node sends a release request of l data units to all nodes along this reservation path.

Each node maintains a variable RESERVE for each of its links, which indicates how many data units have been reserved. At all times RESERVE is less than ρc.

- When a node receives a reservation request for l data units and if RESERVE + l < ρc, then RESERVE: = RESERVE + l and a positive acknowledgment is returned, else RESERVE: = RESERVE + l and a negative acknowledgment is returned.
- When a node receives a release request for l data units then: RESERVE: = RESERVE $-l$.

After the reservation is completed successfully, the reserved traffic is transmitted before the asynchronous traffic. The reserved traffic will be queued, if the link is busy, in the SYNC–QUEUE.

B. The ASYNC-EN Rotation Protocol

The ASYNC-EN (asynchronous enable) control signal is used for realizing a rotation timer on each ring interface (each direction has a separate identical mechanism). The time elapses between two successive arrivals of the ASYNC-EN signal is measured by the ASYNC-EN-TIMER(t), and its current value is stored in the ASYNC-EN-ROTATION-TIME variable. For example, each time the node receives the ASYNC-EN signal

$$\text{ASYNC-EN-ROTATION-TIME:} = \text{ASYNC-EN- TIMER}(t),$$

and

$$\text{ASYNC-EN-TIMER}(t): = 0.$$

Under normal condition the ASYNC-EN rotates around the ring freely, i.e., each node will forward the ASYNC-EN immediately after receiving it. As a result, the rotation time of this signal is the propagation delay around the ring. A node can hold the ASYNC-EN only if it starts to accumulate synchronous messages in its SYNC-QUEUE. By holding the ASYNC-EN the node indirectly signals to upstream nodes not to send asynchronous traffic.

C. Integration of the Asynchronous Traffic

The following algorithm is for the slotted mode access control, with T_s-slot duration and r slots around the ring. Therefore, under free running condition the ASYNC-EN completes a rotation around the ring, every r time slots. By using the ASYNC-EN-TIMER(t) the node determines when it can send asynchronous packets.

The objective of the integration algorithm is on one hand to maximize the potential asynchronous traffic, and on the other hand to minimize the synchronous traffic delay. Two algorithms are defined; one determines when a node can send asynchronous traffic (ASYNC-EN-TIMER(t) threshold), and the other determines when a node with synchronous traffic in its SYNC-QUEUE will hold the ASYNC-EN signal.

Asynchronous integration algorithm:

- A node can transmit asynchronous packets if:

1) it is not satisfied (based on the previous SAT algorithm),
2) it sees an empty slot, and
3) ASYNC-EN-TIMER(t) $\leq r$.

Hold ASYNC-EN algorithm:

- This algorithm determines the SYNC-QUEUE threshold for signaling to the nodes on the ring to stop sending their asynchronous traffic. Let l_i^k be the maximum integer number of packets node i can receive every r time slot (assuming periodic synchronous arrival rate).
- Node i will hold the ASYNC-EN for one time slot if the number of packets in its SYNC-QUEUE is greater than l_i^k and ASYNC-EN-ROTATION-TIME is r (if it is higher, it means that another node has already held the ASYNC-EN signal for one time slot).

VI. Discussion and Conclusions

In this work we have shown how to design a ring network with spatial reuse, while maintaining the functionality and the simplicity of existing ring and bus designs. The MetaRing uses the combination of ring, global fairness algorithm, and additional extensions for fault tolerance, priority handling, routing and real-time traffic support. The solutions presented are suitable for many applications and environments. It ranges from connecting a cluster of high-speed machines, to large local and metropolitan area networks.

The MetaRing architecture unifies, in a simple manner, all the essential LAN properties.

- Immediate or random access under light load, as in *Ethernet and DQDB.*
- Single node can almost fully load the ring, as in *Token-rings and DQDB.*
- Fairness and asynchronous priority levels, as in *IBM Token-ring.*
- Integration of synchronous and asynchronous traffic, as in *FDDI,* but with a stronger fairness property.
- Transmission of variable size packets, in the buffer insertion mode, as in *Ethernet and Token-rings.*
- Minimum Propagation Delay:
  1. For the slotted mode—the delay through the insertion/elastic buffer is minimized.
  2. Logical addressing—the address decoding along the ring is done on one byte, therefore, the cut-through and the table look up delays are minimized.
- Fault Tolerant—the set of solutions presented in this work can be easily extended to operate independently and correctly on every connected segment of the ring.
- Cost Effectiveness—the implementation of this architecture does not require new technology. The design complexity and the level of technology of the MetaRing are the same as token rings (e.g., FDDI), and its performance are better and more reliable. Thus, this solution has much better cost effectiveness characteristics than token rings (i.e., "you get much more for the same amount of money").

Acknowledgment

The authors would like to thank J. Janniello, S. Hai, and S. Shmueli for their contributions to the MetaRing implementation.

References

[1] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Commun. ACM,* vol. 19, no. 7, July 1976.
[2] W. Bux, F. H. Closs, K. Kummerle, H. J. Keller, and H. R. Mueller, "Architecture and design of a reliable token-ring network," *IEEE J. Select. Areas Commun.,* vol. SAC-1, no. 5, pp. 756–765, Nov. 1983.
[3] W. E. Burr, "The FDDI optical data link," *IEEE Commun. Mag.,* vol. 24, no. 5, pp. 18–23, May 1986.
[4] F. E. Ross, "FDDI—A tutorial," *IEEE Commun. Mag.,* vol. 24, no. 5, pp. 10–17, May 1986.
[5] Z. L. Budrikis *et al.,* "QPSX: A queue packet and synchronous circuit exchange," in *Proc. ICCC'86,* 1986, pp. 288–293.
[6] R. M. Newman and J. L. Hullet, "Distributed queueing: A fast and efficient packet access protocol for QPSX," in *Proc. ICCC'86,* 1986, pp. 294–299.
[7] J. L. Hullet and P. Evans, "New proposal extends the reach of metro area nets," *Data Commun.,* pp. 139–147, Feb. 1988.
[8] I. Cidon and Y. Ofek, "Fairness algorithm for full-duplex buffer insertion ring," *U.S. Pat.* 4926418, 1989.
[9] ——, "MetaRing—A full-duplex ring with fairness and spatial reuse," *IBM Res. Rep.,* no. RC 14961, Sept. 1989.
[10] A. A. Lazar, A. T. Temple, and R. Gidron, "MAGNET II: A metropolitan area network based on asynchronous time sharing," *IEEE J. Select. Areas Commun.,* vol. SAC-8, pp. 1582–1594, Oct. 1990.
[11] R. M. Falconer and J. L. Adams, "Orwell: A protocol for an integrated services local network," *Brit. Telecom Technol. J.,* vol. 3, no. 4, pp. 27–35, Oct. 1985.
[12] E. R. Hafner, Z. Nenadal, and M. Tschanz, "Integrated local communications—Principles and realization," *Hasler Rev.,* vol. 8, no. 2, pp. 34–43, 1975.
[13] M. T. Liu and D. M. Rouse, "A study of ring networks," in *Proc. IFIP WG6.4, Univ. Kent Workshop Ring Technol. Based Local Area Networks,* Sept. 1983, pp. 1–39.
[14] H. Ohnishi, N. Morita, and S. Suzuki, "ATM ring protocol and performance," in *Proc. ICC'89,* 1989, pp. 394–398.
[15] P. Heinzmann, H. R. Muller, D. A. Pitt, and H. R. van As, "Buffer-insertion cell-synchronized multiple access (BCMA) on a slotted ring," in *2nd Int. Conf. Local Commun. Syst.: LAN PBX,* Palma Blearic Islands, Spain, June 1991.
[16] I. Cidon and Y. Ofek, "Distributed fairness algorithm for local area networks with concurrent transmissions," in *Proc. 3rd Int. Workshop Distribut. Algorithms,* Sept. 1989, pp. 57–69.
[17] R. Simha and Y. Ofek, "A starvation-free access protocol for a full-duplex buffer insertion ring local area network," *Comput. Networks ISDN Syst.,* vol. 21, no. 2, pp. 109–120, Apr. 1991.
[18] Special Report, "Gigabit network testbeds," *IEEE Comput. Mag.,* vol. 23, pp. 77–80, 1990.
[19] W. Bux and M. Schlatter, "An approximate method for the performance analysis of buffer insertion rings," *IEEE Trans. Commun.,* vol. COM-31, pp. 50–55, Jan. 1983.
[20] A. Hopper and R. M. Needham, "The Cambridge fast ring networking system," *IEEE Trans. Comput.,* vol. 37, no. 10, pp. 1214–1223, Oct. 1988.
[21] D. E. Hubber, W. Steinlin, and P. J. Wild, "SILK: An implementation of a buffer insertion ring," *IEEE J. Select. Areas Commun.,* vol. SAC-1, pp. 766–774, Nov. 1983.

[22] T. Minami *et al.*, "A 200 Mbit/s synchronous TDM loop optical LAN suitable for multiservice integration," *IEEE J. Select. Areas Commun.*, vol. SAC-3, pp. 849–858, Nov. 1985.
[23] R. Cohen, Y. Ofek, and A. Segall, "A new label based source-routing in multi-ring networks," in *3rd Int. Workshop Protocols High-Speed Networks (IFIP WG6.1/WG6.4)*, IBM Res. Rep. 1992.
[24] E. Chang and R. Roberts, "An improved algorithm for decentralized extrema-finding in circular configurations of processes," *Commun. ACM*, vol. 22, no. 5, pp. 281–283, May 1979.
[25] J. Chen, H. Ahmadi, and Y. Ofek, "Performance study of a Gb/s MetaRing," in *16th Conf. Local Comput. Networks*, 1991, pp. 136–147.

Israel Cidon (M'85–SM'90) received the B.Sc. (summa cum laude) and the D.Sc. degrees from the Technion—Israel Institute of Technology in 1980 and 1984, respectively, both in electrical engineering.

From 1980 to 1984, he was a Teaching Assistant and a Teaching Instructor at the Technion. From 1984 to 1985 he was on the faculty with the Electrical Engineering Department at the Technion. In 1985 he joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he has been a Research Staff Member and a manager of the Network Architectures & Algorithms group involved in various broadband networking projects. Since 1990 he is with the Department of Electrical Engineering at the Technion.

Dr. Cidon received the IBM Outstanding Innovation Award for his work on the PARIS high-speed network in 1989. He was a Guest Editor for *Algorithmica* and member of the program committee for INFOCOM'91. Currently, he serves as the Editor for Network Algorithms in the IEEE TRANSACTIONS ON COMMUNICATIONS. His current research interests are in high speed local and wide area networks and distributed algorithms.

Yoram Ofek received the B.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology in 1979, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Illinois-Urbana in 1985 and 1987, respectively.

From 1979 to 1982 he was affiliated with RAFAEL, as a research engineer. During 1983–1984 he was a Fermi National Accelerator Laboratory, Batavia, IL. Since 1987 he is a research staff member at the IBM T. J. Watson Research Center, Yorktown Heights, NY. His main research interests are access-control, routing, flow-control and fairness in local and wide area networks, high-speed optical networks, distributed algorithms and systems, clock synchronization, self-stabilization and fault tolerance.

Dr. Ofek is the program co-chairperson of the Sixth IEEE Workshop on Local and Metropolitan Area Networks. He was on the program committee for INFOCOM '93 and a Guest Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He initiated and has been the leading research activities on the MetaRing and MetaNet architectures.