A Local Fairness Algorithm for Gigabit LAN's/MAN's with Spatial Reuse

Jeane S.-C. Chen, Israel Cidon, *Senior Member, IEEE,* and Yoram Ofek

***Abstract*—In this paper, we present an algorithm to provide local fairness for ring and bus networks with spatial bandwidth reuse. Spatial bandwidth reuse can significantly increase the effective throughput delivered by the network and is, therefore, desirable to be implemented in high-speed LAN/MAN environments. However, spatial bandwidth reuse can result in unfair access among nodes in the network and, thus, a fairness algorithm is needed to regulate the access to the network. A local fairness algorithm views the network as multiplicity of communication resources as opposed to a global fairness algorithm, which views the network as a single communication resource. Our algorithm can be applied to any dual ring or bus architecture such as MetaRing [1], [4]. In the dual bus configuration, when transporting ATM cells, the local fairness algorithm can be implemented using two generic flow control (GFC) bits in the ATM cell header.**

In the performance study section of our paper, we will show that this local fairness algorithm can exploit the throughput advantage offered by spatial bandwidth reuse better than a global fairness algorithm. This is accomplished because it ensures fair use of network resources among nodes which are competing for the same subset of links, while permitting free access to noncongested parts of the network. We will demonstrate the performance advantage of our local fairness scheme by simulating the system under various traffic scenarios and compare the results to that of the MetaRing SAT-based global fairness algorithm. Furthermore, we will show that under certain traffic patterns, the performance of this algorithm achieves the optimal throughput result predicted by the known Max–Min fairness definition [7].

I. Introduction

The trend toward high-speed communication has brought renewed interest in LAN/MAN architectures with spatial bandwidth reuse because of the potential high throughputs that can be delivered by these architectures. In networks with spatial bandwidth reuse packets or cells are removed from the network by their destinations. Access methods with spatial bandwidth reuse can be easily implemented using a buffer insertion or slotted ring or dual bus techniques [8], [11]. In these schemes, a node can transmit a packet at any time as long as its insertion buffer is empty or if it observes an empty slot, namely when no transmission of upstream users to downstream ones has been detected. This local access decision, together with destination packet removal, enables multiple transactions to be carried on distinct segments of the network concurrently and, therefore, significantly increase the effective throughput of the network. By a simple observation, one may realize that when the traffic pattern is homogeneous (uniform), a factor of 2 can be gained in a unidirectional ring structure by introducing spatial bandwidth reuse. (The average distance of a path is half of the ring length.) 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 bandwidth reuse is of four nodes transmitting at the same time (on each direction).

However, because priority is given to upstream traffic in these access schemes, a problem known as "starvation" can happen when some nodes are constantly covered by upstream traffic and not able to access the network for a very long period of time. In order to architect high-performance Gb/s LAN's/MAN's with spatial bandwidth reuse, it is desirable to provide fairness solutions to regulate access in these networks.

Previously, this starvation problem was solved by applying global fairness algorithms. Existing mechanisms introduced in MAGNET [10], Orwell [6], and ATMR [14] are all global fairness algorithms. These algorithms use some "stop-and-go" mechanisms to control the access and have the drawback of being sensitive to large propagation delays. A more efficient and robust global fairness algorithm was introduced in the MetaRing architecture [4]. In this architecture, a control signal called SAT is used to regulate the access in a continuous fashion. In addition, the MetaRing architecture includes five addressing modes, integration of synchronous and asychronous traffic, and multi-ring extensions [4], [5], [12], [13], [17]. The MetaRing architecture, with 100 Mb/s link speed and with aggregate throughput of 700 Mb/s, was prototyped at our IBM T. J. Watson Research Laboratory in 1989. A Gb/s buffer insertion ring called ORBIT, which incorporates many of the early MetaRing ideas has been implemented [2]. A key feature of ORBIT is its "seamless" interoperation with the plaNET wide area network. ORBIT will be deployed in several field trials, including the AURORA testbed that is part of the NSF/DARPA Gigabit Networking Program.

However, all these global fairness algorithms regulate the access to the network by considering the network as a single communication resource. Therefore, they cannot fully utilize the throughput advantages offered by spatial bandwidth reuse, especially under nonuniform traffic conditions.

This work introduces a fault-tolerant local fairness algorithm that regulates the access to these networks and solves the fairness problem with a minimal impact on the network efficiency. The fault recovery event can be triggered by a timeout at any node. The timeout event has a tight bound

Manuscript received June 1992; revised December 1992.

J. Chen and Y. Ofek are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

I. Cidon is with the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa 32000 Israel.

IEEE Log Number 9211670.

0733-8716/93$03.00 © 1993 IEEE

of O (n) (n is the number of nodes), which constitutes a fast recovery procedure. Our earlier work on local fairness [3] did not consider the fault-tolerance issue and, therefore, does not provide the proper time bounds needed for a timeout driven fault-tolerant mechanism and the timeout event had a bound of $O(n^2)$.

A local fairness mechanism views the network as a distributed collection of communication resources and not as a single resource, as in global fairness. The local fairness mechanism is triggered locally, at an arbitrary time, only if potential starvation exists. It regulates the transmissions of the interfering nodes without affecting others.

The distributed local fairness algorithm that is executed on each node alternates between two modes of operation: i) nonrestricted mode, in which a node can transmit at any time by observing the basic access protocol; and ii) restricted mode, in which a node can transmit only a predefined quota of data units (either as packets or cells) before it transits back to the nonrestricted mode. Normally, each node is operating in the nonrestricted mode. When a node detects starvation, it activates some control mechanism which transits itself and some nodes upstream into restricted mode. This restricted mode is later terminated when the access conflict is resolved, and all nodes involved are satisfied.

Two control signals are needed to toggle each node between these two modes of operation: i) **REQ** for initiating the restricted period of operation, and is forwarded upstream over the congested segment of the network; and ii) **GNT** for indicating that the access conflict has been resolved and for terminating the restricted mode of operation. As it will be shown, the REQ/GNT signals facilitate local fairness cycles over the congested parts of the network.

The locality of the fairness cycles provide the algorithm with a wide dynamic range of operation. On one hand, under certain traffic patterns the performance of this algorithm achieves the optimal result predicted by the Max–Min fairness definition described in [7], [9]. On the other hand, under worst-case traffic pattern, the local fairness degenerates spontaneously to the MetaRing's global fairness mechanism [4], and we show that in this case there is an equivalent relation between the local and global fairness mechanisms.

The organization of this paper is as follows. In Section II, we present the basic system configuration and principles. The local fairness algorithms for dual bus and dual ring are presented and their properties are discussed in Section III. A performance study which compares optimal, local, and global fairness algorithms is presented in Section IV.

II. Basic Configurations and Principles

In this section, we present three aspects of the system's configuration and principles: i) topology, ii) access control, and iii) control signalling.

A. Dual Ring or Dual Bus Topologies

The local fairness algorithm presented in this work can be implemented on dual ring and dual bus topologies. Packets can be transmitted in either direction (usually according to a shortest path routing rule), while the control signals for each data flow are sent on the opposite direction. The arrangement of opposite directions for data and control paths is necessary because starvation is caused by the hogging of upstream traffic and, therefore, control signals should be sent upstream to the source of contention. In these topologies, each node in the network will independently execute two local fairness algorithms—one for each direction.

B. Access Control with Spatial Bandwidth Reuse

The system can operate under two basic access control modes: buffer insertion for variable-size frame or slotted for fixed size cells. In both modes, the frames or cells are removed by their destinations using short identity (ID) labels (e.g., 8 bits), so the node can determine very fast whether or not to remove a frame or a cell from the network.

Buffer insertion is a distributed access technique. On the receiving side of each link, there is an insertion buffer (IB) which can store at least one maximal size frame. A node may start a frame transmission at any time as long as its insertion buffer is empty. If traffic arrives when the node is in the middle of transmission, it will be stored in the insertion buffer until this frame transmission is completed. The node cannot transmit anymore until its insertion buffer becomes idle again, i.e., a nonpreemptive priority is given to the ring traffic.

When operated in slotted mode, at the beginning of each slot, there is a busy bit. If this bit is 0, the slot is empty; if it is 1, the slot is full. A node can transmit a cell only if it receives an empty slot. The cell is removed by the destination node, and then the slot becomes empty. The motivation for a slotted mode is to minimize insertion buffer delay.

C. REQ and GNT Control Signaling

The local fairness algorithm uses two control signals: REQ and GNT. We present two transfer methods for these signals which depend on whether the transmission is a variable-size frame or fixed-size cells. Note that the frame transmission is possible only in the buffer insertion mode, while cell transmission is possible in either buffer insertion or slotted mode.

When variable-size frames are transmitted in the buffer insertion mode, independent hardware control signals are used [4]. These signals have the following characteristics: i) each signal is implemented by a redundant codeword in the serial bit stream; and ii) each hardware control signal has a preemptive resume priority, such that it can be sent in the middle of a data frame in a way that does not damage the data frame which it preempts.

When fixed-size cells are transmitted in buffer insertion or slotted mode, the two control signals are implemented by using two bits in a predefined position in the cell header. In the ATM networking environment, we can use two out of the four bits in the generic flow control (GFC) field in the ATM cell header, which has been designated for this purpose.

III. The Local Fairness Algorithm

First, we will present the motivations for developing a local fairness algorithms for dual ring and dual bus networks.

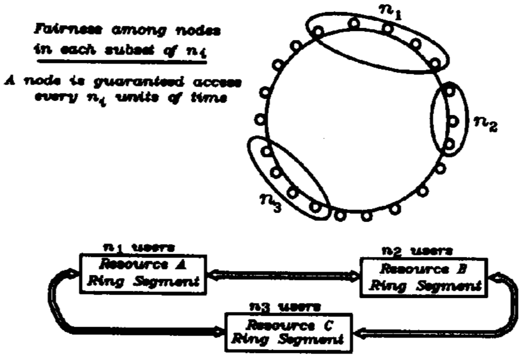


Fig. 1. Local resources on a ring.

Then, we will present a local fairness algorithm for dual bus, followed by the more complex dual ring algorithm. The main difference between the two algorithms is that, in the ring version, it is necessary to include an ID as a parameter of the REQ control signal in order to break the ring's circular symmetry.

A. Motivations

Fairness is conventionally defined in a global way. A global fairness algorithm regulates the access to a network by viewing the whole network as a single resource. Because of this, it has the following two basic characteristics that become drawbacks in networks with spatial bandwidth reuse.

1) It is GLOBAL, i.e., every node sees the same transmission constraints.

2) It is CONTINUOUS, i.e., it operates even if there is no starvation.

The first drawback, being global, is demonstrated in Fig. 1. In this example, there are three independent subsets of users that communicate only among themselves. A reasonable approach is to provide fairness within each subset while maintaining the maximal achievable throughput in each of the subsets. A global fairness algorithm will force all groups to maintain fairness (the same maximal throughput) among themselves, even if they do not interfere at all. The second drawback, being continuous (in time), means that the fairness mechanism is operational even if no node starves. This may result in some unnecessary performance degradation.

These two drawbacks motivate the development of an event-driven, as opposed to continuous, local fairness algorithm that is initiated only when starvation occurs. In addition, the algorithm should only involve segments of interfering, as opposed to global, nodes. In the example of Fig. 1, the local algorithm will be executed independently among the three subsets with no interference or message exchange. Under worst-case load scenarios, the local fairness algorithm can "degenerate" to the global fairness algorithm [4] as will be shown in Section III-E.

B. Local Fairness for Dual Bus Networks

Our local fairness algorithm distinguishes between two basic modes of operation, as shown in Fig. 2. They are:

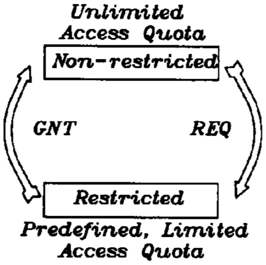


Fig. 2. The local fairness cycle.

- **Nonrestricted Mode:** Nodes can transmit at any time as long as the buffer insertion or slotted protocol permits it (priority to upstream traffic). This mode is identified by a single Free Access (FA) state.
- **Restriced Mode:** Nodes can transmit only a predefined quota of data units (either as frames or cells) before they transmit back to the nonrestricted mode.

Nodes in the nonrestricted mode are not involved in any control signal exchange. However, they may asynchronously trigger the operation of the fairness mechanism upon starvation.

The algorithm uses two types of control signals to facilitate the transition between these two operation modes, and they are:

- **REQ:** This signal initiates the restricted period of operation and is forwarded upstream over the congested segment of the bus.
- **GNT:** This signal is used, when the node is satisfied, to terminate the local fairness cycle.

The two control signals create local fairness cycles over congested segments of the bus. In each cycle, each node can transmit only a predefined quota while it is in the restricted mode, and can have free access (FA) while it is in the nonrestricted mode, as shown in Fig. 2. We note that if the bus is congested, the time interval a node is in the nonrestricted mode can be zero. In this case, a node will transmit one quota in every local fairness cycle.

Fig. 3 demonstrates the basic operation of the algorithm for a buffer insertion access protocol over a dual bus network. Here we assume that only a single node initiates the algorithm, and that there is at least one node upstream to it which has no upstream traffic. A starved node triggers the operation by sending the REQ signal upstream and entering the **tail** (T) state. Upon reception of such a signal, a node enters the restricted mode of operation and, if its upstream is idle, it will enter the **head** (H) state. If this node cannot provide silence (it senses traffic from upstream), it will forward the REQ upstream and enter the **body** (B) state. Upon satisfaction, i.e., transmission of a certain predefined quota, the tail node sends a GNT signal upstream and transits back to the nonrestricted free access (FA) state. Upon receiving this GNT, the node upstream follows similar rules: If it is in the body state, it transits to a tail (T) state and will similarly forward GNT upon satisfaction. If it is in the head state, the local cycle on this segment of the bus is terminated. In this scenario, the algorithm has created

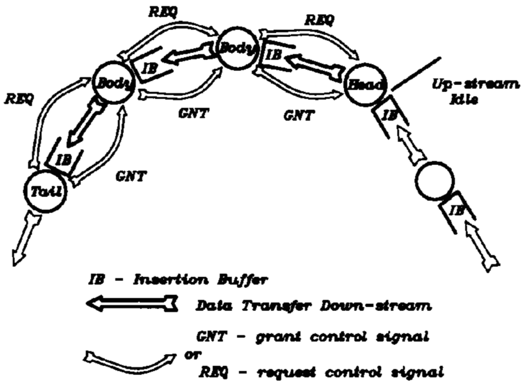


Fig. 3. Local fairness mechanism on a dual bus.

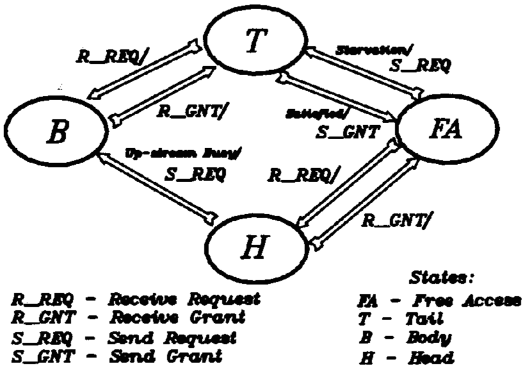


Fig. 4. Bus local fairness—State transition diagram.

a REQUEST PATH which contains unique and distinct head and tail nodes. Each node of the REQUEST PATH is able to transmit the same quota.

The actual algorithm is a bit more complex. Since there might be multiple initiators of the fairness algorithm, we should either merge or sequence these distinct REQUEST PATH's once they overlap. In this work, we merge the REQUEST PATH's since they provide a linear timeout bound at a node for the release of the REQUEST PATH. (In our previous work [3], we did not merge the REQUEST PATH's, which could resblt in a quadratic timeout bound.)

Fig. 4 shows an event-driven state transition diagram of the local fairness algorithm on a dual bus network. The writing on the transition arcs have the following form: **event/action,** indicating the event that causes the transition and the specific action that should be taken (possibly none). The algorithm has only four states with the following transition conditions.

1. **Starvation,** which is true if a node in the nonrestricted mode (FA state) has something to transmit, but could not access the network because its upstream link is continuously busy. As a result, this node will send a REQ signal (S_REQ) to its upstream neighbor and change its state to T.
2. **R_REQ** (receive request), which causes a node in the T state to change its state to B (effectively merging two REQUEST PATH's) and causes a node in the FA state to change its state to H.
3. **Upstream Busy,** which causes a node in the H state to forward a REQ signal (S_REQ) to its upstream neighbor and change its state to B.
4. **Satisfied,** which is true if a node in the restricted mode has transmitted its predefined quota of cells or bytes. If the node is satisfied in the T state, it will forward a GNT signal (S_GNT) to its upstream neighbor and change its state to FA.
5. **R_GNT** (receive grant), which causes a node in the B state to change its state to T and a node in the H state to change its state to FA.

C. Local Fairness for Dual Ring Networks

One additional problem needed to be solved when we apply the dual-bus local fairness algorithm to dual ring networks is deadlock. Deadlock will occur when single or multiple REQUEST PATH's have covered the whole ring and all the nodes are in the B state with no node in the T state. Since only the node in the T state can send GNT upstream (when it is satisfied), the ring will be in deadlock.

In order to break the ring's symmetry and to solve the deadlock problem, each node maintains a variable, REQ_ID, which identifies the original tail node of the request path. The REQ_ID is sent as a parameter of the request control signal in the following format: REQ(REQ_ID). With the help of REQ_ID, two REQUEST PATH's can be merged as follows when they overlap: i) when a node receives REQ(j) and its REQ_ID $\neq j$, it will merge the REQUEST PATH's and transit to body (B) state (unless it is already in this state); and ii) when $j >$ REQ_ID, it will forward upstream the REQ(j) and will assign a new value, j, to its REQ _ID variable.

Basically, we perform a simple election algorithm to ensure that each REQUEST PATH has a single tail and a single head. If $j =$ REQ_ID, it is clear that the same REQUEST PATH covers the whole ring and no head is currently present. In this case, the tail or body node transits to the combined head–tail (HT) state. The formal description of the local fairness algorithm for a node on the ring is depicted in Fig. 5, in terms of an event-driven finite-state machine.

Notations:

- The node's current state: FA, T, H, B, or HT.
- REQ_ID—the REQUEST PATH ID, which has been given by the initial tail node of this path and is a combination of this node ID and a sequence number.
- R_REQ(j)—receive the request control signal with the value j.
- R_GNT—receive the grant control signal.
- S_REQ(REQ_ID)—send the request control signal with the value of the REQUEST PATH ID.
- S_GNT—send the grant control signal.
- $T_{\max}$—the maximum time a node can be in restricted mode before it forwards a GNT signal. This also determines the maximum rotation time for a request signal around the ring, i.e., if a node has forwarded a request signal, then after at most $T_{\max}$ this signal will either get

State	Event	Action	Next State
T1) FA	Starvation	REQ_ID:=I and S_REQ(REQ_ID)	T
T2) FA	R_GNT	None	FA
T3) FA	R_REQ(j)	REQ_ID:=j	H
T4) T	R_GNT	None	T
T5) T	R_REQ(j) and j=REQ_ID	None	HT
T6) T	R_REQ(j) and j≠REQ_ID	If j>REQ_ID, then REQ_ID:=j and S_REQ(REQ_ID)	B
T7) T	Satisfied	S_GNT	FA
T8) B	R_GNT	None	T
T9) B	R_REQ(j) and j=REQ_ID	None	HT
T10) B	R_REQ(j) and j≠REQ_ID	If j>REQ_ID, then REQ_ID:=j and S_REQ(REQ_ID)	B
T11) HT	R_GNT	None	HT
T12) HT	R_REQ(j)	If j>REQ_ID, then REQ_ID:=j	HT
T13) HT	Satisfied	S_GNT	H
T14) H	R_REQ(j)	If j>REQ_ID, then REQ_ID:=j	H
T15) H	Up-stream Busy	S_REQ(REQ_ID)	B
T16) H	R_GNT	None	FA

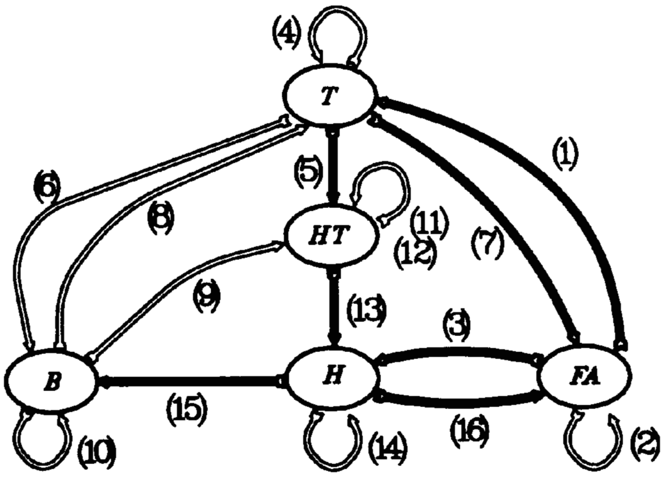


Fig. 5. Ring local fairness—State transition diagram.

back to the node or be terminated. The exact value of $T_{\max}$ is discussed in Section III-F.

D. Properties of the Local Fairness Algorithm

In this section, we will present the correctness and properties of the local fairness algorithm on dual rings (the dual bus algorithm can be viewed as a simple special case).

The initial REQ_ID of node i is $\{ID^i, s^i\}$ and has two parts: i) ID^i, the initial tail (originating node) unique ID (most significant part); and ii) s^i, the sequence number (least significant part) given by the initial tail node of this path.

Each node has a finite set, S, of sequence numbers. Node i can reuse a sequence number, s^i, only if it has not received its own REQ_ID, $\{ID^i, s^i\}$, for a time interval of at least $2T_{\max}$. This immediately implies:

Lemma 1: At any given time, all REQ(REQ_ID) signals sent on the ring are distinct.

Proof: The REQ(REQ_ID) signal is initiated only when a node changes its state from FA to T (T1 in Fig. 5). A node which receives this signal will either forward or terminate it, but will never modify it (T3, T12, and T14). Since we assume that a node cannot be in the restricted mode for more than $T_{\max}$, and after at most $T_{\max}$ a request signal will either get back to its originating node or be terminated. Therefore, when a node initiates a new REQ(REQ_ID), its previous request signal with the same REQ_ID value has been terminated. ■

Lemma 2: At any given time, all nodes with the same REQ_ID have received this value from the same REQ(REQ_ID) signal.

Proof: Assume that when a REQ(REQ_ID) is initiated one of the nodes on the ring has the same REQ_ID value. However, since a node cannot be in the restricted mode with the same REQ_ID value for more than $T_{\max}$ and since the minimum time for initiating two request signals with the same REQ _ID is $2T_{\max}$, the assumption is contradicted. ■

Based on the previous two lemmas, we give the following definition.

Definition 1: A VALID REQUEST PATH is a continuous segment of nodes, in the restricted mode, that have one of the following structures.

1. It starts with a node in the T state, possibly follows by some nodes in the B state, and ends with a node in the H state.
2. It covers the whole ring, where a single node in the HT state and the rest are in the B state.
3. There are three transition cases of VALID REQUEST PATH:

   a.) A GNT signal was sent from a node in the T or HT state (T7 or T13 in Fig. 5) such that the tail is in transition.

   b.) A REQ(j) signal was sent from a node in the FA or H state (T1 or T15) such that the head is in transition.

   c.) A REQ(j) signal was sent from a node in the B state within the VALID REQUEST PATH (T6 or T10).

Theorem 1: If no REQ(j) signal is in transit within a VALID REQUEST PATH, then the REQ_ID's of two adjacent nodes in the path are either the same or the node closer to the tail has a smaller REQ_ID.

Proof: **By Induction.** *Initial Step:* All the nodes in the VALID REQUEST PATH have the same REQ_ID value, which is the initial value each node receives when it enters the restricted mode. *Induction step:* A node in the T, B, H, or HT state receives a REQ(j) signal (T6, T10, T12, or T14 in Fig. 5); if $j \leq$ REQ_ID, then it will not change its REQ_ID value. If the downstream node that sent this signal has already changed its REQ_ID value, then the upstream node has an equal or higher value than its downstream neighbor and the theorem holds. If $j >$ REQ_ID, the node will copy REQ_ID: $= j$ (T6, T10, T12, or T14) and, if it is in the T or B state, the REQ(j) is forwarded upstream. Since a REQ(j) signal is initiated by a downstream node in the T state, it will progress until it reaches the head of the VALID REQUEST PATH or terminate when it reaches a node with a higher REQ_ID value; again, the theorem holds. ■

Corollary 1: If the VALID REQUEST PATH covers the whole ring, there is only one node in the HT state.

Lemma 3: The three transition cases (3a, 3b, and 3c) in Definition 1 lead to either structure 1 or 2 in Definition 1.

Proof: Assuming that the initial network has structure 1 or 2 (Definition 1), we can then prove the three transition cases.

(3a) The GNT signal can be received by either a node in the B state, which changes its state to T (T8), or by a node in the H state, which change its state to FA (T16). In both cases, Definition 1 holds. Note that T2, T4, and T11 are cases in which a GNT signal was received but caused no action or state transition.

(3b) The REQ(j) signal can be received by either a node in the T state, which changes its state to B (T6) or HT (T5), or by a node in the FA state, which change its state to H (T3). In both cases, Definition 1 holds.

(3c) The REQ(j) signal can be received by either a node in the B state, which will remain in the B state (T10) or change its state to HT (T9), or by a node in the H or HT state (T14 or T12), which will remain in the same state. Again, in these cases Definition 1 holds. ■

Theorem 2: At any given time t, the ring contains only VALID REQUEST PATH's and nodes which are in the FA state. There is no signal on the way which can violate this invariant upon its reception.

Proof: We will prove Theorem 2 by an induction on the sequence of possible events. At the time of initialization, all nodes are in the FA state. Assume that some event takes place at time t and assume that the theorem holds until this time. We will show that it still holds after the event of time t by showing that all 16 possible events and state transitions, described in Fig. 5, can be mapped to one of the five cases in Definition 1 (1, 2, 3a, 3b, or 3c). Then, following lemma 3, the proof is completed.

T1, T15 are mapped to 3b; in T2, the node remains in FA state; T3, T4, T8, T14 are mapped to 1; T5, T9, T11, T12 are mapped to 2; T6, T10 are mapped to 3c; T7, T13 are mapped to 3a; and in T16, the node returns to the FA state. ■

E. Local and Global Fairness Equivalence Relation

The local fairness is achieved by limiting the transmission of each node in the restricted mode to some predefined quota of cells or bytes. A node receives a new quota each time it enters the restricted mode. When the load is high, all nodes can be continuously in the restricted mode, which means that a single VALID REQUEST PATH continuously covers the whole ring (Corollary 1).

This scenario is equivalent to the global fairness concept presented in [4]. The global fairness algorithm is based on a single control signal called SAT (from the word SATisfied), which can be viewed as a combination of the REQ and GNT signals. The next discussion briefly describes the SAT-based global fairness algorithm, and shows the equivalent relation between the local and global fairness concepts.

The SAT-based global fairness algorithm: The access on each direction of a dual ring is regulated by a hardware control signal, SAT, which circulates in the opposite direction to the data traffic it is regulating. Fig. 6 describes the basic ring mechanism for one direction. In principle, the node will forward 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 quota since the last time it forwarded the SAT signal.

Local and global fairness equivalence relation: When the network is fully loaded (i.e., each node is covered by upstream traffic and always in the restricted mode), the following will hold.

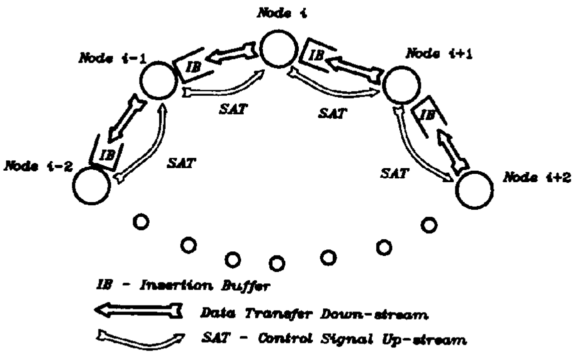


Fig. 6. The SAT-based global fairness mechanism (one direction).

1. The VALID REQUEST PATH covers the whole ring with a single node in the HT state (Corollary 1).
2. When the node in the HT sends a GNT (change state to H), it will immediately be followed by a REQ signal (change state to B).
3. The upstream node that receives the GNT and REQ signals will first enter the T state, and then immediately enter the HT state.

Corollary 2: When a VALID REQUEST PATH covers the whole ring with one node in the HT state, the local fairness algorithm is equivalent to the global fairness algorithm with a single SAT signal.

F. Fault Tolerance

The operation of the network is based on an unreliable transfer of control signals, i.e., no data link control (DLC) protocol is used to ensure reliable transfer of the REQ and GNT signals. Thus, if a REQ or GNT is lost, the transmitting and receiving nodes will not "know" about it. As a result, some nodes or the entire network can deadlock, i.e., remain in the restricted mode for an unbounded time. In the following, we will show how the system returns to normal operation after a node or link failure and intermittent control signal loss.

1) Node or link failure: It is assumed that link and node failures are detected by some data link control protocol. When a link or node is detected faulty, it is removed from the network. In this case, the local fairness algorithm will operate independently on any connected dual bus segment.

2) Control signal loss and timeout: As previously stated, we assume that when the network operates in high speed, the control signals of the fairness algorithm are transferred without error recovery. As a result, if either a GNT or REQ(j) signal is lost, the algorithm will stop. This failure is detected by a timeout event, since a node can be in the restricted mode for some maximum predefined time period. The following two properties are obtained from the fact that the request paths are always merged (T6 in Fig. 5).

Lemma 4: All nodes whose traffic interferes with one another will be part of the same VALID REQUEST PATH.

Proof: Immediately from the algorithm and the VALID REQUEST PATH definition. The VALID REQUEST PATH is always extended upstream to cover all interfering sources and, since overlapping VALID REQUEST PATHs are merged, all interfering nodes are part of the same VALID REQUEST PATH. ■

Lemma 5: $T_{\max}$, the maximum time a node can be in the restricted mode continuously, is at most the time it takes each node to transmit one quota plus one ring's propagation delay, i.e., $T_{\max}$ is proportional to the size of the network.

Proof: Immediately from Lemma 4, and the restriction that a node can transmit one quota while it is in the restricted mode. ■

Thus, if a node resets its local timer each time it enters the restricted mode and if the time period expires after $T_{\max}$ (before the node exits the restricted mode), then we can say that a timeout has occurred.

Fault Tolerant Procedure: If a node in the restricted mode has a timeout event, it will send a GNT signal upstream and will exit the restricted mode, i.e., change its state to FA.

Theorem 3: The fault tolerant procedure ensures that the local fairness algorithm is deadlock free and that, after a failure, the network will return to normal operation.

Proof: To show this, we can observe that in all the restricted mode states (T, B, H, and HT) the node returns to the FA state. The upstream node that receives the GNT global signal will make one of the following transitions: i) T to T (T4); ii) B to T (T8); iii) H to FA (T16); iv) HT to HT (T11); and v) FA to FA (T2). Thus, this node is either in the FA state or T and HT states, in which it will send a GNT signal upstream when it is satisfied.The GNT signal causes the REQUEST PATH to be terminated such that, when a new REQUEST PATH is created, it will be consistent with the VALID REQUEST PATH definition. ■

IV. Performance Study

In this section, we study the throughput performance of the local fairness algorithm presented using simulations. The results are compared to that obtained under: i) the Max–Min fairness condition defined in [7], [9]; ii) pure buffer insertion ring without fairness enforcement; and iii) the global fairness algorithm studied in [1].

We adapt the definition of Max–Min fairness to the dynamic traffic pattern environment of our LAN. The well-known Max–Min fairness [7, pp. 448–453] allocates rates (bandwidth) to users (who each have a path associated with it) according to the following rule.

Max–Min Definition: "*maximize the allocation of each user i subject to the constraint that an incremental increase in i's allocation does not cause a decrease of some other user's allocation that is already as small as i's or smaller.*"

It is also shown that an equivalent definition of Max–Min fairness can be given from the concept of "bottleneck link" as follows. For each user, say i, there is some link k on the path it is using such that k's bandwidth is completely used up, and i's rate is at least as large as the rate of any other user using link k. This link k is called the bottleneck link of i's path [9]. The throughput achievable by every node can be predicted by applying this bottleneck link concept. Given a traffic pattern, we first divide the bandwidth of the most

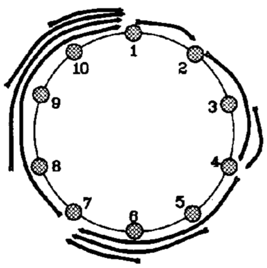


Fig. 7. Traffic scenario 1.

congested links among those nodes competing for them. This process determines the throughputs achievable for these nodes, and the same process is applied iteratively on the residual link bandwidths for the rest of the nodes until the throughput for every node is determined.

An event-driven simulation model for this system was constructed using the RESQ simulation language [15], [16]. In this model, packets generated by nodes attached to the ring arrive at each individual input queue and wait for access to the ring before they are delivered to their respective destinations. The number of nodes attached to the ring, the arrival processes, the packet length distributions, and the destination distributions are all input parameters in the simulation model. The various combinations of these parameters allow us to simulate and study a wide range of system configurations.

For this study, we considered a unidirectional ring with 10 nodes attached. The ring has a transmission bandwidth of 1 Gb/s, and each node generates fixed-size packets with lengths of 1 Kb. The propagation delay on the ring is assumed to be 5 μs/km, and we assumed a ring length of 1 Km with nodes on the ring equally spaced. In order to demonstrate the effectiveness of the local fairness algorithm, we constructed a few traffic scenarios with localized communication patterns. In Fig. 7, the ten links connecting the nodes in the ring are divided into four groups with 1–4 links in each group, respectively. Stations belong to the same group as their most immediate downlinks. Thus, Group 1 contains node 1; Group 2 contains nodes 2, 3; Group 3 contains nodes 4–6; and Group 4 contains nodes 7–10. Each node transmits to the head node in the next group, making the most downstream link in each group the bottleneck resource.

We simulated a fully loaded system, i.e., every node always has something to send, with this traffic configuration implementing the local fairness algorithm. We also simulated the same system with the global SAT algorithm described in [1], [4] and without any fairness control. The throughput achieved by each node under these various algorithms, along with the Max–Min optimal throughput for this traffic configuration, are shown in a table in Fig. 8.

If a fairness algorithm is not provided, then due to the Buffer Insertion protocol, the upstream nodes in each subgroup have an advantage over their downstream neighbors when competing for the same communication resources. This is seen in Fig. 8, where nodes 1, 2, 4, and 7 are able to deliver the maximum nodal throughput of 1 Gb/s while all other nodes are starved with zero throughput. The optimal fairness allocation under the Max–Min definition divides the bandwidth of the bottleneck links equally among those competing for it. Therefore, node 1 should achieve a throughput of 1 Gb/s; nodes 2 and 3 should achieve a throughput of 1/2 Gb/s each; nodes 4–6 each gets a throughput of 1/3 Gb/s; and nodes 7–10, each gets a throughput of 1/4 Gb/s. When applying the global fairness algorithm, because the whole ring is treated as one single resource, the bottleneck link dictates the bandwidth allocation to every node on the ring. Therefore, each node gets a throughput of 0.25, resulting in a total ring throughput of only 2.5 Gb/s. The local fairness algorithm, however, treats each link in the ring as a separate resource and divides the bandwidth equally among members of each competing group. We see that, in this traffic scenario, the throughput results of the local fairness algorithm achieve the optimal ones.

node id	1	2	3	4	5	6	7	8	9	10	Total
no fair	1	1	0	1	0	0	1	0	0	0	4
optimal	1	0.5	0.5	0.33	0.33	0.33	0.25	0.25	0.25	0.25	4
global	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	2.5
local	1	0.5	0.5	0.33	0.33	0.33	0.25	0.25	0.25	0.25	4

Fig. 8. Throughput (Gb/s) comparison under traffic scenario 1.

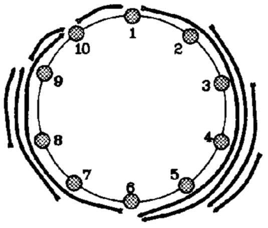


Fig. 9. Traffic scenario 2.

The second traffic scenario we tested is shown in Fig. 9. In this scenario, we perturbed the overlay patterns by shifting the bottleneck links from the most downstream links of each group to the middle links of the group. We also blurred the boundaries of the competing groups for nodes 6–9. The purpose is to investigate the responsiveness of the local fairness algorithm, since in this algorithm the control mechanism is activated by the starved node in each group and traverses upstream to the source of the disturbance. In particular, we want to investigate, under the local fairness mechanism, whether node 9 is forced to have throughput equal to that of nodes 6–8 or if it is able to transmit beyond that and fully utilize link 9. The throughput results for this scenario are shown in Fig. 10 for the cases with no fairness, optimal fairness, global fairness, and local fairness. It is seen that the robustness of the local fairness algorithm permits link 9 to be fully utilized. However, node 9 has a slight advantage over the competing transaction from node 6 and the local fairness algorithm produces near-optimal throughput results. It is also interesting to see that, in this scenario, the local fairness algorithm achieves a higher total throughput than the system without any fairness mechanism.

We further blurred the boundaries of competing groups in traffic scenario 2 by adding a transaction from nodes 5–8 to

node id	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Total*
no fair	1	0	0	0	0	1	0	0	0	1	3
optimal	0.25	0.25	0.25	0.25	0	0.33	0.33	0.33	0.67	1	3.67
global	0.25	0.25	0.25	0.25	0	0.25	0.25	0.25	0.25	0.25	2.25
local	0.25	0.25	0.25	0.25	0	0.3	0.3	0.3	0.7	1	3.6

Fig. 10. Throughput (Gb/s) comparison under traffic scenario 2.

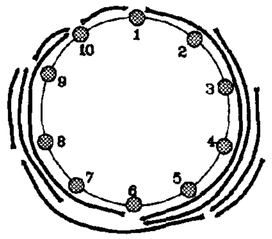
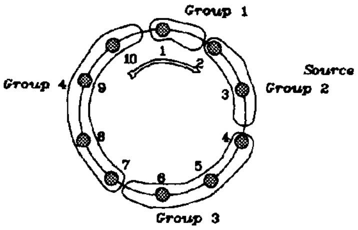


Fig. 11. Traffic scenario 3.

node id	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Total*
no fair	1	0	0	0	0	1	0	0	0	1	3
optimal	0.25	0.25	0.25	0.25	0.33	0.33	0.33	0.33	0.67	1	4
global	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	2.5
local	0.167	0.167	0.167	0.167	0.278	0.278	0.278	0.278	0.722	1	3.5

Fig. 12. Throughput (Gb/s) comparison under traffic scenario 3.



Destination

Source	1	2	3	4	5	6	7	8	9	10
1		1								
2			.5	.5						
3				1						
4					.33	.33	.33			
5						.5	.5			
6							1			
7	.25							.25	.25	.25
8	.33								.33	.33
9	.5									.5
10	1									

Destination Probability Matrix
(All the blanks are zeros)

Fig. 13. Traffic scenario 4.

yield traffic scenario 3. The traffic pattern is shown in Fig. 11, while the performance results are shown in Fig. 12. We see that, in this case, the local fairness algorithm produces less-than-optimal throughputs for each node while preserving, in general, the throughput ratios among the competing nodes. Again, in this scenario, the local fairness algorithm achieves a higher total throughput than the system without any fairness mechanism.

In traffic scenario 4, we take the same grouping as in traffic scenario 1 but introduce randomness in the destination selection. In particular, we let each node transmit to every downstream node in the same competing group and to the head node in the next group with equal probability. The traffic pattern and destination matrix are shown in Fig. 13, while the performance results are shown in Fig. 14.

Recall that, without fairness mechanisms, nodes get to transmit only when they are not covered by upstream traffic. Therefore, for group 3, node 4 transmits all the time, node 5 transmits only when node 4 is sending to 5, and node 6 transmits when 4 is sending to 6 or when 4 is sending to 5 and 5 is sending to 6. Based on the destination probability matrix, it is derived that the throughputs for nodes 4–6 are 1, 1/3, and 1/2 Gb/s, respectively. Fig. 14 verifies this result by simulation. The Max–Min fairness is defined for static requirements, and we have dynamic contention patterns in this traffic scenario. We obtained the "optimal" throughput results in Fig. 14 by applying the Max–Min algorithm to the stationary flow requirements on each link. This results in equalized throughputs for nodes in each group. We cannot conclude with confidence that this is indeed the optimal solution for the dynamic traffic scenario, and it is an open question for future study. However, the local fairness mechanism does not equalize the throughputs for nodes 4–6 and nodes 7–10. In each of these groups, the most downstream node gets the highest throughput and the most upstream node gets the second highest, while the middle nodes get less throughputs. This might be due to the fact that the contention patterns are dynamic. Each node is involved in a different number of control paths depending on its location, and therefore achieves different throughputs. As we have indicated, the issue of what is optimal fairness under the dynamic traffic is open.

node id	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Total*
no fair	1	1	0.5	1	0.33	0.5	1	0.25	0.33	0.5	6.41
optimal	1	0.67	0.67	0.55	0.55	0.55	0.48	0.48	0.48	0.48	5.89
global	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	0.39	3.9
local	1	0.67	0.67	0.55	0.46	0.58	0.46	0.39	0.4	0.56	5.75

Fig. 14. Throughput (Gb/s) comparison under traffic scenario 4.

To summarize, we have demonstrated the robustness of our local fairness algorithm by comparing the throughput results with those of the global SAT algorithm, no fairness control, and the optimal fairness under various traffic scenarios. We showed that, for static traffic, the local fairness algorithm achieves either optimal (scenario 1) or near optimal (scenarios 2 and 3) throughput results. For scenarios 2 and 3, the local fairness mechanism even delivers a higher total throughput than the system without fairness control.

V. Conclusions

We have presented in this paper a local fairness algorithm to be used in high-speed LAN/MAN architectures with spatial bandwidth reuse. In this algorithm, the control mechanism is only triggered when starvation is detected; otherwise, each node in the network is allowed to transmit freely following the basic access protocol. In the event that the control mechanism is triggered, it is only enforced locally, i.e., among the nodes of the network where the conflict occurs. Due to these characteristics, our algorithm can achieve high throughput while providing fairness.

Our algorithm also provides fault-tolerant property. The timeout value for this property is found to have a tight bound of $O(n)$, where n is the number of nodes in the network. This tight bound ensures fast recovery from deadlock conditions. The ability to have fast recovery from deadlock is very important in high-speed networking environments, since in these environments the control signals are transferred without error recovery protection.

In the performance study section, we demonstrated the robustness of our algorithm through simulations. It was found that, for static traffic, our algorithm achieves either optimal or near-optimal throughputs according to the well-known Max–Min fairness definition. We also showed that, for some traffic scenarios, our local fairness mechanism even delivers higher total throughput than the system without fairness control. This demonstrates that, contrary to conventional wisdom, fairness does not always mean tradeoff with system throughput.

ACKNOWLEDGMENT

The authors would like to thank A. Mayer for useful comments and discussions.

REFERENCES

[1] J. Chen, H. Ahmadi, and Y. Ofek, "Performance study of the metaring with Gbps links," in *Proc. 16th Local Comput. Netw. Conf.*, Minneapolis, MN, Oct. 1991, pp. 136–147.

[2] I. Cidon, I. Gopal, P. M. Gopal, R. Guerin, and M. Kaplan, "plaNET and ORBIT: An overview," IBM Res. Rep., 1992.

[3] I. Cidon and Y. Ofek, "Distributed fairness algorithm for local area networks with concurrent transmissions," in *Proc. 3rd Int. Workshop on Distrib. Algorit.*, Sept. 1989, pp. 57–69.

[4] I. Cidon and Y. Ofek, "MetaRing—A full-duplex ring with fairness and spatial reuse," *IEEE Trans. Commun.*, vol. 41, no. 1, pp. 110–120, Jan. 1993.

[5] R. Cohen, Y. Ofek, and A. Segall, "A new label based source-routing in multi-ring networks," in *Proc. 3rd Int. Workshop on Protoc. for High-Speed Netw.*, (IFIP WG6.1/WG6.4), 1992.

[6] 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.

[7] R. G. Gallager and D. P. Bertsekas, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[8] E. R. Hafner, Z. Nenadal, and M. Tschanz, "Integrated local communications—Principles and realization," *Hasler Rev.*, vol. 8, no. 2, pp. 34–43, 1975.

[9] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. COM-29, no. 7, pp. 954–962, July 1981.

[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. 8, no. 8, pp. 1582–1594, Oct. 1990.

[11] M. T. Liu and D. M. Rouse, "A study of ring networks," in *Proc. IFIP WG6.4/Univ. of Kent Workshop on Ring Technol. Based Local Area Netw.*, Sept. 1983, pp. 1–39.

[12] Y. Ofek, "Integration of multi-ring on the MetaRing architecture," in *Proc. 2nd IEEE Workshop on Future Trends of Distrib. Comput. Syst.*, Egypt, 1990, pp. 190–196.

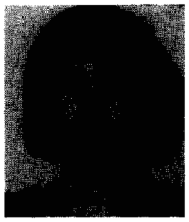
[13] Y. Ofek, "Overview of the MetaRing architecture," *Comput. Netw. and ISDN Syst.*, to appear.

[14] H. Ohnishi, N. Morita, and S. Suzuki, "ATM ring protocol and performance," in *Proc. ICC'89*, 1989, pp. 394–398.

[15] C. H. Sauer, E. A. MacNair, and J. F. Kurose, "The research queueing package version 2: Introduction and examples," IBM Res. Divis., RA 138, 1982.

[16] C. H. Sauer, E. A. MacNair, and J. F. Kurose, "The research queueing package version 2: CMS reference manual," IBM Res. Divis., RA 139, 1986.

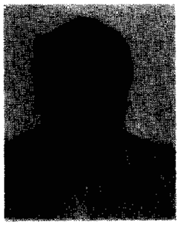
[17] H. Wu, Y. Ofek, and K. Sohraby, "Integration of synchronous and asynchronous traffic on the MetaRing architecture and its analysis," in *Proc. ICC'92*, 1992, pp. 147–153.



Jeane S.-C. Chen was born in Taipei, Taiwan. She received the B.S. degree from the National Chiao-Tung University, the M.S. degree from Washington University, St. Louis, and the Ph.D. degree from Columbia University, New York, in 1990.

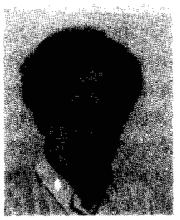
Since 1982, she has been employed by the IBM T. J. Watson Research Center, Yorktown Heights, NY, where she has been involved in data compression for storage of Chinese characters; optimal design and feedback control for electromagnetic devices; design, analysis, and prototyping of high-speed printer actuators; and, currently, modeling and performance analysis of communication networks.

Dr. Chen is a member of the IEEE Communications Society, and is currently the Book Reviews Editor for the *IEEE Communications Magazine*.



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

From 1980 to 1984, he was a Teaching Assistant and a Teaching Instructor at the Technion. From 1984 to 1985, he was with the Department of Electrical Engineering 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 and Algorithms Group involved in various broadband networking projects. Since 1990, he has been with the Department of Electrical Engineering at the Technion. In 1989, he received the IBM Outstanding Innovation Award for his work on the PARIS high-speed network. He currently serves as the Editor for Network Algorithms for the IEEE T RANSACTIONS ON COMMUNICATIONS and an Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING. His research interests include 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, Haifa, Israel, 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 at Fermi National Accelerator Laboratory, Batavia, IL, and from 1984 to 1986 he was with Gould Electronics, Urbana, IL. Since 1987, he has been 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 parallel systems, clock synchronization, self stabilization, and fault tolerance. He initiated and has been leading the research activities on MetaRing and MetaNet architectures.