Improved Fairness Algorithms for Rings with Spatial Reuse

Israel Cidon, *Senior Member, IEEE,* Leonidas Georgiadis, *Senior Member, IEEE,*
Roch Guérin, *Senior Member, IEEE,* and Yuval Shavitt, *Member, IEEE*

***Abstract*—Ring network architectures that employ spatial reuse permit concurrent transmissions of messages over different links. While spatial reuse increases network throughput, it may also cause starvation of nodes. To alleviate this problem, various policies have been suggested in the literature. In this paper, we concentrate on a class of such policies that achieves fairness by allocating transmission quotas to nodes. For such policies, we provide mechanisms for improving delays and increasing overall throughput without compromising fairness.**

***Index Terms*— Buffer insertion, fairness, ring networks, throughput.**

I. Introduction

The recent increase in transmission speeds has resulted in the emergence of new networking standards and architectures. In the local- and metropolitan-area (LAN/MAN) environments, the availability of greater transmission speeds has changed many of the assumptions traditionally made when comparing performances of various architectures [5], [6]. In particular, transmission and buffering delays, which used to be a major concern, now typically represent only a small fraction of the overall end-to-end delay which is dominated by the propagation delay. This shift in the relative importance of the different delay components has led to a renewed interest in rings networks that employ *spatial reuse* as an attractive alternative for high-speed LAN/MAN's.

In a ring network that employs spatial reuse, multiple simultaneous transmissions are allowed as long as they take place over different links. As a result, the total ring throughput can be significantly higher than the capacity of a single link. The recognition of this advantage has generated numerous proposals both for new LAN/MAN architectures [18], [12], [17], [9], [16], [2], and for upgrades of existing standards to support this feature [21], [19], [13].

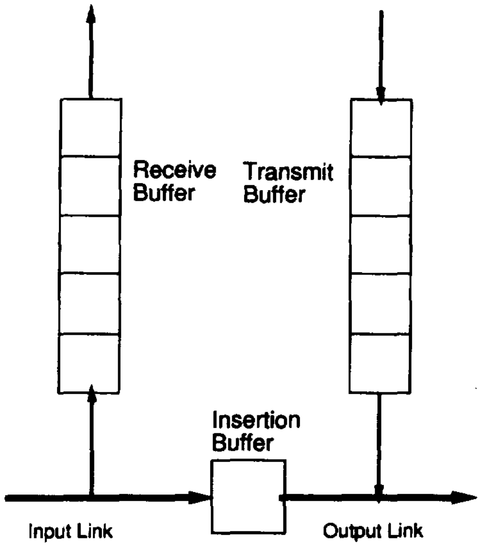


Fig. 1. A station on a buffer insertion ring.

Two common approaches for achieving spatial reuse are the slotted ring and the buffer insertion ring. Both schemes introduce spatial reuse by employing destination removal of packets. In the slotted ring, a fixed number of slots circulate constantly on the ring. A node can use a slot if it is either empty or contains a message destined to that node. In this architecture, messages are of fixed size equal to the amount of data that can be transmitted in a slot. Buffer insertion rings, on the other hand, can support variable size packets, and operate as follows (see Fig. 1). On the receiving side of each link, there is an insertion buffer (IB), which can store at most one maximal size packet. A node may start a packet transmission at any time as long as its IB is empty and there is no ring traffic on the link. If ring traffic is arriving while the node is in the middle of a packet transmission, then this traffic is stored in the IB until the packet transmission is completed. The node cannot transmit anymore until the IB becomes empty and there is no ring traffic on the link, i.e., nonpreemptive priority is given to the ring traffic. If a node is idle, ring traffic not destined to that node cuts through the IB, without delay.

Since in both slotted and buffer insertion rings spatial reuse results in ring traffic having priority, it is clear that without an access control mechanism, heavily loaded nodes can prevent other nodes from accessing the ring. This situation is known as "starvation." To prevent this problem, mechanisms that guarantee fair ring access to all nodes have been proposed in [12] for slotted rings and in [10] and [9] for buffer insertion and slotted rings. Both mechanisms are based on the idea of

Manuscript received June 1994; revised October 1996; approved by IEEE/ACM Transactions on Networking Editor K. S. Vastola. An earlier version of this paper was presented at the IEEE Infocom'94 Conference.

I. Cidon is with the Department of Electrical Engineering, Technion, Haifa 32000, Israel (e-mail: cidon@ee.technion.ac.il).

L. Georgiadis is with the School of Engineering, Department of Electrical and Computer Engineering, Aristotle University, Thessaloniki 54006, Greece (e-mail: leonid@auth.eng.gr).

R. Guérin is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: guerin@watson.ibm.com).

Y. Shavitt is with the Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21209-2694, USA (e-mail: shavitt@cs.jhu.edu).

Publisher Item Identifier S 1063-6692(97)03025-2.

allocating transmission quotas to the nodes. For purposes of clarity, we concentrate on the case of buffer insertion rings, but the algorithms we present are also applicable to slotted rings.

In addition to guaranteeing fair ring access to all nodes, there are several other performance aspects of importance in such networks. One key measure is throughput. It is particularly important that fairness be enforced while the node throughputs are kept as high as possible. Another performance measure is access delay, and more specifically, Head-of-Line (HOL) delay, i.e., the amount of time the first packet in the transmission buffer of a node has to wait before it is inserted in the ring. HOL delay, or rather the range of its variations, is a major component in the transmission *jitter* that the ring introduces [7]. While the objectives of achieving high throughput and low HOL delays are often conflicting, a reasonable tradeoff is demonstrated here through a number of improvements to the distributed fairness algorithm of [9], which is briefly described next. For details the reader is referred to [9].

In [9], a control signal, the SAT (which stands for SATisfied) rotates around the ring, and grants a transmission quota to each node it visits. The direction of SAT rotation can be either the same as the direction of the traffic it regulates or, in a bidirectional ring, opposite to this traffic (see [9] for a discussion). A node can transmit its own traffic whenever the IB is empty and the link is idle, provided it has not exhausted its current quota allocation, i.e., the amount of data it transmitted since it last released the SAT does not exceed the quota it was allocated then. If a node receives the SAT and is "starved," i.e., it has packets waiting to be transmitted and it has not exhausted its quota, then it holds the SAT until it is SATisfied. A node is deemed SATisfied either if it has exhausted its quota or if it has no more packets to transmit. The holding of the SAT ensures that the upstream traffic, which is starving the node by preventing it from transmitting, eventually stops as transmission quotas are not refreshed. Note that in the case of a starved node that holds the SAT, such a node allocates itself a new transmission quota only upon releasing the SAT, i.e., after its transmission requirements for the previous quota have been satisfied. The reader is referred to [9] for a more precise description of the SAT algorithm, but we note that the algorithm is effective, not only in preventing nodes from starving, but also in ensuring that they share ring bandwidth fairly and effectively, i.e., full utilization of the maximally loaded link is achieved, and it is equally shared between nodes contending for it [14]. The SAT algorithm has, however, a number of shortcomings with respect to HOL delay and total ring throughput.

A first drawback of the SAT algorithm is that, when the ring is heavily loaded, a node may wait a long time, proportional to the number of active stations, before gaining access to the ring. This problem is mostly due to the static quota allocation on which the SAT algorithm relies. The choice of a fixed quota size, independent of load conditions on the ring, either impacts throughput if a small quota allocation is selected, or results in degraded access delay if nodes are granted large quotas. A second disadvantage of the SAT algorithm is that the duration of the SAT rotation cycle is determined by the most heavily loaded links on the ring [14], and this may result in unnecessary losses of throughput in asymmetrically loaded rings. Specifically, nodes that transmit packets only over lightly loaded links will exhaust their quota and stop transmitting early in the SAT rotation cycle, while the SAT is being held by some of the nodes whose traffic traverses heavily loaded links.

In this paper, we propose two mechanisms to address each of the above-mentioned problems with the SAT algorithm. The first mechanism, originally presented in [22], is studied in Section II, and consists of a distributed algorithm for adapting the size of the quotas, which results in significant reduction of HOL delays and maintains fairness. The second mechanism (Section III) uses a single control signal called INFO that informs nodes about the quota status of downstream nodes, so that they can decide if transmissions in excess of their quota allocation are permissible. Both mechanisms can be efficiently combined using a single control signal. We briefly discuss this implementation aspect in Section IV. Note that the addition of these mechanisms preserves the distributed and asynchronous nature of the SAT algorithm, e.g., no central controller or reservation phase are needed. Finally, in Section V, we present methods for recovering from various types of errors that can corrupt the control signals on which the proposed mechanisms rely.

An algorithm for increasing the throughput of the ring with spatial reuse has been proposed in [15]. This algorithm is implemented through the use of a control signal called Distributor that is carried along with the SAT and, as with the INFO signal, allows a node to transmit in excess of its quota in certain situations. The algorithm proposed in the current paper can be considered as a generalization of the scheme in [15]. As will be explained in Section III-B, this generalization significantly increases the instances where a node can transmit in excess of its quota, therefore resulting in throughput increase, while at the same time reducing channel access delays.

Another algorithm that addresses the throughput issue has been proposed in [11]. This algorithm uses fairly complex event-driven state machines (with five states per node), two different message types, and the need to carry (as part of the message) and maintain a node ID parameter. The algorithm may produce a considerable number of control messages as each node may invoke at any time an independent phase of the algorithm. This happens because the algorithm requires starved nodes to actively block upstream interference. These control messages must be inserted dynamically into the data stream. In comparison, our INFO algorithm is accomplished through a single rotating signal (similar to the original SAT), and carries only a hop count information (well contained within two bytes). In addition, while the algorithm in [11] successfully addresses the throughput problem (although through a complex implementation), it does not address at all the HOL delay problem which is increased due to the additional contention. It is also not clear how a quota allocation mechanism can be added to it in a rather simple way.

In [20], Picker and Fellman introduce an enhancement to the SCI fairness protocol in order to improve its throughput.

As described in [20], the specific constraints of the SCI standard prevent a solution that includes upstream signaling. The extension is based on maintaining a "go-bit" (the SCI version of the SAT signal) for each node in the ring (if the number of nodes increases beyond the number of bits supported in the control signal, the go-bits are allocated to a group of nodes). Starved nodes only hold the go-bits of interfering nodes (or groups) so that noninterfering nodes are not prevented from transmission. The above solution requires a control message size which is linear with the number of nodes. While a typical SCI ring is not expected to connect a large number of devices, this is not true in a LAN/MAN environment. Our solution requires a message size which is only logarithmic with this number. The solution of [20] does not target the head of the line delay problem for a similar reason (the expected small size and link distances of an SCI ring). Finally, the use of the dual ring and the transmission of control signals in the opposite direction of the traffic increase the spatial reuse, and speed up the reaction to the flow control signals.

II. Improving Packet Delays

Throughout this paper and unless stated otherwise, we consider a bidirectional ring with n nodes that employs the SAT algorithm described in [9]. The SAT signal rotates opposite to the traffic it regulates. A packet generated by a node is sent over the side of the ring for which the destination node is at a minimal hop distance. In case the destination node is at the same distance on each side, either one can be chosen. Therefore, the maximal distance in number of hops that a packet travels is $\lfloor n/2 \rfloor$. We assume that the maximum size of a packet is $L_{\max}$. Since the packets can be of variable size, it may happen that a node does not have enough quota to complete the transmission of a packet. There are various methods to deal with this case. The proposed algorithms are independent of the method used, however, for definiteness, we chose the following approach here. If the remaining quota is insufficient to complete the transmission of a packet and the SAT is not held by the node, the node holds the packet. If the node holds the SAT, or during the next SAT visit, the node adds to the new allocated quota the quota remaining from the current cycle. The node will now be able to transmit the held packet since the new allocated quota should be at least $L_{\max}$ to allow for the transmission of a maximum packet length. After completing transmission of the held packet, the node becomes SATisfied and releases the SAT. Note that, whenever the SAT is released by a node, its remaining quota is at most equal to the quota allocated during the last SAT visit.

The throughput and delay characteristics of the ring depend heavily on the quota allocated to the nodes. Furthermore, the appropriate quota needed to achieve desirable performances depends on the traffic distribution on the ring. For example, when only one node wishes to transmit, the optimal quota that the SAT should grant is the amount that can be transmitted in one round-trip delay (RTD), T_{RD} seconds, i.e., $Q_{RD} = B_w T_{RD}$ bits, where B_w denotes the link bandwidth in bits/second. A smaller quota allocation would not allow the full utilization of the channel in this case since the node will exhaust its quota before the SAT returns to refresh it. Consider, however, the situation where all of the nodes wish to transmit over the same link. Under the previous allocation it can be shown (see the Appendix) that an upper bound on the HOL delay is

$$4T_{RD}(n-1) + 3T_{RD} + 2n\frac{L_{\max}}{B_w}. \tag{1}$$

The worst case node delay is within a multiplicative factor from the upper bound (1). Indeed, it is possible to show that a node directly upstream from a bottleneck link, say node 1, will be prevented from inserting its own packets for at least $2\lfloor n/2 \rfloor T_{RD}$. For example, consider the following scenario.

Example 1: Suppose that all nodes have empty queues. Starting from node 1, the SAT rotates once around the ring granting quotas to all nodes, and arrives back at node 1. Immediately after leaving node 1 for the second time, all of the $\lfloor n/2 \rfloor$ nodes upstream from node 1 generate new packets that need to be transmitted over the outgoing link of this node. Node 1 also generates new packets. Let us assume for simplicity that all of these $\lceil n/2 \rceil$ nodes are close together so that the propagation delay between the two end nodes on that ring segment, i.e., nodes 1 and $\lceil n/2 \rceil$, is negligible. In this scenario, it is easy to see that a packet from node 1 will have to wait until each of the $\lfloor n/2 \rfloor$ upstream nodes transmitted the equivalent of two quota allocations.

Using the SAT algorithm in a ring with 200 nodes and a round-trip delay of 0.5 ms (which corresponds to a ring size of 100 km), the delay in this scenario can be as long as 100 ms, which will be unacceptable to some applications. It is clear from the above discussion that it would be useful if the quota allocated to the nodes could be adapted to reflect the traffic characteristics. We provide, next, two algorithms by which this quota adaptation is achieved.

Algorithm A: Two *counter fields* $CT1, CT2$ are added to the SAT. During a SAT cycle, defined as the time interval between two successive visits of the SAT to node $1, CT1$ contains the sum of node quota reservations during the previous SAT cycle (to be satisfied in the current SAT cycle), and $CT2$ collects the node reservations during the current cycle. Node 1, which acts as a leader node, generates the first SAT and initializes the counter fields to $CT1 = 0, CT2 = 0$; upon subsequent arrivals of the SAT, node 1 copies the contents of $CT1$ to $CT2$ and reinitializes $CT2$ to 0. When node i receives the SAT, it saves the counter value $CT1$ in a register CV, and increases counter $CT2$ by a value r^i which reflects its current quota reservation. The first SAT cycle is used for initializing the reservation process; no node receives quotas during this cycle. During subsequent SAT cycles, upon release of the SAT, node i allocates itself quota $\nu^i = (r_p^i/CV)Q_{\max}$, where r_p^i is the value requested at the previous SAT visit, so that quotas in the current SAT cycle are, in effect, allocated to nodes in proportion to their reserved requests in the previous SAT cycle. In the simplest and practically important case, the requests r^i take the values 0 or 1, i.e., CV counts the number of active nodes. In the rest of the paper, we assume that $r^i \in \{0,1\}$ unless mentioned otherwise.

Algorithm B: A single counter field CT is added to the SAT. CT is initially zero. When node i receives the SAT, it saves the counter value in a register CV, increases the counter by r^i, and decreases it by the value r_p^i, requested at the previous SAT visit. As with Algorithm A, the first cycle is used for initialization. During subsequent cycles, upon release of the SAT, node i again allocates itself quota $\nu^i = (r_p^i/CV)Q_{\max}$.

In the Appendix, we show the following upper bounds on HOL delays for the two algorithms.

HOL Delay Bound for Algorithm A:

$$6\frac{Q_{\max}}{B_w} + 3T_{RD} + 2n\frac{L_{\max}}{B_w}. \qquad (2)$$

HOL Delay Bound for Algorithm B:

$$4(\ln n + 1)\frac{Q_{\max}}{B_w} + 3T_{RD} + 2n\frac{L_{\max}}{B_w}. \qquad (3)$$

As with the case of fixed quota allocations, scenarios can be given under which the HOL delays of a node are within a constant factor of the bounds given above. As will be seen in subsequent sections, in all of our simulations, we found that quite satisfactory performance can be achieved with values of $Q_{\max}$ less that $10T_{RD}$. Therefore, taking into account the fact that the term $L_{\max}/B_w$ is normally very small, and comparing the bound (1) with (2) and (3), we see that both Algorithms A and B can provide significantly better HOL delay bounds than the fixed quota allocation for a typical local area network. Algorithm A has better HOL delay bounds than Algorithm B. On the other hand, Algorithm B is simpler than A since it does not require special action by any of the nodes and uses a single counter field for its operation. Moreover, in all of our simulation studies, Algorithm B achieved HOL delays which were much closer to those predicted from (2) rather than (3). As we will see in Section IV, Algorithm B can be easily combined with our proposed algorithm for throughput increase in a manner that permits rapid adaptation to load conditions. While Algorithm A also could be used for that purpose, it does not appear to have any advantages in this case. We also note that, while the bounds may change by various modifications that will be introduced later to the basic algorithm, the comments above regarding the relative performance of Algorithms A and B will remain the same. For these reasons, we concentrate on the study of Algorithm B in the rest of the paper.

Since quota allocation to a node depends on the request it registered in the previous SAT release, it is possible (when $r_p^i = 0$) for it to be prevented from transmitting for up to two SAT cycles, even though it has messages to transmit. This will occur if the node receives new messages immediately after releasing a SAT on which the node did not make any quota reservations. To avoid such unnecessary delays, we slightly modify Algorithm B so that the SAT grants a small quota $Q_{\min}$ (typically a few kilobytes) to users which request no quota in a cycle. Note that users with low throughput demands can be SATisfied with the minimal allocation, and need not use the SAT counter. An issue that arises in this case is what a node will do if its requested quota turns out to be smaller than $Q_{\min}$. There are two options in this case, either to use the calculated quota, or to pick $Q_{\min}$ as the quota used for the current cycle. Usually, it should be the case that the minimum calculated quota (when all nodes request quotas) is larger than $Q_{\min}$, and therefore this issue does not arise. In any case, the particular choice has little effect on the performance, and for definiteness, we will choose the first option in this paper. In the following example, we illustrate the operation of the algorithm.

Example 2: For simplicity, we consider a unidirectional ring with three nodes. The direction of traffic is from node a to b to c (see Fig. 2). Nodes a and b always transmit to node c and node c always transmits to a. The distance between a and c is negligible. We assume that all packets are of the same size, and that time is measured in packet transmission time units. Let $Q_{RD} = 10, Q_{\max} = 6 \times Q_{RD} = 60$, and $Q_{\min} = 0$. At time $t = 0$, node c holds the SAT and has 20 packets in its host buffer, nodes a and b have 50 packets, and no new packets are generated by the nodes.

At $t = 0$, node c releases the SAT with counter value 1, and does not attempt to transmit since it has not allocated itself quotas yet. Similarly, nodes a and b release the SAT with counter values 2 and 3, respectively, without attempting to transmit. At time $t = 10$, the SAT is back to node c and the counter value is 3. Node c allocates itself $\nu^c = (60/3) = 20$ packets and releases the SAT immediately, with counter value $3 - 1 + 1 = 3$. Nodes a and b perform the same actions upon reception of the SAT, which is taking place just prior to time $t = 20$ since we assumed that the distance between the nodes is negligible. At $t = 20$, the counter is back at node c. Node c inserted 10 packets in the time interval [10, 20), and still has quota for 10 more packet transmissions. It therefore holds the SAT until time $t = 30$ when it exhausts its allocated quota and its host buffer is empty. Then, node c releases the SAT with value $3 - 1 + 0 = 2$. By time $t = 40$, the SAT is received at node a which completes the quota allocated to it in the previous cycle at the same time $(40 - 20 = 20)$. Since node a still has 30 packets to transmit, it reallocates itself quota $\nu^a = 60/2 = 30$ and releases the SAT with counter value 2. Until $t = 40$, node b is not able to insert any packets on the ring because its insertion buffer is busy with packets sent by node a. Therefore, node b holds the SAT. Since node a starts inserting its newly allocated quota on the ring at $t = 40$, node b cannot start inserting its own packets until $t = 70$. At time $t = 90$, node b exhausts its allocated quota, reallocates itself $\nu^b = 60/2 = 30$ packets, and releases the SAT. Node c releases the SAT immediately without altering the counter value since its host buffer is empty. Node a releases the SAT with counter value $2 - 1 + 0 = 1$, and the SAT arrives at node b at $t = 100$. Node b transmits 10 of its allocated quota in the interval [90, 100), and therefore it holds the SAT until the completion of its allocated quota which takes place at $t = 120$. Then, node b releases the SAT with counter value 0. At this time, the queues of all nodes are empty.

While the method of quota adaptation described above significantly reduces HOL delays and provides satisfactory throughput for many traffic patterns, there exist patterns with potential for high spatial reuse, for which the ring may be underutilized. Consider, for example, a ring for which $Q_{\max} =$

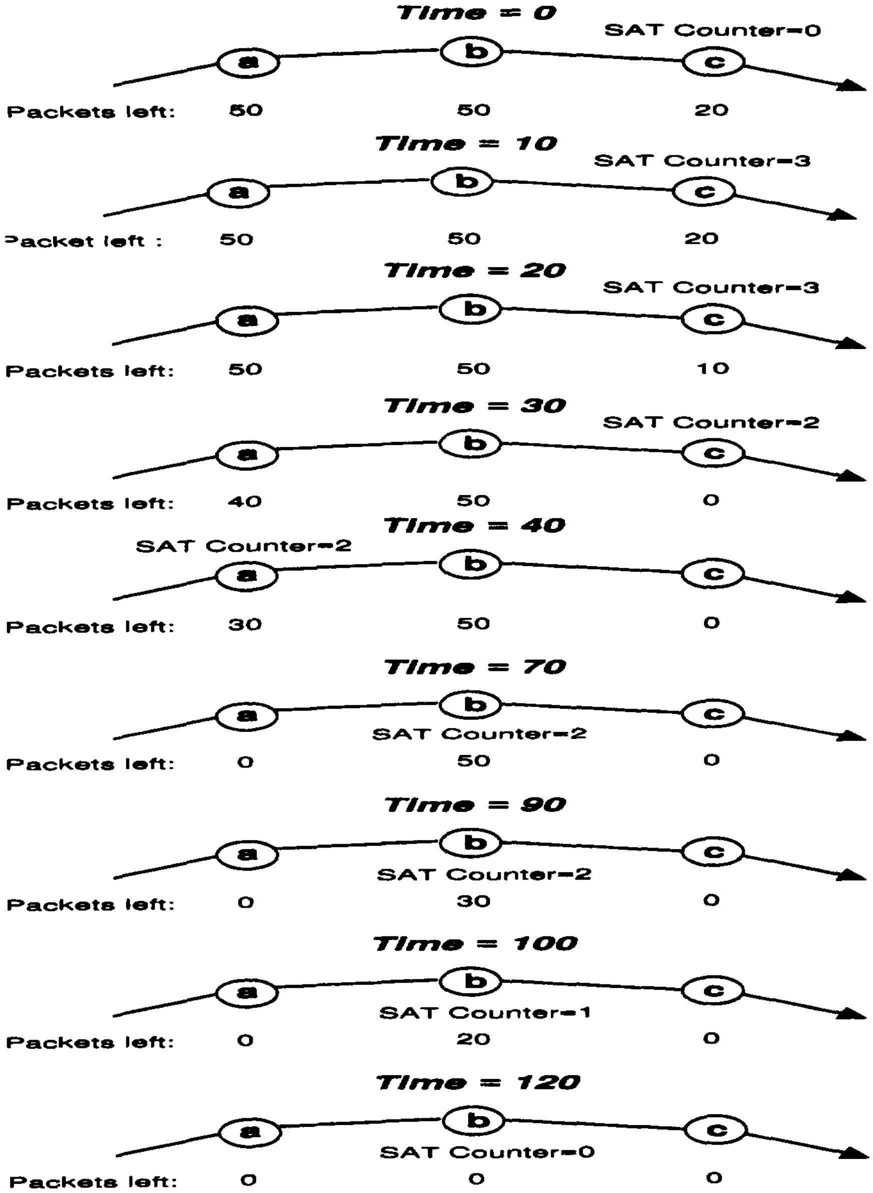


Fig. 2. Example of operation of counter-based quota allocation algorithm.

$10Q_{RD}$, and with 20 active stations which only transmit to their downstream neighbors. Each station receives a quota $\nu^i = 0.5 \times Q_{RD}$, while they should get Q_{RD} to fully utilize their link. In Section III, we describe a mechanism that helps avoid such throughput losses.

A. Simulation Results

To study the properties of the quota adaptation mechanism, we developed a detailed simulation model of the buffer insertion ring. We assume that $r^i \in \{0, 1\}$ and therefore,

$$\nu^i = \begin{cases} Q_{\max} \Big/ \sum_{j=1}^{n} r^j, & \text{for nodes with } r^i = 1 \\ Q_{\min}, & \text{for nodes with } r^i = 0. \end{cases}$$

As stated before, the SAT algorithm without quota adaptation is likely to induce long delays when the ring is heavily loaded. To capture this, the simulations of this section assume that the node buffers are always full. In this environment, an important measure of the incurred delays is HOL delay.

Simulation Model: We simulated a bidirectional ring with 40 nodes. The transmission rate is 1 Gbit/s, the RTD is 500 μs, and the packet size is constant and equal to 1900 bytes. Under the pure SAT algorithm, all nodes are assumed to have quota equal to Q_{RD} so that the individual nodes can fully utilize the ring by themselves. All simulation results are based on at least one million packet transmissions. The measurements refer to one side of the ring. The statistics on the other side are similar.

We simulated the following scenarios.

1) *Uniformly Distributed Destinations:* All nodes have packets to transmit, and packet destinations are uniformly distributed.
2) *A Deprived Node:* Same as above, except that one node, no. 39, is never a destination.

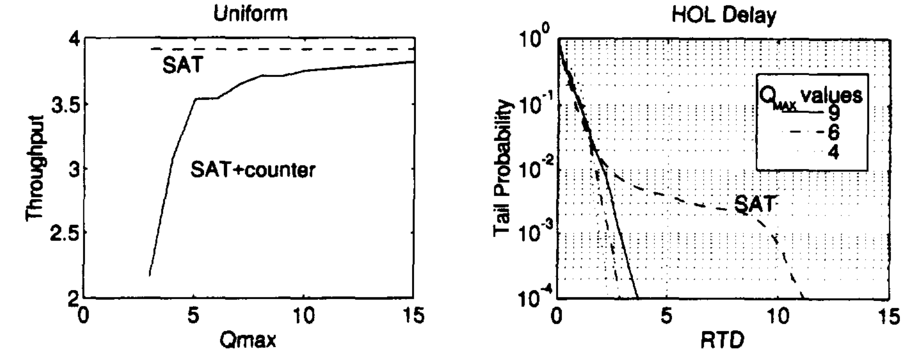


Fig. 3. Destination is uniformly distributed.

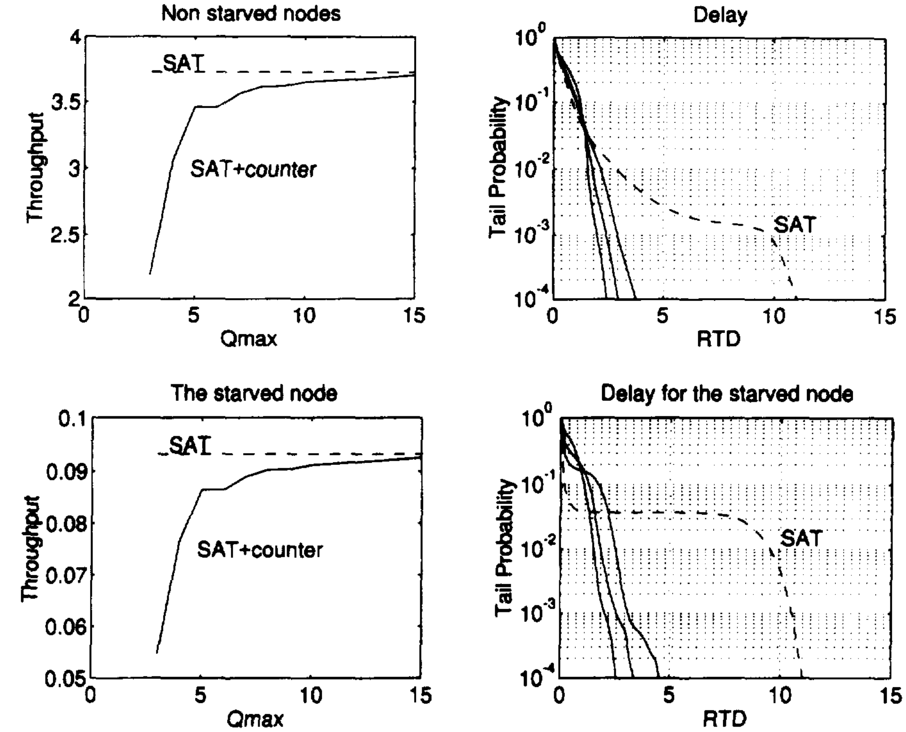


Fig. 4. Node #39 is starved.

3) *A File Server:* All nodes transmit to one specific node (a file server).

1) Uniformly Distributed Destination: Fig. 3 compares the throughput of the SAT and counter algorithm (left) and the HOL-delay tail probability (right). The following observations can be made from the simulation results.

Throughput: The vertical axis describes the throughput of both algorithms. The horizontal axis shows the value of $Q_{\max}$ in multiples of Q_{RD}. As expected, the throughput of the SAT algorithm (dashed line) is close to 4.0 in this scenario. The throughput of the quota adaptation algorithm (solid line) is lower than the throughput of the SAT algorithm and decreases when $Q_{\max}$ decreases. The decrease, however, is less than 10% for $Q_{\max}$ values over $6Q_{RD}$.

Tail Probability: We plot in the right part of Fig. 3 the tail probabilities of the HOL delay for the SAT algorithm when it operates with fixed quota per node equal to Q_{RD} (dashed line), as well as when the quota adaptation algorithm is used with $Q_{\max}$ values of 4, 6, and 9 Q_{RD}. The portion of packets with HOL delay of more than 4 RTD is (close to) zero for the quota adaptation algorithm under all of the tested $Q_{\max}$ values, while for the SAT algorithm, this portion is 0.004, and it drops only by a factor of two for delays of 8–9 RTD. This points to the effectiveness of the quota adaptation algorithm in bounding HOL delays.

2) A Deprived Node: Fig. 4 compares the throughput and tail probability of the HOL delay under the SAT algorithm with and without quota adaptation, when a single node (no. 39) is never a destination. Separate comparisons are made for node 39 and the other nodes. The plots were obtained after simulating the transmission of over four million packets, which translates into more than 100 000 packets/node. For nodes other than no. 39, the deprived node, the results are similar to those of the previous experiment. However, the tail behavior of the HOL delay of node 39 shows an even larger improvement over the SAT algorithm for all of the tested values of $Q_{\max}$.

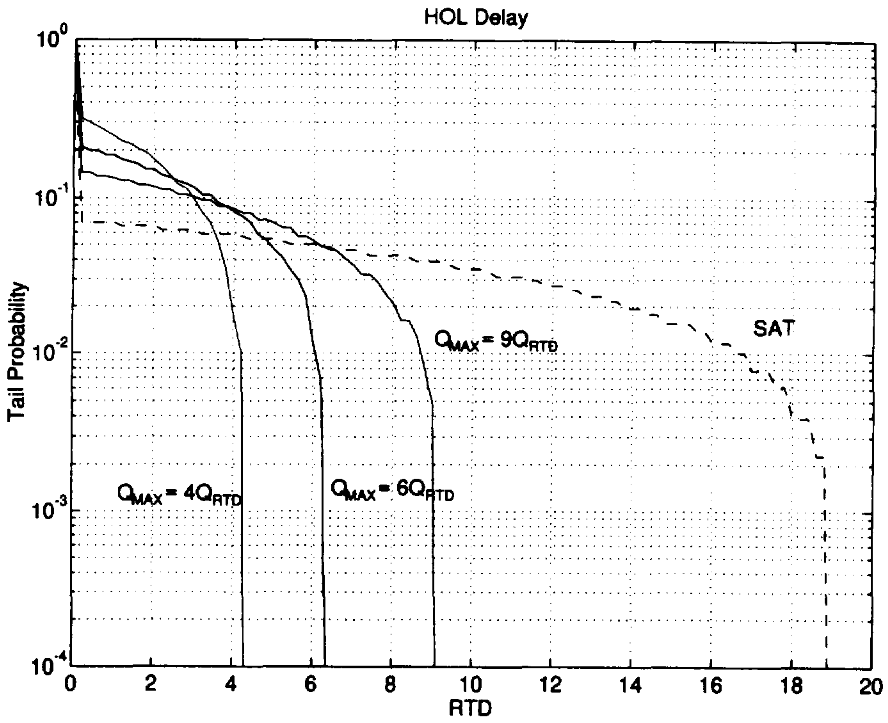


Fig. 5. All traffic goes to one node.

Under the SAT algorithm, about 4% of the packets of node 39 have HOL delays larger than 5 RTD, while the percentage is (almost) zero when the quota adaptation algorithm is used. In fact, without quota adaptation, the HOL delay is over 7.5 RTD for about 3% of the packets and over 9 RTD for 1% of them. This is because node 39 is denied access to the network most of the time unless it holds the SAT and eventually stops transmissions from all other nodes. Since the cycle lengths under the SAT algorithm with fixed allocation are large, the first packet that node 39 transmits in a cycle in general will have a very long HOL delay. This phenomenon is also present even when the quota adaptation algorithm is used, but the lower total quota allocation translates into shorter cycles which improves the tail of the HOL delay.

3) Communication with a File Server: The results for this case are shown in Fig. 5, which only considers the HOL delay since the lack of spatial reuse in this case means that both algorithms have the same throughput. The figure gives the tail probability of the HOL delay without quota adaptation (rightmost curve) and with quota adaptation for different values of $Q_{\max}$ (4, 6, and $9Q_{RD}$ from left to right). The tail probabilities are now larger in all cases since this is a scenario under which maximal HOL delays may occur. However, by controlling $Q_{\max}$, the quota adaptation algorithm is again effective at limiting the tail of the HOL delay.

III. Improving System Throughput

A. Fairness Issues Under Spatial Reuse

Fairness issues arise whenever multiple users attempt to utilize a common resource. From our point of view, the users are the nodes, the resource is the ring, and the objective is to allocate the ring capacity "fairly" to the nodes while keeping the system throughput high. Each node on the ring generates packets destined to some other node. Since we are dealing with asynchronous traffic, packet destinations are generally not known before packet generation time, and are considered random. For such traffic, an appropriate throughput performance criterion for each node is the average number of packets that the node is able to transmit in the long run. Based on this performance criterion, a well known optimization problem that is associated with fairness is the *max–min* optimization [3], where each node gets the largest possible throughput that does not impact nodes with lower throughputs. Specifically, quoting from [3], a max–min fair throughput point $\boldsymbol{v} = \{v^1, \cdots, v^n\}$ on a ring with n nodes has the following property: for $i = 1, \cdots, n, v^i$ cannot be increased while maintaining feasibility without decreasing v^j for some j for which $v^j \leq v^i$, where v^i is the throughput of node i.

While a max–min fair point is satisfactory in many situations, it may be unnecessarily restrictive for the system we consider in this paper. There are situations where, by reducing the throughput of some node j by a very small amount, a tangible increase in the throughput of a node i with an already larger throughput can be achieved (precluded by the max–min solution). To illustrate this, consider the following example.

Example 3: Assume that we have a unidirectional ring with four nodes, and that the direction of traffic is from node 1 to node 4 (see Fig. 6). Nodes 2 and 3 always transmit to node 4, node 4 transmits always to node 1, and node 1 transmits its packets to node 2 with probability 1-ϵ and to node 4 with probability $\epsilon \geq 0$.

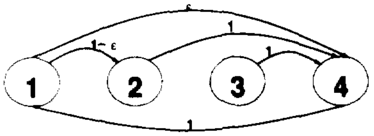


Fig. 6. Example of potential throughput improvement.

It is easy to see that, if $\varepsilon > 0$, the max-min fair vector is

$$\left\{ \frac{1}{2+\varepsilon}, \frac{1}{2+\varepsilon}, \frac{1}{2+\varepsilon}, 1 \right\}.$$

However, another feasible point is the vector

$$\left\{ 1, \frac{1-\varepsilon}{2}, \frac{1-\varepsilon}{2}, 1 \right\}.$$

If we implement a policy that provides the throughputs of the last vector, then we will increase the throughput of node 1 by almost 100% and will decrease the throughputs of nodes 2 and 3 by $(\varepsilon + \varepsilon^2)/2$, which can be arbitrarily small. Mathematically, the max-min fair point has the following discontinuity for the system under consideration: for $\varepsilon = 0$, the max-min fair point is $\{1, 0.5, 0.5, 1\}$, while the limit obtained by considering the max-min fair points for $\varepsilon > 0$ is $\{0.5, 0.5, 0.5, 1\}$.

When all nodes have equal quota, it is shown in [14] that, as the quota sizes increase, the SAT algorithm provides node throughputs that are converging to the point that maximizes the minimum throughput in the network. However, the SAT algorithm equalizes the throughputs of all nodes (that have enough to transmit). Therefore, when ring traffic is asymmetric, a possible increase in the throughput of some nodes that could occur without penalizing other nodes is lost. The quota adaptation algorithm improves the HOL delays of the SAT algorithm, but cannot correct this throughput problem. It has been shown in [14] that the problem can be solved by allocating different quotas to nodes as a function of their traffic distribution. This, however, may not be practical as it requires knowledge of traffic statistics which either may not be available or too variable to estimate accurately. In the next section, we describe a mechanism which, for asymmetric traffic, can provide much larger throughputs than the SAT algorithm, and preserves fairness without requiring knowledge of traffic statistics. The node throughputs resulting from this mechanism are quite close to the max-min fair values. For moderate quota sizes, the undesirable behavior of the max-min fair point described in the previous example is also largely corrected.

B. Throughput Increase Mechanism

The mechanism for increasing throughput is implemented via the use of a control signal called INFO. The INFO, like the SAT, rotates opposite to the traffic it regulates and carries a "hop counter." A node that receives the INFO records the hop counter value, and then increases the hop counter if it is SATisfied or sets it to 1 if it is not. In all cases the node forward the INFO signal to its upstream neighbor without delay. By increasing the hop counter value, the node signals that it is currently SATisfied, and that sending packets through it will no longer interfere with the transmission of its own quota. Once the counter on the INFO signal reaches a specified maximum allowed value, usually the total number of nodes in the ring, it is no longer incremented. When a node has a packet to transmit and it is out of quota, it checks whether the destination of the packet is less than or equal to the last INFO hop counter value; if it is, the packet can be transmitted, and the process is repeated with the next packet. Otherwise, the node refrains from transmitting. Note that since the INFO signal is never delayed, a node's hop counter is updated every RTD time.

The mechanism proposed in this paper, using the INFO signal, can be considered as a generalization of the mechanism proposed in [15], using the Distributor signal, in the following sense. The mechanism in [15] (see also [4] for a description and performance study of this mechanism) allows a node to identify the first downstream node whose queue is nonempty (active node) in the current SAT cycle. Alternatively, the Distributor signal permits a node which has an empty queue upon its receipt of the SAT to indicate its inactive condition to others. This then allows SATisfied nodes to transmit through inactive nodes, even after they have exhausted their quota. The mechanism we propose differs and extends the Distributor signal in two significant aspects. First, because of the use of a separate INFO signal that circulates as fast as possible, the update of a node status is not performed only once per SAT cycle. Typically, since the INFO is never held, it passes through each node several times during each SAT cycle, and therefore provides more timely and accurate information on a node status. Second, the mechanism we propose allows transmission without quotas, not only through idle nodes, but also through SATisfied nodes. These differences in the implementation and the type of information carried by the INFO signal make it much more likely that a node can transmit in excess of its allocated quota which, of course, translates into higher system throughput. This is demonstrated through the following example.

Example 4: Consider a ring with 23 nodes, where 11 consecutive nodes, nodes 1–11, are active and highly loaded, so that their queues are nonempty for a long time. The rest of the nodes are inactive. Consider the side of the ring where traffic flows from nodes 1 to 23. Assume that nodes 2–11 have traffic for node 12, while node 1 has traffic for node 3. We can estimate in a heuristic fashion the node throughputs as follows. Since the traffic from the 10 nodes $2, \cdots, 11$, has to cross the link between nodes 11 and 12, the SAT mechanism assures that the throughput of all these nodes is about 0.1 (under either the INFO or the Distributor mechanism). Consider, now, the throughput of node 1. Since node 2 is always active, although node 1 has traffic for node 3, it (node 1) can never transmit in excess of its quota, and therefore the information in the Distributor signal is of no use. Therefore, its throughput will still be about 0.1. Using the INFO signal, however, node 1 is notified when node 2 is SATisfied. Since node 2 will be busy for about 20% of a SAT cycle time (10% transmitting its own quota traffic and 10% forwarding the quota traffic of node 1), node 1 can use the rest 80% of a SAT cycle time to transmit in excess of its quota. Therefore, its throughput will be close to 0.9.

TABLE I
TRAFFIC PATTERN

Source / Destination	1	3	4	5	6	7	8	9	10	11	18
1											1.0
2		0.5	0.5								
3			1.0								
4				1.0							
5					0.2	0.2	0.2	0.2	0.2		
6						0.25	0.25	0.25	0.25		
7							0.33	0.33	0.33		
8								0.5	0.5		
9									1.0		
10										1.0	
11											1.0
12											1.0
13											1.0
14											1.0
15											1.0
16											1.0
17											1.0
18	1.0										

The fact that the INFO signal circulates independently of the SAT signal and as fast as possible not only makes it possible to provide a node with the appropriate information to identify the first downstream starved node, but it improves access delays as well. To see this, assume that node a has an empty queue and declares itself inactive in the control signal, either INFO or Distributor, but just after this control signal is released, it receives new messages for transmission. If an upstream node has traffic that crosses node a, then under the Distributor mechanism, node a may be blocked for an entire SAT cycle. Under the INFO mechanism, however, the node will be blocked only for about 1 RTD, until the INFO signal arrives back to it and the node declares itself active again.

C. Simulation Results

We consider the following asymmetrically loaded ring. The system consists of 36 nodes, and therefore each node may transmit to 18 destinations on each side of the ring. Let q_j^i be the portion of traffic from node i destined to node j. We have simulated the scenario whose traffic distribution is shown in Table I. Note from the table that all of the packets generated by node 1 have to traverse nodes 2 to 17.

The simulation results for this traffic pattern are given in Table II. The first column gives the max-min fair throughputs of all nodes for reference. Columns 2–4 give the throughputs obtained for the SAT algorithm with a fixed individual node quota of Q_{RD}, and for the combined SAT and quota adaptation (QA) algorithm with $Q_{\max} = 4, 9$, respectively. The INFO signal was used in all cases. For illustration purposes, we present graphically in Fig. 7 the max–min fair node throughputs and the node throughputs corresponding to column 4 in Table II. Note that if the SAT algorithm were used without the INFO signal, node throughputs would all be equal to about 0.123, which corresponds to the minimum throughput achieved by any node for all three experiments (columns 2–4). Furthermore, under the SAT algorithm, increasing the quota allocation of all nodes beyond Q_{RD} can only improve node throughputs up to 0.125, the minimum throughput of the max–min fair point. As can be seen from the results of columns 3 and 4, when the INFO signal is used, node throughputs get closer to their max–min fair values, and the overall ring throughput is more than 2.5 times the ring throughput when only the SAT is used.

We, therefore, see that the INFO signal is successful at improving throughput when congestion varies among the links on the ring. This is achieved without penalizing the nodes with the minimum throughput in the ring and with almost no increase in delays. However, due to the nonzero RTD, small increases in delays may occasionally occur due to inaccurate information concerning the status of nodes. For example, an empty node may have increased the hop counter in the INFO signal although it still had quota left. If it suddenly receives packets, it may have to wait for the INFO signal to come back (1 RTD) to shut off uncontrolled upstream traffic and get the opportunity to finish transmitting its remaining quota. There are various ways to minimize the potential for such delays.

One possible approach is to replace the single rotating INFO signal by event-driven INFO signals, which are triggered on demand. Specifically, whenever a node changes its state from SATisfied to starved or vice versa, it generates an INFO signal. If the node is SATisfied, the INFO signal contains the last INFO value the node received, increased by one. If the node is starved, the INFO signal has the value 1. A SATisfied node that receives an INFO signal with a value different from the one it previously recorded increases the INFO counter by one, and immediately forward the signal. Typically, there will be multiple INFO signals simultaneously on the ring, but unbounded proliferation is avoided by removing INFO signals which correspond to outdated or previously transmitted information. In particular, a starved node that receives an INFO signal simply records its value, but does not propagate it. Similarly, a node that receives an INFO signal with a hop counter value equal to twice the number of nodes in the ring also does not propagate it since this implies that all nodes are satisfied and are aware of this situation. Event-triggered generation of INFO signals does improve delays, but the

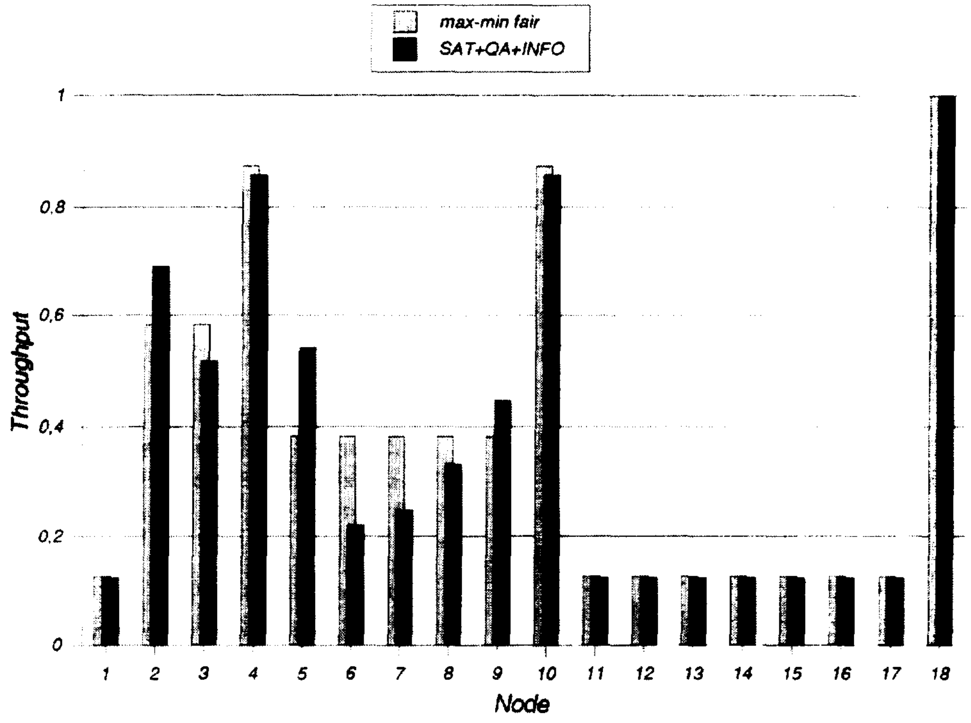


Fig. 7. Achieved and max-min fair node throughputs.

TABLE II
THROUGHPUTS FOR THE TRAFFIC PATTERN OF EXAMPLE 1

node	max-min fair	SAT+INFO	SAT+QA+INFO $Q_{MAX} = 4RTD$	SAT+QA+INFO $Q_{MAX} = 9RTD$
1	0.125	0.123	0.123	0.123
2	0.583	0.732	0.586	0.691
3	0.583	0.497	0.568	0.516
4	0.875	0.859	0.859	0.859
5	0.383	0.600	0.445	0.541
6	0.383	0.226	0.206	0.221
7	0.383	0.250	0.244	0.248
8	0.383	0.334	0.359	0.333
9	0.383	0.434	0.461	0.447
10	0.875	0.859	0.859	0.859
11	0.125	0.123	0.123	0.123
12	0.125	0.123	0.123	0,123
13	0.125	0.123	0.123	0.123
14	0.125	0.123	0.123	0.123
15	0.125	0.123	0.123	0.123
16	0.125	0.123	0.123	0.123
17	0.125	0.123	0.123	0.123
18	1.0	1.0	1.0	1.0

price is an increase in implementation complexity which may outweigh the benefits.

Another positive effect of the INFO signal is that it also avoids some of the undesirable throughput restrictions of a strict max–min approach. To demonstrate this, we simulated the network of Example 3 in Section III-A for a value of $\varepsilon = 0.01$. Both the SAT and the counter algorithms were simulated with and without the INFO signal. The node throughputs of the counter algorithm were found to be very close (for most $Q_{\max}$ values) to those obtained with the SAT algorithm. Therefore, Table III presents only the results obtained when using the SAT algorithm with quota equal to Q_{RD}. The first row gives the max–min fair point for $\varepsilon = 0.01$. Without the INFO, the throughput of node 4 is reduced to 0.4899. When the INFO is used, the throughput of nodes 2 and 3 is reduced slightly (about 0.5%) compared to the pure SAT algorithm, while the throughput of node 1 is almost doubled, and node 4 has the same throughput as the corresponding component of the max–min vector. The resulting throughput vector is very close to the vector that would have been obtained if the traffic of node 1 had absolute priority ($\{1, 0.495, 0.495, 1\}$). We see in this experiment that the use of the INFO signal provides a throughput vector that is more desirable in practice than the max–min fair throughput vector.

TABLE III
$\varepsilon = 0.001$

	Node 1	Node 2	Node 3	Node 4
max-min FAIR	0.4975	0.4975	0.4975	1.0
SAT w/o INFO	0.4899	0.4899	0.4899	0.4899
SAT w INFO	0.9177	0.4877	0.4877	1.0

While the INFO signal can increase throughput beyond that of the SAT with or without the quota adaptation mechanism, it provides no guarantees that this extra throughput will be allocated fairly to the nodes. This is because there are no restrictions on the amount of traffic a node can transmit beyond its allocated quota as long as this traffic does not traverse starved nodes, i.e., it is within the range specified by the last INFO hop counter. Therefore, it is possible that heavily loaded nodes monopolize the extra capacity of the lightly loaded links. For relatively small $Q_{\max}$, on the order of a few Q_{RD}, this phenomenon is not very pronounced since the nodes downstream from the heavily loaded nodes receive the INFO signal first. This means that they typically have a chance to transmit several packets beyond their allocated quota before being interrupted by upstream traffic. However, it is possible to generalize the INFO algorithm to ensure greater fairness when allowing transmissions without quotas, so that the node throughputs better approximate the max–min vector point. The basic idea is to provide a vector of INFO "quotas" to nodes. Nodes that deplete their ith INFO quota can transmit the $i+1$st INFO quota as long as they do not interfere with nodes that did not complete their ith INFO quota. This approach can be implemented through the use of multiple rotating INFO signals. However, besides the increased complexity, the improvements in throughput relative to the single INFO case occur only for relatively large quotas. This will imply large delays, which may not be a desirable alternative.

IV. A Unified Implementation

The quota adaptation algorithm may occasionally result in large delays when a previously inactive node receives a large burst of data which cannot be transmitted using only the minimum allocated quota, $Q_{\min}$. In this case, the node can transmit only $Q_{\min}$ in the current cycle. It must then wait to be visited by the SAT in order to put its request on the counter. However, the request will only be granted at the next SAT visit, and meanwhile, the node can again transmit no more than its minimum quota allocation.

The problem described above can be alleviated if the length of time needed for reserving quotas is reduced, which can be achieved by moving the quota allocation counter from the SAT to the INFO signal. The nodes update the counter field in the same manner as before. Since, when there are active nodes on the ring, the INFO signal rotates several times faster than the SAT, the nodes will now have the opportunity to update the counter much more frequently. This increase in the frequency of counter updates can be used to improve the adaptability of the protocol to changing ring loads. For this, however, the way the value of the counter field is used by the nodes has to change as follows.

- In case the SAT arrives at a node before that node has been able to update the quota allocation counter on the INFO signal to reflect its new requirements, i.e., the SAT arrives before the INFO signal, the node will then act as if it had actually been able to make this update, and allocates itself a quota according to the last received counter value plus its own request. The only drawback of this approach is that a node may allocate itself a quota before it actually has been able to reserve it. This means that the total quota allocated to all nodes may be higher than assumed, and in case of congestion, it is important that this inconsistency be corrected quickly. Fortunately, because the INFO signal rotates much faster than the SAT (especially in the case of congestion) as it is not being held, the inconsistency will be corrected quickly.
- The counter value indicates the load on the network. The nodes can take advantage of the increased frequency of counter updates to adjust their quota allocation according to the current load. Specifically, upon receipt of the INFO signal, a node checks if the quota allocation corresponding to the new INFO quota allocation counter yields a smaller value than its current residual quota. If it does, the residual quota is lowered to this new value. This way, if new nodes become active, the increase in counter value will force the rest of the nodes to reduce their quota, and therefore will allow the new users to participate in the quota allocation with reduced HOL delays. Note that, if the received INFO quota allocation counter is lower than the value used by a node when it last allocated itself quotas, the node is not allowed to increase its quota. This is done to avoid increases in delay that may occur in certain scenarios. For example, in the "file server" scenario, the bottleneck node will have long HOL delays if each of the rest of the nodes completes its quota, empties its queue, and therefore decreases the counter value. In this case, the rest of the nodes would increase their quota allocation, thereby blocking the bottleneck node for a longer time.

As illustrated in the next section, this combination of the INFO signal with the quota allocation counter results in better overall performance.

A. Simulation Results

The main conclusions of the simulations are the following. Under light loads, putting the active node counter on the INFO signal made a small difference compared to the algorithm where the quota allocation counter was rotating with the SAT. This difference remained small, even for ring loads of 90%. However, when some nodes attempt to overload the ring while others only have small or medium throughput requirements, the advantage of rapid adjustment of the allocated quotas becomes apparent. In effect, the significance of putting the quota allocation counter on the INFO is that users with small communication requirements are protected from others that may attempt to flood the network.

In order to demonstrate this, we simulate a network where half of the nodes are overloaded, i.e., always have something to transmit, and the loads for the other half are either light (one quarter) or medium (one half). The traffic pattern we considered had transmissions from all nodes headed in the same destination, i.e., a file server or gateway. This scenario was chosen as it stresses delay performance, which is the quantity of interest when estimating the advantages of a rapidly adjusting quota allocation algorithm. However, it should be pointed out that similar results were also observed for other traffic patterns, e.g., the case of a starved node.

The simulation results for this scenario are displayed in Fig. 8. It shows that the statistics of the total[1] delay at nodes with light and medium loads are significantly better when the active node counter is carried on the INFO signal rather than on the SAT signal. When the INFO signal carries the active node counter, the probability that packets sent by a station with low or medium load experience a total delay greater than 16 RTD is negligible (or zero). On the other hand, if the active node counter is carried by the SAT, this probability is above 1%. It is interesting to see that the HOL delay behavior of overloaded nodes is also improved, while their throughput remains the same. (The graphs are taken from the results of seven simulations, each with approximately 70 000, 175 000,

[1] This means from arrival time to the host buffer to arrival of the first bit to the destination.

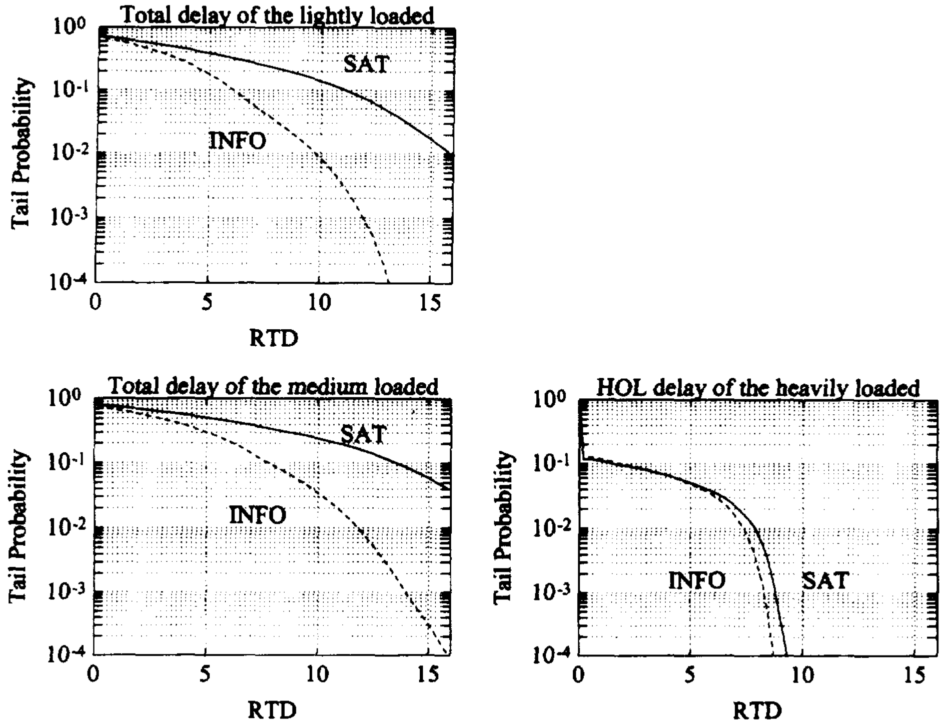


Fig. 8. Unified implementation—all nodes send to a single file server.

and 500 000 packets for nodes with light, medium, and heavy loads respectively.)

V. Error Handling

In this section, we investigate the problem of recovering from various types of errors that can corrupt the control signals on which the algorithms rely. In particular, we show how to protect the algorithms from errors such as signal loss, signal multiplication, and errors in the data carried by the signals.

Signal loss can be easily detected by a time-out mechanism. For the INFO signal that rotates at constant speed, the timeout can be set to $1.5 * T_{RD}$; for the SAT signal, it can be set to 1.5*(max cycle length). Time is measured by a station elected as the leader (leader election is done for other purposes as well, and not especially for this algorithm). If the INFO signal is detected missing, a new INFO signal is generated by the leader, with the quota allocation counter set to zero and an INFO_START bit set, to one. The INFO_START bit indicates to the nodes that this is a new signal with new counter values. A node that receives an INFO signal with the INFO_START bit set, reinitializes its allocation algorithm, i.e., puts r^i on the counter and waits for the SAT to reallocate itself quota. The value used by a node as the current number of active nodes can either be the last one before the INFO loss, or it may decide to wait for a full RTD to receive an updated counter value. Note that the previously described quota adjustment feature ensures that the algorithm recovery is fast and smooth. The leader resets the INFO_START bit to zero after receiving the INFO signal back, but no reinitialization is needed for the hop counter as it simply rebuilds itself when going through each node. SAT loss detection may take longer because of the longer rotation time. Again, as soon as the loss of the SAT is detected by the leader, a new SAT is generated, and the quota allocation algorithm starts anew. Although SAT regeneration may take some time, and in this period no quotas are allocated, the ring remains active (without fairness, though) because of the INFO hop counter mechanism.

SAT multiplication, on the other hand, can result in longer delays for packets of starved nodes. The multiple SAT signals can be merged when two or more reach a starved node, but this process may be slow. Faster solutions can be found in [9]. INFO multiplication is potentially more dangerous from the point of view of the active node counter as it may result in inconsistent increments and decrements of the counter by nodes, i.e., they increment the counter of one signal and perform the decrement on another counter. In order to identify multiple INFO's, we rely on a "random" bit in the INFO signal, which the leader randomly sets to zero or one each time it sees the INFO signal. When the leader receives an INFO signal with a random bit value different from the one it expects, i.e., the one it last set, it discards this INFO signal and sends a new INFO signal with the INFO_START bit set. This is needed to recover from possible inconsistencies in the active node counter values kept at each node. Nodes then reinitialize the allocation algorithm as described above when they receive the new INFO signal. Since the INFO random bit is changed every cycle by the leader, multiple INFO signals are eventually detected (and eliminated) with probability 1.

Error in the hop counter is never a problem since it corrects itself the first time the signal encounters an unsatisfied node. Some errors in the allocation counter are impossible to detect, in particular, if a node increases the counter and then is

TABLE IV
COMPARATIVE PERFORMANCE OF QA AND INFOR MECHANISMS

Mechanism	Better Delay	Max-min Tput
SAT+QA	Yes	No
SAT+INFO	No	Approximately
SAT+QA+INFO	Yes	Approximately

disconnected from the ring. To overcome this, the leader can reinitialize the algorithm every E cycles, where E is a large number that depends on the ring failure probability. Restart of the allocation counter also occurs in a number of other instances, for example, whenever the leader receives a zero allocation counter, or when a node detects a negative value after subtracting its previous allocation. Restarts are again done by setting the INFO_START bit as described before.

VI. SUMMARY

In this paper, we have proposed and evaluated techniques to improve fairness algorithms in ring networks with spatial reuse. We focused on fairness algorithms that rely on the use of transmission quotas to control how stations are allowed to transmit on the ring. Our first goal was to provide greater flexibility in how quotas were allocated so that a better tradeoff between throughput and delay could be achieved. Next, we focused on taking better advantage of traffic locality in the ring, to improve throughput without affecting fairness. Two ideas were presented to achieve these goals. The first is to monitor the demand for access permission through a rotating counter that continuously tracks the number of active nodes on the ring. The second is to use a hop counter that informs nodes about the number of nonstarved downstream nodes. The effectiveness of these improvements was demonstrated by means of simulations which illustrated the gains in access delay and throughput that could be achieved. Table IV shows the comparative performance of the two mechanisms proposed in this paper. The second column represents HOL delay improvement relative to the pure SAT mechanism. Recall that QA stands for the "quota adaptation" mechanism.

APPENDIX

In this Appendix, we prove the HOL delay bounds (2), (3) for Algorithms A and B, respectively. We rely on [1, Theorem 5] which we present below rephrased to conform to our model and notation. Call the node queue of packets that need to be transmitted in the clockwise (counterclockwise) direction, c queue (cc queue). We assume that there is a SAT algorithm operating on the bidirectional ring, but we do not make any assumptions on the quota sizes allocated to each node during a SAT visit.

Theorem 5[1]: Assume that a packet arrives at the head of the line of the c queue (cc queue) at node $i, i = 1, \cdots, n$ at time t_a and it starts being injected in the ring at time t_b. Let $\bar{I}$ be a bound on the number of bits inserted in the c direction (cc direction) of the ring by the rest of the nodes in the interval$[t_a, t_b)$. Then an upper bound on the HOL delay is

$$T_{RD} + 2\left(T_{RD} + \frac{\bar{I}}{B_w} + n\frac{L_{\max}}{B_w}\right).$$

From Theorem 5, we see that, in order to develop bounds on the HOL delay, it is sufficient to determine a bound on the injected traffic while a packet waits at the head of the line. The next two propositions provide such a bound for Algorithms A and B. We first introduce some notation that will be used in the proof of these propositions.

Recall that the SAT rotates on the opposite side of the ring that it regulates. Assume, without loss of generality, that the HOL packet at node i is in the cc queue of that node, and therefore the controlling SAT rotates in the c direction. We also assume that the ordering of nodes is in the c direction of the ring. Let $Q_k(t_a)$ be the quota of node $k, k = 1, \cdots, n$ at time t_a, and let Q_k^a be the additional quota allocated by the SAT to node k during the interval $[t_a, t_b)$. The amount of traffic that can be injected in the cc direction of the ring during $[t_a, t_b)$ is at most

$$\sum_{k=1,k\neq i}^{n} Q_k(t_a) + \sum_{k=1,k\neq i}^{n} Q_k^a.$$

Proposition 1: Under Algorithm A,

$$\bar{I} = 3Q_{\max}.$$

Proof: It is sufficient to show that

$$\sum_{k=1,k\neq i}^{n} Q_k(t_a) + \sum_{k=1,k\neq i}^{n} Q_k^a \leq 3Q_{\max}.$$

At time t_a, let the SAT be on the link between nodes $j-1$ and $j, n \geq j > i$, and let $l(t_a)$ be the number of times that the SAT has visited node i by time t_a since the beginning of system operation. By the operation of the SAT (see the last sentence of the first paragraph in Section II), $Q_k(t_a)$ for $k = j, \cdots, n$ is at most equal to the quota that was allocated to node k during the $(l(t_a) - 1)$th visit of the SAT. Therefore, according to the operation of Algorithm A,

$$\sum_{k=j}^{n} Q_k(t_a) \leq Q_{\max}.$$

Also, $Q_k(t_a)$ for $k = 1, \cdots, j-1$ is at most equal to the quota that was allocated to node k during the $l(t_a)$th visit of the SAT. Observe next that, since a packet waits at the head of the line at node i, the SAT can rotate around the ring at most once in the interval $[t_a, t_b)$. Hence, Q_k^a for $k = j, \cdots, n$ is equal to the quota that will be allocated to node k during the $l(t_a)$th visit of the SAT, and therefore,

$$\sum_{k=1,k\neq i}^{j-1} Q_k(t_a) + \sum_{k=j}^{n} Q_k^a \leq Q_{\max}.$$

Finally, Q^a_k for $k = 1, \cdots, i$ is equal to the quota that will be allocated to node k during the $(l+1)$th visit of the SAT, while $Q^a_k = 0$ for $k = i+1, \cdots, j-1$ since the SAT will not visit such a node before time t_b. Therefore,

$$\sum_{k=1}^{j-1} Q^a_k \le Q_{\max}.$$

From the previous inequalities, we conclude that

$$\sum_{k=1, k\neq i}^{n} Q_k(t_a) + \sum_{k=1, k\neq i}^{n} Q^a_k \le 3Q_{\max}$$

as desired. The cases where the SAT is on the link between nodes $j-1$ and $j, j \le i$ or where the SAT is being held by one of the nodes can be treated similarly. ■

Proposition 2: Under Algorithm B,

$$\bar{I} = 2(\ln\ n + 1)Q_{\max}.$$

Proof: To simplify the argument, assume without loss of generality that $i = 1$. Let the SAT be between nodes $j-1$ and j at time t_a. Also, let $\ell(t_a)$ be the number of times the SAT visited node $j-1$ since the beginning of system operation, and let t_0 be the time that the SAT was released by node $j-1$ for the $(\ell(t_a)-1)$st time. Note that $Q_k(t_a)$ is at most equal to the quota allocated to node k by the SAT visit in the interval $[t_0, t_a)$. Let the value of the SAT counter CT at time t_0 be $m, 1 \le m \le n$. Then, according to Algorithm B, only m of the nodes will be allocated a quota by the SAT visit in the interval $[t_0, t_a)$. This is so since only these m nodes have registered requests for a quota on CT at time t_0. Let $\mathcal{M} \neq \emptyset$ be the set of these m nodes. Also, let the kth of the nodes in $\mathcal{M}$, in the direction of SAT rotation, be node l. The value of CT seen by node l is at least $m-(k-1), 1 \le k \le m$. The value $m-(k-1)$ will occur if none of the nodes in $\{j, \cdots, l-1\}$ registers new requests between time t_0 and the time the SAT visits node l. This implies that the nodes in $\mathcal{M} \cap \{j, \cdots, l-1\}$ subtract one from CT, while the rest of the nodes in $\{j, \cdots, l-1\}$ do not increase CT. Therefore,

$$\begin{aligned}\sum_{k=2}^{n} Q_k(t_a) &\le \left(\sum_{k=1}^{m} \frac{1}{k}\right) Q_{\max} \\ &\le \left(\sum_{k=1}^{n} \frac{1}{k}\right) Q_{\max} \le (\ln\ n + 1)Q_{\max}.\end{aligned}$$

Similarly, if m' is the value of the SAT counter CT when the SAT is released by node $j-1$ for the $\ell(t_a)$st time, we have

$$\sum_{k=j}^{n} Q^a_k \le (\ln\ n + 1)Q_{\max}.$$

Taking into account the fact that $Q^a_k = 0$ for $k = 2, 3, \cdots, j-1$, we conclude

$$\sum_{k=2}^{n} Q_k(t_a) + \sum_{k=2}^{n} Q^a_k \le 2(\ln\ n + 1)Q_{\max}.$$

The cases where the SAT is held by some node at time t_a, or $\mathcal{M} = \emptyset$, are treated similarly. ■

Combining Theorem 5 with Propositions 1 and 2, we derive the bounds (2) and (3). Also, bound (1) follows by arguments analogous to those used in Proposition 2.

REFERENCES

[1] B. R. Bellur and G. H. Sasaki, "A SAT-based network access scheme for fairness in high speed networks," preprint, 1996.

[2] H. R. van As, W. W. Lemppenau, P. Zafiropulo, and E. A. Zurfluh, "CRMA-II: A Gbits/sec MAC protocol for ring and bus networks with immediate access capability," IBM Zürich Res. Lab., Tech. Rep. RZ 2129, May 1991.

[3] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[4] S. Breuer and T. Meuser, "Enhanced throughput in slotted rings employing spatial reuse," in *Proc. IEEE INFOCOM'94*, Toronto, Ont., Canada, pp. 1120–1129.

[5] W. Bux, "Local area subnetworks: A performance comparison," *IEEE Trans. Commun.*, vol. COM-29, pp. 1465–1473, Oct. 1981.

[6] ——, "Performance issues in local-area networks," *IBM Syst. J.*, vol. 23, no. 4, 1984.

[7] J. S.-C. Chen and R. Guérin, "On issues related to real-time applications support in the ORBIT gigabit/sec ring," in *Proc. 2nd Int. IFIP Conf. Broadband Commun. (BB'94)*, Paris, France.

[8] I. Cidon, I. Gopal, and R. Guérin, "Bandwidth management and congestion control in plaNET," *IEEE Commun. Mag.*, vol. 29, pp. 54–63, Oct. 1991.

[9] I. Cidon and Y. Ofek, "Metaring—A full duplex ring with fairness and spatial reuse," *IEEE Trans. Commun.*, vol. 41, pp. 110–120, Jan. 1993. (See also *Proc. IEEE INFOCOM'90.*)

[10] ——"Distributed fairness algorithms for local area networks with concurrent transmissions," in *Distributed Algorithms, 3rd Int. Workshop*, Nice, France. Springer-Verlag, Lecture Notes in Computer Science 392, Sept. 1989, pp. 57–69.

[11] J. Chen, I. Cidon, and Y. Ofek, "A local fairness algorithm for gigabit LAN's/MAN's with spatial reuse," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1183–1192, Oct. 1993.

[12] R. M. Falconer and J. L. Adams, "Orwell: A protocol for an integrated service local network," *Br. Telecom Technol. J.*, vol. 3, pp. 21–35, Apr. 1985.

[13] M. W. Garrett and S.-Q. Li, "A study of slot reuse in dual bus multiple access networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 248–256, Feb. 1991.

[14] L. Georgiadis, R. Guérin, and I. Cidon, "Throughput properties of fair policies in ring networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 718–728, Dec. 1993.

[15] R. Gvozdanovic, "A review of high performance protocols for FDDI follow on LAN," *Proc. 10th Annu. EFOC/LAN'92 Conf.*, Paris, France, pp. 81–88.

[16] P. Heinzmann, H. R. Müller, D. A. Pitt, and H. R. van As, "Buffer-insertion cell-synchronized multiple access (BCMA) on a slotted ring," in *Proc. 2nd Int. Conf. Local Commun. Syst.: LAN and PABX*, Palma, Balearic Islands, Spain, June 1991.

[17] A. Hopper and R. M. Neddham, "The Cambridge fast ring networking system," *IEEE Trans. Comput.*, vol. 37, pp. 1214–1223, Oct. 1988.

[18] D. E. Huber, 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.

[19] M. J. Karol and R. D. Gitlin, "High-performance optical local and metropolitan area networks: Enhancements of FDDI and IEEE 802.6 DQDB," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1439–1448, Oct. 1990.

[20] D. Picker and R. D. Fellman, "Enhancing SCI's fairness protocol for increased throughput," presented at the *IEEE Int. Conf. Network Protocols*, Oct. 1993.

[21] M. A. Rodrigues, "Erasure node: Performance improvements for the IEEE 802.6 MAN," in *Proc. IEEE INFOCOM'90*, San Francisco, CA, pp. 636–643.

[22] Y. Shavitt, "Distributed algorithms for ring networks," (in Hebrew), Master's thesis, Dep. Elec. Eng., Technion—I.I.T., Haifa, Israel, June 1992.

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

From 1984 to 1985, he was with the Faculty of the Department of Electrical Engineering, Technion. In 1985, he joined the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, where he was a Research Staff Member and the Manager of the Network Architectures and Algorithms group involved in various broad-band networking projects such as the Paris/Planet Gigabit testbeds, the Metaring/Orbit Gigabit LAN, and the IBM Broad-Band Networking architecture. In 1994 and 1995, he was with Sun Microsystems Labs, Mountain View, CA, as Manager of High-Speed Networking, working on various ATM projects including OPENET—an open and efficient ATM network control platform. Since 1990, he has also been with the Department of Electrical Engineering, Technion. His research interests are in high-speed wide- and local-area networks, distributed network algorithms, network performance and mobile networks.

Dr. Cidon is a Founding Editor of the IEEE/ACM Transactions on Networking. Previously, he served as the Editor for Network Algorithms for the IEEE Transactions on Communications and as a Guest Editor for *Algorithmica*. In 1989 and 1993, he received the IBM Outstanding Innovation Award for his work on the PARIS high-speed network and topology update algorithms respectively.

Leonidas Georgiadis (SM'95) received the Diploma degree in electrical engineering from Aristotle University, Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees, both in electrical engineering, from the University of Connecticut, Storrs, in 1981 and 1986, respectively.

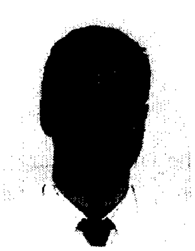
From 1981 to 1983, he was in the Greek Army. From 1986 to 1987, he was a Research Assistant Professor at the University of Virginia, Charlottesville. In 1987, he joined the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member. Since October 1995, he has been with the Telecommunications Department, Aristotle University, Thessaloniki, Greece. His interests are in the areas of high-speed networks, congestion control, mobile communications, scheduling, modeling, and performance analysis.

Dr. Georgiadis is a member of the IEEE Communications Society. In 1992, he received an IBM Outstanding Innovation Award for his work on goal-oriented workload management for multiclass systems.

Roch Guérin (SM'91) received the "Diplôme d'Ingénieur" from the École Nationale Supérieure des Télécommunications, Paris, France, in 1983, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, both in electrical engineering, in 1984 and 1986, respectively.

Since 1986, he has been with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, where he manages the Broadband Networking Department. His current research interests are in the areas of performance analysis and traffic modeling, scheduling policies, and in general quality-of-service issues in high-speed networks. He has recently been leading a project developing an experimental system integrating switching and routing functions, with a particular emphasis toward supporting a wide range of QoS offerings.

Dr. Guérin is a member of Sigma Xi and the IEEE Communications Society, and is an Editor for the IEEE/ACM Transactions on Networking. He was an Editor for the IEEE Transactions on Communications and the IEEE Communications Magazine. In 1994, he received an IBM Outstanding Innovation Award for his work on traffic management in the BroadBand Networking Services Architecture.



Yuval Shavitt (M'97) received the B.Sc. degree in computer engineering (cum laude), the M.Sc. degree in electrical engineering, and the D.Sc. degree from the Technion—Israel Institute of Technology, Haifa, in 1986, 1992, and 1996, respectively.

From 1986 to 1991, he served in the Israel Defense Forces, first as a System Engineer, and the last two years as a Software Engineering Team Leader. He spent the summer of 1992 as a summer student at the IBM T. J. Watson Research Center, Yorktown Heights, NY. Currently, he is a Post-Doctoral Fellow in the Department of Computer Science, The Johns Hopkins University, Baltimore, MD. His research interests include networks and distributed algorithms.