Improved Fairness Algorithms for Rings with Spatial Reuse

I. Cidon, L. Georgiadis, R. Guérin

IBM T. J. Watson Research Ctr.
P.O. Box 704
Yorktown Heights, NY 10598

Y. Shavitt

Department of Electrical Engineering
Technion, Haifa 32000
Israel

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 achieve fairness by allocating transmission quotas to nodes. For such policies, we provide mechanisms for improving delays and increasing overall throughput without compromising fairness.

1 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 [3, 4]. 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/MANs.

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 [15, 10, 14, 8, 13], and for upgrades of existing standards to support this feature [17, 16, 11].

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. 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., non-preemptive 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 [10] for slotted rings and in [9, 8] for buffer insertion and slotted rings. Both mechanisms are based on the idea of 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 [5]. While the objectives of achieving high throughput and low HOL delays are often conflicting, a reasonable trade-off is demonstrated here through a number of improvements to the distributed fairness algorithm in [8], which we describe below.

In [8] a control signal, the SAT (which stands for SATisfied), rotates around the ring and grants a transmission quota to each node it visits. 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. The SAT algorithm is effective not only at preventing nodes from starving but also at ensuring that they share ring bandwidth fairly and effectively, i.e., full utilization of the maximally loaded link equally shared between nodes [12]. The SAT algorithm has, however, a number of shortcomings with respect to access delay (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 SAT rotation cycle is determined by the most heavily loaded links on the ring [12] 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 mechanism to address each of the above mentioned problems of the SAT algorithm. The first mechanism, which is studied in Section 2, consists of a distributed algorithm for adapting the size of the quotas, which both preserves fairness and keeps ring access below a predetermined and easily programmable time. The second mechanism (Section 3) uses a single control signal called INFO that informs nodes about the 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 4. 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. In Section 5, we consider the case where both asynchronous and synchronous traffics coexist on the same ring, and investigate a simple approach that builds on the new mechanisms introduced in previous sections to accommodate both traffic types.

2 Improving Packet Delays

Throughout this paper, we consider a bidirectional ring (extensions to a unidirectional ring are addressed in [6]) with n nodes, that employs the SAT algorithm described in [8]. 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 that a packet travels is $\lceil n/2 \rceil$.

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), Q_{RTD}. 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 the nodes wish to transmit over the same link. Under the previous allocation, the node directly upstream of the bottleneck link may be prevented from inserting its own packets for as long as $2 \times (\lceil n/2 \rceil - 1) \cdot RTD$. This is because in the worst case, this node may have to wait until all the other nodes transmit their quotas from two consecutive SAT cycles. Using the SAT algorithm in a Ring with 200 nodes and a round trip delay of 0.5 milliseconds (which corresponds to a ring size of 100km) this worst case delay can be as long as 100 milliseconds, which may be unacceptable for some applications. It is clear from the above discussion, that the quota allocated to the nodes should be adapted to reflect the traffic characteristics. We provide next a mechanism by which this quota adaptation is achieved.

A *counter field* is added to the SAT. When node i receives the SAT, it saves the counter value in a register, CV, increases the counter by a value, r^i, which reflects its current quota requirements, and decreases it by the value, r_p^i, requested at the previous SAT visit. Upon release of the SAT, node i allocates itself quota $\nu^i = (r_p^i / CV) \times Q_{MAX}$. CV contains the sum of the requests registered by all nodes between two successive visits of the SAT to node i, so that quotas are in effect allocated to nodes in proportion to their requests. Since quota allocation to a node depends on the request it registered in the previous SAT release, it is possible (when $r_p^i = 0$) that it be prevented from transmitting for up to two SAT cycles even though it has messages to transmit. To avoid such potentially long delays, the SAT grants a small quota Q_{MIN} (typically a few Kbytes) 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.

Given Q_{MAX} and Q_{MIN}, the maximum cycle length and the maximum time a packet will wait at the head of its host buffer can be easily calculated. Specifically, the maximal cycle length will occur in the following scenario. Only one node has a request for the current cycle and each of the rest of the nodes has an empty queue. Just before the SAT arrives to each one of these $n-1$ nodes, it receives packets to transmit and it, thus, hold the SAT and transmits for a duration of Q_{MIN}/BW. The maximum amount of data in a SAT cycle is transmitted in this scenario. The cycle length in this case is (for both bidirectional and unidirectional rings):

$$\frac{Q_{MAX}}{BW} + (n-1)\frac{Q_{MIN}}{BW}.$$

The maximal time a packet waits at the head of its host buffer can occur when all nodes transmit their packets over the same node, i. If we make the reasonable assumption that the size Q_{MIN} is at least the size of one maximum packet, every node can transmit at least one packet in a SAT cycle. Under this conditions, in the worst case a packet at the head of the host buffer at node i will have to wait until the rest of the nodes exhaust the quota allocated to them in two SAT cycles and the maximal channel access time is

$$2\left(\frac{Q_{MAX}}{BW} + \left(\left\lceil\frac{n}{2}\right\rceil - 2\right)\frac{Q_{MIN}}{BW}\right).$$

Using these bounds, the system manager can select Q_{MAX} and Q_{MIN} and tune the network based on the delay restrictions of the applications. For example, with $Q_{MAX} \geq Q_{RTD}$ a single station can fully utilize the channel, but when the number of active stations increases the quota allocated to each node decreases in proportion as the sum of all allocated quotas remains constant. While this method of quota adaptation successfully upper bounds HOL delays and provides satisfactory throughput for many traffic patterns, there exist some patterns with potential for high spatial reuse, for which the ring may be under-utilized. Consider, for example, a ring for which $Q_{MAX} = 10Q_{RTD}$, and with 20 active stations which only transmit to their downstream neighbors. Each station receives a quota $\nu^i = .5 \times Q_{RTD}$, while they should get Q_{RTD} to fully utilize their link. In Section 3 we describe a mechanism that helps to avoid such throughput losses.

2.1 Simulation Results

To study the properties of the quota adaptation mechanism, we developed a detailed simulation model of the buffer insertion ring. All the simulation results in this section are for the case were $r^i \in \{0, 1\}$, and

$$\nu^i = \begin{cases} \frac{Q_{MAX}}{\sum_{j=1}^{n} r^j} & \text{for nodes with } r^i = 1 \\ Q_{MIN} & \text{for nodes with } r^i = 0 \end{cases}$$

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 the head of line delay (HOL delay), or the interval between the arrival of a packet to the head of its node buffer and the time its transmission begins.

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

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, #39, is never a destination. 3)**A File Server.** All nodes transmit to one specific node (a file server).

2.1.1 Uniformly Distributed Destination.

Figure 1 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_{RTD}. 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 is decreasing with Q_{MAX}. The decrease, however, is less than 10% for Q_{MAX} values over $6Q_{RTD}$.

Tail probability: We plot here the tail probabilities of the HOL delay for the SAT algorithm when it operates with fixed quota per node equal to Q_{RTD}, as well as when the quota adaptation algorithm is used with Q_{MAX} values of 4, 6, and 9 Q_{RTD}. The portion of packets with HOL delay of more than 4 RTD is (close to) zero for the quota adaptation algorithm under all 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.

2.1.2 A Deprived Node

For nodes other that #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 the tested values of Q_{MAX}. Under the SAT algorithm, about 5% of the packets of node #39 have HOL delays larger than 5RTD, while the percentage is (almost) zero when the quota adaptation algorithm is used. Details of this and the next experiment can be found in [6].

2.1.3 Communication with a File Server

Since there is no spatial reuse in this case, the throughput is the same under both algorithms. The tail probabilities are now larger under both algorithms, since this is the scenario under which maximal HOL delays may occur. However, through the use of Q_{MAX} the HOL delay is again effectively controlled when the quota adaptation algorithm is used.

3 Improving System Throughput

3.1 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 mostly 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 [2], where each node gets the largest possible throughput that does not impact nodes with lower throughputs.

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 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 1 Assume that we have a unidirectional ring with 4 nodes and that the direction of traffic is from node 1 to node 4 (see figure 2). 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 $1-\varepsilon$ and to node 4 with probability $\varepsilon \geq 0$.
It is easy to see that if $\varepsilon > 0$, the max-min fair vector

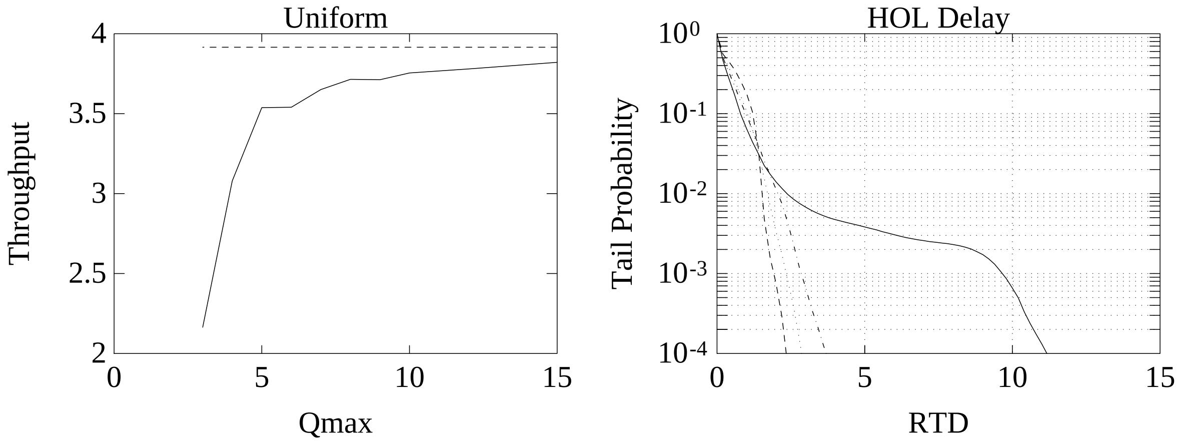

Figure 1: Destination is uniformly distributed

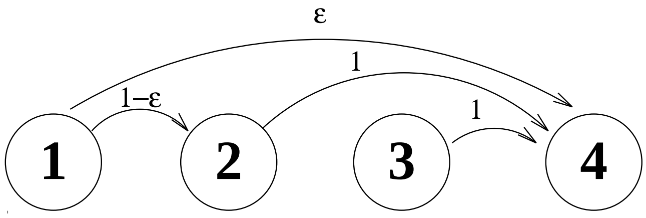

Figure 2: example

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 $\{1, .5, .5, 1\}$, while the limit obtained by considering the max-min fair points for $\varepsilon > 0$ is $\{.5, .5, .5, 1\}$.

When all nodes have equal quota, it is shown in [12] 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 throughput of some nodes without penalizing others is lost. The quota adaptation algorithm improves the HOL delays of the SAT algorithm but cannot correct this throughput problem. It can be shown [12], that the problem can be solved by allocating different quotas to nodes as a function of their traffic distribution. This may, however, not be practical, and 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 last example is also largely corrected.

3.2 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 forwards 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.

3.3 Simulation Results

We consider the following assymetrically 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 following scenario. Node 1 as well as nodes 11 through 17 send all their traffic to node 18 Nodes 3, 4, 9, 10 and 11 send all their traffic to their down stream neighbor. Hence, $q_3^4 = q_4^5 = q_9^{10} = q_{10}^{11} = q_{18}^1 = 1$. Node 2 splits its traffic equally to nodes 3 and 4, hence $q_2^3 = q_2^4 = 0.5$. Node 5 splits its traffic among equally to nodes 6 through 10, hence, $q_5^j = .2 : \ j = 6, \ldots, 10$, And similarly, $q_6^j = .25 : \ j = 7, \ldots, 10$, $q_7^j = .33 : \ j = 8, 9, 10$, $q_8^9 = q_8^{10} = .5$. Note that all the packets generated by node 1 have to traverse nodes 2 to 17. Table 1 shows the results obtained from a simulation of the ring with this traffic pattern. The first column gives the max-min fair throughputs of all nodes for reference. Columns 2, 3 and 4 give the throughputs obtained for the SAT algorithm with quota Q_{RTD}, and for the quota adaptation algorithm with $Q_{MAX} = 4, 9$, respectively. The INFO signal was used in both cases. Note that if the SAT algorithm was 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, 3, 4). Furthermore, under the SAT algorithm, increasing the quota allocation of all nodes beyond Q_{RTD} 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 simulated next the network of Example 1 in Section 3.1 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 very close (for most Q_{MAX} values) to those obtained with the SAT algorithm. Therefore, Table 2 presents only the results obtained when using the SAT algorithm with quota equal to Q_{RTD}. The first row gives the max-min fair point for $\varepsilon = 0.01$. Without the INFO, the throughput of node 4 is reduced to .4899. When the INFO is used, the throughput of node 2 and 3 is reduced slightly (about .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 ($\{1,.495,.495,1\}$). 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.

While the INFO signal can provide increased throughput compared to the SAT with or without the quota adaptation mechanism, it does not provide guarantees that this extra throughput will be allocated fairly to the nodes. It is possible to generalize the INFO algorithm to ensure greater fairness in allowing transmissions without quotas, so that throughputs are closer to 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.

4 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. This problem 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. When there are active nodes on the ring the INFO signal rotates several times faster

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

Table 1: Throughputs for the traffic pattern of Section 3.3.

	Node 1	Node 2	Node 3	Node 4
MAX-MIN FAIR	0.4975	0.4975	0.4975	1.0
SAT without INFO	0.4899	0.4899	0.4899	0.4899
SAT with INFO	0.9177	0.4877	0.4877	1.0

Table 2: $\varepsilon = 0.01$

than the SAT. In this situation, the nodes can reserve quotas much faster, and typically allocate themselves the desired quota at the next SAT visit. In case the SAT arrives at a node before it has had a chance to update the quota allocation counter on the INFO, the node will 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. Allocation of quotas at other nodes will then have to be adjusted to reflect this increase in demand. This is achieved by requiring that nodes lower their quota to match the current system load, each time they receive a new value of the quota allocation counter. 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. 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 as this can result in significant delay increases for other nodes in a number of scenarios.

4.1 Simulation Results

The details of the simulation can be found in [6] The main conclusions 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 ring for 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.

5 Integration

In this section we address the problem of integrating two types of traffic with different service requirements: (i) synchronous traffic which requires guaranteed bandwidth, and (ii) asynchronous traffic with less rigid constraints on packet delay. In addition to having different service requirements, these two types of traffic also differ in their arrival patterns, with synchronous traffic having regular arrivals and asynchronous traffic being typically bursty in nature. Both types of traffic will coexist in broadband integrated networks. It is, therefore, important to understand, in the context of the ring network studied in this paper, how these two traffic types can efficiently share the ring bandwidth while being provided with the appropriate grade of service. There are two main aspects to this issue, that reflect the different requirements of the two types of traffic: (1) Guarantee throughput and delay for synchronous traffic; (2) Minimize the impact of (1) on the throughput and delay of asynchronous traffic.

There are various approaches to achieve such traffic integration, that provide different trade-offs between complexity and overall efficiency. In this section we limit ourselves to a simple method that ensures good performance to synchronous traffic with only limited throughput degradation for asynchronous traffic. This approach can be implemented without modification to the ring protocol described in the previous sections. As a result, the approach is somewhat conservative as it focuses on "protecting" synchronous traffic by assuming worst case conditions for its generation. However, because of the new INFO mechanism available on the ring, the impact of this conservative approach on asynchronous traffic is limited. We now proceed with a brief description of our traffic integration scheme and identify its key characteristics.

Synchronous sessions will reserve bandwidth in a way similar to that described in [8]. Session j of node i computes the amount of bandwidth α_{ij} it needs and proceeds to reserve this bandwidth on all the associated links. We limit the amount of bandwidth allocated to synchronous traffic on any link to no more than α_{MAX}. This is imposed to ensure that interferences between synchronous traffic from different sessions remain minimal, so that the delay and jitter experienced by synchronous traffic are kept low. Another motivation for limiting the maximum amount of synchronous traffic on any link is to maintain adequate asynchronous throughput on the ring. In a typical LAN environment a reasonable value for α_{MAX} is around 0.5.

The asynchronous traffic uses the quota allocation algorithm described in the previous section. The major modification introduced by the presence of synchronous traffic is in how the total ring quota allocation is determined, i.e., the quantity Q_{MAX} that is shared among all nodes. The value used for this total quota allocation determines the actual volume of asynchronous traffic transmitted during one rotation of the SAT signal. In order to account for the presence of synchronous traffic, the total quota allocation is reduced in proportion to the volume of this traffic. Specifically, the total ring quota to be allocated among nodes for transmission of asynchronous traffic is set equal to $(1-\alpha)Q_{MAX}$, where α is the maximum amount of bandwidth allocated to synchronous traffic on any ring link. The value of α is assumed known to all nodes through some standard control mechanism such as the utilization update algorithm of [7]. It could also be obtained using an algorithm based on two rotating fields, such as the Token Ring priority algorithm [1]. In addition to protecting synchronous traffic by limiting the amount of asynchronous traffic that can compete with it, we also give priority to synchronous packets over asynchronous ones at each node. This means that the presence of synchronous packets waiting for transmission prevents the transmission of any local asynchronous packet (even if quotas are present) until the synchronous queue is empty. Furthermore, a non-empty synchronous queue is also viewed as an indicator of "node starvation" and will result in the SAT being held. Similarly, it will also trigger the resetting of the INFO hop counter. Note that all this assumes the use of some control mechanism, e.g., leaky bucket, to ensure that the volume of synchronous traffic is limited to its agreed upon intensity.

It should be apparent from the above discussions that the proposed traffic integration method heavily favors synchronous traffic, and should therefore easily guarantee that it sees adequate performances. This has indeed been verified in the many simulations scenarios we have considered in [6]. The more important issue is then to assess the impact of this approach on the level of performance seen by asynchronous traffic. The simulation experiments concentrated, therefore, on understanding the performance implications of the proposed mechanism on the asynchronous traffic. The results show that while asynchronous throughput is slightly degraded it still remains at a reasonable level thanks to the availability of the INFO mechanism (see [6]).

References

[1] Token ring access method and physical layer specification. ANSI/IEEE Standards 802.5-1985 (ISO DIS 8802/5).

[2] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, second edition, 1992.

[3] W. Bux. Local area subnetworks: A performance comparison. *IEEE Trans. Commun.*, COM-29(10):1465–1473, October 1981.

[4] W. Bux. Performance issues in local-area networks. *IBM Systems Journal*, 23(4), 1984.

[5] J. S.-C. Chen and R. Guérin. On issues related to real-time applications support in the orbit gigabit/sec ring. In *Proc. 2nd Intl. IFIP Conf. Broadband Commun. (BB'94)*, Paris, France, March 1994. (See also IBM Research Report RC 18234, August 1992).

[6] I. Cidon, L. Georgiadis, R. Guérin, and Y. Shavitt. Improved fairness algorithms for rings with spatial reuse. Technical report, IBM T. J. Watson Research Center, 1994.

[7] I. Cidon, I. Gopal, and R. Guérin. Bandwidth management and congestion control in plaNET. *IEEE Commun. Mag.*, 29(10):54–63, 1991.

[8] I. Cidon and Y. Ofek. Metaring – a full duplex ring with fairness and spatial reuse. *IEEE Trans. Commun.*, 41(1):110–120, January 1993. (See also Proc. INFOCOM'90).

[9] Israel Cidon and Yoram Ofek. Distributed fairness algorithms for local area networks with concurrent transmissions. In *Distributed Algorithms 3rd International Workshop, Nice, France*, pages 57 – 69. Springer-Verlag, September 1989. Lecture Notes in Computer Science 392.

[10] R. M. Falconer and J. L. Adams. Orwell: A protocol for an integrated service local network. *Br. Telecom Technol. J.*, 3(4):21–35, April 1985.

[11] M. W. Garrett and S.-Q. Li. A study of slot reuse in dual bus multiple access networks. *IEEE J. Select. Areas Commun*, 9(2):248–256, February 1991.

[12] L. Georgiadis, R. Guérin, and I. Cidon. Throughput properties of fair policies in ring networks. *IEEE Trans. Networking*, 1(6), December 1993.

[13] 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. Second Intl. Conf. Local Commun. Sys.: LAN and PABX*, Palma, Balearic Islands, Spain, June 26–28 1991.

[14] Andrew Hopper and Roger M. Neddham. The cambridge fast ring networking system. *IEEE Transactions on Computers*, 37(10):1214 – 1223, October 1988.

[15] D. E. Huber, W. Steinlin, and P. J. Wild. SILK: An implementation of a buffer insertion ring. *IEEE J. Select. Areas Commun.*, SAC-1(5):766–774, November 1983.

[16] 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*, 8(8):1439–1448, October 1990.

[17] M. A. Rodrigues. Erasure node: Performance improvements for the IEEE 802.6 MAN. In *Proc. INFOCOM'90*, pages 636–643, San Francisco, CA, June 1990.