A Steady State Bound for Resilient Packet Rings

*Changcheng Huang, +Harry Peng, *Fengjie Yuan, and +John Hawkins
* Carleton University, + Nortel Networks
huang@sce.carleton.ca

***Abstract*-Resilient Packet Ring (RPR) is a new technology being standardized in the IEEE 802.17 working group. This paper presents the performance of 1TB-RPR, a proposal made to the working group. 1TB-RPR deploys a rate-based fairness algorithm rather than the quota-based algorithm that has been widely used in legacy ring schemes. This fairness algorithm significantly reduces access delays under steady state and allows RPR to be scalable in today's MAN/WAN environments. A ring access delay bound under steady state is given. The bound is then proved using both analytical and simulation approaches. Furthermore we show that the bound is tight by constructing a worst-case traffic scenario. It is shown that straight overloading scenarios may not be the worst case.**

I. INTRODUCTION

A Resilient Packet Ring (RPR) [1] network is a ring-based architecture that consists of two counter-rotating rings with each station connecting to two adjacent stations over a link pair. In the past three decades, various ring technologies have been proposed in literature and some of them have been standardized [2,3,4]. Token ring [2], for example, is one of the earliest ring protocols that have been standardized. In a token ring network, a token is passed from node to node. A node is allowed to transmit packets only when it holds the token. Token ring networks have two main characteristics:

1) A packet on the ring can only be removed from the ring at its source node, an approach called source stripping.

2) A node can release the idle token only when the packet it transmitted has returned.

Several proposals [4] were made to incorporate the spatial reuse concept in the early 90's based on the buffer insertion ring technology, where a small buffer was inserted on the transit path in each node. In these schemes, a node can transmit packets as long as its insertion buffer is empty. Because packets are removed from the ring at their destination nodes rather than source nodes, spatial reuse can be achieved. This approach clearly gives the transit traffic stream higher priority than local add-in traffic streams. Downstream nodes may therefore suffer the so-called starvation problem if upstream nodes keep bursting traffic. To solve this problem, fairness algorithms must be implemented in association with buffer insertion ring technology. It is highly desirable that a good fairness algorithm should maximize throughputs and minimize access delays. There are in general two types of fairness algorithms [5]: global fairness vs. local fairness. MetaRing [3], a well-known scheme that supports spatial reuse, deployed a global fairness scheme at the beginning. In this scheme, a control signal SAT (which stands for SATisfied) rotates around the ring. The quota of a node is renewed every time SAT visits the node. A node can transmit its local traffic whenever its insertion buffer is empty and it has not exhausted its quota. Access delays can be reduced by adaptively assigning quota based on information about downstream nodes carried by SAT. The major drawback of this global fairness is that quotas can only be renewed every round trip through the whole ring. As shown in [6], the maximum access delays are within the order of round trip times. When the ring network is overloaded, the access delays seen by each node will oscillate between zero and the maximum value depending on when a packet comes in relative to the recent SAT visit. A local fairness algorithm as discussed in [5] can reduce the maximum access delays from the round trip of a ring to the round trip of a congestion span only if traffic is non-uniform. It also suffers from the oscillation between zero and maximum value due to its quota-based nature.

The RPR scheme discussed in this paper tries to solve the oscillation problem by using a rate-based control approach rather than the quota-based one adopted by all the aforementioned schemes. It divides a congestion period into two stages: transit and steady state. In general the transit behavior of a RPR network is similar to MetaRing with local fairness. But the access delays under steady state are significantly smaller than transit state and they do not depend on either the ring size or the size of the congestion span. This difference may not be useful if congestion periods are short. But it is well known that Internet traffic shows strong self-similar nature, where congestion periods are typically long and sustained [7,8,9]. The improvement over access delays under steady state allows RPR to scale to a much larger ring sizes (e.g. 2000km) and much higher ring speeds (e.g. 10 Gbps or above) so that it can be applied to MAN/WAN applications. This is the major driver for RPR technology because it is hard to compete with Ethernet for any new technologies in LAN environments.

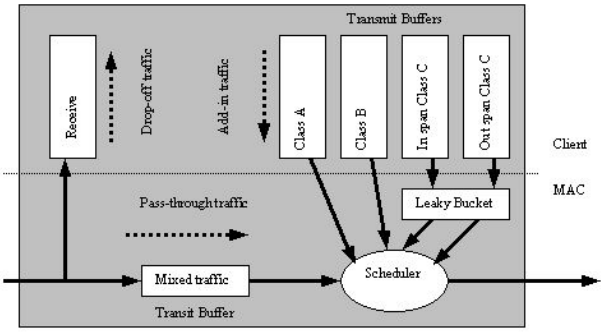

Fig. 1. A typical 1TB-RPR node.

0-7803-7975-6/03/$17.00 (C) 2003

In this paper, a bound for the access delays under steady state is developed. It will be shown that the bound is much smaller than the bounds in [6].

II. THE 1TB-RPR SCHEME

A typical 1TB-RPR node is shown in Fig. 1. The 1TB-RPR scheduling algorithm is also based on buffer insertion ring technology where pass-through packets always have priority over add-in packets from the transmit buffers. Because the pass-through traffic has absolute priority over the add-in traffic, only a very small transit buffer (1 or 2 packets) is required. This significantly simplifies the hardware implementation of the MAC. But on the other hand all add-in traffic streams may experience ring access delays. For high priority traffic (Class A and Class B) this ring access delay contributes a delay jitter that must be minimized. To reduce the ring access delay, a fairness algorithm based on feedback control is designed to control the access of the total bandwidth for all nodes during periods of congestion. To detect congestion, the fairness algorithm uses two trigger conditions: one triggered by a high utilization, and one triggered by a high ring access delay. When congestion is detected, the 1TB-RPR-fairness algorithm uses explicit congestion notifications to manage bandwidth on the ring so that weighted bandwidth fairness is achieved and ring access delays are minimized. The weight assigned to a node represents how much bandwidth the node requires for low priority traffic during periods of congestion.

AS shown in Fig. 1, the scheduler chooses data packets from five queues:

1) Packets from transit buffer.
2) Packets from Class A transmit buffer.
3) Packets from Class B transmit buffer.
4) Packets from in-span Class C transmit buffer.
5) Packets from out-of-span Class C transmit buffer.

Class A and Class B traffic is engineered according to a Committed Information Rate (CIR) and is not subject to the advertised rates generated by the fairness algorithm in downstream nodes. A Class C traffic flow is regulated by a token bucket where its token rate is controlled by the advertised rate received from the down stream nodes during periods of congestion and its bucket size is pre-configured to achieve the maximum throughput.

Similar to [5], the 1TB-RPR scheme is a local fairness algorithm where fairness is regulated over a congestion span rather than the whole ring. This can have an improvement if traffic is non-uniform. Fig.2 gives an example of a congestion span, which is defined as the span of all nodes contributing to the congestion on a link. A congestion span typically consists of a head node, several chain nodes and a tail node. A node that detects a congested outgoing link is defined as the head node. The head node knows the whole congestion span because each node tracks the IDs of all source nodes with traffic passing through it. Based on the utilization of its outgoing link, which is also called a downstream link, the head node

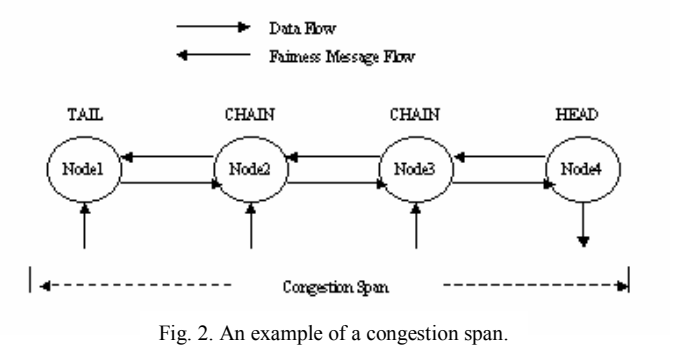

Fig. 2. An example of a congestion span.

calculates a fair rate for Class C traffic and then advertises it to the upstream nodes in the span when congestion happens. The initial advertised rate is normalized by the sum of all the weights assigned to the nodes within the congestion span. Having received the normalized advertised rate from the downstream node, each node calculates its target rate by multiplying the normalized advertised rate with its own weight and then applying the rate to its leaky bucket for the out-of-span Class C traffic. Assume there be N nodes (node 1 – node N) in a congestion span. Let ρ_i be the token rate of the leaky bucket in node i, ω_i be the weight assigned to node i, U_T be the target utilization, C be the link speed and C_H be the mean rate of high priority traffic (Class A and B) on the outgoing link of node N, then we will have

$$\rho_i = \frac{\omega_i}{\sum_{j=1}^{N} \omega_j}(U_T C - C_H). \quad (1)$$

Using this scheme the 1TB-RPR-fairness algorithm distributes any spare capacity to all the nodes in the congestion span in a weighted fashion.

III. A BOUND FOR ACCESS DELAYS UNDER STEADY STATE

Based on the scheduling algorithm described in the last section, we have the following observations:

1) The ring access delays for high priority add-in traffic (Class A and Class C) are caused by bursts of pass-through traffic. The larger the bursts, the longer the access delays. On the other hand, the low priority add-in traffic can be blocked by either bursts of pass-through traffic or empty leaky bucket. The access delays caused by empty leaky bucket depend on the token refilling rate of the leaky bucket. When ring speeds are high, these rates are typically set to high values decided by the fairness share of the total capacity for low priority traf-

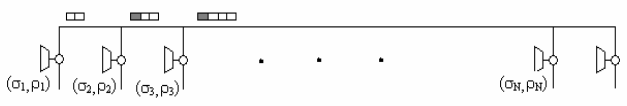

Fig. 3. The concatenation of bursts in a congestion span.

fic as shown by equation (1). For example, if the token rate is set to 300Mbps, the maximum access delay caused by empty token bucket is 40 μs for a maximum packet size of 1500 bytes. This is very different from the legacy quota-based ring schemes where the quota renewal times depend on the round trip times of a ring or a congestion span. Because, the delays caused by empty leaky bucket are small and easy to calculate for high speed rings, we are going to focus on the access delays caused by bursts on the pass-through traffic which are similar to the access delays seen by high priority add-in traffic.

2) Each node in a congestion span can generate traffic bursts contributing to the access delays seen by down stream nodes. Because the peak rates of the high priority add-in traffic of upstream nodes are shaped strictly according to CIR's, the maximum high priority burst a node can generate is one packet. Different from high priority traffic, low priority add-in traffic streams of upstream nodes are shaped by token buckets which allow much larger bursts decided by their bucket sizes. So in this paper, we will neglect the bursts caused by high priority traffic (i.e. $C_H = 0$).

3) As shown in Fig.3, the bursts generated by upstream nodes can sometimes concatenate together to form a longer burst when they reach downstream nodes. Clearly the longest burst seen by a downstream node can be decided by the possible aggregation of the longest bursts generated by all upstream nodes in a congestion span. Because propagation delays are constant, they do not contribute extra bursts. The transit buffers may contribute extra bursts, but have very little impact because their sizes are too small. So in the following, we will neglect propagation delays and transit buffers to simplify our analysis.

We are only interested in developing a steady state bound in this paper. Steady state means that the fairness algorithm has been triggered during a sustained congestion period, each node in a congestion span has applied a target rate to its leaky bucket based on the advertised rate. For more detailed studies on transit behavior, readers can refer to [10]

Lemma 1: For a congestion span with *N+1* nodes where each node *i* is regulated by a leaky bucket with parameters (σ_i, ρ_i), then the access delays for high priority traffic at the head node (Node *N+1*) is bounded by

$$B_N = \frac{\sum_{i=1}^{N} \sigma_i}{C - \sum_{i=1}^{N} \rho_i} \tag{2}$$

Proof:

The constraints imposed by the leaky bucket in node *i* are as follows: If $A_i(\tau, t)$ is the amount of flow that leaves the leaky bucket and enters the ring in time interval (τ , t], then

$$A_i(\tau, t) \le \sigma_i + \rho_i (t - \tau), \forall t \ge \tau \ge 0. \tag{3}$$

Define a burst to be an interval *B* such that for any τ, $t \in B$, $\tau \le t$,

$$\sum_{i=1}^{N} A_i(\tau, t) = (t - \tau)C \tag{4}$$

If $B = [t_1, t_2]$, from (3)(4), we have

$$B = t_2 - t_1 \le \frac{\sum_{i=1}^{N} \sigma_i}{C - \sum_{i=1}^{N} \rho_i} \tag{5}$$

In the above proof, we have assumed a fluid model [11]. The errors introduced by a fluid model are small and have been well studied. The above approach is similar to [11] with a major difference: In [11], it is assumed that there is an infinite buffer between the leaky buckets and the scheduler while in our case there is no buffer at all after the leaky buckets. In equation (2), $\sum_{i=1}^{N} \sigma_i$ is the maximum number of tokens that the first *N* nodes in a congestion span can accumulate, *C* is the token consumption rate while the outgoing link of Node *N* is busy, and $\sum_{i=1}^{N} \rho_i$ is the total token replenishing rate. It is easy to see now that the bound stated in equation (2) is a bound imposed by the over-all token supply.

Although Lemma 1 has shown that equation (2) is a bound, it is not necessary a tight bound unless we can find a real traffic scenario with access delays that can actually reach the bound. It has been shown in [11] that greedy sessions, sessions that use as many tokens as possible, can reach the bound as calculated by equation (2) for a GPS (Generalized Processor Sharing) multiplexing system. Because any overloading sessions are greedy sessions, it is very easy to find a scenario that can achieve the bound for GPS. Unfortunately this is not the case for our RPR scheme. This is because our congestion span does not have any buffer between its leaky buckets and its scheduler. Therefore the down stream nodes will lose tokens when they are blocked by the traffic from their upstream nodes if their buckets are full. From equation (2) we can see that it will make the burst shorter if any token is lost. Therefore concurrent greedy sessions will not be the worst case in an RPR system. In the following we will construct a special deterministic traffic scenario for an RPR system to achieve the bound.

Lemma 2: The RPR bound in equation (2) is tight for ring access delays of a 1TB-RPR system.

Proof:

We use a constructive approach to prove that the bound is tight. We will show that we can always find a traffic scenario in which the maximum ring access delay equals to the RPR bound for a set of arbitrary parameters that satisfy equation (1).

Our deterministic traffic scenario is shown in Fig.4. Also shown in Fig. 4 are the dynamics of their corresponding leaky buckets. We assume that all buckets are full at t_0. The source

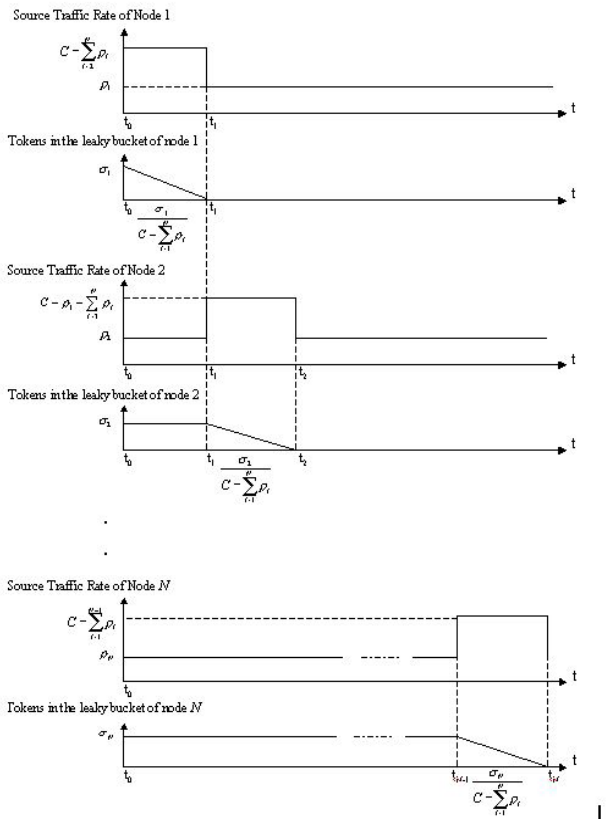

Fig. 4. Source traffic rates and tokens of the N nodes in the congestion span.

traffic rate of node 1 is set to $C-\sum_{i=2}^{N}\rho_i$, while the source traffic rates of node 2 to node N are set to ρ_2, $\rho_3 \cdots \rho_N$. Therefore the total amount of traffic the N nodes add on to the ring is

$$C-\sum_{i=2}^{N}\rho_i+\rho_2+\rho_3+\cdots+\rho_N=C.$$

That is to say that the ring is busy from t_0. It should be noted that, at this moment, the number of tokens in the bucket of node 1 is decreasing while all other buckets maintain the same. We set the source traffic rate of node 1 at $C-\sum_{i=2}^{N}\rho_i$ until the leaky bucket in node 1 runs out of tokens at t_1. It is easy to see that the busy period $[t_0, t_1]$ will be

$$B_1=\frac{\sigma_1}{C-\sum_{i=2}^{N}\rho_i-\rho_1}=\frac{\sigma_1}{C-\sum_{i=1}^{N}\rho_i}$$

As shown in Fig. 6, when node 1 runs out of tokens at t_1, we set the source traffic rate of node 1 to ρ_1. At the same time, we set the source traffic rate of node 2 to $C-\rho_1-\sum_{i=3}^{N}\rho_i$, and keep this value until the leaky bucket in node 2 runs out of tokens at t_2 while all other nodes stay at their original rates. This busy period $[t_1, t_2]$ will be

$$B_2=\frac{\sigma_2}{C-\rho_1-\sum_{i=3}^{N}\rho_i-\rho_2}=\frac{\sigma_2}{C-\sum_{i=1}^{N}\rho_i}.$$

When there are no tokens left in the leaky bucket of node 2 at t_2, the source traffic rate of node 2 goes back to ρ_2, and the source traffic rate of node 3 goes up to $C-\sum_{j=1}^{2}\rho_j-\sum_{i=4}^{N}\rho_i$.

We can repeat this process until we finish all the nodes in a congestion span as shown in Fig. 6. Therefore the total burst length will be

$$B=B_1+B_2+\cdots+B_N=\frac{\sigma_1}{C-\sum_{i=1}^{N}\rho_i}+\frac{\sigma_2}{C-\sum_{i=1}^{N}\rho_i}+\cdots+\frac{\sigma_N}{C-\sum_{i=1}^{N}\rho_i}=\frac{\sum_{i=1}^{N}\sigma_i}{C-\sum_{i=1}^{N}\rho_i} \quad (6)$$

From (6) we can see that the maximum burst length in this specific case is exactly equal to the bound for ring access delays as defined in (2).

From Fig. 4 we can see that Node 1 is not greedy until t_1, Node 2 is not greedy until t_2 and so on and so forth. This is very different from the worst-case scenario in [2] where all the sessions are greedy from time t_0.

IV. Simulation Results

In this section, we will show some simulation results. Our simulation model is implemented in OPNET, a powerful simulation tool. All links are assumed to be running at 10Gbps. We are going to focus on the hub applications that are likely the worst case in terms of access delays because the whole ring will become a single congestion span. The propagation delay on each link is set to70 μs (~15km). Unless stated otherwise, all weights are set equal for all the nodes and target utilization is set to 95%. All the access delays are measured at the head nodes.

In the proof of Lemma 1 and 2 it was assumed that traffic followed a fluid model. In practice all traffic streams are packetized. To check the impact of packetization, we have conducted several simulations with 8 nodes as a congestion span. In Fig. 5, the pair of dotted lines is the calculated bounds and the simulated maximum access delays under the deterministic traffic scenario as identified in Lemma 2 with packet size of 12272 bits. The pair of solid lines is the results with packet size of 4224 bits. The reason that the calculated bounds are different with different packet sizes is because 1TB-RPR allows their leaky buckets to have a maximum deficit of 1 packet size. It is easy to see that there are errors

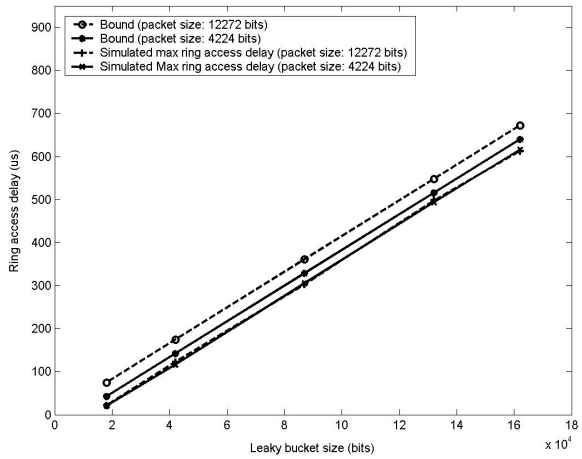

Fig. 5. Ring access delay bounds and simulated max access delays under worst case traffic scenarios with different packet sizes.

between calculated and simulated results. But the errors become smaller when the packet size becomes smaller. Therefore it can be concluded that these errors are introduced by packetization. In general they are small and negligible.

As pointed out in the last section, greedy sessions do not guarantee to achieve the bound. This is where a RPR network is different from a system such as GPS. A simulation result is shown in Fig.6 together with the calculated bounds. In this simulation, the ring is loaded with concurrent greedy sessions. As it can be easily seen, the simulated results are much smaller than their corresponding bounds. The larger the bucket sizes, the larger the differences. This is because the upstream nodes may burst longer time causing more tokens to be lost at downstream nodes.

.

V. CONCLUSIONS

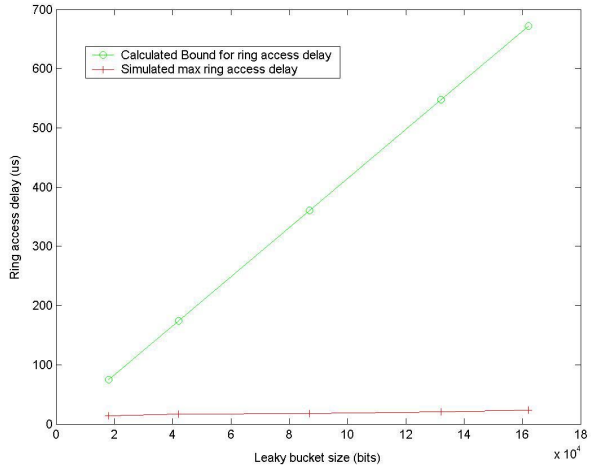

Fig. 6. Ring access delay bounds and simulated max ring access delays under concurrent greedy sessions with different leaky bucket sizes.

Different from all earlier ring technologies such as MetaRing, the 1TB-RPR scheme uses a rate-based fairness algorithm rather than quota-based approach. This allows it to significantly reduce the access delays under steady state. This greatly improves the performance of the ring networks during a sustained congestion period, a scenario very likely to happen for Internet traffic due to its strong self-similarity. Furthermore, the access delays under steady state do not depend on the ring sizes and therefore allow RPR to scale for MAN/WAN applications.

In this paper, we have developed a bound for access delays under steady state. The bound is much smaller than the bounds found in [6], which are at the order of round trip times. Simulation results have shown that, the actual access delays are typically much smaller than the bound we have developed although we have proved the bound is tight with a deterministic traffic scenario. For the behavior during transit period, readers are referred to [10] for further details.

REFERENCES

[1] Cole, N., Hawkins, J., Green, M., Sharma, R. and Vasani, K., "Resilient Packet Rings for Metro Networks," available at: http://www.rpralliance.org/, Aug. 2001.

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

[3] Cidon, I. and Ofek, Y., "MetaRing – A Full-Duplex ring with Fairness and Spatial Reuse," *IEEE/ACM Transaction on Communication*, vol. 41, pp. 110 – 120, Jan. 1993.

[4] Breuer, S. and Meuser T., "Enhanced Throughput in Slotted Rings Employing Spatial Slot Reuse," in *Proceedings of IEEE INFOCOM'94*, Toronto, Jun. 1994.

[5] Chen, J., Cidon, I., and Ofek Y., "A Local Fairness Algorithm for Gigabit LAN's/MAN's with Spatial Reuse," *IEEE JSAC*, vol. 11, no. 8, Oct. 1993.

[6] Cidon, I., Georgiadis, L., Guerin, R, and Shavitt, Y.,"Improved Fairness Algorithms for Rings with Spatial Reuse", *IEEE/ACM Transactions on Networking,* vol. 5, no. 2, pp.190-204, Apr. 1997.

[7] Leland, W. E., Taqqu, M., Willinger, W., and Wilson, D., "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," IEEE/ACM Trans. Networking, vol. 2, no.1, Feb. 1994.

[8] Huang, C., Devetsikiotis, M., Lambadaris, I., and Kaye, A. R., " Self-Similar Modeling of Variable Bit Rate Compressed Video: A Unified Approach," in *Proceedings of IEEE/ACM SIGCOMM'95*, Cambridge, Aug. 1995.

[9] Norros, I., " Queueing Behavior Under Fractional Brownian Traffic," in *Self-Similar Network Traffic and Performance Evaluation*, edited by K. Park and W. Willinger, John Wiley & Sons, 2000.

[10] Francisco, M., Yuan, F., Huang, C., and Peng, H., " A Comparison of Two Buffer Insertion Ring Architectures with Fairness algorithms", accepted to ICC'93, Anchorage, May, 2003.

[11] Parekh, A. K. and Gallager, R. G., "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Transaction on Networking*, vol. 1, no. 3, pp. 344 – 357, June 1993.