A Deterministic Bound for the Access Delay of Resilient Packet Rings

Changcheng Huang, *Member, IEEE,* Harry Peng, and Fengjie Yuan

***Abstract*— Resilient Packet Ring (RPR) is a new technology being standardized by the IEEE 802.17 working group. This paper presents a ring access delay bound under steady state. The bound is then proved analytically. Furthermore we show that the bound is tight by constructing a worst-case traffic scenario. It is shown that straight overloading scenarios are not the worst case.**

***Index Terms*— Multiple access protocols, performance analysis, high-speed networks.**

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. Token ring [2], for example, is one of the earliest ring protocols that have been standardized. MetaRing [3], a well-known scheme that supports spatial reuse, deploys a quota-based fairness scheme with maximum access delays within the order of ring round trip times [4].

The RPR scheme discussed in this paper tries to minimize the access delay 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. The transit behavior of a RPR network is similar to MetaRing. 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 a 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 [5]. The improvement over access delays under steady state allows RPR to scale to much larger ring sizes (e.g. 2000km) and much higher ring speeds (e.g. 10 Gb/s or above) so that it can be applied to MAN/WAN applications.

In this paper, a bound for the access delays under steady state is developed. It is shown that the bound is much smaller than the bounds in [4].

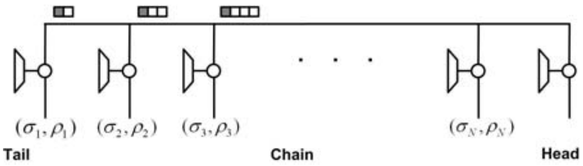

Fig. 1. An example of a congestion span and burst concatenation.

II. The CM-RPR Scheme

A typical RPR node with single transit buffer deploys a priority-based scheduler. The scheduler under the conservative mode (CM) defined in the draft RPR standard chooses data packets from five queues according to the following order: packets from transit buffer, packets from Class A transmit buffer, packets from Class B transmit buffer, packets from in-span Class C transmit buffer, packets from out-of-span Class C transmit buffer. Because the pass-through traffic has absolute priority over the add-in traffic, only a very small transit buffer (one or two 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 low priority traffic (Class C) at each node of a congestion span during periods of congestion. Fig. 1 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. Based on the utilization of its downstream link, the head node calculates a fair rate for Class C traffic and then advertises it to the upstream nodes in the span when congestion happens. Having received the normalized advertised rate from the downstream node, each node calculates its target rate and then applies 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 at node i, w_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

$$\rho_i = \frac{w_i}{\sum_{j=1}^{N} w_j}(U_T C - C_H) \qquad (1)$$

Manuscript received May 15, 2004. The associate editor coordinating the review of this letter and approving it for publication was Prof. Michael Devetsikiotis.

Changcheng Huang is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada (e-mail: huang@sce.carleton.ca).

Harry Peng is with Nortel Networks, Ottawa, Canada (e-mail: hpeng@nortelnetworks.com).

Fengjie Yuan was with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada.

Digital Object Identifier 10.1109/LCOMM.2005.01025.

Using this scheme, the CM-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 add-in traffic flows are caused by either bursts of pass-through traffic or their own shapers (i.e. empty leaky buckets). The access delays caused by their own shapers are small during a congestion period when ring speeds are high. So we are going to focus on the access delays caused by the bursts of the pass-through 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 (Committed Information Rate), the maximum high priority burst a node can generate is one packet. On the other hand, 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. $C_H = 0$). We are interested in the case that low priority pass-through traffic bursts block high priority add-in traffic flows at the down stream nodes;
3) As shown in Fig. 1, 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. 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 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 and each node in a congestion span has applied a target rate to its leaky bucket for Class C traffic based on the advertised rate.

Theorem 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 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 $(\tau, t]$, then [6]

$$A_i(\tau, t) \leq \sigma_i + \rho_i(t - \tau), \forall t \geq \tau \geq 0 \tag{3}$$

Define a burst seen at Node $N+1$ to be an interval B such that $\forall \tau, t \in B, \tau \leq t$,

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

where T_i is the propagation delay from node i to node $N+1$. If $B = [t_1, t_2]$, from (3)(4), we have

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

$\diamondsuit$

It is interesting to note that the above bound does not depend on the propagation delay. The above approach is similar to [6] with a major difference: In [6], 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. Therefore the bound in (2) is not tight in [6] but is tight for our system as it is shown in the following paragraphs.

Although Theorem 1 has shown that (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 [6] that greedy sessions, sessions that use as many tokens as possible, are likely to be the worst-case scenario 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 the conservative mode RPR scheme. This is because our congestion span does not have any buffer between its leaky buckets and its scheduler. Therefore the downstream nodes will lose tokens when they are blocked by the traffic from their upstream nodes if their buckets are full. From (2) we can see that it will make the burst shorter if any token is lost. In the following we will construct a special deterministic traffic scenario for an RPR system to achieve the bound.

Theorem 2. The RPR bound in (2) is tight for the ring access delay of a CM-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 (1).

Our deterministic traffic scenario is shown in Fig. 2. Also shown in Fig. 2 are the dynamics of their corresponding leaky buckets. We assume that bucket i is full at $t_0 - T_1, i = 1, \cdots, N$. The ingress traffic rate at node 1 transmit buffer is set to $C - \sum_{i=2}^{N} \rho_i$ at $t_0 - T_1$, while the ingress traffic rates at node 2 to node N are set to $\rho_2, \rho_3, \cdots, \rho_N$. Therefore the pass-through traffic rate at node $N+1$ at time t_0 is

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

Therefore the ring is busy at node $N+1$ 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 stay the same. We set the ingress traffic rate of node 1 to $C - \sum_{i=2}^{N} \rho_i$ until the leaky bucket in node 1 runs out of tokens at $t_1 - T_1$.

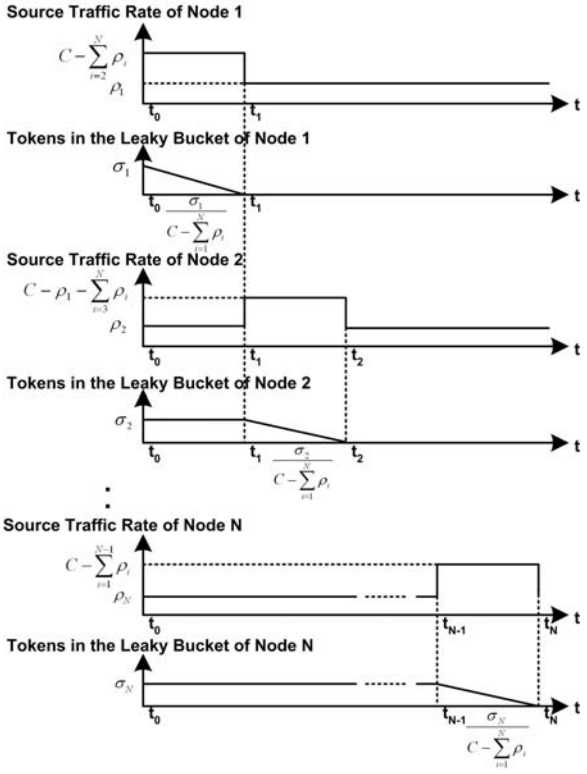
Source Traffic Rate of Node 1
$C-\sum_{i=2}^{N}\rho_i$
ρ_1
t_0
t_1
t
Tokens in the Leaky Bucket of Node 1
σ_1
t_0
$\frac{\sigma_1}{C-\sum_{i=1}^{N}\rho_i}$
t_1
t
Source Traffic Rate of Node 2
$C-\rho_1-\sum_{i=3}^{N}\rho_i$
ρ_2
t_0
t_1
t_2
t
Tokens in the Leaky Bucket of Node 2
σ_2
t_0
t_1
$\frac{\sigma_2}{C-\sum_{i=1}^{N}\rho_i}$
t_2
t
Source Traffic Rate of Node N
$C-\sum_{i=1}^{N-1}\rho_i$
ρ_N
t_0
t_{N-1}
t_N
t
Tokens in the Leaky Bucket of Node N
σ_N
t_0
t_{N-1}
$\frac{\sigma_N}{C-\sum_{i=1}^{N}\rho_i}$
t_N
t

Fig. 2. Source traffic rates and tokens of the N nodes in a congestion span.

It is easy to see that the length of 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. 2, when node 1 runs out of tokens at t_1, we set the ingress traffic rate of node 1 to ρ_1 . At the time $t_1 - T_2$, we set the ingress 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 $t_2 - T_2$ while all other nodes stay at their original rates. The length of 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 $t_2 - T_2$, the ingress traffic rate of node 2 goes back to ρ_2 , and the ingress traffic rate of node 3 goes up to $C - \sum_{i=1}^{2} \rho_i - \sum_{i=4}^{N} \rho_i$. We can repeat this process until we finish all the nodes in a congestion span as shown in Fig. 2. Therefore the total burst length will be

$$B = B_1 + B_2 + \cdots + B_N = \frac{\sum_{i=1}^{N} \sigma_i}{C - \sum_{i=1}^{N} \rho_i} \tag{6}$$

$\diamondsuit$

From (6) we can see that the maximum burst length in this specific case is exactly equal to the bound for the ring access delay as defined in (2). From Fig. 2 we can see that Node 2 is not greedy until $t_1 - T_2$, Node 3 is not greedy until $t_2 - T_3$ and so on and so forth. This is very different from the worst-case scenario in [6] where all the sessions are greedy from time t_0.

IV. Conclusions

Different from all earlier ring technologies such as MetaRing, the CM-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 [4], which are at the order of ring round trip times.

References

[1] N. Cole *et al.*, "Resilient Packet Rings for Metro Networks," available at: http://www.rpralliance.org/, Aug. 2001.

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

[3] 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.

[4] I. Cidon *et al.*, "Improved fairness algorithms for rings with spatial reuse," *IEEE/ACM Trans. Networking*, vol. 5, pp. 190-204, Apr. 1997.

[5] W. E. Leland *et al.*, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1-15, Feb. 1994.

[6] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344-357, June 1993.