A New Access Control Scheme for Metropolitan Packet Ring Networks

Helen Tang+, Ioannis Lambadaris+, Hamid Mehrvar*, Jerome Talim+ and Ashraf Matrawy+
+ Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada
*Nortel Networks, Ottawa, Ontario, Canada
Emails: +{yhtang, ioannis, jtalim, amatrawy}@sce.carleton.ca, *hamidm@nortelnetworks.com

Abstract- *This paper focuses on fair medium sharing in metropolitan packet ring networks*[1]*. We propose a new fair access control scheme called Virtual Source Queuing (VSQ) for these networks. A simple feedback control scheme for VSQ to avoid packet loss on the ring is also proposed. We compare the implementation complexity and performance of VSQ with those of the IEEE 802.17 Resilient Packet Ring (RPR) standard. The simulation and analytical results show that our scheme can achieve fairness guarantee with less convergence time. It also has less access delay and requires ring buffer compared to RPR.*

Keywords**:** Resilient Packet Ring (RPR), Virtual Source Queuing (VSQ), One-Hop-Backpressure control, Medium Access Control (MAC).

I. INTRODUCTION

An important trend in networking is the migration of packet-based technologies from Local Area Networks (LAN) to Metropolitan Area Networks (MAN). The rapidly increasing volume of data traffic in metro networks is challenging the capacity limits of the existing transport infrastructures that are based on circuit-oriented technologies such as SONET and ATM. Resilient Packet Ring (RPR) is a Metropolitan Area Network (MAN) technology supporting data transfer among stations interconnected in a dual-ring configuration [1]. Unlike a token ring where *source-removal* is used and only one packet can be transmitted at a time, RPR uses *destination-removal* which allows concurrent transmission over different segments of a ring. As a result, the total ring throughput of an RPR-ring can be significantly higher than the capacity of a single link. However, since the ring bandwidth becomes a shared medium, a key challenge for RPR is to design a Medium Access Control (MAC) scheme that ensures all nodes have fair access to the ring [2].

IEEE 802.17 RPR standard [1] specifies the RPR MAC that is based on the Buffer Insertion Ring (BIR) [3] technology. The basic operation of BIR is as follows. There is an insertion buffer at every node interface to solve the conflict between the data already flowing on the ring and data ready to be transmitted by a node. *"Strict priority"* is given to the ring traffic. Downstream nodes may experience so-called *starvation* problem if upstream nodes keep sending traffic and prevent downstream nodes from accessing the ring.

In an attempt to solve the starvation problem associated with BIR, RPR standard defines a fairness algorithm as part of its MAC scheme. The fairness algorithm is based on fair rate estimation and feedback control and works as follows. Once starvation is detected at a node, a fair rate is estimated by this node and sent to all the upstream nodes. The upstream nodes will then throttle their traffic to the received fair rate. After a convergence period, the starvation will clear and the fair rate increases until next starvation happens.

The RPR fairness algorithm, however, has two problems. The first is bandwidth oscillation due to the inaccuracy in fair rate estimation. The second is long access delay for the node traffic as a natural result of feedback rate control scheme.

In order to solve the oscillation problem of RPR fairness algorithm, another fairness algorithm called Distributed Virtual-time Scheduling in Rings (DVSR) has been recently proposed [6]. DVSR provides a different way of estimating the fair rate by approximating fair queuing with per-ingress counters of packet arrivals. The accuracy of the fair rate estimation is improved with DVSR, therefore, bandwidth oscillation is reduced compared to RPR. However, DVSR did not address the access delay issue of RPR since it still relies on rate feedback control to achieve fairness.

In this paper, we propose a new MAC scheme called Virtual Source Queuing (VSQ). Unlike RPR and DVSR schemes that use fair rate estimation and feedback for fairness, VSQ uses fair scheduling to provide guaranteed fair medium access. As a result, VSQ eliminates the oscillation and access delay problems related to RPR. The implementation complexity of VSQ is also studied and compared with that of RPR MAC.

The rest of this paper is organized as follows. In Section II, we provide an overview on the RPR MAC scheme. The proposed VSQ MAC scheme is introduced in Section III. Performance comparison in terms of fairness guarantee and access delay between VSQ and RPR is presented in Section IV. Section V addresses the implementation complexity of VSQ in comparison with RPR. Finally, the concluding remarks are given in Section VI.

II. OVERVIEW OF RPR MAC SCHEME

In this section, we describe the basic operations of RPR MAC. Readers are referred to [1] for full details. The notations used in this paper are listed in Table I.

A. RPR Node Architecture

Fig. 1 shows a simplified RPR node architecture. Only one direction of the ring is shown here, but the reverse direction is similar.

All ingress traffic entering the ring passes a rate controller module which includes virtual destination queuing (VDQ) to

---

[1] This work is supported in part by Communications and Information Technology Ontario (CITO) and Mathematics of Information Technology and Complex System (MITACS).

TABLE I. NOTATIONS

N	Number of ring nodes. RPR limits N to 256.
sizeMtu	Size of Max Transfer Unit, default 9216 Bytes.
linkRate (R_L)	Line rate of the ring
fairRate	Estimated fair rate by the starved node.
reservedRate (R_{re})	Rate provisioned for HP traffic
unreservedRate(R_{un})	Rate not provisioned for HP traffic
agingInterval	The interval at which the *fairRate* is calculated. 100µsec for 10Gbps link.
advertisingInterval	Interval the *fairRate* is advertised. Default 10µsec for 10Gbps link.
stqLowThrehold	A level of LP TB occupancy at or above which node is considered to be starved.
stqHighThreshold:	A level of LP TB occupancy at or above which LP node traffic is no longer admitted. Range: [*3 * sizeMtu, stqFullThreshold - sizeMtu*].
forwardRate	The rate of LP ring traffic at a node.
addRate	The rate of LP node traffic at a node.
(σ_i, ρ_i)	Parameters of the token bucket at node *i*. σ_i : token size ($\sigma_i \geq$*sizeMtu*); ρ_i : token rate.

avoid head of line blocking. RPR nodes have traffic monitor module to measure both node and transit traffic rate. These measurements are used by the fair rate estimation/feedback module to compute a rate for each upstream node and feedback the rate through a control message. Once an upstream node receives this control message, it adjusts its rate controller to the computed rate. The priority scheduler serves traffic from Ingress Buffer and Transit Buffer. The threshold module prevents the transit buffer overflow.

The RPR defines multiple classes of traffic in *Ingress Buffer (IB)*,. Here we simplify traffic classes into 2 priorities: high priority (HP) and low priority (LP). Since the HP class has stringent packet loss and delay requirements, the required bandwidth to carry this traffic is provisioned with *reservedRate*. On the other hand, the LP class traffic has loose delay and loss requirements, allowing statistical multiplexing to be used to increase the utilization of ring bandwidth.

The RPR standard draft defines two implementations for *Transit Buffers (TB)*: 1-TB and 2-TB. The 1-TB implementation uses one transit buffer for both HP and LP traffic while the 2-TB implementation uses separate buffers for HP and LP traffic. The scheduler serves the ingress buffers and the transit buffers based on *strict priority*. In 1-TB design, the priority order is: transit buffer, HP ingress buffer, and LP ingress buffer. For 2-TB implementation, the priority order is: HP transit buffer, HP ingress buffer, LP transit buffer and LP ingress buffer. The advantage of the 1-TB design is its hardware simplicity, however, it is possible that LP ring traffic blocks HP node traffic from accessing the ring. This is because once the LP traffic from upstream nodes gets on the ring, it becomes ring traffic and assumes higher priority than the HP traffic of a downstream node. On the other hand, the 2-TB design offers relatively lower access delay for HP node traffic that has higher priority than LP ring traffic. The disadvantage, however, is that the LP transit buffer must be large enough to store the LP ring traffic while yielding the ring access to HP node traffic. In both 1-TB and 2-TB designs, LP ring traffic has priority over LP node traffic. Therefore, a heavily loaded upstream node can hog the ring bandwidth for a very long time and cause starvation for its downstream nodes.

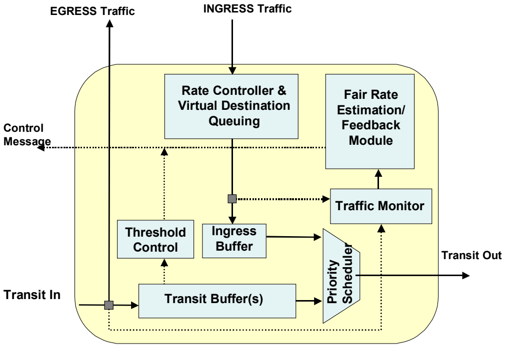


Figure 1. RPR Node Architecture.

B. *RPR Fairness Algorithm*

To prevent the *starvation* problem for LP ingress traffic, the RPR fairness algorithm uses explicit rate feedback to control the amount of LP traffic that each node can put on the ring. Since HP traffic is provisioned, fairness algorithm only applies to LP traffic. In this paper, the traffic we study is LP traffic unless otherwise specified.

RPR fairness algorithm assumes two operation modes: Aggressive Mode (AM) and Conservative Mode (CM). The operation of RPR fairness algorithm is illustrated in Fig. 2 where a part of a ring (three connected nodes) is shown. At each *agingInterval,* every node checks its starvation status based on the conditions specific to the mode AM or CM (AM mode is shown in Fig. 2.). Once starvation is detected, the starved node estimates a *fairRate* and sends a feedback control message including the *fairRate* to the upstream nodes in the opposite direction. Upstream nodes apply the *fairRate* as the token rate of its token bucket so that its *addRate* is throttled to the *fairRate*.

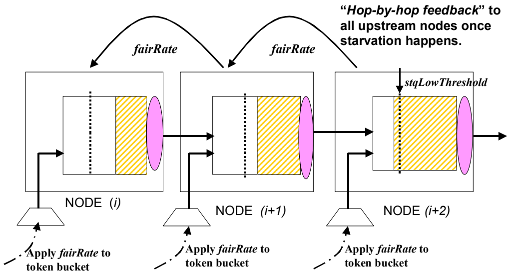
In AM mode, the starvation is detected when the *stqLowThreshold* of LP TB is exceeded. The initial *fairRate* is set to the *addRate* of the starved node. The underline logic is that all nodes should have the same *addRate*. If the throughput of a starved node drops, all other nodes should follow. Once the starvation is clear, the *fairRate* ramps up and gradually get closer to the optimal fair rate.

In CM mode, the starvation can be triggered in two ways:

a) when the outgoing link usage *(forwardRate + addRate)* of a node exceeds a threshold. The threshold is 80% of the link rate by default and the remaining bandwidth is reserved for HP traffic;

b) when the Head of line (HOL) timer expires, which means a packet at the head of the LP Ingress Buffer has waited for too long.

The *fairRate* is set to be the *unreservedRate* divided by the number of *active* nodes that have had at least one packet passing the starved node in the last *agingInterval.* After the starvation clears, the *fairRate* gradually ramps up. In summary, RPR fairness control process requires 4 steps as listed in Fig. 2: starvation detection, fair rate estimation, hop-by-hop broadcast and fair rate application.



Procedure of the RPR feedback control to ensure fair access:
Step 1: **Detect starvation (AM mode is shown).**
Step 2: **Estimate fair rate:** ***fairRate.***
Step 3: **Broadcast the** ***fairRate*** **to all upstream nodes "hop-by-hop".**
Step 4: **Upstream nodes apply the** ***fairRate*** **to their token buckets.**

Figure 2. Illustration of RPR feedback fairness control scheme

III. VIRTUAL SOURCE QUEUING SCHEME

A. Limitations of RPR Fairness Algorithm

The RPR fairness algorithm can reduce the starvation problem. However, it has two limitations. The first is bandwidth oscillation due to the inaccuracy in fair rate estimation. The results in [4],[5] have shown that with "unbalanced node traffic" scenario, RPR fairness algorithm is prone to severe and permanent oscillations. Such oscillations may result in significant throughput loss [6]. The second is long access delay for the node traffic since there is a delay from the time the fair rate is calculated to the time the fair rate is taken into effect at all nodes [4]. This is a natural result of feedback control scheme.

We believe that the priority scheduler used in RPR MAC is the root cause of the starvation problem. The existing fairness algorithms have not tackled this fundamental problem. Instead, they attempt to achieve fair medium access by regulating the ring access through fair rate estimation and feedback control.

B. Virtual Source Queuing (VSQ)

An ideal MAC scheme for packet rings should assure fairness as well as maximize the throughput on each segment of the ring. The main idea of the proposed approach is to provide a fair share of the bandwidth to all the competing nodes. We use the weighted so*urce-based fairness* definition stated in the RPR standard [1]: on any given segment on the ring, the available bandwidth is allocated to each node using the segment in proportion to its relative weight. This definition is also used in DVSR scheme under the name of "ingress node aggregated" fairness [6]. In what follows, we formulate the fair medium access problem and propose our solution.

<u>Problem Statement:</u> Let us consider a unidirectional ring with *N* nodes. For each node *i* $(1 \le i \le N)$, we define L_i as the bandwidth of the outgoing link between node *i* and node $(i+1) \bmod N$. All the nodes that send traffic through L_i are called the ***sources*** of L_i. The problem of fair medium access translates to how to fairly allocate L_i among all the competing sources.

To solve the above problem, we provide a virtual queue for traffic from each possible source of L_i. The total number of virtual queues at each node is $(N-1)$. Q_{ij} $(1 \le j \le N)$ represents the virtual queue at node *i* with source *j*. A weight η_{ij} is assigned to Q_{ij} based on the relative importance of that node. The option of setting different weights to different nodes could result in prioritizing the nodes and achieving differentiated QoS. Any scheduling algorithm that provides fair sharing can be used to serve these virtual source queues. The choice of scheduling algorithm is implementation-specific. In this paper, we choose Weighted Round Robin (WRR) [7] due to its simplicity.

Since Metro packet ring networks have limited number of nodes, e.g., RPR standard limits the number of nodes on a ring to 256[1]. Thus, we argue that scalability is not an issue for VSQ in metro packet ring networks.

C. One-Hop-Backpressure control for VSQ

In order to have a lossless transit path for the packet rings, we design a simple O*ne-Hop-Backpressure* control scheme.

We use two sets of thresholds for each virtual queue: *VSQThreshold* and *VSQLowThreshold (VSQThreshold > VSQLowThreshold)*. When a queue *(i)* at node *(n)* exceeds the upper threshold, *VSQThreshold*, a feedback control message *OFF(i)* is sent from node *(n)* to the upstream neighbor *(n-1)*. *OFF(i)* indicates that the traffic from queue *(i)* at node *(n-1)* is excessive and may lead possible buffer overflow at node *(n)*. Therefore, node *(n-1)* should decrease its transmission rate by pausing servicing packets from queue *(i)* until it receives an *ON(i)* signal from node *(n)*. The *ON(i)* signal is triggered when the queue *(i)* descends below the lower threshold, *VSQLowThreshold.* While the scheduler pauses the service rate of queue *(i)*, the utilization of link *(n-1)* does not decrease because the scheduler is work-conserving so that other queues can be served. As a result, the maximum buffer occupancy at node *(n)* is expected to be nearly bounded by *VSQThreshold*. More details on the buffer size will be given in Section IV-C.

This operation of VSQ feedback control is very simple and effective because the One-Hop-Backpressure message works very fast. This feedback control only needs to check the queue length upon reception and transmission of packets.

On the other hand, RPR feedback control is much more complex because it is used not only to prevent packet loss but also to enforce fair access. There are 4 steps involved in the RPR feedback control operations as shown in Fig. 2. The feedback control messages that contain the *fairRate* information are sent *periodically* "hop-by-hop" to *all* upstream nodes. The simplicity in VSQ feedback control also results in less transit buffer requirement compared to RPR as shown later in Section IV-C.

D. VSQ Node Architecture

Fig. 3 illustrates the architecture of VSQ MAC. Compare to the RPR MAC architecture in Fig. 1, the main difference is that VSQ eliminates the fair rate estimation/feedback module, traffic monitor, rate controller and VDQs. The reason is that VSQ uses the fair scheduler to serve all the queues fairly so that every node can have guaranteed fair share of the ring bandwidth without fair rate estimation and feedback as needed in RPR.

Another benefit of VSQ is that the LP traffic of different sources is isolated from each other. Therefore, a greedy source that sends excessive traffic may suffer delay or loss while a well-behaved source has immediate access to its guaranteed fair share. This brings a strong motivation for misbehaved

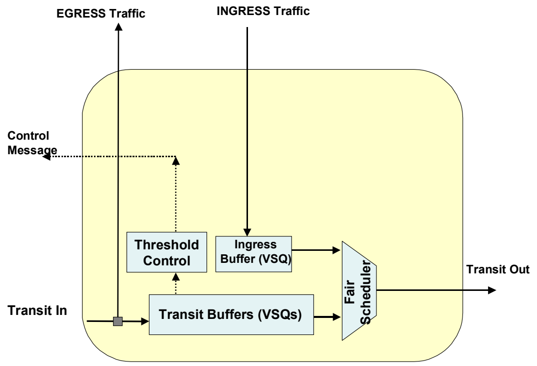


Figure 3. VSQ node architecture

sources to control their traffic generation.

In summary, VSQ scheme has the following advantages:

- Access guarantee: Every node has immediate access to its fair share of bandwidth without waiting for the fair rate feedback control to takes into effect.
- Fairness guarantee: Fair medium access is ensured without bandwidth oscillation because the scheduler fairly serves all the virtual source queues.
- Maximizing link utilization: Link is always fully utilized because there is no bandwidth oscillation.
- Simplicity: There is no need for fair rate estimation and feedback.
- Isolation: Well-behaved sources are isolated from greedy ones.

IV. PERFORMANCE EVALUATION

In this section, we first study the fairness property of VSQ and compare the convergence time with RPR and DVSR. We then derive the access delay bound, average access delay and end-to-end delay for VSQ and compare them with those of RPR.

A. Simulation Results on Fairness Guarantee

We use OPNET simulations to show the fairness property of VSQ. As an illustrative example, we use the "parking lot scenario" described in [6] and depicted Fig. 4. The link rate is set to be 622*Mbps*. Sources 1, 2, 3, and 4 start sending *248.8Mbps* to node 5 at times 0 *sec*, 0.1 *sec*, *0.2 sec* and 0.3 *sec,* respectively. In this example, the output link of node 4 is shared by sources 1, 2, 3 and 4. Equal weights have been assigned to all the four sources.

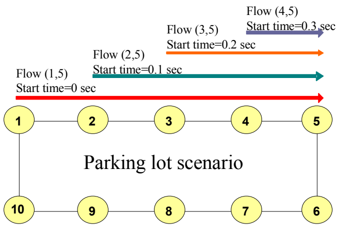


Figure 4. Parking lot simulation scenario

TABLE II. FAIR RATE CONVERGENCE TIME COMPARISON

	RPR	DVSR	VSQ
Convergence time	18-50msec	2 msec	0.1msec
Dependence on RTT	Increases with RTT	Increases with RTT	No impact from RTT

Our simulation results show that VSQ can ensure fair rate without bandwidth oscillation. The fair rate convergence time of VSQ is much smaller than those of RPR and DVSR. In Table II, we compare the fair rate convergence time of VSQ with the results in [6]. The convergence times are 50msec for RPR-AM, 18 msec for RPR-CM, two ring round trip times (RTT), 2 msec, for DVSR and *0.1msec* for VSQ. The convergence times of both RPR and DVSR are very sensitive to the RTT. For example, if the RTT delays increased by a factor of 10 to be *10 msec*, DVSR takes approximately *22 msec* to converge [6]. On the other hand, the RTT has no impact on VSQ convergence time because VSQ does not need fair rate estimation and feedback control for fairness.

B. Comparison of Access Delay Upper Bound

In this subsection, we will show that VSQ access delay bound is smaller than RPR delay bound. Some of the notations used here are previously listed in TABLE I.

1) Access Delay Upper bound for VSQ

The maximum waiting time for a packet at the head of a virtual source queues is for the scheduler to serve a packet from all other *(N-1)* virtual source queues. Let us assume that all the packets are of the maximum size of *sizeMtu* and the switching time between the queues is negligible, the access delay upper bound for VSQ is given by:

$$VSQ_{bound} = sizeMtu * (N-1) / unreservedRate. \tag{1}$$

2) Access Delay Upper bound for RPR

Since RPR fairness algorithm is based on fair rate estimation and feedback control, a starved node can only access the ring after the rate feedback control takes into effect. As a result, the access delay bound equals to the *responseTime* which is the summation of the following three parts given in [1]:

$$\begin{aligned} RPR_{bound} &= responseTime \\ &= fairnessAgingInterval + MaxFRTT + MaxFRED \end{aligned} \tag{2}$$

1) *fairnessAgingInterval*: the maximum amount of time for the fairness algorithm to detect the starvation. The default value is 100µs for 10Gbps.
2) *maxFRTT (Maximum Fairness Round Trip Time):* the propagation time of a fair rate message around the ring (D_r: ring length *(k*m), link delay: *5µs/km)*: *maxFRTT =5µs * D_r + advertisingInterval * (N-1).*
3) *maxFRED (Maximum Fair Rate Enforcement Delay)*: the time for the fair rate to be taken into effect at all ring nodes. It is different for 1-TB and 2-TB design.

For 1-TB design, the token bucket credit limits of all the upstream nodes have to be cleared before the fair rate can take into effect, so *maxFRED* is given in [1],[8] as:

$$MaxFRED_{(1-TB)} = \frac{\sum_{i=1}^{N-1}\sigma_i}{R_L - \sum_{i=1}^{N-1}\rho_i}$$

In 2-TB design, all upstream LP transit buffers have to be drained before the fair rate can take into effect. Assuming that all upstream LP transit buffers are at *stqHighThreshold* when they receive the new fair rate, thus, *maxFRED* is given in [1] as:

$$maxFRED_{(2\text{-}TB)} = stqHighThreshold * (N-1)/unreservedRate.$$

We compare the access delay bound of VSQ and RPR through the following calculations:

$$\frac{RPR_{bound(1-TB)}}{VSQ_{bound}} = \frac{fairnessAgingInterval + \max FRTT + \max FRED_{(1-TB)}}{(sizeMtu*(N-1)/unreservedRate)}$$

$$> \frac{\sum_{i=1}^{N-1}\sigma_i}{sizeMtu} \ge \frac{sizeMtu*(N-1)}{sizeMtu} = N-1,$$

$$\frac{RPR_{bound(2-TB)}}{VSQ_{bound}} = \frac{fairnessAgingInterval + \max FRTT + \max FRED_{(2-TB)}}{(sizeMtu*(N-1)/unreservedRate)}$$

$$> \frac{stqHighThreshold}{sizeMtu} \ge \frac{sizeMtu*3}{sizeMtu} = 3.$$

The results show that RPR bound is much larger than VSQ bound, which means in the worst case, node traffic experiences less access delay with VSQ than with RPR

C. Comparison of Average Packet Delay

To evaluate whether the virtual source queues and round-robin scheduler on the ring introduce any additional delay with respect to RPR, we compare the average access delay and end-to-end delay between RPR and VSQ scheme. The average end-to-end packet delay mainly has two components: a) average access delay and b) average transit delay as the waiting and transmission time at all intermediate nodes on the ring. The derivation of the delays for symmetric traffic pattern is given in the Appendix. Here we show some numerical results based on (A.1)-(A.8).

We consider a uni-directional OC-48 rate ring with N=8 nodes. Packet size is set to be 500 bytes. Weights are assumed to be equal for all nodes in VSQ scheme. We plot the access delay and end-to-end delay versus load for both RPR and VSQ schemes in Fig. 5 and Fig. 6. For the 8-node ring, the maximum load of each node is 25% (equals to system load of 100%) due to the aforementioned benefit of *destination-removal* as opposed to 12.5% with *source-removal.* As we can see from Fig 5, VSQ has less access delay than RPR for all load conditions. As for the end-to-end delay, Fig. 6 shows that when the load of each node is below 20% (which equals to system load of 80%), VSQ has less end-to-end delay. When the node load is above 20%, VSQ and RPR has similar end-to-end delay. The reason is that at high load, the queuing delay dominants the end-to-end delay, thus, the advantage of VSQ on access delay diminishes.

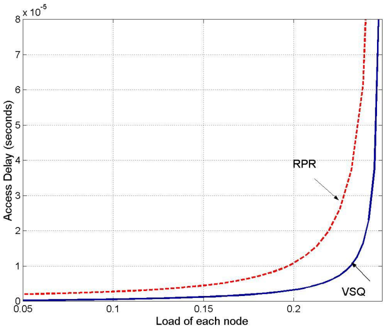


Figure 5. Average Access delay comparison between RPR and VSQ.

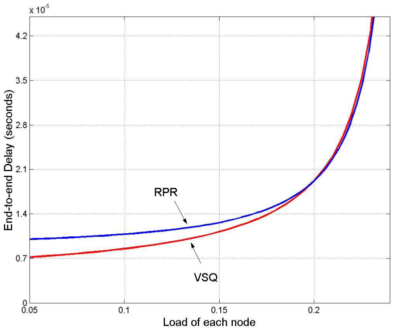


Figure 6. Average end-to-end delay comparison between RPR and VSQ.

To validate the analytical results demonstrated in the Appendix, we present our simulation results on the end-to-end delay for VSQ and compare them with the numerical results. The parameters of the simulations are the same as the numerical ones. As shown in Fig. 7, the simulation results (drawn with symbols) closely tracks the numerical results (plotted in lines).

V. COMPLEXITY COMPARISON OF VSQ AND RPR

In terms of implementation complexity, VSQ and RPR differ in the following three aspects:

a) packet address processing,

b) feedback control scheme, and

c) transit buffer requirement.

In the following sub-sections, we address these three aspects and show that VSQ can use the same addressing scheme as

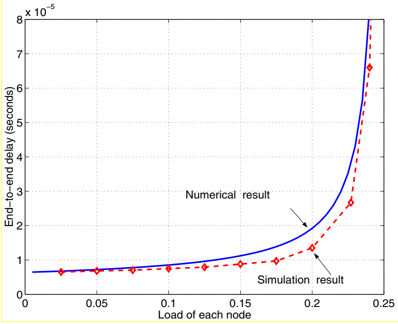


Figure 7. Comparison of numerical result with simulation result.

RPR and it requires less complex feedback control as well as less buffer requirement than RPR.

A. Packet Address Processing

RPR standard defines globally unique 48 bit IEEE 802.17 addresses [1] for the source and the destination addresses of the packets on the rings. VSQ uses the same addressing scheme. When a node receives packet from the ring, it checks both source and destination addresses. The destination address is first checked to decide whether the packet is destined to this node. The source address is checked with both VSQ and RPR but for different purposes:

- With VSQ: the source address is checked to decide which virtual source queue it should be put in.
- With RPR: the source address is checked for counting the number of active nodes to estimate the *fairRate*.

Therefore, there is no additional overhead for VSQ to check the sources address of an incoming packet and place it into the appropriate virtual source queues.

B. Feedback Control

Feedback control is used in both VSQ and RPR. However, the objective and operation are very different as was described in Section III-C.

C. Transit Buffer Requirement

For VSQ, the transit buffer size is chosen so as to hold all the LP traffic before the *One-Hop-Backpressure* signal reaches the upstream neighboring node. The time it takes for the signal reaches the upstream neighbor is *one-hop-delay*. The maximum traffic rate to a virtual source queue is *unreservedRate* (R_{un}). Therefore, the maximum transit buffer size at a node (*sizeVSQ*) for a ring with N nodes ($N-1$ virtual source queues) is given by:

$$sizeVSQ=(VSQThreshold+one\text{-}hop\text{-}delay * R_{un})*(N-1). \quad (3)$$

For RPR, the 2-TB design also requires a large transit buffer for LP traffic to reduce access delay for HP ingress traffic. The buffer size requirement has been studied in [1]. The maximum LP transit buffer size at an RPR node *(sizeRPR)* is set to be able to hold all the LP transit traffic that could be received while waiting for upstream nodes to apply the *fairRate*. The worst case is all the transit traffic are LP with *lineRate* and the node traffic are HP with rate of *reservedRate*(R_{re}) so that the transit buffer size grows with the rate of R_{re}[1].

$$sizeRPR=stqHighThreshold+ R_{re} * responseTime. \quad (4)$$

The second term is the maximum amount of HP node traffic that the node is expected to receive during the *responseTime* which has been given in (2).

Following is an illustrative example for comparing the transit buffer size requirement of VSQ and RPR.

Example: We consider a ring with $N=100$, $D_r=1000km$, every hop distance is 10km, $R_L=10Gbps$, $R_{re}=2Gbps$.

For VSQ, we set the *stqThreshold =100KB* and we have:

$$sizeVSQ= (100KB + 5\mu s/km * 10\ km*8Gbps/8) *(N-1)$$
$$= 0.15N-0.15\ (MB)=14.85\ (MB).$$

For RPR, in order to be comparable to VSQ calculation, we set *stqHighThreshold =100KB*(N-1)=9.9MB*. We have:

$$responseTime = fairnessAgingInterval+maxFRTT+maxFRED$$
$$= 100\ \mu s + 6ms +100KB*(N-1)*(N-1)/R_{un}$$

By plugging the data to (4) and applying algebra, we have:

$$sizeRPR= stqHighThreshold+ R_{re} * responseTime$$
$$=025N^2+0.05N+1.45\ (MB)= 256.45\ (MB),$$

which is more than 17 times the size of VSQ buffer requirement of 14.85 MB. The buffer size requirement versus N for this example in shown in Fig. 8. We see that *sizeRPR* grows with N^2 while *sizeVSQ* grows linearly with N.

VI. CONCLUSIONS

In this paper, we presented a new MAC scheme called VSQ for metro packet ring networks. Unlike the existing RPR MAC that relies on fair rate estimation and feedback to regulate the medium access, VSQ applies virtual queues and fair scheduler to guarantee fair medium access.

The performance results have shown that VSQ can provide fairness guarantee with no bandwidth oscillation and less convergence time compare to RPR MAC. Both the *worst-case* and the *average* access delays of VSQ are less than those of RPR. Furthermore, we propose a simple and effective one-hop-backpressure control for VSQ to avoid packet loss for limited transit buffer space. The transit buffer requirement in VSQ is shown to be much less than in RPR 2-TB design. We believe that VSQ is more cost effective than RPR MAC and could work as a new fairness mode with the RPR technology.

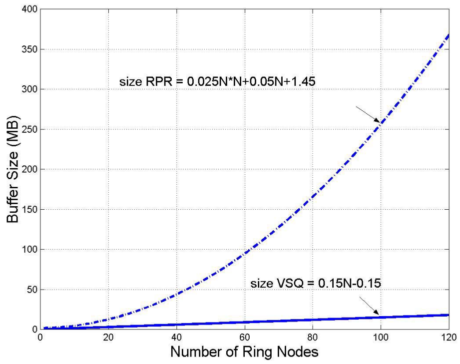


Figure 8. Transit buffer size requirement comparison

Appendix: Average Packet Delay Analysis For VSQ

For VSQ delay analysis, we consider an *N*-node unidirectional ring in which each node is modeled as a queuing system with *N* virtual source queues. Each queue is assumed to have infinite waiting room. For each queue Q_{ij}, $(1 \le i,j \le N)$ the service limit k_{ij} represents the number of packets that can be transmitted in each service cycle for Q_{ij}.. W_{ij} represents the delay at queue Q_{ij}.

To make the open network of queuing model mathematically tractable, certain assumptions are made [9].

- Symmetric traffic pattern: each node sends its traffic to all the other nodes with equal probability of $1/(N-1)$.
- Packet arrival process at every node is independent and follow Poisson process with λ packets/sec.
- All nodes have the same distribution of packet length with first moment, $\overline{x}$ and second moment, $x^{(2)}$.
- The propagation delay on the ring is neglected since it is constant and the same for both VSQ and RPR.
- "Store-and-forward" mechanism on the ring and hence, the transmission time at different nodes is additive.

Let λ_{ij} represent the arrival rate to Q_{ij} , we have:

$$\lambda_{ij} = \begin{cases} \lambda(\frac{j-i-1}{N-1}), & i<j, \\ \lambda(\frac{N-1-i+j}{N-1}), & i>j, \\ \lambda, & i=j. \end{cases} \tag{A.1}$$

Since there is no exact solution for the waiting time W_{ij}, we use the approximation on $E(W_{ij})$ proposed in [10]:

$$E[W_{i,j}] = \frac{(1-\rho_{ij})(1-\rho_i) + \frac{\rho_{ij}}{k_{ij}}(2-\rho_i)}{1-\rho_i - \frac{\lambda_{ij}s}{k_{ij}}} * \frac{D_i + \frac{s}{1-\rho_i}\sum_{l=1}^{N}\frac{\rho_{il}^2}{k_{il}}}{\sum_{l=1}^{N}[\rho_{il}(1-\rho_{il}) + \frac{\rho_{il}^2(2-\rho_i)}{k_{il}(1-\rho_i)}]}, \tag{A.2}$$

where

$$\rho_{ij} = \lambda_{ij}\beta \, , \rho_i = \sum_{j=1}^{N}\rho_{ij},$$

$$D_i = \frac{\rho_i \sum_{j=1}^{N}\lambda_{ij}\beta^{(2)}}{2(1-\rho_i)} + \frac{\rho_i s^{(2)}}{2s} + \frac{s}{2(1-\rho_i)}[\rho_i^2 - \sum_{j=1}^{N}\rho_{ij}^2],$$

and β *is* the mean of the packet service time, $\beta^{(2)}$ *is* the second moment of the packet service time, *s* is the first moment of the switchover time between queues and $s^{(2)}$*:* the second moment of the switchover time between queues. For simplicity, we assume that the switchover time is negligible and hence $s = s^{(2)} = 0$.

When node *i* sends a packet to node *j*, it travels the following path:

$$i \rightarrow (i+1) \bmod N \rightarrow (i+2) \bmod N \rightarrow \ldots \rightarrow j.$$

By definition, the average access delay is $E(W_{ii})$. The average end-to-end packet delay from node *i* to *node j*, $E(T_{j,i})$, is the summation of average queuing delay at all the nodes the packet traverses:

$$E(T_{j,i}) = E(W_{i,i}) + E(W_{(i+1)\bmod N,i}) + \ldots + E(W_{(j-1)\bmod N,i}). \tag{A.3}$$

The mean end-to-end queuing delay for packets originating at node *i* is given as the average over all the possible destinations:

$$E(T_i) = \frac{\sum_{j=1}^{N-1} E(T_{(i+j)\bmod N,i})}{N-1} \tag{A.4}$$

For a balanced network, all $E(T_i)$s are equal. Without loss of generality, we choose $i=1$ and derive the average end-to-end queuing delay *E(T)* as follows:

$$\begin{aligned} E(T) = E(T_1) &= \frac{E(T_{2,1}) + E(T_{3,1}) + E(T_{4,1}) + \ldots + E(T_{N,1})}{N-1} \\ &= \frac{\sum_{i=1}^{1}E(W_{i,1}) + \sum_{i=1}^{2}E(W_{i,1}) + \sum_{i=1}^{3}E(W_{i,1}) + \ldots + \sum_{i=1}^{N-1}E(W_{i,1})}{N-1} \\ &= \frac{1}{N-1}\sum_{i=1}^{N-1}(N-i)*E(W_{i,1}) \end{aligned} \tag{A.5}$$

The average distance a packet traverse is *(N/2)*. Thus, the end-to-end packet delay is:

$$E(D_{e2e}) = E(T) + \beta * (N/2) \, . \tag{A.6}$$

For RPR delay analysis, the mean delay for the node traffic, *E(W)*, and the mean delay for the ring traffic, *E(W')* are [11]:

$$E(W) = \frac{(\lambda_1+\lambda_2)\beta^2}{(1-\lambda_2\beta)[1-(\lambda_1+\lambda_2)\beta]}, E(W') = \frac{(\lambda_1+\lambda_2)\beta^2}{1-\lambda_2\beta}, \tag{A.7}$$

where λ_1 is the ring traffic rate and λ_2 is the node traffic rate. For symmetric traffic pattern, we have $\lambda_1 = \lambda(N/2-1)$ and $\lambda_2 = \lambda$. *E(W)* represents the average access delay. The average end-to-end delay has been given in [12]:

$$E(D_{e2e}) = E(W) + \beta + H*(E(W') + \beta) \tag{A.8}$$

where *E(W)* and *E(W')* are given in (A.7) and *H* can be interpreted as the mean number of intermediate nodes encountered by a packet on the ring. For symmetric traffic case $H = N/2-1$.

References

[1] "IEEE Standard 802.17-2004" June 2004, available at: http://www.ieee802.org/17.

[2] "An introduction to Resilient Packet Ring Technology," a white paper by the Resilient Packet Ring Alliance, Oct. 2001

[3] W. Bux, M. Schlater, "An approximation method for the performance analysis of Buffer Insertion Rings," *IEEE Trans. on Comm.,* vol. 31, no. 1, pp. 50-55, January, 1983.

[4] M. Francisco, F. Yuan and C. Huang, "A comparison of two buffer insertion ring architectures with fairness algorithms," *Proc. of IEEE ICC'03*, Anchorage, USA, May 2003, pp. 605-609.

[5] V. Gambiroza, P. Yuan and E. Knightly, "The IEEE 802.17 Media Access Protocoal for High-Speed Metropolitan-Area Resilient Racket Rings," to appear in *IEEE Network.*

[6] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, E. Knightly, "Design, Analysis, and Implementation of DVSR: A Fair, High Performance Protocol for Packet Rings," *IEEE/ACM Trans. on Networking,* 12(1), Feb. 2004.

[7] J. Wang and Y. Levy, "Managing performance using weighted round-robin," *Proceedings of the fifth IEEE Symposium on Computers and Communications, 2000*, pp. 785 –792.

[8] C. Huang, H. Peng, F. Yuan and J. Hawkins, "A steady state bound for resilient packet rings," *Proc. of IEEE Globecom'03,* San Francisco, USA, Dec. 2003, pp 4054 - 4058.

[9] H. Tang, "Fair Medium Access Control for Spatial Reuse Ring Netowrks," Ph.D. thesis proposal, Carleton University, Nov. 2003.

[10] S. W. Fuhrmann and Y.T. Wang, "Analysis of cyclic service systems with limited service: bounds and approximations," *Performance Evaluation* 9 (1988) 35-54.

[11] L. Kleinrock, *Queuing Systems*, Vol. 2: Computer Applications, (Willey-Interscience, 1976).

[12] D. A. Schupke and A. Riedl, "Packet Transfer Delay Comparison of a Store-and-Forward and a Cut-Through Resilient Packet Ring," *International Zurich Seminar on Broadband Communications (IZS)*, Zurich, Switzerland, February 19-21, 2002