

Virtual Queuing: An Efficient Algorithm for Bandwidth Management in Resilient Packet Rings

Arash Shokrani, Siavash Khorsandi, Ioannis Lambadaris, and Lutful Khan

Broadband Network Laboratory

Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada
{shokrani, khorsand, ioannis}@sce.carleton.ca, lkhan@radialink.com

Abstract- Resilient Packet Ring (RPR) is being devised as part of IEEE 802.17 standard, where fairness in bandwidth allocation among ring nodes, efficiency in resource utilization, and a low computational complexity are the main requirements. Although recent efforts have improved the performance of the RPR fairness algorithms to have acceptable steady-state behavior, we demonstrate that current algorithms suffer from extreme unfairness and throughput loss in some dynamic traffic scenarios. In this paper¹, we address the bandwidth management in RPR. First, we propose a general fairness model for packet rings. Then, a new algorithm for bandwidth management in RPR called Virtual Queuing (VQ) is introduced. We study the fairness properties of VQ algorithm both analytically and with simulation results. Compared to the RPR standard fairness algorithms that suffer from a throughput loss of up to 28% in some cases, the throughput loss with VQ is less than 2%. Comparing to another algorithm, called Distributed Virtual-time Scheduling in Rings (DVSR), VQ has a lower computational complexity and a better performance in a dynamic traffic environment. We show that the average throttled rate of the head node in a congestion span can be up to 80% for DVSR. With VQ, it is less than 4% in all cases.

I. INTRODUCTION

Resilient Packet Ring (RPR) IEEE 802.17 technology is a new MAC layer for metro-ring networks [1], [2]. It is aimed to provide high throughput, fault tolerance, and spatial reuse in a ring network. Spatial reuse, which is multiple concurrent transmissions over different parts of the ring, is provided in RPR by packet removal at the destinations. However, spatial reuse can result in congestion, and consequently unfairness among different nodes in accessing the ring bandwidth. Hence, a bandwidth allocation mechanism is required to ensure fairness among RPR nodes. Fairness in ring networks has been of particular attention [3]-[5]. It is also the main challenge in RPR [6].

The Bandwidth management in RPR is based on the concept of explicit rate feedback. When an RPR node (station) becomes congested, it calculates a fair rate and advertises it through a control message to its upstream nodes contributing to the congestion. When the upstream nodes receive the fair rate, they adjust the rate of their local traffic accordingly. As a result, the congested node will be able to add more of its local traffic to the ring.

¹ This research was supported by a grant from Communications and Information Technology Ontario (CITO).

The main proposed RPR fairness algorithms are the RPR draft standard algorithm, which operates in Aggressive Mode (RPR-AM) or Conservative Mode (RPR-CM), and Distributed Virtual-time Scheduling in Rings (DVSR) [1], [6], [7]. RPR-AM and RPR-CM algorithms suffer from extreme unfairness and throughput loss in some *dynamic* traffic scenarios. In particular, RPR-AM algorithm suffers from permanent oscillations and RPR-CM has a low convergence speed, which results in throughput loss in some unbalanced traffic scenarios [6]. DVSR is another fairness algorithm that performs the bandwidth allocation based on the concept of virtual-time. Compared to RPR-AM and RPR-CM algorithms, DVSR converges much faster. However, in a dynamic traffic environment, it can result in overestimation of the local fair rate and extreme starvation and unfairness at the local node.

In this paper, first we introduce a new fairness model for packet rings that is a generalization of Ring Ingress Aggregated with Spatial reuse (RIAS) fairness model in [6]. Then, we propose a new algorithm called Virtual Queuing (VQ) for fair rate calculation in RPR with a lower computational complexity, better fairness properties, and the same convergence scale as DVSR.

The rest of this paper is organized as follows. In Section II, the RPR node architecture is explained and a new mathematical model for fairness in RPR is proposed. Then the performance of the current RPR algorithms is evaluated. In Section III, we present Virtual Queuing algorithm and characterize its properties. Section IV contains our simulation results, and conclusions are drawn in Section V.

II. AN OVERVIEW OF RESILIENT PACKET RING

A. RPR Node Architecture

The ring in RPR is bidirectional and consists of two unidirectional counter-rotating ringlets. Fig. 1 illustrates the general RPR node architecture, where only the traffic of one of the ringlets is shown. At each node, the arriving traffic from the ring is dropped if destined to that node. Otherwise, it is forwarded to the transit buffer. The transit buffer may be implemented in two modes: single queue or dual queue. In single queue mode, the Low Priority (LP) and High Priority (HP) transit packets are forwarded into a single queue. In this mode, the scheduler gives the service priority to the transit traffic over the local traffic, which guarantees a lossless ring.

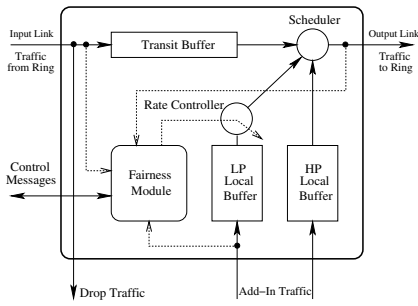


Fig. 1. The RPR node architecture

In dual queue mode, there are two transit queues for HP and LP transit traffic to improve the delay performance of the HP traffic [8]. In this mode, the scheduler gives the highest priority to the HP transit traffic. Then, packets from local buffers are forwarded prior to packets from the LP transit queue. When the LP transit queue is almost full, the scheduler gives the service priority to the transit queues over the local buffers.

An RPR node with a single transit queue is congested, when the utilization at the output link of its scheduler exceeds a certain threshold (e.g. 95% of the available bandwidth of the link) or if its local traffic experiences a long delay to access onto the ring. In dual queue mode, congestion is triggered when the LP transit queue depth exceeds a threshold. Note that in RPR, HP traffic has a *reserved* bandwidth throughout the ring. Hence, we do not consider HP traffic in our discussions in this paper. LP traffic is called fairness eligible (FE) as its rate is controlled by the fairness algorithm.

B. Fairness Model

In this section, we propose a fairness model for packet rings that is a generalization of RIAS fairness model in [6]. The stated objective of the bandwidth management mechanism in RPR is to provide per-station fairness in utilizing the ring bandwidth. Given the same per-station weights (or priority) and the same bandwidth demands, the ring bandwidth should be equally divided among the competing stations. Let us denote a source-destination flow from node i to node j by $f(i,j)$ and an Ingress Aggregated (IA) super-flow transiting node n and originated at node i by $sf(i,n)$. The ring bandwidth management consists of two components:

1) *Ring behavior*: It deals with calculating a fair rate at each node in order to maintain inter-station fairness. From a fairness point of view, IA super-flows are visible and not the individual source-destination flows. The advertised fair rates regulate the rate of IA super-flows so that none of the stations suffers from starvation.

2) *Source behavior*: It deals with intra-station fairness and allocating the bandwidth to the individual flows at each node based on the received fair rates from downstream nodes.

Source behavior should be optimized to maximize spatial reuse in the ring. That is, if some of long-haul flows face congestion, other flows destined to the nodes before the congestion point must be able to claim the unused bandwidth. However, fairness in the ring should not depend on any particular

assumption on the source behavior. RIAS fairness assumes a maximal spatial reuse source behavior. In the following, we provide an alternative fairness definition that does not have this constraint.

Let C be the available bandwidth on a link, also called *unreserved_rate*, r_i^n be the rate of $sf(i,n)$, and F_n be the local fair rate at node n . Note that r_i^n is a non-increasing function of n , that is, $r_i^n \geq r_i^{n+1}$. In the particular case when $n=i$, r_n^n represents the total rate of local fairness eligible traffic measured after the local rate controllers (shapers).

Definition 1: The set of rates $\mathbf{F}=\{F_n, \forall n\}$ and alternatively the matrix of super-flow rates $\mathbf{R}=\{r_i^n, \forall i,n\}$ are defined to be fair if they meet the following criteria simultaneously:

$$\begin{aligned} r_i^n &\leq F_n, \forall i,n, \\ F_n &= \max_min^*(C, \{r_i^n, \forall i\}) \\ &= \max_min(C, \{r_i^n, \forall i\}) + \alpha [C - \sum_i r_i^n]^+, \end{aligned} \quad (1)$$

where $[x]^+ = \max\{x, 0\}$ and $0 < \alpha \leq 1$. The distinction between \max_min^* and \max_min is essential. When the bandwidth is not utilized, \max_min^* results in a value that is larger than the rate of any of the current super-flows. This is controlled by parameter α . For $\alpha > 0$, the unused bandwidth can be claimed by the backlogged flows. As a result, any backlogged super-flow will have a bottleneck node m , where $\sum_i r_i^m = C$.

Let us assume that the local traffic is buffered on a per-destination basis at each station and $r_{i,j}$ be the service rate of queue j at node i . We have $r_i^n = \sum_{j>n} r_{i,j}$ and the set of rates $\{r_{i,j}, \forall j\}$ is called *feasible* if $\sum_{j>n} r_{i,j} \leq F_n, \forall i,n$. An example of a feasible source behavior is $r_{i,j} \leq \min_{i<n<j} \{F_n/N_n(i)\}$, where $N_n(i)$ is the number of flows at node n originated at station i . More details about this fairness model can be found in [10].

Example 1: Let us consider the scenario in Fig. 2 and assume a fair behavior at the ring level and that all flows are backlogged. Flow $f(1,30)$ is facing congestion in the downstream and can send only at 4% of the link capacity. Let $F_2=F_3=0.5$. With a maximal feasible source behavior, we have $r_{1,30}=0.04$, $r_{1,4}=0.46$, and $r_{2,4}=0.5$. With the source behavior, which is given above as an example of a feasible behavior, we have $r_{1,30}=0.04$, $r_{1,4}=0.25$, $r_{2,4}=0.5$. This is not a fair solution since (1) is not satisfied for $n=2$ and $n=3$, that requires $F_2 > 0.5$ and $F_3 > 0.5$. The solution that satisfies (1) is $F_2=F_3=0.64$, $r_{1,30}=0.04$, $r_{1,4}=0.32$, and $r_{2,4}=0.64$. From a ring point of view, this is also a fair solution.

C. Current RPR Fairness Algorithms

1) *RPR-AM*: Let *add-rate* be the insertion rate of FE traffic at the local node and *forward_rate* be the service rate of its transit traffic, which are measured by each node over a fixed time-interval length of T seconds called *aging_interval* (or Co-

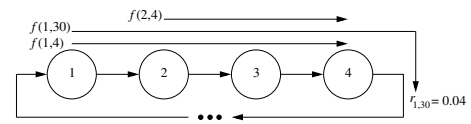


Fig. 2. A scenario to compare feasible and maximally feasible source behaviors

ntrol Interval). At the end of each *aging interval*, an RPR node checks its congestion status. In RPR-AM, a congested node advertises the low-pass-filtered of its *add_rate* as the fair rate to its upstream nodes. The low-pass-filtering is an exponential averaging with parameter *lpCoef* as explained in [7]. When congestion clears the fair rate is set to *uneserved_rate*, C . The nodes in the congestion span directly apply the received fair rates to their local FE traffic, if it is less than C . Otherwise, their allowed insertion rate for FE traffic is ramped up.

2) *RPR-CM*: In Conservative Mode, when congestion is detected for the first time, the local fair rate is set to an initial value of C divided by the number of active stations. An active station is the one that has transmitted at least one packet over the past *aging interval*. This fair rate is advertised to the upstream nodes. If the congested node remains congested and *add_rate+forward_rate* is greater than a high threshold (e.g. $0.95 \cdot C$), the fair rate is ramped down. If *add_rate+forward_rate* is less than a low threshold (e.g. $0.80 \cdot C$), the fair rate is ramped up. This process is iterated only once in every fairness round trip time (FRTT) to allow the network to settle with the new rates.

3) *DVSR*: In DVSR, rates of individual IA flows are measured. The fair rate is calculated as a function of these measured rates. Hence, a better decision in terms of the fair rate calculation is made at the expense of further complexity of measuring per-flow rates. Assume an arbitrary node n , and let r_i^n be the arrival rate of traffic to node n originating at node i including node n . It can be shown that the DVSR fair rate, F_n , satisfies the following conditions [6]:

$$F_n = \begin{cases} \max\{r_i^n\} + (C - \sum_i r_i^n), & \text{if } \sum_i r_i^n < C, \\ (C - \sum_{\forall r_i^n < F_n} r_i^n) / I, & \text{otherwise} \end{cases} \quad (2)$$

where I is the number of flows with rates more than or equal to F_n , that is, $I = |\{i: r_i^n \geq F_n\}|$. This is in fact a *max-min** operation on the observed traffic rates of the stations. It is implicitly assumed that r_i^n represents the bandwidth demand of station i . Note that F_n cannot simply be calculated from (2) as it has a recursive form. The algorithm for calculation of the fair rate in DVSR requires a sort and a loop operation with complexity $O(N \log_2 N)$, where N is the number of IA flows at node n [6]. The complexity of RPR-AM and RPR-CM algorithms is invariant of N .

D. Evaluation of the Current RPR Fairness Algorithms in Dynamic Traffic Scenarios

To evaluate these schemes, we consider the *parking lot* scenario of Fig. 3, which is widely studied as a benchmark scenario [6]. In an n -node parking lot, nodes $1, 2, \dots, n$ send traffic to node $n+1$, where node n is the head node and node 1 is the tail node in the congestion span. We consider a simple 2-state source model called on-off source. It fluctuates between a high rate, S_h , and a low rate, S_l . Let T_h and T_l be the sojourn times of the high (on) and the low (off) states, respectively. In the following, we discuss the potential performance problems in RPR-AM, RPR-CM, and DVSR algorithms.

1) To evaluate CM-RPR algorithm, a 2-node parking lot scenario was considered. An on-off source is introduced to the head node (node 2) and node 1 can send at C . The simulation parameters are as follows: $C=100\text{Mbps}$, $T=1\text{msec}$, $lpCoef=16$, $rampUpCoef=rampDnCoef=32$, $S_h=C/2$, $S_l=0.5\text{Mbps}$, $T_h=2T$, and $T_l=T$. Fig. 4 shows the output link utilization at node 2. Due to slow convergence, CM-RPR cannot adapt to traffic changes quickly enough. In this case, a throughput loss of 27% is observed

2) For RPR-AM algorithm, we take the same scenario as for RPR-CM except that $T_l = 10T$ to see the impact of traffic imbalance on the throughput. Fig. 5 presents the advertised fair rate of the head node. This rate is not applied at the head node according to the RPR standard. Due to the oscillations of the fair rate, a total throughput loss of 16% is observed.

3) DVSR is more vulnerable to traffic variations towards the tail. A 4-node parking lot scenario was considered. The on-off source is introduced at the tail (node 1), while other nodes can send at C . We have $T=1\text{msec}$, $S_h=C/4$, $S_l=C/10$, and $T_h=T_l=T$. Fig. 6 depicts the advertised fair rate and also throughput of the head node. Due to overestimation of the fair rate, the head node faces periodic starvations and has received substantially a lower share of the bandwidth than its upstream nodes. The average throttled rate of the head node is 38%. This will be further discussed in Section III-C.

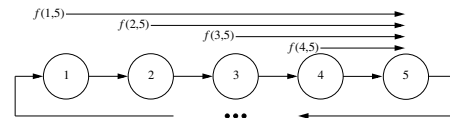


Fig. 3. Parking lot scenario

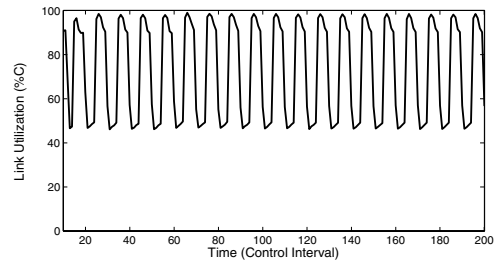


Fig. 4. Link utilization of the head node in CM-RPR

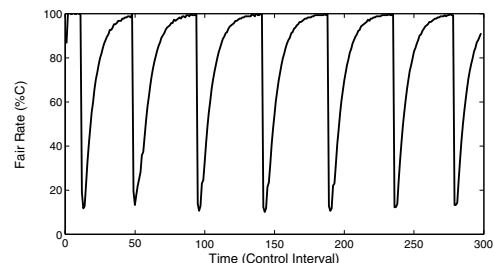


Fig. 5. Oscillations of the RPR-AM fair rate

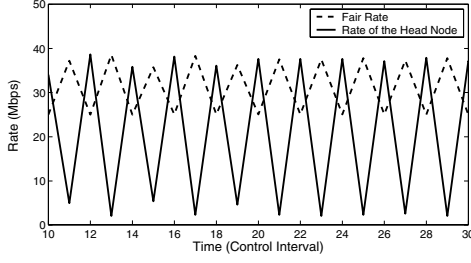


Fig. 6. The DVSR fair rate and the rate of node 4

III. VIRTUAL QUEUEING: A DISTRIBUTED FAIR BANDWIDTH MANAGEMENT ALGORITHM FOR RPR

A. VQ Algorithm Description

In RPR, we are dealing with a distributed rate control system (See Fig. 7). To achieve fairness, a global fair rate should be calculated so that if all of the rate controllers are set accordingly, congestion does not occur. The fair rate calculation algorithm should be designed in such a way that a distributed implementation with a limited complexity, speedily converges (in an average sense) to a solution that is close to the fair solution. The observed traffic at a node ($r_i^n(k)$) is not an accurate indication of the bandwidth demand of a station, as it is subject to constraints imposed by the advertised fair rate and also by the limitation of the available bandwidth. Therefore, applying \max_min^* to the observed rates does not necessarily result in the global fairness. Several iterations may be needed to adjust the errors in demand estimations. A more accurate estimation of the demands results in a faster convergence.

The average rate of the arrival process to an arbitrary node n during control interval k is equal to

$$r^n(k) = \sum_i r_i^n(k) = \sum_{i \neq n} r_i^n(k) + r_n^n(k). \quad (3)$$

The first term in (3) is the average rate of the transit traffic at node n and the second term is the average rate of the local traffic at node n to the *virtual queue*, in control interval k . We have $\sum_{i \neq n} r_i^n(k) \leq C$ and $r_n^n(k) \leq F_n(k)$. Note that $r_n^n(k)$ is the amount of the local traffic conforming to the current fair rate, whether it is serviced or buffered. Therefore, $r^n(k) \leq C + F_n(k)$. When $r^n(k) > C$, the local node does not receive its fair share. The eligible traffic in the excess of the available bandwidth is buffered at the local node. Therefore, each node can be modeled as a virtual queue with a service rate of C and the average arrival rate of $r^n(k)$.

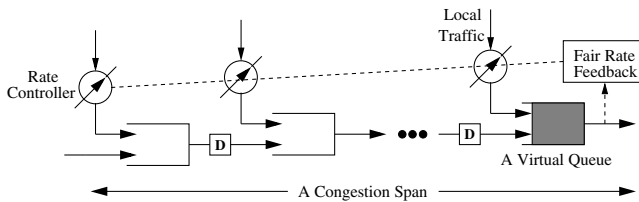


Fig. 7. A distributed rate control Model

Proposition 1: Any ring fairness algorithm that maintains $\sum_i r_i^n(k) \leq C, \forall n$, results in a fair solution.

Proof: Assume an arbitrary node n and let $r_{\max} = \max_i \{r_i^n(k)\}$. If $\sum_i r_i^n(k) < C$, we have $F_n(k) > r_{\max}$, $r_{\max} = \max_min(C, \{r_i^n, \forall i\})$ and we can find $0 < \alpha \leq 1$ so that (1) is satisfied. Hence, it is a fair solution. If $\sum_i r_i^n(k) = C$, then we have $F_n(k) = r_{\max}$. In this case, we can write $r_{\max} = (C - \sum_{i \in V} r_i^n(k)) / |\{i: r_i^n(k) = r_{\max}\}| = \max_min(C, \{r_i^n, \forall i\})$, where $V = \{i: r_i^n(k) < r_{\max}\}$. As $\sum_i r_i^n(k) = C$, $F_n(k) = r_{\max} = \max_min^*(C, \{r_i^n(k), \forall i\})$, which is a fair solution based on Definition 1. ■

The ramification of this proposition is significant. It indicates that any rate control mechanism that stabilizes the virtual queue at every node will result in a fair solution. However, a speedy convergence is the main requirement. We have already demonstrated that performing \max_min^* operation on the observed rates may overestimate the fair rate in a dynamic traffic environment. In designing VQ algorithm, instead of taking the observed rates as the bandwidth demands, all of the flows sending at or above the current fair rate are assumed to be backlogged. Hence, their bandwidth demands over the next control period are assumed to be equal to C . The goal is to match the total arrival rate of the virtual queue at node n to the available bandwidth, that is, $r^n(k) \leq C$. VQ is an algorithm that performs this in a simple manner.

Let $F_n(k)$ be the fair rate advertised by congested node n for control interval k . We need the following measured parameters at node n in control interval k to calculate the fair rate for control interval $k+1$:

- $N_i(k)$: Traffic received at congested node n from node i .
- $N_n(k)$: The maximum of local queue size and local traffic entering the virtual queue at node n .
- E_i : Total amount of arriving traffic from node i to congested node n , conforming to the last advertised fair rate, $F_n(k)$.
- E_S : Total amount of arriving traffic from rate-limited nodes.
- E_U : Total amount of arriving traffic from input-limited nodes.

The rate of input-limited nodes is below the advertised fair rate and a flow is considered to be rate-limited if it fully utilizes its available fair share. Table 1 shows the fair rate calculation procedure in VQ algorithm.

Table 1: VQ fair rate calculation algorithm

$$\begin{aligned}
 N_i(k) &= r_i^n(k) \cdot T \\
 N_n(k) &= \max\{r_n^n(k) \cdot T, local_queue_size\} \\
 E_i &= \min\{N_i, T \cdot F_n(k)\}, \forall i \\
 S &= \{i: E_i \geq T \cdot F_n(k)\} \\
 U &= \{i: E_i < T \cdot F_n(k)\} \\
 E_S &= \sum_{i \in S} E_i \\
 E_U &= \sum_{i \in U} E_i \\
 \text{if } (E_S \neq 0 \ \&\& \ E_U < C \cdot T) & \quad f = (C \cdot T - E_U) / E_S \quad (7) \\
 \text{if } (E_U \geq C \cdot T) & \quad f = (C \cdot T) / (E_U + E_S) \\
 \text{if } (E_S = 0 \ \&\& \ E_U < C \cdot T) & \quad f = 1 \\
 F_n(k+1) &= \min\{f \cdot F_n(k), C\}
 \end{aligned}$$

This algorithm is robust with respect to the size of the control interval, T . Only the eligible part of traffic is used in the fair rate calculation. At the steady state, E_i is equal to N_i but in the transient period, N_i can exceed E_i before the congestion feedback takes effect. However, for the purpose of the fair rate adjustment, the latest fair rate is applied.

E_S is the total amount of arriving traffic from rate-limited nodes and E_U is the total amount of arriving traffic from input-limited nodes. When $E_S=0$ and $E_U < C \cdot T$, none of the flows is throttled. Therefore, the node does not need to change the fair rate and the rate increase/decrease factor, f , is equal to 1. When $E_U > C \cdot T$, the fair rate is very large and should be rapidly decreased to match the total rate of incoming traffic to the ring capacity. When $E_S > 0$ and $E_U < C \cdot T$, the bandwidth demand of the input-limited flows is deducted from the ring capacity and the remaining bandwidth is divided among the rate-limited nodes. It can be shown that this is equivalent to a *max_min** operation with a low computational complexity. The difference between VQ and DVSR is in setting the parameter α in (1). In DVSR it is set to 1, while in VQ algorithm it is dynamically set to $\alpha=1/|S|$. When there are N active stations contributing to the congestion at node n , the fair rate will be no less than C/N , that is, $F_n \geq C/N$. For $F_n(k) = C/N$, we have $f \geq 1$. Therefore, $F_n(k+1) \geq C/N$.

B. Rate Feedback Mechanism

VQ is a fairness algorithm designed to work within current RPR framework. When a node is congested, special control packets are dispatched every T seconds (control interval) to carry advertised fair rates of the stations. Each station will write its advertised fair rate in a designated field in the control packet passing through that node. If a control packet is not received during the past T seconds, the station creates a new control packet and forwards it to upstream nodes. The size of the control interval has an impact on the amount of control overhead, which should be limited to a reasonable value.

C. Characterizing Fairness Properties of VQ

From a fairness point of view, the head node in a congestion span receives the worst performance. We consider the amount of the local traffic of the head node throttled below its fair share to characterize the fairness properties of VQ algorithm. The accumulated throttled traffic (ATT) of the head node over a period of time is expressed in bytes and is equal to the increase in the size of the virtual queue at the head node over that period (Fig. 7). Also, the average amount of throttled traffic of the head node in the unit of time, $\langle ATT \rangle$, is expressed in bps and measures the under-allocation of the head node. Let $r(k)$ be the service rate of the local traffic of the head node, and $F(k)$ be its fair share in control interval k . ATT and $\langle ATT \rangle$ can be calculated as follows:

$$ATT(L) = \sum_{k=1}^L (F(k) - r(k)) \cdot T, \quad (9)$$

$$\langle ATT \rangle (L) = \sum_{k=1}^L (F(k) - r(k)) / L. \quad (10)$$

We can express $\langle ATT \rangle$ in percentage of the average service rate of the head node as follows:

$$\langle ATT \rangle \% = 100 \cdot \langle ATT \rangle / (\sum_{k=1}^L r(k) / L). \quad (11)$$

The amount of throttled traffic of the head node depends on the network and the traffic scenario. However, to draw some general properties, we consider the following three cases: *steady state*, where all of the flows keep their current rates; *turn-on*, where some of the flows increase their rates by C_u in total; and *turn-off*, where some of the flows decrease their rates by C_d in total.

The following propositions provide some upper bounds for the accumulated throttled traffic (ATT) of the head node in VQ and DVSR algorithms in a dynamic traffic environment, where the bandwidth demand of the stations may increase (turn-on) or decrease (turn-off). The proofs of these propositions are not given in this paper due to the space limitations. Details are available in [9]. Let $N > 0$ be the number of backlogged flows, M be the total number of active nodes, R be the rate of the local traffic of the head node, and $g(x,y,z) = 0.5 \cdot \epsilon \cdot (2+\epsilon)/(1+\epsilon) + x \cdot [\log(\min\{y, 1/x\}/z) / \log(1+x)]^+$, where $\epsilon = \lfloor y / \max\{z, 1/x\} - 1 \rfloor^+$.

Proposition 2: In a congestion period, where the total demand exceeds the available bandwidth, we have the following properties for VQ algorithm:

- 1) For any consecutive turn-on control intervals (without a turn-off), where the rate of the stations increases by C_u in total, we have $ATT_{VQ,on} < g(C/R, C/\max\{C/M, (C-C_u)/N\}, N) \cdot C \cdot T$.
- 2) For any consecutive turn-off control intervals (without a turn-on), where the rate of the stations decreases by C_d in total, we have $ATT_{VQ,off} = 0$.
- 3) Over a period of time with a turn-on and a turn-off sequence, which is called a cycle, we have $ATT_{VQ,cycle} < g(C/R, C/\max\{C/M, (C-C_u)/N\}, N) \cdot C \cdot T - C_d \cdot T$.

Proposition 3: In a congestion period, we have the following properties for DVSR algorithm:

- 1) For any consecutive turn-on control intervals (without a turn-off), where the rate of stations increases by C_u in total, we have $ATT_{DVSR,on} = ATT_{VQ,on}$.
- 2) For a turn-off control interval where the rate of stations decreases by C_d in total, it can be shown that $ATT_{DVSR,off} < g(C/R, N, N/(1+(N-1) \cdot C_d/C)) \cdot C \cdot T$.
- 3) Over a period of time with a turn-on and a turn-off sequence, we have $ATT_{DVSR,cycle} = ATT_{VQ,cycle} + ATT_{DVSR,off}$ and $ATT_{DVSR,cycle} > 0$.

From Proposition 3, the accumulated throttled traffic of the head node over a cycle is always non-zero for DVSR. Hence, we have shown that the bounds given in [6] for a single node cannot be extended to a multi-node ring scenario. In fact, the ATT of the head node may not be bounded over a period of congestion and a permanent deviation from the fair rates can occur. However, ATT can be zero in VQ over a period of time. In some scenarios, it may grow, but with a much slower pace compared to that of DVSR. Fig. 8 illustrates the ATT bounds of a cycle for VQ and DVSR with $N = 3$ and $M = 10$ for $R = 0.1$ and $R = 1$ as a function of $C_u (= C_d)$. All rates are normalized to the link rate. A point of diminishing and no return is observed at $C_u = 0.7$ for this particular case. During turn-on period, for $C_u \geq 0.7$ the final fair rate and hence $ATT_{VQ,on}$ remains flat but since C_d increases, it results in decreasing $ATT_{VQ,cycle}$ for VQ.

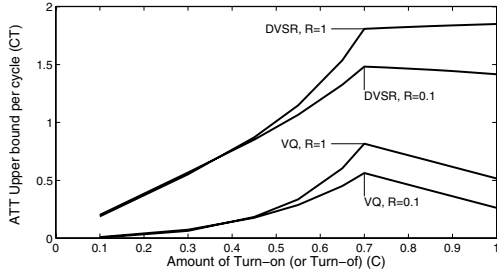


Fig. 8. ATT Upper bound per turn-on/turn-off cycle

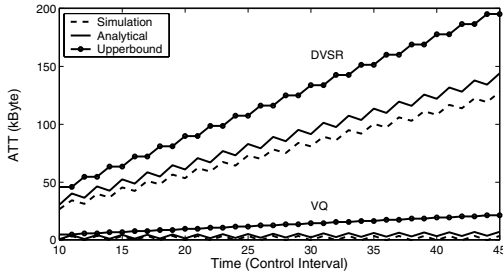


Fig. 9. Comparing *ATT* upper bound with simulation and analytical results

To show the tightness of the upper bounds, we compare these bounds with the analytical and the simulation results for some worst-case scenarios. Fig. 9 shows the results for a 4-node parking lot scenario of Fig. 3 with an on-off source at the tail node. The simulation parameters are as described in Section II-D. A close match between the model and the simulation results, and an acceptable margin for the upper bound are observed. It can be seen that when the tail node is in the low state, *ATT* decreases as the head node is able to send more traffic than its fair share. It should be noted that the difference between the upper bound and the actual values is accumulated over time.

IV. SIMULATION RESULTS

We conducted simulations in OPNET to compare the performance of VQ, DVSR and the RPR standard algorithms. We considered 100Mbps links with 50 μ sec propagation delay (i.e., a distance of 10km between each pair of nodes) and control interval (T) of 1msec. Our study is concentrated on convergence times and fairness properties of these algorithms in overload conditions, as these conditions pose greater problems and are more pressing. The simulations are preformed with a small packet size of 512 bits to minimize the effect of the packet size.

We considered a homogeneous traffic in a 4-node parking lot scenario (Fig. 3), where each node 1 to 4 generated 100Mbps of low priority traffic meaning that the network was overloaded to 400%. Nodes 4, 3, 2, and 1 start sending traffic at 0, 0.1, 0.2 and 0.3 seconds, respectively. Fig. 10 shows the normalized service rate of the head node in VQ. When N nodes are on, the fair rate is C/N . This figure shows speedy convergence in VQ after each change in the traffic.

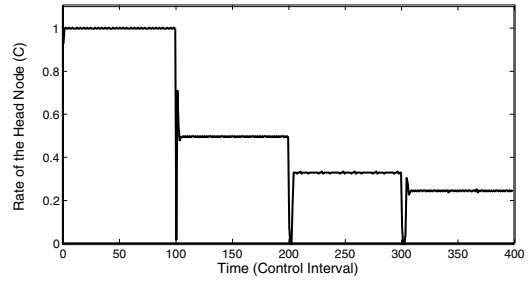


Fig. 10. Convergence of VQ with homogeneous traffic

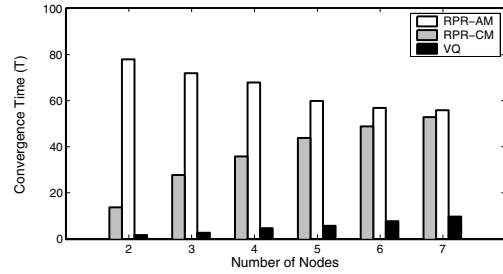


Fig. 11. Convergence time in the synchronized parking lot

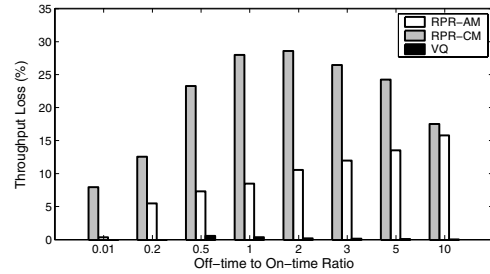


Fig. 12. Comparing throughput loss of AM, CM and VQ

Fig. 11 compares convergence times of RPR-CM, RPR-AM and VQ algorithm in a synchronized parking lot scenario with 3-8 nodes, where all nodes start sending traffic at the same time. In this scenario $lpCoef = 16$, and $rampUpCoef = rampDnCoef = 32$. The convergence time of RPR-AM decreases with the number of nodes. The reason is that the final fair rate decreases with the number of nodes and it is achieved faster in low pass filtering process. However, convergence time in CM-RPR increases with the number of nodes. It can be seen that the convergence time of VQ is much less than RPR-CM and RPR-AM.

Due to the slow convergence, RPR-CM and RPR-AM algorithms result in throughput loss with dynamic traffic.

Fig. 12 compares the throughput loss of these algorithms in a 2-node parking lot scenario, where an on-off source is introduced at the head node. The on-off source is generating 50Mbps traffic in the high state and 0.5 Mbps in the low state. The upstream node is backlogged. We have $T_h = 5T$, while T_l varied from $0.01T_h$ to $10T_h$. The throughput loss with CM-RPR is up to 28% and with AM-RPR up to 17%. CM-RPR shows a non-monotonic behavior with off-time to on-time ratio (T_l/T_h). When the off-time is large, RPR-CM converges and throughput loss is reduced. For small off-times, the released band-

width by the downstream node is not that much and hence the loss is less. However, RPR-AM does not converge and the throughput loss increases with the off-time. In VQ, the throughput loss is less than 2% in all cases.

To compare VQ and DVSR, we considered a 4-node parking lot topology (Fig. 3), where an on-off source with $S_h=25\text{Mbps}$, $S_f=10\text{Mbps}$, and $T_h=T_f$ is introduced at the tail. The other nodes are backlogged. In this scenario, T_h+T_f is changed from $2T$ to $16T$. Fig. 13 shows $\langle ATT \rangle$ in percentage of the average service rate of the head node, $\langle ATT \rangle\%$. It can be seen that as T_h+T_f is increased, $\langle ATT \rangle\%$ reduces as DVSR converges. In the next scenario we keep $T_h+T_f=2T$, and change the number of nodes (M) from 2 to 7. The on-off source is always introduced at the tail with $S_h=100/M$ Mbps and $S_f=0.5\text{Mbps}$, while the other nodes are backlogged. Fig. 14 shows $\langle ATT \rangle\%$ for DVSR and VQ. It can be seen that $\langle ATT \rangle\%$ in DVSR drastically grows with the number of nodes. This increase is much less for VQ. As it is shown in Figs. 13, and 14, the average throttled rate of the head node in VQ is less than 4% in all cases.

V. CONCLUSION

In this paper, we first proposed a fairness model for packet rings and then a new algorithm for fair rate calculation in RPR called Virtual Queuing (VQ) was presented. It is based on a virtual queuing model of the system. The calculated fair rate aims at stabilizing the queuing system and maximizing the utilization. The convergence speed of VQ is much better than the RPR standard algorithms and is same as DVSR, while it is computationally less complex than DVSR, as it does not require a sort operation. It was also shown through simulations and analysis that VQ has better fairness properties in a dynamic traffic environment. As such, VQ is a powerful technique for fair and efficient bandwidth management in packet rings.

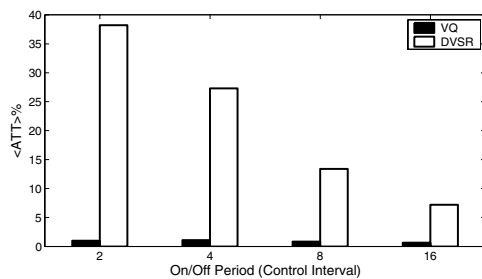


Fig. 13. Comparing $\langle ATT \rangle\%$ of DVSR and VQ for different On-Off period

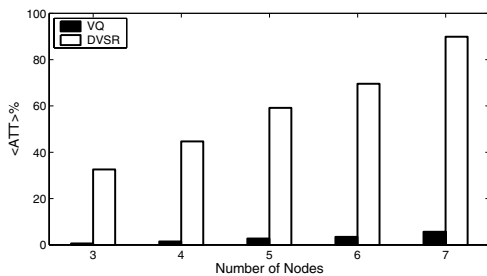


Fig. 14. Comparing $\langle ATT \rangle\%$ of DVSR and VQ for the different number of nodes

REFERENCES

- [1] F. Davik, M. Yelmez, S. Gjessing and N.Uzun, "IEEE 802.17 Resilient packet rings tutorial," *IEEE Communications Magazine*, March 2004.
- [2] M. Karol, and R. Gitlin, "High performance optical, local and metropolitan area networks: Enhancements of FDDI and IEEE 802.6 DQDB," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 8, pp. 1439-1449, October 1990.
- [3] I. Cideon, and Y. Ofek, "MetaRing – a full-duplex ring with fairness and spetial reuse," *IEEE Transaction on Communications*, vol. 41, no. 1, pp. 110-120, January 1993.
- [4] D. Tsiang, and G. Suwala, "The Cisco SRP MAC layer protocol," *IETF RFC 2892*, August 2000.
- [5] I. Cidon, L. Georgiadis, R. Guerin, and Y.Shavitt, "Improved fairness algorithms for rings with spatial reuse," *IEEE/ACM Transactions on Networking* 5(2), April 1997.
- [6] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly, "Design, analysis, and implementation of DVSR: A fair, high performance protocol for packet rings," *IEEE/ACM Transactions on Networking* 12(1), February 2004.
- [7] IEEE. IEEE Standard 802.17: Resilient Packet Ring (D3.0), <http://ieee802.org/17>, November 2003.
- [8] D. Schupke, 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, Februray 19-21, 2002.
- [9] S. Khorsandi, A. Shokrani, and I. Lambadaris, "A Novel Fairness Model and an Efficient Algorithm for Bandwidth Management in Resilient Packet Rings," Carleton University Technical Report, August 2004.
- [10] S. Khorsandi, A. Shokrani, and I. Lambadaris, "A New Fairness Model for Resilient Packet Rings," accepted for presentation in *ICC05*, Seoul, Korea, May 2005.