

On Modeling of Fair Rate Calculation in Resilient Packet Rings*

Arash Shokrani, Ioannis Lambadaris, and Jérôme Talim
Broadband Networks Laboratory

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

Abstract

The Resilient Packet Ring (RPR) IEEE 802.17 is a new Medium Access Control (MAC) protocol for high-speed ring networks. In order to achieve fairness among nodes, a fairness algorithm is employed at each RPR node. In this paper, we develop an analytical model for fair rate calculation in the standard RPR Aggressive fairness algorithm in parking lot scenario. Our approach is to model the fair rate calculation by a non-linear discrete-time low-pass filter. We verify our model by simulation results and analyze the convergence of the fairness algorithm. Further, the effect of various parameters on convergence time is investigated.

1. Introduction

The Resilient Packet Ring (RPR) IEEE 802.17 standard is a new Medium Access Control (MAC) protocol for metro-ring networks. It is devised to achieve many objectives such as high utilization, fault recovery, and fairness, which are not simultaneously available in the current technologies [5],[2]. Packets are removed from the ring at their destinations and as a result RPR can support *spatial reuse*, that is, multiple concurrent transmissions over different parts of the ring [1]. However, it may cause congestion and unfairness among different nodes in accessing the ring bandwidth. Therefore, a bandwidth allocation (fairness) algorithm is required to provide a fair ring access to all nodes.

The ring access scheme in RPR is based on Buffer Insertion Ring (BIR) method, in which every node (station) on the ring has an insertion (transit) buffer [7]. The ring traffic transiting a node may be temporarily stored in the transit buffer. A node is allowed to add its local traffic (add traffic) to the ring as long as its transit buffer is empty. In other words, ring traffic has non-preemptive priority over local traffic at each node. This means that the transit traffic at each node can block the local traffic of that node in

accessing onto the ring. Hence, all nodes should be forced to adjust the insertion rate of their local traffic (add-rate) according to their fair shares.

The objective of the fairness algorithm in RPR is to fairly distribute the available bandwidth on any link between the local traffic of all competing nodes on that link. We define the true *fair rate* as the rate at which every node should add its local traffic to the ring, without starving its downstream nodes. When a node is congested, it calculates a first approximation to the true fair rate since the exact value of the true fair may not be known in case of congestion. The calculated fair rate is then advertised to the upstream nodes through a control message. As the upstream nodes receive the control message, they adjust their add-rates according to the received fair rate. This process may repeat until the calculated fair rate converges to the true fair rate. The time it takes for the fairness algorithm to converge is commonly referred to as the *convergence time*, which is an important criterion in evaluating performance and efficiency of an RPR fairness algorithm.

In [2] and [3], convergence times of different RPR fairness algorithms are investigated based on the simulation results. However, an accurate analytical model would provide a solution to evaluate the performance of different RPR fairness algorithms without having to develop a simulation model. In this paper, we present an *accurate* analytical model for fair rate calculation in the standard RPR fairness algorithm. This analysis is done for the parking lot scenario depicted in Fig. 1, in which all nodes 1, 2, ..., and n send traffic to node 0 (Hub), sharing the link between node 1 and node 0. This scenario is one of the traditional and challenging benchmark scenarios in RPR, and is studied in

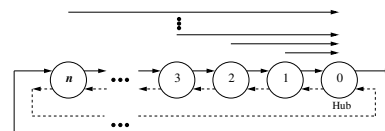


Figure 1. Parking lot scenario

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

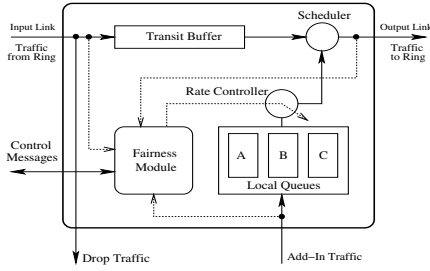


Figure 2. RPR node architecture

standardization process of RPR, as well as in [3], to compare performance of different RPR fairness algorithms. Using the analytical model, we study the performance of the RPR standard fairness algorithm in terms of stability, and convergence time. We also investigate the impact of various system parameters on the performance of the fairness algorithm.

In Section 2 the RPR node architecture, the notion of fairness in RPR, and the RPR standard fairness algorithm are described. In Section 3, we present our analytical model of fair rate calculation. Section 4 deals with the numerical results. Finally, some conclusions are given in Section 5.

2. An Overview of Resilient Packet Ring

RPR supports three traffic classes: high-priority low-delay class A; medium-priority bounded-delay class B; and low-priority best-effort class C. The bandwidth is allocated for class A and a subclass of class B. However, a portion of this allocated bandwidth is not reserved and can be reclaimed by the lower priority classes if not used. The fairness algorithm dynamically divides the unreserved bandwidth between the remaining portion of class B and class C traffic of the competing nodes. The traffic controlled by the fairness algorithm is called fairness eligible (FE).

The ring in RPR is bidirectional and consists of two unidirectional counter-rotating ringlets. Fig. 2 presents an RPR node architecture, where only one direction of the ring traffic 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, transit packets with different priorities 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. The fairness module implements the fairness algorithm, transmits control messages, and controls the rate limiters of the local traffic.

2.1. RPR Fairness Model

The objective of the fairness algorithm in RPR is to fairly distribute the available bandwidth on a given link between FE traffic of all competing nodes on that link [4]. The bandwidth share of each node on a link should be proportional to the relative weight assigned to that node. In other words, a per-station fairness in utilizing the ring bandwidth should be maintained. In this paper, we only consider FE traffic as the rate of the reserved/allocated traffic is not controlled by the fairness algorithm.

2.2. RPR Standard Fairness Algorithm

Congestion detection and fair rate calculation in RPR nodes are performed at the end of every fixed time interval called *control interval*. Each RPR node measures the service rate of its FE local traffic and that of the transit traffic, at the output of the scheduler. The *add-rate* (respectively, the *forward-rate*) is the *average* amount of FE local traffic (respectively, transit traffic) serviced per unit of time; both are measured at the end of each control interval. Let us define the following notations:

- C is the available bandwidth on a link.
- Control interval $k > 0$ with length of T seconds, is the time interval $((k-1) \cdot T, k \cdot T]$.
- α is a low-pass filtering coefficient ($0 < \alpha < 1$).
- $r_i^a(k)$ is the add-rate of node i in control interval k and $\tilde{r}_i^a(k)$ is its low-pass filtered version defined as follows:

$$\tilde{r}_i^a(k) = (1 - \alpha) \cdot \tilde{r}_i^a(k-1) + \alpha \cdot r_i^a(k). \quad (1)$$

- $r_i^f(k)$ is the forward-rate of node i in control interval k and $\tilde{r}_i^f(k)$ is its low-pass filtered obtained similar to (1).
- $u_i(k) = r_i^a(k) + r_i^f(k)$ is the usage rate of the output link of node i in control interval k , and $\tilde{u}_i(k)$ is its low-pass filtered version. Intuitively, we have

$$\tilde{u}_i(k) = \tilde{r}_i^a(k) + \tilde{r}_i^f(k) = (1 - \alpha) \cdot \tilde{u}_i(k-1) + \alpha \cdot u_i(k). \quad (2)$$

For simplicity, assume that all rates are normalized to C . Therefore, $u_i(k) \leq 1$ and $r_i^a(k) \leq 1 - r_i^f(k)$.

1) *Congestion Detection*: Assume that node i is implemented in *single-queue* mode. At the end of control interval k , node i detects congestion if $\tilde{u}_i(k)$ exceeds a certain threshold τ (e.g., $0.95 \cdot C$); or a packet at the head of local buffers of that node experiences a long delay to get onto the ring.

2) *Fair Rate Calculation*: The RPR standard fairness algorithm operates in two modes: *Aggressive Mode* (AM) and *Conservative Mode* (CM) [4]. We assume that all nodes are operating in AM, as this is the default mode of operation in RPR. In AM, when node i detects congestion in control interval k , it advertises $\tilde{r}_i^a(k)$ as the fair rate to its upstream nodes. If node i is not congested, the fair rate is set to 1.

3) *Control Message Transmission*: We assume that there is a single control message circulating along the ring within each control interval. The control message contains separate fields for the calculated fair rate of all nodes. At the end of each control interval, one of the nodes initiates transmission of the control message. It inserts its advertised fair rate in a designated field and sends the control message to its upstream node. Upon receiving a control message, node i adjusts the rate of its local FE traffic so that its add-rate for destinations, which are in the downstream of node j , does not exceed the advertised fair rate of node j . It also updates the corresponding field in the control message with its most recently calculated fair rate and forwards the message to its upstream node.

3. Problem Description

We consider the parking lot scenario depicted in Fig. 1, which is a benchmark scenario for evaluating an RPR fairness algorithm. The performance of an RPR fairness algorithm should be evaluated in overloaded conditions. Otherwise congestion does not occur and there is no need for a rate control mechanism. Hence, we assume that nodes 1, 2, ..., and n in Fig. 1 have infinite amount of traffic in their local buffers destined to node 0. In Fig. 1, the solid-line ringlet is the traffic path to node 0. The control message is transmitted backward on the dashed-line ringlet. Without loss of generality, we only consider FE traffic as the fairness algorithm only controls the rate of FE traffic throughout the ring. All nodes compete for the bandwidth on the link between node 1 and node 0. As a result, this link is the most congested link. We assume that all nodes have equal weights. Therefore, the fair share of each node is $\frac{1}{n}$ of the link bandwidth. We also assume that the fairness algorithm operates in AM. Note that the transit queue mode is independent of the fairness algorithm and only specifies how the congestion is triggered [4]. In this analysis, we assume that the transit queue at each node is implemented in *single-queue* mode.

In the following, we first assume that the link propagation delay is zero and derive the fair rate equation. Then, we incorporate a deterministic link propagation delay into the model and analyze the problem in a more realistic case.

3.1. Analysis of the Model Without Link Propagation Delay

In this section we ignore the link propagation delay in fair rate calculation and then determine the conditions with respect to which the convergence time of the fairness algorithm is finite. Let $\ell_i(k)$ be the rate limiter value of the local traffic of node i at the beginning of control interval k . This is the maximum rate at which node i is allowed to add traffic to the ring destined to node 0. The add-rate of node i in control interval k is also bounded by $1 - r_i^f(k)$ and hence we have

$$r_i^a(k) = \min \{1 - r_i^f(k), \ell_i(k)\}. \quad (3)$$

1) *Congestion Detection Time*: Starting at time 0, each node has traffic at link rate to transmit to node 0. The initial rate limiter value of all nodes is equal to the link rate: $\ell_i(1) = 1, i = 1, 2, \dots, n$. Recall that at each node the transit traffic has priority over local traffic of that node. As a result, before congestion detection, only node n , which is the most upstream node, can transmit at the link rate, i.e., $r_n^a(k) = 1$; other nodes will not be able to add their local traffic to the ring before congestion detection. Hence, $r_i^a(k) = 0, i = 1, \dots, n - 1$. Note that as node n sends traffic at the link rate to node 0, before congestion detection we have $u_i(k) = 1, i = 1, \dots, n$.

We assume that congestion is triggered at node i only when $\tilde{u}_i(k)$ exceeds τ . As congestion is detected through a low-pass filtering approach, it takes several control intervals for all nodes to detect congestion. For node 1, in particular, we have $\tilde{u}_1(0) = 0$ and $u_1(k) = 1$, for all k prior to congestion detection. Therefore, one can derive $\tilde{u}_1(k)$, using (2), as follows:

$$\tilde{u}_1(k) = 1 - (1 - \alpha)^k. \quad (4)$$

Node 1 detects congestion at the end of control interval \hat{k} , where $\hat{k} = \min \{k : \tilde{u}_1(k) > \tau\}$. Using (4) we obtain

$$\hat{k} = \left\lceil \frac{\ln(1 - \tau)}{\ln(1 - \alpha)} + 1 \right\rceil, \quad (5)$$

where $\lceil x \rceil$ is the largest integer number which is smaller than or equal to x . Note that nodes 2, 3, ..., and n , detect the congestion at the same time as node 1.

2) *Advertised Fair Rate*: Assume that node 1 initiates the control message transmission at the end of each control interval. As the size of the control message in RPR is relatively short, we can ignore its transmission and processing time. We have also assumed that link propagation delay is zero in this ideal model. As a result, the time it takes for a control message to reach the furthest upstream node in the ring (i.e., node n), is zero. This means that all nodes receive and apply the fair rate with no delay after it is advertised.

For $k \leq \hat{k}$, only node n can add traffic to the ring. Therefore, we have

$$\tilde{r}_i^a(\hat{k}) = 0, \quad i = 1, 2, \dots, n-1. \quad (6)$$

When congestion is detected, node 1 advertises its calculated fair rate, i.e., $\tilde{r}_1^a(\hat{k}) = 0$, to its upstream nodes. Node $i > 1$ receives the fair rate instantly and sets its rate limiter value to the received fair rate, i.e., $\ell_i(\hat{k} + 1) = \tilde{r}_1^a(\hat{k}) = 0$. Note that the rate limiter value of a node only changes when that node receives a fair rate from one of its congested downstream nodes. Therefore, $\forall k, \ell_1(k) = 1$, as there is no congested node on downstream of node 1. Hence, from (3) we have

$$\forall k, r_1^a(k) = \min \{1 - r_1^f(k), \ell_1(k) = 1\} = 1 - r_1^f(k). \quad (7)$$

As $\forall i, u_i(k) = r_i^a(k) + r_i^f(k)$, it follows from (7) that $\forall k, u_1(k) = 1$. Therefore, $\tilde{u}_1(k)$ can always be obtained from (4). Note that $\tilde{u}_1(k)$ in (4) is an increasing function of k and consequently, $\tilde{u}_1(k) > \tau, k \geq \hat{k}$. This means that node 1 is congested for all $k \geq \hat{k}$. As a result, node 1 calculates and advertises a fair rate to its upstream nodes in every control interval after congestion detection.

– In control interval $k = \hat{k} + 1$, as $\ell_i(\hat{k} + 1) = 0, i > 1$, it follows from (3) that $r_i^a(\hat{k} + 1) = 0, i > 1$. In other words, node $i > 1$ does not add any traffic to the ring. Consequently, $r_1^f(\hat{k} + 1) = 0$, and using (7), $r_1^a(\hat{k} + 1) = 1$. Therefore, according to (1), the new calculated fair rate of node 1 is equal to $\tilde{r}_1^a(\hat{k} + 1) = (1 - \alpha) \cdot 0 + \alpha \cdot 1 = \alpha$.

Note that node $i > 1$ is not congested in control interval $\hat{k} + 1$ as $u_i(\hat{k} + 1) = 0, i > 1$. Hence, node 1 is the only congested node and all upstream nodes adjust their local rates to the advertised fair rate of node 1. At the end of control interval $\hat{k} + 1$, node 1 advertises $\tilde{r}_1^a(\hat{k} + 1)$ as the fair rate to its upstream nodes.

– In control interval $(\hat{k} + 2)$, we have $\ell_i(\hat{k} + 2) = \tilde{r}_1^a(\hat{k} + 1) = \alpha, i > 1$. This means that each node $i > 1$ can add traffic to the ring up to the rate α . Using (3), it follows that $r_i^a(\hat{k} + 2) = \min \{1 - r_i^f(\hat{k} + 2), \alpha\}, i > 1$. For transit traffic of node 1, which is the aggregated traffic from nodes 2, 3, ..., and n , we have $r_1^f(\hat{k} + 2) = \min \{1, (n-1) \cdot \alpha\}$. Hence, the add-rate of node 1 is $r_1^a(\hat{k} + 2) = 1 - r_1^f(\hat{k} + 2) = \max \{0, 1 - (n-1) \cdot \alpha\}$. Then, node 1 calculates its new fair rate as $\tilde{r}_1^a(\hat{k} + 2) = (1 - \alpha) \cdot \alpha + \alpha \cdot r_1^a(\hat{k} + 2)$, which is advertised to its upstream nodes.

– In general, in control interval $k > \hat{k}, \ell_i(k) = \tilde{r}_1^a(k-1), i > 1, r_1^f(k) = \min \{1, (n-1) \cdot \tilde{r}_1^a(k-1)\}$, and the add-rate of node 1 is obtained as

$$r_1^a(k) = (1 - (n-1) \cdot \tilde{r}_1^a(k-1))^+, k > \hat{k}, \quad (8)$$

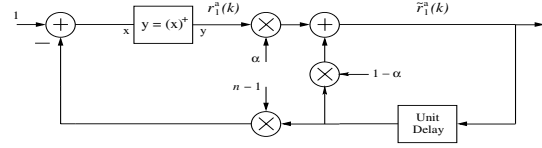


Figure 3. Low-pass filter model of fair rate calculation.

where $(x)^+$ denotes $\max \{0, x\}$. Consequently, the fair rate of node 1 for $k > \hat{k}$ can be calculated as follows:

$$\tilde{r}_1^a(k) = (1 - \alpha) \cdot \tilde{r}_1^a(k-1) + \alpha \cdot (1 - (n-1) \cdot \tilde{r}_1^a(k-1))^+. \quad (9)$$

It can be seen that $\tilde{r}_1^a(k)$ depends on α, n , and the previous calculated fair rate, $\tilde{r}_1^a(k-1)$. Therefore, we can model the fair rate calculation process at node 1 using the first-order non-linear discrete-time filter depicted in Fig. 3, where the unit delay is equal to one control interval. The advertised fair rate of node 1 is the response of the filter to a unit step function applied at $k = \hat{k} + 1$, with initial condition $\tilde{r}_1^a(\hat{k}) = 0$.

3) *Condition for Finite Convergence Time of the Fair Rate:* Consider the filter in Fig. 3 and assume that the non-linear block is bypassed (i.e., $y = (x)^+$ replaced by $y = x$). In this case, the system is a first-order *linear* discrete-time filter with transfer function

$$H(z) = \frac{\alpha}{1 + (\alpha \cdot n - 1) \cdot z^{-1}}. \quad (10)$$

This linear system is stable when its pole is inside the unit circle, which yields

$$\alpha < \frac{2}{n}. \quad (11)$$

We have analytically proved in [6] that the stability condition of the non-linear filter is given by (11) as well. Moreover, we have shown that the response of the non-linear stable system to the unit step function converges to $\frac{1}{n}$, i.e., the true fair rate.

Note that node 1 is assumed to be the most congested node and all other nodes adjust their local rates based on the advertised fair rate of node 1. To maintain this condition, the congestion threshold, τ , should be set to a value greater than $1 - \frac{1}{n}$ and close to 1. In this case, the other nodes are not congested for $k > \hat{k}$, and always set their rates to the advertised fair rate of node 1.

3.2. Analysis of the Model With Link Propagation Delay

We now consider the effect of the link propagation delay on fair rate calculation. Up to the congestion detection, this

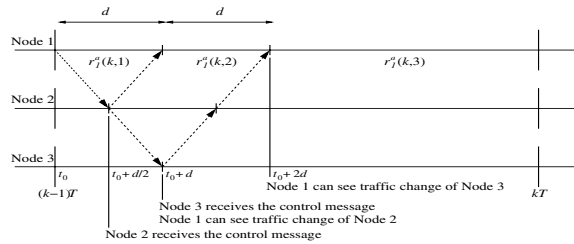


Figure 4. Effect of propagation delay on advertising the fair rate.

system works like the ideal model and the congestion detection time can still be calculated from (5). We define d as the one-hop *round-trip* propagation delay. In this case the propagation delay of each link is equal to $\frac{d}{2}$. Fig. 4 illustrates the effect of link propagation delay on the add-rate of node 1 in a scenario with $n = 3$. It presents a typical control interval k , at which node 1 sends a control message to its upstream nodes.

Assume that node 1 advertises $\tilde{r}_1^a(k-1)$ as the fair rate to the upstream node 2 at $t_0 = (k-1) \cdot T$ seconds. Node 2 receives the control message at $(t_0 + \frac{d}{2})$ seconds. Upon receiving the control message, node 2 sets its local traffic rate controller to the advertised fair rate and forwards the control message to node 3.

At $(t_0 + d)$ seconds, node 1 can see the effect of traffic change of node 2 due to the advertised fair rate $\tilde{r}_1^a(k-1)$. Note that the forward-rate (and consequently the add-rate) of node 1 in time interval $(t_0, t_0 + d]$ does not depend on the advertised fair rate $\tilde{r}_1^a(k-1)$ yet. At the same time, node 3 receives the control message forwarded by node 2. It also sets its rate controller to the advertised fair rate. Node 1 will see the effect of the rate adjustment of node 3 at $(t_0 + 2 \cdot d)$ seconds. In general, node 1 can see the traffic change of node $j > 1$ due to the most recently advertised fair rate at $(t_0 + (j-1) \cdot d)$ seconds. In order to simplify the problem, we assume that the time it takes for the congested node 1 to see the traffic change of node n , which is the furthest node contributing to congestion, is less than the length of control interval. That is,

$$(n-1) \cdot d < T. \quad (12)$$

Therefore, after advertising a new fair rate at the beginning of each control interval, congested node 1 can see the effect of the traffic change of all nodes contributing to congestion in the same control interval. Hence, it can make a solid estimation of the true fair rate. As node 1 gradually sees the effect of traffic changes of its upstream nodes, its add-rate changes during a control interval. In this case, the average add-rate of node 1 during a control interval should be measured in order to calculate the fair rate. In [6] we show that

the average add-rate of node 1 in control interval $k > \hat{k}$ is calculated as

$$r_1^a(k) = (1 - (n-1) \cdot \frac{d}{T}) \cdot r_1^a(k, n) + \frac{d}{T} \cdot \sum_{j=1}^{n-1} r_1^a(k, j), \quad (13)$$

where $r_1^a(k, j) = (1 - (n-j) \cdot \tilde{r}_1^a(k-2) - (j-1) \cdot \tilde{r}_1^a(k-1))^+$, $1 \leq j \leq n$.

The calculated fair rate of node 1 in control interval k , $\tilde{r}_1^a(k)$, is obtained by substituting (13) in (1). One can see that $\tilde{r}_1^a(k)$, $k > \hat{k}$, depends on α , n , $\tilde{r}_1^a(k-2)$, and $\tilde{r}_1^a(k-1)$. As a result, using a method similar to the one applied to the ideal case ($d = 0$), we can model the fair rate calculation at node 1 by a second-order non-linear discrete-time filter.

We have investigated and proved several properties of this model in [6]. We observed that the stability domain of the second-order nonlinear filter is similar to the corresponding linear filter, which is obtained by bypassing all non-linear blocks in (13). Also, we proved that the linear system with propagation delay is less sensitive to the value of α compared to the linear system without propagation delay. Finally, we showed that the maximum stability domain for the linear system with delay is achieved when $(n-1) \cdot d = 0.5 \cdot T$.

4. Numerical Results and Analysis

In this section, we compare the results from the analysis with the ones from simulations to examine the accuracy of the developed model. We also analyze the convergence of the calculated fair rate in parking lot scenario. The simulation model is implemented in OPNET. We consider 100Mbps links, packet length of 512 bits, and $T = 1\text{msec}$.

1) *Effect of α on Fair Rate Oscillations:* The oscillations of the fair rate result from an overestimation or underestimation of the rates. As the oscillations contribute to the convergence time, they are part of the performance of the fairness algorithm. Fig. 5 presents both analytical and simulation results of the calculated fair rate in a scenario with $n = 8$, $d = 0.1T$, $\tau = 0.95$, for $\alpha = 0.05, 0.15$, and 0.25 . One can first note that analytical and simulation results are very close. In fact, the maximum relative error in all cases is less than 2%. Our analytical model is a very accurate approximation of the real system, and therefore may be used to evaluate the performance of the RPR fairness algorithm. In Fig. 5, congestion is detected at control intervals 59, 19, and 11 for $\alpha = 0.05, 0.15$, and 0.25 , respectively. One can see that as α increases, congestion detection time decreases. However, the calculated fair rate converges to the true fair rate with a larger number of oscillations. The reason is that when α is large, the latest measured data has more weight in the calculation of the next fair rate, according to (1), and

the true fair rate may not be estimated accurately. Thus, the fair rate may be overestimated, then underestimated, and so on.

2) *Effect of Propagation Delay on Fair Rate Oscillations*: Fig. 6 illustrates the effect of propagation delay on the fair rate calculation for $n = 4$ and $\alpha = 0.5$. Both analytical and simulation results are shown. In this case, $\hat{k} = 5$. As delay increases, the congested node sees the effect of traffic changes due to the last advertised rate in a smaller portion of the control interval, which changes the value of the calculated rate. For example for $d = 0.2T$, the first overshoot of the fair rate occurs with one control interval delay comparing with the other two cases due to the propagation delay. The maximum relative error of the simulation results in all cases is less than 1.5%.

3) *Effect of Propagation Delay on Stability of Fair Rate*: We define $\alpha_n^*(d)$ as the supremum value of α such that the calculated fair rate does converge to $\frac{1}{n}$ in an n -node scenario with d one-hop round-trip propagation delay. In other words, for $\alpha > \alpha_n^*(d)$, the advertised fair rate has permanent oscillations. In the ideal model ($d = 0$), (i) the calculated fair rate in control interval k , depends on the calculated fair rate in control interval $k - 1$ according to (9); and (ii) $\alpha_n^*(d = 0) = \frac{2}{n}$ (See (11)). In the realistic model (for $d > 0$), the calculated fair rate in control interval k depends on the fair rate calculated in the control intervals $k - 1$ and $k - 2$ (See (13)). As it mentioned in Section 3.2, we observed that the system with delay is less sensitive to the value of α compared to the system without propagation delay. Therefore, $\alpha_n^*(d > 0) > \alpha_n^*(d = 0)$. Fig. 7 shows $\alpha_n^*(d)$ versus normalized propagation delay ($\frac{d}{T}$) for different number of nodes, n . It can be seen in Fig. 7 that $\alpha_n^*(d > 0)$ is greater than $\alpha_n^*(d = 0)$. When $d \rightarrow 0$, the model behaves similar to the ideal model ($d = 0$) and $\alpha_n^*(d) \rightarrow \frac{2}{n}$. In [6], we have proved that for some cases a system with delay exhibits a unique maximum value of $\alpha_n^*(d)$ for $\frac{d}{T} = \frac{1}{2(n-1)}$. This can be seen in Fig. 7 where

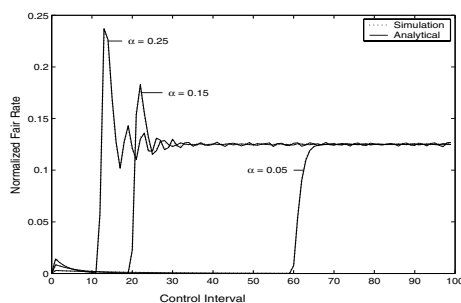


Figure 5. Normalized calculated fair rate for $n = 8$, $\tau = 0.95$, and $d = 0.1T$.

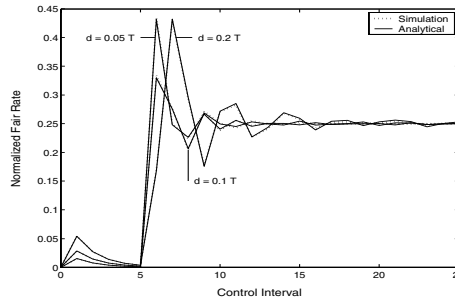


Figure 6. Normalized calculated fair rate for $n = 4$, $\tau = 0.95$, and $\alpha = 0.5$.

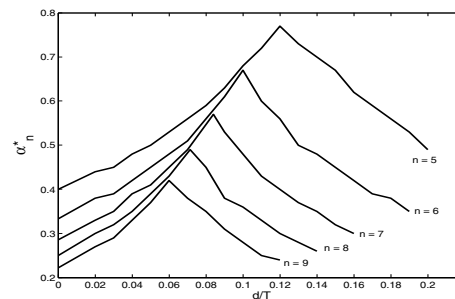


Figure 7. Variation of α_n^* due to the propagation delay changes.

for a given n , the maximum value of $\alpha_n^*(d)$ is achieved approximately for $\frac{d}{T} = \frac{1}{2(n-1)}$.

4) *Effect of α on Fair Rate Convergence Time*: We define $k_{c,n}$ as the convergence time of the system with n nodes, i.e., $k_{c,n} = \min \{k : |r_i^a(k) - \frac{1}{n}| \leq \varepsilon, i = 1, 2, \dots, n\}$, for a given ε , $0 < \varepsilon < 1$. Fig. 8 shows the convergence time for different number of nodes versus α for $d = 0.1T$. Convergence time consists of two components: (i) congestion detection time; and (ii) search-time defined as the time between congestion detection and convergence of the fair rate, when the nodes search for the true fair rate. Congestion detection time is independent of the number of nodes contributing to the congestion, and is a decreasing function of α for a fixed congestion threshold τ , according to (5). As α increases, congestion detection time decreases but on the other hand the calculated fair rate experiences more oscillations and search-time becomes larger.

It is noticeable in Fig. 8 that for a constant value of n and small values of α (close to 0), convergence time is a decreasing function of α . The reason is that detection time (which is a decreasing function of α) is the dominant part in total convergence time, when α is close to 0.

As we increase α , detection time drops while search-

time rises. Therefore, convergence time does not change significantly and remains in the same range in some extent. For example, it can be seen in Fig. 8 that convergence time for $n = 4$ does not change significantly for α such that $0.2 < \alpha < 0.6$. Nonetheless when α gets closer to $\alpha_n^*(d = 0.1T)$ (which is for example around 0.7 for $n = 4$ in Fig. 8), the search-time is the main source of delay in convergence time and its drastic increase.

We also observe that when α is relatively small (e.g., $\alpha < 0.05$), if $n_1 > n_2$, we have $k_{c,n_1} < k_{c,n_2}$. The reason is that for small values of α , the calculated fair rate does not oscillate and instead increasingly converges to the true fair rate (this is shown in Fig. 5 for $\alpha = 0.05$). For the true fair rates we have $\frac{1}{n_1} < \frac{1}{n_2}$, hence $k_{c,n_1} < k_{c,n_2}$.

It can also be seen that as α increases and estimated fair rate oscillates, the system with more nodes faces more delay in converging to the true fair rate. The reason is that for $n_1 > n_2$, we have $\alpha_{n_1}^*(d) < \alpha_{n_2}^*(d)$ and if α is within stable domain of both systems, it is closer to $\alpha_{n_1}^*(d)$ than $\alpha_{n_2}^*(d)$ and we can conclude that convergence time of the system with more node is larger. Also, any change in propagation delay can increase or decrease $\alpha_n^*(d)$ according to Fig. 7. For example, we observe in Fig. 7 that, by decreasing the link propagation delay from $d = 0.1T$ to $d = 0.02T$, $\alpha_n^*(d)$ decreases for $n = 6, 7$, and 8. Fig. 9 depicts the fair rate convergence time for different number of nodes for $d = 0.02T$. By comparing graphs in Figs. 8 and 9, it can be seen that $\alpha_n^*(0.02T) < \alpha_n^*(0.1T)$. From results of Figs. 8 and 9, we heuristically observe that for a system with propagation delay, the convergence time of the fair rate is relatively small for α such that $\frac{1}{n} < \alpha < \frac{2}{n}$.

5. Conclusion

We develop an analytical model for fair rate calculation in the standard RPR fairness algorithm in parking lot scenario. Our approach is to model the fair rate calculation

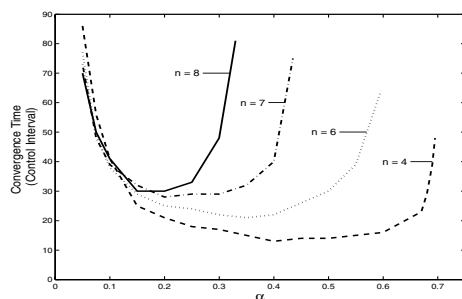


Figure 8. Convergence time vs. α with $\varepsilon = 0.01$ and $d = 0.1T$.

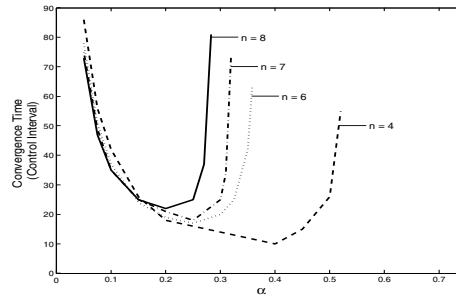


Figure 9. Convergence time vs. α with $\varepsilon = 0.01$ and $d = 0.02T$.

process as a nonlinear discrete-time low-pass filter. Simulation results confirm the model accuracy. This model can be used to evaluate performance of the RPR fairness algorithm in terms of stability and convergence time. Using this model, we studied the effect of various system parameters on stability and convergence of the calculated fair rate. Finally, we concluded that by choosing the low-pass filter coefficient α so that $\frac{1}{n} < \alpha < \frac{2}{n}$, the convergence time of the fair rate in parking lot scenario will be within its minimum range.

References

- [1] I. Cidon, L. Georgiadis, R. Guerin, and Y. Shavitt. Improved fairness algorithms for rings with spatial reuse. *IEEE/ACM Transactions on Networking*, 5(2):190–204, April 1997.
- [2] F. Davik, M. Yilmaz, S. Gjessing, and N. Uzun. IEEE 802.17 resilient packet ring tutorial. *IEEE Communication Magazine*, 42(3):112–118, March 2004.
- [3] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafar, and E. Knightly. Design, analysis, and implementation of DVSR: A fair, high performance protocol for packet rings. *IEEE/ACM Transactions on Networking*, 12(1):85–102, February 2004.
- [4] IEEE. IEEE standard 802.17: Resilient packet ring (draft version 3.1). <http://ieee802.org/17>, February 2004.
- [5] M. Karol and R. Gitlin. High performance optical, local and metropolitan area networks: Enhancements of FDDI and IEEE 802.6 DQDB. *IEEE JSAC*, 8(8):1439–1449, October 1990.
- [6] A. Shokrani, I. Lambadaris, and J. Talim. An analytical model for fair rate calculation in resilient packet rings. *Carleton University Technical Report*, (SCE-04-16), August 2004.
- [7] R. Simha and Y. Ofek. A starvation-free access protocol for full-duplex buffer insertion ring local area networks. *Journal of Computer Networks and ISDN Systems*, 21, 1991.