Modeling and Stability Analysis 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— **Resilient Packet Ring (RPR), which is standardized as IEEE 802.17, is a new Medium Access Control (MAC) protocol for high-speed metro-area ring networks. RPR supports spatial reuse and, therefore, maintaining fairness among different nodes in accessing the ring bandwidth is a challenging task in RPR. In order to achieve fairness among nodes, a fairness algorithm is employed at each RPR node. When a node is not given access to the ring it becomes congested. In this case, the fairness algorithm calculates and advertises a fair rate to all upstream nodes contributing to the congestion point. Consequently, the congested node will be able to add its local traffic to the ring. In this paper, we develop an analytical model for fair rate calculation in the standard RPR fairness algorithm in the parking lot scenario. We first ignore the link propagation delay and model the system using a non-linear discrete-time low-pass filter. We, then, consider the link propagation delay and develop a more realistic model. We verify our model by simulation results and analyze the convergence of the fairness algorithm. Further, the effect of various parameters on the convergence time is investigated. Finally, we determine the low-pass filter coefficient in order to ensure that the convergence time of the algorithm in the parking lot scenario is within its minimum range.**

I. 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 several objectives including high utilization, fault recovery, and fairness, which are not simultaneously available in the current technologies [1], [2], [3]. 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 [4], [5]. Although the spatial reuse improves throughput of the ring, it may cause congestion and unfairness among different nodes in accessing the ring bandwidth. Therefore, a bandwidth allocation algorithm is required in order to provide a fair ring access to all nodes.

The ring access scheme in RPR is based on the Buffer Insertion Ring (BIR) method, in which every node (station) on the ring has an insertion (transit) buffer [6]. 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, the ring traffic has non-preemptive priority over the add traffic. This means that the transit traffic at each node can block the local traffic of that node in accessing the ring.

To illustrate the fairness problem in RPR, consider the *parking lot* scenario (Fig. 1) in which all nodes 1, 2, …, and n have infinite local traffic to send to node 0 (Hub) sharing the link between node 1 and node 0. To ensure fairness, each node should receive $\frac{1}{n}$ of the link bandwidth in this example. Hence, all nodes should be forced to adjust the insertion rate of their local traffic (add-rate) according to their fair shares. Without such a scheme, node n, as the most upstream node in this scenario, can take over all of the bandwidth and starve its downstream nodes.

The objective of the fairness algorithm in RPR is to distribute fairly the available bandwidth on any link among 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 becomes 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. Notice that as the calculated fair rate is an approximation to the true fair rate, the congestion may not be cleared at the congested node. In this case, the congested node calculates another approximation of the fair rate and advertises it to the upstream nodes. This process may repeat until the calculated fair rate converges to the true fair rate. The time needed for the fairness algorithm to converge is commonly referred to as *convergence time*, which is an important criterion in evaluating performance and efficiency of an RPR fairness algorithm.

In [3] and [7], 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. 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

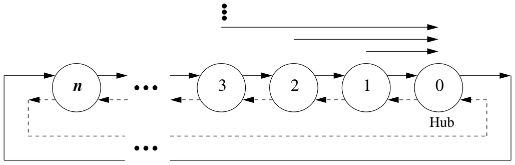

Fig. 1. Parking lot scenario

0-7803-9277-9/05/$20.00/©2005 IEEE

(Fig. 1). This scenario is one of the traditional and challenging benchmark scenarios in RPR. It is studied in standardization process of the RPR as well as in [7], to compare the performance of different RPR fairness algorithms. Using this 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.

The rest of this paper is organized as follows. In Section II the RPR node architecture, the notion of fairness in RPR, and the RPR standard fairness algorithm are described. Section III presents our analytical model for fair rate calculation. Section IV deals with the numerical results. Finally, Section V concludes this paper.

II. An Overview of Resilient Packet Ring

The RPR MAC supports three service classes: high priority class A, medium priority class B, and low priority class C. Class A provides an allocated guaranteed data rate with low end-to-end delay and jitter bound. Class B is divided into two subclasses; B-CIR (committed information rate) and B-EIR (excess information rate). The bandwidth is allocated for B-CIR subclass which provides guaranteed data rate with bounded end-to-end delay and jitter. Any class B traffic beyond allocated B-CIR is considered as B-EIR subclass, which has access to additional best effort data transmission. Class C provides a best-effort traffic service with no allocated or guaranteed data rate and no bounded end-to-end delay or jitter. It is necessary to mention that only traffic subclass B-EIR and class C are subject to the fairness algorithm and called fairness eligible (FE) traffic. Class A and subclass B-CIR have allocated bandwidth and are not considered for fairness determination in RPR.

A. RPR Node Architecture

RPR employs 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 removed 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 the single-queue mode, all transiting packets with different priorities are forwarded into a single queue called *primary transit queue* (PTQ). In this mode, the scheduler gives service priority to the transit traffic over the local traffic which guarantees a lossless ring. In the dual-queue mode, there are two transit queues: *primary transit queue* (PTQ) for class A traffic and *secondary transit queue* (STQ) for class B, and class C traffic. In this mode, the scheduler gives highest priority to the PTQ. Then, packets from local buffers are forwarded prior to packets from the STQ. When the STQ is almost full, the scheduler gives service priority to the transit queues over the local buffers. Employing the dual-queue transit buffer improves the delay performance of the class A traffic [8].

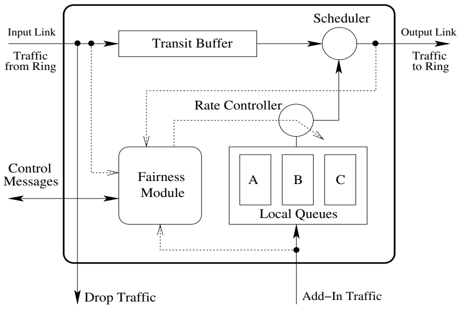

Fig. 2. RPR node architecture

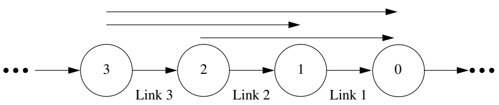

Fig. 3. An Example for the RPR fairness model

At each node, the MAC client (local) traffic is buffered in the local queues. This traffic is queued based on different priorities. The rate of the local traffic at each node is controlled by rate controllers. The fairness module implements the fairness algorithm and adjusts the rate controllers of the local FE traffic. It measures the rate of the local add traffic and that of the transit traffic and uses these measurements to calculate the fair rate. In case of congestion, the fairness module sends a control message which include the fair rate, to the upstream nodes contributing to the congestion. The control message is transmitted on the opposite direction of the traffic flow through the opposite ringlets.

B. RPR Fairness Model

The objective of the fairness algorithm in RPR is to distribute fairly the available bandwidth on a given link among all competing nodes on that link [9]. In other words, a per-station fairness in utilizing the ring bandwidth should be maintained in RPR. The bandwidth share of each node on a link should be proportional to the relative weight assigned to that node. To illustrate the fairness model consider the scenario depicted in Fig. 3. In this scenario, node 3 has local FE traffic destined to nodes 1 and 0 and node 2 sends FE traffic only to node 0. Assume that all demands are infinite and all nodes have equal weights. Therefore, the available bandwidth on link 2 should be equally shared between FE traffic of nodes 2 and 3. As a result, each of the flows originating from node 3 receives $1/4$ and the flow originating from node 2 receives $1/\,2$ of the available bandwidth on link 2.

C. RPR Standard Fairness Algorithm

Congestion detection and fair rate calculation in the 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. Furthermore, all measured rates are smoothed by low-pass filtering. Let us define the following notations:

- C is the unreserved link capacity called *unreserved* rate.
- T is the length of a control interval in seconds and control interval $k > 0$ is the time interval $\big((k-1)\cdot T, k\cdot T\big]$.
- 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 version defined as
$$\tilde{r}_i^f(k) = (1-\alpha)\cdot\tilde{r}_i^f(k-1) + \alpha\cdot r_i^f(k). \quad (2)$$
- $u_i(k) = r_i^a(k) + r_i^f(k)$ 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
$$\tilde{u}_i(k) = \tilde{r}_i^a(k) + \tilde{r}_i^f(k) = (1-\alpha)\cdot\tilde{u}_i(k-1) + \alpha\cdot u_i(k). \quad (3)$$

For simplicity, assume that all rates are normalized to the unreserved rate. 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 the 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 the local buffer of that node experiences a delay more than a predefined delay threshold to access the ring.

2) Fair Rate Calculation: The RPR standard fairness algorithm operates in two modes: *Aggressive Mode* (AM) and *Conservative Mode* (CM) [9]. The differences between these two modes are in the fair rate calculation and rate adjustment in case of congestion. In this paper, 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 sets its advertised fair rate to $\tilde{r}_i^a(k)$. This fair rate is advertised to the upstream nodes of node i. If node i is not congested, the fair rate is set to 1, i.e. normalized unreserved rate.

3) Control Message Transmission: We assume that there is a single control message which circulates along the ring within each control interval. It 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. This node inserts its advertised fair rate in a designated field in the control message and sends the control message to its upstream node. Upon receiving the control message, node i:

– adjusts the insertion rate of its local FE traffic so that its add-rate for destinations beyond node j, does not exceed the advertised fair rate of node j;

– updates the corresponding field in the control message with its most recently calculated fair rate and forwards this message to its upstream node.

The next section presents an analytical model for fair rate calculation in RPR.

III. Problem Description

We consider the parking lot scenario (Fig. 1) which is a benchmark scenario for evaluating an RPR fairness algorithm. The performance of an RPR fairness algorithm should be evaluated in an overloaded network; otherwise, congestion does not occur and there is no need for a rate control mechanism. Hence, we assume that in Fig. 1 nodes 1, 2, . . ., and n have infinite amount of traffic in their local buffers destined to node 0 (hub). 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 becomes the most congested one. 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 congestion is triggered [9]. In this analysis, we assume that the transit queue at each node is implemented in the single queue mode. 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.

In the following sections, this problem is analyzed for two cases. First, we assume that the propagation delay of the links is zero and investigate the convergence of the fairness algorithm. Then, we incorporate the deterministic link propagation delay into the model and analyze the problem in a more realistic case.

A. Analysis of the Model without Link Propagation Delay

In this section we ignore the link propagation delay and model the fair rate calculation in the RPR standard fairness algorithm. Then, we determine the conditions with respect to which the convergence time of the fairness algorithm is finite. Let us define $\ell_i(k)$ as 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
$$r_i^a(k) = \min\big\{1 - r_i^f(k), \ell_i(k)\big\}. \quad (4)$$

1) Congestion Detection Time: Starting at time 0 (or during control interval $k = 1$), each node has traffic at the link rate to transmit to node 0. The initial rate limiter value of all nodes is equal to the link rate: $\ell_i(1) = 1$, $i = 1,2,\ldots,n$. Note that at each node the transit traffic has priority over the local traffic of that node. As a result, 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 the congestion detection. Hence, $r_i^a(k) = 0$, $i = 1,\ldots,n-1$. Note that as node n sends traffic at the link rate to node 0, before the congestion detection we have $u_i(k) = 1$, $i = 1,\cdots,n$.

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 the congestion detection. Therefore, one can derive $\tilde{u}_1(k)$ using (3) as follows:

$$\tilde{u}_1(k) = 1 - (1-\alpha)^k. \tag{5}$$

Node 1 detects congestion at the end of control interval $\hat{k}$, where $\hat{k} = \min\left\{k : \tilde{u}_1(\hat{k}) > \tau\right\}$. Using (5) we obtain

$$\hat{k} = \left\lfloor \frac{\ln(1-\tau)}{\ln(1-\alpha)} + 1 \right\rfloor, \tag{6}$$

where $\lfloor x \rfloor$ is the largest integer number which is equal to or less than 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 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 the transmission and the processing time of the control message throughout the ring. Moreover, we have assumed that the link propagation delay is zero in this ideal model. As a result, the time needed for a control message to reach the furthest upstream node in the ring (i.e. node n) is zero. This means that all upstream nodes of node 1 receive the control message right after it is transmitted. For $k \leq \hat{k}$, only node n can add traffic to the ring. Therefore, we have

$$\tilde{r}_i^a(\hat{k}) = 0, \quad i = 1,2,\ldots,n-1. \tag{7}$$

When congestion is detected, node 1 advertises its calculated fair rate, i.e. $\tilde{r}_1^a(\hat{k}) = 0$, to its upstream nodes. When node $i > 1$ receives the control message, it sets its rate controller value to the received fair rate, i.e. $\ell_i(\hat{k}+1) = \tilde{r}_1^a(\hat{k}) = 0$. Note that the rate controller value of a node only changes when that node receives a fair rate from one of its downstream congested nodes. Therefore, $\forall k$, $\ell_1(k) = 1$, as there is not a congested node on downstream of node 1. Consequently, from (4) we have

$$\forall k,\ r_1^a(k) = \min\left\{1 - r_1^f(k), \ell_1(k) = 1\right\} = 1 - r_1^f(k). \tag{8}$$

As $\forall i$, $u_i(k) = r_i^a(k) + r_i^f(k)$, it follows from (8) that $\forall k$, $u_1(k) = 1$. Therefore, $\tilde{u}_1(k)$ can always be obtained from (5). Note that $\tilde{u}_1(k)$ in (5) is an increasing function of k and, consequently, $\tilde{u}_1(k) > \tau$, $\forall k \geq \hat{k}$. 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 the upstream nodes in every control interval after the congestion detection.

- In control interval $k = \hat{k}+1$, as $\ell_i(\hat{k}+1) = 0$, $i > 1$, it follows from (4) 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 (8) $r_1^a(\hat{k}+1) = 1$. Therefore, according to (1) the new calculated fair rate of node 1 is equal to $\tilde{r}_1^a(\hat{k}+1) = (1-\alpha)\cdot 0 + \alpha \cdot 1 = \alpha$.

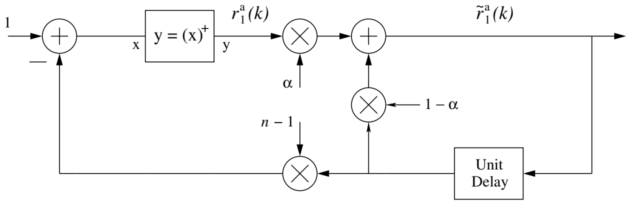

Fig. 4. Low-pass filter model of fair rate calculation.

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 (4), it follows that $r_i^a(\hat{k}+2) = \min\left\{1 - r_i^f(\hat{k}+2), \alpha\right\}$, $i > 1$. For the transit traffic of node 1, which is the aggregated traffic from nodes 2,3,…,n, we can write $r_1^f(\hat{k}+2) = \min\left\{1, (n-1)\cdot\alpha\right\}$. Hence, the add-rate of node 1 is $r_1^a(\hat{k}+2) = 1 - r_1^f(\hat{k}+2) = \max\left\{0, 1-(n-1)\cdot\alpha\right\}$. 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 $k > \hat{k}$, $\ell_i(k) = \tilde{r}_1^a(k-1)$, $i > 1$, $r_1^f(k) = \min\left\{1, (n-1)\cdot\tilde{r}_1^a(k-1)\right\}$, and the add-rate of node 1 is obtained as

$$r_1^a(k) = \left(1 - (n-1)\cdot\tilde{r}_1^a(k-1)\right)^+, k > \hat{k}, \tag{9}$$

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

$$\tilde{r}_1^a(k) = (1-\alpha)\cdot\tilde{r}_1^a(k-1) + \alpha\cdot\left(1-(n-1)\cdot\tilde{r}_1^a(k-1)\right)^+. \tag{10}$$

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 nonlinear discrete-time filter depicted in Fig. 4, where the unit delay is equal to one control interval. The advertised fair rate of node 1 is the response of this 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. 4 and assume that its 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

$$H(z) = \frac{\alpha}{1 + (\alpha\cdot n - 1)\cdot z^{-1}}\ . \tag{11}$$

We refer to the obtained system as the *corresponding linear system* of the nonlinear model. This corresponding linear system is stable when its pole is inside the unit circle, which yields

$$\alpha < \frac{2}{n}. \tag{12}$$

We have proved in Proposition 1 in Appendix that the stability condition of the nonlinear system depicted in Fig. 4 is given by (12) as well. That is, if $\alpha < \frac{2}{n}$, the convergence time of the calculated fair rate is finite.

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 local rates to the advertised fair rate of node 1.

B. 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 system works like the ideal model and the congestion detection time can still be calculated from (6). We define d as the one-hop *round-trip* propagation delay. In other words, the propagation delay of each link is equal to $\frac{d}{2}$. Fig. 5 illustrates the effect of the 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. It also shows the time when the upstream nodes receive the control message.

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 $\big(t_0, t_0 + d\big]$ does not depend on the advertised fair rate $\tilde{r}_1^a(k-1)$ yet.

At the same time ($t_0 + d$ seconds) , node 3 receives the control message forwarded by node 2. This node also sets its rate controller to the advertised fair rate. Node 1 observes 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 $\big(t_0 + (j-1) \cdot d\big)$ seconds. In order to simplify the problem, we assume that the time needed 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,

$$(n-1) \cdot d < T. \tag{13}$$

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. As a result, it can make a more solid estimation of the true fair rate. Note that because of the link propagation delay, node 1 gradually sees the effect of the traffic change of its upstream nodes. Therefore, 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. Let us define the following notations:

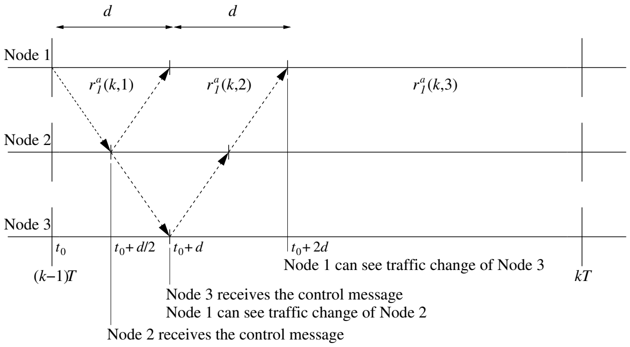
d
d
Node 1
Node 2
Node 3
$r_1^a(k,1)$
$r_1^a(k,2)$
$r_1^a(k,3)$
t_0
$(k-1)T$
$t_0+d/2$
t_0+d
t_0+2d
Node 1 can see traffic change of Node 3
kT
Node 3 receives the control message
Node 1 can see traffic change of Node 2
Node 2 receives the control message

Fig. 5. Effect of propagation delay on advertising the fair rate.

- Subinterval j $(1 \leq j \leq n-1)$ of control interval $k > \hat{k}$ is the time interval $\big((k-1) \cdot T + (j-1) \cdot d, (k-1) \cdot T + j \cdot d\big]$. At the beginning of subinterval j of control interval k, node 1 can see the change of the add-rate of node j due to the fair rate advertised at the beginning of control interval k.
- $r_1^a(k,j)$ is the add-rate of node 1 in subinterval j of control interval k. In other words, $r_1^a(k,j)$ corresponds to the rate adjustment of node j due to the fair rate advertised at the beginning of control interval k.
- $r_1^a(k,n)$ is the add-rate of node 1 in time interval $\big((k-1) \cdot T + (n-1) \cdot d, k \cdot T\big]$. This represents the add-rate of node 1 when the effect of rate adjustment of all upstream nodes can be seen by node 1.

In Fig. 5, the subintervals of control interval k with the corresponding add-rates of node 1 are shown for $n = 3$. Now consider node 1 at the beginning of control interval $k > \hat{k}$, when fair rate $\tilde{r}_1^a(k-1)$ is just advertised to the upstream nodes. We can make the following observations on the add-rate of node 1 during control interval k:

– During subinterval $j = 1$, node 1 cannot see the effect of the latest advertised rate, $\tilde{r}_1^a(k-1)$. The traffic in this interval corresponds to the previous advertised rate, $\tilde{r}_1^a(k-2)$. Using (9), the add-rate of node 1 in this interval is $r_1^a(k,1) = \big(1 - (n-1) \cdot \tilde{r}_1^a(k-2)\big)^+$.

– In subinterval $j = 2$, traffic change of node 2 can be seen as d seconds has elapsed since the latest rate advertisement. However, the add-rates of nodes 3, 4, …, and n still correspond to the previous advertised fair rate. In this case, $r_1^a(k,2) = \big(1 - (n-2) \cdot \tilde{r}_1^a(k-2) - \tilde{r}_1^a(k-1)\big)^+$.

– In general for subinterval j $(1 \leq j \leq n-1)$ we have

$$r_1^a(k,j) = \big(1 - (n-j) \cdot \tilde{r}_1^a(k-2) - (j-1) \cdot \tilde{r}_1^a(k-1)\big)^+. \tag{14}$$

– During time interval $\big((k-1)T + (n-1)d, kT\big]$, the forward-rate of node 1 only corresponds to the latest advertised rate. Hence, the add-rate of node 1 in this time interval is obtained as,

$$r_1^a(k,n) = \big(1 - (n-1)\cdot \tilde{r}_1^a(k-1)\big)^+. \qquad (15)$$

Now we can calculate the add-rate of node 1 in control interval $k > \hat{k}$ by obtaining time-average of its add-rate over this control interval as follows:

$$r_1^a(k) = \big(1 - (n-1)\cdot \frac{d}{T}\big)\cdot r_1^a(k,n) + \frac{d}{T}\cdot \sum_{j=1}^{n-1} r_1^a(k,j), \quad (16)$$

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

Note that calculation of the add-rate of node 1 in (16) is a recursive function as its value in control interval k depends on $\tilde{r}_1^a(k-1)$ and $\tilde{r}_1^a(k-2)$. The calculated fair rate of node 1 in control interval k, $\tilde{r}_1^a(k)$, is obtained by substituting (16) 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 $\tilde{r}_1^a(k-1)$. Therefore, using similar method used for the ideal case ($d = 0$), we can model the fair rate calculation at node 1 by a second-order nonlinear discrete-time filter. The calculated fair rate of node 1 will be the response of the second-order nonlinear discrete-time filter to the unit step function applied at $k = \hat{k} + 1$.

In order to calculate the fair rate of node 1 in control interval $k > \hat{k}$, we need to know the initial values of the fair rate, i.e. $r_1^a(\hat{k}-1)$, and $r_1^a(\hat{k})$. We define $\tilde{r}_1^a(\hat{k}-1) \triangleq 1$ and calculate the exact value of the first advertised fair rate, $r_1^a(\hat{k})$. During control intervals 2, 3, ..., and $\hat{k}$ node 1 could not send any traffic. Hence,

$$r_1^a(k) = 0, \quad k = 2,3,\ldots,\hat{k}. \qquad (17)$$

However, in the first control interval, node 1 can send traffic at the link rate for a short period of time before traffic of node 2 reaches its transit buffer. This time is equal to the one-hop link propagation delay, i.e. $\frac{d}{2}$. Similar to node 1, nodes 2, 3, ..., and $n-1$ can only send traffic at the link rate for $\frac{d}{2}$ seconds in the first control interval. After that, only node n sends traffic until congestion is detected. The average add-rate of node 1 in the first control interval is

$$r_1^a(1) = \frac{d}{2T} \qquad (18)$$

Using (1), (17) and (18) we can calculate the fair rate at the time of congestion detection as follows:

$$\tilde{r}_1^a(\hat{k}) = \alpha \cdot (1-\alpha)^{\hat{k}-1}\cdot \frac{d}{2T} \qquad (19)$$

Now we can calculate the value of the fair rate, $\tilde{r}_1^a(k)$, for any $k > \hat{k}$ using (1), (16), (19), and $\tilde{r}_1^a(\hat{k}-1) \triangleq 1$.

We implemented this analytical model in MATLAB to analyze the effect of the propagation delay and the low-pass filter coefficient on the calculated fair rate. We observed similar properties in the model with the propagation delay and in its corresponding linear system, which is obtained by bypassing all non-linear blocks in (16). For instance, it seems that the stability domain of the second-order nonlinear system (with respect to α for a given n) is similar to its corresponding linear system. In Appendix, we have proved that the corresponding linear system of the model with the propagation delay is stable for a wider range of α compared to the corresponding linear system of the model without the propagation delay (Proposition 2). We have also shown that the maximum range of α with respect to which the corresponding linear system of the model with the propagation delay remains stable, is achieved when $(n-1)\cdot d = 0.5\cdot T$ (Proposition 3).

In the analysis of the model with the link propagation delay, we assumed that (13) is maintained. However, if $(n-1)\cdot d > T$, the calculated fair rate in control interval k $\big(\tilde{r}_1^a(k)\big)$ would depend on $\tilde{r}_1^a(k-1)$, $\tilde{r}_1^a(k-2)$, $\tilde{r}_1^a(k-3)$, and etc., which makes the calculation more complex.

IV. Numerical Results and Analysis

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

A. Effect of α on Fair Rate Oscillations

The oscillations of the fair rate (before converging to the true 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 RPR fairness algorithm. Fig. 6 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. 6, 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, the 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.

B. Effect of Propagation Delay on Fair Rate Oscillations

Fig. 7 illustrates the effect of the propagation delay on the fair rate calculation for $n = 4$ and $\alpha = 0.5$. Both analytical and simulation results are shown. In this case, congestion is detected at the end of the 5^{th} control interval ($\hat{k} = 5$). As the propagation 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

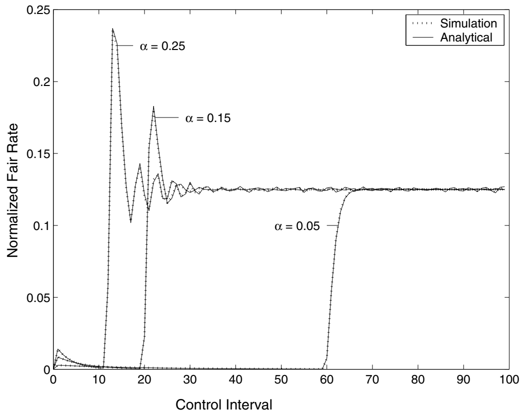

Fig. 6. Normalized calculated fair rate for $n = 8$, $\tau = 0.95$, and $d = 0.1T$.

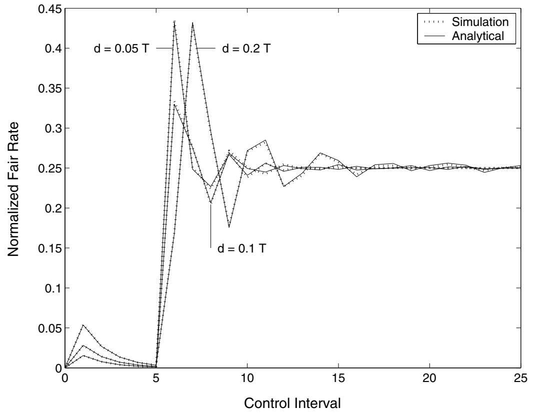

Fig. 7. Normalized calculated fair rate for $n = 4$, $\tau = 0.95$, and $\alpha = 0.5$.

of the calculated rate. For example for $d = 0.2 \cdot T$, 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%.

C. Effect of Propagation Delay on Stability of Fair Rate

We define $\alpha_n^*(d)$ as the supremum value of α such that $\forall \alpha < \alpha_n^*(d)$, the calculated fair rate does converge to $\frac{1}{n}$ in an n-node scenario with d one-hop round-trip propagation delay. 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 (10); and (ii) $\alpha_n^*(d = 0) = \frac{2}{n}$ (See (12)). In the realistic model ($d > 0$), the calculated fair rate in control interval k depends on the fair rate calculated in the control intervals $k - 1$ and $k - 2$ (See (16)). As it was mentioned in Section III-B, we observed through analytical results that the system with the propagation delay is stable for a wider range of α compared to the system without the propagation delay. Therefore, $\alpha_n^*(d > 0) > \alpha_n^*(d = 0)$. In Proposition 2 in Appendix, we have proved this property of our model for a particular case. Fig. 8 shows $\alpha_n^*(d)$ versus the normalized propagation delay ($\frac{d}{T}$) for different number of nodes. These results are obtained using the analytical model. It can be seen in Fig. 8 that $\alpha_n^*(d > 0)$ is greater than $\alpha_n^*(d = 0)$. As $d \to 0$, the model behaves similar to the ideal model ($d = 0$) and $\alpha_n^*(d) \to \frac{2}{n}$. In Appendix, we have proved that the corresponding linear system of the model with the propagation delay, which is obtained by bypassing all nonlinear blocks in (16), exhibits a unique maximum value of $\alpha_n^*(d)$ for $\frac{d}{T} = \frac{1}{2 \cdot (n-1)}$. This property can also be observed for the actual model (which includes the nonlinear blocks) in Fig. 8, where for a given n, the maximum value of $\alpha_n^*(d)$ is achieved approximately for $\frac{d}{T} = \frac{1}{2 \cdot (n-1)}$ (or equivalently $(n - 1) \cdot d = 0.5 \cdot T$).

D. Effect of α on Fair Rate Convergence Time

We define $k_{c,n}$ as convergence time of the system with n nodes, i.e. $k_{c,n} = \min\left\{k : \left|r_i^a(k) - \frac{1}{n}\right| \le \frac{\varepsilon}{n}, i = 1, 2, \ldots, n\right\}$, for a given ε ($0 < \varepsilon < 1$). Fig. 9 shows the analytical results of the convergence time for different number of nodes versus α for $d = 0.1T$. The convergence time consists of two components: (i) congestion detection time; and (ii) *search-time* which is defined as the time between the congestion detection and convergence of the fair rate, when nodes search for the true fair rate. The 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 (6). As α increases, the congestion detection time decreases but on the other hand the calculated fair rate experiences more oscillations and, hence the search-time increases.

It is noticeable in Fig. 9 that for a constant value of n and small values of α (close to 0), the convergence time is a decreasing function of α. The reason is that the congestion detection time (which is a decreasing function of α) is the dominant part in the convergence time, when α is close to 0.

As we increase α, the congestion detection time drops while the search-time rises. Therefore, the convergence time does not change significantly and remains in the same range in some extent. For example, it can be seen in Fig. 9 that the convergence time for $n = 4$ does not change significantly for α such that $0.2 < \alpha < 0.6$. Nonetheless, as α gets closer to

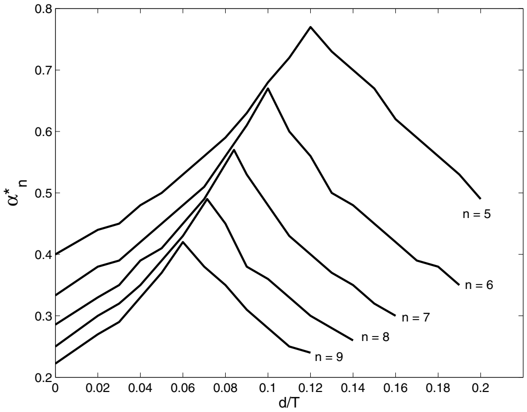

Fig. 8. Variations of α_n^* versus the normalized propagation delay.

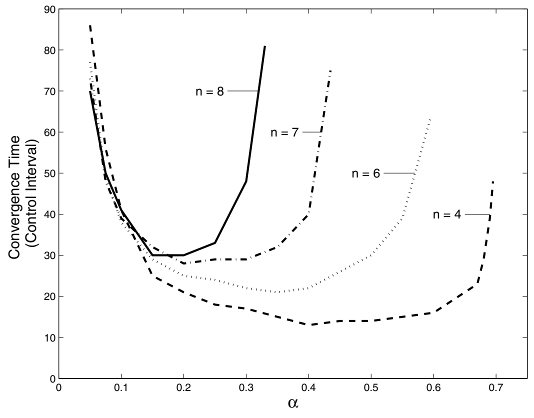

Fig. 9. Convergence time for various values of n and α with $\varepsilon = 0.01$ and $d = 0.1T$.

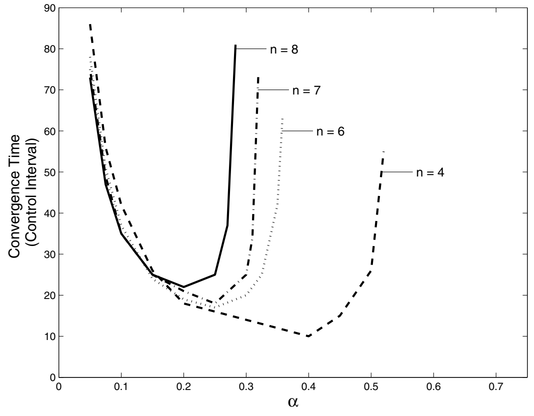

Fig. 10. Convergence time for various values of n and α with $\varepsilon = 0.01$ and $d = 0.02T$.

$\alpha_n^*(d = 0.1T)$ (which is for example around 0.7 for $n = 4$ in Fig. 9), the search-time is the main source of the delay in the 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 when α is small, the calculated fair rate does not oscillate and instead increasingly converges to the true fair rate (this is shown in Fig. 6 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 the stable domain of both systems, it is closer to $\alpha_{n_1}^*(d)$ than $\alpha_{n_2}^*(d)$. In this case, we can conclude that convergence time of the system with more node is greater. In addition, any change in the propagation delay can increase or decrease $\alpha_n^*(d)$ according to Fig. 8. For example, we observe in Fig. 8 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. 10 depicts the analytical results of the convergence time for different number of nodes for $d = 0.02T$. By comparing graphs in Figs. 9 and 10 it can be seen that $\alpha_n^*(0.02T) < \alpha_n^*(0.1T)$. From results of Figs. 9 and 10, we heuristically observe that for a system with the propagation delay, the convergence time of the fair rate is relatively small for α such that $\frac{1}{n} < \alpha < \frac{2}{n}$.

V. Conclusion

We developed an analytical model of the fair rate calculation in the standard RPR fairness algorithm for the parking lot scenario. Our approach was to model the fair rate calculation process as a nonlinear discrete-time low-pass filter. Simulation results confirmed 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 the system parameters on stability 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 the parking lot scenario is within its minimum range.

Appendix

In this appendix, we study some properties of the analytical model presented in Section III. We use *Equivalent Gains* method described in [10], to analyze stability conditions of a nonlinear system. This method is an intuitive technique where a nonlinear component is replaced by an equivalent gain and, then, the stability of the system with the equivalent gain is examined. A brief description of the equivalent gains method is provided next.

Definition 1: A function $\phi(x)$, with a scalar input and a scalar output, is assumed to belong to the sector $[k_1,k_2]$ if for all inputs x we have

$$k_1 \cdot x^2 \leq \phi(x) \cdot x \leq k_2 \cdot x^2, \tag{20}$$

where k_1 and k_2 are allowed to be $-\infty$ or $+\infty$. Note that k_1 and k_2 do not necessarily represent the minimum and maximum slopes of $\phi(x)$.

To analyze stability of a nonlinear system using the equivalent gains method, first the corresponding k_1 and k_2 of the nonlinear block of the system should be determined. Then, the nonlinear block should be replaced by a linear gain $k_{eq} \in [k_1,k_2]$. Finally, the stability of the linear system is examined at each point in the interval $[k_1,k_2]$.

A. Stability Analysis of the Model without Link Propagation Delay

In this section we analyze the stability condition of the fair rate calculation model without the propagation delay. In Section III-A, we showed that the fair rate in the model without the link propagation delay can be calculated by the nonlinear difference equation (10). In the following proposition, we investigate the condition with respect to which the fair rate calculation system is stable and the convergence time of the calculated fair rate is finite.

Proposition 1: For a given n, the nonlinear system depicted in Fig. 4 is stable when $\alpha < \frac{2}{n}$.

Proof: We use the equivalent gain theory to investigate the stability of this system. The nonlinearity in the system depicted in Fig. 4 is $\phi(x) = (x)^+ = \max\{0,x\}$. One can verify that the function $\phi(x) = \max\{0,x\}$ belongs to the sector $[0,1]$, i.e., $k_1 = 0$ and $k_2 = 1$. Using the equivalent gain method, the stability condition of the nonlinear system depicted in Fig. 4 can be analyzed by examining the stability of the equivalent linear system for all $k_{eq} \in [0,1]$. Replacing the nonlinear block in Fig. 4 by a linear gain k_{eq} yields the equivalent linear system with the following transfer function:

$$H_{k_{eq}}(z) = \frac{\alpha \cdot k_{eq}}{1 + \big(\alpha \cdot (k_{eq} \cdot (n-1) + 1) - 1\big) \cdot z^{-1}}. \qquad (21)$$

This system has a real pole at $z = 1 - \alpha \cdot (k_{eq} \cdot (n-1) + 1)$ and is stable if $-1 < z < 1$. Combining the inequalities $0 \le k_{eq} \le 1$, and $0 < \alpha < 1$, we can derive the following: $\alpha \le \alpha \cdot (k_{eq} \cdot (n-1) + 1) \le \alpha \cdot n$. Therefore, $1 - \alpha \cdot n \le z \le 1 - \alpha$ and the linear system is stable if $\alpha < \frac{2}{n}$. This means that the equivalent linear system is stable for all $k_{eq} \in [0,1]$ if $\alpha < \frac{2}{n}$. Therefore, the nonlinear system depicted in Fig 4 is stable when $\alpha < \frac{2}{n}$. ■

B. Stability Analysis of the Model with Link Propagation Delay

It was shown in Section III-B that the fair rate calculation in the system with the propagation delay can be modeled by a second-order nonlinear discrete-time filter. The approach used to analyze the stability of the model without the propagation delay can be applied to this more realistic system in order to determine its stability conditions: After substituting (16) in (1), the fair rate calculation is modeled as a second-order nonlinear filter with a unit step function as its input. Then, each nonlinear block i $\big(\phi(x) = (x)^+ = \max\{0,x\}\big)$ is replaced by a corresponding linear gain k_{eq}^i. Finally, the stability of the equivalent linear system is investigated for all $k_{eq}^i \in [0,1]$. However, there are n nonlinear blocks in (16) and the stability analysis of such a system would be very complicated. In the following, we only investigate the stability of the system with the propagation delay for $k_{eq}^i = 1$, $\forall i$. In other words, we analyze the stability of this system when all nonlinear blocks are bypassed. By substituting (16) in (1) and replacing each nonlinear block i by a linear gain $k_{eq}^i = 1$ we have

$$\begin{aligned}\tilde{r}_1^a(k) &= (1-\alpha) \cdot \tilde{r}_1^a(k-1) \\ &+ \alpha \cdot \Bigg[\Big(1 - (n-1) \cdot \tfrac{d}{T}\Big) \cdot \big(1 - (n-1) \cdot \tilde{r}_1^a(k-1)\big) \\ &+ \frac{d}{T} \cdot \sum_{j=1}^{n-1} \big(1 - (n-j) \cdot \tilde{r}_1^a(k-2) - (j-1) \cdot \tilde{r}_1^a(k-1)\big)\Bigg]. \qquad (22)\end{aligned}$$

The transfer function corresponding to the linear difference equation in (22) is as follows:

$$H_d(z) = \frac{N(z)}{D(z)}, \qquad (23)$$

where

$$N(z) = \alpha \cdot \Big(1 + (n-1) \cdot (n-2) \cdot \frac{d}{2T}\Big), \qquad (24)$$

and

$$\begin{aligned}D(z) = 1 - \Big((1 - \alpha \cdot n) + \alpha \cdot n \cdot (n-1) \cdot \frac{d}{2T}\Big) \cdot z^{-1} \\ + \ \alpha \cdot n \cdot (n-1) \cdot \frac{d}{2T} \cdot z^{-2}. \qquad (25)\end{aligned}$$

Note that equation (22) represents the difference equation of the response of $H_d(z)$ to the unit step function. We also have

$$H(z) = H_d(z)\Big|_{d=0}, \qquad (26)$$

where $H(z)$ is the transfer function described in (11).

When n and α are fixed, let $\alpha_n^*(d)$ denotes the supremum value of α such that $\forall \alpha < \alpha_n^*(d)$, $H_d(z)$ is stable, i.e., its poles are within the unit circle. For $H(z)$, we have $\alpha_n^*(0) = 2/n$ according to (12).

Proposition 2: For a fixed n and any $d > 0$ we have $\alpha_n^*(0) < \alpha_n^*(d)$.

Proof: $H_d(z)$ has two poles as follows:

$$z_1 = \frac{1}{2} \cdot \Big(1 - \gamma + \gamma \cdot \delta/2 + \sqrt{f(\gamma)}\Big),$$

and

$$z_2 = \frac{1}{2} \cdot \Big(1 - \gamma + \gamma \cdot \delta/2 - \sqrt{f(\gamma)}\Big),$$

where $\gamma \triangleq \alpha \cdot n$, $\delta \triangleq (n-1) \cdot d/T < 1$, and $f(\gamma) \triangleq (1 - \gamma + \gamma \cdot \delta/2)^2 - 2 \cdot \gamma \cdot \delta$. $H_d(z)$ is stable when $|z_1| < 1$ and $|z_2| < 1$. For a given n and an arbitrary $d > 0$, the poles may be a real or a conjugate pair depending on the sign of $f(\gamma)$. In order to determine the sign of $f(\gamma)$ let us solve the equation $f(\gamma) = 0$ for γ. It has two real poles: $\gamma_1 = (\delta/2 + 1 - \sqrt{2 \cdot \delta})/(\delta/2 - 1)^2$ and $\gamma_2 = (\delta/2 + 1 + \sqrt{2 \cdot \delta})/(\delta/2 - 1)^2$, with $\gamma_1 < \gamma_2$. The following three cases are distinguished based on the position of γ with respect to γ_1 and γ_2:

1) $\gamma < \gamma_1$: In this case $f(\gamma) > 0$ and both poles z_1 and z_2 are real with $z_1 < z_2$. As $\delta < 1$, one can check that $\gamma_1 < 1$, thus $\gamma < 1$. Consequently, we can show by inspection that $1 - \gamma + \gamma \cdot \delta/2 > 0$ and, as a result, $z_2 > 0$. Since z_1 is an upper bound of z_1 and z_2 and $z_1 < 1 - \gamma + \gamma \cdot \delta/2 < 1 - \gamma/2 < 1$, both poles are within the unit circle and the system is always stable.

2) $\gamma_1 < \gamma < \gamma_2$: In this case, $f(\gamma) < 0$ and $H_d(z)$ has two conjugate poles such that $|z_1|^2 = |z_2|^2 = \gamma \cdot \delta/2 = \alpha \cdot n \cdot \delta/2$. Therefore, $H_d(z)$ is stable whenever $\alpha \cdot n \cdot \delta/2 < 1$. This means that $\alpha_n^*(d) = 2/(n \cdot \delta)$. As $\delta < 1$, we have $\alpha_n^*(d) > \alpha_n^*(0) = 2/n$.

3) $\gamma > \gamma_2$: In this case $f(\gamma) > 0$ and both poles are real. As $\gamma > \gamma_2$, we can show that $(1 - \gamma + \gamma \cdot \delta/2) < 0$ and $z_2 < z_1 < 0$. This means that $H_d(z)$ is stable as long as z_2 is inside the unit circle. We have $0 > z_2 > 1 - \gamma + \gamma \cdot \delta/2 > 1 - \gamma = 1 - \alpha \cdot n$. Note that $z = 1 - \alpha \cdot n$ is the pole of $H(z)$. In other words, for

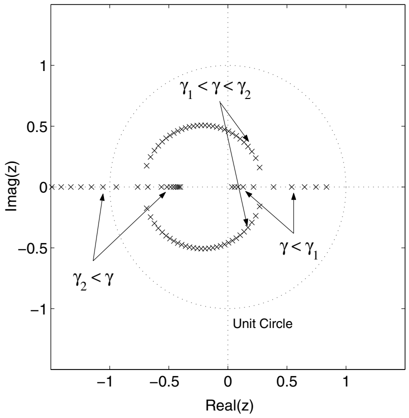

Fig. 11. Locations of $H_d(z)$ poles in the z-plane due to variation of α for $n = 8$ and $d = 0.05T$.

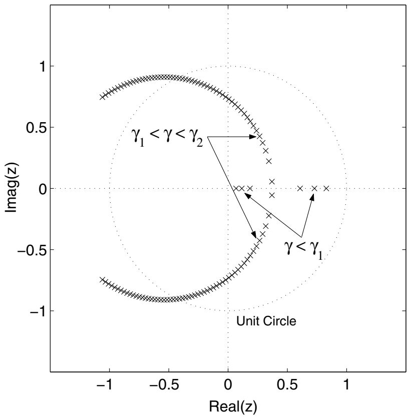

Fig. 12. Locations of $H_d(z)$ poles in the z-plane due to variation of α for $n = 8$ and $d = 0.1T$.

some value of α, the pole of $H(z)$ is outside the unit circle while z_2 is still inside the unit circle. That is, $\alpha^*_n(d) > \alpha^*_n(0)$. ■

Proposition 3: For a given n, the maximum value of $\alpha^*_n(d)$ is achieved when $(n-1) \cdot d = 0.5 \cdot T$.

Proof: Recall that for any $\gamma < \gamma_1$, both poles of $H_d(z)$ are within the unit circle. As a result, the determination of $\alpha^*_n(d)$ depends on the position of γ with respect to γ_2. We distinguish two cases as follows:

1) $\delta < 0.5$: We prove that when $\delta < 0.5$, $\alpha^*_n(d)$ is an increasing function of δ. One can check that when $\delta < 0.5$, we have $\gamma_2 < 4$ and, consequently, the inequality $\alpha \cdot n \cdot \delta < 2$ (stability condition of $H_d(z)$) holds for any $\gamma_1 < \gamma < \gamma_2$. In other words, $H_d(z)$ is stable for any $\gamma < \gamma_2$. When $\gamma > \gamma_2$, we showed that z_1 and z_2 are real and $z_2 < z_1 < 0$. Therefore, $H_d(z)$ is stable as long as z_2 remains within the unit circle. From the definition of $a^*_n(d)$, one can notice that in this case $a^*_n(d)$ is obtained by solving the equation $z_2 = -1$ (as for any $\alpha < a^*_n(d)$, z_2 remains within the unit circle). This yields $\alpha^*_n(d) = 2/\big((1-\delta) \cdot n\big)$. Hence, for $\delta < 0.5$, $\alpha^*_n(d)$ is an increasing function of δ. Fig 11 shows the locations of $H_d(z)$ poles due to variations of γ in an example with $\delta < 0.5$. It can be seen that z_2 is real when it falls outside the unit circle for some $\gamma > \gamma_2$.

2) $\delta > 0.5$: We prove that when $\delta > 0.5$, $\alpha^*_n(d)$ is a decreasing function of δ. For $\delta > 0.5$, we can check that $\gamma_2 > 4$. In this case, the inequality $\alpha \cdot n \cdot \delta < 2$ (stability condition of $H_d(z)$) is satisfied only for γ such that $\gamma_1 < \gamma < \gamma_2$ and $|z_1|^2 = |z_2|^2 = \gamma \cdot \delta/2 < 1$. As a result, for any $\gamma = \alpha \cdot n < 2/\delta$, both poles are within the unit circle. From definition of $\alpha^*_n(d)$ it follows that $\alpha^*_n(d) = 2/(\delta \cdot n)$, which is a decreasing function δ. Fig 12 depicts the locations of $H_d(z)$ poles due to variations of γ in an example with $\delta > 0.5$. It can be seen that z_1 and z_2 are conjugate when they fall outside the unit circle for some $\gamma_1 < \gamma < \gamma_2$.

As $\alpha^*_n(d)$ is an increasing function of δ for $\delta < 0.5$ and it is a decreasing function of δ for $\delta > 0.5$, the maximum value of $\alpha^*_n(d)$ is achieved when $\delta = (n-1) \cdot d/T = 0.5$. ■

References

[1] 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.

[2] 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.

[3] F. Davik, M. Yilmaz, S. Gjessing, and N. Uzun, "IEEE 802.17 Resilient Packet Ring Tutorial," *IEEE Communication Magazine*, vol. 42, no. 3, pp. 112-118, March 2004.

[4] I. Cidon, L. Georgiadis, R. Guerin, and Y.Shavitt, "Improved fairness algorithms for rings with spatial reuse," *IEEE/ACM Transactions on Networking*, vol. 5, no. 2, pp. 190-204, April 1997.

[5] D. Tsiang, and G. Suwala, "The Cisco SRP MAC layer protocol," *IETF Networking Group, RFC 2892*, August 2000.

[6] 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,* vol. 21, 1991.

[7] 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*, vol. 12, no. 1, pp. 85-102, February 2004.

[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] IEEE, IEEE Standard 802.17: Resilient Packet Ring (Draft Version 3.1), *http://ieee802.org/17*, February 2004.

[10] G. F. Franklin, J. D. Powell, M. L. Workman, *Digital Control of Dynamic Systems*, Addison-Wesley Longman Publishing Co. Inc., Boston, MA, 1997.