An Analytical Bound for Convergence of the Resilient Packet Ring Aggressive Mode Fairness Algorithm

Fredrik Davik
Simula Research Laboratory
University of Oslo
Ericsson Research Norway
Email: bjornfd@simula.no

Amund Kvalbein
Simula Research Laboratory
Email: amundk@simula.no

Stein Gjessing
Simula Research Laboratory
Email: steing@simula.no

***Abstract*— Resilient Packet Ring (RPR) is a new standard, designated IEEE standard number 802.17, for MAN and WAN dual ring topologies. RPR uses the buffer insertion principle as a basis for its medium access control protocol. In this paper, we analyze parts of the aggressive mode of the RPR fairness protocol. We look at a congested node, and utilize control systems theory to analyze the stability of the associated fairness algorithm. In particular, we discuss how the settings of the two important parameters ageCoef and lpCoef influence the stability of an RPR-network. At the end of the paper we present simulated scenarios in order to illustrate our results.**

Keywords: Resilient Packet Ring, Fairness, Control Systems, Simulations

I. Introduction

In a ring network, when nodes transmit more data than the ring bandwidth can sustain, we have a resource allocation problem. The solution to the problem is twofold: i) a fair distribution of the bandwidth must be defined, and ii) a policy for this fair distribution of bandwidth must be implemented and enforced.

A recent addition to the protocol family for ring topologies is the IEEE standard for Resilient Packet Rings (RPR), IEEE standard 802.17 [1]. RPR uses the RIAS principle for fair allocation of bandwidth [2]. This means that when a link is congested, the available bandwidth should be fairly (according to the RIAS definition) distributed between all nodes that transmit data over this link.

RPR belongs to the class of ring networks based on the buffer insertion principle [3], [4]. Legacy ring technologies based on the same principle include SCI [5], CRMA-II [6], [7] and MetaRing [8]. Other classes of ring technologies are slotted medium access control protocols (Cambridge Ring [9], ATMR [10]) and token based protocols (IEEE 802.5 Token Ring [11] and FDDI [12]).

RPR's policy for fair division of ring bandwidth is enforced by the fairness algorithm. There are two modes of operation for this algorithm. These two modes are respectively the *conservative* mode of operation, discussed in [13], and the *aggressive* mode of operation, discussed in [13], [14]. The *conservative* mode of operation, uses a form of rate control, where the congested node (i.e. the node immediately upstream of a congested link) issues a rate change command (back-pressure message, also called a fairness message) and then waits to the see the resulting effect. When the estimated waiting period has elapsed, if the resulting effect is not the RIAS fair division of bandwidth, a new rate value is calculated and distributed to upstream nodes. The estimation of the waiting period is based on periodic measurements, and is denoted the *Fairness Round Trip Time* (*FRTT*).

For the *aggressive* mode of operation, a waiting period is not estimated. Instead the congested node periodically sends fairness messages upstream, containing the congested station's estimate of the fair rate. This estimate is calculated based on measurements of the congested node's own send rate and statically configured parameters. These statically configured parameters will typically be set based on heuristic guidelines or expert knowledge/simulation results. Obvious weaknesses with this approach is the risk of mis-configurations that may result in instabilities and/or underutilization of the network.

In this paper, we show analytically how these parameters should be set in a system. By this, we reduce the risk of network instability as well as easing the task of configuring a Resilient Packet Ring network.

In section II we give a short introduction to the RPR fairness algorithm and the vocabulary used in the rest of the paper. In section III, our main analysis of the *aggressive* fairness algorithm is developed. Then, in section IV, we will present and discuss simulation scenarios and associated simulation results we use to verify our analytical results. Finally in sections V and VI, we refer to related work, conclude and give some directions to further work.

II. The RPR fairness algorithm

RPR defines three data packet service classes: high priority (class A), medium priority (class B) and low priority (class C). The high and a configured portion of the medium-priority

traffic is sent using reserved bandwidth[1], while the rest of the medium and the low-priority traffic uses the remaining (unreserved) bandwidth. In a congested situation, the unreserved bandwidth is divided between the contending nodes by the fairness algorithm. Traffic sent using the unreserved bandwidth is referred to as *fairness eligible*, since the fair send rate for this traffic is governed by the fairness algorithm.

An RPR node may contain two separate insertion buffers, known in RPR as transit queues. With such a design, transit traffic of class A is assigned to the Primary Transit Queue (PTQ), while transit traffic of classes B and C use the Secondary Transit Queue (STQ). In this paper we assume that all nodes have two transit queues. The high-level architecture of an RPR node is shown in Fig. 1.

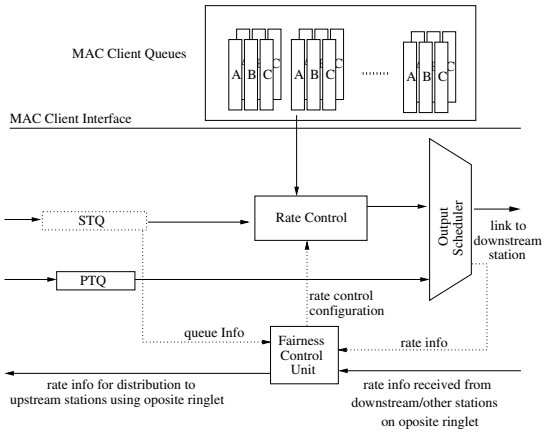


Fig. 1. Generic RPR node design, showing a node's attachment to the ring for transferral of data in the east direction and transferral of fairness messages (rate info) in the west direction. The solid lines indicates the flow of data through the node. The dotted lines indicates the exchange of control/configuration information between node internal function blocks.

When the demand for bandwidth over a link is higher than the available capacity, the link becomes congested. We also say that the node upstream from this link is congested. In a congested node, the STQ is filling up, because the node itself adds traffic at the same time as class B or C traffic is transiting the node. As long as the STQ is only partially filled, the bandwidth on the out-link is equally divided between the local add traffic and the transit traffic, and the node is still in the *uncongested* state. However, when the STQ-occupancy exceeds a threshold, denoted *low*, the node is by definition *congested* [1], and actions (described in section III) will be taken to alleviate congestion. If the STQ continues to fill and the STQ-occupancy exceeds a *high*-threshold, the scheduling rules of RPR causes the local (fairness eligible) add traffic to stop.

A congested node constitutes the *head* of a so called *congestion domain*. The node furthest away, and upstream from the congested node, that contributes to the congestion, is defined as *tail* of the congestion domain. Hence a *congestion domain* contains all nodes that contribute to the congestion of a certain link, including any nodes in between, even if these nodes are not contributing to the congestion. Fig. 2 shows an example of a *congestion domain*.

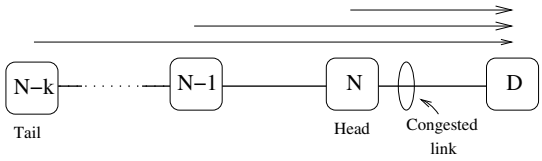


Fig. 2. Nodes N, N-1,..., N-k all send traffic to node D. Thus node N becomes the most congested node in the segment. Node N then becomes head of the *congestion domain* while node N-k, which is the node furthest away, contributing to the congestion, becomes the tail of the congestion domain.

It is the responsibility of the *head* to alleviate congestion by dividing the available bandwidth of the congested link between all contending nodes, so that each gets its fair share of the capacity of the congested link. The *head* does this by calculating a so called *fair rate*, and advertises this rate upstream to all nodes in the *congestion domain*. No node, having received this rate message, is then allowed to send at a rate higher than the (received) fair rate, over the congested link. As mentioned above, there are two versions of this fairness algorithm. In this paper we analyze the so called *aggressive* version.

III. Aggressive Mode Rate Control Configuration

In general, in feedback control systems not using predictive methods, a rule of thumb for design of the control system is that the time constant for the controller part of the system, should not be faster than the time constant of the system itself. Thus, given the time constant of the controller, this introduces a constraint with respect to which systems can be safely controlled. In an RPR-network, the time constant of the system (a *congestion domain* with its associated nodes) will change dynamically, depending on the load situation on the ring. The time-constant of the system consists of a fixed part and a dynamic part.

The fixed part is the propagation delay from the *head* of the *congestion domain* to the *tail* and back. Also, the per node processing of fairness messages can be considered fixed. However, the processing delay, is so small that it has no significance for the scenarios considered in this paper.

The dynamic part consists of two sub-parts: i) the queuing delay, experienced by a fairness message, when propagating from the *head* to the *tail*, and ii) the time required to empty the transit queues in the path between *tail* and the *head* before the rate change at the *tail* is observable at the *head*.

In this section, we show how the setting of two parameters, named $ageCoef$ and $lpCoef$, relates to the steady-state and transient-response behavior of the rate control algorithm for a *congestion domain head*.

On an RPR-network, the *head* of a *congestion domain* calculates the fair division of send rates for traffic traversing

[1]Actually, some of the bandwidth called reserved in this paper may dynamically be allocated to fairness eligible traffic. This simplification, however, does not influence the results presented.

the congested link. The calculation is done at periodic intervals (every aging interval). According to the standard, the aging interval is 100μs (for line rates $\geq$ 622Mbit/s). In the RPR standard, the calculation of the rate is specified as two cascaded low-pass filters, also referred to as rate-counters. The first filter, takes the weighted sum of locally added traffic, denoted $x(n)$, during the current aging-interval, n, and the previous filter output-value, $addRate(n-1)$. The resulting sum is the new output-value of the filter, $addRate(n)$. This value is then fed into the 2^{nd} low-pass filter, which takes the weighted sum of its input, $addRate(n)$, and the previous output-value of the filter, $lpAddRate(n-1)$. The resulting sum is the new output-value of the filter, $lpAddRate(n)$.

The calculated value, $lpAddRate(n)$, becomes the *head*'s new estimate of the fair rate, and is distributed in the form of a rate change command, in RPR known as a *fairness message*, to the upstream neighbors. The filtering process described above can be formulated in terms of a discrete-time 2^{nd}-order low-pass filter, with a sampling period which equals the aging interval. The filtering processes is described formally below.

Let the amount of locally added traffic during aging interval n be denoted $x(n)$, and let:

$$p_1 = \frac{ageCoef-1}{ageCoef}, \text{ where} \quad ageCoef \in \{1,2,4,8,16\} \tag{1}$$

Then, from the standard, we have:

$$addRate(n) = addRate(n-1) \cdot p_1 + x(n) \tag{2}$$

$$lpAddRate(n) = \frac{1}{lpCoef}\left(lpAddRate(n-1) \cdot (lpCoef-1) + addRate(n)\right) \tag{3}$$

Where: $lpCoef \in \{16, 32, 64, 128, 256, 512\}$

This can be modeled as a 2^{nd} order digital low-pass filter as shown in Fig. 3. In the figure, the box with the marking z^{-1} denotes that the value on the output of the box is delayed one sampling period (i.e. one aging interval) as compared to the value on the input of the box. The filter input and output-values, denoted respectively $X(z)$ and $Y(z)$ is the Z-domain representations of the discrete time-domain signals $x(n)$ and $y(n)$, where $y(n) = lpAddRate(n)$.

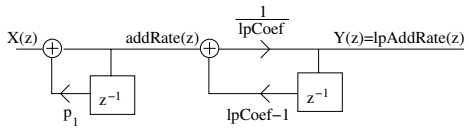


Fig. 3. Block Diagram of the two Cascaded First-Order Low-Pass Filters Yielding the Fair Rate Based on the Congestion Head's Own Send Rate $x(n)$

Let:

$$p_2 = \frac{lpCoef-1}{lpCoef} \tag{4}$$

Then the transfer function, $H(z)$, of the 2^{nd} order low-pass filter shown in Fig. 3 can be written as shown in (5) below.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{lpCoef} \cdot \frac{z^2}{(z-p_1)(z-p_2)} \tag{5}$$

Given this model of the *aggressive* fairness algorithm's rate control mechanism, using discrete control systems' analysis techniques, we can deduce some of its basic properties. For the purpose of this paper, the most important system properties are the transient response and stability. In this paper, we use a simplistic approach, where we from the properties of the rate control part of the system, i.e. the 2^{nd} order low-pass filter, deduce some properties, and then we utilize the findings to determine some boundary operating conditions for a system. Specifically, we determine the time-constant of the filter when applied to a step input and use this time-constant to determine the boundary conditions of the system.

If a node, transiting traffic from upstream nodes, has sent no traffic on the ring lately, the value of its $lpAddRate$ rate-counter will be equal or close to zero. Assume that the node is currently uncongested. Also assume that both the aggregate demand of upstream nodes and the local demand for bandwidth, over the link connected to the node's downstream neighbor, exceed half of the available bandwidth. Then the scheduling rules of RPR will result in equal division of bandwidth between local and transit traffic. With the sum of demands being greater than the available bandwidth, this will result in the secondary transit queue (STQ) starting to fill. As long as the STQ-occupancy stays below the high threshold, the local add-rate, denoted $x(n)$, will be constant and equal to half of the available bandwidth. Once the STQ-occupancy exceeds the high threshold, the scheduling rules of RPR causes the local add-rate, $x(n)$, to be 0, to avoid overflow of the STQ. The value of $x(n)$ will remain 0 until the STQ-occupancy falls below the *high* threshold.

The scheduling rules described above, are summarized in (6) below:

$$x(n) = \begin{cases} bandwidth & \text{, STQ-occupancy} = 0 \\ \frac{bandwidth}{2} & \text{, } 0 < \text{STQ-occupancy} < \textit{high} \\ 0 & \text{, otherwise} \end{cases} \tag{6}$$

These scheduling rules, controlling the value of the node's local add-rate, $x(n)$, together with the configuration of the 2^{nd} order low-pass filter shown in Fig. 3, control the behavior of the $lpAddRate(n)$ rate-counter in the *congestion domain head*. Now, let us consider the scenario where a node's $lpAddRate = 0$ and the STQ-occupancy is close or equal to 0. The node is currently transiting data from upstream nodes at the available bandwidth of the link connected to the downstream neighbor. Next, the node starts sending local (add) traffic over the same link, which at some point in the near future will make the node *head* of a *congestion domain*. From the time where the node starts transmitting and onwards, the behavior of its $lpAddRate$ rate-counter is characterized by a set of cycles. A cycle consist of two consecutive periods. In the first period, $x(n) = \frac{bandwidth}{2}$, and thus $lpAddRate(n)$ monotonically increases until the STQ-occupancy exceeds the *high* threshold. At that point, the second period starts. In the second period, $x(n) = 0$, and thus $lpAddRate(n)$ starts monotonically decreasing. The second period ends when STQ-occupancy falls below the *high* threshold. Thus the cycle is

concluded.

The behavior of the rate control algorithm for a *congestion domain head*, resembles that of the step response of a second order continuous-time system. In (7) below, $G(s)$ is the Laplace transform of the general 2^{nd}-order continuous-time system function, $g(t)$:

$$G(s) = \mathcal{L}(g(t)) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{7}$$

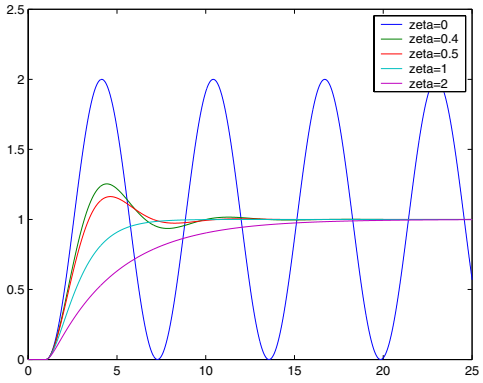


Fig. 4. Unit step response for 2nd order continuous-time system.

For an unstable second order continuous-time system, the system output value, in response to a step input, represented in Fig. 4 above, with the plot where $\zeta = 0$, oscillates between the max and min values determined by the system's inherent boundaries, with a frequency as determined by the system's natural frequency, ω_n. For a corresponding stable system, represented by the plots in Fig. 4 where $\zeta \in \{0.4, 0.5, 1, 2\}$, the system output in response to the same input, converges to the same steady-state value. The speed of convergence and the presence and magnitude of oscillations during convergence however, depends on the setting of ζ.

Given the cyclic behavior of $lpAddRate(n)$, the stability of the *aggressive* mode fairness algorithm, can be studied in terms of the step-response of the rate-control (fairness) algorithm executing in the head-node. To determine whether an RPR-network will stabilize or not, it is sufficient to analyze the first period of a cycle (described above) that will result from a given configuration of the $ageCoef$ and $lpCoef$ parameters. The analysis consists of determining the time-constant, τ, for a given configuration. Once this time-constant has been decided, this represents an upper bound for the time-constant of a system to be safely controlled using this configuration. Below, an analytical expression for the value of τ is derived.

As discussed above, it is safe to assume that the input to the system, $x(n)$, during the first half of the initial cycle, can be described in terms of a step function:

$$x(n) = \begin{cases} 0, & n < 0 \\ \Delta, & \text{otherwise, where } \Delta = \frac{bandwidth}{2} \end{cases} \tag{8}$$

Then, the corresponding Z-transform, $X(z)$, of the input signal, $x(n)$, is:

$$X(z) = Z\{x(n)\} = \frac{\Delta \cdot z}{z - 1} \tag{9}$$

If we evaluate the output of the system, $Y(z)$, in response to the input, $X(z)$, we have (from (5) and (9)) the relationship:

$$Y(z) = \frac{\Delta}{lpCoef} \cdot \frac{z^3}{(z-1)(z-p_1)(z-p_2)} \tag{10}$$

If we perform a partial fraction expansion on (10) above before performing the inverse Z-transform on the signal $Y(z)$, we get the (discrete) time domain representation of the output signal:

$$y(n) = Z^{-1}\{Y(z)\} = \tag{11}$$
$$\frac{\Delta}{lpCoef} \cdot \left(\frac{1}{(1-p_1)(1-p_2)} + \frac{p_1^{n+2}}{(p_1-1)(p_1-p_2)} + \frac{p_2^{n+2}}{(p_2-1)(p_2-p_1)} \right)$$

This function resembles an exponential ramp function, consisting of a constant part and two parallel first order-filter functions with a steady-state value:

$$y_{ss} = \lim_{n\to\infty}(y_n) = \frac{\Delta}{lpCoef \cdot (1-p_1)(1-p_2)} = ageCoef \cdot \Delta \tag{12}$$

As the ratio $\frac{p_2}{p_1}$, from the configuration of parameters $ageCoef$ and $lpCoef$, becomes larger than 1, the transient behavior, and thus the function's time-constant, is determined more by the last fraction, denoted $y_2(n)$, where:

$$y_2(n) = \frac{\Delta}{lpCoef} \cdot \frac{p_2^{n+2}}{(p_2-1)(p_2-p_1)} \tag{13}$$

Thus in these cases, it is sufficient to study $y_2(n)$ to determine the transient response of the 2^{nd} order filter. $y_2(n)$ together with the first fraction from (11), represents a first order filter system. Given that the input to the filter, $x(n)$, is constant during the observation interval, the value of $y_2(n)$, evaluated at discrete points $n \in \{0, 1, 2, ..., N\}$, will match exactly the corresponding points on a continuous-time filter with the same output-function as shown in (13). In the case of the continuous-time filter, however, the output-values is defined for all values of $t \geq 0$ (not only at discrete times, as is the case of $y(n)$). To find the time-constant of the digital filter, we analyze the filter in terms of output-function for an equivalent continuous-time filter-function (we replace the discrete variable n in (11) with the continuous variable t).

For a first order continuous-time (filter) system, the system time-constant, τ, in response to a step function applied to the system input, is defined as shown in (14) [15].

$$\frac{y_{2_{ss}}}{\tau} = \frac{d}{dt}\left(y_2(t)\right)|_{t=0} \tag{14}$$
$$= \frac{d}{dt}\left(\frac{\Delta}{lpCoef} \cdot \frac{p_2^{t+2}}{(p_2-1)(p_2-p_1)} \right)\Bigg|_{t=0}$$
$$= \frac{\Delta}{lpCoef} \cdot \frac{ln(p_2) \cdot p_2^2}{(p_2-1)(p_2-p_1)}$$

$$\Rightarrow \tau = \left(\frac{y_{2_{ss}} \cdot lpCoef (p_2-1)(p_2-p_1)}{\Delta \cdot ln(p_2) \cdot p_2^2} \right) \tag{15}$$

In (14), $y_{2_{ss}}$ represents the steady-state value of y_2 as $t \to \infty$. In the case of an exponential increasing function with a non-zero start value ($y_2(0) \neq 0$), the $y_{2_{ss}}$ factor in the numerator on the left hand side can be replaced with the expression $y_{2_{ss}} - y_2(0)$. Evaluating y_2 at 0 and ∞ gives:

$$y_{2_{ss}} = lim_{t\to\infty}(y_2(t)) = 0 \quad (16)$$

$$y_2(0) = \frac{\Delta}{lpCoef} \cdot \frac{p_2^2}{(p_2-1)(p_2-p_1)} \quad (17)$$

Substituting $(y_{2_{ss}} - y_2(0))$ for $y_{2_{22}}$ in (14), we get:

$$\tau = \frac{-1}{ln(p_2)} = \frac{-1}{ln(\frac{lpCoef-1}{lpCoef})} \approx lpCoef \quad (18)$$

The final simplification is obtained when using a power-series expansion of the $ln(\frac{lpCoef-1}{lpCoef})$ expression in the denominator.

Note that in (18) above, the time-constant is specified in units of t, which for the discrete filter corresponds to the number of sampling-periods. As noted at the start of the chapter, the sampling period equals one aging interval.

As an example, given a system with $ageCoef = 4$ and $lpCoef = 64$, the resulting value of τ from (18) above becomes $\tau \approx 64$ [aging intervals] $= 6.4ms$. This means that in a scenario with a node connected upstream of a congested link, having the configuration shown above, the maximum system time-constant (as discussed at the start of this chapter) equals 6.4ms. Failure to operate within these boundary conditions may result in an unstable system, i.e. the system fails to converge to the fair division of send rates over the congested link.

In an RPR-network, when the time-constant of the rate-controlling (fairness) algorithm executing in the *head* node is too fast compared to the system's time-constant, the system will be unstable. The symptoms of an unstable RPR-network, like that shown in Fig. 6 b), resembles those of a 2^{nd} order continuous-time system shown in Fig. 4 above with $\zeta = 0$. In the case of an unstable RPR-network, the output of the contending stations will oscillate between a max and a min value, where the max value is limited to the value of the available bandwidth and the min value is limited to 0. When the time-constant of the rate controlling (fairness) algorithm equals or exceeds the system's time-constant, the system will be stable. When this is the case, the throughput of the contending stations converges to the (RIAS) fair division of bandwidth. This behavior is shown in Fig. 6 a). Depending on the ζ "equivalent" setting of the RPR-network, the speed of convergence and presence and magnitude of oscillations is decided.

In this chapter, we have discussed how the setting of the two parameters, named $ageCoef$ and $lpCoef$, relate to the steady-state and transient-response behavior of the fairness algorithm for a *congestion domain head*, executing the *aggressive* mode of the RPR fairness algorithm. In the next chapter, we will illustrate the findings using our RPR simulator model implemented within the OPNET Modeler framework [16].

IV. Simulation Results

In this section, we describe the simulation scenario used to illustrate and evaluate the analytical results described above.

For the scenario used, often referred to as a "hot"-receiver scenario, we aim to show the relation between the link length (propagation delay) and the system settling time. With system settling time, we mean the time used from congestion occurs, until a fair division of send rates has been established by the fairness algorithm. For an unstable configuration, the settling time will be infinite, as the fairness algorithm never converges to the fair division of send rates.

Our simulation scenario is illustrated in Fig. 5. Node 4 is the "hot" receiver, receiving traffic from nodes 0 - 3. These nodes all try to send at their maximum allowed rate, making the link between nodes 3 and 4 the most congested link. We use a fixed packet size of 500B, and the STQ buffer thresholds *full*, *high* and *low* are set to: 254400, 63625 and 31812 bytes.

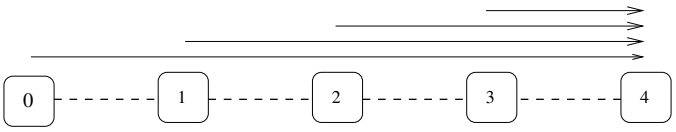
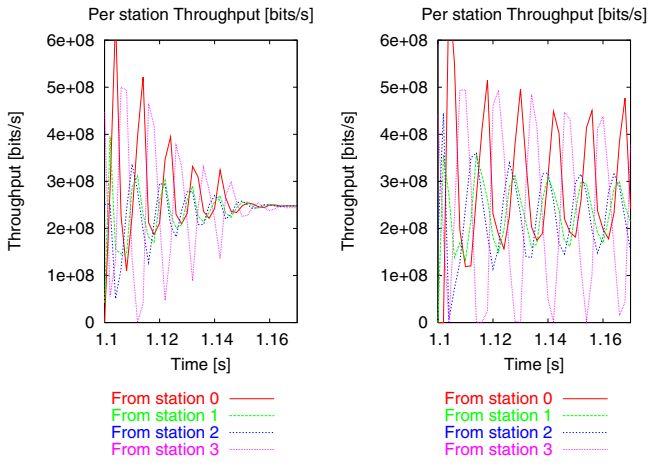


Fig. 5. In this hot receiver scenario, nodes 0, 1, 2 and 3 send at their maximum allowed rate to the "hot" receiver – node 4.

A. Fairness Convergence as a Function of Link Delay

In this simulation experiment, we want to illustrate the similarity between the system response of an RPR-network and the step-response of a second-order continuous time system with various values for the ζ variable shown in (7) above. We show two sets of output-curves, one (Fig. 6 a)) where the aggregate link-delay (system time-constant) is within the stability bounds of the RPR-network and thus, the RPR-network converges to the fair division of link bandwidth. The other set (Fig. 6 b)) shows the throughput-curves for a scenario where the system time-constant exceeds the stability bounds of the RPR-network. Correspondingly, for this scenario, the RPR-network does not converge to the fair division of link bandwidth. For both sets, $lpCoef = 64$ and $ageCoef = 4$.

As noted above, Fig. 6 a) shows a scenario where the configuration of the $lpCoef$ parameter is set to a value causing



a) link-delay is $500\mu s$ b) link-delay is $750\mu s$

Fig. 6. Throughput of traffic received by station 4 in hot receiver scenario.

the fairness algorithm to converge to the fair division of send rates. The link-delay in this scenario is 500μs, thus the aggregate link-propagation delay (from node 3 to node 0 and back) is 3ms. The corresponding analytical system time-constant is given by (18) and gives a bound of 64 aging intervals (i.e. 6.4 ms). When we compare the throughput convergence of the RPR-network with the convergence of the continuous-time system shown in Fig. 4 we can observe the similarity. For a continuous-time system, with a value of $\zeta < 1$, the convergence to the steady-state value is characterized by exponential (decreasing) oscillations. The lower the value of ζ, the higher the oscillations will be. Finally, when ζ becomes zero, the system will never converge and the oscillations are bounded by the physical boundaries of the system.

Similarly, for our RPR-network, if the time-constant, τ, of the rate-controller, is large compared to the system time-constant, the convergence towards fair rate will proceed slowly and surely without oscillations. If τ is in the (lower) proximity of the time-constant of the RPR-network (as the one showed in Fig. 6 a)), the RPR-network converges to the fair division of rates with exponential decreasing oscillations. Finally, as illustrated in Fig. 6 b), when the link delay is increased to 750μs, the value of τ clearly becomes too low compared to the system time-constant. As shown, for this configuration, the throughput of the contending nodes experience oscillations that are approximately constant in amplitude and the system does not converge to a steady-state (fair rate) value.

Given the analytical results presented in section III, the observant reader may find the behavior plotted in Fig. 6 b) unexpected. The analytical results indicate that given a configuration where $lpCoef$ is set to 64, the rate-controller should be able to handle networks where the system-delay is bounded upwards to 6.4ms. In the case of 750μs link-delays, the aggregate propagation delay from node 3 to node 0 and back is 4.5ms. Additionally fairness messages may incur more than 100μs processing delay per node traversed, when propagating from the *head* to the *tail*, resulting in a delay close to 5ms. Finally, queueing and scheduling delays experienced in both the upstream path (experienced by rate messages sent from *head* to *tail*) and the downstream path adds to the total delay. Thus in total, the system time-constant becomes too large compared to the value of τ.

B. Fairness Convergence as a Function of Total Link Delay

By varying the link length of each link, we can control the aggregate propagation delay from the *head* to the *tail* and back in the *congestion domain*. For each of the allowed values of $lpCoef$ (shown in (3)), while keeping the value of $ageCoef$ constant at the RPR default setting of 4, we run a sequence of simulations, increasing the link-delay for each iteration. The obtained throughput-data are subsequently processed to find the time it takes for the fairness algorithm to reach a stable state. Based on [15], we define a state as stable when no observed values deviate more than 5% from the mean value over a 50 ms time interval. Formally, we define our system to be stable at time t_0, if all values sampled in the interval $\langle t_0, t_0 + t_s \rangle$ ($t_s = 50ms$) are within $\pm 5\%$ of the mean value in the same interval:

$$\begin{aligned} max(X(t)) &< \overline{X(t)} * 1.05, \quad t_0 < t < t_0 + t_s \\ min(X(t)) &> \overline{X(t)} * 0.95, \quad t_0 < t < t_0 + t_s \end{aligned} \tag{19}$$

We use the "hot"-receiver scenario described above, and measure the time it takes from a link (the link between nodes 3 and 4 in Fig. 5 above) becomes congested, to the system has stabilized at the fair division of sending rate for each source.

For each allowed value of the *lpCoef* parameter, we plot the convergence-time required to reach the stable state as defined in (19) above. The variable in the plot is the aggregate of link propagation-delays between the *head* and *tail* node and back. The results are shown in Fig. 7 below. The abscissa value when the stabilization time becomes infinite is the point where, for the corresponding system time-constant and $lpCoef$ setting, the system no longer satisfies the stability criterion of (19) above.

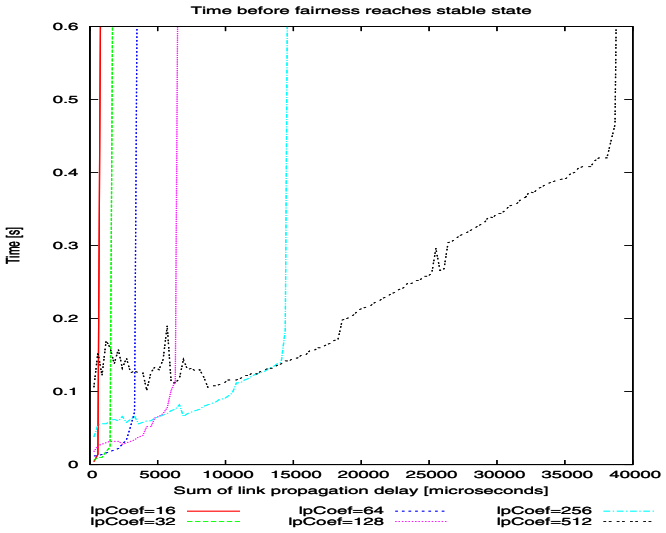


Fig. 7. The time it takes before the fairness algorithm stabilizes at the fair rate for each node sending over a congested link for varying link lengths and *lpCoef* settings.

As seen in the figure, the point where the system (for the various $lpCoef$) settings no longer converges are lower than those of the analytical expression in (18) ($lpCoef$=16: 1.6ms, $lpCoef$=32: 3.2ms, $lpCoef$=64: 6.4ms, $lpCoef$=128: 12.8ms, $lpCoef$=256: 25.6ms, $lpCoef$=512: 51.2ms). As discussed for the previous simulation experiment, we believe that the major cause of this is the scheduling and queuing delays on the path between *head* and *tail*.

V. Related Work

Analytical performance evaluation of a complex system like an RPR-network is extremely hard to perform precisely. Several papers are published for buffer insertion rings presenting analytical performance evaluation based on queueing theory

[17], [18], [19]. Similarly, we can find analytical studies of RPR based on queueing theory [20]. Other relevant work in the area of buffer insertion rings is the work presented in [21]. In this paper, Scott et al. studied the performance of the buffer insertion ring SCI analytically, but did not include the Fairness protocol. For RPR in general, several papers have been published, studying different performance aspects. In [22], Huang et al. present a thorough analysis of ring access delays for RPR-networks with nodes that contain a single transit queue. In [2], Gambiroza et al. focus on the operation of the RPR fairness algorithm and their alternative proposal, DVSR, and its ability, for some given load scenarios to converge at the fair division of rates according to their RIAS fairness reference model. More general fairness in insertion buffer rings have been studied by several [23], [24], [25], [26].

VI. Conclusion

When the demand for bandwidth in a Resilient Packet Ring is higher than the supply, the fairness algorithm running in the most congested node (the *head* of a *congestion domain*) is responsible for calculating the fair rate that all nodes sending over the congested link, must adhere to. In this paper we have developed a discrete control system model of RPR's *aggressive* mode fairness algorithm rate control mechanism running in *congestion domain* heads. We are not aware of any other contribution that evaluate the performance of RPR-networks using control systems theoretic approaches. We have discussed the model and deduced some of its basic properties. We have argued that the time-constant, τ, of the rate-controller must not be larger than the time-constant of the system it controls, which consists of all nodes contributing to the congestion (the *congestion domain*). Based on this we have shown that the setting of some important system parameters, $ageCoef$ and $lpCoef$, can be analytically estimated based on the size of the controlled system (the size of the *congestion domain*). This new insight makes the configuration of an RPR-network sounder, and less prone to configuration errors. Finally, we have supported our analytical findings by simulations that indicate that our analytical model yields a reasonable upper bound for the convergence of the Resilient Packet Ring *aggressive* mode fairness algorithm.

In future work, it would be interesting to extend our control system model to include all the nodes that contribute to the congestion. It would also be interesting to model the fairness algorithm even more precisely, in order to find even closer correspondence between the model and the simulated results.

VII. Acknowledgements

We want to thank Nils Damm Christophersen for providing valuable insights to the analytical section of this paper.

References

[1] IEEE Computer Society, "IEEE Std 802.17-2004," September 24 2004.

[2] 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 Trans. Networking*, vol. 12, no. 1, pp. 85–102, 2004.

[3] E. Hafner, Z. Nendal, and M. Tschanz, "A Digital Loop Communication System," *IEEE Trans. Commun.*, vol. 22, no. 6, pp. 877 – 881, June 1974.

[4] C. C. Reames and M. T. Liu, "A Loop Network for Simultaneous Transmission of Variable-length Messages," in *Proceedings of the 2nd Annual Symposium on Computer Architecture*, vol. 3, December 1974.

[5] "IEEE Standard 1596-1990," IEEE Standard for Scalable Coherent Interface (SCI).

[6] H. R. van As, W. W. Lemppenau, P. Zafiropulo, and E. Zurfluh, "CRMA-II: A Gbit/s MAC Protocol for Ring and Bus Networks with Immediate Access Capability," in *Proceedings of the Ninth Annual European Fibre Optic and Local Area Networks Conference (EFOC/LAN'91)*, London, June 1991, pp. 262 – 277.

[7] W. W. Lemppenau, H. R. van As, and H. R. Schindler, "Prototyping a 2.4 Gbit/s CRMA-II dual-ring ATM LAN and MAN," in *Proceedings of the 6th IEEE Workshop on Local and Metropolitan Area Networks*, 1993, pp. 17 – 18.

[8] I. Cidon and Y. Ofek, "MetaRing - A Full Duplex Ring with Fairness and Spatial Reuse," *IEEE Trans. Commun.*, vol. 41, no. 1, pp. 110 – 120, January 1993.

[9] R. Needham and A. Herbert, *The Cambridge Distributed Computing System*. London: Addison-Wesley, 1982.

[10] "ISO/IECJTC1SC6 N7873," January 1993, specification of the ATMR Protocol (V.2.0).

[11] "IEEE Standard 802.5-1989," IEEE Standard for Token Ring.

[12] F. E. Ross, "An Overview of FDDI: the Fiber Distributed Data Interface," *IEEE J. Select. Areas Commun.*, vol. 7, no. 7, pp. 1043 – 1051, September 1989.

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

[14] F. Davik and S. Gjessing, "The Stability of the Resilient Packet Ring Aggressive Fairness Algorithm," in *Proceedings of the 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN2004)*, Mill Valley, CA, April 25-28 2004, pp. 17–22.

[15] C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2nd ed. Prentice-Hall, 1991.

[16] "OPNET Modeler. http://www.opnet.com." [Online]. Available: http://www.opnet.com/

[17] M. T. Liu, G. Babic, and R. Pardo, "Traffic analysis of the distributed loop computer network (DLCN)," in *Proc. Nat. Telecommun. Conf.*, December 1997.

[18] A. Thomasian and H. Kanakia, "Performance Study of loop networks using buffer insertion," *Comput. Networks*, vol. 3, pp. 419–425, December 1979.

[19] W. Bux and M. Schlatter, "An Approximate Method for the Performace Analysis of Buffer Insertion Rings," *IEEE Trans. Commun.*, vol. 1, January 1983.

[20] P. Yue, Z. Liu, and J. Liu, "High performance fair bandwidth allocation algorithm for resilient packet ring," in *Proceedings of the 17th International Conference on Advanced Information Networking and Applications*, March 27-29 2003, pp. 415 – 420.

[21] S. L. Scott, J. R. Goodman, and M. K. Vernon, "Analysis of the SCI Ring," University of Wisconsin, Madison, Tech. Rep. 1055, November 1991.

[22] C. Huang, H. Peng, F. Yuan, and J. Hawkins, "A steady state bound for resilient packet rings," in *Global Telecommunications Conference, (GLOBECOM '03)*, vol. 7. IEEE, December 2003, pp. 4054–4058.

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

[24] I. Kessler and A. Krishna, "On the cost of fairness in ring networks," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 306–313, 1993.

[25] D. Picker and R. Fellman, "Enhancing SCI's fairness protocol for increased throughput," in *IEEE Int. Conf. On Network Protocols*, October 1993.

[26] J. Schuringa, G. Remsak, and H. R. van As, "Cyclic Queuing Multiple Access (CQMA) for RPR Networks," in *Proceedings of the 7th European Conference on Networks & Optical Communications (NOC2002)*, Darmstadt, Germany, June 2002, pp. 285 – 292.