The IEEE 802.17 Media Access Protocol for High-Speed Metropolitan-Area Resilient Packet Rings

V. Gambiroza, P. Yuan, and E. Knightly

***Abstract*— The Resilient Packet Ring (RPR) IEEE 802.17 standard is under development as a new high-speed technology for metropolitan backbone networks. A key performance objective of RPR is to simultaneously achieve high utilization, spatial reuse, and fairness, an objective not achieved by current technologies such as SONET and Gigabit Ethernet nor by legacy ring technologies such as FDDI. The core technical challenge for RPR is the design of a fairness algorithm that dynamically throttles traffic to achieve these properties. The difficulty is in the distributed nature of the problem, that upstream ring nodes must inject traffic at a rate according to congestion and fairness criteria downstream. This article provides an overview of the RPR protocol with a focus on medium access and fairness.**

I. Introduction

Rings are the dominant topology in metropolitan backbones primarily for their protection properties, that is, even under a link failure, full connectivity among all ring nodes is maintained. Moreover, rings have reduced deployment costs as compared to star or mesh topologies as ring nodes are only connected to their two nearest neighbors vs. to a centralized point (star) or multiple points (mesh). Unfortunately, current technology choices for high-speed metropolitan rings provide a number of unsatisfactory alternatives. A SONET ring can ensure minimum bandwidths (and hence fairness) between any pair of nodes. However, use of static circuits prohibits unused bandwidth from being reclaimed by other flows and results in low utilization. On the other hand, a Gigabit Ethernet (GigE) ring can provide full statistical multiplexing, but suffers from poor utilization and unfairness. Low utilization arises because the Ethernet spanning tree protocol [12] requires that one link be disabled to preclude "loops," thereby preventing traffic from being forwarded along the true shortest path to the destination. Unfairness occurs in GigE in the topology of Figure 1 for example, in which nodes will obtain different throughputs to the core or hub node depending on their spatial location on the ring and input traffic patterns. Finally, legacy ring technologies such as FDDI do not employ spatial reuse. That is, by using a rotating token such that a node must have the token to transmit, only one node can transmit at a time.

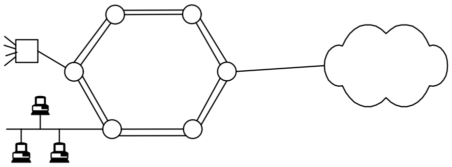


Fig. 1. Illustration of Bidirectional Packet Ring

The IEEE 802.17 Resilient Packet Ring (RPR) working group was formed in early 2000 to develop a standard for bi-directional packet-switched metropolitan rings. Unlike FDDI (as well as token ring, and DQDB), the protocol supports destination packet removal so that a packet will not traverse all ring nodes and spatial reuse can be achieved. However, allowing spatial reuse introduces a challenge to ensure fairness among different nodes competing for ring bandwidth. Consequently, a key performance objective of RPR is to simultaneously achieve high utilization, spatial reuse, and fairness.

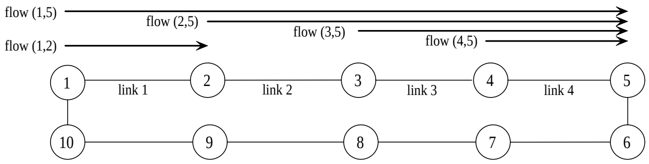


Fig. 2. Parallel Parking Lot Scenario

To illustrate spatial reuse and fairness, consider the depicted scenario in Figure 2 in which four infinite demand flows share link 4 in route to destination node 5. In this "parallel parking lot" example, each of these flows should receive 1/4 of the link bandwidth. Moreover, to fully exploit spatial reuse, flow (1,2) should receive all excess capacity on link 1, which is 3/4 due to the downstream congestion.

The key technical challenge of RPR is design of the bandwidth allocation algorithm that can dynamically achieve such rates, namely the fairness algorithm. Note that to realize this goal, some coordination among nodes is required. For example, if each node performs weighted fair queueing [12], a local operation without coordination among nodes, flows (1,2) and (1,5) would obtain equal bandwidth shares at node 1 so that flow (1,2) would receive a net bandwidth of 1/2 vs. the desired 3/4. Thus, RPR algorithms must throttle traffic at ingress points based on downstream traffic conditions to achieve these rate allocations.

In this article, we describe the RPR fairness algorithm as defined in the IEEE 802.17 protocol [8]. We present the algorithm's design objectives, describe the RPR node architecture, outline the key components of the fairness algorithm, and present a set of simulation results regarding fairness, throughput, and convergence times.

The authors are with the ECE/CS Departments at Rice University, Houston, TX. URL http://www.ece.rice.edu/networks.

II. Overview of Ring Technologies

In this section, we present the design objectives of the Resilient Packet Ring fairness algorithm in the context of legacy technologies, GigE and SONET.

A. Legacy Ring Technologies

Obsolete technologies such as FDDI, token ring and DQDB allow only one packet to transmit at a time, and do not support destination packet removal. Thus, each packet circulates the entire ring before being removed. As a consequence, spatial reuse cannot be achieved.

Achieving spatial reuse in rings was widely studied in the context of generalizing token ring protocols (see [7], [16] and the references therein). A notable example is the MetaRing protocol [3], which we briefly describe as follows. MetaRing attained spatial reuse by replacing the traditional token of token rings with a 'SAT' (satisfied) message designed so that each node has an opportunity to transmit the same number of packets in a SAT rotation time. While providing significant throughput gains over token rings, the coarse granularity of control provided by holding a SAT signal limits such a technique's applicability to RPR. For example, the protocol's fairness properties were found to be highly dependent on the parameters as well as the input traffic patterns [1]; the SAT rotation time is dominated by the worst case link prohibiting full spatial reuse; etc.

B. Gigabit Ethernet

Gigabit Ethernet rings are characterized by low cost, ease of manageability, and simple integration with existing equipment. Commercial implementations also allow priority classes which service providers can employ to support applications such as voice over IP (e.g., [14]).

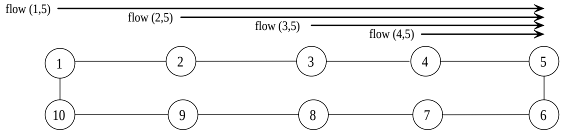


Fig. 3. Parking Lot Scenario

However GigE rings have four limitations. First, GigE lacks a distributed fairness algorithm which can result in undesirable distributions of bandwidth among nodes. For example, consider the "Parking Lot" scenario of Figure 3 in which all flows have traffic demands given by the full link capacity normalized to 1. In this case, the desired bandwidth allocations are 0.25 to each flow. For the case of GigE, at node 4, flow (4,5) has demand 1 as do flows (1,5) through (3,5) in *aggregate*. Because GigE services traffic locally in FIFO order, flow (4,5) will obtain 0.5 throughput and aggregated flows (1,5) through (3,5) will obtain 0.5 throughput. Therefore, GigE will achieve bandwidth allocations of .125, .125, .25 and .5 to respective flows (1,5) through (4,5). Thus, without any distributed bandwidth allocation algorithm, flows obtain throughputs which are not fair.

Second, the spanning tree protocol prohibits loops in Ethernet networks. Consequently, one link must be effectively disabled to ensure a loop-free network. In Figure 3, if the disabled link is between nodes 5 and 6, then traffic originating from node 6 and destined to node 5 must traverse all ring nodes vs. being forwarded on a 1-hop path.

Third, when a link or node fails, an Ethernet Ring requires re-computation of the spanning tree. As the spanning tree protocol can take 100's of milliseconds to seconds to converge, faults can cause significant application disruptions. A further impact of slow recovery can occur if the fault is perceived by the layer three routing protocol. In such cases, BGP and OSPF routes may be recomputed causing potentially long-time-scale disruptions that span regions well beyond the metro area. For example, a fault detected by BGP may cause an Autonomous System to switch from its primary to secondary metro service provider. Such changes can have vast and unpredictable effects on performance and link loads.

Finally, while GigE can provide simple traffic prioritization rules, it does not have mechanisms for providing guaranteed bandwidth, delay, and delay jitter, as do alternate technologies such as SONET and RPR.

C. SONET

SONET rings provide point-to-point circuits among ring nodes. With a dedicated circuit, SONET provides guaranteed bandwidth, delay, and delay jitter. Moreover, SONET provides fast recovery from faults typically in the 10's of msec range, thereby making link failures nearly transparent to most applications and higher layers.

The primary limitation of SONET rings is bandwidth inefficiency. If all nodes require circuits to all other nodes, then an N-node ring suffers from the "N-squared" problem as each node requires $N-1$ circuits. Even for moderate values of N, the total ring capacity is quickly exhausted. In a realistic setting, nodes will communicate with a subset of nodes representing Internet Data Centers, gateways to the Internet backbone, multiple corporate sites, etc. somewhat mitigating this effect. Regardless, static circuits preclude nodes from using unused bandwidth on other links and from bursting beyond their circuit rate. Thus, overall, circuit-switched rings must operate at considerably lower utilization than packet-switched rings.

D. RPR

Like Ethernet but unlike legacy technologies, RPR supports destination packet removal thereby enabling spatial reuse. To exploit bandwidth efficiencies of spatial reuse while simultaneously ensuring fairness, RPR employs a distributed fairness algorithm as described below. Unlike Ethernet, RPR does not employ the spanning tree protocol as ring topologies purposefully have "loops" which spanning tree is designed to avoid. Consequently, packets on RPR rings are forwarded on the shortest-hop-count path to the ring destination node.

Second, the "resilience" aspect of RPR ensures that if any single node or bi-directional link fails, an alternate path is constructed within 50 msec. In particular, RPR's dual ring topology allows traffic to be forwarded among all non-failed stations after a link or node failure by steering traffic away from the failure. For example, considering Figure 3, RPR's primary path between nodes 4 and 5 is the one-hop path. However, if the link between nodes 4 and 5 fails, nodes 4 and 5 will communicate via the path 4-3-2-1-10-9-8-7-6-5.

Finally, RPR defines multiple traffic classes. Class A provides a circuit-like transport between two ring nodes with a guaranteed data rate, low end-to-end delay, and jitter bounds. Thus, Class A supports performance guarantees similar to that of SONET, yet without the coarse bandwidth granularity imposed by the SONET hierarchy (OC-3, OC-12, etc.).

Like Class A, Class B also provides a committed bandwidth guarantee, but also allows clients to burst above the committed rate, with excess traffic treated fairly among all competing non-guaranteed traffic. This overcomes a limit of true circuits (e.g., SONET) that prevents unused bandwidth from being reclaimed by other nodes or traffic classes.

Finally, Class C provides a best effort service with no allocated or guaranteed rate and no bounds on end-to-end delay and jitter. Class C traffic is opportunistic in that it reclaims unused bandwidth from Class B. Achieving fairness for Class C traffic and excess Class B traffic is an important challenge for RPR that is addressed by the RPR fairness algorithm described below.

III. RPR Fairness and Spatial Reuse

The goal of the RPR fairness algorithm is to simultaneously achieve fairness, high utilization, and spatial reuse via a distributed and dynamic bandwidth allocation algorithm.

A. Parking Lot Scenarios

Fairness alone is most easily illustrated with the classical "parking lot" scenario of Figure 3. In this example, node 5 represents a gateway to a core or hub node, and nodes 1 - 4 connect access networks. A metro service provider desires to provide equal bandwidth allocations to each of its customers accessing via different (perhaps multiple) nodes. Thus, RPR targets to achieve bandwidth allocations of 1/4 of the link capacity to each of the depicted 4 flows.

To fully achieve spatial reuse, excess bandwidth on the ring must be fully utilized provided that the fairness constraints are satisfied. A simple example of spatial reuse is presented in the Parallel Parking Lot scenario of Figure 2 which contains a single additional flow between nodes 1 and 2. In this case, this one-hop flow between nodes 1 and 2 should obtain throughput .75 since that capacity would otherwise be unused given the bottleneck link between nodes 4 and 5.

Achieving spatial reuse together with fairness introduces new challenges in protocol design. Namely, with destination packet removal and without any token mechanisms, any pair of nodes can potentially communicate at any rate up to link saturation limits. Thus, the goal of the RPR fairness algorithm is to provide a distributed protocol to throttle flows at their ring-ingress points to their ring-wide fair rates. Thus, in the above example of the Parallel Parking Lot, the flow between nodes 1 and 5 must be throttled at node 1 to rate .25 such that the flow between nodes 1 and 2 can attain rate .75.

B. RIAS Reference Model

In general, these ideal allocations can be described generally via the RIAS (Ring Ingress-Aggregated with Spatial reuse) fair reference model. The RIAS reference model as introduced in [10], formally defined in [6], and is now incorporated into IEEE 802.17 standard's targeted performance objective [8]. RIAS Fairness has two key components. The first component defines the level of traffic granularity for fairness determination at a link as an ingress-aggregated (IA) flow, i.e., the aggregate of all flows originating from a given ingress node. The targeted service model of packet rings justifies this: to provide fair and/or guaranteed bandwidth to the networks and backbones that it interconnects. Thus, the RIAS reference model ensures that an ingress node's traffic receives an equal share of bandwidth on each link relative to other ingress nodes' traffic on that link. The second component of RIAS fairness ensures maximal spatial reuse subject to this first constraint. That is, bandwidth can be reclaimed by IA flows when it is unused either due to lack of demand or in cases of sufficient demand in which flows are bottlenecked elsewhere.

A final example of RPR and RIAS fairness contrasts it with other fairness objectives typically employed in the Internet via TCP or ATM protocols. TCP provides bandwidth allocations that are approximately *proportional fair* in that a flow's throughput is inversely proportional to its round-trip time (see [9], [13]). One interpretation of proportional fairness is that a flow's bandwidth is scaled to resources consumed. For example, in the parking lot scenario, the flow traversing four links consumes four times the resources of the one-hop flow, and therefore receives a lower bandwidth allocation. Thus, the proportional fair allocations for the parking lot example are .12, .16, .24, and .48 from left to right.

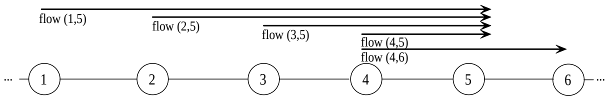


Fig. 4. Two-Exit Parking Lot Scenario

Similarly, flow-based max-min fairness [2], widely studied in the context of ATM networks (e.g., [11], [15]) and elsewhere, also differs from RIAS fairness. This is best illustrated in the "two exit" scenario of Figure 4. Here, the RIAS fair rates of the flows originating from nodes 1, 2, and 3 are still .25. However, ingress node 4 now has two flows on bottleneck link 4 and must divide its ingress-aggregated link fair rate of .25 among these two flows such that each receives rate .125. In contrast, flow based max-min fair allocation would give all 5 flows a rate of 0.2.

In summary, as each node of a ring connects one or more clients, packet rings have a unique fairness requirement that each *ingress* node's traffic receives a fair bandwidth share on each link for which it demands traffic. Thus, RPR has a new fairness objective (RIAS) of providing fair bandwidth to ingress nodes and without bias to spatial location. Consequently, RPR has a new fairness algorithm designed to achieve these objectives.

IV. RPR Fairness Algorithm

In this section, we describe the basic operation of the Resilient Packet Ring (RPR) fairness algorithm [8]. Due to space constraints, our description necessarily omits many details and focuses on the key mechanisms for bandwidth arbitration. Readers are referred to the standards documents for full details and pseudocode.

Throughout, we consider committed rate (Class B) and best effort (Class C) traffic classes in which each node obtains a minimum bandwidth share (zero for Class C) and reclaims unused bandwidth in a weighted fair manner. We omit discussion of Class A traffic which has guaranteed rate and jitter, as other nodes are prohibited from reclaiming unused Class A bandwidth.

A. RPR Node Architecture

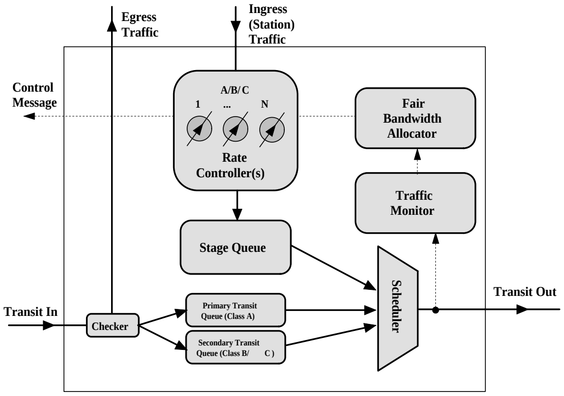


Fig. 5. Generic RPR Node Architecture

The architecture of a generic RPR node is illustrated in Figure 5. First, observe that all station traffic entering the ring is first throttled by rate controllers. In the example of the Parallel Parking Lot, it is clear that to fully achieve spatial reuse, flow (1,5) must be throttled to rate 1/4 at its ring ingress point. Second, these rate controllers are at a destination-based granularity. Traffic is divided into two categories based on whether it's going over the congested link or not, i.e., if a single link is congested, an ingress node only throttles its traffic forwarded over that link. RPR also supports a type of virtual output queueing analogous to that performed in switches to avoid head-of-line blocking.

Next, RPR nodes have measurement modules (byte counters) to measure serviced station traffic and transit traffic. These measurements are used by the fairness algorithm to compute a feedback control signal to throttle upstream nodes to the desired rates. Nodes that receive a control message use the information in the message, perhaps together with local information, to set the bandwidths for the rate controllers.

The final component is the scheduling algorithm that arbitrates service among station and transit traffic. In *single-queue mode*, the transit path consists of a single FIFO queue referred to as the Primary Transit Queue (PTQ). In this case, the scheduler employs strict priority of transit traffic over station traffic. In *dual-queue mode*, there are two transit path queues, one for guaranteed Class A traffic (PTQ), and the other for Class B and C traffic, called Secondary Transit Queue (STQ). In this mode, the scheduler always services Class A transit traffic first from PTQ. Class A station traffic will be served right after PTQ if STQ is not full. Otherwise, the scheduler serves STQ traffic in advance to ensure a lossless transit path. For fairness eligible traffic, i.e., excess Class B and Class C traffic, the scheduler employs round-robin service among the transit traffic in STQ and the station Class B and C traffic until a buffer threshold is reached for STQ. If STQ reaches the buffer threshold, STQ transit traffic is always selected over station traffic for fairness consideration.

In both cases, the objective is to ensure hardware simplicity (for example, avoiding expensive per-flow or per-ingress queues on the transit path) and to ensure that the transit path is lossless, i.e., once a packet is injected into the ring, it will not be dropped at a downstream node.

B. RPR Fairness Algorithm

There are two modes of operation for the RPR fairness algorithm. The first, termed Aggressive Mode (AM), evolved from the Spatial Reuse Protocol (SRP) [17] currently deployed in a number of operational metro networks. The second, termed Conservative Mode (CM), evolved from the Aladdin algorithm [4]. Both modes operate within the same framework described as follows. A congested downstream node conveys its congestion state to upstream nodes such that they will throttle their traffic and ensure that there is sufficient spare capacity for the downstream station traffic. To achieve this, a congested node transmits its local fair rate upstream, and all upstream nodes sending to the link must throttle to this same rate. After a convergence period, congestion is alleviated once all nodes' rates are set to the fair rate. Likewise, when congestion clears, stations periodically increase their sending rates to ensure that they are receiving their maximal bandwidth share.

There are two key measurements for RPR's bandwidth control, *forward_rate* and *add_rate*. The former represents the service rate of all transit traffic and the latter represents the rate of all serviced station traffic. Both are measured as byte counts over a fixed interval length *aging_ interval*. Moreover, both measurements are low-pass-filtered using exponential averaging with parameter 1/*LPCOEF* given to the current measurement and 1-1/*LPCOEF* given to the previous average. In both cases, it is important that the rates are measured at the output of the scheduler so that they represent serviced rates rather than offered rates.

At each *aging_ interval*, every node checks its congestion status based on conditions specific to the mode AM or CM. When node n is congested, it calculates its *local_ fair_rate*$[n]$, which is the fair rate that an ingress-based flow can transmit to node n. Node n then transmits a fairness control message to its upstream neighbor that contains *local_ fair_rate*$[n]$.

If upstream node $(n-1)$ receiving the congestion message from node n is also congested, it will propagate the message upstream using the minimum of the received *local_ fair_rate*$[n]$ and its own *local_ fair_rate*$[n-1]$. The objective is to inform upstream nodes of the minimum rate they can send along the path to the destination. If node $(n-1)$ is not congested but its *forward_rate* is greater than the received *local_ fair_rate*$[n]$, it forwards the fairness control message containing *local_ fair_rate*$[n]$ upstream, as this situation indicates that the congestion is due to transit traffic from further upstream. Otherwise, a null-value fairness control message is transmitted to indicate a lack of congestion.

When an upstream node i receives a fairness control message advertising *local_ fair_rate*$[n]$, it reduces its rate limiter value for flows going over the congested link, termed *allowed_rate_congested*, which is the sum of allowed service rate of flow (i, j), for all values of j, such that n lies on the path from i to j. The objective is to have upstream nodes throttle their own station rate controller values to the minimum rate it can send along the path to the destination. Consequently, station traffic rates will not exceed the advertised *local_ fair_rate* value of any congested node in the downstream path of a flow. Otherwise, if a null-value fairness control message is received, it increments *allowed_rate_congested* by a fixed value such that it can reclaim additional bandwidth if one of the downstream flows reduces its rate. Moreover, such rate increases are essential for convergence to fair rates even in cases of static demand.

The main differences between AM and CM are congestion detection and calculation of the local fair rate which we discuss below. Moreover, by default AM employs *dual-queue mode* and CM employs *single-queue mode*.

C. Aggressive Mode (AM)

Aggressive Mode is the default mode of operation of the RPR fairness algorithm and its logic is as follows. An AM node n is said to be congested whenever

$$\textit{STQ\_depth}[n] > \text{ low\_threshold}$$

or

$$\textit{forward\_rate}[n] + \textit{add\_rate}[n] > \text{ unreserved\_rate},$$

where as above, *STQ* is the transit queue for Class B and C traffic. The threshold value *low_threshold* is a fraction of the transit queue size with a default value of 1/8 of the *STQ* size.[1]

When a node is congested, it calculates its *local_ fair_rate* as the normalized service rate of its own station traffic, *add_rate*, and then transmits a fairness control message containing *add_rate* to upstream nodes.

Considering the parking lot example in Figure 3, if a downstream node advertises *add_rate* below the true fair rate (which does indeed occur before convergence), all upstream nodes will throttle to this lower rate; in this case, downstream nodes will later become uncongested so that flows will increase their *allowed_rate*. This process will then oscillate more and more closely around the targeted fair rates for this example.

D. Conservative Mode (CM)

Each CM node has an access timer measuring the time between two consecutive transmissions of station packets. As CM employs strict priority of transit traffic over station traffic via single queue mode, this timer is used to ensure that station traffic is not starved. Thus, a CM node n is said to be congested if the access timer for station traffic expires or if

$$\textit{forward\_rate}[n] + \textit{add\_rate}[n] > \text{ low\_threshold.}$$

Unlike AM, *low_threshold* for CM is a rate-based parameter that is a fixed value less than the link capacity, 0.8 of the link capacity by default. In addition to measuring *forward_rate* and *add_rate*, by searching the header for the ingress node ID of each packet, a CM node also measures the number of *active* stations that have had at least one packet served in the past *aging_ interval*.

If a CM node is congested in the current *aging_ interval*, but was not congested in the previous one, the *local_ fair_rate* is computed as the total unreserved rate divided by the number of *active* stations. If the node is continuously congested, then *local_ fair_rate* depends on the sum of *forward_rate* and *add_rate*. If this sum is less than *low_threshold*, indicating that the link is under utilized, *local_ fair_rate* ramps up. If this sum is above *high_threshold*, a fixed parameter with a default value that is 0.95 of the link capacity, *local_ fair_rate* will ramp down.

Again considering the parking lot example in Figure 3, when the link between nodes 4 and 5 is first congested, node 4 propagates rate 1/4, the true fair rate. At this point, the link will still be considered congested because its total rate is greater than *low_threshold*. Moreover, because the total rate is also greater than *high_threshold*, *local_ fair_rate* will ramp down periodically until the sum of *add_rate* and *forward_rate* at node 4 is less than *high_threshold* but greater than *low_threshold*. Thus, for CM, the maximum utilization of the link will be *high_threshold*, hence the name "conservative."

V. SIMULATION EXPERIMENTS

In this section, we use simulations to study the performance of RPR and provide comparisons with a Gigabit Ethernet (GigE) Ring that has no distributed bandwidth control algorithm and simply services arriving packets in first-in first-out order

We first study RPR and GigE in the context of the basic RIAS goals of achieving spatial reuse and fairness for both open-loop UDP flows and closed-loop TCP flows. Second, we study RPR convergence times and associated temporal dynamics under step-function traffic inputs. Finally, we address cases of unbalanced traffic that can cause oscillations and throughput degradation in RPR.

All simulation results are obtained with our publicly available *ns-2* implementation of RPR.[2] Unless otherwise specified, we consider 622 Mbps links (OC-12), 200 KB STQ buffer size, 1 kB packet size, and 0.1 msec link propagation delay between each pair of nodes.

[1] *unreserved_rate* is the link capacity minus the reserved rate for guaranteed traffic. As we consider only best-effort traffic, *unreserved_rate* is the link capacity in the rest of this paper.

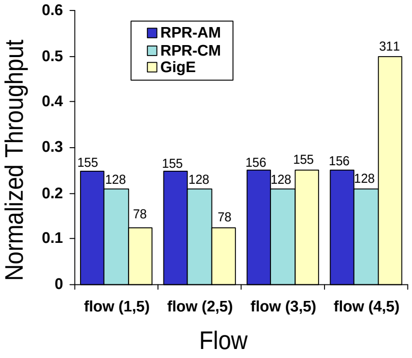


Fig. 6. Throughput in the Parking Lot

A. *Fairness and Spatial Reuse*

1) Fairness in the Parking Lot: We first consider the parking lot scenario with a ten-node ring as depicted in Figure 3 and widely studied in the IEEE 802.17 standardization process. Four constant-rate UDP flows (1,5), (2,5), (3,5), and (4,5) each transmit at an offered traffic rate of 622 Mbps, and we measure each flow's throughput at node 5. We perform the experiment with RPR and GigE (for comparison, the GigE link rate is set to 622 Mbps) and present the results in Figure 6. The figure depicts the average normalized throughput for each flow over the 5 second simulation, i.e., the total received traffic at node 5 divided by the simulation time. The labels above the bars represent the un-normalized throughput in Mbps.

We make the following observations about the figure. First, RPR-AM achieves the correct RIAS fair rates (622/4) to within ±1%. In contrast, without a distributed bandwidth control mechanism, GigE fails to ensure fairness, with flow (4,5) obtaining 50% throughput share whereas flow (1,5) obtains 12.5%. Next, observe that RPR-CM achieves throughputs of 128 Mbps, 82% of the ideal value, illustrating the throughput penalty of CM control. Finally, other experiments (not shown) with Pareto on-off flows with various parameters yield nearly identical average throughputs.

B. *TCP Traffic*

1) Inter-node Performance Isolation: Unfairness among congestion-responsive TCP flows and non-responsive UDP flows is well established. However, suppose one ingress node transmits only TCP traffic whereas all other ingress nodes send high rate UDP traffic. The question is whether RPR can still provide RIAS fair bandwidth allocation to the node with TCP traffic, i.e., can RPR provide inter-node performance isolation? The key issue is whether RPR's reclaiming of unused capacity to achieve spatial reuse will hinder the throughput of the TCP traffic.

To answer this question, we consider the same parking lot topology of Figure 3 and replace flow (1,5) with a single long-lived TCP Reno flow (representing a large file transfer for example). The remaining three flows are each constant rate UDP flows with rate 0.3 (186.6 Mbps). Simulation results indicate that the single TCP flow (1,5) is still able to obtain its full RIAS share of 155 Mbps. This is easily achieved provided that the MAC client buffer is sufficiently large to avoid dropping the TCP traffic before it enters the ring. As described previously, once packets enter the ring, they are not dropped downstream.

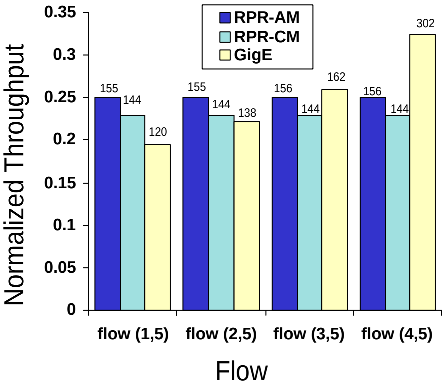


Fig. 7. Throughput for TCP Micro-Flows

2) RIAS vs. Proportional Fairness for TCP Traffic: Next, we consider the case that each of the four flows in the parking lot is a single TCP micro-flow, and present the corresponding throughputs for RPR and GigE in Figure 7. As expected, with a GigE ring the flows with the fewest number of hops and lowest round trip time receive the largest bandwidth shares (cf. Section III). However, RPR-AM and RPR-CM eliminate such spatial bias and provide all ingress nodes with an equal bandwidth share, independent of their hop count and round trip time.

3) Spatial Reuse in the Parallel Parking Lot: We now consider the spatial reuse scenario of the Parallel Parking Lot (Figure 2) again with each flow offering traffic at the full link capacity. As described in Section III, the rates that achieve IA fairness while maximizing spatial reuse are 0.25 for all flows except flow (1,2) which should receive all excess capacity on link 1 and receive rate 0.75.

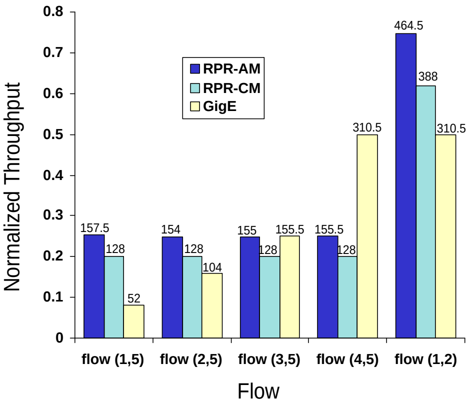


Fig. 8. Spatial Reuse in the Parallel Parking Lot

Figure 8 shows that RPR-AM achieves the targeted RIAS

[2] available at http://www.ece.rice.edu/networks/RPR

fair rates by exploiting per-destination queues and rate limiters at node 1. In contrast, as with the Parking Lot example, GigE favors downstream flows for the bottleneck link 4, and diverges significantly from the RIAS fair rates.

4) Convergence Time: In this experiment, we study the convergence time of the fairness algorithms using the parking lot topology and UDP flows with normalized rate 0.4 (248.8 Mbps). The flows' starting times are staggered such that flows (1,5), (2,5), (3,5), and (4,5) begin transmission at times 0, 0.1, 0.2, and 0.3 seconds respectively.

Figure 9 depicts the throughput over windows of duration 1 msec for RPR AM and CM. Observe that RPR-AM takes approximately 50 msec to converge whereas RPR-CM takes approximately 18 msec. Since RPR's control update interval is fixed to 0.1 msec, this corresponds to 500 and 180 control messages for AM and CM to converge respectively.

C. Oscillations Under Unbalanced Traffic

For our final experiments, we describe cases in which the RPR fairness algorithm is not able to converge even under constant-rate traffic inputs. In particular, for scenarios in which different flows have different RIAS rates, the RPR fairness algorithms can permanently oscillate. There are multiple adverse effects of such oscillations, including throughput degradation and increased delay jitter. The key issue is that the congestion signals *add_rate* for Aggressive Mode and (*link capacity/number of active stations*) for Conservative Mode do not always reflect the true fair rates and hence nodes oscillate in search of the correct fair rates. Precise mathematical conditions for oscillation are presented in [6]. Here, we present illustrative examples.

1) Aggressive Mode: Recall that without congestion, rates are increased until congestion occurs. In AM, once congestion occurs, the input rates of all nodes contributing traffic to the congested link are set to the minimum input rate. However, this minimum input rate is not necessarily the RIAS fair rate. Consequently, nodes over-throttle their traffic to rates below the RIAS rate. Subsequently, congestion will clear and nodes will ramp up their rates. Under certain conditions of unbalanced traffic, this oscillation cycle will continue permanently and lead to throughput degradation.

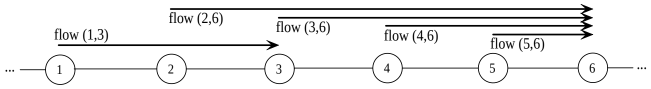


Fig. 10. Upstream Parallel Parking Lot

An illustrative example is presented in the "Upstream Parallel Parking Lot" scenario of Figure 10. In this case, under constant rate traffic inputs in which each flow demands the full link capacity, the four right-most flows will converge to their fair rate of 0.25 as in the Parking Lot. However, with RPR-AM, flow (1,3) will permanently oscillate between rates 0.25 and 0.75 as opposed to transmitting continuously at its RIAS fair rate of 0.75. This occurs because when flow (1,3) reaches its fair rate of 0.75, link 2-3 becomes congested. Upon congestion, node 2 transmits a message to node 1 containing node 2's add rate of 0.25 which forces node 1 to throttle flow (1,3) to that same rate. Subsequently, congestion clears and node 1 is able to gradually increase its rate upon receiving null congestion messages. Repeating the cycle, flow (1,3)'s throughput permanently oscillates between rates 0.25 and 0.75 as illustrated in Figure 11. This results in an average throughput degradation of 14% below the RIAS rate of 0.75.

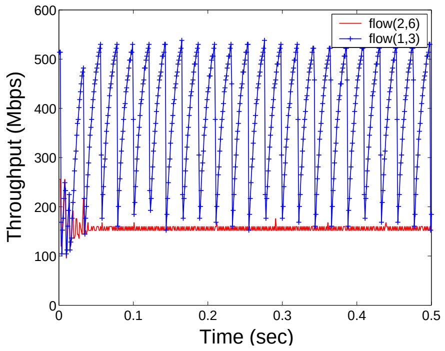


Fig. 11. Oscillation in the Upstream Parallel Parking Lot

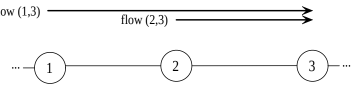


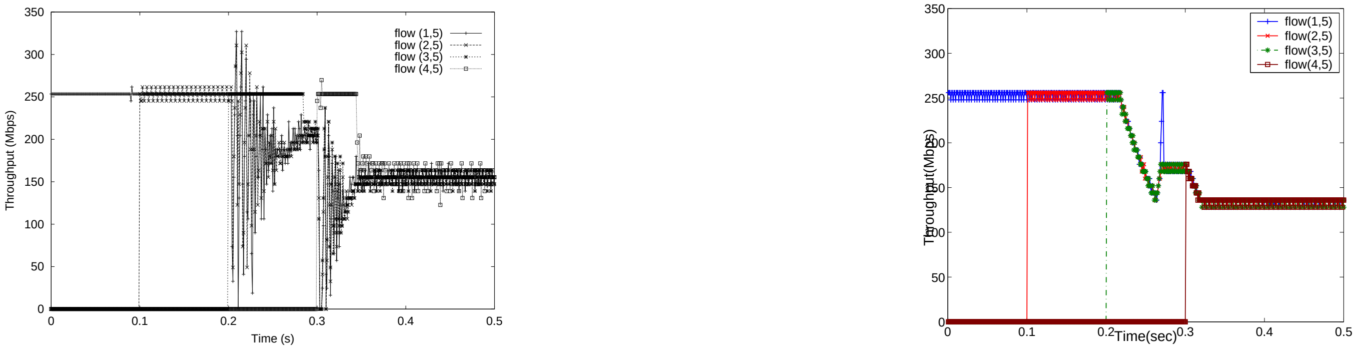
Fig. 12. Oscillation Scenario

The extent of the throughput degradation depends on the rate of the downstream flow. To explore this effect, we consider the two flow scenario of Figure 12 and vary the rate of flow (2,3) in different simulations and report the resulting throughput degradation of flow (1,3). In particular, Figure 13 depicts throughput loss of flow (1,3) vs. the downstream flow (2,3) rate. Observe that the throughput loss can be as high as 26% depending on the rate of the downstream flow. Moreover, observe that the throughput loss is non-monotonic. Namely, for downstream input rates that are very small, the upstream rate controller value drops dramatically but quickly recovers as there is little congestion downstream. For cases with higher rate downstream flows, the range of oscillation for the upstream rate controller is smaller, but the recovery to full rate is slower due to increased congestion. Finally, if the offered downstream rate is the fair rate (311 Mbps here), the system is "balanced" and no throughput degradation occurs.

Finally, we note that unbalanced traffic is also problematic for Conservative Mode. With CM, the advertised rate is determined by the number of *active* flows when a node first becomes congested for two consecutive *aging_intervals*. If a flow has even a single packet transmitted during the last *aging_interval*, it is considered *active*. Consequently, permanent oscillations also occur with low rate flows and unbalanced traffic. We refer interested readers to [6] for a complete discussion of oscillation in RPR-CM.

VI. Conclusions

In this paper, we presented an overview of the IEEE 802.17 Resilient Packet Ring Protocol. The key design goals of RPR



(a) RPR Aggressive Mode

(b) RPR Conservative Mode

Fig. 9. Algorithm Convergence Times

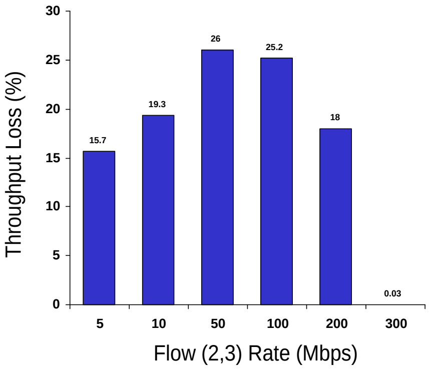


Fig. 13. RPR-AM Throughput Loss

are (1) fast recovery from faults, (2) guaranteed bandwidth, delay, and delay jitter, for class A and B traffic, and (3) spatial reuse, fairness, and high throughput for opportunistic class C traffic and excess class B traffic, as defined by the RIAS fairness reference model. We showed that in contrast to GigE, RPR is highly successful in achieving these design goals in many scenarios. Moreover, we showed that scenarios with unbalanced traffic provide an open problem for fairness algorithm design in packet rings as explored in [5], [6] for example. Such algorithms could potentially be invoked as new fairness-algorithm modes within the RPR framework.

VII. Acknowledgements

The authors are grateful to Necdet Uzun (Cisco) and Hui Zhang (Turin Networks and CMU) for insightful discussions on the RPR fairness algorithm.

References

[1] G. Anastasi and L. Lenzini. Performance evaluation of a MetaRing MAC protocol carrying asynchronous traffic. *Journal of High Speed Networks*, 6(1), 1997.

[2] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

[3] I. Cidon and Y. Ofek. Metaring - a full-duplex ring with fairness and spatial reuse. *IEEE Transactions on Communications*, 41(1):110–120, January 1993.

[4] A. Mekkittikul et al. Aladdin Proposal for IEEE Standard 802.17, Draft 1.0, November 2001.

[5] V. Gambiroza, Y. Liu, P. Yuan, and E. Knightly. High-performance fair bandwidth allocation for resilient packet rings. In *Proceedings of the 15th ITC Specialist Seminar*, Wurzburg, Germany, July 2002.

[6] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly. Design, Analysis, and Implementation of DVSR: A Fair, High Performance Protocol for Packet Rings. To appear in *IEEE/ACM Transactions on Networking*, 11(6), December 2003.

[7] L. Georgiadis, R. Guerin, and I. Cidon. Throughput properties of fair policies in ring networks. *IEEE/ACM Transactions on Networking*, 1(6):718–728, December 1993.

[8] IEEE. IEEE Standard 802.17: Resilient Packet Ring (Draft Version 1.1), October 2002. http://ieee802.org/17.

[9] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[10] E. Knightly, L. Balzano, V. Gambiroza, Y. Liu, S. Shaefor, P. Yuan, and H. Zhang. Achieving High-Performance with Darwin's Fairness Algorithm, IEEE 802.17 Standard Presentation, March 2002.

[11] H. T. Kung and R. Morris. Credit based flow control for ATM networks. *IEEE Network*, 9(2):40–48, March 1995.

[12] J. Kurose and K. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2nd edition, 2003.

[13] L. Massoulie and J. Roberts. Bandwidth sharing: objectives and algorithms. In *Proceedings of IEEE INFOCOM '99*, New York, NY, March 1999.

[14] Phonoscope Metro Service Provider. www.phonoscope.com, Houston, TX.

[15] C. Su, G. de Veciana, and J. Walrand. Explicit rate flow control for ABR services in ATM networks. *IEEE/ACM Transactions on Networking*, 8(3):350–361, June 2000.

[16] L. Tassiulas and J. Joung. Performance measures and scheduling policies in ring networks. *IEEE/ACM Transactions on Networking*, 4(5):576–584, October 1995.

[17] D. Tsiang and G. Suwala. The Cisco SRP MAC Layer Protocol, August 2000. Internet RFC 2892.