Congestion Control in Resilient Packet Rings

Dongmei Wang, K.K. Ramakrishnan, Charles Kalmanek,
Robert Doverspike, Aleksandra Smiljanić
AT&T Labs-Research, Florham Park, NJ 07932
{mei, kkrama, crk, rdd, aleks}@research.att.com

***Abstract*— Congestion control in ring based packet networks is challenging due to the fact that every node in the network runs both a rate adaptation algorithm, analogous to an endpoint algorithm in other network architectures, and a rate allocation algorithm, analogous to switch-based algorithms in other network architectures. This paper describes a congestion control algorithm for IEEE 802.17 Resilient Packet Rings called the Enhanced Conservative Mode algorithm that aims to avoid congestion and achieve a fair rate allocation for fairness eligible traffic in the case of a single bottleneck. We first present analysis to show that existing approaches for RPR congestion control (aggressive and conservative mode) have deficiencies. We present simulation results showing that the proposed enhanced conservative mode congestion control algorithm, is a significant improvement. In conjunction with other mechanisms specified in the IEEE 802.17 MAC, the proposed algorithm achieves high utilization on the ring with minimal starvation and oscillations, allows sources to fast start, and provides quality of service for multiple classes of service that require rate, delay and jitter guarantees.**

I. Introduction

Since a large and growing fraction of metropolitan access network traffic is packet data, there is a huge opportunity to achieve statistical multiplexing gains by integrating packet switching into metro access networks [1]. To support data traffic, the access network must support both bursty and delay-sensitive applications with dependable SLAs. The network must also support legacy private line TDM services, possibly via circuit emulation. Today, legacy services are supported using SONET ring technologies which provide fixed bandwidth pipes, offering stringent QoS guarantees, while not necessarily getting the benefit of statistical multiplexing. Migrating these services to emerging packet-based networks requires such networks to have good support for multiple classes of service.

There are several competing technologies for metro access networks: this paper focuses on the Resilient Packet Ring (RPR) [10] technology being standardized in IEEE 802.17. RPR is a Media Access Control (MAC) protocol designed for dual counter-rotating access rings that potentially replace traditional SONET rings. Nodes on an RPR ring transport frames from a source to a destination node by encapsulating the payload from the client (of the RPR MAC) with an RPR header. A recent tutorial on RPR appears in [6].

RPR is designed to support spatial re-use for unicast traffic, by using "destination-stripping" of frames, where the receiving node (also called a "station") removes the frame from the ring. Thus, the overall network capacity is increased as the path from the destination to the source is available for other traffic (similar in concept to "buffer insertion" rings). Transit traffic receives priority over locally sourced traffic at a station. However, if a station is already transmitting a packet, incoming transit traffic is buffered at a "transit queue". In this framework, when the network is congested, it is important that upstream stations do not starve downstream stations. It is also important that a node's throughput and delay performance not depend on its position on the ring, relative to the other active stations. This problem is somewhat similar to that observed previously in the IEEE 802.6 Distributed Queueing Dual Bus (DQDB) [9]. Thus, fairness and congestion control are important issues to address in an RPR ring.

Various fairness algorithms have been proposed for ring networks. In MetaRing [4], [5] fairness is provided by a token that circulates through the ring. In these schemes, when the number of credits per token visit is large enough to ensure high ring utilization, the ring access delay is increased. Several researchers have addressed fairness in a ring network with multiple congested links [2], [8], [12]. These schemes are significantly more complex than the RPR congestion control and fairness schemes studied here.

Congestion control in rings has several unique characteristics when compared with traditional congestion control schemes. Ring nodes act as both end-systems, which insert traffic onto the ring, and switches, which forward transit traffic. Every node on the ring is responsible for detecting congestion, estimating the fair rate of the flows contributing to congestion, and generating the appropriate feedback to upstream nodes in a timely manner so as to prevent buffer overflows. In addition to this, the node acts as an end-system, responding to feedback and adjusting its local transmit rate. Another issue is that in typical end-end congestion control an endpoint estimates and makes a control decision based on the end-to-end Round Trip Time (RTT). Whereas, in a ring the congestion control protocol must be designed to estimate the RTT of the flows contributing to congestion and thus operate on the time scale of the local "congestion domain".

A key part of the RPR design is its framework for distributed congestion control (in this paper, we use the terms 'RPR congestion control' and 'RPR fairness' interchangeably) for best effort traffic. In this framework, when a span is "congested", the node adjacent to the congested link (the "head" of a "congestion domain") sends a locally computed "fair rate" upstream in a Fairness Control Message (FCM). FCMs provide the backpressure mechanism to control the transmission (local "add") rate of upstream nodes. RPR nodes generate FCMs very frequently (potentially once every 100 microseconds), which enables rapid determination of the rate at which upstream stations

Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04)
1092-1648/04 $ 20.00 IEEE

can add traffic to the ring.

We summarize some of the competing goals for an effective RPR congestion control algorithm here.

- fair allocation of bandwidth to source nodes that transmit over a congested link
- "fast start" by sources even with small on-chip MAC transit buffers relative to the link speeds, in contrast to the large packet buffers used in high-speed switches and routers (our simulations use 256 Kbytes for 622 Mbps link speeds)
- high ring utilization
- non-starvation of downstream nodes
- enable fairness eligible traffic to co-exist with higher priority traffic, while allowing the higher priority traffic to achieve its rate, delay and jitter guarantees
- support for a wide range of operating parameters, and minimal oscillations in the throughput of individual stations
- scale to a large number of stations
- the scheme must work well with "higher-layer" end-to-end schemes that may be used, such as those used with TCP.

In a high-speed ring, this implies that the feedback based congestion control must be fast and precise to achieve all of the objectives described above.

Previous work on feedback based congestion control has taught us that it is important to consider the following attributes when designing a good algorithm:

- Accuracy of the fair rate estimation for sources contending for the bottleneck bandwidth.
- Frequency of fair rate estimation, to allow timely response to transient changes in network load.
- Need to balance implementation complexity with the amount of information used for accurate estimation of the fair rate.

This paper presents our proposed robust congestion control algorithm that addresses the above and works within the framework laid out by the IEEE 802.17 Working Group [10]. Initially, the IEEE 802.17 working group had proposed two modes of operations for the fairness algorithm, which we describe in Section III of this paper. The primary difference between these modes is the information used at a congested node to compute the "fair rate":

- **Conservative Mode**: A congested node allocates bandwidth proportionally among the active nodes sending traffic over its congested link, thus requiring nodes to maintain the number of active stations over an estimation time window.
- **Aggressive Mode**: A congested node throttles back all upstream sources to a smoothed version of its local "add" rate, avoiding the need to maintain the number of active stations.

The Conservative and Aggressive modes have been described and compared in [2], [6], [7] and [13]. In this paper we define a detailed set of network configurations and workload scenarios, described in Section V, report the results of our rigorous evaluation, through simulation, of all the design alternatives proposed in the 802.17 working group, and propose an improved congestion control algorithm. Our simulations reveal that both the aggressive and original conservative schemes have potential performance problems. In particular, the aggressive mode exhibits significant oscillations in throughput under some conditions. The design principles underlying the original conservative mode were intended to estimate the fair rate and avoid these oscillations in the throughput. However, the original conservative mode exhibited poor convergence properties, and resulted in starvation of downstream stations, as we show later. Thus, our algorithms modify the original conservative mode scheme to overcome the performance problems discovered by our evaluation. In particular, we

- provide a more accurate estimate of the fair rate and the amount by which it is increased during the time a station remains congested,
- enable an earlier reaction to incipient congestion by introducing an intermediate threshold for the transit buffer occupancy that causes a reduction of the computed fair rate,
- re-estimate the local fair rate more quickly, at the onset of congestion and when a node is severely congested,
- evaluate the frequency and accuracy of the fair rate estimation that is needed for good performance, and
- identify and evaluate the need to shape the aggregate "fairness eligible" traffic forwarded by a node.

We show that our algorithm scales to large ring sizes and a large number of stations, provides high utilization and fast start of newly active stations, while using only a small transit buffer at each station. In addition to the simulation analysis of our algorithm, we also provide comparisons to the schemes that existed prior to our work, where appropriate. Our proposed algorithm has since been accepted as the (final) conservative mode for RPR congestion control in the IEEE 802.17 standard [10].

Section II gives a brief overview of the classes of service supported by RPR, and Section III presents the RPR fairness framework, including the Aggressive, Original Conservative and our proposed Enhanced Conservative modes. Section IV summarizes our simulation framework and Section V presents simulation results. Section VI summarizes our contributions to the RPR fairness algorithm.

II. RPR Service Classes

To support a range of performance requirements, RPR defines three classes of service: Class A, B and C, with strict priority between them. Class A supports traffic requiring bandwidth and jitter guarantees. Class A is further divided into Class A0, which receives the most stringent delay guarantees, and Class A1, which may be subject to some jitter. Class B supports traffic requiring rate guarantees, specified as a committed information rate (CIR) and an excess information rate (EIR). Class C supports best effort traffic. These QoS classes support services similar to those supported by the IETF Diffserv classes [3] (e.g., EF, AF and BE classes.)

The RPR MAC is responsible for scheduling the access to the ring between local "add" traffic and transit traffic. An important goal of the RPR MAC is that, following the rules for IEEE 802 media, once a packet has been accepted by the MAC, it should not be dropped due to congestion. Transit traffic may be buffered at the node's transit buffer when local add traffic gets access to the ring. This allows for a small cushion, so that the locally added packet completes transmission, and there are no partial transmissions. Highly congested stations give priority to transit traffic over traffic that is inserted at the local

station. Therefore, a station may be starved when upstream stations fully utilize a congested link. To avoid node starvation and priority inversion, the fairness algorithm must avoid allowing any station to become highly congested.

RPR nodes may implement a single transit buffer for all classes or dual transit buffers, with the primary transit buffer (PTQ) for Class A (higher priority), and the secondary transit buffer (STQ) for Class B and C (lower priority). Class A traffic and the CIR portion of Class B traffic are subject to admission control. Class C traffic and the EIR portion of Class B traffic are considered to be fairness eligible (FE). The congestion control/fairness algorithm of interest in this paper allocates the available bandwidth under congestion among stations sending fairness eligible traffic. The MAC allows unused Class A1 and CIR of Class B bandwidth to be reclaimed for FE traffic.

III. RPR Congestion Control and Fairness

RPR uses a hop-by-hop congestion control framework designed to support robust, responsive source-based weighted fairness for the case of a single bottleneck link. The fairness algorithm attempts to limit the amount of fairness eligible (FE) traffic crossing a congested link to the capacity available for FE traffic. To achieve this, nodes periodically compute a "local fair rate" and advertise this fair rate information to upstream nodes in fairness control messages (FCM). Upstream nodes derive from the received FCM, a limit for the rate at which they can transmit traffic ("add rate") through a downstream congested link. This limit is a rate that is no more than the station's weighted fair share of the capacity of the congested link.

The fair rate advertised by a node is either 1) the node's locally computed fair rate (based on the capacity of link adjacent to the node), normalized to a station with unit weight, 2) the fair rate received from the downstream node, if that value is smaller, or 3) a "full rate", which is a reserved value, when upstream nodes are not contributing to downstream congestion. The specifics of the fair rate computation differ in the aggressive, original conservative and enhanced conservative modes and will be presented below.

To support spatial re-use, the fairness algorithm is designed to operate within a "congestion domain." This framework aims to ensure that nodes which are not contributing to congestion on a bottleneck link do not have their local "add rate" limited unnecessarily. The node whose locally computed fair rate controls a set of upstream nodes is called the "Head" of the congestion domain, while the first upstream node that advertises the "full rate" is called the "Tail" node. Note that any of the nodes in the congestion domain may become congested (as a result of the STQ buffer occupancy exceeding a threshold) as the dynamics of the traffic sources change with time, even though there may only be a single bottleneck, as viewed by the steady state traffic load. As a result, the node acting as the "Head" node in the congestion domain may continually change, for example as sources come on (send traffic) and go off (stop sending traffic). Thus, the node that controls the fair rate at which upstream nodes may send traffic can be constantly changing, even during the process of convergence when a set of nodes start up.

Each node maintains local variables measuring the rate at which it has added traffic through a downstream congested link (AddRateCongested) and the total added traffic from a station (AddRate) [1]. The rate at which a node can add traffic is limited by the corresponding values AllowedRate and AllowedRateCongested. Specifically, if the node is sending traffic through a congested link, it limits its AddRateCongested to AllowedRateCongested while the node limits its overall AddRate to AllowedRate. Both the add rates and allowed rates are computed once each AgingInterval. The AgingInterval is used to smooth the measured rates of traffic (to overcome the effects of transient burstiness on calculations.)

We define several variables used below. "L" is a node's locally computed fair rate, and "A_l" is the advertised normalized fair rate received in a FCM. "W_l" is the weight associated with the local node and $\sum W$ is the sum of the weights of the upstream active nodes including the local node (if a node sent a packet past the local node in the last AgingInterval, it is considered active.) "R" is the "unreserved rate", which is the link capacity minus the bandwidth reserved for Class A0 traffic. R is an upper bound on the capacity available for FE traffic. Finally, $LpAddRate$ and $LpFwRate$ are exponentially smoothed measured rates of traffic added and forwarded by the node respectively, computed once each AgingInterval. The allowed rates are computed as:

- If the node's adjacent downstream link is congested, then $AllowedRate = L$. Otherwise, the AllowedRate increases gradually up to R as:

$$AllowedRate = AllowedRate + (R - AllowedRate)/\gamma$$

  where γ (increase coefficient) has a default value of 64.
- If a downstream link is congested, $AllowedRateCongested$ is set to:

$$AllowedRateCongested = \min(L, W_l \times A_l)$$

  Otherwise (e.g., when a node receives a "full rate" advertisement in its FCM), it ramps up to R as the AllowedRate does above.

A node determines that it is congested when its STQ occupancy exceeds a specified "low threshold." When the STQ length exceeds a "high threshold", this is interpreted as the onset of severe congestion, and the congested node stops adding FE traffic to the ring to relieve congestion and to ensure that traffic already on the ring will not be lost. Thus, to avoid "starvation" of the congested node, it is critical that the fairness mechanism manages upstream demand carefully. Several features of the enhanced conservative mode described below are designed to avoid the STQ length exceeding the high threshold.

A. Achieving "Fast Start" and Preventing Packet Loss

One of the fundamental goals of the RPR congestion control mechanism is to allow sources to start at the full rate, so as to achieve high link utilization. Another reason for this "fast start" in RPR is because RPR is meant to be used as a metro transport technology, and a multitude of higher layer protocols and applications need to be supported, with diverse throughput and latency requirements. As a result, it is highly desirable that sources are allowed to transmit as fast as possible, when they

[1] Each node uses a local ring topology database and the MAC address of the Head node adjacent to the downstream congested link, which is carried in the FCM, to compute these variables.

start up, while still ensuring no packets are lost on the ring and overall long term fairness is achieved by the congestion control algorithm.

When the transit buffer exceeds the high hreshold for the STQ buffer occupancy for FE traffic, the local station stops inserting traffic, thus ensuring that all the transit traffic can be transmitted on the out-bound interface of the ring. The threshold is set to account for at least the packets in flight on the span between the congested node and the upstream station. With a larger transit buffer, the local station may be able to transmit more packets before it is shut off. The policy of the congestion control algorithm is to ensure that access is arbitrated in a fair and equitable manner (hence the term "fairness algorithm" being used synonymously with congestion control algorithm in the IEEE 802.17 Working Group.) Therefore, an additional measure by which the congestion control algorithm is evaluated is the length of time a downstream station is starved even under transient overloads.

The congestion control algorithm must react fast enough to bring the transmission rate of sources that "fast start" down to their fair rate, while ensuring all the other goals outlined in Section I are met. RPR incorporates the capability to react quickly to congestion by having a feedback mechanism that sends fairness control messages (FCM) frequently - typically once every 100 μ seconds. The algorithm we propose in this paper is one of the first schemes that enables stations to fast start without losing transit traffic even with relatively small transit buffers.

B. Aggressive Mode Local Fair Rate Calculation

The key difference between the various modes (aggressive, original conservative and enhanced conservative modes) is the way the congested node computes the local fair rate, L. In the aggressive mode, when a node transitions to the congested state, L is calculated, and then updated every aging interval, as:

$$L = LpAddRate;$$

In the aggressive mode (as in the conservative modes), a congested node stops adding FE traffic to the ring when the STQ length exceeds the high threshold. With the aggressive mode, a node exits the congestion state as soon as its STQ length goes below the low threshold (unlike the conservative modes described below).

Table I shows the key state transitions at the congested node that influence the calculation of the local fair rate. The station samples the STQ occupancy every 'Aging Interval' (typically 100 $\mu secs$.) In the uncongested state (UNCG), when the STQ occupancy exceeds the low threshold, the local fair rate L is set to the local station's low pass filtered add rate and the station transitions to the congested state (CGST, shown in Row 2 of Table I). When the station is in the congested state (STQ $\geq$ low threshold), L remains set to the low pass filtered add rate. When the STQ drops below low threshold, the station once again transitions to the uncongested state (UNCG, shown in Row 4).

One of the motivations for the aggressive mode was simplicity, since it avoids the need to estimate the number of active nodes for the initial fair rate estimation, and also does not use an adaptation mechanism to fine tune the computed local fair rate. However, we will show below in the experimental section that this causes significant oscillations in the throughput achieved by individual nodes, since L can change dramatically when the STQ length crosses the low threshold.

C. Original Conservative Mode Local Fair Rate Calculation

The conservative modes (both the original and enhanced conservative modes) differ significantly from the aggressive mode in the algorithm used by the congested node to compute the local fair rate, L. In the conservative mode, when a node first transitions from the "uncongested" to "congested" state, it estimates an initial fair rate based on the link capacity available for FE traffic and the number of active nodes (and their weights). Then, the congested node continually adapts its computed fair rate L, while it is congested. The node reduces its local fair rate multiplicatively as long as its secondary transit buffer (STQ) is above a specified "high threshold", and increases its local fair rate when STQ goes below the "low threshold". The node maintains the local fair rate L when the STQ buffer occupancy is between the low and high thresholds, and continues to advertise that rate. The high threshold is typically set as $0.25 \times STQ\_size$ and the low threshold is typically set at $0.5 \times high\_threshold$. The node exits the congested state when its local fair rate reaches the "unreserved rate", (at which point it advertises the "full rate" to upstream stations in its FCM.) unlike the aggressive mode.

The key state transitions for the original conservative mode are shown in Table II. The action in each of the rows of the table is primarily the calculation of L, as per the equations described above.

The original conservative mode in contrast to the aggressive mode adjusts the fair rate, L, while the station remains in the congested state (CGST). In the uncongested state (UNCG), when the STQ buffer occupancy reaches the "low threshold" (in Row 2), the congested node computes the local fair rate L based on the number of active stations as an equal share of the unreserved capacity, and the node transitions to the congested state (CGST).

$$L = W_l \times R/ActiveStations$$

The congested node adapts L periodically, based on an initially fixed round trip time for the network. When the STQ occupancy increases above the high threshold, the node reduces L multiplicatively as (shown in Row 5 of Table II):

$$L = L - L/\beta$$

Here, β (the ramp down coefficient) has a default value of 64. This is also the point at which the node stops inserting add traffic. If this buffer occupancy persists, the node can be starved.

When the congested node's buffer occupancy, STQ length, drops below the low threshold, L is increased additively, (applying the action in Row 6 periodically, every Fairness RTT (FRTT)) in proportion to the difference between the current value of L and the unreserved rate, R as:

$$L = \min(R, L + (R - L)/\beta)$$

In contrast to the aggressive mode, the conservative mode stays in the congested state (CGST) until it reaches the unreserved rate, R (as shown in Row 4). The node exits its congested state to the uncongested state (UNCG) when $L \geq R - \beta$. The head node advertises in the Fairness Control Message, the fair rate a station with unit weight may transmit at, by dividing L with the local weight:

$$A_l = L/W_l$$

The original conservative mode did not attempt to perform a weighted fair allocation of the rates. Rather, every station was considered to have the same unit weight.

Current state		Row	Next state	
State	Condition		Action	State
INIT		1	$L = R$	UNCG
UNCG	AgingInterval Expired &&$(STQ > STQLowThreshold)$	2	$L = LpAddRate$	CGST
	AgingInterval Expired	3	$L = R$	UNCG
CGST	AgingInterval Expired &&$(STQ <= STQLowThreshold)$	4	$L = R$	UNCG
	AgingInterval Expired	5	$L = LpAddRate$	CGST

TABLE I

LOCAL FAIR RATE CALCULATION TABLE FOR AGGRESSIVE MODE

Current state		Row	Next state	
State	Condition		Action	State
INIT		1	$L = R$	
UNCG	AgingInterval Expired &&$(STQ > STQLowThreshold)$	2	$L = W_l \times R/ActiveStations$ Reset RTT	CGST
	AgingInterval Expired	3	$L = R$	UNCG
CGST	AgingInterval Expired &&$L \geq R - \beta$	4	$L = R$	UNCG
	Aging Interval Expired &&$STQ > STQHighThreshold$ && RTTWorthOfIntervalPassed	5	$L = L - L/\beta$ Reset RTT	CGST
	AgingInterval Expired &&$STQ <= STQLowThreshold$ && RTTWorthOfIntervalPassed	6	$L = \min(R, L + (R - L)/\beta)$ Reset RTT	

TABLE II

LOCAL FAIR RATE CALCULATION TABLE FOR ORIGINAL CONSERVATIVE MODE

Current state		Row	Next state	
State	Condition		Action	State
INIT		1	$L = R$	UNCG
UNCG	AgingInterval Expired && $(STQ > STQLowThreshold)$	2	if $(LpAddRate < (W_l \times R)/\sum W)$ $L = W_l \times (R - LpAddRate)/(\sum W - W_l)$ else $L = W_l \times R/\sum W$, Reset FRTT	CGST
	AgingInterval Expired	3	$L = R$	UNCG
CGST	AgingInterval Expired &&$(L \geq R - \beta)$	4	$L = R$	UNCG
	AgingInterval Expired && $STQ > STQMediumThreshold$ && FRTTWorthOfIntervalPassed	5	$L = L - L/\beta$ Reset FRTT	CGST
	AgingInterval Expired && $STQ < STQLowThreshold$ && FRTTWorthOfIntervalPassed	6	$L = \min(R, L + (R - (LpAddRate + LpFwRate))/\alpha)$ Reset FRTT	
	AgingInterval Expired && $(STQ > STQHighThreshold)$	7	if $(LpAddRate/W_l < LpFwRate/(\sum W - W_l))$ $L = \min(L, W_l \times (LpAddRate + LpFwRate)/\sum W)$	

TABLE III

LOCAL FAIR RATE CALCULATION TABLE FOR ENHANCED CONSERVATIVE MODE

D. Enhanced Conservative Mode Local Fair Rate Calculation

We describe our proposed enhanced conservative mode protocol in this section.

Since RPR fairness operates over a set of nodes in a "congestion domain," rather than end-to-end as in traditional congestion control schemes, each congested node has to dynamically estimate the round-trip time of the congestion domain, known as the Fairness RTT (FRTT). The congested node re-computes its local fair rate once per FRTT, represented as FRTTWorthofIntervalPassed in Tables II and III. The FRTT accounts for the time taken by the FCM to travel from the head to tail node, for the change to take effect, and then to be reflected in the traffic observed at the head node, including queueing delays in the STQs of the intermediate nodes between the "tail" and "head." One of our contributions was to determine via extensive simulations (not reported here) the frequency and accuracy of FRTT measurement needed to achieve acceptable system performance and avoid starvation of downstream nodes.

The enhanced conservative mode uses a weighted fair allocation of the available capacity when computing the local fair rate, unlike the original conservative mode, which did not handle node weights in the local fair rate calculation. Each node keeps track of the active stations traversing its adjacent links, and their associated weights, which are distributed using RPR's topology discovery mechanism.

The enhanced conservative mode significantly improves estimation of the local fair rate, L, at each of the steps in the state machine at the congested node. Our first enhancement is computing the initial value of L more accurately than the conservative mode, when a node first enters the congested state. When the node transitions from an uncongested state to the congested state (when the STQ buffer occupancy exceeds the low threshold), L is calculated as:

$$\text{If } LpAddRate < (W_l \times R)/\sum W: \quad L = W_l \times (R - LpAddRate)/(\sum W - W_l); \tag{1}$$

$$\text{Else:} \quad L = W_l \times R/\sum W;$$

The first part in the equation accounts for the case when the local node has a small local demand. The congested node more

accurately estimate the fair rate and reallocates the unused rate to upstream stations.

Once in the congested state, the node adapts its local fair rate once every FRTT. The enhanced conservative mode introduces a third threshold for STQ buffer occupancy, the "medium threshold", which is mid-way between the low and high thresholds. If STQ occupancy goes above the medium threshold, this is used as an indication of increased congestion, causing the congested node to start reducing traffic from upstream nodes before its STQ goes above high threshold. Thus, the enhanced conservative mode provides an early action for incipient congestion, thus reducing the likelihood of starvation of the congested node. The congested node reduces its local fair rate multiplicatively when STQ is above the medium threshold. The congested node reduces the local fair rate L it computes and advertises to upstream nodes as:

$$L = L - L/\beta \quad (2)$$

β (the ramp down coefficient) has a default value of 64.

When the STQ length falls below the low threshold, this is interpreted as a reduction in the level of congestion, and the node allows L to ramp up as:

$$L = \min(R, L + (R - LpAddRate - LpFwRate)/\alpha) \quad (3)$$

where α (the ramp up coefficient) is a configurable parameter and has a default value of 64. In the equation above, $(R - LpAddRate - LpFwRate)$ is a more accurate estimate of the left-over capacity that can be used for fairness eligible traffic, to correctly ramp up L.

With the enhanced conservative mode, we have separated the coefficients for ramp up and ramp down. α may be set to a larger value than β (i.e., smaller increase steps) for large rings. A larger value of α significantly improves the scalability of the scheme when there are a large number of active stations. It avoids increasing of the STQ buffer occupancy at downstream stations above the high threshold and thus reduces the likelihood of starving them.

When the estimate of the number of active nodes used to compute the initial fair rate (when the station first transitions to the congested state in Row 2 of Table III) is too small, it leads to an inaccurate initial local fair rate estimate, thus potentially causing the local node to be starved due to severe congestion. The enhanced conservative scheme introduces a new action in the state machine, (Row 7 in the Table) which recomputes L with an updated value of $\sum W$, thus aggressively reducing the advertised fair rate. This computation is done once every aging interval when the node is severely congested, rather than waiting for an FRTT. This feature of the enhanced conservative mode allows it to rapidly react to node starvation, and is a key improvement over the original scheme:

If $(STQ > HighThreshold \;\&\&\; LpAddRate/W_l < LpFwRate/(\sum W - W_l))$:

$$L = min(L, W_l \times (LpAddRate + LpFwRate)/\sum W) \quad (4)$$

The node exits the congested state when the locally computed fair rate exceeds the "unreserved rate" (at which point it advertises the "full rate" to upstream stations in its FCM.)

Table III presents the key state transitions at the congested node, related to the computation of the local fair rate. The table follows the structure of the original conservative mode, as shown in Table II, except for the introduction of an additional row (Row 7) which reflects the action taken by the node when it is severely congested. The main difference between the schemes is in the actions, where the local fair rate is computed using the equations described above. Further, the reduction in the fair rate when the node is in the congested state (CGST) happens earlier when the STQ occupancy exceeds the medium threshold, as shown in Row 5.

In summary, our proposed enhancements includes a set of essential changes to more accurately estimate the local fair rate at the congested node. This includes changes to carefully account for the traffic added by the node when it is sending less than its fair rate because it has a low demand (equation (1)). This is necessary to reallocate the unused portion of the head node's fair share to other nodes. A second contribution includes a set of changes to more rapidly re-estimate the local fair rate in equation (4) when the node is severely congested (row 7 of the fairness state machine in Table III), rather than relying on the relatively slow mechanisms in equations (2), (3) and (5) for ramping up and ramping down the rate when the node is congested. We also compute the increase in the local fair rate more accurately, using equation (3). The third contribution established the need to understand the frequency and accuracy of round trip estimation. Our fourth contribution validated the need for the MAC to shape the combined rate of local "add" traffic and transit traffic at each node to the "unreserved rate," which is the link rate less the bandwidth reserved for Class A0 traffic. We describe this in more detail in Section V

IV. Experimental methodology

Our results are based on NS-2 [11] simulations, with an initial implementation of the RPR MAC obtained from Rice University. We extended the simulator and added support for RPR's conservative mode fairness scheme. We use configurations of nodes in a unidirectional ring ranging from 6 to 20 nodes (see Figures 1, 6, and 8). All spans for a given topology have identical propagation delays (either 2 ms or 0.6ms), and the link speed is 622 Mbps. For the traffic matrix, we have generally used a "parking lot" configuration. For the workload, we used UDP (constant rate) sources, with both greedy and non-greedy flows. The greedy flows have a demand equal to the link rate. The non-greedy UDP flows have a constant rate of 50 Mbps. We also report experimental results with flows that go ON and OFF with a pattern of new flows coming ON that generally causes the congestion domain to become larger. Flows going OFF were used to demonstrate how other nodes reclaim the bandwidth that was released. The STQ buffer size is modest, set to 256 Kbytes and the RPR MAC client buffer size was set to 1000 packets. Packet size was set at 978 bytes. The "Aging Interval" and "Advertisement Interval" were set to 100 microseconds. The throughput at each source node is evaluated with an averaging interval of 5 milliseconds. The link utilization is calculated as the total number of bytes transmitted over the link from time $t = 0$.

V. Simulation results

We first compare the performance of the three different modes with different scenarios, where sources generate UDP

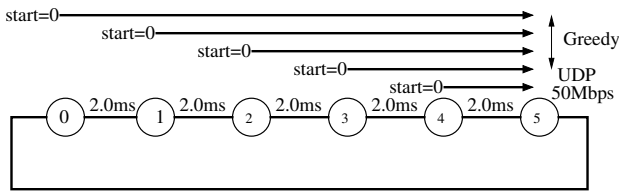


Fig. 1. Simulation configuration with steady flows

and TCP traffic. We then focus on evaluating the performance of the enhanced conservative mode.

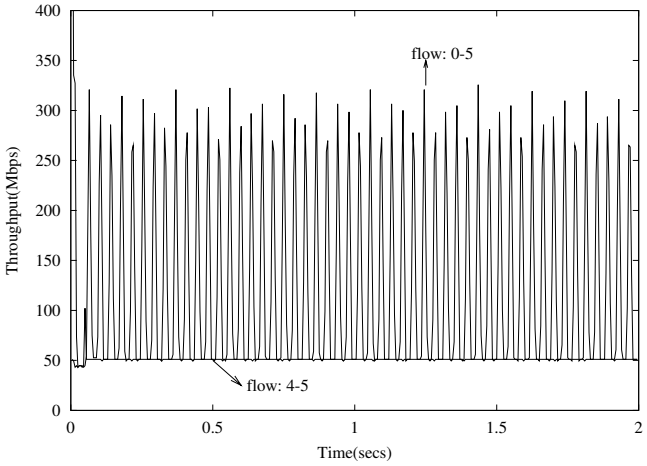
A. Performance with Constant Rate(UDP) flows

For the first set of experiments, we use the configuration shown in Figure 1. The flow sourced at node 4 is a limited UDP flow with a constant rate of 50 Mbps, while the other four flows are greedy UDP streams.

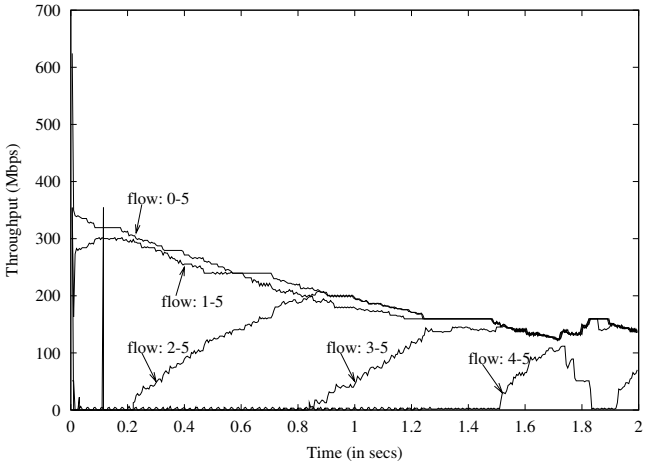
Figure 2(a) shows the throughput of the flows with the aggressive mode [10], exhibiting considerable oscillation for the throughput of the flow from source node 0 (similar behavior was observed for source nodes 1 through 3). The throughput of the flow originating from node 4 remains at 50 Mbps. The oscillatory behavior for the upstream flows is because the "head node" (node 4) advertises its own AddRate, which is limited to 50 Mbps, every time it transitions to the congested state. This causes a dramatic reduction of the transmit rate from the upstream nodes, which relieves the congestion at the head node. The upstream nodes are then allowed to ramp up to the "Full" (unreserved) rate, which again creates congestion at node 4 and the cycle repeats.

Figure 2(b) shows the throughput of the flows with the original conservative mode. The throughput of the upstream flows tends to converge toward the fair share after a considerable period of time. Further, flow 4-5 is starved most of time, and oscillates afterwards, over a long time scale. The upstream flows (2-5, 3-5) are also starved for significant periods of time. Node starvation is due to the fact that upstream stations are sending too much traffic, and the STQ occupancy remains above the high threshold at the starved nodes.

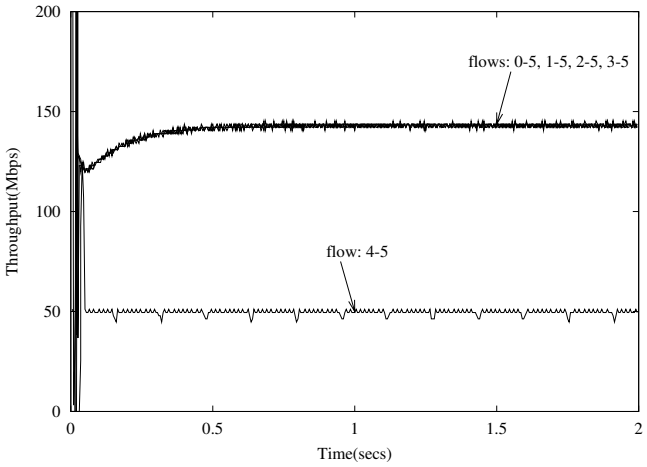
In Figure 2(c), we show the throughput for all the flows with the enhanced conservative mode, which demonstrates a dramatic improvement compared to Figure 2(a) and (b). Notice that the sources start at the full link rate, but very rapidly reduce to a rate of 120 Mbps, which is the equal share for the 5 flows, within approximately 2.5 round trip times (the congestion domain's fixed round trip propagation time (RTT) is 20 milliseconds.) The bandwidth left unused by node 4 is re-allocated to the other nodes: in the time period between 50 and 500 milliseconds, the rates of the flows from stations 0 through 3 ramp up from 120 Mbps to 143 Mbps as a result of the accurate estimation of the fair rate allocated to upstream stations, in equation (3). Node 4's throughput remains at 50 Mbps (limited by the demand from that source node). No packets are lost from the secondary transit buffer, even though the sources start at full rate, and the STQ size is only 256 Kbytes. Figure 3 shows that both enhanced and original conservative mode achieve more than 97% bandwidth utilization on the most congested link 4-5. In contrast, the corresponding link utilization achieved with the aggressive mode is approximately 86%, as a result of the considerable oscillations in the throughput of the upstream nodes.



(a) Aggressive mode



(b) Original conservative mode



(c) Enhanced Conservative mode

Fig. 2. Throughput of individual flows with different modes with UDP Traffic for configuration shown in Fig. 1.

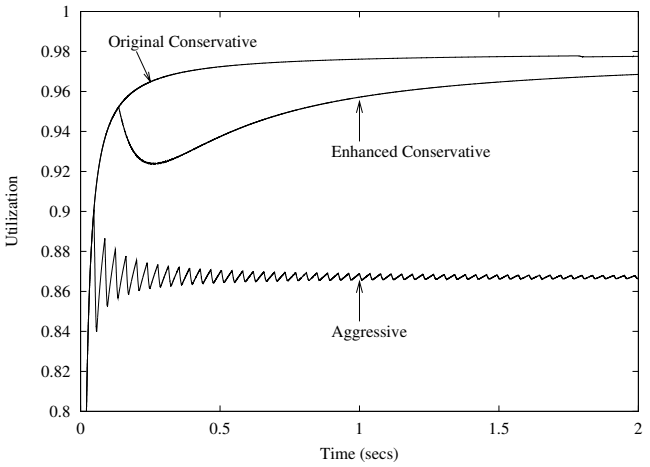


Fig. 3. Link utilization on the bottleneck link between node 4 and 5, with UDP traffic.

One issue that typically arises in examining the performance of congestion control mechanisms at the datalink layer is how it interfaces with a window-based transport protocol such as TCP. We used the same configuration shown in Figure 1, with the difference that each greedy UDP flow is replaced with ten TCP flows (for a total of 40 TCP flows), so as to create bursty, but greedy TCP flows. The throughput behavior with TCP flows were similar to that with constant rate UDP flows, as seen in Figure 2 (we did not include these results to limit the length of the paper.) We believe that the enhanced conservative mode has the appropriate characteristics for a datalink layer congestion control mechanism and works well with the higher layer TCP mechanisms. We now present additional results for the enhanced conservative mode, to demonstrate its effectiveness, in a range of scenarios.

B. Performance with ON-OFF flows

We demonstrate the responsiveness of the enhanced conservative mode to dynamic changes in demand and the congestion domain RTT in this experiment. We use a set of greedy UDP flows that start and stop at different times as shown in Figure 4. This pattern results in the tail of the congestion domain (whose head node is at node 4) to gradually expand from node 3 up to node 0. The flows $3-5$ and $4-5$ are stable and long-lived flows while the others turn ON and OFF at different times (see points S1, S2, S3 and S4 in Figure 5.) Because a new flow triggers the downstream node to re-evaluate the local fair rate quickly according to equation (4), we observe from Figure 5 that the flows converge to their new fair shares within $2-4$ RTTs. Because a new flow starts at the full rate, this presents a sudden load on the downstream station. When the STQ buffer occupancy goes above the high threshold, a downstream congested node stops adding traffic (starving itself), thus reducing the congestion on its downstream link. However, the period during which the node is starved is relatively small due to equation (4).

We observe that, as new flows turn on, the rate at which each of the flows transmit stabilizes to the fair rate (in the figure, see for example interval A1, where the two flows $3-5$ and $4-5$ converge to $300Mbps$; in interval A2, where three flows $2-5$, $3-5$ and $4-5$ converge to $200Mbps$; in interval A3, where four flows $1-5$, $2-5$, $3-5$ and $4-5$ converge to $150Mbps$;

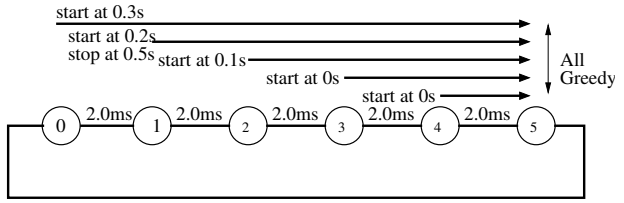


Fig. 4. Network configuration and traffic pattern to examine performance with on-off flows

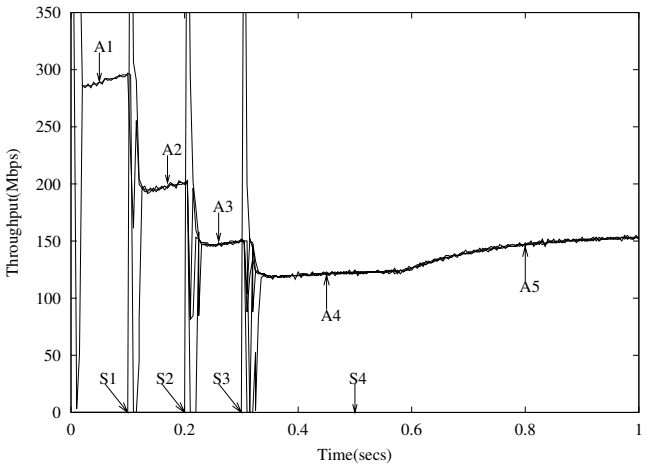


Fig. 5. Throughput of ON-OFF flows.

and in interval A4, where five flows $0-5$, $1-5$, $2-5$, $3-5$ and $4-5$ converge to $120Mbps$). As a flow from node 1 turns off (at point S4), the STQ length at the downstream node quickly drops below the low threshold, which triggers the ramp-up of the local fair rate calculated at the head node according to equation (3). Then, we observe that the four flows $0-5$, $2-5$, $3-5$ and $4-5$ gradually reclaim the bandwidth released by flow $1-5$ in the interval A5.

C. Achieving spatial reuse

To demonstrate the spatial reuse capability of the RPR enhanced conservative mode, we use the configuration shown in Figure 6 which has two congestion domains. The first congestion domain is from node 0 to node 3 (which we call FD1) with the congested link being $2-3$. The flow from node 0 to 3 is source limited at $60Mbps$. The other domain is from node 3 to node 9 (called FD2) with the congested link being between nodes 8 and 9. The congestion domains do not overlap with each other. Since our objective is to achieve source based fairness, the fair rate in FD1 is $281Mbps$ for nodes 1 and 2. The fair rate in the domain FD2 is about $100Mbps$.

We show the throughput of the individual flows in Figure 7. As expected, the throughput of the individual flows in FD2 con-

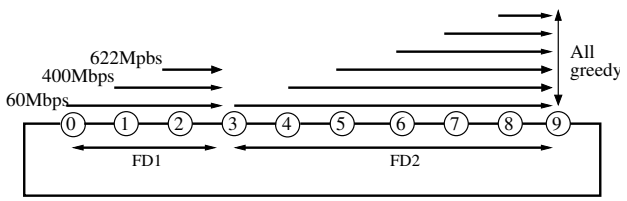


Fig. 6. Simulated networks with two bottleneck links: link $2-3$ and $8-9$. The link propagation delay is set to $0.6ms$.

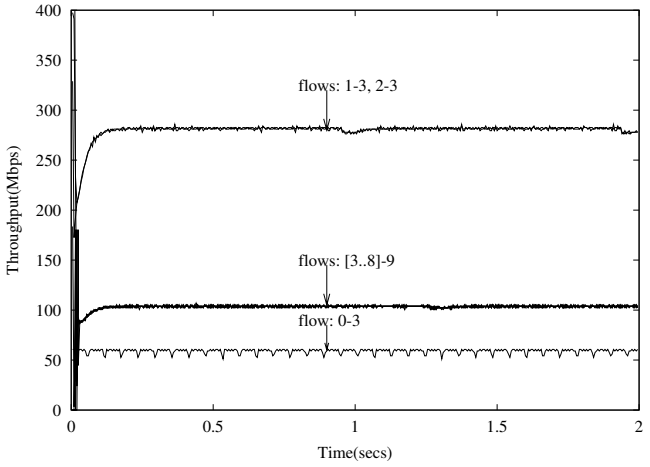


Fig. 7. Throughput with two congestion domains, demonstrating effective spatial re-use.

verges to the fair rate of $100Mbps$, and the throughput of flows $1-3$ and $2-3$ converge to the fair rate of $281Mbps$ very rapidly. The experiment demonstrates that the conservative mode is able to achieve both spatial reuse and rapid convergence to a fair allocation.

D. Achieving QoS with multiple classes of traffic

Another set of important requirements for RPR is to guarantee the jitter and delay for high priority class A traffic, guaranteeing the committed rate (CIR) for class B traffic and fairness of FE traffic, all while achieving high link utilization.

A contribution of ours to the RPR congestion control mechanism was examining the need to shape the total FE traffic forwarded by each node. This is needed to allow the co-existence of fairness eligible (subject to congestion control) traffic and the higher priority guaranteed Class A_0 traffic. We observed there is a fundamental need to match the shaper credit increment and decrement rates, to avoid starvation of the congested node. Shaping the aggregate FE (transit plus add) traffic to the "unreserved rate" matches the rate at which shaper credits are incremented, thus ensuring isolation of the different traffic classes (we have not included more details and simulation results on this, due to space limitations.)

We demonstrate the coexistence of multiple QoS classes using the 20 node network configuration shown in Figure 8. The most downstream flow between nodes $18-19$ is a class A0 (reserved) flow starting at $t = 0$ with a rate of $200Mbps$. A Class B flow with a CIR of $60Mbps$ and an EIR of $562Mbps$ starts at $t = 0$ between nodes $10-19$. All the other upstream flows are greedy UDP (Class C) flows, which start at time $t = 0.5$ seconds. We start the greedy UDP flows slightly later than class A0 and Class B flows to observe the impact of the sudden start-up of a large number (17) of upstream greedy Class C flows.

As shown in Figure 9, initially the Class A0 traffic between nodes $18-19$ is at $200Mbps$, and the remaining bandwidth is made available to the Class B flow (approximately $422Mbps$). At time $t = 0.5$, the Class C flows start up and achieve a rate of approximately $20Mbps$ each. The Class A0 flow is not impacted, and remains at $200Mbps$, thus ensuring that the rate, delay (since the PTQ buffer exclusively used for Class A0 is

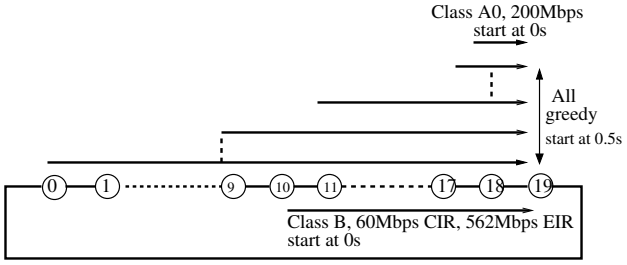


Fig. 8. Simulated networks with mixture of class A0, class B and class C traffic. The link propagation delays are set at 0.2ms.

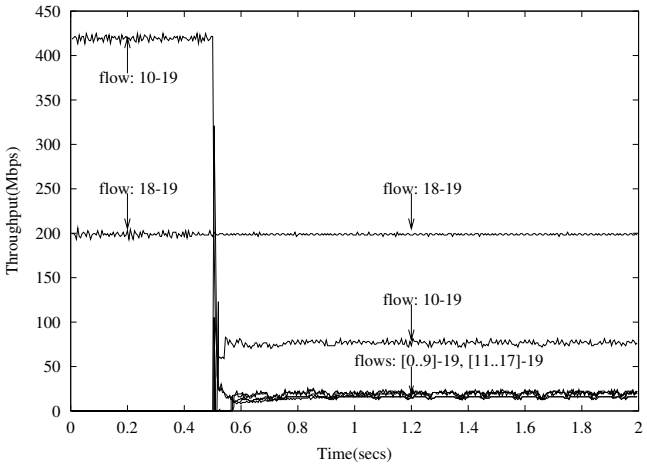


Fig. 9. Throughput with mixture of class A0, Class B and class C traffic

limited in size) and jitter guarantees for the flow are maintained. The throughput of the Class B flow from station $10-19$ drops to about $80Mbps$. This shows that the flow gets its guaranteed CIR rate of $60Mbps$, plus a fair share of the bandwidth available for FE traffic, of $20Mbps$. We thus observe that the enhanced conservative mode achieves rate and delay guarantees for reserved traffic (Class A0), CIR rate guarantees for burstable traffic (Class B), and fairness among all the FE flows.

E. Achieving Weighted Fairness

So far, we always set the weight at every source node as 1. In this experiment, we use a slightly different scenario and traffic pattern to demonstrate the ability of the proposed scheme to achieve source node based weighted fairness. The configuration is shown in Figure 10.

Figure 11 shows the throughput of the individual source nodes, demonstrating that all the source nodes are able to get the share of the bandwidth proportional to their weights.

VI. Summary and Conclusions

In this paper, we provided an analysis of Resilient Packet Ring (RPR) congestion control/fairness algorithms and pro-

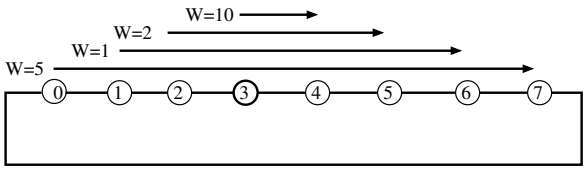


Fig. 10. Simulated networks with weighted source nodes

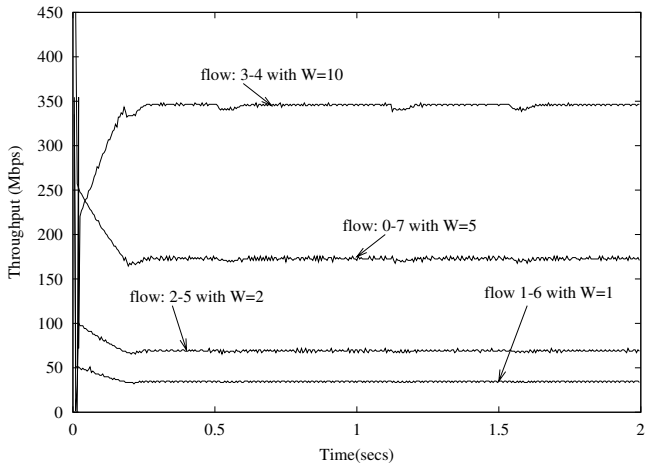


Fig. 11. Throughput with weighted source nodes: flow $0-7$ converges to about 170Mbps; flow $1-6$ converges to about 35Mbps, flow $2-5$ converges to about 70Mbps, and flow $3-4$ converges to about 350 Mbps.

posed an improved and robust congestion control algorithm that works within the original RPR framework to achieve a source-based fair rate allocation. The IEEE 802.17 Working Group proposed two modes of operation for the algorithm: Aggressive and Conservative Mode. Through our extensive simulations we discovered that both the aggressive and original conservative schemes had potential performance problems. In particular, the aggressive mode exhibited significant oscillations in throughput under some conditions. The original conservative mode exhibited poor convergence properties, and resulted in starvation of downstream nodes. We presented a subset of our simulations that justify these assertions.

Designing a robust fairness algorithm for RPR is challenging, due to the high link speeds, small buffer sizes, and desire to have nodes "fast start" so as to achieve high utilization. The improved fairness algorithm we proposed achieves this, taking advantage of the fast hop-by-hop feedback control using a set of mechanisms that rapidly relieves congestion when it occurs. In particular, it demonstrates the ability to adaptively determine the fair share for the contending nodes on the ring, even if the initial estimate is inaccurate. The congested node uses an increase-decrease algorithm to adjust its estimate of the fair share. We showed that the nodes converge to a fair share allocation within a small number of round-trip times. Further, the scheme re-allocates bandwidth unused by a source node having a demand that is less than its fair share. We demonstrated that our algorithm minimizes oscillations in a source's throughput and effectively avoids starvation even under severe congestion - a feature that is superior to existing mechanisms, and highly desirable in a transport network. We also found that it is important for nodes on the ring to estimate the actual congestion domain RTT (not reported here), to match the frequency on adaptation at the congested nodes to the fundamental control frequency of the ring. Finally, we demonstrated that the performance targets for Class A and B traffic can be met while co-existing with a large number of greedy Class C traffic demands.

VII. Acknowledgments

We thank the authors of [13] for establishing the initial framework for the conservative mode. We also gratefully appreciate the contributions of Necdet Uzun whose significant technical input and insight helped improve the scheme. We also thank the members of the various Fairness Adhoc Groups created within the auspices of the IEEE 802.17 Working group for helping refine the scheme further.

References

[1] Thomas S. Afferton, Robert D. Doverspike, Charles R. Kalmanek and K. K. Ramakrishnan, "Packet-Aware Transport for Metro Networks," *IEEE Communications Magazine*, March 2004, pp.120-127.

[2] Harmen Van As, "Fairness Benchmarking of MACs," *http://grouper.ieee.org/groups/802/17/proceedings.html*, January 2002.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, Dec. 1998 (http://www.ietf.org/rfc/rfc2475.txt?number=2475)

[4] I. Cidon, and Y. Ofek, "Metaring- A Full Duplex Ring with Fairness and Spatial Reuse," *IEEE Transactions on Communications*, October 1993, pp. 110-120.

[5] I. Cidon, L. Georgiadis, R. Guerin, Y. Shavitt, "Improved Fairness Algorithms for Rings with Spatial Reuse," *IEEE/ACM Transactions on Networking*, April 1997, pp: 190-304.

[6] F. Davik, M. Yilmaz, S. Gjessing, N. Uzun, "IEEE 802.17 Resilient Packet Ring Tutorial," *IEEE Communications Magazine*, March 2004, pp: 112-118.

[7] M. J. Francisco, F. Yuan, C. Huang, and H. Peng, "A Comparison of Two Buffer Insertion Ring Architectures with Fairness Algorithms," *Proceedings of ICC'03, Anchorage*, May, 2003.

[8] 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 on Internet Traffic Engineering and Traffic Management*, Wurzburg, Germany, July 2002.

[9] E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuck, "Improving the Fairness of Distributed-Queue-Dual-Bus Networks," *Proceedings of IEEE Infocom'90*, June, 1990.

[10] IEEE, "IEEE Standard 802.17: Resilient Packet Ring," *http:/ieee802.org/17* (standard specification in progress).

[11] The Network Simulator "ns-2," *http://www.isi.edu/nsnam/ns/*.

[12] J. H. Schuringa, G. Remsak, H.R. van As, A. Lila, "Cyclic Queueing Multiple Access (CQMA) for RPR Networks," *European Conference on Networks & Optical Communications*, June 2002, Darmstadt, Germany.

[13] F. Yuan, C. Huang, H. Peng, and J. Hawkins, "Performance Analysis of Resilient Packet Rings with Single Transit Buffer," *Proceedings of IEEE ICT'2002*, Beijing, June, 2002.