Improved fairness algorithm to prevent tail node induced oscillations in RPR

Yan F. Robichaud and Changcheng Huang
Optical Networks Laboratory
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
{yan, huang}@sce.carleton.ca

Abstract – **The IEEE P802.17 workgroup has standardized a ring network architecture and associated protocol called Resilient Packet Ring (RPR). RPR tries to overcome previous MAN technologies shortcomings in high-speed networks. The RPR fairness mechanism suffers from severe permanent oscillations under certain conditions. Several propositions were made to solve this problem but all needed significant modifications of the current RPR protocol. In this paper, we propose an improved algorithm to prevent tail node induced oscillations. It requires only simple modifications to the current fair rate advertisement mechanism of RPR. Our simulation results show that the improvements work under various conditions and increase the average throughput of RPR.**

Keywords – **Resilient Packet Ring, Bandwidth Management, Media Access Control, Fairness Algorithm.**

I. INTRODUCTION

The IEEE P802.17 workgroup has standardized a ring network architecture and associated protocol called Resilient Packet Ring (RPR) [1]. RPR tries to overcome previous MAN technologies shortcomings in high-speed networks ([2], [5]). Its salient features include dual counter-rotating ring topology with simple operation and management; automatic topology discovery; fast protection mechanism; interworking with the IEEE 802 family of networks; multiple class of services; support of efficient statistical multiplexing, dynamic bandwidth reclamation and spatial reuse through destination-stripping to attain high utilization; simple shortest path routing through either inner or outer ringlets; and distributed fairness algorithm with traffic shaping to share fairness eligible traffic among competing nodes, avoiding starvation problems.

The MAC Client stores the packets to be sent in its transmit buffers. The MAC layer scheduler then adds the frames to the appropriate ringlet in the downstream direction from station to station until the destination is reached. Stations that are locally sourcing traffic when receiving frames from the ring will temporarily store the frames in a transit buffer. Two configurations are supported: a station with only one small transit buffer (1TB) or one small high-priority transit buffer and a second larger low-priority transit buffer (2TB). The RPR scheduler prevents losing frames by insuring that no transit buffers can overflow. Three classes of service are available: high-priority guaranteed bandwidth, low-delay, low-jitter class A; medium priority, low-jitter, bounded delay class B; and best-effort class C. ClassB and classC can reclaim unused bandwidth.

The rest of this paper is organized as follows: Section II explains in more detail the RPR fairness algorithm and some of its shortcomings. Section III presents an improved algorithm to the fairness mechanism to prevent the tail node induced oscillations. Section IV describes the simulation scenarios and parameters used. Section V analyses the results. Finally, we summarize and conclude our findings.

II. FAIRNESS ALGORITHM

The RPR fairness algorithm goal is to distribute unallocated and unused reclaimable bandwidth fairly among the competing nodes and use this bandwidth to send fairness eligible traffic (classC and classB-EIR). The algorithm is executed when a node detects a congestion indication on its output link. In the case of a 1-transit buffer configuration, this indication is given by the averaged link rate becoming greater than a low threshold (usually 80% of the unreserved link bandwidth) or the head of line delay for transmit packets is greater than a certain limit. In the case of 2-transit buffer configuration, the indication is given by either the link rate becoming greater than the unreserved rate or when the secondary transit queue occupancy becomes greater than a low threshold. When these conditions occur, the output link is deemed congested and the node triggers the fairness algorithm. The congested node calculates a fair rate to be advertised to upstream nodes in order to decrease their added traffic. This congested node becomes the head node of the fairness domain. There are two methods of calculating the advertised fair rate, a conservative mode and an aggressive mode. Previous papers have shown that both modes can create oscillations under certain conditions. Recently, the final draft of the RPR protocol [1] solves some of the issues regarding the conservative mode oscillations, making it more stable and quicker to reach the appropriate fair rate under constant traffic patterns (see Figure 1 and Figure 2).

The calculated fair rate is advertised along the upstream nodes until a node considers that the next upstream node does not contribute to the head node congestion. This node, the tail node, resets the advertised rate to the full link rate (if it is not itself more congested), to allow its upstream nodes to send as much traffic as they need. This tail node behavior can produce permanent oscillations in certain scenarios which degrades the average throughput ([3], [4], [7], [6]). We study these two scenarios in this paper, shown in Figure 3 (2-node, aggressive mode) and Figure 6 (Parallel Parking Lot, conservative mode).

To prevent these oscillations under unbalanced traffic scenarios, several proposals were made: In [7], a dynamic bandwidth allocation algorithm called Distributed Virtual-time Scheduling in Rings (DVSR) is proposed. Each node counts

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

Authorized licensed use limited to: Carleton University. Downloaded on October 6, 2008 at 14:58 from IEEE Xplore. Restrictions apply.

the total packet arrivals and calculates a lower bound of temporally and spatially aggregated virtual time. A frame is circulated around the ring to distribute that information among all the nodes to approximate a Generalized Processor Sharing scheduler [10], which can be used by the ingress nodes to determine the per-destination fair rate of the various flows. In [8], the Distributed Bandwidth Reallocated in Rings algorithm is proposed. Each node uses local and remote information to compute a per-destination fair rate by transmitting the results to downstream nodes. Finally, in [9] a distributed scheduler named Virtual Source Queuing is proposed which guarantees a fair access to all the incoming flows to ingress nodes using a simple feedback scheme. However, all of these proposed schemes necessitate significant modifications to the current RPR protocol. The next section presents an improved algorithm requiring only simple modifications to the current fair rate advertisement mechanism of RPR.

III. IMPROVED FAIRNESS ALGORITHM

We propose an improved algorithm to the fairness domain determination mechanism to detect the congested/uncongested toggling state that creates the traffic oscillations. In [1] Table 10.10, the advertised fair rate state machine will advertise a full link rate if the tail node is not itself congested, if it detects downstream congestion and if the amount of forwarded traffic from upstream nodes is less than the received fair rate. As shown in Figure 4 and Figure 7, this simple algorithm creates tail node induced permanent oscillations that degrades the average throughput. The improvements avoid this congested/uncongested toggling by letting the tail node advertise the maximum rate at which the upstream nodes can send traffic through the tail node link. This is done by detecting that the congestion comes from too much upstream traffic and not because the tail node itself wants to send more locally sourced traffic than its fair share and that the upstream nodes would consume more bandwidth than available thus creating an unnecessary congested condition. The tail node then advertises the maximum rate to avoid this toggling condition. The proposed modifications to the state machine are detailed in Table I and II. The only modifications needed are in rows 12a and 14a of the state machines. Note that we are introducing the parameter rateLowThreshold for 2-buffer configuration for this case.

MINE	IAmTail==TRUE
	lpAddRate <
	allowedRate &&
	lpFwdRate >
	lpAddRate &&
	clientBufPkts == 0
		12a	ad =
			rateLowThreshold-
			lpAddRate
			if (ad<0) ad = 0
			frame.fairRate = ad;
				M2

	--	12b	frame.fairRate = normLocalFairRate;	M2
M2	--	12c	frame.saCompact = myMacAddress; frame.ttl = MAX_STATIONS; frame.ri = Other(myRi);	SEND

TABLE I

Improved fairness algorithm (see [1], Table 10.10)

FULL	lpFwdRate >
	receivedRate
		14a	ad =
			rateLowThreshold -
			lpAddRate
			if (ad<0) ad = 0
			frame.fairRate = ad;
				F2

	--	14b	frame.fairRate = FULL_RATE;	F2
F2	--	14c	frame.saCompact = myMacAddress; frame.ttl = MAX_STATIONS; frame.ri = Other(myRi);	SEND

TABLE II

Improved fairness algorithm (see [1], Table 10.10)

IV. SIMULATION PARAMETERS

We have designed an OPNET simulation model compliant with P802.17 Draft 3.3. In particular, the implementation used in this study shapes the secondary transit queue (STQ) with shaperD.

Figure 3 and Figure 6 show the topology of the networks simulated. They are composed of six stations linked by a ring with a circumference of approximately 240km. Each hop propagation delay equals 0.2ms. The traffic flows used in all simulations are classC traffic to destination node 0 (Poisson greedy traffic with mean packet length of 444 bytes).

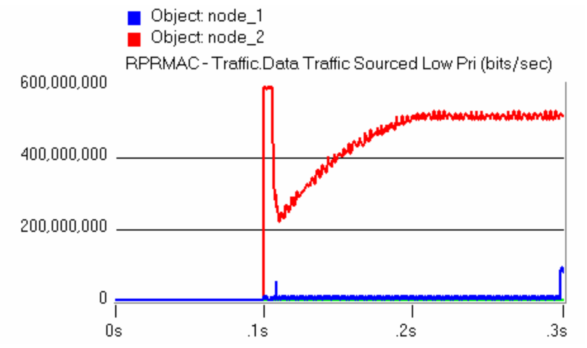


Figure 1. 2-node scenario does not create oscillations when using the conservative algorithm.

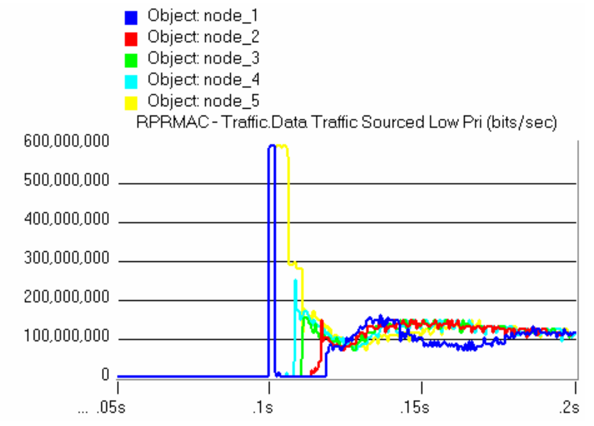


Figure 2. The conservative algorithm oscillates and settles to the fair rate within 7 RTT in the parking lot scenario.

Authorized licensed use limited to: Carleton University. Downloaded on October 6, 2008 at 14:58 from IEEE Xplore. Restrictions apply.

The conservative fairness algorithm tries to stabilize fairness eligible traffic between two target rates, low and high rate thresholds (see Figure 2, parking lot scenario where all nodes send traffic to destination node 0). All the stations weights are equal, thus the fairness algorithm will reduce each station added traffic to the calculated fair rate.

Parameter	**Value**
Unreserved bandwidth	95%
Target Rates	
rateLowThreshold	85% (of unres. BW)
rateHighThreshold	95% (of unres. BW)
MAC Access delay thresholds	0.1 ms
Primary Transit Queue (PTQ)	
PTQ Size	32 KB
Secondary Transit Queue (STQ)	
STQ Size	256 KB
stqLowThreshold	32 KB
stqMedThreshold	48 KB
stqHighThreshold	64 KB
Varia	
rampCoef	16
rampCoef per decay interval	1024
lpCoef	64

TABLE III
Simulation parameters used.
Parameters not shown use P802.17 Draft 3.3 default values.

V. RESULTS ANALYSIS

A. *2-node 99%/1% scenario*

Figure 3 shows the traffic flows and their associated rate. In the case of nodes using the conservative mode, no oscillations occur when using the standard RPR algorithm (see Figure 1). This is due to the fact that the algorithm tries to reach an equilibrium point between a low and a high rate threshold. When this occurs, the algorithm stabilizes and continues to advertise this fair rate. The uncongested condition will occur if node 2 decreases significantly its add rate for a long enough period of time (depending on the ramping up coefficient). Then the cycle will start again.

In the case of nodes using the aggressive algorithm, oscillations occur because there is no equilibrium point to set into. The congested/uncongested states are triggered by the buffer occupancy threshold. Figure 4 shows these oscillations using the 2-transit buffer configuration. From 0.1s to 0.3s node 1 adds only a small amount of traffic (1% of link rate), and node 2 is greedy and sends as much traffic as possible. From 0.3s to 0.4s, we show that when node 1 becomes greedy, both nodes stabilize to the fair rate. Finally, from 0.4s to 0.5s, when node 1 does not add any traffic, node 2 can use up all available bandwidth without oscillations. The problematic part is then in the 0.1s to 0.3s time period, creating oscillations that impair the average utilization. Figure 5 shows the same scenario with the improved fair rate advertising algorithm. We see that the oscillations are reduced and node 2 can still reclaim all available bandwidth when node 1 stops adding traffic. In this case, there is a significant improvement in average link utilization, from 85% to 95%.

B. *Parallel Parking Lot Scenario*

Figure 6 shows the Parallel Parking Lot scenario traffic flows and their associated fair rate share. Figure 7 shows the tail node induced oscillations. From 0.1s to 0.2s all 5 greedy nodes transmit traffic, constrained by the advertised fair rate they receive. We observe that node 5 rate oscillates because of the toggling in the advertised fair rate sent by node 4, the tail node. In this simulation, we show that when node 4 stops adding traffic, the rates of all remaining nodes stabilize to the fair rate. In particular, node 5 is able to reclaim all available bandwidth on links 5 and 4, exploiting the spatial reuse of RPR. Figure 8 shows the same scenario with the improved fair rate advertising algorithm. We note that the oscillations are gone and node 5 can still reclaim the bandwidth when node 4 stops adding traffic. In this case, there is a slight improvement in the average link utilization, from 80% to 84%.

VI. CONCLUSION

The RPR fairness mechanism suffers from severe permanent oscillations under certain conditions. In this paper, we propose an improved algorithm to the current fair rate advertisement mechanism of RPR to prevent tail node induced oscillations. The improved algorithm detects the conditions which normally trigger the congested/uncongested toggling state causing the oscillations. The tail node then advertises the maximum rate which avoids triggering congestion and oscillations. Our simulation results show that the improvements work under various conditions and can increase significantly the average throughput of RPR without major modifications to the protocol. Future work will include a more extensive analysis including various traffic patterns.

REFERENCES

[1] P802.17 RPR Draft 3.3. http://grouper.ieee.org/groups/802/17/

[2] F. Davik, M. Yilmaz, S. Gjessing and N. Uzun, “IEEE 802.17 Resilient Packet Ring Tutorial”, IEEE Communications Magazine, pp. 112-118, March 2004.

[3] F. Yuan et al., “Performance Analysis for Resilient Packet Rings with Single Transit Buffer”, IEEE ICT 2002, June 2002.

[4] M.J. Francisco et al. “A Comparison of Two Buffer Insertion Ring Architectures with Fairness Algorithms”, ICC03, May 2003.

[5] Y. Robichaud, C. Huang, J. Yang and H. Peng, “Access Delay Performance of Resilient Packet Ring under Bursty Periodic Class B Traffic Load”, in proceedings of International Conference on Communications (ICC04), Paris, June 2004.

[6] P. Yuan, V. Gambiroza and E. Knightly, "The IEEE 802.17 media access protocol for high-speed metropolitan-area resilient packet rings", IEEE Network, vol 18, no 3, pp 8-15, May-June 2004.

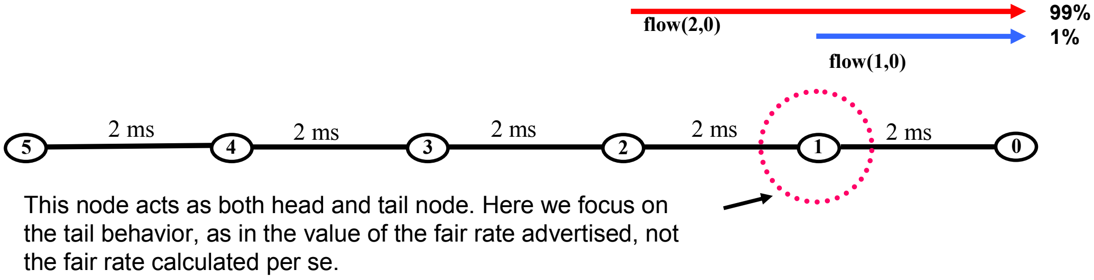
[7] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, E. Knightly, "Design, Analysis, and Implementation of DVSR: A Fair, High Performance Protocol for Packet Rings", IEEE/ACM Transactions on Networking, vol 12, no 1, pp 85-102, Feb 2004.

[8] P. Yue, Z. Liu and J. Liu, “High performance fair bandwidth allocation algorithm for resilient packet ring”, AINA’03, March 2003.

[9] H. Tang, I. Lambadaris, H. Mehrvar and J. Talim, “A New Access Control Scheme for Metropolitain Packet Ring Networks”, to appear in proceedings of the 10th IEEE Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, Dallas, 29 November 2004.

[10] A.K. Parekh and R. G. Gallagher, “A generalized processor sharing approach to flow control in integrated services networks: the single-node case”, IEEE/ACM Tran. on Networking, vol 1, no 3, June 1993.

Authorized licensed use limited to: Carleton University. Downloaded on October 6, 2008 at 14:58 from IEEE Xplore. Restrictions apply.



This node acts as both head and tail node. Here we focus on the tail behavior, as in the value of the fair rate advertised, not the fair rate calculated per se.

Figure 3. 2-node 99%/1% scenario with tail node induced oscillations.

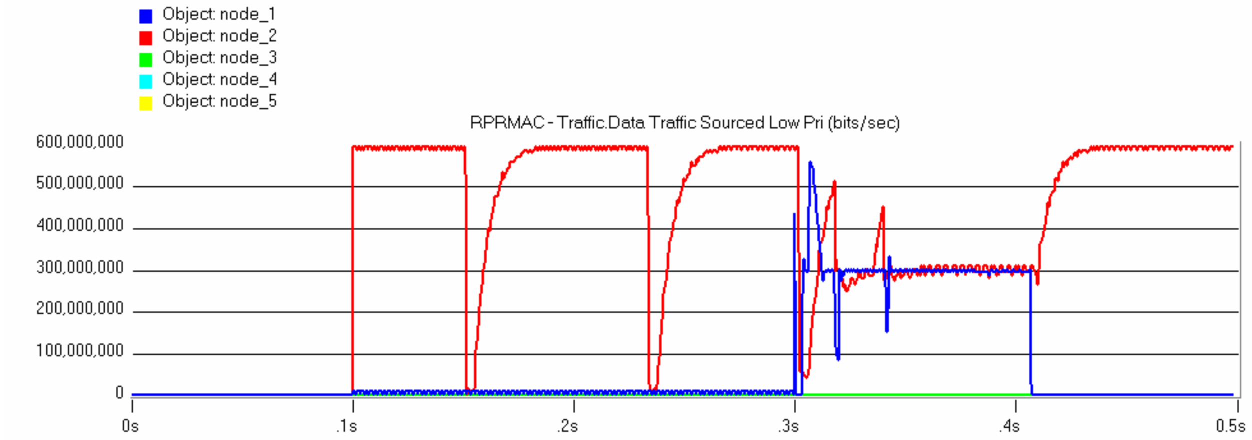


Figure 4. Simulation results obtained with 2-transit buffer configuration and aggressive mode showing severe, permanent oscillations.

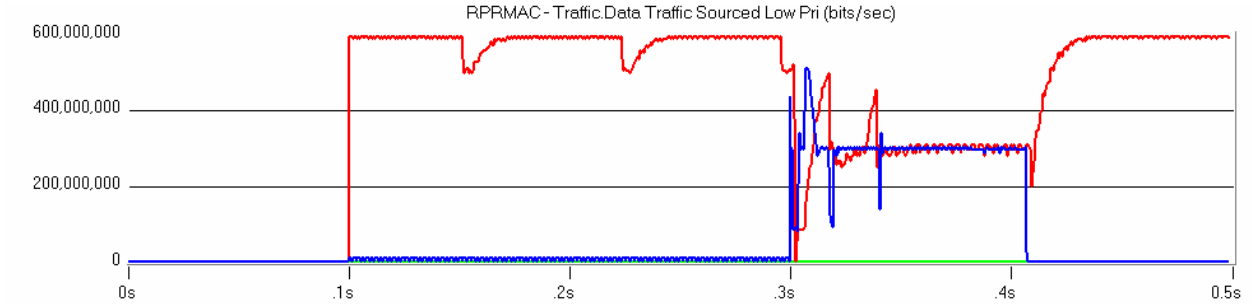
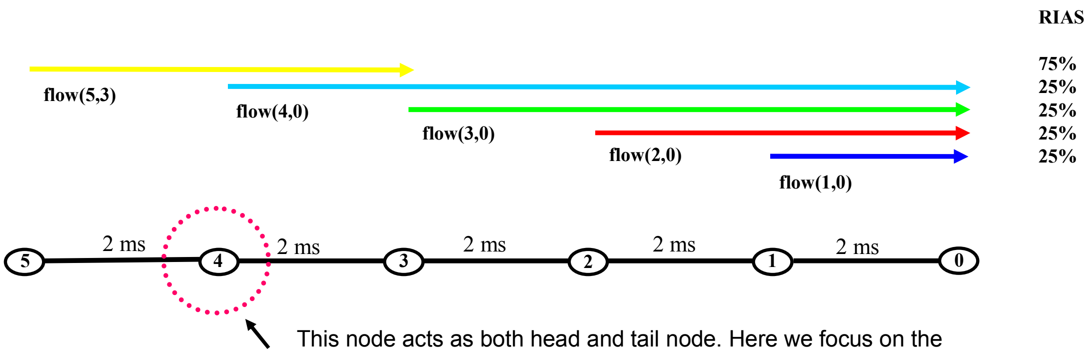


Figure 5. Simulation results obtained with 2-transit buffer configuration and aggressive mode using the improved fairness algorithm showing reduced oscillations.

Authorized licensed use limited to: Carleton University. Downloaded on October 6, 2008 at 14:58 from IEEE Xplore. Restrictions apply.



This node acts as both head and tail node. Here we focus on the tail behavior, as in the value of the fair rate advertised, not the fair rate calculated per se.

Figure 6. Parallel Parking Lot scenario. Node 4 becomes the tail of the fairness congestion domain which triggers oscillations in the traffic sent by Node 5.

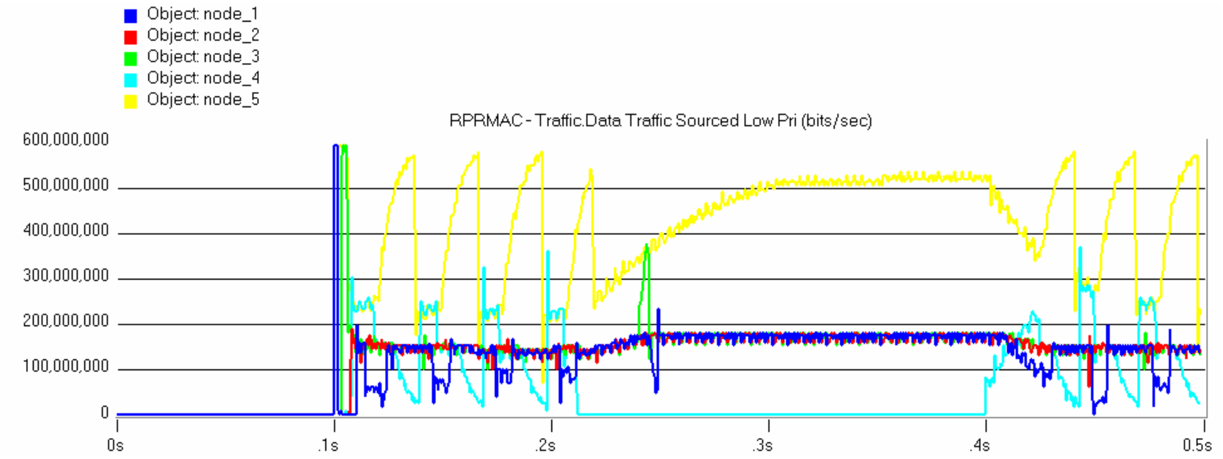


Figure 7. Simulation results showing the severe permanent oscillations when using 1-transit buffer configuration and conservative mode.

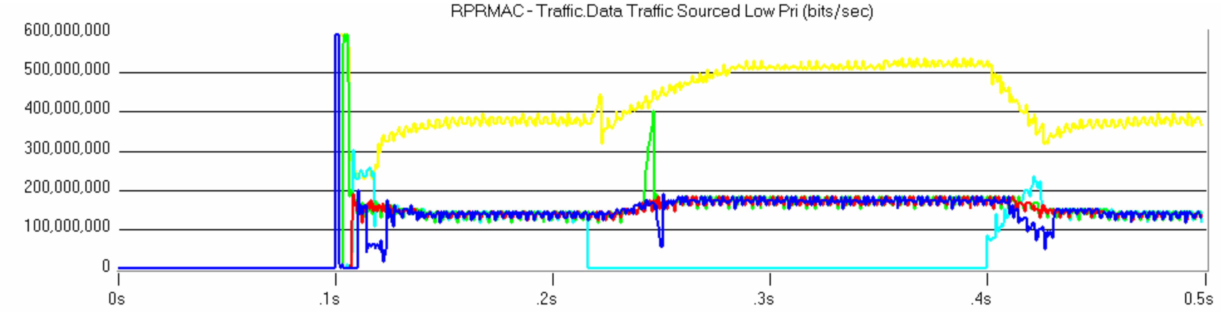


Figure 8. Simulation results obtained with 1-transit buffer configuration and conservative mode using the improved fairness algorithm removing oscillations.

Authorized licensed use limited to: Carleton University. Downloaded on October 6, 2008 at 14:58 from IEEE Xplore. Restrictions apply.