A Comparison of Two Buffer Insertion Ring Architectures with Fairness Algorithms

Mark Joseph Francisco, Fengjie Yuan, Changcheng Huang
{mfranc, fyuan, huang} @ sce.carleton.ca
Advanced Optical Networks Laboratory
Department of Systems and Computer Engineering, Carleton University

Harry Peng
hpeng@nortelnetworks.com
Nortel Networks

Abstract – **Buffer Insertion Rings (BIR) are known to provide higher throughputs than other competing ring technologies. With the introduction of spatial reuse, MANs and LANs are at a greater advantage of maximizing bandwidth efficiency. Spatial reuse introduces the concept of congestion and Fairness Algorithms are needed to police the fair access of the low priority traffic on the ring. Two architectures are studied in this paper, Mono Transit Buffer (MTB) and the Dual Transit Buffer (DTB). Different from earlier BIR architectures, the congestion control mechanisms studied in this paper are rate based and traffic streams are regulated using leaky buckets. It has been shown through simulations that both architectures exhibit oscillatory behavior under certain congestion conditions. MTB oscillates due to the overreaction of rate estimations, whereas DTB oscillates due to the buffer threshold settings. We show that by correctly setting parameters, oscillations can be dampened to achieve fair throughputs for all nodes contributing to the congestion.**

I. INTRODUCTION

Metropolitan Area Networks (MANs) and Local Area Networks (LANs) have seen many changes in recent years to accommodate the increased demands of its users. New technologies have been designed to provide efficient bandwidth usage over common existing technologies, such as Ethernet and FDDI rings. Access methods of rings have been extensively studied, compared and refined in [1, 2, 3, 4, 11]. With the new transmission speeds and increased necessity for bandwidth efficiency, a recent interest has been developed in spatial reuse in ring architectures, such as those described in [5, 6, 7]. Spatial reuse networks are a very attractive alternative for high speed MANs and LANs.

Spatial Reuse allows multiple simultaneous transmissions to occur at the same time, as long as bandwidth is available. The overall throughput of a network using spatial reuse can be significantly higher than that of a network without spatial reuse. The introduction of Spatial Reuse into a technology also introduces the notion of congestion, also called *starvation* in [6] and [8]. Congestion is a state a node may enter if it has not been given access to send its packets for a lengthy period of time. Congestion also leads to a problem in fairness, meaning that under a congested network, each node should have fair access to the globally shared resource, which is the ring. Fairness algorithms can be found in other ring technologies, such as MetaRing [6] and SRP [5].

In this paper, we present two BIR architectures with spatial reuse and Fairness Algorithms. The Fairness Algorithms use a backward explicit congestion notification mechanism and can be described as a type of closed loop feedback control system. Congested nodes are required to send a fair rate to the nodes contributing to the congestion, to which they adjust their transmission rates. Given that the Fairness Algorithm is a closed loop feedback control system, we show through simulations that oscillations can occur in the overall throughput, if parameters in the congestion control mechanism are incorrectly set. The Fairness Algorithm is bound by the ring delay, which is described in detail in [9]. These two ring architectures provide a new area of research, since these Fairness Algorithm are rate based, rather than slot or quota based, such as the MetaRing [6]. The two methods are described here and compared in terms of throughput and utilization.

The paper is organized as follows: Section II will describe the BIR technology and the Fairness Algorithms. Simulation results will be provided in Section III. Section IV contains an explanation of the simulation results and present improvements to the results. Conclusions will be drawn in Section V.

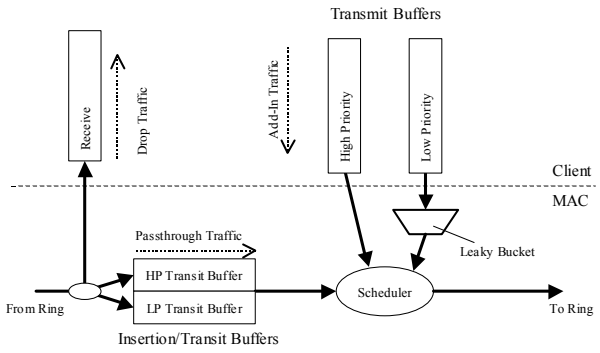

Fig. 1. A Typical BIR MAC Architecture

II. Buffer Insertion Ring And Fairness Algorithm

The Buffer Insertion Ring Architecture presented in this paper is similar to that found in [5] and [8] and is summarized here. The ring is a full-duplex Buffer Insertion Ring with counter rotating rings. The clockwise ring is denoted as the O-ring (outer ring) whereas the counter clockwise ring is called the I-ring (inner ring).

Fig. 1 illustrates the general MAC architecture of the BIR studied in this paper. Only one direction of traffic is shown here, but the reverse is similar with the exception of the Transmit buffers and the drop buffer.

In most MANs and LANs, different priority of service is provided for the unified network. Voice data can be classified as High Priority traffic, which suffers no loss and a low bounded delay. Low priority traffic can be attributed to other non-essential or non-critical data, which may suffer a longer delay and even packet loss, but still meets the application requirements. The BIR presented here show two transmit buffers in the Client side. A High Priority and Low Priority Transmit Buffer exists. A leaky bucket is used to regulate the transmission of traffic of the Low Priority Transmit Buffer. The leaky bucket has an adjustable leak rate. If the transmit buffer is full, new packets are dropped. The transit buffers can be configured such that it can have one transit buffer, or two transit buffers.

The spatial reuse option used by this network creates a necessity for a Fairness Algorithm to allow all nodes on the network to gain equal access to the ring. It is important to note that only Low Priority Traffic is subjected to the Fairness Algorithm. When congestion is detected, the Congestion Notification Message is created and is sent upstream to the nodes contributing to the congestion. The *Congestion Notification Message* contains a field called "*fair rate*". As the message propagates upstream, the receiving node adjusts their leaky bucket leak rates to the *fair rate*. Fair rate calculations are made every Sample Period, which is typically set to 100 μs. The method of calculating the *fair rate* is described below.

The following two subsections describe the Mono Transit Buffer and the Dual Transit Buffer designs. The scheduling algorithm, the congestion detection mechanism and the calculation of the *fair rate* are described.

A. Mono Transit Buffer

Fig. 1 shows a typical BIR MAC architecture which is similar to that presented in [11]. In the MTB architecture, a single transit buffer amalgamates Low Priority and High Priority traffic into one mixed traffic buffer on the ring. The traffic in that buffer has the utmost priority. The advantage of the MTB configuration is that it simplifies the hardware implementation since it is less complex, but also means that traffic on the ring, which is a mix of High Priority and Low Priority traffic, can block the transmitted High Priority and Low Priority traffic waiting to gain access onto the ring. The MAC scheduler sends traffic in the following order:

1) Packets sent from the Mixed Transit Buffer.

2) Packets sent from the High Priority Transmit Buffer.

3) Packets sent from the Low Priority Transmit Buffer.

Congestion, in the MTB architecture, can be triggered in two ways:

1) *If the link usage exiting a node exceeds a threshold.* In this case, the threshold is typically set to 95% of the link's unreserved bandwidth. The remaining bandwidth is reserved specifically for High Priority Traffic. The congestion status is lifted when the link usage falls below the threshold.

2) *If the Head of Line Timer expires.* The Head of Line Timer indicates whether a packet at the head of the Low Priority Transmit Buffer has waited a fixed length of time and is deemed unfair. If it has waited too long, the timer expires, and the node is congested.

When congestion occurs, the initial fair rate is calculated by (in the case of no high priority traffic)

$$\rho_i = \frac{CU_t}{\omega} \qquad (1)$$

ρ_i is the fair rate at node i, where node i is the congested node. C is the link rate of the ring. U_t is the link utilization. ω is the number of nodes contributing to the congestion. The rate calculation requires that the node keeps track of the contributing nodes in ω. As traffic passes through the node, it is counted. The count is reset every Sample Period.

B. Dual Transit Buffer

Fig. 1 actually shows the Dual Transit Buffer (DTB) configuration. A similar architecture has been presented in [5] and is described below. Two transit buffers exist for Low Priority and High Priority Traffic. As the traffic arrives from the ring, the traffic is either dropped if destined for that node, or placed in the HP or LP Transit buffer, according to the traffic classification. The dual transit buffer allows High Priority traffic to have the utmost priority and is never blocked by Low Priority traffic passing through the node. The disadvantage is that the LP Transit Buffer must be large to allow traffic bursts from all the other buffers and it governs the amount of High Priority traffic. This will make the hardware design more difficult and more parameters for traffic engineering. Different from MTB, the scheduling algorithm will be as follows:

1) Packets sent from the High Priority Transit Buffer.

2) Packets sent from the High Priority Transmit Buffer.

3) Packets sent from the Low Priority Transmit Buffer.

4) Packets sent from the Low Priority Transit Buffer.

Congestion can be triggered *if the LP Transit Buffer depth exceeds a threshold.* Two thresholds are defined: The LO_THRESHOLD and the HI_THRESHOLD. When the LP Transit Buffer depth exceeds the LO_THRESHOLD, then the node is congested. If the buffer depth exceeds HI_THRESHOLD, then the LP Transit traffic has been starved and transmitted LP traffic is blocked to let the LP Transit buffer gain access to the ring. Congestion is lifted when the buffer depth falls below LO_THRESHOLD. DTB can achieve 100% link utilization since passthrough Low Priority Traffic can be buffered on the ring.

When congestion occurs, the fair rate is calculated by monitoring the add-in traffic sourced by the congested node. This rate is called *my_usage* and is described in detail in [5]. *My_usage* is set as the fair rate in the congestion notification message. The heuristic logic underlining this algorithm is that all nodes should have the same transmitted throughput. If the throughput of a congested node drops, all other nodes should follow.

III. Simulation Scenario

Simulations of a 16-node hub network were conducted in OPNET Modeler. The scenario consisted of 15 nodes sending traffic on the I-ring to Node 0, the Hub. This scenario was called the Hub Scenario and tests the last node's response to an overloaded network, as seen in [10]. If all nodes send more than ρ_i in (1), given U_t is 1 for DTB, Node 1, the node before the hub, will be congested and the fairness algorithm would activate.

The network was overloaded to 150% to drive the congested node into deep congestion. This stressful scenario was for pathological purpose. The network was not unstable because overflowed packets were dropped at transmit buffers. Given the link rate being OC192 (9.953 Gbps), each contributing node (Node 1 to Node 15) sent 1 Gbps of Low Priority traffic starting at 0.01 seconds. Packets were distributed trimodally to emulate real traffic profile, meaning 60% of the packets were 64 bytes, 20% were 512 bytes and 20% were 1518 bytes. The distance between the nodes was 15 km, roughly 70 μs delay. The total ring delay was 1.12 ms, for a ring circumference of 240 km. The Sample Period was set to 100 μs, meaning congestion detection and fair rate calculations are made at this interval. Observations were made on Node 1, since this node is the congested node before the hub.

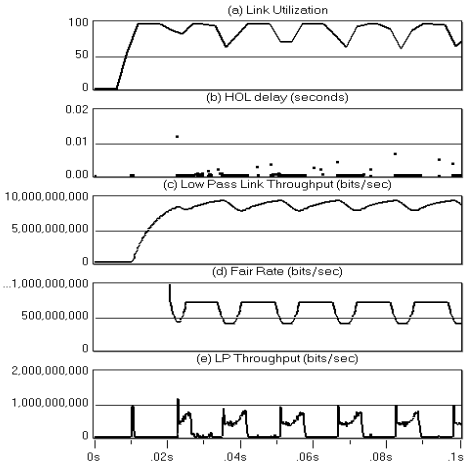

Fig. 2. Results for MTB default scenario

A. Mono Transit Buffer

The first scenario tested was the MTB configuration and will be referred to as the MTB Default Scenario. The *HOL Delay Threshold* was originally set to 10 ms and the *Link Utilization* was set to 95%. Fig. 2 shows the simulation results for this scenario.

The graphs in Fig. 2 are explained below:

a) *Link Utilization*: This graph shows the actual utilization of the link as a percentage of the link's capacity.
b) *HOL Delay*: This graph shows the HOL delay. This is used to determine congestion when the timer exceeds the HOL Delay Threshold, which is set to 10 ms in the default scenario.
c) *Low Pass Link Throughput*: This graph shows the total transit and transmit traffic byte count, run through a low pass filter. If the value exceeds 95% of the link bandwidth, congestion is detected.
d) *Fair Rate*: This graph shows the fair rate advertised from Node 1 to the upstream nodes contributing to the congestion.
e) *Low Priority Throughput*: The actual throughput of transmitted Low Priority traffic by Node 1.

Note that all five graphs share the same timescale. The primary observation that can be made from Fig. 2 is that the Link Utilization (Fig. 2 a), the Fair Rate (Fig. 2 d) and the LP Throughput (Fig. 2 e) oscillate. Congestion is triggered by the HOL Delay exceeding the threshold at 0.02 seconds (Fig. 2. b) and subsequent oscillations are caused by the Low Pass Link Throughput (Fig. 2 c) exceeding the maximum bandwidth threshold coupled with the oscillation of the Fair Rate (Fig. 2 d). An explanation of the oscillation will be made in Section IV.

B. Dual Transit Buffer

An identical scenario was tested replacing all MTB nodes with DTB nodes, called the DTB Default Scenario. Since DTB does not use the HOL Threshold as a measure of congestion, the size of the LP Transit Buffer thresholds were configured to detect congestion. The LO_THRESHOLD, which detects the congestion, was set to 64 kilobytes and HI_THRESHOLD was set to 128 kilobytes. Fig. 3 shows results from this scenario.

The graphs in Fig. 3 are similar to those in Fig. 2, which the exception to Fig. 3 b: *LP Transit Buffer Usage* is the depth of the LP Transit Buffer.

Note that all four graphs in Fig. 3 share the same timescale. The oscillations are the primary observation that can be made from Fig. 3. The period of the oscillations is tighter than those of the MTB Default Scenario (Fig. 2). Oscillations are caused by a toggling of the Congestion status. A further examination will be made in Section IV part B.

IV. Results Analysis

In Section III we saw that oscillations occur for both the MTB and the DTB default scenarios. We will see that, for both default scenarios, the way we can control and remove these oscillations are different for the two architectures. For MTB, increasing the interval of advertisements can dampen oscillation. On the other hand, reducing the sensitivity of the

congestion threshold for DTB can lift oscillation. In the next two subsections, we will go into depth why each scenario oscillated and will show how we can dampen the oscillation by intuitively setting specific parameters.

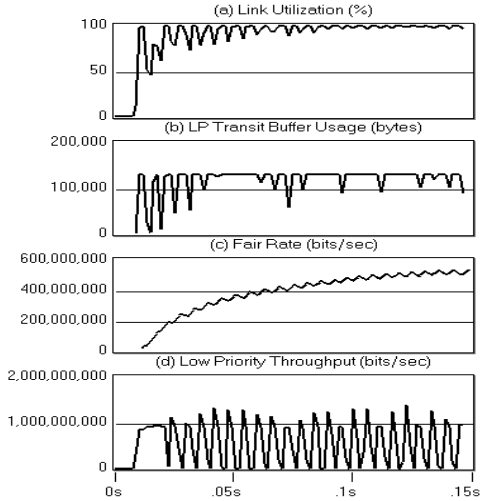

Fig. 3. Results for the DTB Default Scenario

A. Mono Transit Buffer

The closed loop feedback system of the Fairness Algorithm implies that it is bounded by the lag between the time the fair rate is defined and the time it is taken into effect. If the congested node does not allow enough time to make a solid calculation of the fair rate, then it will either miss or overshoot the optimal fair rate value.

In Fig. 2, we saw that the oscillation is started by the congestion being triggered by the HOL Timer. When Node 1 is congested, the fair rate is calculated based on the number of nodes contributing to the congestion at Node 1. This calculated rate missed the true fair rate.

The fair rate that should have been calculated would be 9.95 Gbps / 15 nodes = 663 Mbps per node. The rate that was calculated was below 500Mbps due to the low pass nature of the calculation, which means that the advertised rate missed by more than 160 Mbps. Since it missed, the feedback control system had to compensate, but in the case of the MTB Default Scenario, not enough time was given to make a clear assertion of the fair rate.

Fig. 4 illustrates some end-to-end times from Node 1. For a congestion notification message to reach Node 2, 70 µs would need to pass. A full Round Trip Time (RTT) would be needed for Node 1 to see the traffic changes after Node 2 adjusts its leaky bucket leak rate to the fair rate. The same observation can be made of the nodes upstream. In the MTB default scenario, Node 1 is recalculating the fair rate 100 µs after sending the first congestion notification message. This allows only enough time for Node 2 to see the congestion notification message and adjust its leaky bucket leak rate. Node 1 will only see this new throughput until 140 µs. Thus, Node 1 is not giving enough time for Node 2 to recalculate the fair rate, but it also is not giving enough time for all nodes in the congested span.

For Node 1 to make an accurate fair rate estimation, it would need the RTT of the furthest contributing node before calculating the next fair rate. In this case, the RTT of Node15 would be 2 * 980 µs, which is 1960 µs. By decoupling the fair rate calculation with the congestion notification generation, then enough time would be given for Node 1 to make an accurate fair rate estimation. We introduce the concept of the Advertisement Interval. The Congestion Notification Message is sent every Advertisement interval, instead of every Sample Period. By setting the Advertisement Interval to 2ms, the oscillations found in the MTB Default Scenario should dampen quickly.

Fig. 5 shows improvements to the MTB Default Scenario. The Advertisement Interval was set to 2ms, which allowed enough time for Node 1 to make a more accurate fair rate estimation. Notice that the throughput (Fig. 5 e) is still blocked from 0.01 seconds to 0.025 seconds, because link utilization is 100% (Fig. 5 a). After the HOL Timer expires, the traffic comes to a steady state and the link utilization stabilizes at 0.05 seconds.

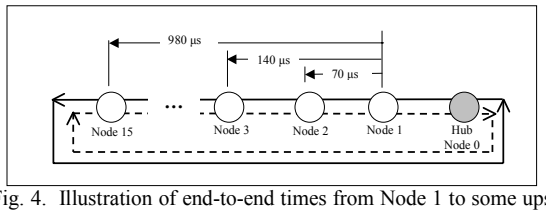

Fig. 4. Illustration of end-to-end times from Node 1 to some upstream nodes.

B. Dual Transit Buffer

In Fig. 3, the buffer size was set to 1 Megabyte and LO_THRESHOLD was set to 64 kilobytes, while HI_THRESHOLD was set to 128 kilobytes. When the buffer depth exceeded LO_THRESHOLD, Congestion Notification Messages are generated and sent upstream, but not enough time was allocated before the buffer depth reached HI_THRESHOLD, at which point the locally transmitted traffic was blocked. Blocking the transmitted traffic and flushing the LP Transit Buffer toggles the congestion status, which was the source of the oscillation.

On an OC192 link, the buffer can be filled from LO_THRESHOLD to HI_THRESHOLD in approximately 50 µs. The buffer size should be large enough to allow the congestion notification message to take effect, before blocking the transmitted traffic.

By setting the thresholds very large to accommodate the round trip time, oscillation can be avoided. The exact buffer size needed depends on the add-in traffic throughput at the congested node (this decides how much bandwidth is left for transit path) and the round trip time.

Fig. 6 shows an improvement to the DTB Default Scenario. The LO_THRESHOLD was set to 450 kilobytes and HI_THRESHOLD was set to 900 kilobytes. The throughput (Fig. 6 d) shows no oscillations at 0.1 seconds. The LP Transit buffer depth oscillates (Fig. 6 b) in the first 0.03 seconds, but comes to a steady state, which is congested. The Fair Rate (Fig. 6 c) comes to a steady state quicker than that of the DTB Default scenario, which forces

the LP Throughput and the overall link utilization to achieve a steady state earlier. An assumption can be made that if the LO_THRESHOLD was larger, steady state can be achieved much faster.

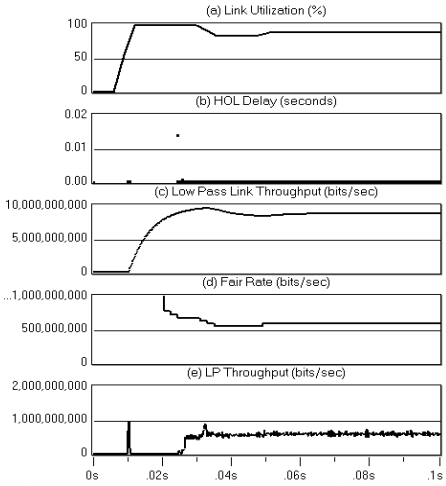

Fig. 5. Improvements to the MTB Default Scenario

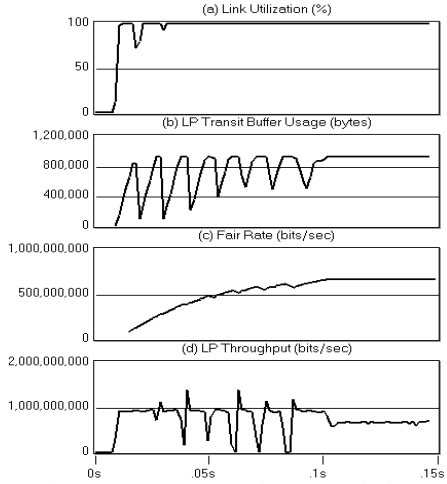

Fig. 6. Improvements to the DTB Default Scenario

V. CONCLUSION

In this paper, two buffer insertion ring architectures have been presented. The first examined was the mono transit buffer (MTB) architecture and the other was the dual transit buffer (DTB) architecture. The two architectures had similar Fairness Algorithms, but calculated congestion and the fair rate in different ways. Simulations showed that under the same HUB scenario, the Fairness Algorithm, behaved in a somewhat similar manner. Our simulations showed that both configurations oscillated when responding to a step increase of traffic.

The MTB configuration oscillated because of the overreaction of the rate estimation. Simulations showed that if the Advertisement Interval was increased to the RTT of the further node in the congestion span, the oscillations were dampened.

The DTB configuration oscillated because of the Fairness Algorithm's sensitivity to the congestion thresholds in the LP Transit Buffer. It was found that if the LO_THRESHOLD was increased, oscillations could be dampened.

REFERENCES

[1] M. Karol, R. Gitlin, "High Performance Optical Local and Metropolitan Area Networks: Enhancements of FDDI and IEEE 802.6 DQDB", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 8, pp. 1439-1448, October 1990.

[2] M. A. Marsan, F. Neri, "Performance Comparison of Four MAC Protocols for Integrated MANs", *Telecommunications Symposium, 1990. ITS '90 Symposium Record, SBT/IEEE International*, pp. 104-112, September 1990.

[3] H.R. Muller et al., "DQMA and CRMA: New Access Schemes for Gbit/s LANs and MANs", *INFOCOM '90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. The Multiple Facets of Integration, Proceedings IEEE*, vol. 1, pp. 185-191, June 1990.

[4] I. Cidon et al., "Improved Fairness Algorithms for Rings with Spatial Reuse", *IEEE/ACM Transaction on Networking*, Vol. 5, No. 2, pp. 190-204, April 1997.

[5] D. Tsiang, G. Suwala, "The Cisco SRP MAC Layer Protocol", *Internet Engineering Task Force (IETF) Request for Comments 2892*, August 2000.

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

[7] S.Breuer, T. Meuser, "Enhanced Throughput in Slotted Rings Employing Spatial Slot Reuse", *INFOCOM '94 Networking for Global Communications, 13th Proceedings IEEE*, vol. 3, pp. 1120-1129, June 1994.

[8] R. Simha, Y. Ofek, "A Starvation-free Access Protocol for a Full-duplex Buffer Insertion Ring Local Area Network", *Ninth Annual International Phoenix Conference on Computers and Communications 1990, Proceedings*, pp. 531-538, March 1990.

[9] D. A. Schupke, "Packet Transfer Delay of the SRP Ring", *26th Annual IEEE Conference on Local Computer Networks 2001, Proceedings of LCN*, pp. 464-465, November 2001.

[10] G. Anastasi, L. Lenzini, B. Meini, "Performance Evaluation of a Worst Case Model of the MetaRing MAC Protocol with Global Fairness", *Proceedings of International Zurich Seminar on Digital Communications*, February1996.

[11] W. Bux, M. Schlater, "An Approximate Method for the Performance Analysis of Buffer Insertion Rings", *IEEE Transaction on Communications*, vol. COM-31, no.1, pp. 50-55, January 1983.