Best-Effort Bandwidth Reservation in High Speed LANs using Wormhole Routing

Bruce Kwan Po-Chi Hu Nicholas Bambos Leonard Kleinrock *

University of California at Los Angeles

Los Angeles, CA 90024-1594

Hong Xu

Cisco Systems

San Jose, CA 95134-1706

Joe Touch †

USC/Information Sciences Institute

Marina del Rey, CA 90292-6695

March 22, 1996

Abstract

Current quality of service schemes require switching nodes to identify and manipulate packets within large switch buffers. These methods are effective for networks with switching nodes that contain sufficient resources, such as in ATM networks. However, in wormhole routing networks like Myrinet, the emphasis on providing low latency and high link speeds at a low cost precludes the use of intelligent switches with large buffers. In this paper, we present a simple scheme that provides "best effort" bandwidth reservation to delay sensitive traffic in a wormhole routing LAN. The idea behind this scheme is that low priority messages are segmented into smaller packets before transmission into the network while high priority messages are left unsegmented. Switches employ round robin scheduling to resolve streams contending for the same output port. Consequently, larger packets are given more bandwidth than segmented traffic. We demonstrate this idea via simulation and present results relating the optimal low priority packet length to the distance the packet must travel to reach its destination.

*This work was supported by the Advanced Research Projects Agency, ARPA/CSTO, under Contract DABT63-93-C-0055 "The Distributed Supercomputer Supernet- A Multi Service Optical Intelligent Network"

†This work is supported by the Advanced Research Projects Agency through Ft. Huachuaca contract #DABT63-93-C-0062 entitled "Netstation Architecture and Advanced Atomic Network". The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Army, the Advanced Research Projects Agency, or the U.S. Government.

1 Introduction

Switches in wormhole routing networks (related to cut-through routing) forward arriving bytes before the entire packet has been received, provided the desired output port is available [KK79, DS86, NM93, AV94, DG92, FRU92, HK95, KD91]. This technique greatly reduces propagation delay. Thus message latency is extremely low under light traffic conditions. In addition, delay does not depend on the number of hops taken as is the case in store and forward networks. This method of routing packets is popular in multicomputers [AS88] and is appearing in the LAN arena through products such as Autonet [RS91] and Myrinet [BCF+95]. Myricom's Myrinet is used in the Supercomputer Supernet (SSN) project [KGB+95],which is a joint research project between UCLA, Jet Propulsion Laboratory, and Aerospace Corp, being funded by the Advanced Research Projects Agency. The project goal is to build a network that supports multimedia real-time traffic, distributed cluster computing, and traditional datagram packets. The network is composed of electronic LAN clusters connected via an optical backbone. One of its key purposes is to support distributed computing by a high speed flexible interconnection of supercomputers and workstation clusters. The LAN clusters are connected using Myrinet hardware. The choice of Myrinet is governed by its high throughput (640 Mb/s) and low latency capabilities inherent in its wormhole routing technology.

An important aspect of the SSN project is the support of bandwidth reservation for high priority traffic across both LAN and MAN distances. This is a necessary element of high speed networks that carry delay sensitive data. To effectively support distributed applications, communication overhead must be kept to a minimum. Similarly, real time multimedia data such as video and voice requires low latency performance. If messages arrive after a certain deadline, they are useless at the destination and must be dropped. To avoid large packet "losses", network resources must be used properly to provide adequate service. In addition, this improved service for high priority traffic must be accomplished while continuing to provide adequate service to low priority datagram traffic.

Recently, work has been done in this area, specifically for ATM networks. Methods include statistical multiplexing, weighted fair queueing, and jitter control algorithms [Par94, PG93, Gol90, VZF91]. Generally, these schemes require intelligent switching nodes and sufficient buffering to identify and manipulate packets as they flow through the network. Each of these methods are effective in providing performance guarantees for cell-based networks such as ATM networks.

However, these algorithms do not translate well to current implementations of wormhole routing networks. Wormhole routing LANs, such as Myrinet, provide very low latency and high link speeds at the expense of control of the packets after entering the network. Indeed, Myrinet switches are not capable of complex operations and their buffers are minimal, only providing enough space to support simple backpressure flow control. Consequently, implementation of the quality of service schemes mentioned above is not possible in current wormhole routing LANs. In fact, any intelligent resource allocation for wormhole routing networks

must be coordinated at the end hosts.

This paper presents a simple bandwidth reservation scheme suitable for wormhole routing LANs. This scheme has properties similar to "best effort" methods [Cla88] in that it does not provide guaranteed throughput and delay performance but adds the functionality of preferential bandwidth allocation. It does this without the overhead of using traditional connection-oriented approaches (ATM signaling). Rather, it keeps the best-effort (connectionless) feature of wormhole routing. This work is primarily aimed at supporting distributed applications on clusters of high performance workstations and supercomputers where occasional short bursts of data must be delivered with minimal delay. The results are also suitable for real-time multimedia data that can withstand some degree of loss (i.e. voice). However, traffic that requires strict delay bounds must employ a centralized quality of service reservation scheme in conjunction with our approach. Efforts to develop a centralized scheme are underway at both USC/ISI and at UCLA [GKK+96].

In this early report, simulation experiments are used to provide insight into the dynamics and performance of this bandwidth reservation scheme. One result relates the optimal low priority traffic segment size with the number of hops the message must travel to reach its destination. Due to lack of space only key ideas and results are presented here. A full treatment of the subject including analytical models for performance evaluation is in preparation [KHB+96].

The remainder of the paper is organized as follows: section 2 gives a description of the simulator and traffic model, section 3.1 provides examples of how packetization can be used to reserve bandwidth and give priority to a specific traffic stream, section 3.2 presents a study on the effects of segment size and traffic load, section 3.3 describes the impact of this scheme on low priority traffic, section 4 describes how introducing priority at the switching nodes affect performance, and section 5 provides a brief comment on issues concerning implementation. Conclusions and further research ideas are then given in sections 6 and 7.

2 Experiment Design

We use a simulator, developed for the SSN project, written in a C-based parallel programming language called Maisie [BaCG+96]. To accurately model wormhole routing, the simulator performs discrete byte (flit) simulation. The simulator can model wormhole routing networks with any topology of host interfaces and electronic switches.

Each switch has a small input port slack buffer (80 bytes in actual Myrinet switch) that is used to support the backpressure link level flow control scheme. When the buffer fills up to a certain threshold, a *STOP* control signal is sent back along the incoming link and is propagated all the way to the source if necessary. A *GO* symbol is transmitted after the buffer level drops below another threshold.

Message stream arrivals to a host follow a Poisson distribution. Host messages that are not immediately

transmitted are queued at the host. Message lengths follow an exponential distribution with some mean worm length. All links between switches are assumed to be 25 meters long when accounting for propagation delay. The simulator can segment worms into fixed sized packets to support the segmentation scheme. Priority is implemented at the host in all simulations. This means that high priority worms have non-preemptive priority over low priority worms. This current study only considers 2 levels of priority. However, the scheme presented can be extended to support multiple priorities.

Deadlock is a problem in wormhole routing networks. To avoid it, a deadlock avoidance routing scheme is used. Up/down routing is used in networks such as Myrinet and Autonet [OK92] to avoid deadlock and is implemented in the simulator. Up/down routing avoids deadlock by restricting the paths of worms to a set of deadlock-free routes.

Performance metrics of interest are message network latency and throughput. We define message latency as the time between the moment a packet enters the network and the time the head of the worm arrives at its destination. For messages that are segmented, the message latency is the time between the entrance of the first segment into the network and the time that the head of the last segment arrives at its destination. Throughput of a host is the fraction of time that the host spends transmitting bits.

Offered load is the fraction of link capacity at which a host attempts to send. When referring to load, it is a reference to the offered load at each host.

3 The Dual Priority Segmentation Scheme

3.1 The Bandwidth Reservation Effect

The basic idea behind this dual priority scheme is that low priority messages are segmented into smaller packets before transmission into the network while high priority messages are left unsegmented. To support this scheme, the switching nodes in the network use round robin scheduling to resolve streams contending for the same output port. The network's response to this type of traffic shaping is to give more network bandwidth to hosts transmitting large messages. The technique of using round robin switching with variable sized packet lengths has been addressed in the literature with respect to packet networks [Nag87, DKS89].

Below a two host example is observed. In this simple topology, two hosts transmit to a single destination (see Figure 1). It is clear how different worm length ratios between the two traffic streams can affect throughput and delay (see Figure 2 and Figure 3).

In the simulation, the hosts on the left side continuously transmit to the destination host on the right (offered load is 1.0 for each host). Message lengths follow an exponential distribution with a mean of 1000 bytes. Arrival times of messages follow a Poisson distribution. H_1 is a traffic source that requires higher priority and thus transmits without segmenting the worms into smaller packets. H_2 transmits messages after

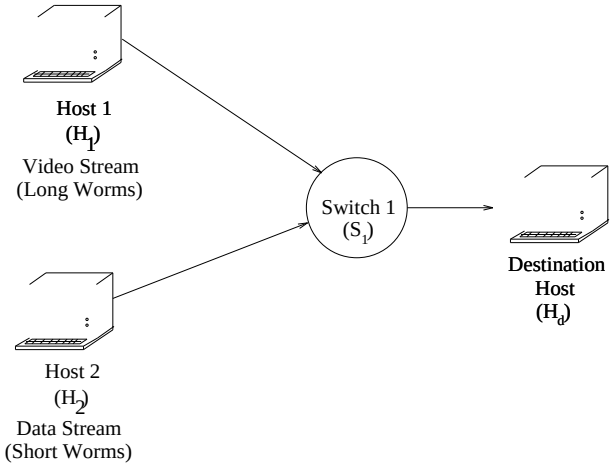

Figure 1: Simple two host topology

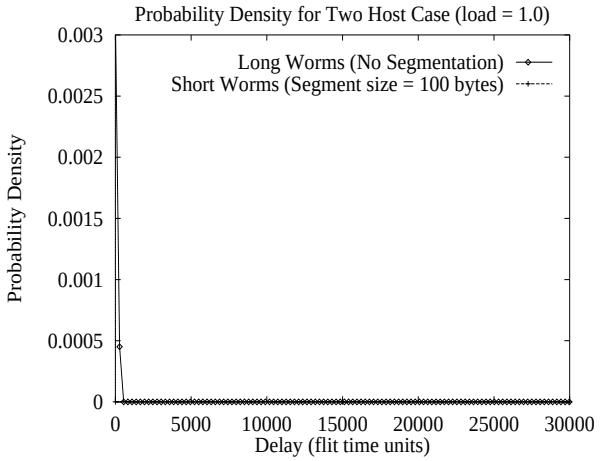

Figure 2: Probability density of random network delays occuring in the two host topology

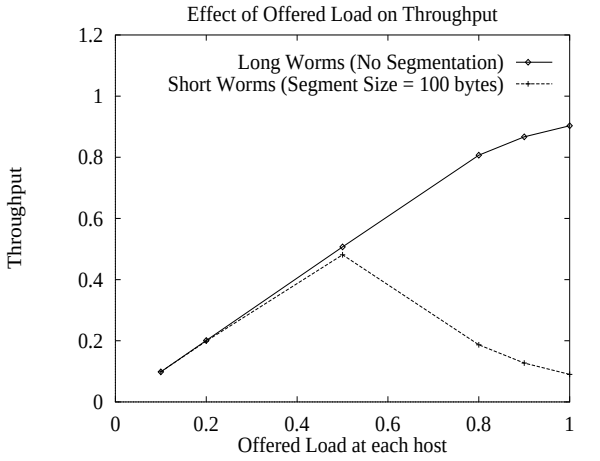

Figure 3: Throughput for two host topology

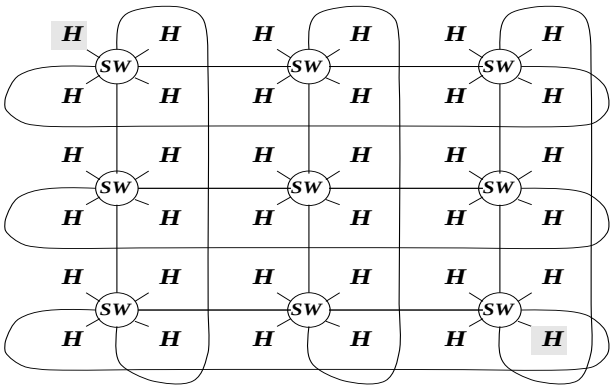

Figure 4: 3x3 torus topology

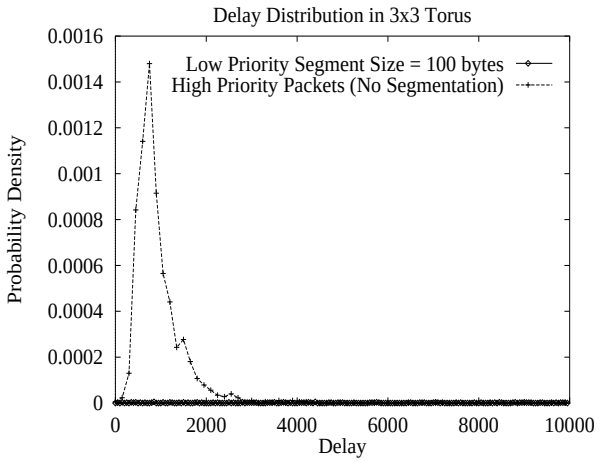

Figure 5: Delay distribution for 3x3 torus example. Average message length = 1000 bytes, Load = 1.0.

segmenting them into smaller worms of 100 bytes.

Figure 2 shows the probability density of delay for each stream. As expected, the delay distribution for the large messages is much tighter than the delay distribution for small messages. In Figure 3, throughput for each stream across different offered loads is shown. From the figure, it is observed that each host can transmit all of its messages as long as the load is below 0.5. When the load of each host rises above 0.5, the stream with long messages captures more throughput to continue to transmit all of its messages. The throughput for the host transmitting long messages begins to suffer when its load rises above 0.9 since the switch employs round robin scheduling to resolve contending streams.

Next, a LAN consisting of a 3x3 torus of switches with 4 hosts attached to each switch (see Figure 4) is examined. This topology is used to examine how worm length ratios can be used in a more complex network topology to provide improved throughput and delay. In this set of experiments, a specific host (shaded host in upper left corner in Figure 4) in the 3x3 torus topology always attempts to transmit its traffic using

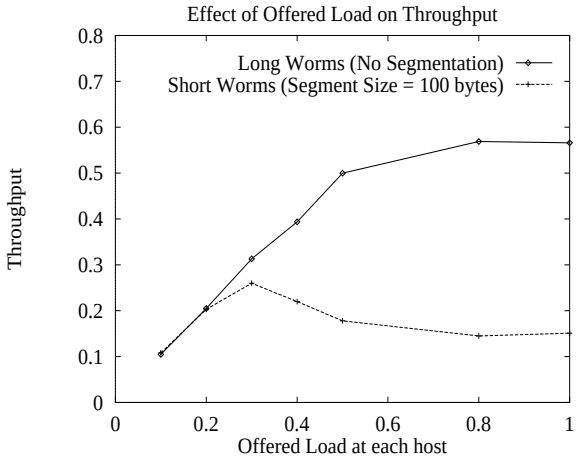

Figure 6: Throughput for 3x3 torus example

long message lengths to a different specific host (shaded host in bottom lower right corner). All other hosts transmit to each other according to a uniform destination distribution. Arrival and wormlength distributions are the same as in the two host example.

Figure 5 shows the delay distribution for the high priority stream and for a low priority stream. The results show that even as the topology complexity increases, using message length ratios continues to be an effective method of reserving bandwidth. In Figure 6, a graph of the throughput for each stream is shown. As in the simple two host case, the throughput rises for both streams up to a certain threshold (0.2 in this case) and then the throughput curves split. The high priority traffic continues to capture more bandwidth while the low priority traffic bandwidth suffers. The reason why the ratio here is clearly not 10:1 is because there are more hosts competing with the high priority traffic stream for the same host.

3.2 Low Priority Segment Size: Higher Order Blocking Effects

In this section, the effects of the segment size of low priority worms on the delay performance of high priority streams is examined. First, the light traffic region is observed in a 3x3 torus topology. All messages have uniform destination distribution and exponentially distributed lengths with a mean of 1000 bytes. Every host transmits both high and low priority traffic. The fraction of high priority traffic generated at each host is 0.3. High priority messages are sent without any segmentation. In contrast, low priority messages are segmented into fixed length worms of 100 bytes. Under light traffic conditions, blocking events are rare. Consequently, when blocking does occur, it is usually due to a single worm blocking another. We call this *1st order blocking*. Thus, decreasing the length of low priority segments lowers delay for high priority traffic. Our results in Figure 7 reflect this observation.

When the traffic load increases, blocking events become more common. The added blocking alters the

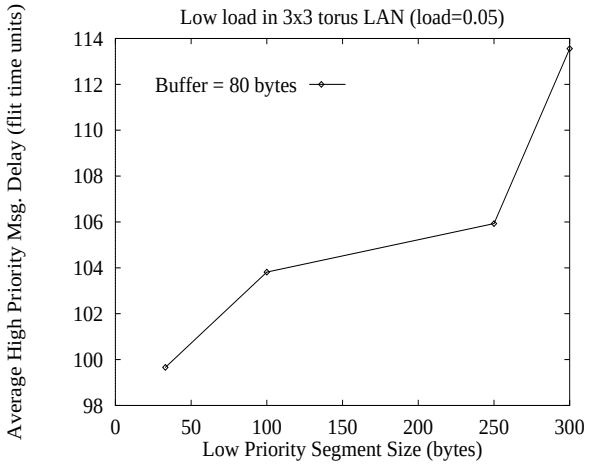

Figure 7: Average Delay for high priority messages in a lightly loaded 3x3 torus LAN.

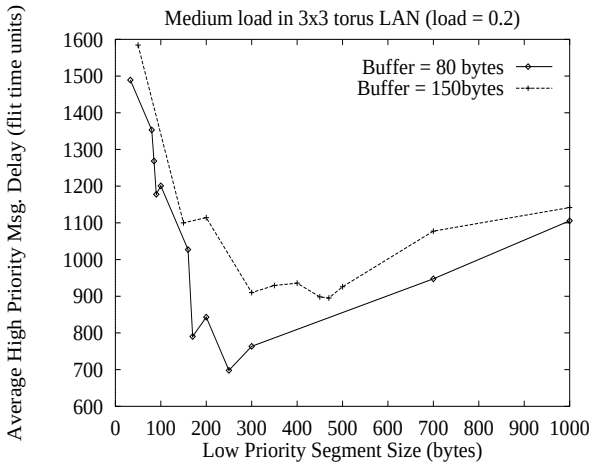

Figure 8: Average Delay for high priority messages in a medium loaded 3x3 torus LAN with load = 0.2. Optimal fixed segment size is 250 bytes when switch buffer is 80 bytes. Optimal fixed segment size is 450 when switch buffer is 150 bytes.

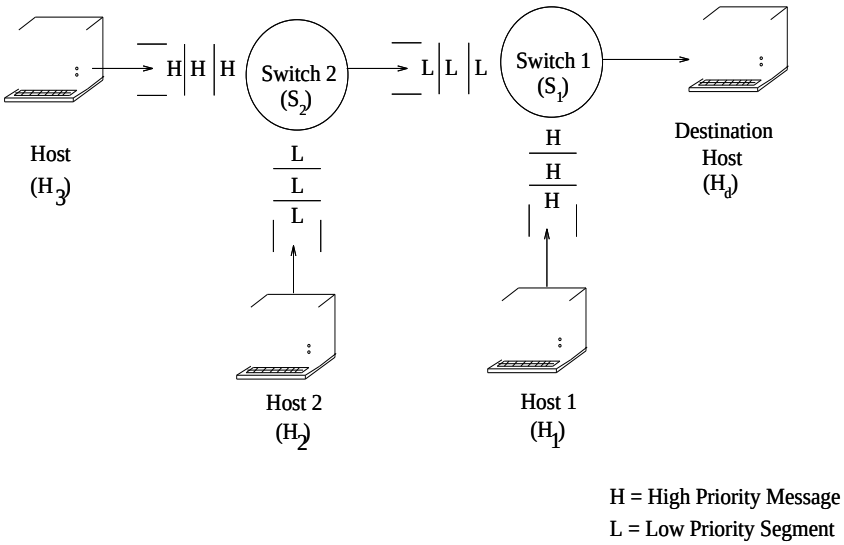

Figure 9: Higher order blocking develops under medium to heavy loads. H_1 is transmitting messages that block H_2 messages. H_2 messages in the switch buffer block H_3 from transmitting. Consequently, H_1 is indirectly blocking H_3. Optimal segment size is related to the distance the message must travel under medium to high traffic loading.

effect of low priority segment size on high priority message delay, as observed in Figure 8. The minimum delay is observed for a low priority fixed segment length of 250 bytes for the case where the switch slack buffer is 80 bytes at each input port.

When congestion in the network increases, more blocking occurs, in the sense that worms that are blocked themselves may end up blocking other worms. We call this *higher order blocking*. In principle, the goal of segmenting low priority messages is to give better delay performance to high priority messages. However, when segments are too small, the *higher order blocking* overshadows the *1st order blocking* and increases the average delay for high priority traffic.

To understand this effect, let us consider the following example shown in Figure 9, where 3 hosts, H_1, H_2, and H_3, transmit messages to a destination host, H_d. H_1 and H_3 have high priority messages to send while H_2 has low priority messages to transmit. In this scenario, H_2 has transmitted several small segments that fill the input slack buffer of S_1. Because switches employ round robin scheduling to resolve output port contention, H_3 must wait at least 3 cycles of transmissions of H_1 and H_2 before it is able to transmit. Specifically, H_3 is delayed by 3 high priority worm transmissions as well as 3 low priority transmissions before being able to transmit. In view of this, one can observe that in order to reduce the effect of *higher order blocking*, the low priority segment size should be on the order of the total amount of buffering available across the distance the packet must travel. In this case, the low priority segment size should be set to the size of the switch buffer at S_1. Messages from H_3 would then only be delayed for a single cycle of transmissions, one high priority message from H_1 and the single low priority segment filling the input buffer between S_1

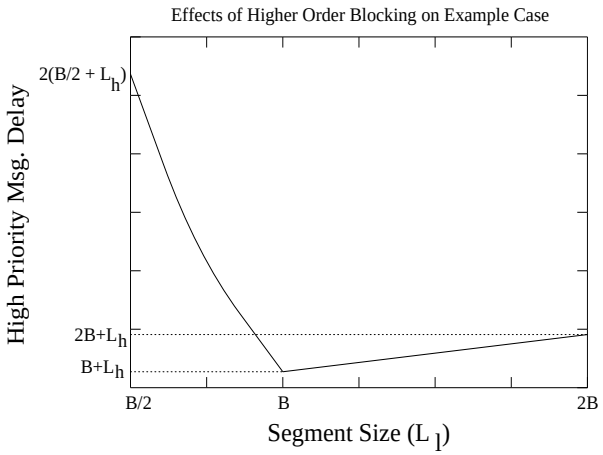

Figure 10: Delay for D_{H_3}. Optimal segment size is B.

and S_2. More generally, the delay of a message from H_3 (D_{H_3}) can be described as follows:

$$D_{H_3} = \begin{cases} \frac{B}{L_l}(L_l + L_h) & : \quad L_l \leq B \\ (L_l + L_h) & : \quad L_l > B \end{cases}$$

where switch buffers are of length B, high priority worms have an average length of L_h and low priority worms have a segment length of L_l. In this particular case, the low priority segment length (L_l) should be set to B in order to minimize D_{H_3} (See Figure 10).

This explains why the larger fixed segment size (250 bytes for case where switch buffer is 80 bytes) (see Figure 8) performs best in our experiment. In the 3x3 torus network, the maximum number of hops through the network is 3. When the switch buffer is 80 bytes, the segment size should be on the order of 240 bytes. However, when the low priority segment size increases beyond this point, high priority message delay also suffers. Low priority segments only need to be long enough to prevent added cycles of delay. If the packet size increases beyond this point, the low priority worms hamper the high priority delay performance directly via *1st order blocking*. Thus, Figure 8 illustrates the importance of optimal segment size for low priority messages. When the size of low priority messages is larger (or smaller) than the optimal value, the average delay of high priority messages suffers. The other curve (where switch buffer is 150 bytes) depicted in Figure 8 confirms this result. In the case where buffer sizes are set to 150 bytes at each switch input port, the optimal low priority fixed segment size is 450 bytes.

The reason for local minima appearing around 80 bytes and 160 bytes (for the buffer=80 bytes case) and around 150 bytes and 300 bytes (for the buffer=150 bytes case) is due to the fact that the effects of the *higher orders of blocking* drop off in stages. Since some of the low priority traffic only travels 1 hop, this fraction of the traffic benefits high priority traffic when the segment size is set to 80 bytes (for the buffer=80 bytes case). This occurs because low priority segments only need to be long enough to prevent the added

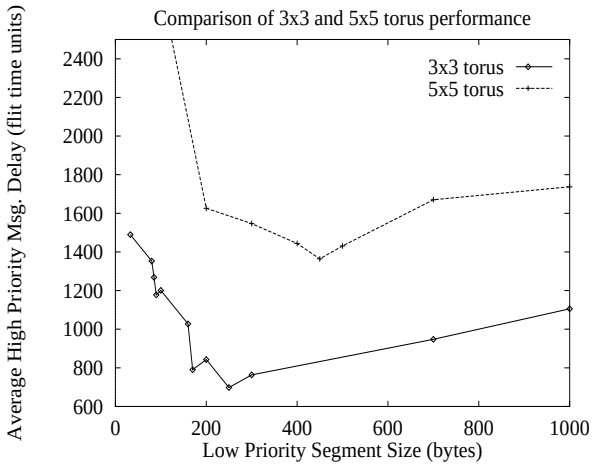

Figure 11: Comparison of average Delay for high priority messages in a 3x3 and a 5x5 torus LAN with medium traffic loading. Optimal fixed segment size is dependent upon the diameter of the network. For the 3x3 case, optimal fixed segment size is 250 bytes and for the 5x5 case, the optimal fixed segment size is 450 bytes.

cylces of delay. This is similarly true for the traffic travelling 2 hops. The overall minimum occurs when low priority segments are ensured to be long enough to avoid the *higher orders of blocking* for all traffic. This observation leads to the development of an adaptive scheme for setting the low priority segment size. From simulation experiments, this adaptive scheme provides a 10% decrease in delay for high priority traffic.

Adaptive Low Priority Segment Size Rule: Low priority segment size (L_l) should be set as follows:

$$L_l = B \cdot h$$

where B is the buffer size at each switch and h represents the number of hops the low priority message must travel.

Increasing the network diameter has the same effect as increasing the buffer size at each switch. Figure 11 shows the delay of high priority traffic with respect to low priority segment size for both a 3x3 torus and a 5x5 torus. In the 5x5 case, a certain fraction of the traffic travels 6 or 7 hops due to the up/down routing. This results in a higher than expected optimal segmentation value. Local minima do not appear in the 5x5 case because the measured simulation results do not provide adequate resolution.

To determine the load at which the *higher order blocking* effect begins to take place, a simulation experiment is run to find the crossover point. Figure 12 is a graph of offered load versus the average delay of high priority messages for four different specifications of segment size for low priority messages. The larger segment size cases (adaptive, 250 bytes and 300 bytes) perform worse than the smallest segment size case (33 bytes) when the load is under 0.15. However, as the load rises above 0.15, the 33 byte segment size case begins to perform much worse than all other segment size cases. This occurs due to the *higher orders of blocking*. The case where the adaptive rule is used performs the best among the larger segment size cases.

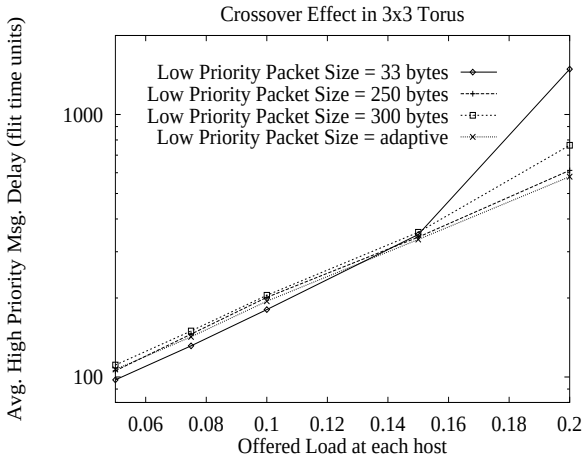

Figure 12: Offered load v.s. avg. delay for high priority messages in a 3x3 torus LAN. As load increases above 0.15, larger packet size for low priority stream is necessary to improve delay performance of high priority stream.

We conclude from this result that the low priority segment size should always be set to be on the order of the distance the message must travel (adaptive rule). Although performance of high priority messages suffer under very low loads, avoiding the severe degradation that occurs with small segment sizes under medium and high traffic loads make this design choice worthwhile.

3.3 Impact on Low Priority Traffic

The dual priority segmentation scheme provides increased bandwidth and reduced delay for high priority traffic. This improvement comes with a cost. Low priority traffic delay suffers. According to our results, high priority traffic can experience up to a 40% decrease in average delay under medium loads in a 3x3 torus topology when low priority messages are segmented. Under the same conditions, low priority traffic average delay increases more than threefold. This increase occurs because low priority traffic is segmented and consequently suffers from multiplexing delay since messages are of little use until the final segment of the message arrives to the destination host. However, only datagram traffic is segmented, which is not delay sensitive. In addition, a wormhole routing network such as Myrinet runs at 80 MBytes/s and thus latencies will remain on the order of a few hundred microseconds even when segmentation is used.

4 Effects of Priority Support at Switching Nodes

To improve performance of our scheme, we implement nonpreemptive priority at the switch. Introducing priority at the switch ensures that high priority messages (including short high priority messages) always receive preferential treatment at the switches. As expected, delay performance under this scheme is better

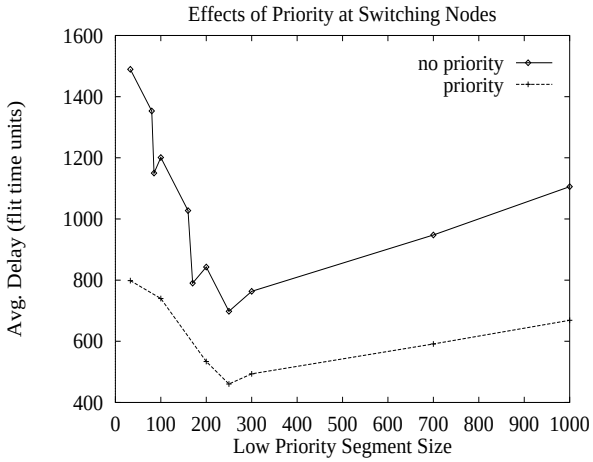

Figure 13: Low priority segment size v.s. avg. delay for high priority messages in a 3x3 torus LAN. Effects of priority support at switching nodes. Load = 0.2, ratio of high priority worms = 0.3. Priority improves performance but this graphs emphasizes the importance again of properly setting the low priority segment size.

(see Figure 13). However, Figure 13 continues to emphasize the need for properly setting the segment sizes of low priority messages.

A drawback of implementing priority support at a switching node is that switches become more complex. This degrades switching speeds and increases implementation costs.

5 Implementation Details

Implementation of the segmentation based bandwidth reservation scheme is simple. In the Myrinet architecture, the network interface unit (NIU) handles the mapping and routing. Consequently, it follows that the adaptive segmentation scheme should be implemented at the NIU level since the segment size is set according to the number of hops that a message must travel. The NIU dynamically sets the Maximum Transfer Unit (MTU) based on the priority level of the message and its route. Reassembly of the segmented messages takes place at a higher layer. The NIU processor would be overtaxed were it responsible for this functionality. The processor is already responsible for handling flow control, DMA and other functions.

The main concern is that a rogue traffic source may flood the network with long worms. This may be prevented by implementing a policing algorithm at each host that restricts the amount of high priority data that can be transmitted over some period of time. In addition, the policing algorithm must also handle imposing a limit on the high priority worm length so that a particular station does not secure an unfair amount of bandwidth as compared with other high priority traffic sources.

6 Conclusions

The dual priority segmentation scheme effectively reserves bandwidth for specified traffic streams. It is capable of providing improved delay and throughput performance to support delay sensitive traffic for applications such as distributed supercomputing and multimedia real-time services.

From our simulation experiments, a relationship between the optimal low priority segment size and the distance the message must travel can be observed. This relationship arises clearly when traffic loads increase and higher orders of blocking occur more frequently. To reduce the effects, the low priority segment size is set using the adaptive scheme described in section 3.2.

7 Future Work

We are currently investigating analytical methods of evaluating performance of the multipriority extension of this scheme. In addition, we are studying the performance of our segmentation scheme in a network that employs an optical backbone used to cover MAN distances.

To improve performance of our scheme, we are considering the use of virtual channels at the switch. The implementation of virtual channels would help alleviate the effects of *higher order blocking* experienced under medium to heavy traffic conditions. The problem with this improvements is that the switch complexity increases and switching speed suffers.

As mentioned earlier, the work in this paper addresses a best-effort scheme for providing improved bandwidth for high priority traffic. As a part of the SSN project, we are studying the support of stricter quality of service conditions (delay bounds, jitter control) in wormhole routing networks. In [GKK+96], a comparison of three different schemes using dedicated channels, virtual channels, and a synchronous token passing protocol called Hyper Token Ring [BN93, BN92, KBN97] are examined and evaluated.

For applications where the high priority traffic remains active for longer time scales (video streams), global information must be coordinated using a centralized quality of service manager entity. An effort to provide performance guarantees for these types of applications such as video and interactive visualization is now underway as a part of the ATOMIC-2 project at USC/ISI [TDX+95].

8 Acknowledgements

Special thanks to Prof. M. Gerla, Prof. E. Gafni, Prof. J. Cong and their students from UCLA, Dr. L. Bergman and S. Monacos from Jet Propulsion Laboratory, and Dr. J. Bannister from Aerospace Corp. for fruitful discussions concerning the SSN project that contributed to this paper.

References

[AS88] W.C. Athas and C.L. Seitz. Multicomputers: Message-passing concurrent computers. *IEEE Computer Magazing*, 21(8):9–25, August 1988.

[AV94] Vikram S. Adve and Mary K. Vernon. Performance analysis of mesh interconnection networks with deterministic routing. *IEEE Transactions on Parallel and Distributed Systems*, 5(3):225–246, March 1994.

[BaCG+96] Rajive Bagrodia, Yu an Chen, Mario Gerla, Bruce Kwan, Jay Martin Prasasth Palnati, and Simon Walton. Parallel simulation of a high-speed wormhole routing network. *PADS '96*, 1996.

[BCF+95] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, et al. Myrinet: A gigabit-per-second local area network. *IEEE Micro*, 15(1):29–36, February 1995.

[BN92] N. Bambos and A. Nguyen. Optimal message flow in ring netorks with spatial reuse. In *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 2360–2363, December 1992.

[BN93] N. Bambos and A. Nguyen. Queueing dynamics and throughput of ring networks with spatial reuse. In *Proceedings of the 31st Allerton Conference on Communications, Control and Computing*, pages 576–587, October 1993.

[Cla88] D. Clark. The design philosophy of the DARPA internet protocols. In *Proceedings ACM SIGCOMM '88*, 1988.

[DG92] Jeffrey T. Draper and Joydeep Ghosh. Multipath e-cube algorithms (meca) for adaptive wormhole routing and broadcasting in k-ary and n-cubes. In *The 6th International Parallel Processing Symposium*, pages 407–410, March 1992.

[DKS89] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. In *Proceedings of the ACM SIGCOMM*, pages 1–12, 1989.

[DS86] William J. Dally and Charles L. Seitz. The torus routing chip. *Journal of Distributed Computing*, 1(3):187–196, 1986.

[FRU92] Sergio Felperin, Prabhakar Raghavan, and Eli Upfal. A theory of wormhole routing in parallel computers. In *33rd Annual Symposium on Foundations of Computer Science*, pages 563–572, 1992.

[GKK+96] Mario Gerla, B. Kannan, Bruce Kwan, Emilio Leonardi, Fabio Neri, Prasasth Palnati, and Simon Walton. Quality of service support for high speed wormhole routing networks. *Submitted to International Conference on Network Protocols*, 1996.

[Gol90] S.J. Golestani. A stop-and-go queueing framework for congestion management. In *Proceedings of the ACM SIGCOMM*, pages 8–18, 1990.

[HK95] Po-Chi Hu and Leonard Kleinrock. A queueing model for wormhole routing with timeout. In *Proceedings of the 4th International Conference on Computer Communications and Networks*, pages 584–593, Sept. 1995.

[KBN97] B. Kwan, N. Bambos, and A. Nguyen. Hyper token ring: A high speed ring network with spatial reuse. *To Be Submitted to INFOCOM*, 1997.

[KD91] J. Kim and C.R. Das. Modeling wormhole routing in a hypercube. In *11th International Conference on Distributed Computing Systems*, pages 386–393, May 1991.

[KGB+95] L. Kleinrock, Mario Gerla, Nicholas Bambos, Jason Cong, Eli Gafni Larry Berman, Joseph Bannister, and Steve Monacos. Optimic: A scalable distributed all-optical terabit network. *to appear in Journal of High Speed Networks Special Issue on Optical Networks*, 1995.

[KHB+96] Bruce Kwan, Po-Chi Hu, Nicholas Bambos, Leonard Kleinrock, Joe Touch, and Hong Xu. Segmentation-based bandwidth reservation in wormhole routing networks. *UCLA Technical Report*, 1996.

[KK79] P. Kermani and L. Kleinrock. Virtual cut-through: A new computer communication switching technique. *Computer Networks*, 3:267–289, 1979.

[Nag87] John B. Nagle. On packet switches with infinite storage. *IEEE Transactions on Communications*, 35(4):435–438, April 1987.

[NM93] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *IEEE Computer Magazine*, 26(2):62–76, February 1993.

[OK92] Susan S. Owicki and Anna R. Karlin. Factors in the performance of the an1 computer network. *Performance Evaluation Review*, 20(1), June 1992.

[Par94] Craig Partridge. *Gigabit Networking*. Addison-Wesley, first edition, 1994.

[PG93] A.K.J. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[RS91] Thomas L. Rodeheffer and Michael D. Schroeder. Automatic reconfiguration in autonet. In *Thirteenth ACM Symposium on Operating Systems Principles*, pages 183–197, 1991.

[SV95] M. Shreedhar and George Varghese. Efficient fair queueing using deficit round robin. In *Proceedings of the ACM SIGCOMM*, pages 231–242, 1995.

[TDX+95] J. Touch, A. Deschon, H. Xu, T. Faber, T. Fisher, and A. Sachdev. Atomic-2: Production use of a gigabit lan. In *Gigabit Networking Workshop '95 at INFOCOM'95*, April 1995.

[VZF91] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. In *Proceedings of Tricomm '91*, June 1991.