On The Effects of Message Scheduling for Packet Switching Interconnect Fabric

Ludmila Cherkasova, Vadim Kotov, Tomas Rokicki
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94303

Abstract.

This report explores the impact of different message scheduling strategies on the performance of a packet-switched network under bursty traffic conditions and different routing strategies.

Deterministic routing strategies are attractive because they are cheap and fast to implement. However, possible drawbacks include lower throughput, significantly increased message latency under heavy traffic, and high contention for resources. Adaptive routing strategies are more flexible but inherently more complex which may result in slower routing.

We investigate the trade-offs involved in using different routing strategies while intelligently scheduling the packets from various messages for injection into the interconnect. This paper presents the results of a simulation study designed to answer this question for realistic *bursty traffic* workloads. In particular we compare deterministic and two forms of adaptive strategies under three different message scheduling algorithms: FIFO, Round Robin and Alpha scheduling. Our results indicate that for some types of bursty traffic with high volume of short messages, adaptive routing does not improve the interconnect performance, either in latency or in throughput. These performance results can be achieved by using either Round Robin or Alpha scheduling, which both tend to smooth traffic burstiness.

Contents

1 Introduction

This paper covers an investigation of the effects of message scheduling on the performance of a packet switched interconnect network under different routing strategies. Message scheduling orders the injection of packets from different messages to improve the interconnect performance.

As a basis for our modelling, we use the *PO2* interconnect as outlined in [CDKR94, CDKRR94]. The *PO2* topology is the same as that of the Mayfly [Davis92]: the elements are combined in a wrapped hexagonal mesh topology to form a low latency, high capacity interconnection fabric for scalable parallel processing systems containing up to hundreds of processing elements (PEs).

PO2 supports the transfer of messages which may vary in length but are physically transferred as a series of fixed-length packets. The network interface assumes all responsibility for fragmentation and reassembly, and notifies the receiving PE only when the complete message has been placed in the receiving PE's memory.

We will consider how much the interconnect performance might be improved by using special message (packet) scheduling algorithms. As a related question we will address the selection of appropriate routing strategies. We will investigate how much adaptivity in routing is necessary and sufficient for the efficient transfer of different types of traffic.

We will investigate the following three routing strategies:

The *Deterministic* strategy uses only a unique minimal path for every source and destination pair.

The *Best Paths* strategy allows any of the minimal paths to be selected, and thus is a minimal adaptive routing strategy.

The *Derouting* strategy is a non-minimal routing strategy which allows a limited number of deroutes from the minimal path.

We compare the performance of these different strategies under different kinds of bursty traffic and different message scheduling algorithms.

Applications are primarily concerned with transmitting variable-length messages which are independent of the particular packet size chosen by a network implementation. The interface to the network is responsible for segmentation of the messages into packets and reassembly at the other end. This situation dramatically changes the traffic pattern because instead of uniformly distributed packets, there are bursts of packets going from some source node to some destination node.

Bursty traffic generates a different type of port contention. Instead of occasional packets competing briefly for the same port, two bursts compete for some port during a longer period of time. If this contention is not controlled, packets can build up in the preceding nodes, leading to more contention and eventually complete interconnect saturation [Jain92]. To prevent this situation from happening a flow control mechanism based on "backpressure" is used.

The message latency must include the time packets from the message spend waiting for insertion into the interconnect. Thus, we investigate *message scheduling*, which attempts to minimize this waiting time (especially for short messages), and we show how appropriate scheduling improves the overall interconnect performance. We show how the Alpha message scheduling algorithm proposed in [CR94] can improve the interconnect performance from two to three times over FIFO message scheduling. The Alpha algorithm is tunable in favor of minimizing the latency of short messages.

The performance results using Round Robin message schedule are quite competitive against the interconnect performance using Alpha message scheduling for a number of traffic patterns. However, Round Robin schedule implementation is more difficult, and has some drawbacks we discuss in this paper.

The remainder of the paper presents our results in more detail. Section 2 describes the structure and basic features of the fabric and introduces three possible routing strategies. Section 3 defines bursty workloads and compares interconnect performance under different routing strategies and FIFO message scheduling. Section 4 introduces the Alpha scheduling algorithm on the *PO2* interconnect and compares its performance under different routing strategies and for different types of bursty traffic. Section 5 investigates Round Robin message scheduling on the *PO2* interconnect performance.

2 The *PO2* Interconnect

The *PO2* interconnect topology is a continuous hexagonal mesh which permits each node in the fabric to communicate with its six immediate neighbors. Figure 1 illustrates a sample of interconnect fabric containing nineteen nodes (only one axis is wrapped for clarity). The seventh port (the PE port, also not shown) connects each node to its corresponding processor.

It is convenient to define the size of the interconnect by the number of nodes on each edge. For example, the interconnect shown in Figure 1 represents an $E3$ interconnect. The total number of nodes in an En interconnect is $3n(n-1)+1$. Thus an $E3$ interconnect has nineteen nodes, while an $E6$ consists of ninety-one nodes.

Messages traveling through the interconnect are split into fixed-length *packets*. The first few words of a packet comprise the packet *header* which contains the source and destination addresses of the packet as well as a unique message and packet identifier.

Each node of *PO2* includes a routing device with six half-duplex links providing connections to the adjacent nodes. There is a single port that communicates with the processing element (*PE*). Each interconnect node has a centralized buffer pool consisting of twenty FIFO packet-sized *buffers* to store incoming packets (both in-transit and originating from the PE). There are also two crossbar switches. One connects the internal ports to the buffer pool and the other connects the buffer pool to the external ports.

Each port uses a FIFO discipline within a node to transfer the packets waiting for that port. When packets are waiting on a pair of ports in both the nodes they connect, the ports alternate directions between the nodes.

A *PO2* node can reject a packet if no buffers are available. An extension of this mech-

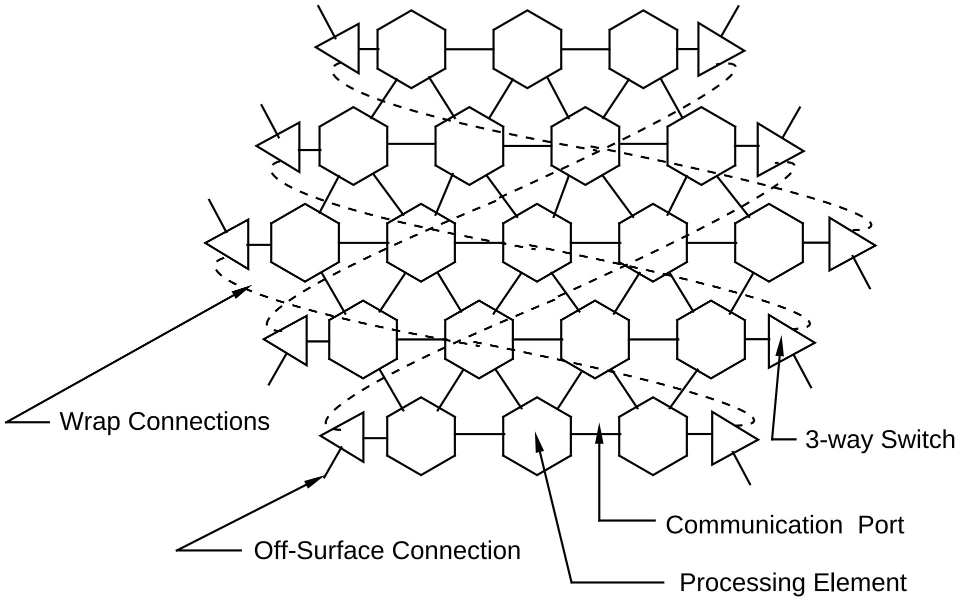

Figure 1: *PO2* Topology

anism is used to provide some measure of backpressure flow control on message bursts: a node rejects a packet if it already contains a waiting packet from the same message. Our results indicate that the backpressure provided by this mechanism justifies the costs of implementation.

The *PO2* design also supports *virtual cut-through* [Fujimoto83] in that as soon as the destination address has been received in a buffer, the next direction for that packet can be calculated. The decoupled input and output of the *PO2* FIFOs permit the head of the packet to be forwarded concurrently with reception of the tail of the packet. Once the packet leaves a buffer and correct receipt has been signalled by the receiving node, the buffer and port become free for subsequent transactions.

If a router is unable to immediately forward a packet, then the packet is stored in a buffer. Buffers are ordered by the packets arrival time in the node. Each buffered packet has a list of ports (in preferred order) that it is waiting on. Whenever some port becomes free, it checks stored packets waiting for it and routes the oldest waiting packet.

The main parameters of the *PO2* model are:

- We assume that each port permits a byte of information to be transmitted in 1 time unit. Each standard packet is 160 bytes long and hence takes 160 time units to transmit.
- The PE port has an additional overhead of 80 time units to establish a connection into the interconnect, and 20 time units to establish a connection out of the interconnect. These overheads on PE ports occur before any real packet data is transferred; the actual packet data transfer occurs at the rated bandwidth of the port, and the additional delay is not propagated (through virtual cut-through) to

the internal ports.

- To receive a packet header and to compute the next available direction takes 12 time units.
- There are 20 buffers in each node.

3 Bursty Traffic and FIFO Message Scheduling

While performance evaluation of packet-switched interconnects has focused on the latency of packets, applications are more concerned with the overall latency of variable-sized messages. Thus, in order to obtain meaningful performance results, we need to define *bursty traffic* workloads. These workloads are defined primarily by a message length distribution. Since we also consider priority traffic, we must also declare some percentage of the messages as priority messages.

Rather than considering many different message length distributions, we consider only bimodal distributions consisting of short messages and long messages. We define short messages to be from one to five packets in length, and we give each length equal probability. A five-packet message contains 640 bytes of payload. We choose a size of twenty-five packets, with 3,200 bytes of payload, for long messages; this is about the size of a disk or memory page.

We shall define our workloads by the percentage of long messages in the traffic; this was the primary variable defining the workloads. For instance, a workload with "10% long messages" contains 10% of long messages, and 90% of short messages. The average message length for this workload is 5.2 packets. Given a traffic density u between zero and one, we generate new messages using a negative exponential distribution with an average interarrival time of $5.2/u$.

A primary focus of the *PO2* design is to minimize the latency of short messages, possibly trading off long-message latency for short-message latency. Message latency is measured from the moment the message is sent by the application or operating system, as defined by the moment the message appears on the interconnect job list, to the moment all the packets of the message appear at the destination. Thus, this time includes both the queue wait time and the interconnect transit time. To compare different workload types and different message lengths, it is convenient to define a "normalized" average message latency which is not highly dependent on the message length. We define this normalized message latency as the total message latency divided by the message length.

We also measure and report the packet latency, as measured from the moment when PE port in the source node starts to inject the packet, until the moment when the packet is completely ejected from the interconnect by the PE at the destination node. Packet latency does not include any queue wait time.

Figure 2 and Figure 3 show the normalized average message latency and packet latency corresponding to a workload with 10% long messages using the three different routing strategies. The messages are injected into the interconnect in FIFO order; we shall present in a later section quite different results showing the effects of intelligent message scheduling.

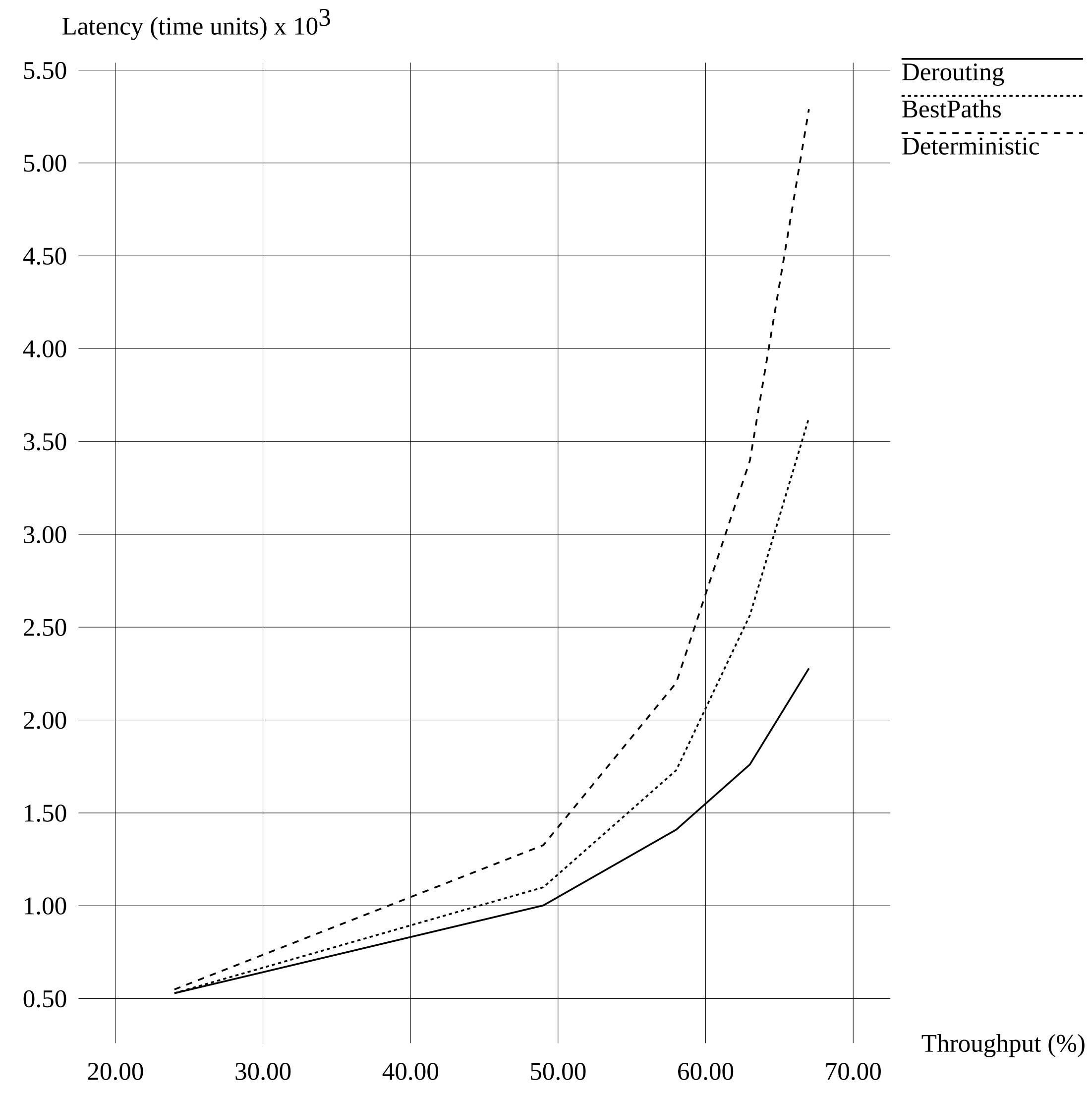

Figure 2: Normalized Average Message Latency for Different Routing Strategies and 10% Long Messages Workload

Figure 2 indicates that the Derouting strategy provides the best overall message latency, followed by the Best Paths and finally the Deterministic strategies. Interestingly, the packet latencies illustrated by figure 3 are in precisely the opposite order, with Deterministic providing the best overall packet latency.

This phenomena is partly explained by examining the port utilization under the different strategies, as shown in Figure 4. The solid black line represents ideal PE port utilization. The percentage of PE port utilization deviation from that line shows the amount of “busy waiting” on the PE port due to the backpressure flow control mechanism. Such busy waiting occurs when the PE port is available but the node rejects the packet because the node already has a waiting packet from the same burst. For a 67% applied

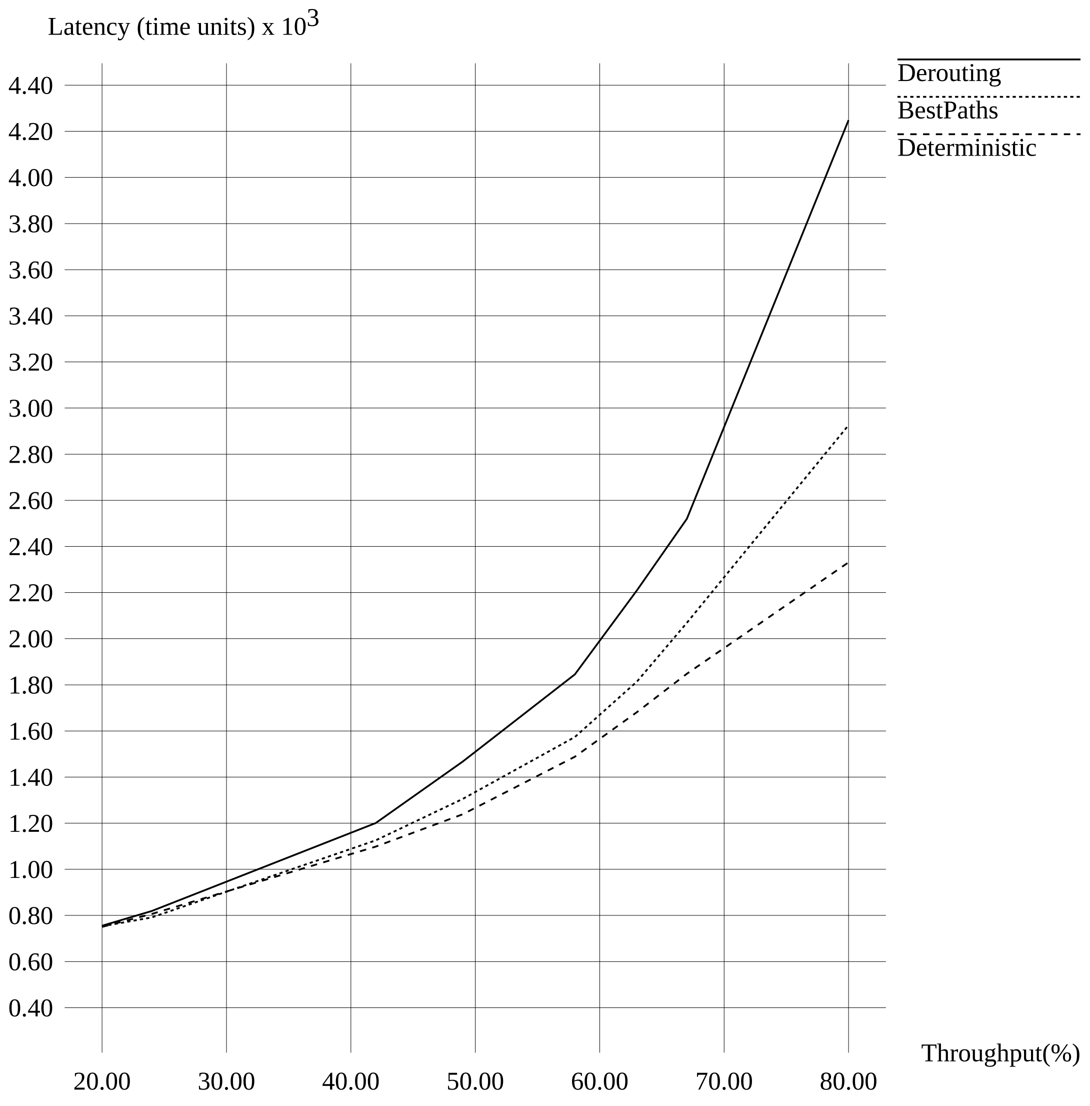

Figure 3: Packet Latency for Different Routing Strategies and 10% Long Messages Workload

load, the PE port utilization for the *Derouting* strategy is 69%, while for the *Best Paths* strategy it is 73% and for the *Deterministic* strategy it reaches 77%. This shows that with the less adaptive strategies packets spend their time waiting in the message queue. The less adaptive routing strategies have fewer available paths from source node to destination node. This strengthens the backpressure effect.

The more adaptive strategies maintain fewer packets in the queue, and therefore more packets inside the interconnect, for a given traffic load. With so many packets inside the interconnect, contention is higher, and the packets spend longer trying to reach the destination node and competing there for the destination PE port. Thus, the Derouting strategy leads to better overall utilization of interconnect fabric resources and provides better overall message latency. The backpressure mechanism under the *Best Paths* and

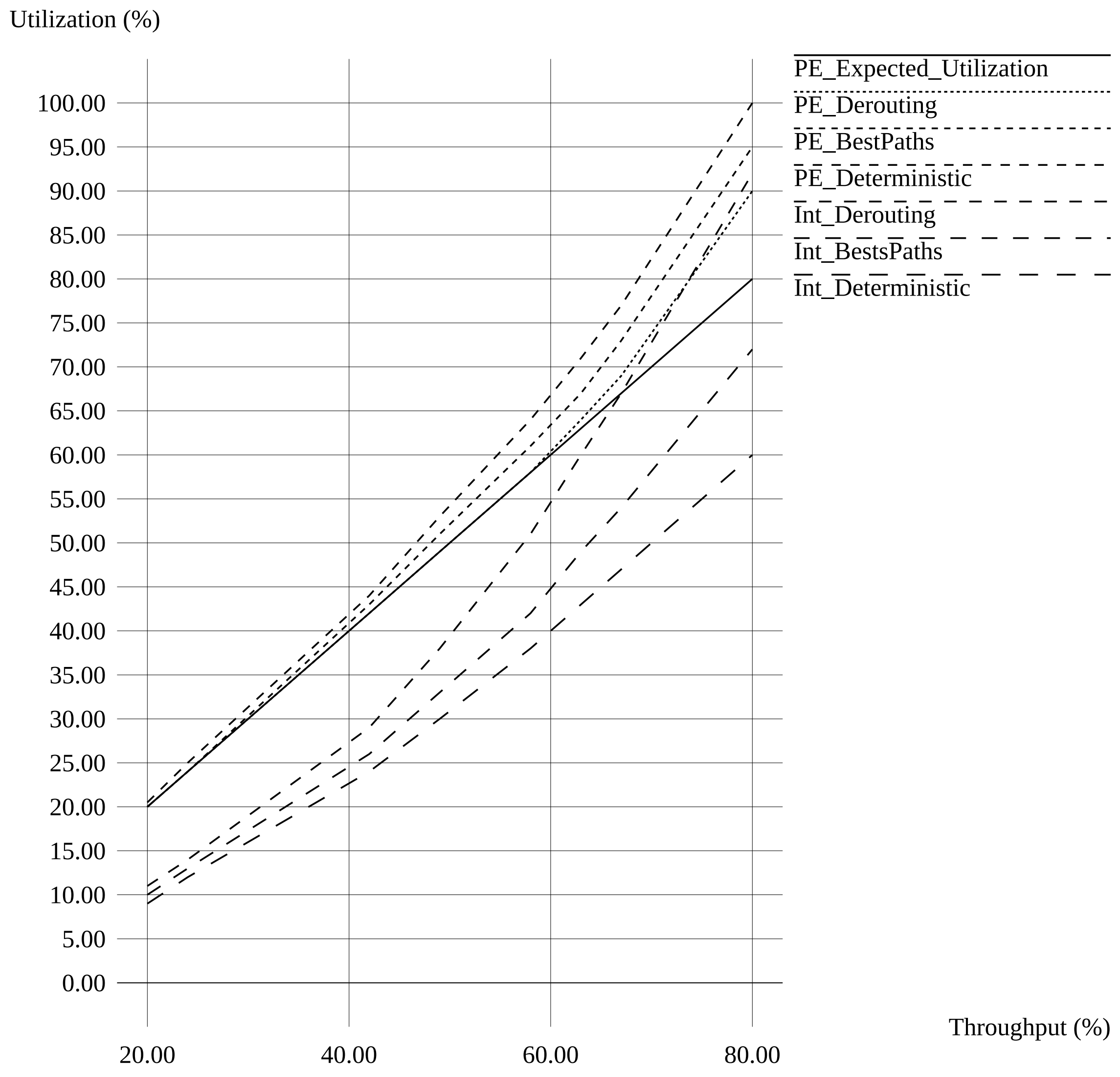

Figure 4: Port Utilization for Different Routing Strategies and 10% Long Messages Workload

Deterministic strategies has a significant impact, especially under heavier traffic, forcing packets and messages to wait outside the interconnect.

However under heavier traffic additional adaptivity exacts a cost. For 67% traffic utilization, the internal port utilization for the *Derouting* strategy is 67%, while for the *Best Paths* strategy is 54% and for the *Deterministic* strategy it reaches 47%. This shows that the *Derouting* strategy under uses significantly more resources: packets are allowed to deviate from the minimal path, using additional ports and buffers. This moves the bottleneck in the interconnect from PE port to its internal ports.

This phenomena is even more pronounced with a higher percentage of long messages. Figure 5 and Figure 6 show the normalized average message latency and packet latency

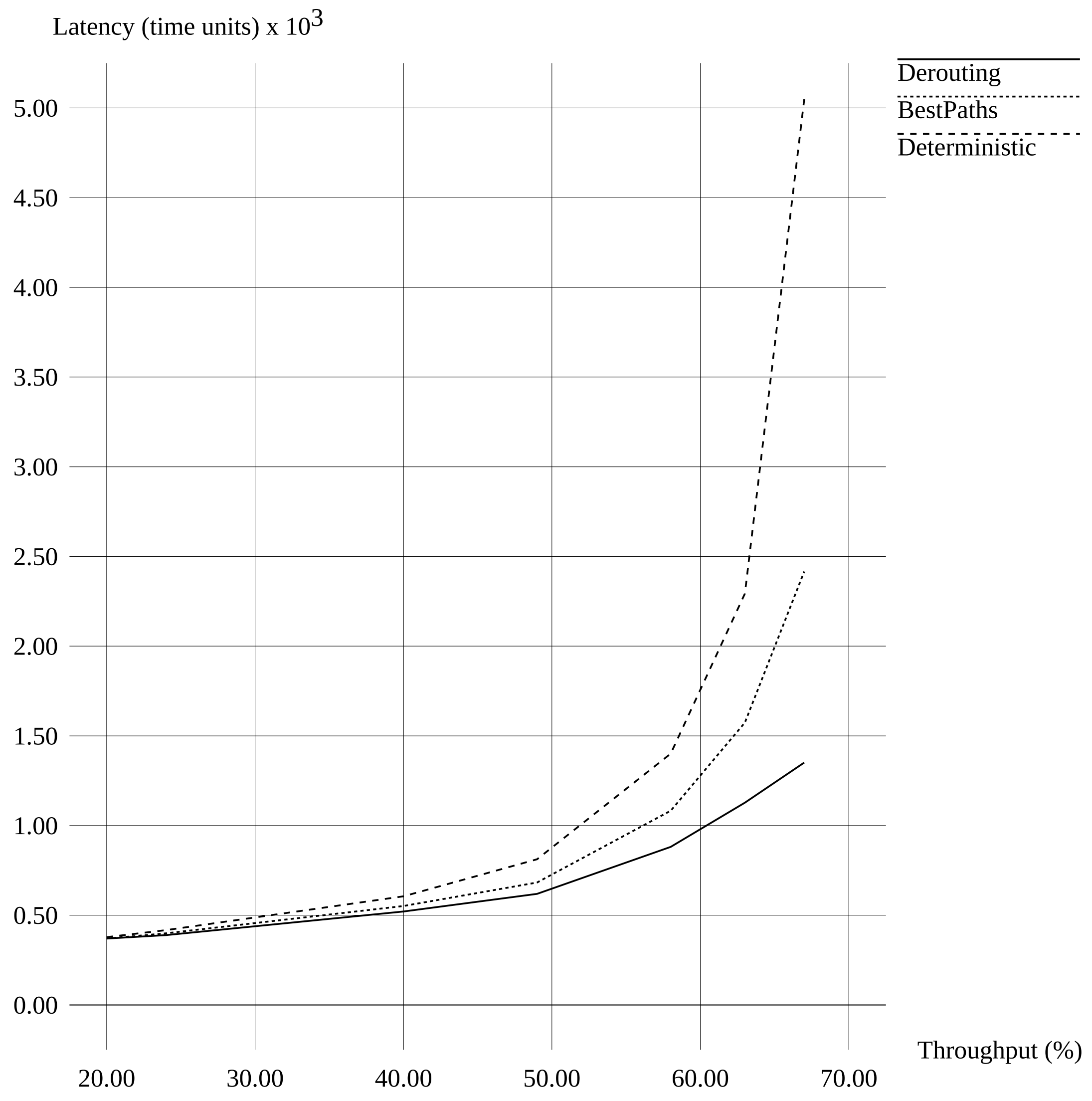

Figure 5: Normalized Average Message Latency for Different Routing Strategies and 80% Long Messages Workload

corresponding to a workload with 80% long messages. Figure 7 shows the PE and internal ports utilization generated by this workload. Again, the Derouting strategy provides the best overall message latency, followed by the Best Paths and finally the Deterministic strategies. However, the Derouting strategy reveals a few "dangerous" characteristics as well. First of all, at high traffic loads and on large interconnects, the internal port utilization rises higher than PE port utilization, at which point the internal ports limit the bandwidth of the interconnect. This happens much more quickly for the Derouting strategy, as illustrated by Figure 4 and Figure 7. In addition, the ability of a single message to distribute packets across a significant percentage of the interconnect fabric in the presence of contention reduces the efficiency of backpressure control flow mechanism. This raises the likelihood of interconnect saturation and deadlock.

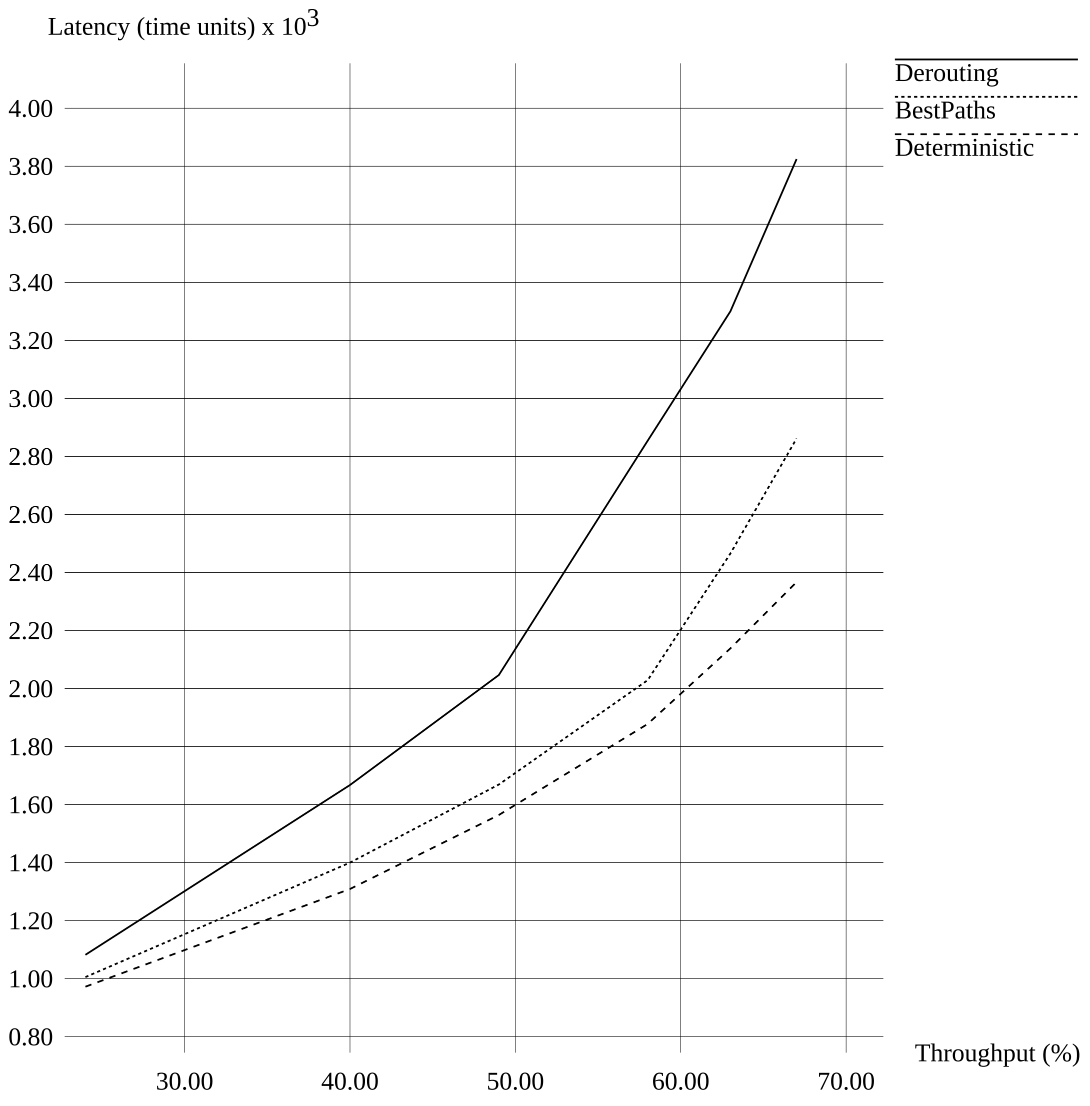

Figure 6: Packet Latency for Different Routing Strategies and 80% Long Messages Workload

4 Bursty Traffic and Alpha Message Scheduling

In the previous section, we described how the Derouting strategy is more effective than the other routing strategies with FIFO message insertion. In this section, we describe a more intelligent message insertion algorithm, called *alpha scheduling* [CR94] that largely eliminates the difference between the strategies while significantly decreasing the overall message latency. Indeed, [CR94] shows that suitable selection of a message insertion strategy can increase the effective performance of the interconnect by the factor of two or three over naive FIFO or round-robin strategies. In addition, alpha scheduling makes it easy to optimize the latency of short messages, while avoiding starvation for all message classes.

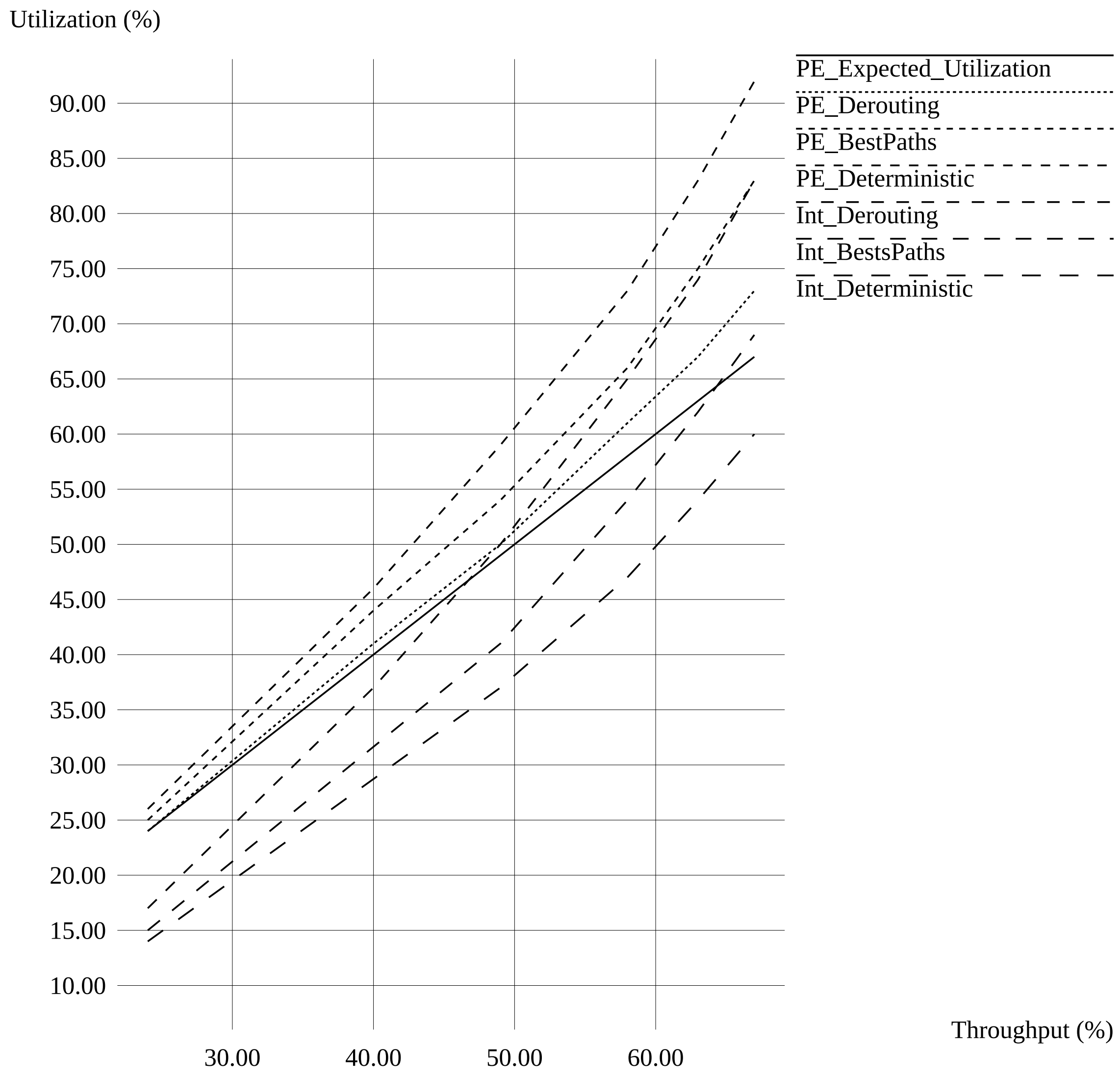

Figure 7: Port Utilization for Different Routing Strategies and 80% Long Messages Workload

With alpha scheduling, the messages waiting to be inserted are stored in a priority queue. Three parameters control the ordering of messages in the queue:

- The node parameter c is a "clock" that starts at zero and increments for each packet inserted into the interconnect through the current node. This value is easily kept bounded, either by resetting it to zero whenever the message queue empties, or by performing a scan through the priority queue on those rare occasions when the clock is about to exceed some maximum.
- The message parameter l is the number of packets in the message that have not yet been sent. Initially this is just the length of the message. As each packet is sent

out, the message priority is decremented by α to keep the head message priority up to date. Alternatively, the priority can be recalculated before considering the preemption of the head message.

- The tuning parameter α controls the balance between fairness and latency minimization; it can range from 0 to ∞.

Messages are inserted into the delivery queue with a priority of $c + \alpha l$. Messages with low priorities get delivered first. A new message inserted into the queue with a priority lower than that of the sending message preempts the sending message.

If $\alpha = 0$, then this strategy is simply FIFO.

If $\alpha = \infty$, then this strategy is simply shortest-packet first; this is optimal for average message latency.

If $\alpha = 1$ or some other finite positive value, then this strategy will not allow any application to be delayed indefinitely by the other applications, as long as there are a finite number of applications and each can insert a finite number of concurrent messages. Larger α yields better average latency; smaller α yields less potential delay of long messages.

The first results we present are for a workload of 10% long messages. Figure 8 shows the effect of alpha message scheduling on the latency of short messages using the Deterministic strategy. Different lines on this graph illustrate the effect of scheduling with different values of α ranging from 0 to 4.

Figure 9 shows the effect of alpha message scheduling on the latency of long messages using the Deterministic strategy.

When $\alpha = 0$, that is, when the strategy is simply FIFO, the latency of short messages is poor. Using $\alpha = 4$ improves the latency of short messages by a factor of five, and also improves the overall message latency by a factor of three. This, therefore, has a much greater effect on the overall performance than the use of the Derouting strategy, without the increased resource demands.

In general, alpha scheduling allows short messages to interrupt longer messages in such a way that overall queue waiting time is decreased. Indeed, it is easy to prove that short messages have a latency close to optimal. The trade-off is that the latency of long messages increases slightly. In the presence of the backpressure control mechanism, the latency increase is very small. If a long message is being inserted through a congested region, backpressure will quickly stall the queue, rejecting further packets from that message, so the long message would be delayed anyway. Injecting a few short messages significantly decreases their waiting time, without significantly affecting the overall latency of the long message. Effectively, this automatic fragmentation of long messages by short ones acts to decrease traffic burstiness and randomize the traffic pattern by interleaving "chunks" of long message with short messages, decreasing the congestion caused by a few coincident long messages.

Figure 10 and Figure 11 shows the effect of alpha message scheduling on the latency of short messages using the Best Paths strategy and Derouting strategy. Different lines on

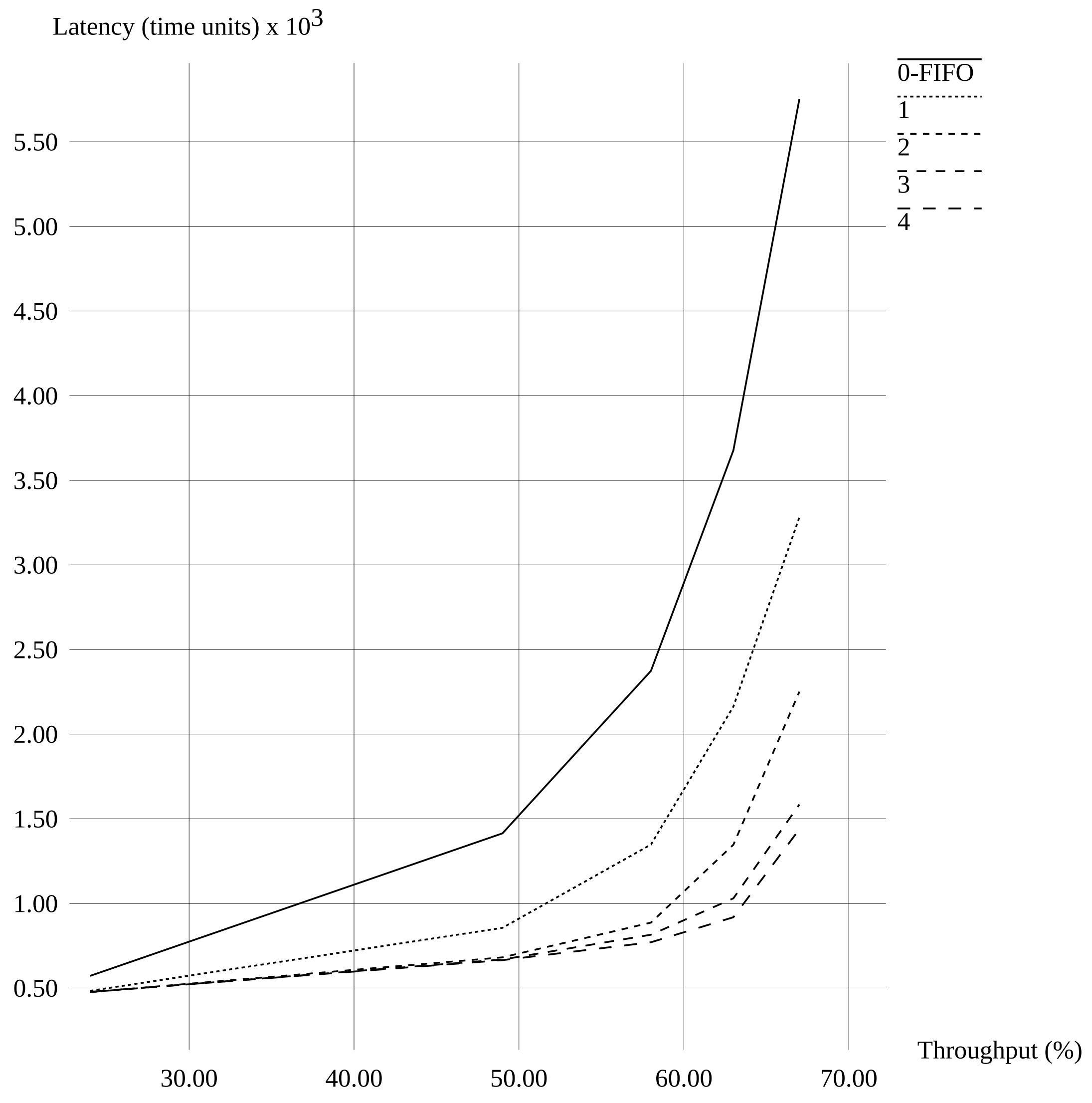

Figure 8: The Effect of Alpha Message Scheduling on Latency of Short Messages Using the Deterministic Strategy (Workload of 10% Long Messages)

this graph illustrate the effect of scheduling with different values of α ranging from 0 to 4. The latency of short messages decreases significantly for all the routing strategies.

Figure 12 shows the effect of alpha scheduling on message latency for the different strategies as the α parameter varies. All the points correspond to a traffic density of 67% and a long message percentage of 10%. Figure 12 shows that alpha scheduling with $\alpha = 8$ not only significantly improves the short message latency and overall interconnect performance, but it also makes the different routing strategies essentially indistinguishable in terms of latency.

Figure 13 shows the effect of alpha scheduling on the latency of one-packet messages

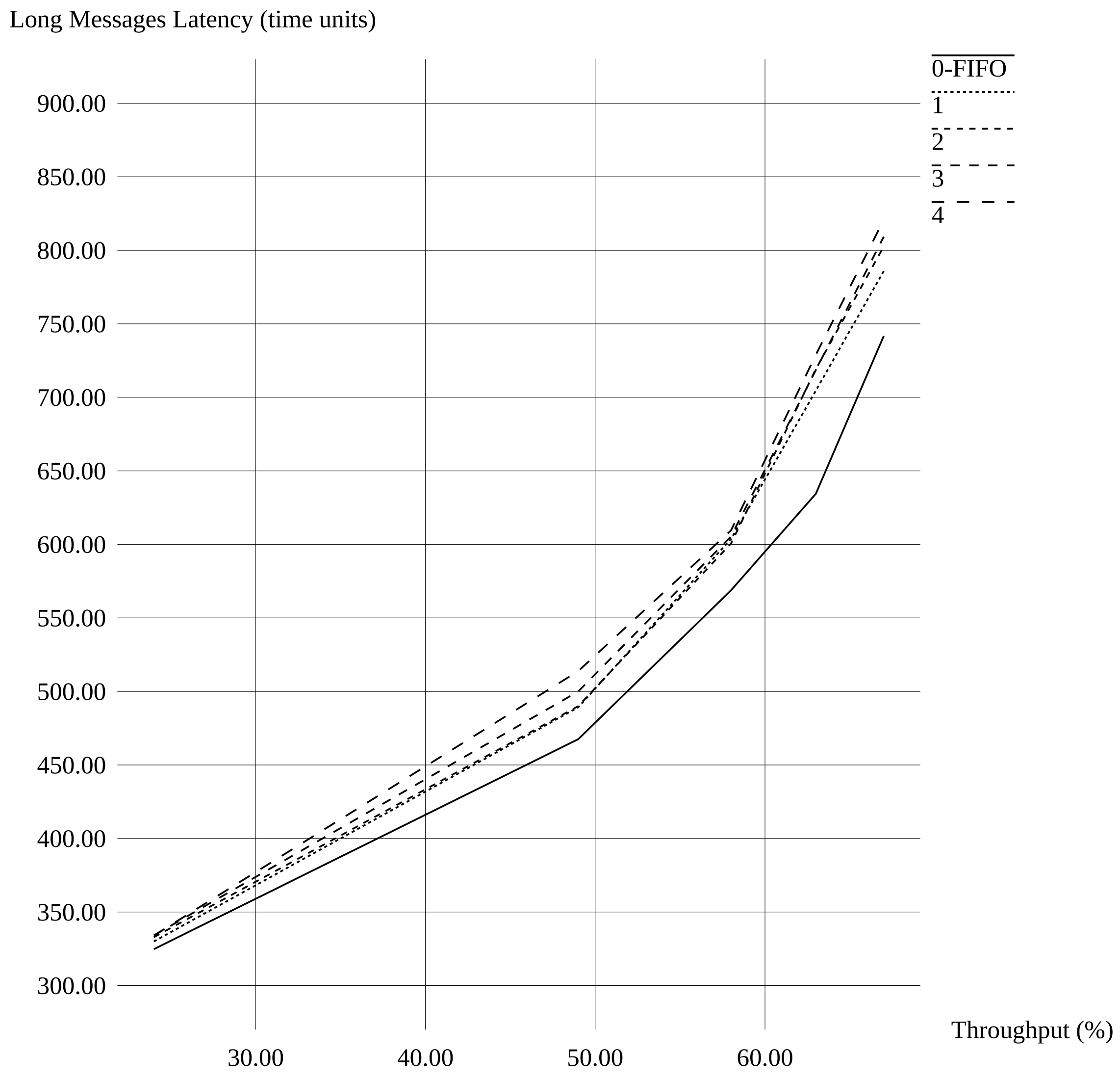

Figure 9: The Effect of Alpha Message Scheduling on Latency of Long Messages using the Deterministic Strategy (Workload of 10% Long Messages)

for the different routing strategies, as the α parameter varies. All the points correspond to a traffic density of 67% and a traffic pattern with 10% of long messages. Figure 13 more explicitly shows where the main contribution of alpha scheduling lies. In Figure 13, there are two lines for each strategy: packet latency and message latency. The difference between them is a waiting time for the one-packet size message to be inserted into the interconnect. Figure 13 shows that the packet latency through the interconnect is independent of alpha message scheduling and the value of alpha. Message latency decreases primarily through a decreased waiting time for the message to be inserted into the interconnect.

Alpha scheduling does not affect the interconnect performance as dramatically for all

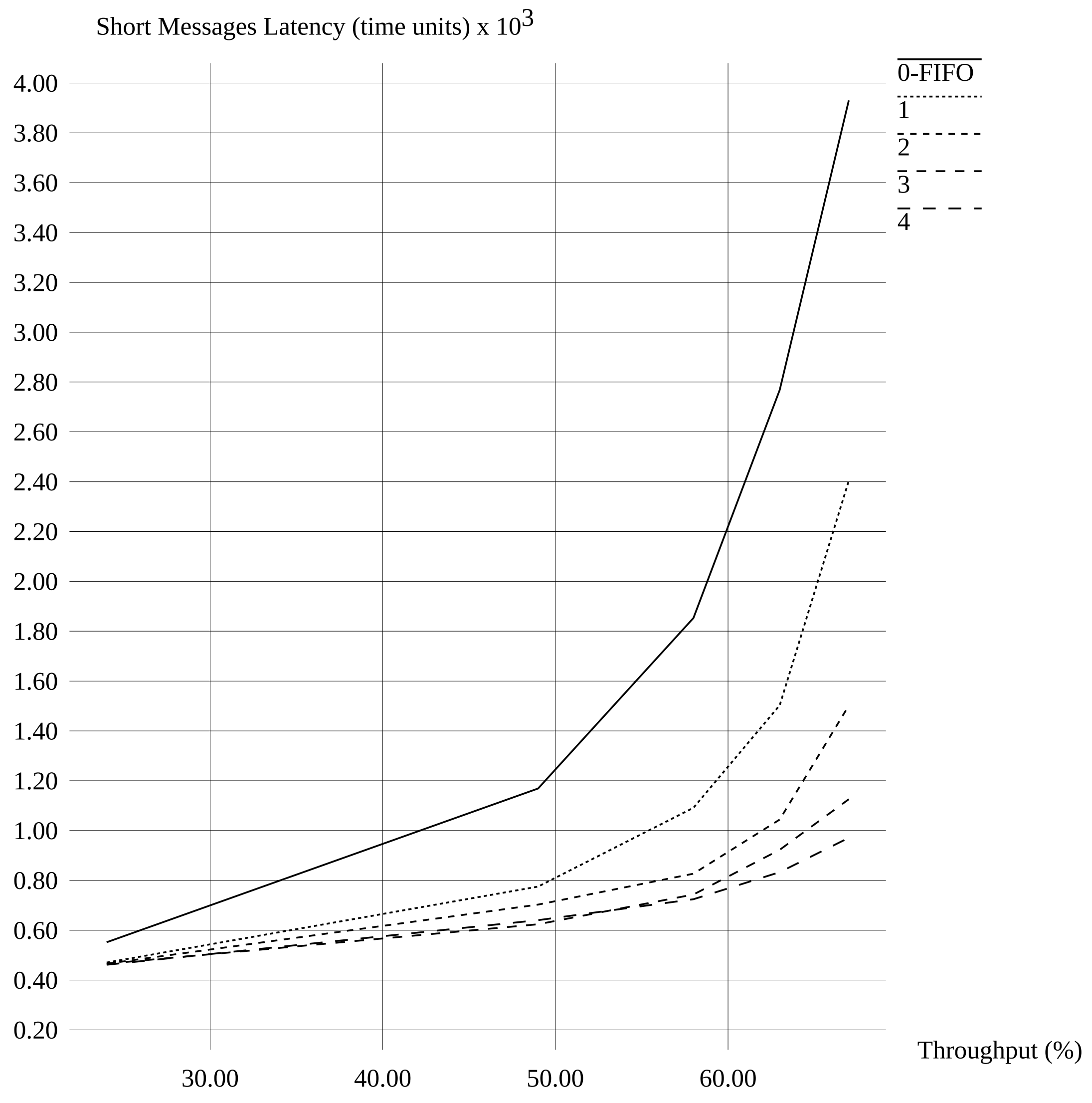

Figure 10: The Effect of Alpha Message Scheduling on Latency of Short Messages Using the Best Paths Strategy (Workload of 10% Long Messages)

workloads. For example, with 80% of long messages, alpha scheduling has less of an impact on average message latency, as shown by Figure 14, where the Derouting strategy again has the best overall message latency. All the points of correspond to a traffic density of 67%. Figure 14 illustrating the effect of scheduling with different values of α ranging from 0 to 10.

We still can see significant improvement of the latency of short messages for each of the three routing strategies shown in Figure 15. Moreover, Derouting strategy is beat by the Best Paths routing strategy with an α of 8 or more.

The explanation of the canging tendency in closeness of the different routing strategies is

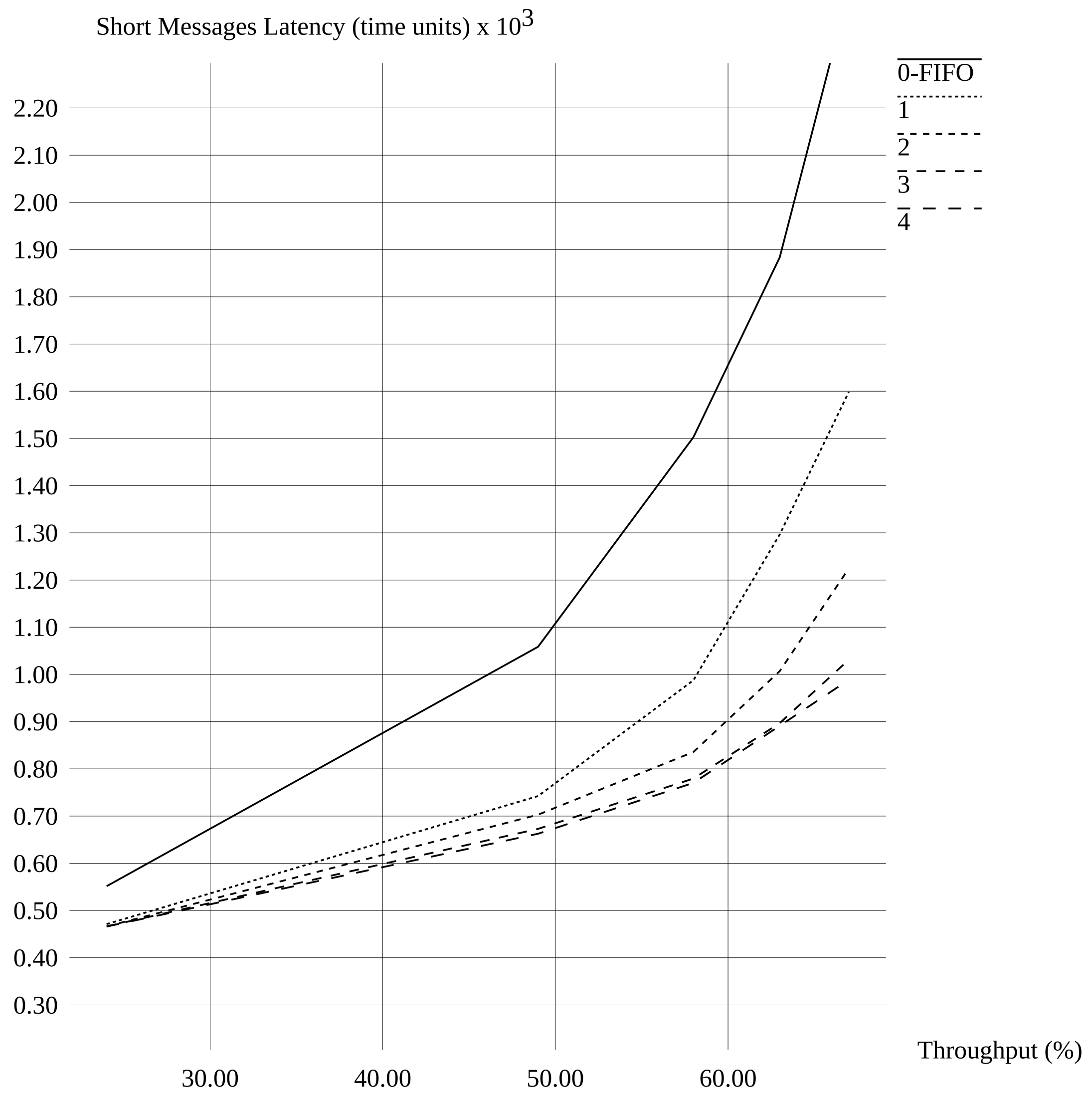

Figure 11: The Effect of Alpha Message Scheduling on Latency of Short Messages Using the Derouting Strategy (Workload of 10% Long Messages)

partly concludes in the changing ratio of packets belonging to short and long messages. The corresponding curves of the packets belonging to short and long messages are shown in Figure 16.

For workload with 80% of long messages, 97% of the packets belong to long messages; only 3% are from the short messages. Thus, there are not enough short messages to break up the long messages and to utilize the “busy waiting” time on PE port. Figure 16 illustrates the fraction of packets from long and short messages for the different workloads. As was noticed before, the backpressure mechanism is stronger for Best Paths and Deterministic strategies while transmitting long messages. Thus, there is a shortage of short messages used by scheduling to optimize short message latency and

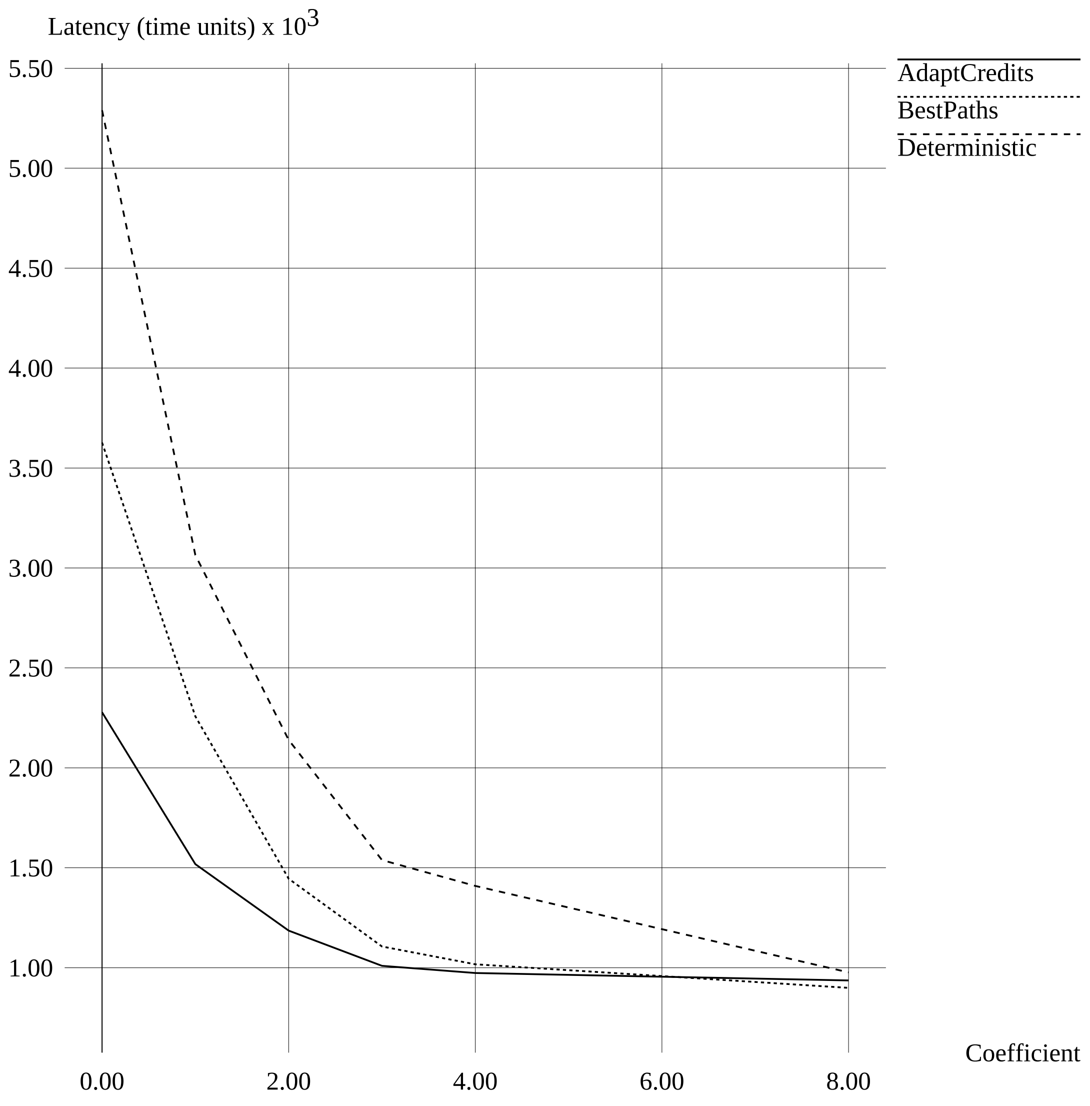

Figure 12: The Effect of Alpha Message Scheduling for Different Strategies (Workload of 10% Long Messages, Throughput 67%)

to smooth the effect of the backpressure mechanism.

Interestingly, it is precisely such a workload that demonstrates how important flow control is; with a lot of very long messages and a lot of adaptivity, the interconnect is easily saturated or even deadlocked.

With 10% long messages, the proportion of packets belonging to long and short messages is 48% and 52%, respectively; there are sufficient short messages to intersperse among the packets of the long messages to take best advantage of the interconnect resources.

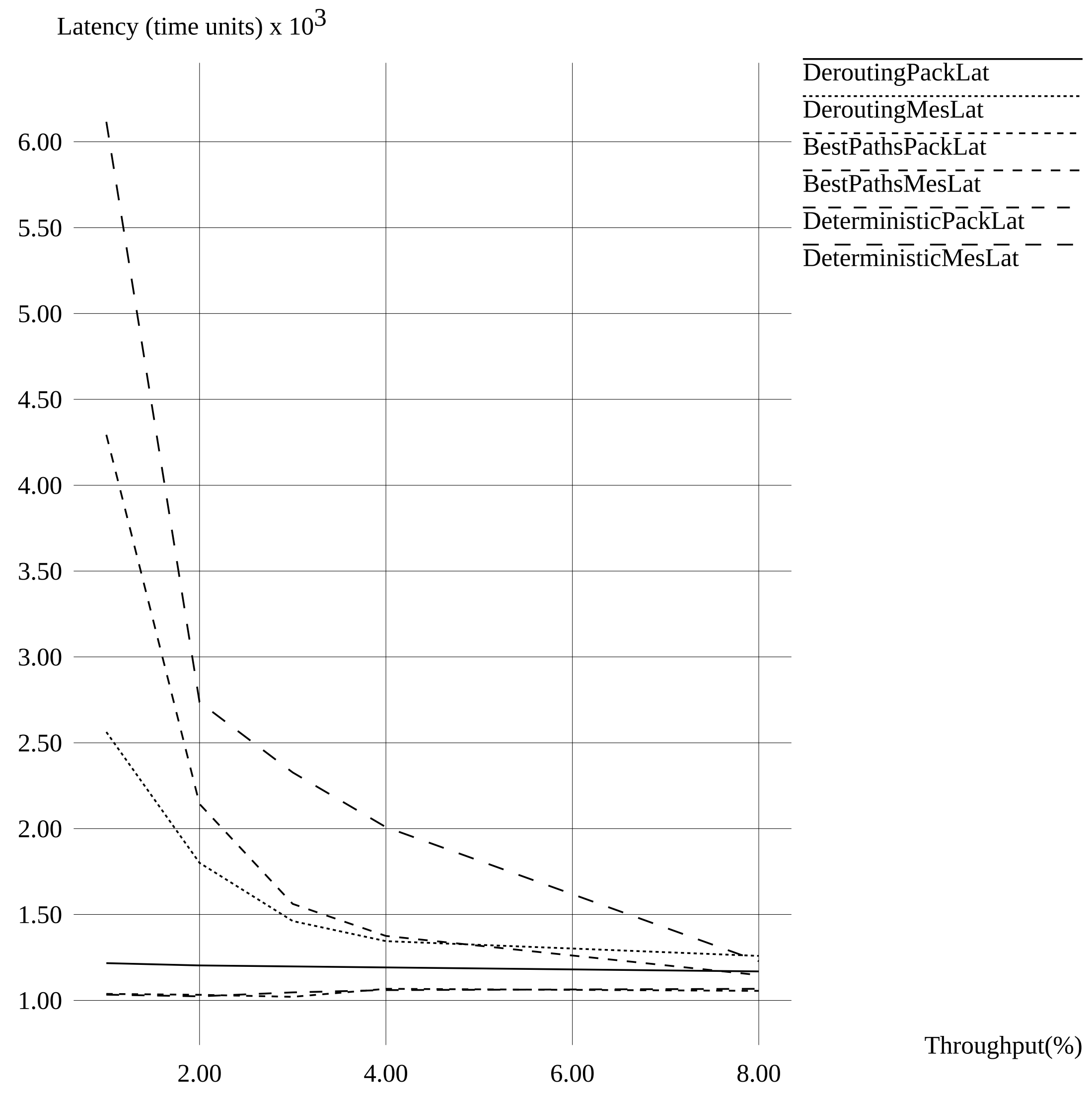

Figure 13: The Effect of Alpha Message Scheduling on Latency of One-Packet Messages for Different Strategies (Workload of 10% Long Messages, Throughput 67%)

5 Bursty Traffic and Round Robin Scheduling

Another scheduling algorithm, called round robin, iterates through the messages currently in the message queue, interleaving packets from outstanding messages. If we assume that new packets are inserted at the end of the message queue, then this algorithm is maximally fair; each application with an outstanding message will receive the same share of the bandwidth. It also guarantees delivery, since there are a finite number of applications.

In the paper [CR94], comparing different scheduling algorithms, we show that the av-

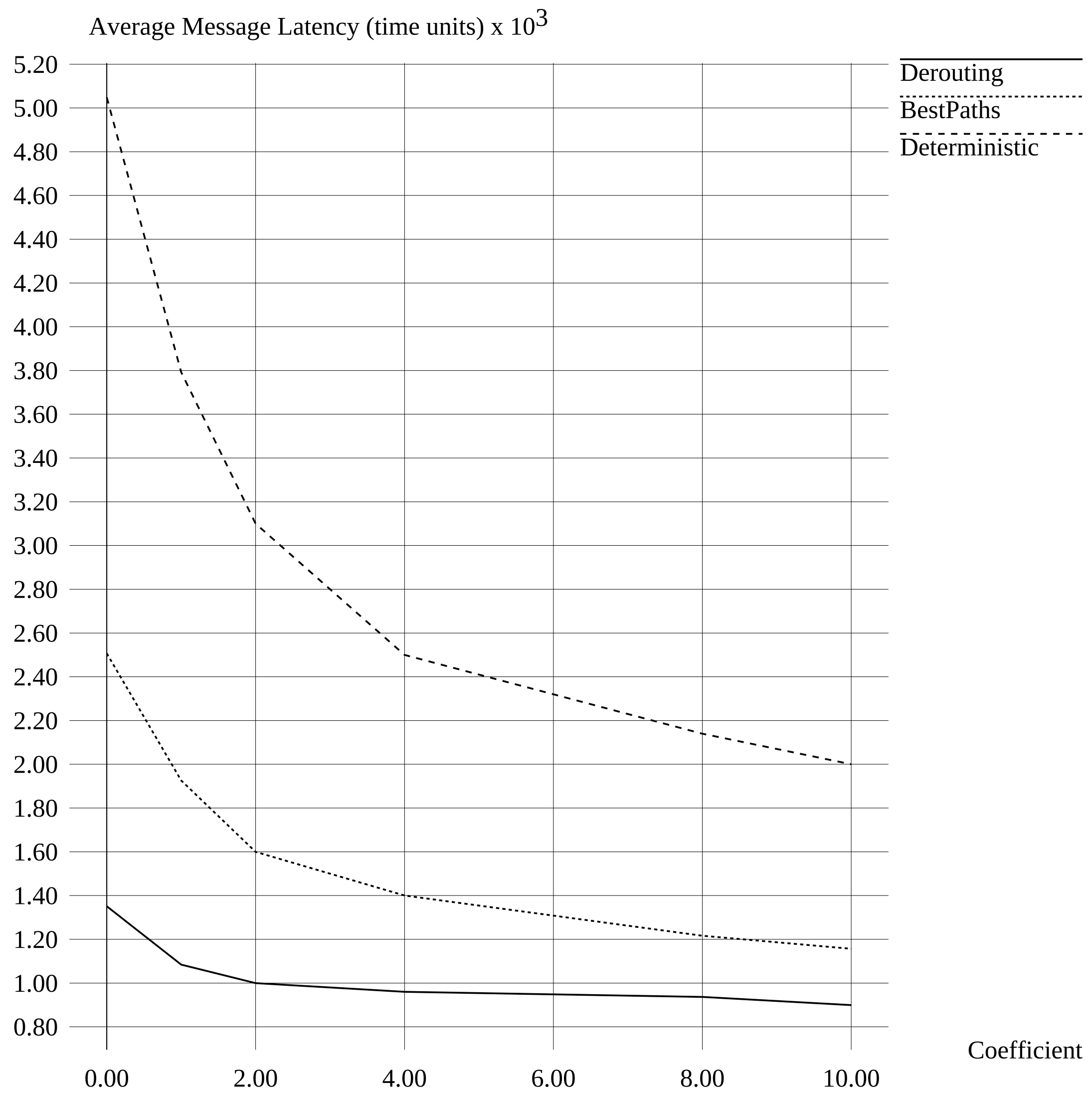

Figure 14: The Effect of Alpha Message Scheduling for Different Strategies (Workload of 80% Long Messages,Throughput 67%)

erage latency is not optimal; interleaving a short and a long message delays the short message by about a factor of two without changing the latency of the longer message. The worst-case average latency is when the final packets for all messages are sent at approximately the same time; this is possible with round robin scheduling. If all messages are about the same length, the average message latency is twice as bad as the optimal value. The increase in latency for a one-packet control message, on the other hand, is proportional to the number of applications; this is much better than with the FIFO scheduling algorithm.

However, in a presence of the backpressure control flow mechanism, the interconnect behaviour and the functioning of its PE ports are different. The longer the message, the

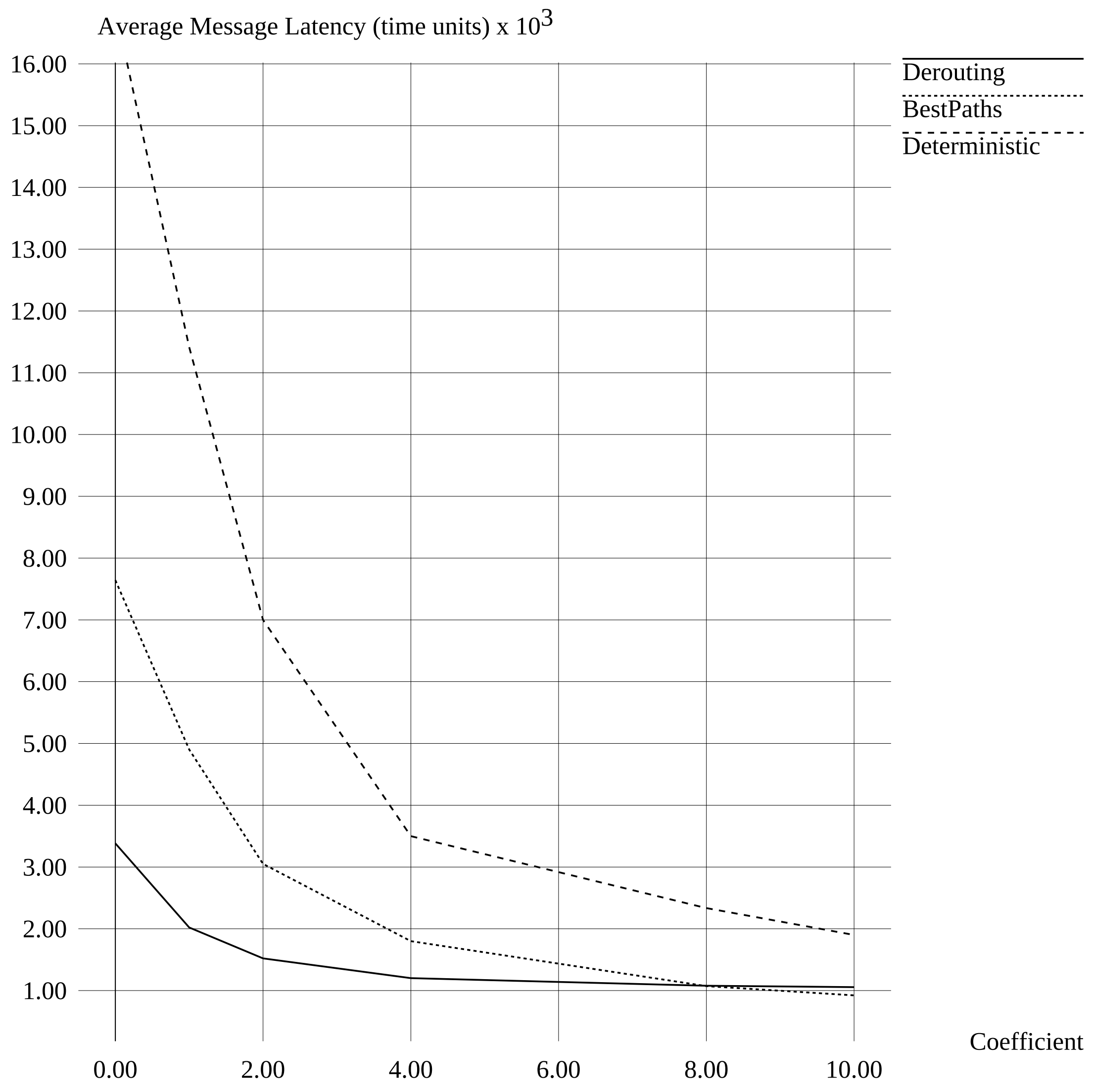

Figure 15: The Effect of Alpha Message Scheduling on Latency of Short Messages for Different Strategies (Workload of 80% Long Messages, Throughput 67%)

stronger is the effect of backpressure. Under these circumstances, interleaving packets from the different messages is a very good idea.

The first results we present are for a workload of 10% long messages. Figure 17 shows the normalized message latency for different routing strategies, comparing round robin scheduling algorithm against FIFO scheduling algorithm. Round robin always beats FIFO scheduling.

Round robin message scheduling has another distinct characterization: the difference in latency between the three routing strategies is insignificant. This result is consistent with the observation we made when comparing the three routing strategies under

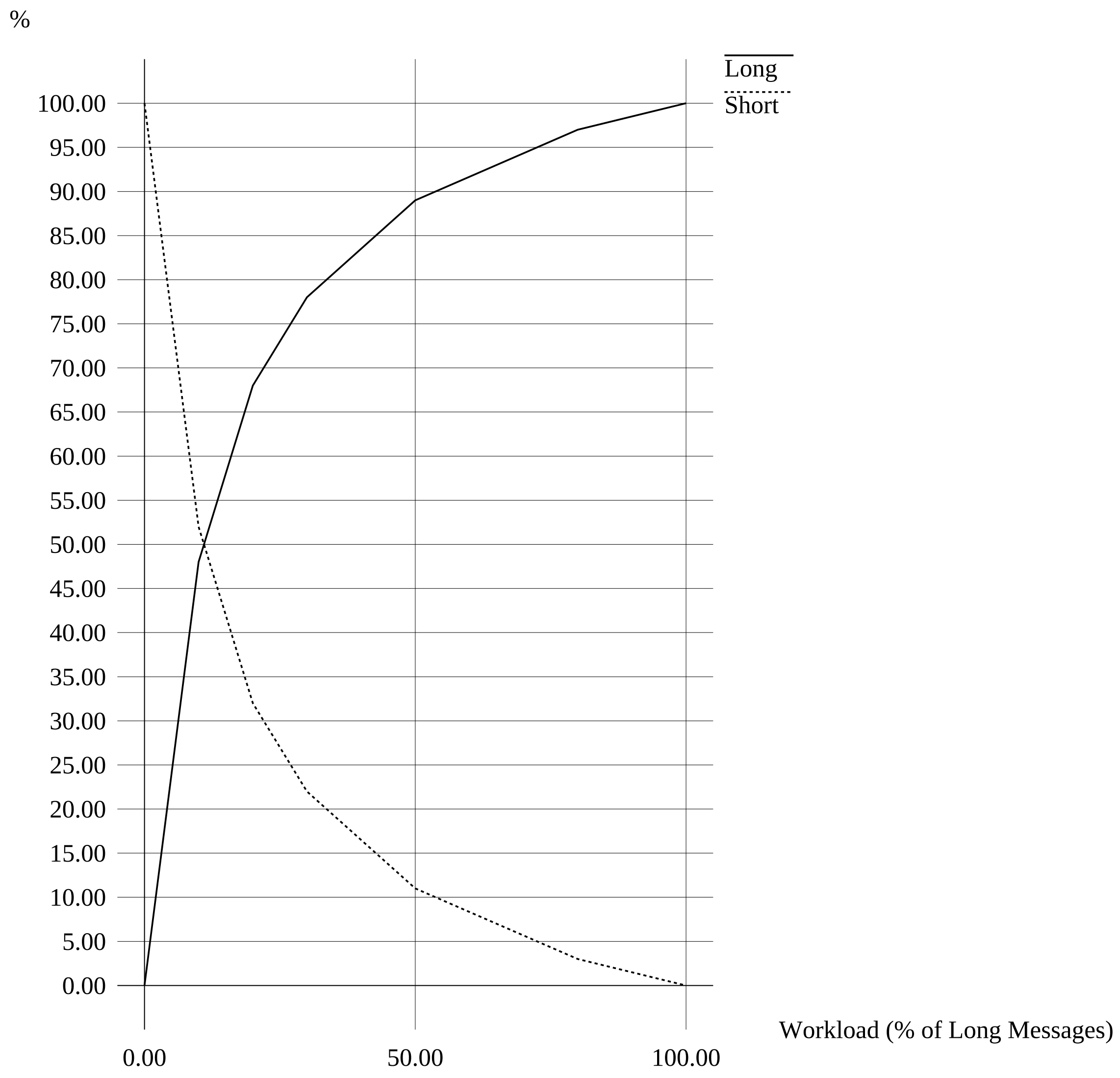

Figure 16: Proportion of Packets Belonging to Short and Long Messages for Different Workloads

uniform random traffic consisting of one-packet size messages. The latency results for the three strategies are practically indistiguishable. Round robin message scheduling interleaves the packets belonging to different messages, essentially decreasing the "burstiness" of traffic. This makes the traffic more like uniform random traffic, under which the difference between the three routing strategies also disappears.

Figure 18 compares the short message latency obtained by using the round robin scheduling algorithm against alpha scheduling (with parameter $\alpha = 8$) for a workload of 10% long messages.

Figure 19 shows the overall message latency for round robin scheduling against alpha

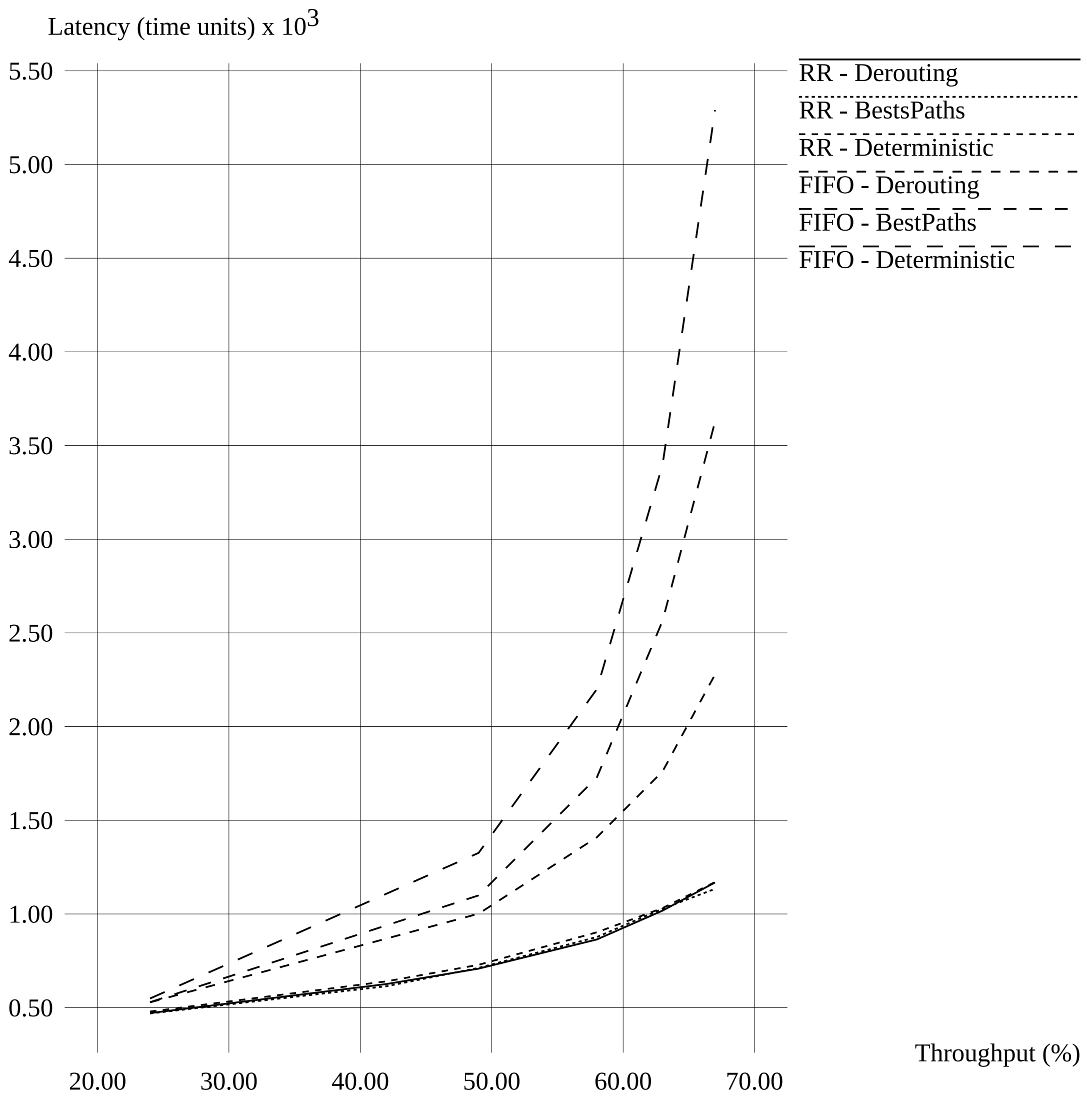

Figure 17: The Effect of Round Robin Scheduling on Normalized Average Message Latency against FIFO Algorithm Using Different Routing Strategies (Workload of 10% Long Messages)

scheduling (with parameter $\alpha = 8$) for the same workload of 10% long messages. It shows that alpha scheduling provides better performance results.

Figure 20 shows the overall message latency for round robin scheduling against alpha scheduling (with parameter $\alpha = 10$) for a workload with 80% long messages. It shows that Derouting strategy using alpha scheduling provides slightly better results than Derouting strategy using round robin scheduling. However Best Paths strategy using alpha scheduling provides slightly worse results than Best Paths strategy using round robin scheduling because of a higher probability of PE port busy waiting.

In general, alpha scheduling shows either better or very similar latency results when

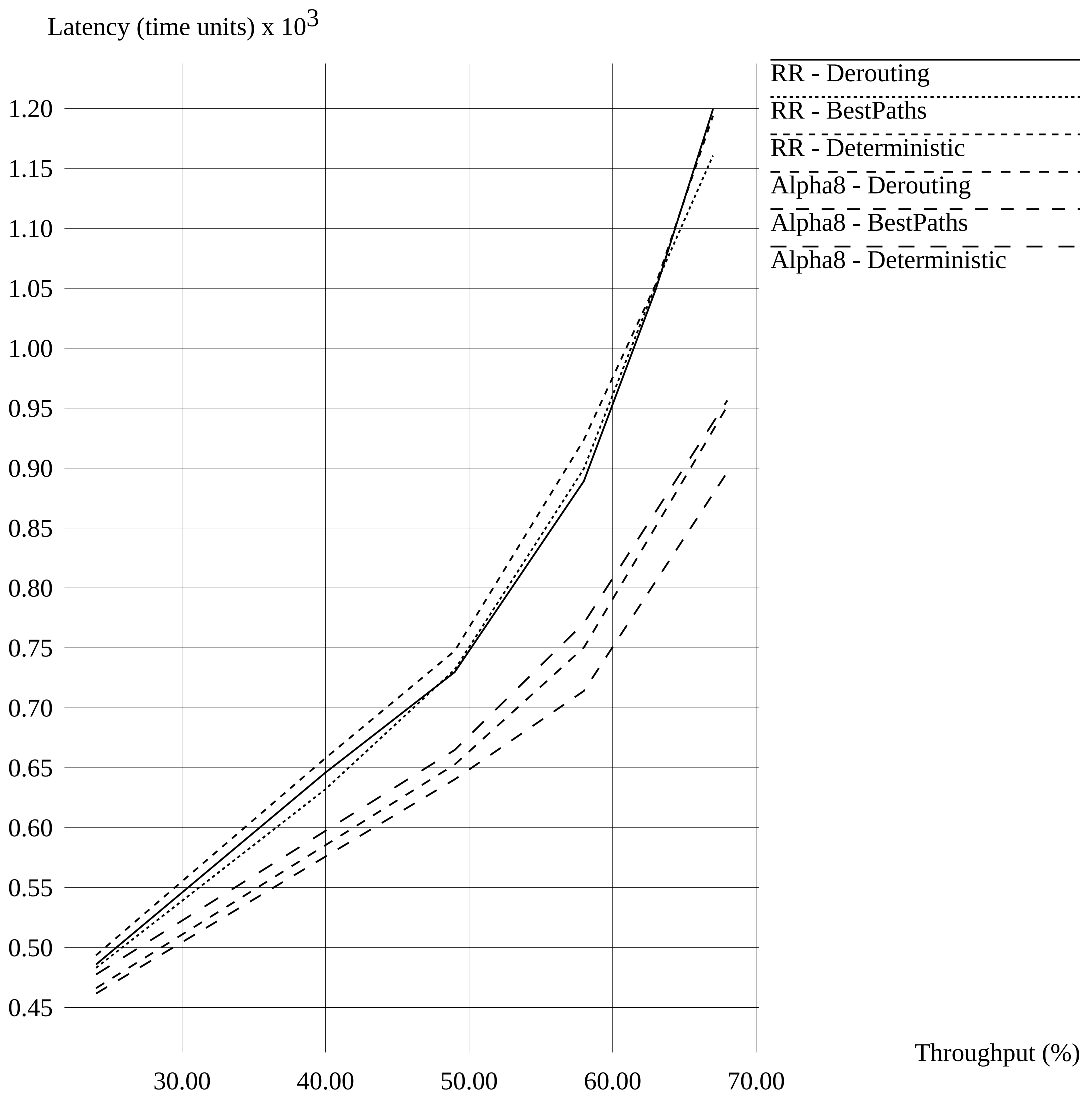

Figure 18: Round Robin Scheduling vs Alpha Scheduling ($\alpha = 8$) for Short Message Latency Using Different Routing Strategies (Workload of 10% Long Messages)

compared with round robin scheduling. The performance benefit of using alpha scheduling gets more pronounced as the message queue length increases, because in this case, round robin scheduling interleaves the packets from all the outstanding messages that increases the latency of each message proportionally to the message queue length. For even longer message queues, round robin strategy becomes even less efficient.

A possible problem with round robin scheduling is that the 'current message' changes with every packet. Depending on how access to the actual message body is done, this can have negative effects on cache hit rates. Round robin scheduling may strain a finite buffer pool used to store message data on an interface board. Finally, each message has a certain amount of state that will constantly need to be switched. If we are sending

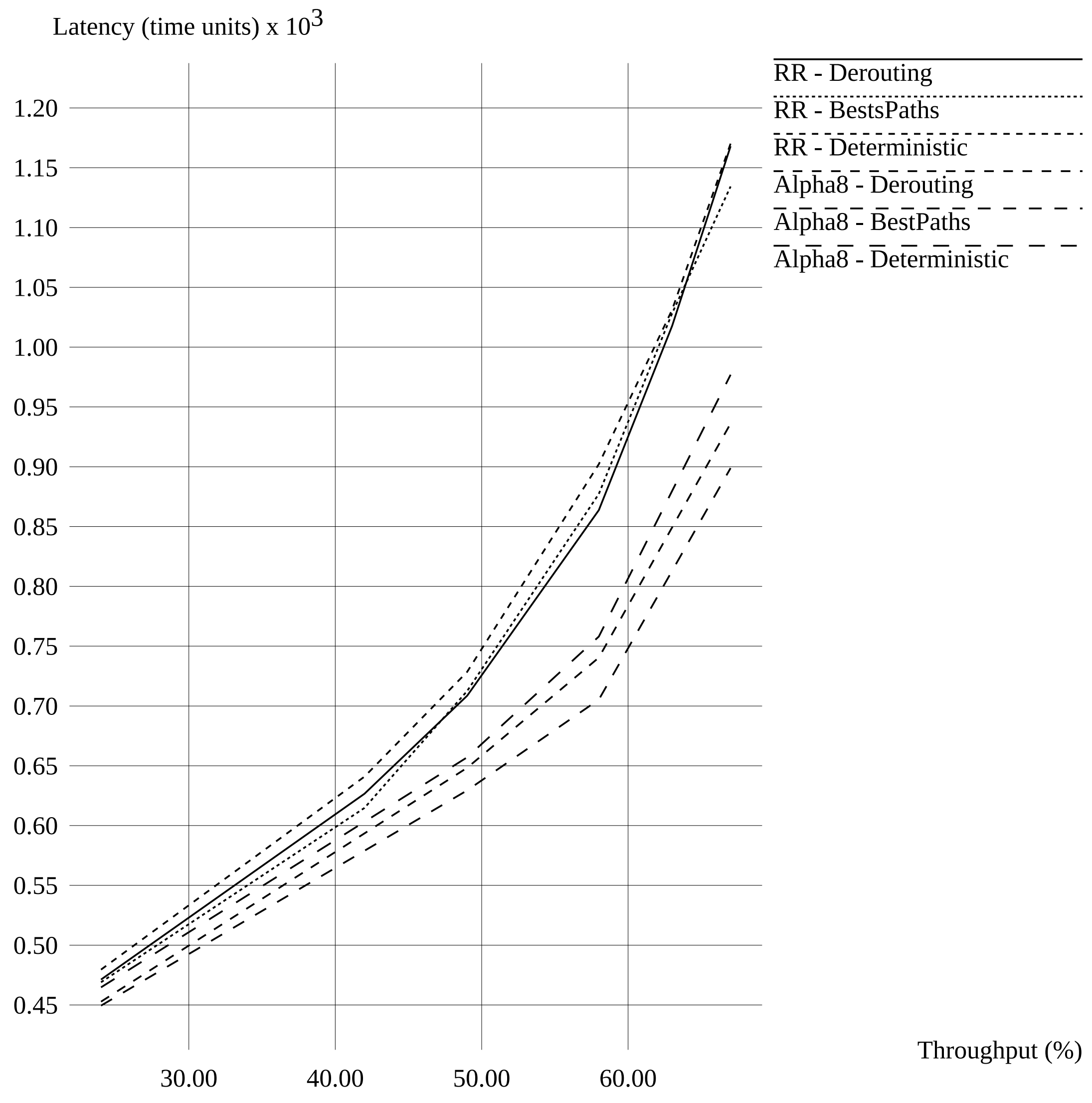

Figure 19: Round Robin Message Scheduling vs Alpha Scheduling ($\alpha = 8$) for Normalized Average Message Latency Using Different Routing Strategies (Workload of 10% Long Messages)

more than a million packets a second, these state switches might have a large negative impact.

A minor variant of the round robin strategy is to always insert new packets at the front of the message queue. In this case, messages of only one packet go out 'immediately'. Even in this variant, short messages of length two or more suffer in latency. In addition, always inserting the short packets in the front of the queue allows a few applications generating many short messages to indefinitely starve a long message.

However, in many cases, the performance results obtained by using alpha scheduling vs

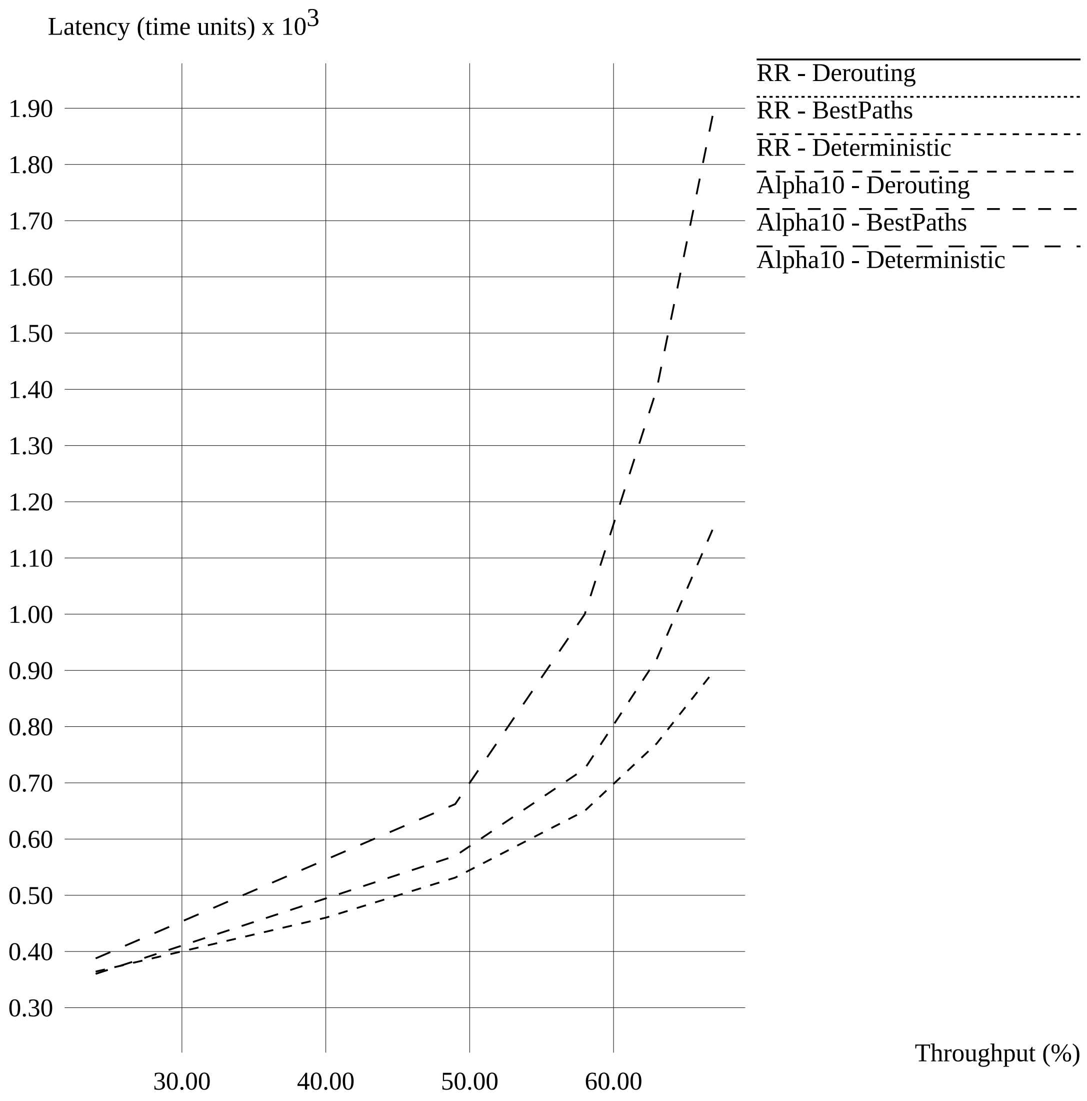

Figure 20: Round Robin Message Scheduling vs Alpha Scheduling (α = 10) for Normalized Average Message Latency Using Different Routing Strategies (Workload of 80% Long Messages)

round robin scheduling for a switch fabric with backpressure control flow mechanism are close enough in observed latency that we propose designers choose the one that is simpler to implement.

6 Conclusion

In this report, interconnect fabric performance under bursty traffic loads was studied. The main results compare the interconnect performance under three routing strategies using different message scheduling algorithms: FIFO, alpha scheduling, round robin algorithm.

With FIFO message scheduling the Derouting strategy provides the best performance results, followed by the Best Paths and finally the Deterministic strategies.

A new scheduling algorithm, alpha scheduling, improves the interconnect performance 2-3 times over the results provided by FIFO scheduling. The results also indicate that the performance advantage of the Derouting strategy disappears when the workload contains a sufficient mix of long and short messages. This is especially important when some degree of backpressure flow-control is desired. Adaptivity decreases the efficacy of backpressure because of the larger number of nodes that can be populated with packets from a particular message. In addition, the internal port utilization rises as derouting frequency increases, lowering the effective maximum throughput of the interconnect for large networks. This higher port utilization threatens network saturation and deadlock.

The use of round robin scheduling makes the performance of all three routing strategies essentially indistinguishable. Round robin scheduling shows slightly worse results than alpha scheduling.

The similarity in performance of round robin and alpha scheduling is mostly due to the presence of backpressure control flow mechanism. With backpressure, interleaving of packets belonging to different messages can be beneficial. In the absence of backpressure, the performance advantage of using alpha scheduling versus round robin scheduling might be up to two times [CR94]. The performance benefit of using alpha scheduling increases as the message queue lengthens.

We conclude that the strategies with less adaptivity, such as the Best Paths and Deterministic strategies, might perform as well as Derouting strategy when intelligent message scheduling is used.

7 Acknowledgements

We especially thank Al Davis, Ian Robinson and Robin Hodgson for useful discussions and remarks during our “scheduling” efforts based on their design of an interconnect fabric, and Chris Hsiung for constant managerial support and active participation during all the stages of the research.

8 References

[AC93] Aoyama, K. and A. A. Chien. The Cost of Adaptivity and Virtual Lanes. Preprint, July 1993.

[CDKR94] Cherkasova, L., A. Davis, V. Kotov, and T. Rokicki. Colored Petri Net Methods for Performance Analysis of Scalable High-Speed Interconnects. In Proceedings of International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'94).

[CDKRR94] Cherkasova, L., A. Davis, V. Kotov, I. Robinson, and T. Rokicki. How Much Adaptivity is Required for Bursty Traffic? To be published.

[CR94] Cherkasova, L. and T. Rokicki. Alpha Message Scheduling for Packet-Switched Interconnects. To be published.

[Chien93] Chien, Andrew A.: A Cost and Speed Model for k-ary n-cube Wormwhole Routers. In *Proceedings of Hot Interconnects '93, A Symposium on High Performance Interconnects*, August 1993.

[Dally89] Dally, W. J. et al. The J-Machine: A Fine-Grain Concurrent Computer. In *Proceedings of the IFIP Conference*, North-Holland, pp. 1147–1153, 1989.

[DS87] Dally, W. J. and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. J. *IEEE Transactions on Computers*, Vol.C–36, No. 5, 1987.

[Davis92] Davis A. Mayfly: A General-Purpose, Scalable, Parallel Processing Architecture. J. *LISP and Symbolic Computation*, vol.5, No.1/2, May 1992.

[Fujimoto83] Fujimoto R. M. VLSI Communication Components for Multicomputer Networks. Ph.D. Thesis, University of California at Berkeley, August 1983.

[Jain92] Jain, R.: Myths About Congestion Management in High-speed Networks. Internetworking: Research and Experience, Vol. 3, pp. 101–113, 1992.

[Seitz84] Seitz, C. L. "The Cosmic Cube". J.*Communications of the ACM*, Vol.28, No. 1, pp. 22-33, January 1984.

[Wille92] Wille, R. A High-Speed Channel Controller for the Chaos Router. Technical Report TR-91-12-03, University of Washington, 1992.