PAPER *Special Issue on Architecture, Algorithms and Networks for Massively Parallel Computing*

A Comparison of Blocking and Non-Blocking Packet Switching Techniques in Hierarchical Ring Networks

G. Ravindran† *and* **M. Stumm**†, *Nonmembers*

SUMMARY This paper presents the results of a simulation study of blocking and non-blocking switching for hierarchical ring networks. The switching techniques include wormhole, virtual cut-through, and slotted ring. We conclude that slotted ring network performs better than the more popular wormhole and virtual cut-through networks. We also show that the size of the node buffers is an important parameter and that choosing them too large can hurt performance in some cases. Slotted rings have the advantage that the choice of buffer size is easier in that larger than necessary buffers do not hurt performance and hence a single choice of buffer size performs well for all system configurations. In contrast, the optimal buffer size for virtual cut-through and wormhole switching nodes varies depending on the system configuration and the level in the hierarchy in which the switching node lies.

key words: Networks, Switching, Wormhole, Virtual Cut-through, Hierarchical Ring Networks, Slotted Rings

1. Introduction

Shared memory multiprocessors based on hierarchical unidirectional ring networks are interesting alternatives to popular direct networks such as 2D meshes or tori. Unidirectional rings allow for simple network node designs and simple node to ring interfaces, allowing them to be clocked at faster rates. They require fewer connections at the node to ring interface, allowing for wider data paths and thus shorter message sizes. Moreover, rings allow easy addition and removal of nodes at arbitrary locations. In many ways, unidirectional ring can be considered the simplest way to connect multiple processing modules using point-to-point interconnection. The topology of hierarchical ring systems allows exploitation of the spatial locality of memory accesses often exhibited in parallel programs, which is critical to size scalability, and it allows efficient implementation of broadcasts. A number of shared memory multiprocessor systems with hierarchical ring structure have been proposed and built, including KSR-1 [3], Hector [16] and NUMAchine [2].

In this paper, we analyse and compare the performance of blocking and non-blocking switching techniques in hierarchical ring networks, using a detailed flit-level simulator. The switching technique determines how packets are forwarded through the network and has a significant impact on performance. Blocking networks block packets when buffers become full, while non-blocking networks drop packets if this happens, using negative acknowledgments and timeouts to recover.

We consider three switching techniques in particular: wormhole (WH), virtual cut-through (VCT), and slotted ring. Wormhole and virtual cut-through are common switching techniques used in mesh and cube networks [5]. A flow-control signal is back propagated to the previous node when blocking is necessary. The primary difference between the two switching techniques is that wormhole may stall packets across (possibly multiple) links, while a virtual cut-through node will accept the head of a packet only if it can buffer the entire packet and will thus always free up the incoming link. We use wormhole as the representative blocking technique, because we found the performance of wormhole and virtual cut-through to be similar (within 5% across all workloads) and wormhole always performs better in hierarchical ring networks.

For non-blocking networks, we adapt virtual cut-through to drop packets whenever local buffer space is insufficient for holding the incoming packet, and we consider slotted ring techniques. Slotted rings send packets as several equal sized cells that are routed independently.

We evaluate in detail how these switching schemes perform. We found that the size of the buffers in the switching nodes is a critical parameter in both blocking and non-blocking networks; performance suffers if the buffers are too small, or interestingly, if they are too large. For system sizes up to 128, we found that the optimal buffer size varies more with the cache line size than the system size. For virtual cut-through systems with more than 2 levels of hierarchy, the optimal node buffer size also depends on the level in the hierarchy in which the node lies. Finally, we find that the non-blocking slotted networks perform better (by more than 10%) than the other two switching techniques.

We limit our study to 3 levels of hierarchy and 128 nodes. The study can easily be extended to include additional levels of hierarchy and/or more nodes, but we believe that the performance of the system then starts to deteriorate due to bisection bandwidth constraints inherent in ring hierarchies [13].

Manuscript received January, 1996.
Manuscript revised March, 1996.
†The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada M5S 1A4

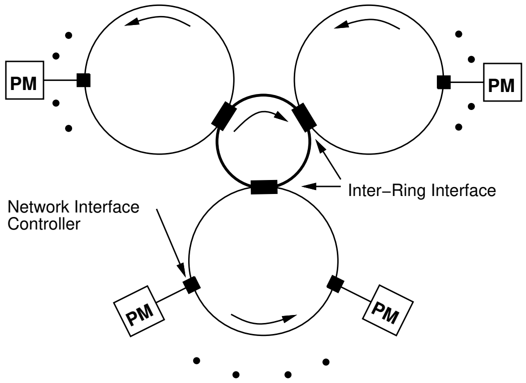

Fig. 1 Example hierarchical ring system with two levels.

2. Simulated System

2.1 System Description

Figure 1 shows a two level hierarchical ring system. It consists of P processing modules connected by a hierarchy of uni-directional rings. Each processing module (PM) contains a processor, a local cache and a portion of the main memory. All processing modules are connected to lowest level rings, which we also refer to as *local rings*. A *global ring* connects several of these local rings. The channel width (data path) of the ring is assumed to be 128 bits wide, based on the NUMAchine design [2].

The system provides a flat, global (physical) address space, and each PM is assigned a unique contiguous portion of that address space, determined by its location. All processors can transparently access all memory locations in the system. The target memory is determined by the address of the memory being accessed. Local memory accesses do not involve the network. Remote memory accesses require that a request packet be sent to the target memory, followed by a response packet from the target memory to the requesting processor.

Packets sent over the rings are of variable size and are transferred in flits, bit-parallel, along a unique path through the ring hierarchy. There are two types of network nodes: *Network Interfaces* (NIC) connect processing modules (PM) to local rings and *Inter-Ring Interfaces* (IRI) connect two rings of adjacent levels. The NIC switches 1) incoming packets from the ring to a PM, 2) outgoing packets from the PM to the ring, and 3) continuing packets from the input link to the output link. The IRI controls the traffic between two rings and is modelled as a 2 x 2 crossbar switch with input FIFO queues for each ring. The routing delay through the NIC and the IRI is assumed to be one network cycle. We assume all communication occurs synchronously; that is, within a clock cycle, each ring node can transfer one flit to the next adjacent node on the same ring (if the link is not being blocked), or to or from the processing module it connects to. No distinction is made between a phit (physical transfer unit) and a flit in our study.

We consider systems of three different sizes: a 16×4, 64 processor system (with 16 PMs and 4 local rings connected to a global ring), a $16 \times 3 \times 2$, 96 processor system, and a $16 \times 4 \times 2$, 128 processor system. Each topology is optimal for the given system size, as determined by a previous study [13]. For each system, we considered three different cache line sizes: 32, 64 and 128 bytes, requiring 3, 5, and 9 flits respectively to transfer them across the network (assuming a channel width of 128 bits and a header flit containing the routing information).

2.2 Simulator

The simulator we use reflects the behavior of the system at the register-transfer level on a cycle-by-cycle basis. It was implemented using the *smpl* simulation library and uses the batch means method of output analysis with the first batch discarded to account for initialization bias [11]. A base simulator was validated against measurements taken from the Hector prototype, a non-blocking hierarchical slotted ring architecture [8],[16]. The base simulator was then extended to model other switching techniques, such as wormhole and non-blocking virtual cut-through.

Our measure of performance is average memory access latency, the time between when a request is first issued and the corresponding response is received. This includes any timeouts and retransmissions that might occur in the non-blocking networks. Latency is inversely proportional to the throughput of the system (i.e. the number of memory accesses per second).

2.3 Benchmark Description

We use synthetic micro-benchmarks to drive our simulator in order to accurately evaluate the performance of the interconnection network under controlled conditions. A series of memory references (i.e. cache misses) is generated at each processor by a variant of the Multiprocessor Memory Reference Pattern (M-MRP) address generator of Saavedra et.al., originally developed to measure the performance of real systems [14]. More formally, an M-MRP is a set of P uniprocessor memory reference patterns, one for each processor, each accessing its own region of the memory address space. The access regions of the processors may overlap.

An M-MRP in our simulation is characterized by two attributes: 1) the size of the memory region, R, accessed by each processor, and 2) the cache miss rate, C, of each processor. By varying each of these attributes, we can exercise the interconnect in a specific and predictable way and measure how the network responds

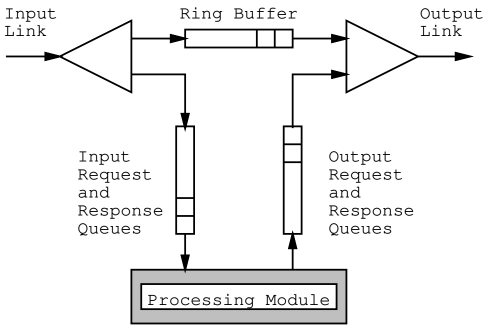

Fig. 2 Network Interface (NIC).

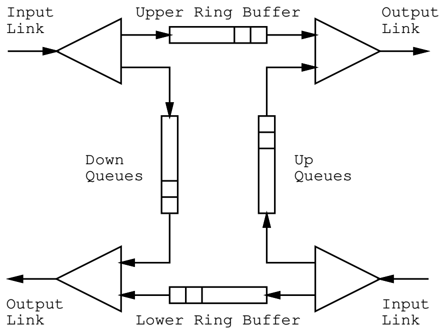

Fig. 3 Inter-Ring Interface (IRI).

under controlled conditions. Parameter R, the size of memory access region, allows us to control different degrees of locality and thus the sharing between processors. Indirectly, R controls the amount of bisection traffic, the traffic through the global ring. R is varied from $1/P$ to 1, where P is equal to the number of processors in the system. We assume that the memory access region of a processor is located such that the processor-local memory is centered in the region. The sequence of memory references in a given memory region is uniformly distributed and independent across the region. Parameter C, the cache miss rate, is varied from $1/100$ to $1/25$ to control the offered load rate in the network. Throughout, we assume a read/write ratio of 7:1.

3. Network Switching

Network Interface Controllers (NICs) are used to connect processing modules to the network at the local ring level, and Interring Interfaces (IRIs) connect lower rings to upper rings. Possible implementations of these network nodes are depicted in Figures 2 and 3.

The NIC has a FIFO *ring buffer* to temporarily store transit packets arriving from the network not destined to the local PM when the output link is currently transmitting another packet. If the ring buffer is empty and no packet is currently being transmitted, then an incoming transit packet will be forwarded to the output link directly, bypassing the ring buffer. The NIC also has a FIFO input queue for storing packets destined for the local PM and a FIFO output queue for storing packets originating from the PM destined for nodes elsewhere in the network. Both of these are split into request and response queues to avoid circular dependencies that can cause deadlock [7]. Priority for transmission to the next node is given to ring packets, either waiting in the bypass buffer or having just arrived from the previous node. Otherwise, if there are packets in one of the output queues then priority is given to response packets over request packets.

The IRI has two ring queues, one for the lower ring and one for the upper ring. It also has a down queue and an up queue. The down queue buffers packets arriving from the upper ring destined for the lower ring, while the up queue buffers packets arriving from the lower ring destined for the upper ring. Both the down and the up queues are also split into request and response queues. Switching takes place independently at the lower and upper ring sides. Priority is given to packets that do not change rings. Arriving transit packets block and are placed in the ring buffer if the output link is in the middle of transmitting a packet from the up or down queue or when packets are already waiting in the ring buffer.

We assume that the size of NIC ring buffers in wormhole switches is fixed at 3, while it is fixed at a size large enough to accommodate the maximum packet size in virtual cut-through switches. In our study, we vary the size of the IRI buffers, but assume that all buffers in an IRI are of equal size. Slotted rings don't require ring buffers (as explained below).

3.1 Wormhole Switching

We use wormhole switching to represent blocking networks [4],[6]. In wormhole switching, a packet is sent as a sequence of flits with the header flit containing the routing information. A flit is the smallest unit on which flow control is performed. There is no distinction here between a flit and a phit (physical transfer unit), since both are assumed to be the same size as the channel (128 bits). As flits are forwarded, a packet may be spread out over multiple links, and a packet is hence sometimes referred to as a *worm* in this context. Since only the head flit of a packet contains routing information, it is essential that the flits of a packet not be interleaved with flits of another packet. The head flit of a packet acquires network resources (links and buffer slots) as it proceeds through the network, while the tail flit then frees them.

When a packet cannot move forward because the next link is busy, it is blocked in place and continues to hold the resource it just acquired. When the local buffers become full, flow control will prevent further transmission over the incoming link, possibly causing

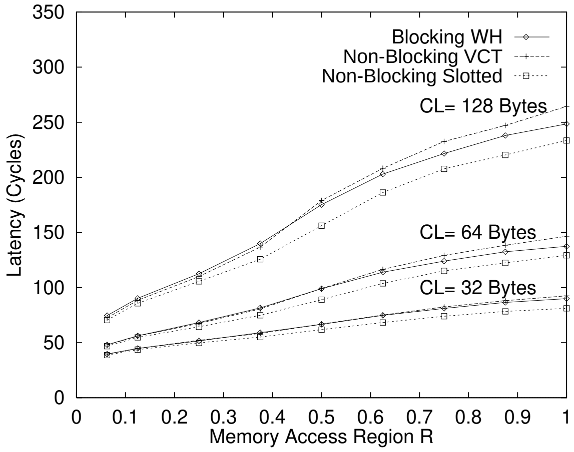

Fig. 4 Memory access latency as a function of memory access region, R, for a 16×4 system with 64 processors and a cache miss rate of $C = 1/25$.

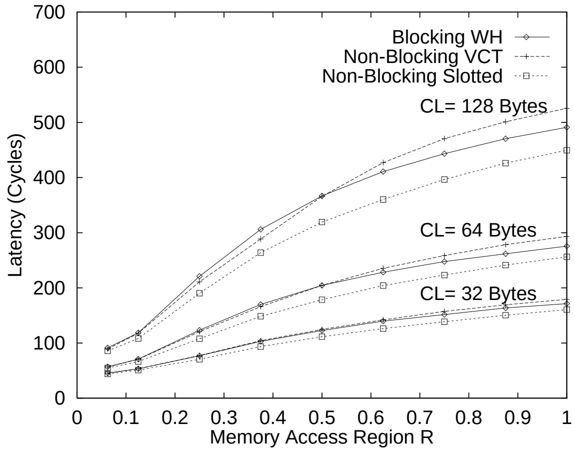

Fig. 5 Memory access latency as a function of memory access region, R, for a $16 \times 4 \times 2$ system with 128 processors and a cache miss rate of $C = 1/25$.

the connected output buffers to fill as well. Under heavy load conditions, a single hot spot may cause tree saturation effects, where contention for a hot spot can back propagate to affect other traffic that has no need for the hot spot resource [12]. The global ring can be a source of tree saturation due to the constant bisection bandwidth of the global ring.

Wormhole networks have poor link utilization under heavy loads because they saturate from contention [15] well before they exhaust their bandwidth. However, under light loads, link contention occurs infrequently and when it occurs, tends to be short lived; having packets wait in the network is less expensive than dropping them and having them to be resent at a later time.

An often mentioned advantage of wormhole switching is that it requires only small buffers at network nodes due to the availability of blocking, thereby allowing the implementation of the router to be small and fast [4], [6]. However, we will show that wormhole switching nodes in hierarchical ring need more than a few buffer slots for optimal performance, but that too many of them can also hurt performance.

3.2 Virtual Cut-though Switching

We have adapted the virtual cut-through switching technique [10] to realize non-blocking networks. In virtual cut-though, as in wormhole, a packet is sent as a sequence of flits that may not be interleaved with other packets, with the head flit containing routing information. A virtual cut-through switching node, however, will accept the head flit of a packet only if it has enough buffer space to accept the entire packet. The non-blocking variant drops the packet whenever it cannot accept it. If the dropped packet is a request, then a Negative Acknowledgement (NACK) packet is sent back to the source NIC. Because ring transit packets are given priority, packets are dropped only when IRI up/down queues or NIC input queues are full.

Dropped packets are recovered by using NACK packets and end-to-end timeouts. Whenever a request packet is sent, the source NIC keeps a copy of the request and starts a timer. If a response is received before the timer expires, then the timer is cleared and the copy can be discarded. If the source NIC receives a NACK for its request, then it resends the request and resets the timer. If the timer expires, then it is assumed that either the response packet or NACK was dropped, and the request is resent with simultaneous resetting of the timer. It should be noted that the timeout in such systems has to be chosen large, larger than the maximum round trip latency (including all possible buffering).

3.2.1 Slotted Ring Switching

Slotted ring networks are considerably different from either virtual cut-through or wormhole routed networks [1], [2], [9], [16]. In a slotted ring, packets are divided into equal sized cells that are routed independently. Each cell, the size of a phit (physical transfer unit), has its own routing information: the first cell of a packet carries the full target memory address, while the remaining cells of the packet only identify the destination PM and contain sequencing information. The fact that the target PM address and sequencing information must be repeated in each cell allows the cells to be routed independently, but adds approximately 10% overhead to the size of the data paths in a 128 processor system (7 bits to address 128 processors and 4 bits to sequence up to 9 flits [2]). Correspondingly, we assume the slotted ring channel is 140 bits wide.

Routers for slotted ring networks are similar to those of virtual cut-through or wormhole routers except that there are no ring buffers. Because links are acquired and then released in the same clock cycle for

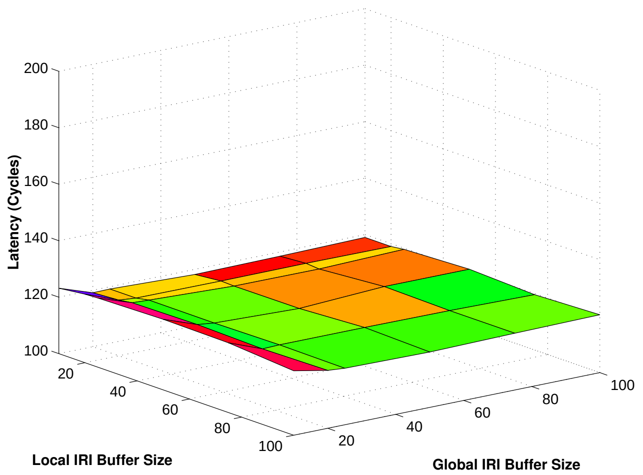

Fig. 6 Memory access latency as a function of IRI buffer size for access pattern with high locality ($R = 0.25$), a 128 ($16 \times 4 \times 2$) processor wormhole switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

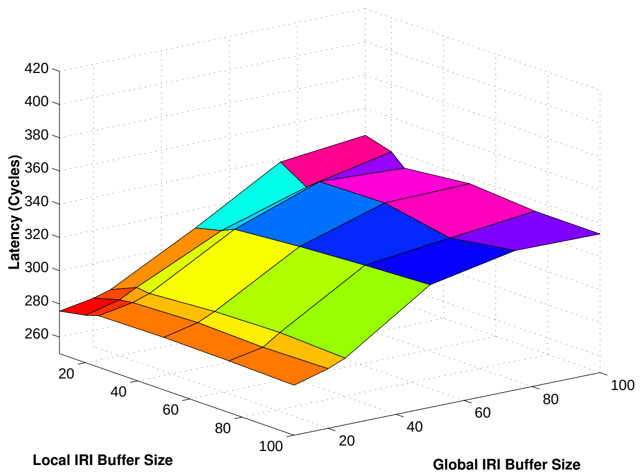

Fig. 8 Memory access latency as a function of IRI buffer size for access pattern with low locality ($R = 1.0$), a 128 ($16 \times 4 \times 2$) processor wormhole switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

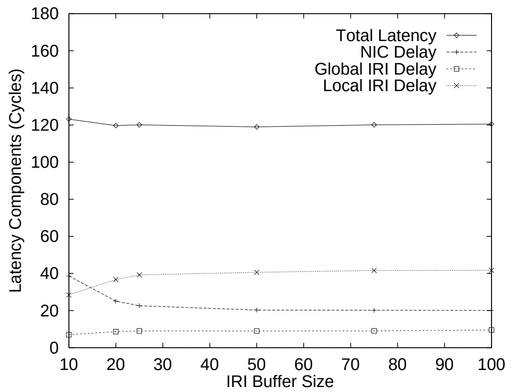

Fig. 7 Components of the memory access latency for access pattern with high locality ($R = 0.25$), a 128 ($16 \times 4 \times 2$) processor wormhole switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

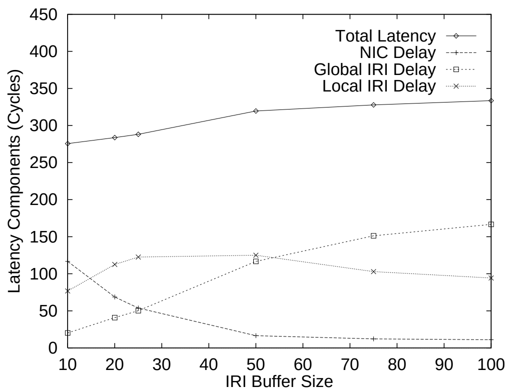

Fig. 9 Components of the memory access latency for access pattern with low locality ($R = 1.0$), a 128 ($16 \times 4 \times 2$) processor wormhole switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

the transmission of a single cell, an incoming transit cell can always be transmitted on the outgoing link without having to be buffered. Cells are dropped only when they cannot be accommodated in IRI up/down queues or NIC input queues. The destination NIC that reassembles packets destined for the attached PM will not accept a packet unless it arrives complete. We assume that a network node will discard all of the remaining cells of a packet once one cell has to be dropped. Again, NACKS and timeouts are used to recover packets whose cells were dropped.

4. Comparative Performance

In this section, we compare the performance of the switching techniques described above. Each system we consider in this section is configured with optimal sized buffers, as discussed in Section 5.

Figure 4 plots the round trip latency of the average memory access as a function of the memory access region size, R, for a two-level, 64-processor (16×4) hierarchical ring network and the relatively high cache miss rate of $C = 1/25$. There are three sets of curves in each plot, one for each cache line size of 32, 64, and 128 bytes. Within each set, there is a curve for each type of switching technique considered.

Figure 4 shows that the slotted ring performs better than the other two protocols for all cache line and memory access region sizes. With high locality in the memory access pattern ($R < 0.5$), the performance improvement of slotted ring over the other two switching techniques is low to moderate (5%–10%), and for low memory access locality ($R > 0.5$), the improvement in performance is moderate to high (10%–15%). The difference in performance between slotted ring and the other two switching techniques is largely independent

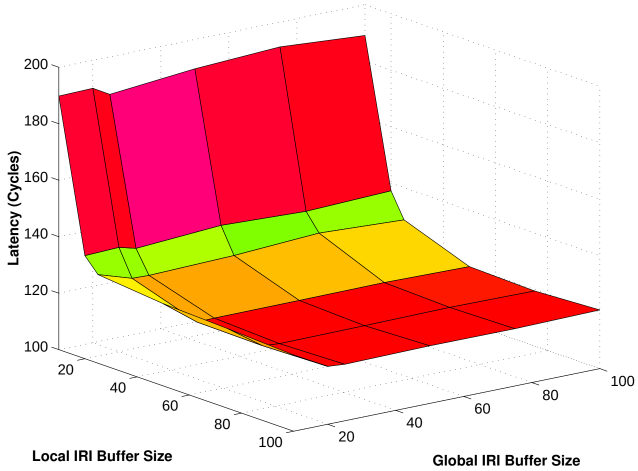

Fig. 10 Memory access latency as a function of IRI buffer size for access pattern with high locality ($R = 0.25$), a 128 ($16 \times 4 \times 2$) processor virtual cut-through switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

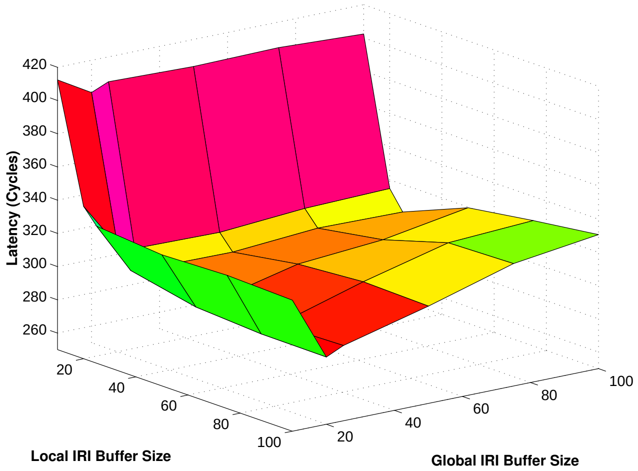

Fig. 12 Memory access latency as a function of IRI buffer size for access pattern with low locality ($R = 1.0$), a 128 ($16 \times 4 \times 2$) processor virtual cut-through switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

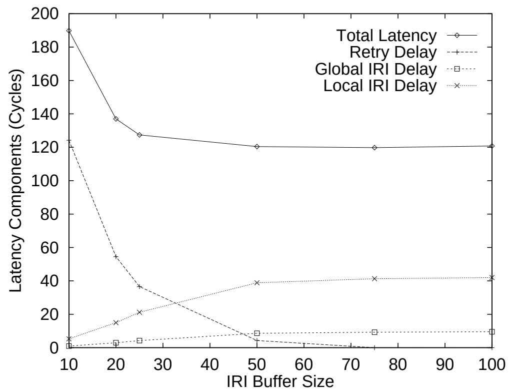

Fig. 11 Components of the memory access latency for access pattern with high locality ($R = 0.25$), a 128 ($16 \times 4 \times 2$) processor virtual cut-through switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

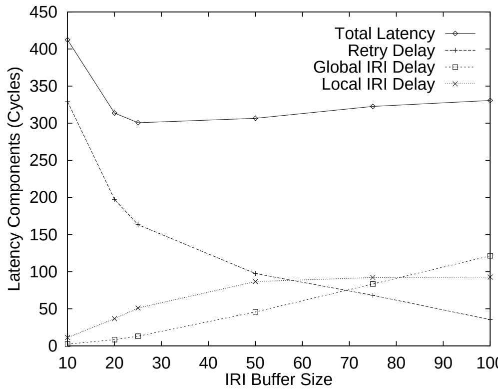

Fig. 13 Components of the memory access latency for access pattern with low locality ($R = 1.0$), a 128 ($16 \times 4 \times 2$) processor virtual cut-through switched system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

of the cache line size.

Figure 5 shows that slotted ring also performs better than wormhole and cut-through on a larger 128 ($16 \times 4 \times 2$) processor system (by 5%–10% for $R < 0.5$ and by 10%–16% for $R > 0.5$). It is also interesting to note that for this system size, virtual cut-through performs marginally better than wormhole (by about 6%) when $R < 0.5$, while wormhole performs better than virtual cut-through when $R > 0.5$, particularly for the large cache line size of 128 bytes.

5. Performance Impact of IRI Buffers

The performance of hierarchical ring networks is sensitive to the size of the IRI buffers in the nodes. For optimal performance of virtual cut-through ring hierarchies, the IRI buffer sizes must be chosen independently at each level, and for both wormhole and virtual cut-through, larger than optimal buffer size hurts performance.

The 3-dimensional figures 6, 8, 10, 12, 14, and 15 depict the average memory access latencies for different IRI buffer sizes. The system in this case has 128 processors ($16 \times 4 \times 2$), a cache line size of 64 bytes, and a relatively high cache hit rate of $C = 1/25$. The size of the local (level–1) and global (level–2) IRI buffers are varied along the x and y axes respectively.

5.1 Wormhole Switching

Figure 6 assumes a relatively high degree of locality in the memory access pattern with $R = 0.25$, while figure 8 assumes poor locality with $R = 1.0$. The two figures show that the size of the local IRI buffers are not overly critical to performance. The global IRI buffer sizes also do not affect performance much for high memory ac-

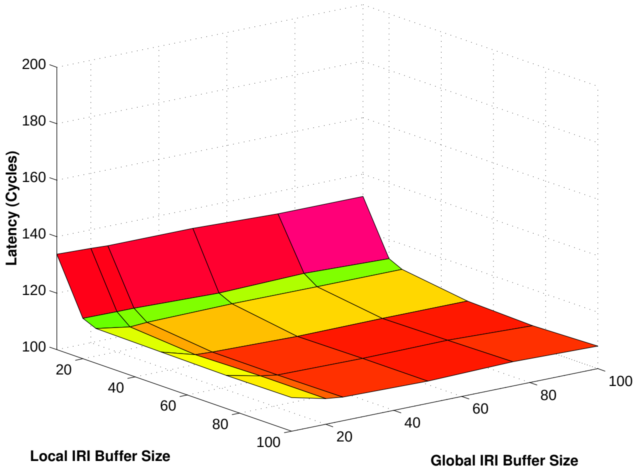

Fig. 14 Memory access latency as a function of IRI buffer size for access pattern with high locality ($R = 0.25$), a 128 ($16 \times 4 \times 2$) processor slotted ring system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

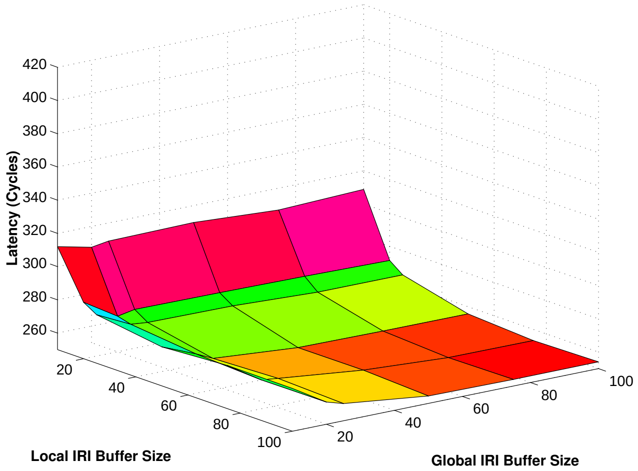

Fig. 15 Memory access latency as a function of IRI buffer size for access pattern with low locality ($R = 1.0$), a 128 ($16 \times 4 \times 2$) processor slotted ring system, a 64 byte cache line size, and a cache miss rate of $C = 1/25$.

cess locality[†]. But, for low memory access locality (figure 8), performance worsens with increasing global IRI buffer sizes. This behavior is partly due to the particular configuration of the system we are modelling, where the global ring connects only two mid-level rings and where a blocked worm at the global ring can prevent other traffic from utilizing the ring. In fact, our results indicate (not shown) that the utilization of the global ring decreases as the global IRI buffer sizes are increased. It is better to have the blocking of worms occur away from the hot spot, in this case the global ring.

Figures 7 and 9 show the components that make up the memory access latency for high and low locality. In these figures it is assumed that the size of the buffers in both local and global IRIs are the same. The figures show that for both high and low locality, the delays due to queueing and blocking at the global IRIs increase with the buffer sizes (although it is less pronounced for the high locality access pattern), while the delays at the NICs decrease as the buffer sizes become large. The local IRI delays increase initially in both cases as the buffer size is increased, and it attains a steady value for high locality memory access patterns, while it starts to decrease after reaching a maximum for low locality memory access patterns. In the latter case, the decrease in NIC and local IRI delays does not make up for the increase in global IRI delays when buffer size is large and hence the increase in total latency.

The difficulty with wormhole switched hierarchical ring networks, as evident in Figure 8, is that a larger global buffer size hurts performance for low memory locality access patterns.

[†]The effect of the global IRI buffer size is more pronounced with large cache line sizes.

5.2 Virtual Cut-through Switching

Figures 10 and 12 show the memory access latency for a non-blocking virtual cut-through hierarchical ring system with 128 processors. In this case, the size of the local IRI buffers is critical. For high access locality (figure 10), the larger the global and local IRI buffers the better. For low access locality, performance improves dramatically when local IRI buffer sizes are increased initially, but then becomes more sensitive to the global IRI buffer size. After this point, performance improves initially with an increase in global IRI buffer size, but then begins to gradually become poorer as the global IRI buffer sizes become large. The optimal global IRI buffer size is found to be around 20.

No single buffer size for both global and local IRIs is optimal across all access patterns. However, it is possible to identify an IRI buffer configuration (with different global and local IRI buffer sizes) that performs reasonably well across a wide range of access patterns; for the system considered this configuration lies in the vicinity of (50,20).

Figures 11 and 13 depict the latency components as a function of buffer size. As the IRI buffer sizes increase, the latency component due to retries (for dropped packets) drops to a very small value. However, local and global IRI delays increase with buffer size. While the local IRI delays tend to stabilize after an initial increase, for access patterns with poor locality global IRI delays continue to increase resulting in an increase in latency.

5.3 Slotted Rings

Figures 14 and 15 depict the memory access latency for a non-blocking slotted ring system with 128 processors. In this case, the performance is significantly affected by

Cache Line size	System size	Blocking WH		Non-Blocking VCT		Slotted	
		L	G	L	G	L	G
32 Bytes	64 P	6		25		>25	
	96 P	6	6	25	20	>25	>25
	128 P	6	6	25	20	>25	>25
64 Bytes	64 P	10		50		>50	
	96 P	10	10	50	20	>50	>50
	128 P	10	10	50	20	>50	>50
128 Bytes	64 P	18		100		>100	
	96 P	18	18	100	20	>100	>100
	128 P	18	18	100	20	>100	>100

Table 1 Optimal local(L) and global(G) IRI buffer sizes for different system and cache line sizes. WH: Blocking wormhole; VCT: Non-blocking virtual cut-through; Slotted: Non-blocking slotted ring.

the size of the local IRI buffer size and to a lesser extent by the size of the global IRI buffer size. However, unlike the case of non-blocking virtual cut-through, the variation in latency is small across buffer sizes. The most important observation is that in contrast to wormhole and virtual cut-through, the performance improves continuously with increasing global and local IRI buffer sizes for both low and high locality memory access patterns, until it reaches a minimum and then remains constant thereafter.

5.4 Other Cache Line and System Sizes

We also simulated 64, 96 and 128 processor systems with cache line sizes of 32, 64 and 128 bytes to consider how these parameters affect optimal buffer sizes. For each switching technique, the basic shape of the memory latency graph stayed the same, but was shifted. The optimal IRI buffer sizes were largely unaffected by the system sizes we considered. However, we found that the optimal IRI buffer sizes tended to double when the cache line size is doubled. This is true for both the global and local IRI buffers except for non-blocking virtual cut-through networks, where the optimal global IRI buffer sizes do not increase with cache line size. These results are summarized in Table 1 that lists the optimal local and global IRI buffer sizes for a variety of parameters. In the case of the slotted ring, the values listed represent minimal values, since latencies monotonically decrease with buffer size.

6. Concluding Remarks

From our simulation study, we conclude that the slotted ring switching technique is a good choice for hierarchical rings, not only because it performs better than other switching techniques across all access patterns and system configurations, but perhaps more importantly, because it is easier to identify optimal buffer sizes for the switching nodes independent of *i*) the memory access patterns, *ii*) the network size, and *iii*) the level in the hierarchy in which the nodes lie. This makes it possible to be conservative in choosing the size of the buffer; IRI's with a buffer size of say 100 will provide the best performance for all system sizes up to 128 processors, regardless of the level at which they are employed. In contrast, the optimal buffer size in blocking wormhole is sensitive to the access pattern and in virtual cut-through is sensitive to the levels in the hierarchy. In both these networks, choosing a buffer size too large will hurt performance.

References

[1] L.A. Barroso and M. Dubois, "Performance evaluation of the slotted ring multiprocessor," *IEEE Trans. on Computers*, 44(7), pp. 878-890, 1995.
[2] S. Brown et al, "The NUMAchine multiprocessor", CSRI-TR-324, CSRI, University of Toronto, 1995.
[3] H. Burkhardt et al., "Overview of the KSR1 computer system," KSR-TR 9202001, Kendall Square Research, 1992.
[4] A.A. Chien, "A cost and speed model for k-ary n-cube wormhole routers," in *Proc. Symp. on Hot Interconnects*, 1993.
[5] W.J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE Trans. on Computers*, 39(6), pp. 775-785, 1990.
[6] W.J. Dally and C. L. Seitz, "The Torus routing chip," *Journal of Distributed Computing*, 1(3), pp 187-196, 1986.
[7] D.B. Gustavson, "SCI and related standards projects," *IEEE Micro*, 12(1), pp. 10-22, 1992.
[8] M. Holliday and M. Stumm, "Performance evaluation of hierarchical ring-based shared memory multiprocessors", *IEEE Trans. on Computers*, 43(1), pp. 52-67, 1994.
[9] A. Hooper and R.C. Williamson, "Design and use of an integrated cambridge ring", *IEEE Journal on Selected Areas of Communication*, 1(5), pp. 775-784, 1983.
[10] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Computer Networks*, 3(4), pp. 267-286, 1979.
[11] M.H. MacDougall, *Simulating Computer Systems: Techniques and Tools*, MIT Press, 1987.
[12] G. Pfister and A. Norton, "Hot spot contention and combining in multistage interconnect networks," *IEEE Trans. on Computers* C32(10), pp. 943-948, 1995.
[13] G. Ravindran and M. Stumm, "Hierarchical ring topologies and the effect of their bisection bandwidth constraints," in *Proc. Intl. Conf. on Parallel Processing*, pp. I/51-55, 1995.
[14] R.H. Saavedra et al, "Characterizing the performance space of shared memory computers using micro-benchmarks," in *Proc. Hot Interconnects*, 1993.
[15] K.G. Shin and S.W. Daniel, "Analysis and implementation of hybrid switching," in *Proc. Intl. Conf. on Computer Architecture*, pp. 211-219, 1995.
[16] Z.G. Vranesic et al, "Hector: A hierarchically structured shared-memory multiprocessor," *IEEE Computer*, pp. 72-78, 1991.

Govindan Ravindran is a Ph.D. candidate in the Department of Electrical and Computer Engineering at the University of Toronto, Toronto, Canada, where he holds canadian commonwealth fellowship.
His research interests include multiprocessor architectures and real time systems. He is a student member of IEEE Computer Society, the Association of Computing Machinery and the Instrument Society of America.

Michael Stumm is a professor in the Department of Electrical and Computer Engineering and the Department of Computer Science at the University of Toronto, Toronto, Canada. He received a diploma in mathematics and a Ph.D. in computer science at the University of Zurich in 1980 and 1984, respectively.
Stumm's research interests are in the areas of computer systems, in particular, operating systems for distributed and parallel systems and multiprocessor architectures. He is a member of the IEEE Computer Society and the Association of Computing Machinery.