Simulative Performance Analysis of Distributed Switching Fabrics for SCI-based Systems

M. Sarwar[1] and A.D. George[2]

High-performance Computing and Simulation (HCS) *Research Laboratory*
[1] Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering
[2] Department of Electrical and Computer Engineering, University of Florida

Abstract

This paper presents the results of a performance study on 1D and 2D *k*-ary *n*-cube switching fabrics for the Scalable Coherent Interface through high-fidelity simulation with analytical verification. These topologies have been widely cited in the literature as the target of studies on parallel algorithms and represent a promising basis for the design of efficient distributed switching fabrics for SCI. Performance is characterized for multiprocessor networks composed of simple SCI rings, counter-rotating SCI rings, unidirectional and bidirectional SCI tori, and SCI tori with rings of uniform size.

Keywords: BONeS; Discrete-event simulation; High-performance networks; Scalable Coherent Interface; Switching fabrics

1. Introduction

The performance and scalability of high-speed computer networks have become critically important characteristics in the design and development of advanced distributed and parallel processing systems. Many applications require or benefit from the use of an interconnect capable of supporting shared memory in hardware, and chief among such interconnects for multiprocessors is the Scalable Coherent Interface. However, in order for SCI interconnects to scale to ever larger system sizes and support a host of embedded and general-purpose applications, a distributed switching fabric is required that will scale with the number of nodes. One of the most promising families of topology for distributed switching fabrics is the *k*-ary *n*-cube topology, a family originally investigated for high-end supercomputing and referenced widely in the literature as the target for algorithm mapping.

The SCI standard is targeted towards increasing the bandwidth of backplane buses and became an IEEE standard in March of 1992 [1]. It improves on the bandwidth of buses by using high-speed, ring-connected, point-to-point links. With a link speed of 1 GB/s (i.e. a gigabyte per second), addressing for up to 64K nodes, and a cache-coherence protocol for distributed shared-memory systems, the popularity of SCI for use in large multiprocessors has continued to increase. Sequent, Cray, and HP-Convex are among the parallel computer vendors that have developed proprietary implementations of SCI for their high-end systems. Sequent developed the IQ-link implemented in their Numa-Q 2000 system to connect groups of four processors in a ring structure [2]. Cray developed the SCX channel, also known as the GigaRing, capable of sustained half-duplex bandwidths of 900MB/s [3-4]. The HP-Convex Exemplar Series uses the SCI-based Coherent Torroidal Interconnect (CTI) to interface hypernodes consisting of 8 processing units each.

SCI has also gained recognition in the workstation cluster market. To date, Dolphin Interconnect Solutions has emerged as the leading manufacturer of SCI adapter cards and switches for clusters. The Dolphin switch relies on a bus-based internal switch architecture called the B-link. The B-link is capable of a bandwidth ranging from 200MB/s to 400MB/s depending on the operating clock speed. Sun has adopted the Dolphin implementation of SCI, dubbed CluStar, for their Enterprise Cluster systems. Recently, Dolphin introduced a dual-ported PCI/SCI adapter card from which to construct unidirectional 2D torus topologies for SCI. Data General, in collaboration with Dolphin, has developed a chipset for their AV20000 Enterprise server to interface SCI to Intel's Standard High Volume (SHV) server nodes [5]. In addition, Dolphin and Siemens jointly developed a PCI-SCI bridge to be used in the I/O subsystems of the Seimens RM600E Enterprise Server systems.

While Dolphin's B-link bus provides a cost-effective approach to internal switch architecture, it is limited in its scalability and support for multidimensional network topologies. In this paper, a crossbar-based SCI switch model is presented that does not suffer from these limitations. The switch uses routing tables that are automatically generated at startup and which guarantee the shortest path to each packet's final destination. The performance of *k*-ary *n*-cube systems is explored by conducting experiments with a fixed ring size over a variable number of total nodes. The *k*-ary *n*-cube family consists of direct networks with *n* dimensions and *k* nodes per dimension, and members include rings, meshes, tori, hypercubes, etc. These networks provide excellent scalability through a constant node degree (i.e. fixed number of ports per node despite the size of the system), and low latencies through smaller diameters (i.e. fewer hops when transferring packets from source to destination). Additionally, they provide topologies that have served as targets for many studies on the mapping of parallel algorithm graphs for multiprocessing and multicomputing. Work by Dally [6], Chung [7], and Reed et al. [8] provide detailed analytical analyses of generic *k*-ary *n*-cube networks.

By contrast, in this paper we concentrate on a simulative approach of applied research to determine the performance of *k*-ary *n*-cube networks constructed with SCI. Through simulation with high-fidelity models for SCI, more accurate results can be obtained to study the relationship and impact of selected switching topologies for SCI multiprocessor networks. The remainder of this paper is organized as follows. Section 2 introduces the Scalable Coherent Interface and its basic operation. Section 3 describes the SCI switch model, and Section 4 presents the performance simulation results for several *k*-ary *n*-cube topologies. Finally, conclusions and directions for future research are discussed in Section 5.

2. Scalable Coherent Interface

The basic SCI topology is a ringlet. The ringlet is constructed using SCI interfaces at each node. The nodes communicate using unidirectional point-to-point links. Each link is 18-bits wide – 16 data bits also referred to as an SCI symbol, a flag bit, and a clock bit. The flag bit is used to identify the type of packet being transmitted on the data lines while the clock provides synchronous communications between nodes. SCI uses a split-transaction protocol where each transaction consists of two subactions. The subaction can be either a request or a response from

a source, followed by an echo indicating whether the request or response was accepted at the destination, as illustrated in Fig. 1.

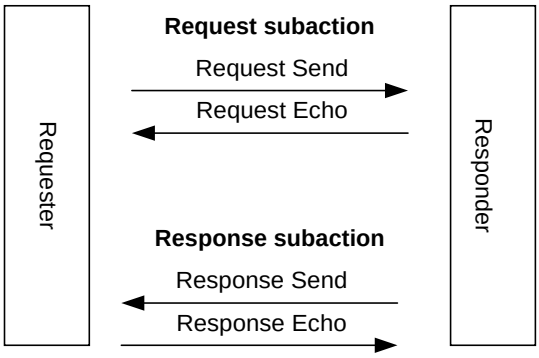

Fig. 1. Split transaction used in SCI.

A typical transaction on a single SCI ring begins with a *request-send* packet from the requestor to the responder. The responder then returns a *request-echo* packet to the requester indicating that the request has been received. After processing the request, the responder sends a *response-send* packet to the requester and receives a *response-echo* packet in return. Some subactions, such as the *move*, do not have a corresponding response subaction.

The basic architecture of an SCI interface is illustrated in Fig. 2. Incoming packets to the interface pass through an address decoder. If the packet is destined for the local node, the decoder places it into the request or response input queue. If the packet is destined for another downstream node, it is forwarded to the bypass FIFO. To output a packet, the SCI node must have sufficient free space in its bypass FIFO to hold all incoming symbols. When there are no packets waiting in the output queue or there is insufficient free space in the bypass FIFO for the output queue data to be sent, data from the bypass FIFO is transmitted on the node's output link. If the bypass queue is empty, then *idle* symbols are transmitted. *Idle* symbols also carry flow-control information and at least one must precede any *send* or *echo* packet. This flow control information is used to inhibit upstream nodes from sending data when the bypass FIFO must be emptied to allow the output queue to be emptied.

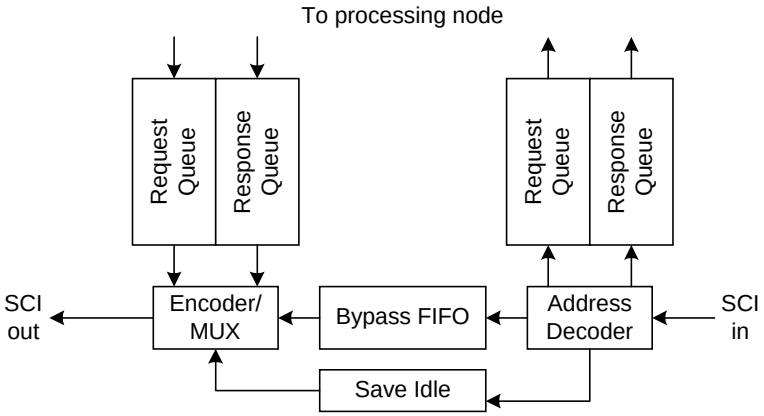

Fig. 2. The SCI interface.

Larger SCI networks are based on multiple ringlets connected together to create more complex topologies through the use of agents. An agent is essentially an SCI-to-SCI bridge used to interconnect two or more rings. The use of agents alters the transaction communication protocol slightly as illustrated in Fig. 3. The figure depicts two nodes communicating via an agent. The agent serves two purposes – it is the responder for one ringlet and the requester for the second. In step 1, the agent accepts the request on behalf of the responder. In step 2, the request is forwarded onto the responder's ring by the agent. Step 3 shows the agent accepting the response and forwarding it back to the requestor on the first ring in step 4.

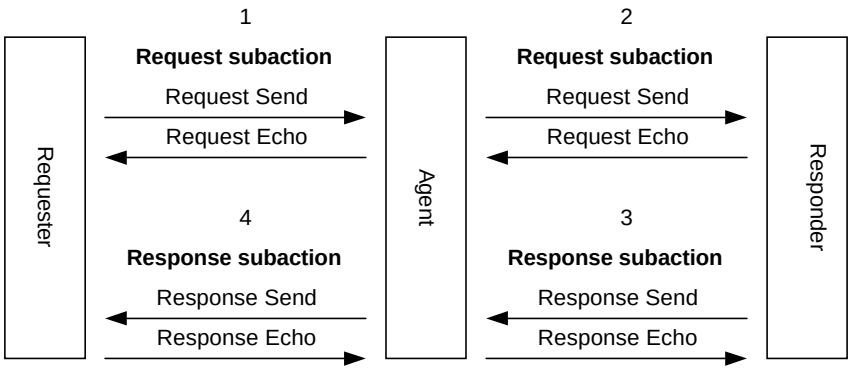

Fig. 3. Transaction protocol through an agent.

In order to explore the performance of SCI for different topologies in the distributed switching fabric, high-fidelity SCI interface and agent-based switch models were constructed. These event-driven models are accurate down to the SCI clock cycle, providing a level of detail virtually identical to the real network hardware. Switching times are dependent on the packet length and are not assumed to be constant for all packets. Packet processing times in the host nodes attached to each of the switches are not included, since the main interest is in this study is on the performance of the SCI interconnection network.

3. SCI distributed switch model

The proposed switch model uses a crossbar in order to achieve a higher throughput than the B-link bus. The simplest of all SCI switches consists of two back-to-back interfaces where the input queues of one feed into the output queues of the other and vice versa. This interface configuration is limited in that it only allows the bridging of two ringlets. In order to increase the number of rings connected at a single point, a switch must provide additional SCI interfaces, one per attached ring.

Wu and Bogaerts have studied several SCI switch models consisting of internal rings, buses, and crossbars for use in multistage SCI networks [9-11]. Their results confirm that although high throughputs can be achieved using an internal bus, better performance is attainable by using a crossbar-based switch. The two crossbar-based switches studied by Wu and Bogaerts are the CrossSwitch and Switchlink. The CrossSwitch provides a single output queue for each output port, regardless of the number of input ports. This structure requires input ports to block in cases where there is contention on an output queue. In contrast, the SwitchLink [12] provides one output queue for each

input port at every output port. This multi-queue approach provides a private path from each input queue to one and only one output queue, hence eliminating contention for a single output queue and minimizing head-of-line (HOL) blocking.

The switch model developed for this research is based on the multi-queue concept used in the SwitchLink. Fig. 4 illustrates the switch model in a 4-port configuration. By incorporating a crossbar and multiple SCI interfaces into each node, several rings can be connected at a single point. In addition, if the switch's ability is extended to service a processing unit, the resulting switch/node model can be used to construct and simulate any standard ring-based topology with distributed switching. Examples of such topologies include dual-ring topologies, the *k*-ary *n*-cube family of topologies, and Manhattan street networks. The basic premise of these topologies is to use each node not only as a processing unit but also as a switching point for packets.

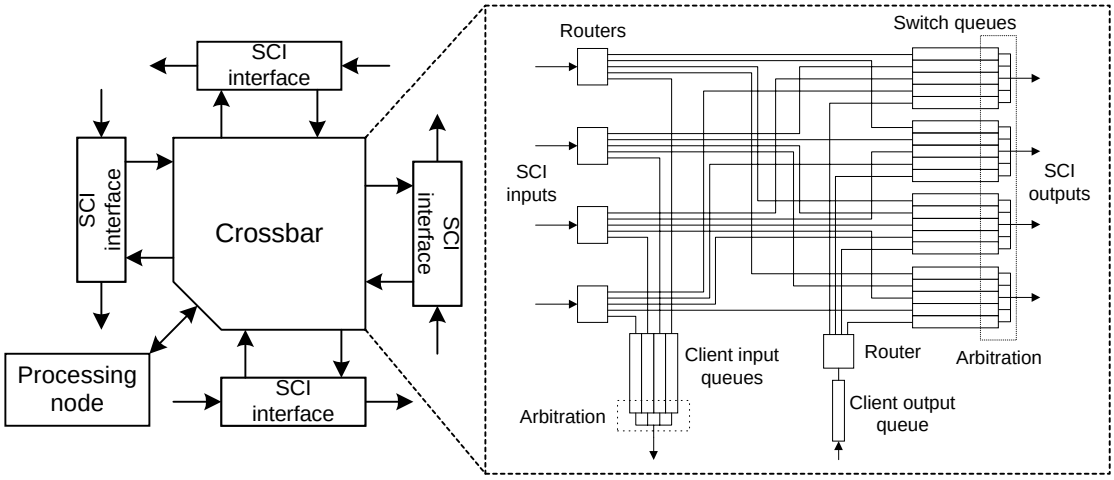

Fig. 4. Switch model in a 4-port configuration.

4. Simulation experiments and results

This section presents simulation results that are used both to verify the correctness of the model and to evaluate the performance of SCI in various topologies. The models were constructed using the Block Oriented Network Simulator (BONeS), a commercial CAD tool from Cadence Design Systems [13-14]. BONeS is an event-driven simulation engine primarily used for modeling and analysis of computer systems and communications networks. Models are created using core library blocks and custom-built blocks. Each block has an equivalent C++ implementation that, when executed, simulates the function of the block. The BONeS model created for this research is a high-fidelity performance model. The SCI protocol is modeled down to the bit level and functionally implemented within the model. Unlike coarse-grained models that assume a fixed packet size and hence fixed switching, queuing, and transmission times, the BONeS model relies on the packet size to measure time. Hence, all delays that a packet would experience if it existed in a real SCI network are modeled to a high degree of accuracy.

Each of the simulations described in this section was executed for a total of 200 microseconds of simulated time to achieve measurements in steady state. It was determined through experimentation that the models reached an

equilibrium point with a constant traffic flow after approximately 20-30 microseconds into the simulation. The time to reach this equilibrium point depended on the size of the topology, with the largest topology reaching equilibrium after approximately 30 microseconds. The simulation time for each experiment in the study was approximately 10-50 minutes depending on the size and complexity of the topology. The model shown in Fig. 4 is comprised of over 4300 BONeS blocks including many custom-built blocks. This large number of blocks accounts for the large simulation times as the topology size increases.

The switch models constructed in BONeS consist of two-port, three-port, and four-port variants. The functions of the switch include servicing the attached processing node(s) as well as providing distributed switching capabilities to the entire network. Table 1 shows the value of several parameters common to all simulated topologies.

Table 1
Model parameter assignments

Parameter	Value
Link Speed	1 GB/sec (as defined by the SCI standard)
Queue Lengths	5 packets deep
Routing Decision Time	10 ns
Packet Switching Time	2 ns per 2-byte symbol
Clock Frequency	500 MHz
Packet Destinations	Uniformly distributed random destination
Transaction Type	64-byte Move (64-byte payload, 16-byte overhead)
Mean Time Between Packets	64/(Total Offered Load/*N*) seconds

Most of the parameters in Table 1 are self explanatory, but a few are worth noting. The routing decision time is the penalty imposed by a multi-port switching node when a packet must be switched off one ring and onto another. The packet switching time is the amount of time it takes to switch one 2-byte SCI symbol. Each port of the switch has a switching capacity of 1 GB/s. Since SCI uses 16-bit wide data lines, two bytes of data are transmitted each clock cycle, and hence the 500 MHz clock yields a link speed of 1 GB/s.

Throughput measurements are obtained by summing the data received by all nodes for a fixed period of time, and dividing the total number of bytes received by the period of time, yielding a total effective throughput for the entire system. This throughput does not include any overhead due to packet headers, idle symbols, echos, etc. The total offered load is divided equally among the nodes and sent to randomly distributed destinations. Since a responseless, 64-byte *move* operation was used, only one-way latencies are measured. The *move* subaction was selected in order to maximize the data throughput of the network. The latency of a packet is measured from the time the packet is generated and placed on the output queue until the time it reaches the destination node and is removed from the input queue.

Three sets of experiments were performed on the SCI model to gauge its performance under a variety of network topologies. First, unidirectional and bidirectional ring topologies were simulated and used to verify the models using the simpler, more easily predictable, 1D configurations. Next, unidirectional and bidirectional 2D tori topologies were constructed to study SCI under a more scalable environment. Finally, a uniform-ring torus topology that is inherently better at tolerating faults is studied to compare its performance with a conventional torus.

4.1 One-dimensional topologies (k-ary 1-cubes)

In order to verify analytically the performance of the SCI model, a unidirectional-ring topology can be used. To determine the theoretical performance of such a ring, consider an *N*-node ring transmitting 64-byte *move* packets. The *move* packets consist of 64 bytes of data and 16 bytes of overhead including the source and destination node addresses, time-out information, and a CRC. The SCI specification also requires that all packets be followed by a 2-byte *idle* symbol for flow-control purposes. The request packet can therefore be considered to contain 82 bytes or 41 SCI symbols (i.e. a 40-symbol request packet and one idle symbol). Similarly, *echo* packets require 10 bytes or 5 symbols (i.e. a 4-symbol *echo* and one *idle* symbol).

Because of the register-insertion structure of SCI, it can be difficult to determine the exact performance of a ring under heavy loading because packets may have to wait in output queues for an undetermined amount of time. However, the throughput at saturation can be approximated by considering a store-and-forward SCI ring. In such a ring, each node would transmit a complete packet to its downstream neighbor. The downstream node would wait for the entire packet to be received before forwarding it to the next node. Such a system would have poor latency performance compared to the pipelined communication in SCI. However, if all packets are the same size and travel the same distance to their destinations, then a store-and-forward system can make maximum use of the bandwidth. At the beginning of a "cycle", each node transmits an entire packet to its downstream neighbor. All nodes will receive the incoming packet from their upstream neighbor at the same time and will simultaneously begin sending the packet to the next node downstream in the next "cycle". At some point, all packets will simultaneously reach their destinations and each will be replaced on the network by a corresponding *echo* packet. The *echo* packets will continue around the ring in the same manner until the original source nodes receive the acknowledgement of the *echo* at the same time and can then send another packet.

With a random distribution of data, each packet on a unidirectional SCI ring will traverse an average of *N/2* links before arriving at its destination. Using this expression and the packet sizes mentioned above, the analytical estimate for best-case effective throughput with an SCI ring using a store-and-forward approximation is found to be:

$$Peak\ Effective\ Throughput = \frac{(N\ nodes)\times(64\ bytes)}{\left(41\times\left(\frac{N}{2}\right)+5\times\left(N-\frac{N}{2}\right)symbols\right)\times(2\ ns/symbol)} = 1.39\ GB/s \qquad (1)$$

The numerator of this equation represents the amount of data that is transmitted simultaneously, 64 bytes $\times$ *N* nodes. The denominator calculates the amount of time required for all *N* transactions to complete. This time is split into the time required for the 41-symbol request packets to arrive at their destinations halfway around the ring and the time required for the 5-symbol *echo* packets to return. As expected, the peak throughput is independent of ring size and has a value of 1.39 GB/s.

To verify the correct operation of our CAD model in terms of the SCI standard, a simple unidirectional-ring topology was simulated. Fig. 5 shows the effective throughputs obtained from simulation experiments with the 4-, 6-, 8-, and 10-node system models with unidirectional rings. As illustrated in the figure, the total effective throughput

with any of the unidirectional rings approaches 1.35 GB/s at saturation. The results of the single-ring simulations closely agree with the analytical results and thus help to verify the accuracy of the model.

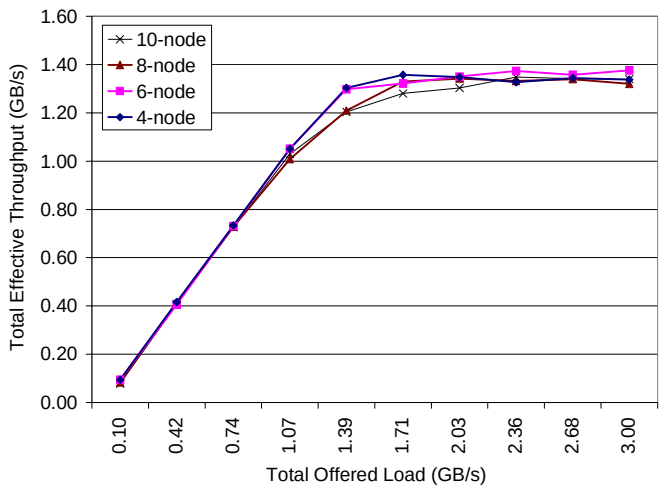

Fig. 5. Throughput versus offered load for the unidirectional ring.

By adding a counter-rotating ring, higher throughputs and lower latencies can be achieved because all links between nodes are now bidirectional. A packet sent on such a network can take the shortest path to its destination. However, the SCI standard requires that the corresponding echo packet must continue its flow on the same ring as the data packet it acknowledges, thereby taking a potentially longer path back to the source node. To determine an upper bound on the throughput of a dual counter-rotating ring network, the previous peak effective throughput in Eq. 1 can be modified as follows:

$$Peak\ Effective\ Throughput \le \frac{(N\ nodes)\times(64\ bytes)\times(2\ packets)}{\left(41\times\left(\frac{N}{4}\right)+5\times\left(N-\frac{N}{4}\right)symbols\right)\times(2\ ns/symbol)} = 4.57\ GB/s \qquad (2)$$

This formulation results in a peak throughput of 4.57 GB/s. With a dual counter-rotating ring network, each node can transmit two 64-byte data packets at the same time – one on each ring. The numerator is therefore modified to allow twice the number of packets transmitted simultaneously. The denominator reflects the fact that request packets taking the shortest path will traverse $N/4$ links on average for a uniform distribution. Again, the peak throughput is independent of ring size. However, the value of the peak throughput is less than the ideal of 4 times that of a unidirectional ring. This drop in the peak performance is attributed to the longer distance that an *echo* packet must travel to return to the originating node. Since nodes cannot transmit to themselves, the average number of links a packet must traverse is actually $N^2/[4(N-1)]$ for even N, or $(N+1)/4$ for odd N, but both expressions approach $N/4$ as N increases. Constructed by substituting these terms into Eq. 2, Table 2 shows the peak throughputs produced by the analytical model.

Table 2
Peak effective throughput versus system size for the counter-rotating ring networks

System	Analytical Peak Throughput (GB/s)	Simulated Peak Throughput (GB/s)	Ratio (Sim./Anal.)
4-node	3.76	3.45	92%
6-node	4.05	3.64	90%
8-node	4.19	3.68	88%
10-node	4.27	3.61	85%

Fig. 6 shows the effective throughputs obtained from the simulation experiments for the 4-, 6-, 8-, and 10-node system models with counter-rotating ring networks, and their results are also included for comparison purposes in Table 2. The larger topologies saturate at approximately 3.6 GB/s, and an average difference of about 11% is indicated between the results from simulation versus analytical estimation. The primary cause for this gap is the lack of queuing delay in the analytical model. For single-ring networks, packets arrive consecutively into each node and are processed accordingly. However, with counter-rotating ring networks, packets may arrive into each node simultaneously from two distinct links. Thus, the queuing delay into the single receiving queue for FCFS processing at the destination node is included in the models and thus reflected in the simulation results to more accurately represent the real system.

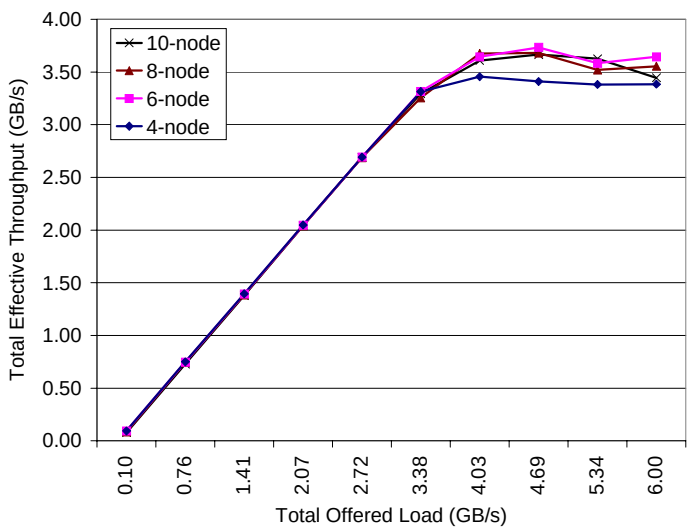

Fig. 6. Throughput versus offered load for the counter-rotating ring network.

Table 3 compares the simulation results on average latency for the unidirectional and counter-rotating ring networks at light loads. The measurements at light loads were taken using an offered load of 0.6 GB/s, which is far below the saturation point for either topology. As expected, the latencies for the counter-rotating ring networks are smaller than single ring networks of equal size.

Table 3
One-way latencies for the unidirectional and counter-rotating ring networks under light loading

System	Single Ring Network Latency (μs)	Counter-Rotating Ring Network Latency (μs)	Ratio (CRR/SR)
4-node	0.176	0.132	75%
6-node	0.225	0.146	65%
8-node	0.282	0.164	58%
10-node	0.344	0.184	53%

The counter-rotating ring systems provide approximately 2.7 times higher throughput and 25-47% lower latency than the single-ring systems. However, a limitation with both of the multiprocessor networks presented above is that the total effective throughput is fixed and does not scale as a function of the number of nodes. Moreover, the latencies continue to increase as the number of nodes increases. In order to achieve higher throughputs and lower latencies, other topologies using multiple rings in two dimensions are considered next.

4.2 Two-dimensional topologies (k-ary 2-cubes)

In order to judge the performance of SCI under a more scalable, two-dimensional topology, models for networks of various sizes were constructed featuring unidirectional and bidirectional tori. Fig. 7 shows a 3×3 unidirectional 2D torus and a 3×3 bidirectional 2D torus. In general, 2D tori will scale better than simple, 1D rings since adding more nodes to a torus also involves adding more rings and thereby more bisection bandwidth. The torus structure allows for a more even distribution of the load and should provide lower latencies and higher throughputs.

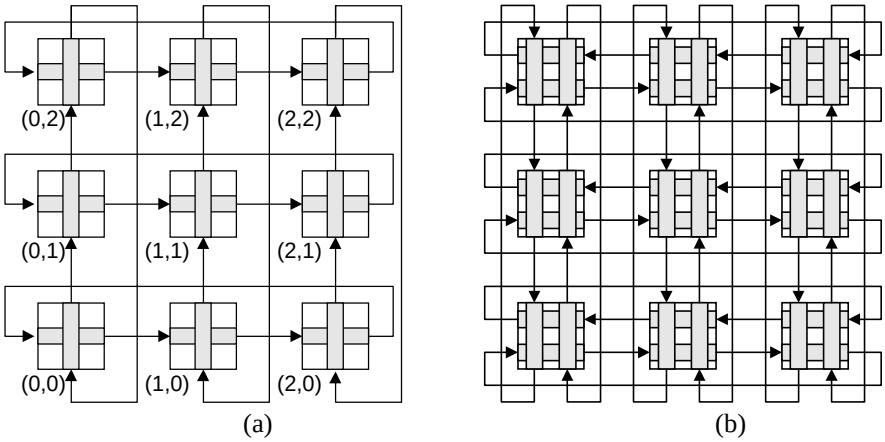

Fig. 7. 3×3 tori configurations, including the (a) unidirectional torus and (b) bidirectional torus.

In order to formulate an analytical expression for the maximum throughput in a unidirectional 2D torus, consider a $\sqrt{N} \times \sqrt{N}$ torus with N nodes. The same approach used previously to formulate analytical models for SCI rings can be applied if the average number of hops for a uniform distribution is known. To find this average distance, the sum of the distances from an arbitrary starting node to each of the other $N-1$ nodes must be found. If the starting node is chosen to be $(0,0)$ at the lower-left node in a system such as that shown in Fig. 7(a), then it can be shown that the distance to any node (i,j) is $i+j$. Therefore, the average distance to any of the $N-1$ nodes is:

$$Average\ Distance = \frac{\sum_{i=0}^{\sqrt{N}-1}\left[\sum_{j=0}^{\sqrt{N}-1}(i+j)\right]}{N-1} = \frac{N}{\sqrt{N}+1} \qquad (3)$$

For the *echo* packets to travel from node *(i,j)* back to node *(0,0)* on a unidirectional torus requires *(k-i)+(k-j)* hops. Substituting this expression into the equation above yields exactly the same result. Using this equation for average distance and taking a similar approach as previously used with the counter-rotating ring networks, the peak throughput for a unidirectional 2D torus is found to be:

$$Peak\ Effective\ Throughput = \frac{(N\ nodes)\times(64\ bytes)\times(2\ packets)}{\left(41\times\left(\frac{N}{\sqrt{N}+1}\right)+5\times\left(\frac{N}{\sqrt{N}+1}\right)symbols\right)\times(2\ ns/symbol)} = 1.39\times(\sqrt{N}+1)\ GB/s \qquad (4)$$

Unlike the 1D ring cases, the peak throughput in a unidirectional 2D torus scales with $N^{1/2}$. As the number of nodes in the torus increases, so does the total available bandwidth. Table 4 shows analytical best-case peak throughputs for various sized tori. The throughput results from the simulation experiments of the unidirectional 2D torus are shown in Fig. 8 and compared to the analytical results in the table.

Table 4
Peak effective throughput versus system size for the unidirectional torus

System	Analytical Peak Throughput (GB/s)	Simulated Peak Throughput (GB/s)	Ratio (Sim./Anal.)
9-node (3x3)	5.56	5.10	92%
16-node (4x4)	6.95	6.21	89%
25-node (5x5)	8.34	7.54	90%
36-node (6x6)	9.73	8.67	89%

For the unidirectional tori, the simulated peak throughputs are consistently lower than the analytically predicted values but follow an approximately parallel track as the number of nodes is increased. Owing to the better accuracy in the simulation models, one reason the simulated throughputs are approximately 10% smaller is that there is a possibility for busy-retries to occur in a 2D torus that are not accounted for in the analytical model. Busy-retries can occur when multiple packets on a ring attempt to switch through the same switching node to a second ring. The input queues of the switching node may become full while the node waits for available bandwidth on the second ring. The resulting *busy-retry* packets on the first ring represent overhead and thus wasted bandwidth, and reduce the throughput from the theoretical maximum.

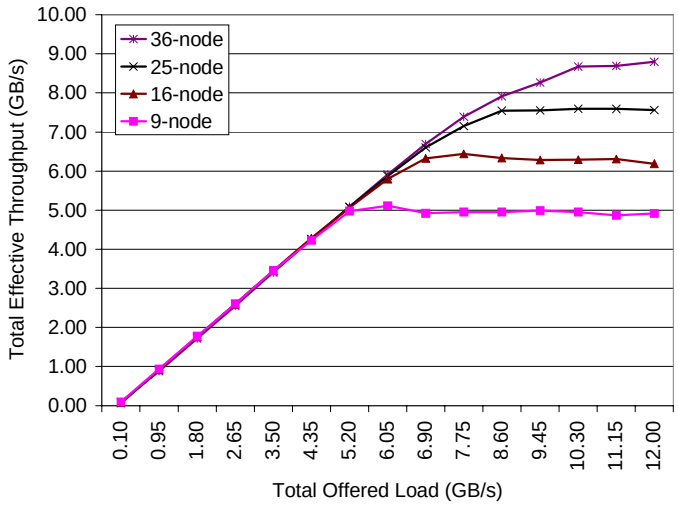

Fig. 8. Throughput versus offered load for the unidirectional torus.

A comparison of the latency results from the simulation experiments with light loads on the counter-rotating ring and unidirectional tori networks are shown in Table 5. The latencies are measured at an offered load of 0.6GB/s for both topologies. Even though the diameters of the tori are smaller than the corresponding diameters of the counter-rotating ring topologies, the tori exhibit a higher latency. This increase of approximately 10-20% in latency is attributed to the additional ring-to-ring switching delay not encountered by packets in the counter-rotating ring topology.

Table 5
One-way latencies for the dual counter-rotating ring and unidirectional torus networks under light loading

System	Counter-Rotating Ring Network Latency (μs)	Unidirectional Torus Network Latency (μs)	Ratio (UT/CRR)
8-node	0.164	–	–
9-node	0.170	0.206	121%
10-node	0.184	–	–
16-node	0.221	0.241	109%

By adding counter-rotating rings to the unidirectional torus structure, a bidirectional 2D torus is created. This topology is illustrated in Fig. 7(b). Each node must have four SCI ports to achieve such a structure. Fig. 9 shows the throughput results from the simulation experiments run on the 9-, 16-, 25-, and 36-node system models with bidirectional tori.

As an analytical estimate for the bidirectional tori, the throughput might simply be expected to be approximately four times that of the unidirectional torus due to twice the number of output ports at each node and half the average distance that each packet must now travel. Again, this supposition is not the case as evidenced by the simulation results for the bidirectional tori in Fig. 9. The difference is attributed to several factors. First, *echo* packets are again required to take the non-optimal path by continuing their flow on the same ring used by their corresponding request packets. Second, increased contention occurs within each switch in the bidirectional torus causing additional busy-retries packets to be generated, since every switching node now has two input ports that may contend for access to

the same output port. Finally, the problem is exacerbated by the fact that, like *echo* packets, each *busy-retry* packet must continue on the same ring as its request packet, thereby taking a non-optimal path back to the source node.

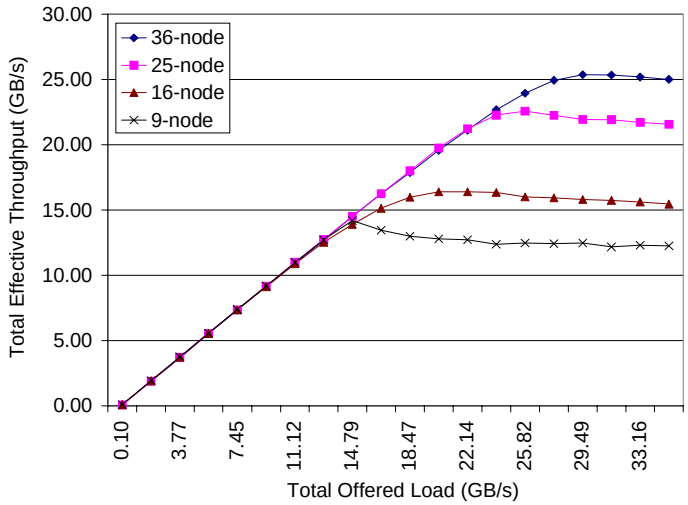

Fig. 9. Throughput versus offered load for the bidirectional torus.

Table 6 compares the latencies of the unidirectional tori to those of the bidirectional tori for light loads. The bidirectional tori are found to be 18-29% lower in latency than with their unidirectional tori counterparts due to the decrease in the average distance between nodes. A full 50% reduction in latency cannot be achieved for the same reasons cited above.

Table 6
One-way latencies for the unidirectional and bidirectional torus networks under light loading

System	Unidirectional Torus Network Latency (μs)	Bidirectional Torus Network Latency (μs)	Ratio (BT/UT)
9-node	0.206	0.162	79%
16-node	0.241	0.197	82%
25-node	0.293	0.212	72%
36-node	0.312	0.222	71%

4.3 Uniform-ring torus topologies

As a final case in the study of 2D distributed switching fabrics for SCI multiprocessor networks, a series of system models was developed and simulated using the uniform-ring topology. As illustrated in Fig. 10 where N = 3x3, each node in a uniform-ring torus possesses exactly four network interfaces.

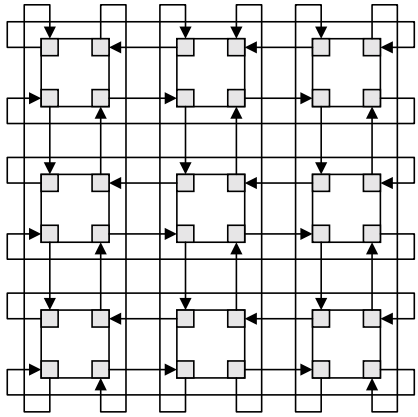

Fig. 10. A 3x3 uniform-ring torus.

In this example, each node connects to four different rings with a fixed size of four nodes per ring. This topology maintains a constant ring size even as the number of nodes continues to increase. By contrast, all conventional tori with N nodes have a ring size of $N^{1/2}$. A summary of these characteristics is provided in Table 7.

Table 7
Summary of 2D torus network characteristics

Network	Total Number of Rings	Number of Nodes per Ring
Unidirectional torus	$2\times\sqrt{N}$	$\sqrt{N}$
Bidirectional torus	$4\times\sqrt{N}$	$\sqrt{N}$
Uniform-ring torus	N	4

One of the advantages of the uniform-ring torus is that its topology is inherently more tolerant of faults. As the number of nodes N increases, the number of nodes per ring stays constant while the total number of rings in the network scales up more quickly than in conventional torus networks. Since a faulty link renders an entire ring unusable, it follows that the smaller the size of the ring the fewer the number of nodes affected by the fault. The price of such fault tolerance is a potential reduction in performance due to the additional switching time required to reach the final destination. Such is the case since fewer nodes per ring implies that the average packet will experience a higher diameter for all but the smallest tori, and thus more ring-to-ring switching in reaching its destination than a comparable packet would experience on a conventional torus. Of course, for some applications, the fault-tolerant aspects of the network may mitigate the concern over any potential losses in performance. For instance, fault tolerance in the SCI network becomes an important factor in high-dependability applications such as those presented by Hudgins and Schroeder [15] and by Perkins et al. [16]. The extent of the impact of the potential performance degradation depends on the nature of the network protocol and switch architectures used, and this impact is explored next in terms of a uniform-ring torus based on SCI.

Fig. 11 shows the effective throughputs obtained from simulation experiments with the 9-, 16-, 25-, and 36-node system models with uniform-ring tori, and compares them to the throughputs obtained from the bidirectional tori of the same size. For the smaller systems the impact of ring count and switching delay is of minor import. For the 9-

node systems, the bidirectional torus has less nodes per ring (i.e. 3 versus 4) and more total rings (i.e. 12 versus 9), and their performance is comparable. The same is true for the 16-node systems, where the total number of rings is equal for the two topologies (i.e. 16 in both cases) as is the number of nodes per ring (i.e. 4 in both cases). As the system size grows larger, the total number of rings provided by the uniform-ring torus scales faster than that with the bidirectional torus, but by contrast the number of nodes per ring in the bidirectional torus continues to scale with $N^{1/2}$ while the uniform-ring holds constant. For example, in the case of the 36-node systems, the bidirectional torus is comprised of 24 rings with 6 nodes per ring while the uniform-ring torus is comprised of 36 rings with 4 nodes per ring. As the results in Fig. 11 demonstrate, when constructing such fabrics for SCI-based multiprocessor networks, the added aggregate bandwidth provided by the additional rings in the uniform-ring torus is approximately balanced by the switching bottlenecks brought on by fewer nodes per ring.

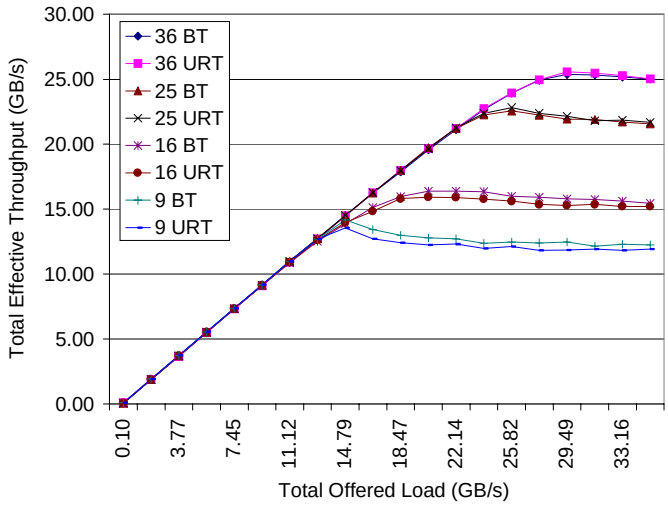

Fig. 11. Throughput versus offered load for the bidirectional and uniform-ring tori.

Table 8 compares the latency results from the simulation experiments at light loads with both topologies. As the system size increases, the latencies with the uniform-ring torus increase more rapidly due to the additional switching required to reach the destination. Due to its small size and for the reasons cited above, the 9-node case is the exception in that the uniform-ring network's latency is slightly lower than that of the bidirectional torus. However, as the system grows to 16 nodes and more, the fixed number of rings in the uniform-ring torus dictates more hops from source to destination resulting in a degradation in latency performance. However, this degradation is mitigated to some degree with the use of fast switches for SCI, but the two latency curves diverge as N increases.

Table 8
One-way latencies for the bidirectional and uniform-ring torus networks under light loading

System	Bidirectional Torus Network Latency (μs)	Uniform-Ring Torus Network Latency (μs)	Ratio (URT/BT)
9-node	0.162	0.142	88%
16-node	0.197	0.202	103%
25-node	0.212	0.233	110%
36-node	0.222	0.277	125%

Thus, the results indicate that these two 2D torus topologies for SCI-based multiprocessor networks achieve comparable total effective throughput. The additional rings incorporated in the uniform-ring topology, and the extra switching day between them, add to the average latency experienced by packets. However, the evidence suggests that the increase in latency scales at a moderately low rate with SCI-based fabrics.

5. Conclusions

In this paper, the performance of several promising SCI topologies for distributed multiprocessor networks was examined through high-fidelity, CAD-based simulation with analytical verification. One-dimensional SCI networks were the first cases studied, since they form the basis for all SCI networks. Two-dimensional tori were explored next, culminating in a case study on uniform-ring SCI networks for high-performance, fault-tolerant multiprocessor networks.

It was shown both analytically and through simulation that the aggregate throughput of single- and dual-ring SCI systems is independent of the network size. From a theoretical standpoint, the total effective throughput of a multiprocessor system based on a single SCI ring is bounded above at 1.39 GB/s, with a more practical limit of approximately 1.35 GB/s as determined through detailed simulations. By contrast, systems constructed from dual, counter-rotating SCI rings are bounded above at 4.57 GB/s, with a practical limit of approximately 3.6 GB/s. As such, the throughput of an dual-ring SCI network was found to be approximately 2.7 times higher than the single ring, and the latency was 25-47% lower for systems up to 10 nodes with the evidence suggesting a latency improvement of over 50% for larger systems.

SCI networks can be made scalable by increasing the number of rings in the topology and adding an additional dimension to the single and dual-ring systems. The two-dimensional topologies (i.e. tori) demonstrated a more scalable throughput that increases as a function of the system size. In the unidirectional tori, analytical results indicate that the throughputs scale according to $1.39 \times (\sqrt{N} - 1)$ GB/s and simulation results found a practical limit approximately 10% lower, where N represents the number of nodes in the system. However, the latencies were actually 10-20% higher for the unidirectional SCI tori versus counter-rotating rings, due to the ring-to-ring switching delay required in a torus fabric.

The throughputs of the bidirectional torus topologies were found to be between 2.5 to 3 times higher than the throughputs of their unidirectional counterparts. Ideally, the throughputs of these systems should be approximately four times the throughputs of the unidirectional ones. However, the increase in the number of *busy-retry* packets brought on by increased contention in the distributed switches, coupled with the longer distances traveled by both the *echo* and *busy-retry* packets, reduce the total effective throughput that can be obtained. The bidirectional tori were found to have 18-29% lower latency for systems up to 36 nodes, with the gap widening for larger systems. The evidence suggests that these latencies will approach but not attain the 50% reduction indicated by the theoretical minimum.

Finally, through the use of several simulation experiments, it was determined that despite an increase in ring-to-ring switching delays brought on by a higher average number of hops from source to destination, the uniform-ring

networks nevertheless achieve throughputs comparable to their bidirectional torus counterparts. The evidence indicates that the additional aggregate bandwidth inherent to the uniform-ring topology balances the additional switching delays that are imposed. However, while the uniform-ring topology does provide more inherent capability for fault tolerance, the results indicate a latency increase of up to 25% for systems up to 36 nodes, and even more for larger systems.

The results presented in this paper represent the first high-fidelity simulations of SCI multiprocessor networks with *k*-ary *n*-cube topologies for investigations into the performance scalability of SCI in terms of throughput and latency versus topology size and dimension. The models and simulations provided in this paper were intended to represent a distributed shared-memory multiprocessor constructed from computers connected via an SCI system-area network. However, the results could also be applied on a lower level where the individual distributed switching nodes are each attached to a processor, memory module, or I/O controller.

Several activities are anticipated for future research in this area. For instance, the SCI networking results presented herein can be extended with application-oriented results to study the performance levels provided to particular distributed parallel algorithms and applications. These activities can be pursued in terms of trace-driven simulations, where access patterns are sampled from real applications and the traces drive or stimulate the models, or with execution-driven simulations where real applications execute on virtual prototypes via simulation. In addition to a broader study from an application level, the study of three-dimensional SCI fabrics and beyond is also anticipated. Finally, whereas this paper focuses on performance attributes, future work will include the study of dependability attributes with the development and analysis of fault-tolerance mechanisms in and for SCI through simulation.

Acknowledgements

This work was sponsored in part by the National Security Agency. The authors also acknowledge contributions by Matthew Chidester, Robert Todd, and William Phipps in our lab for their suggestions and assistance in the development of our high-fidelity CAD model for SCI.

References

[1] IEEE, SCI: Scalable Coherent Interface, *IEEE Approved Standard IEEE 1596-1992*, 1993.

[2] T. D. Lovett, R. M. Clapp, R. J. Safranek, NUMA-Q: An SCI-Based Enterprise Server, *White Paper, Sequent Computer Systems, Inc*. 1996.

[3] S. Scott, The SCX channel: A new, supercomputer-class system interconnect, *Proceedings of the 7th International SCI Workshop*, Santa Clara, California, August 1995.

[4] S. Scott, The GigaRing Channel, *IEEE Micro*, vol. 16, no. 1, pp. 27-34, February 1996.

[5] R. Clark, SCI Interconnect Chipset and Adapter: Building Large Scale Enterprise Servers with Pentium Pro SHV Nodes, *White Paper, Data General Corporation*, 1996.

[6] W. J. Dally, Performance Analysis of *k*-ary *n*-cube Interconnection Network, *IEEE Transactions on Computers*, vol. 29, no. 6, pp. 775-785, June 1990.

[7] T. Y. Chung, Cost-performance Trade-Offs in Manhattan Street Network versus 2-D Torus, *IEEE Transactions on Computers*, vol. 43, no.2, pp. 240-243, February 1994.

[8] D. A. Reed, D. C. Grunwald, The Performance of Multicomputer Interconnection Networks, *IEEE Computer*, pp. 63-73, June 1987.

[9] B. Wu, SCI Switches, *Proceedings of the International Data Acquisition Conference on Event Building and Event Data Readout in Medium and High Energy Physics Experiments*, Fermilab in Batavia, Illinois, October 1994.

[10] B. Wu, A. Bogaerts, B. Skaali, A Study of Switch Models for the Scalable Coherent Interface, *Proceedings of the Sixth Conference on Data Communication Systems and their Performance*, Istanbul, Turkey, October 1995.

[11] B. Wu, A. Bogaerts, R. Divia, E.H. Kristiansen, H. Muller, B. Skaali, Several Details of SCI Switch Models, *University of Oslo/CERN, internal report*, November 1993.

[12] R. Divia, SwitchLink V1.0, *CERN DRDC RD24 Project, Internal note*, November 1992.

[13] A. George, R. Fogarty, J. Markwell, and M. Miars, An Integrated Simulation Environment for Parallel and Distributed System Prototyping, *Simulation*, vol. 75, no. 5, pp. 283-294, May 1999.

[14] K. Shanmugan, V.S. Frost, and W. LaRue, A Block-Oriented Network Simulator (BONeS), *Simulation*, vol. 58, no. 2, pp. 83-94, February 1992.

[15] C. E. Hudgins, J. E. Schroeder, Applying Commercial Real-time Data Networks to Future Military Avionics, *Microprocessors and Microsystems*, vol. 21, pp. 21-28, 1997.

[16] A. E. Perkins, A. D. Birch, R. C. W. Davies, Simulation of Future System Data Networks, *Microprocessors and Microsystems*, vol. 20, pp. 485-494, 1997.