

# Simulation Experiments of a High-Performance RapidIO-based Processing Architecture

J. Adams,<sup>1</sup> C. Katsinis,<sup>2</sup> W. Rosen,<sup>2</sup> D. Hecht,<sup>2</sup> V. Adams,<sup>2</sup> H.V. Narravula,<sup>2</sup>  
S. Sukhtankar,<sup>2</sup> and R. Lachenmaier<sup>3</sup>

**Abstract:** *In this paper we describe the results of our simulation analysis of a high-performance processing architecture based on the RapidIO network protocol. RapidIO is a 10 Gb/s, low-latency packet-switched interconnect technology designed for processor-to-processor, processor-to-memory, and processor-to-peripheral interconnects. Two network topologies were simulated, a simple network consisting of an 8-port switch and 8 processing nodes and a more extensive network consisting of five 8-port switches and 24 processing nodes. Results indicate that latencies as low as 92 ns for a remote 64-bit read request/response transaction may be achieved in an unloaded single-switch system. The effectiveness of various flow control mechanisms provided by the protocol are also explored, and when used in combination a 10% increase in link utilization is achieved.*

## 1 Introduction

Recently a number of high-speed, low-latency networking standards have emerged that are intended for high-performance parallel processing applications based on commodity networks. These standards include RapidIO [1], HyperTransport, Jbus [4], 10 Gigabit Ethernet [2], InfiniBand [5], and Myrinet [6]. RapidIO is an attractive choice for these applications for several reasons. Among these are its high data rates (8–16 Gb/s), potential for extremely low latencies, flat address space, efficient cache coherency mechanism between clusters of processors, support for all needed micro-processor and I/O transactions, high level of fault tolerance, and commercial support. In addition, RapidIO is transparent to existing applications

and operating system software and is designed to support scalability and future enhancements and extensions.

Latency is a critical parameter in the performance of any parallel processing machine and the actual protocol employed by the interconnection network is a key determinant of network latency. We have carried out a number of simulation experiments that accurately model the RapidIO protocol to evaluate the latency of a RapidIO-based system. In this paper we present the results of our simulations as well as a brief review of the RapidIO protocol. Two networks were evaluated, the first a simple 8-node system with a single 8-port switch employing cut-through routing and the second a multi-stage network with five 8-port switches and 24 nodes. Latency was determined over a wide range of operating conditions from very light loading through saturation. The results indicate that latencies as low as 92 ns for a remote 64-bit read request/response transaction may be achieved in an unloaded single-switch system. The effectiveness of several implementation-dependent flow control mechanisms that are supported by the protocol are also evaluated.

## 2 Network Protocol

RapidIO is a high-speed, packet-switched, full-duplex architecture designed to be a chip-to-chip and board-to-board interconnect solution for passing data and control information. The architecture was designed for tightly coupled processor-to-processor, processor-to-memory, and processor-to-peripheral interconnects where low latency and high bandwidth are required. The RapidIO protocol features an efficient header and variable payload sizes up to 256 bytes. The address space is flat and supports up to 65,000 nodes. The architecture supports all needed microprocessor and I/O transactions and is transparent to existing applications and operating system

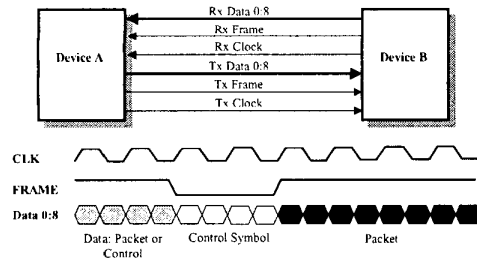
<sup>1</sup> Rydal Research and Development, Inc., Rydal, PA 19046

<sup>2</sup> ECE Department, Drexel University, Philadelphia, PA 19104

<sup>3</sup> Naval Air Warfare Center, China Lake, CA 93555

software. The RapidIO architecture is partitioned into a three-layer hierarchy (logical, transport, and physical layers) that provides a feature set capable of supporting the needs of both processors and peripherals and is transparent to software. These features include the ability to support all non-coherent memory operations, atomic operations, unaligned memory transfers, Global Shared Memory (GSM), and message passing. Of these layers the physical layer is most important for the purpose of simulation and analysis of latency and throughput.

The physical layer transmits, receives, tracks/manages packet flow, and handles transmission errors on the links of the interconnect fabric. These links are comprised of either 8-bit or 16-bit parallel asynchronous interfaces accompanied by clocking signals and framing signals that signifies the start of either a control symbol or packet on the link, and allows for the use of embedded control symbols within packet transmissions. An example of a typical link and its operation can be seen in Figure 1, and is set to operate at transmission rates from 500 MB/s to a potential 8 GB/s per link direction.



**Figure 1: Typical RapidIO link**

The physical layer does not permit packets to be dropped by the fabric and, through the use of strict ordering rules and priorities, insures that packets are delivered in order within flows and that end-to-end deadlocks are avoided. In the event that a packet or control symbol is received in error, and the condition is not fatal, the physical layer specification defines a set of hardware protocol recovery mechanisms that allow the fabric to recover and reestablish communications without packet loss or the need for software intervention.

Flow Control of data through a RapidIO fabric is handled by three mechanisms, backpressure through the use of acknowledgment and

retry control symbols, throttling through the use of a user-defined number of control symbols between packets, and buffer status tracking of the receiving device by means of various control symbols returned to the sending device. The uses of these mechanisms are implementation dependent and do not prohibit the use of worming techniques to move a packet through the fabric.

The protocol allows control packets to be inserted at any time during the transmission of a data packet.

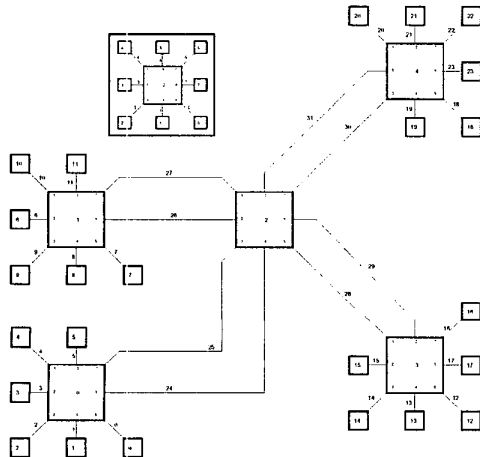
### 3 Simulation

To explore the associated latencies and throughput of a RapidIO fabric we have developed an extensive RapidIO simulator that models all major components within a system (processors, links, and switches) and all internal components of a switch (input/output ports and associated queues and control logic). For simplicity the link length between components is determined by the number of clock cycles required to transmit a single datum across the link, and a clock cycle is the time it takes to transmit a single datum. The size of a datum is dependent on the port size, where in the case of the 8-bit interface the datum would be a byte, or in the case of the 16-bit interface the datum would be a half-word. The simulation models the complexities of a RapidIO switch by implementing the examination of packet headers, determining priority and routing, and queuing packets to the appropriate output port. If resources are available within the switch packets are forwarded from input to output port in a cut-through fashion. Data transmissions on the links conform to the protocol and implement the use of embedded symbols when required.

Flow control on the individual links within the simulation is implemented in an aggressive manner to maximize bandwidth utilization on the links. Specifically backpressure is utilized such that a transmitting device will send packets until all available link packet-Ids have been utilized or until the estimated buffer space of the receiving device equals zero. The receiving device in this process only embeds an acknowledge symbol into the outgoing data stream upon the correct reception of a packet. If the receiving device's input buffer is close to full the device will then embed an idle symbol into the outgoing packet stream when an input buffer space becomes free. Simulation results show that

this process yields a performance increase of 10% in terms of link utilization when compared to the flow control mechanisms individually.

The two performance measurements explored in these simulations are link utilization and end-to-end packet latency. The end-to-end packet latency is defined as the total time elapsed from packet creation, queuing time at the source node, and time for the network to complete packet delivery at the destination node. The interarrival time between packet creation is exponentially distributed, and the packet size is uniformly distributed with mean equal to 112 bytes. The destination of each packet is also uniformly determined from among all other nodes although, alternatively, a certain form of locality can be forced by specifying in the simulation the actual distribution of packets based on the distance between the source and the destination.



**Figure 2: Simulation Networks**

Two networks were simulated, one a simple cluster consisting of one 8-port switch and 8 processing nodes and a more extensive network consisting of five 8-port switches and 24 processing nodes (Figure 2). All links within the two networks only utilize the 8-bit RapidIO interface, and in the second, more extensive network, two links may be used in parallel to alleviate the bottleneck effect of the central switch. The clock rate of the simulation is set to represent a 500 MHz RapidIO clock (clocked on both edges), so that one clock cycle in the simulation equals 1ns.

## 4 Results

For the simple 8-node network, figure 3 shows the end-to-end latency observed as the link utilization varies. Because some of the link bandwidth is used by control symbols, this network saturates at about 95% link utilization. When no congestion occurs, the head of a packet traverses the switch in 17 clock cycles, resulting in a lower limit of approximately 140 clock cycles for the expected end-to-end latency (of packets with mean size of 112 bytes) as also shown in Figure 2. Consequently, a typical 64-bit read request/response transaction, which results in the equivalent of two 20-byte messages, will incur a latency of 90 clock cycles when no congestion occurs in the switch. This result is consistent with analytical estimates of 80 ns for a single remote 64-bit read [3].

Figure 3 also shows the average end-to-end latency for the more extensive 24-node network, for four different types of operations. These operations include the use of either one or both of the parallel links connecting the switches and whether or not locality was implemented in determining packet destination. If locality was used the process was such that 70% of all packets passed through only one switch, and the remaining 30% passed through 3 switches to arrive at their destination. For the case when packet destinations are selected uniformly over all nodes and only one of the parallel links is utilized between the switches, the large traffic through the center switch causes early saturation and average link utilization remains below 33% (saturation curve marked 1L in figure 3). For the case when both links are utilized and packet generation destination remains uniform, the average link utilization increases to 66% (saturation curve marked 2L in figure 3). For the cases when locality is utilized (marked LOC in figure 3), one can see that traffic on the link fabric is more equalized and the performance improves so much that the two-link case (marked 2L LOC in figure 3) approaches the performance of the simpler 8-node network.

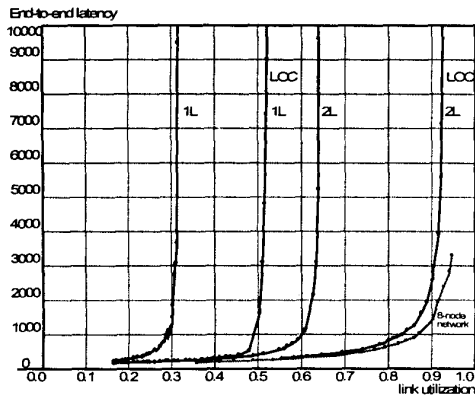


Figure 3: Latency Vs. Utilization

- 1L: 24-node network utilizing 1 link between switches and no locality
- 1L LOC: 24-node network utilizing 1 link between switches with locality
- 2L: 24-node network utilizing 2 links between switches and no locality
- 2L LOC: 24-node network utilizing 2 links between switches with locality

## 5 Conclusions

In this paper we presented the results of our simulation analysis of a high-performance processing architecture based on the RapidIO network protocol. Two networks were evaluated, the first a simple 8-node system with a single 8-port switch employing cut-through routing and the second a multi-stage network with five 8-port switches and 24 nodes. Latency was determined over a wide range of operating conditions from very light loading through saturation. The 8-node network saturates at about 95% of link utilization network due to the fact that some of the link bandwidth is used by control symbols. Latencies as low as 92 ns for a remote 64-bit read request/response transaction were achieved in an unloaded single-switch system. For the 24-node multistage network

average link utilization remained below 33%, limited by early saturation of the center switch. This increased to 66% when the links between the central switch and each of the other switches were doubled. For the cases when locality is utilized traffic on the link fabric is more equalized and the performance improved so much that the two-link case approached the performance of the simpler 8-node network. The simulations also indicated that performance could be improved by as much as 10% by careful implementation of the flow control mechanisms supported by RapidIO.

## 6 Acknowledgements

This work was supported by the U.S. Navy under contract N68936-00-C-0064 to Rydal Research and Development, Inc., and through subcontract 204050 to Drexel University.

## 7 References

1. RapidIO Interconnect Specification Rev. 1.1, RapidIO Trade Association, March 2001.
2. <http://www.manta.iccc.org/groups/802/3/ac/>
3. D. Bouvier, "RapidIO Technical Overview", presented at the RapidIO European Symposium, April 18, 2001, Sophia Antipolis, France.
4. R. Rama, "Sun Microsystems' JBUS: A Bus Architecture for Embedded On-chip Multiprocessing", presented at the Embedded Processor Forum, June 12, 2001, San Jose California.
5. InfiniBand Architecture Specification Release 1.0, InfiniBand Trade Association, October 24, 2000
6. Boden, N. J., et al., Myrinet: A gigabit per second local area network, IEEE Micro (1995), 29-36.