Reconfigurable and adaptive photonic networks for high-performance computing systems

Avinash Kodi[1,*] and Ahmed Louri[2]

[1]Department of Electrical Engineering and Computer Science, Ohio University, 322D Stocker Center, Athens, Ohio 45701, USA

[2]Department of Electrical and Computer Engineering, University of Arizona, 1230 East Speedway Boulevard, Tucson, Arizona 85719, USA

*Corresponding author: kodi@ohio.edu

Received 2 December 2008; revised 17 April 2009; accepted 26 April 2009; posted 13 May 2009 (Doc. ID 104803); published 10 June 2009

As feature sizes decrease to the submicrometer regime and clock rates increase to the multigigahertz range, the limited bandwidth at higher bit rates and longer communication distances in electrical interconnects will create a major bandwidth imbalance in future high-performance computing (HPC) systems. We explore the application of an optoelectronic interconnect for the design of flexible, high-bandwidth, reconfigurable and adaptive interconnection architectures for chip-to-chip and board-to-board HPC systems. Reconfigurability is realized by interconnecting arrays of optical transmitters, and adaptivity is implemented by a dynamic bandwidth reallocation (DBR) technique that balances the load on each communication channel. We evaluate a DBR technique, the lockstep (LS) protocol, that monitors traffic intensities, reallocates bandwidth, and adapts to changes in communication patterns. We incorporate this DBR technique into a detailed discrete-event network simulator to evaluate the performance for uniform, nonuniform, and permutation communication patterns. Simulation results indicate that, without reconfiguration techniques being applied, optical based system architecture shows better performance than electrical interconnects for uniform and nonuniform patterns; with reconfiguration techniques being applied, the dynamically reconfigurable optoelectronic interconnect provides much better performance for all communication patterns. Based on the performance study, the reconfigured architecture shows 30%–50% increased throughput and 50%–75% reduced network latency compared with HPC electrical networks.

OCIS codes: 200.0200, 200.4650.

1. Introduction

The relentless quest for processing speeds in the range of teraflops and beyond has accelerated the need for scalable, parallel, high-performance computing (HPC) systems [1]. For these systems to be scalable and attain the desirable performance, the interconnection network that connects the processors must itself be scalable in both size and bandwidth. It is widely accepted that Moore's law [and the more recent International Technology Roadmap for Semiconductors (ITRS)] growth rate in available transistors will continue for at least the next decade, thereby strengthening further growth of HPC systems. Near-term projections call for HPC systems with computing power in the hundreds of teraflops, off-chip bandwidth in the range of 4–20 Tbits/s and aggregate interprocessor communication bandwidth or network bandwidth around 40 Tbits/s [2–4]. However, with the ITRS projects, although per-chip performance continues to improve at a rate of approximately four times, the total off-chip input/output (I/O) bandwidth (pin count times the bit rate per pin) will increase by approximately 2.7 times [4]. As clock rates increase to the multigigahertz range, and this difference in improvement rates continue, electrical signaling and interconnect problems, such

0003-6935/09/220E13-11$15.00/0

as skin effect, cross talk, electromagnetic interference (EMI), dielectric imperfections, attenuation, clock skew, and power dissipation are predicted to become the ultimate bottlenecks for HPC systems both at the chip-to-chip and board-to-board levels [2,3,5–7]. The major bottleneck is the limited bandwidth at higher bit rates and longer communication distances. This electrical technology limitation, if not dealt with, will create major bandwidth imbalances in future HPC systems and will significantly affect their performance and scalability.

The limited bandwidth and connectivity of electrical interconnects can have negative effects on key performance measures of HPC systems, which includes execution time or processor latency, processor utilization, and network latency. The limited bandwidth causes the processor to stall for a longer time, while waiting for the required data, and consequently lower its utilization. For the network, the limited connectivity and bandwidth results in longer queueing and routing delays throughout the network switches because of extensive multiplexing of numerous signals onto limited serial I/O links [8]. Moreover, important communication functions such as broadcasting and multicasting (required for synchronization and cache coherence protocols) could lead to highly contended and concentrated access to shared data for a short duration [5,9]. During this duration, the network bandwidth will be further reduced since only a fraction of the effective bandwidth could be utilized. Clearly, the combined effect of technological and architectural problems will create a major performance bottleneck and could become a fundamental impediment to future scalable HPC systems.

A. Optical Interconnects for High-Performance Computing Systems

Optical interconnects offer several well-known advantages for HPC systems such as higher spatial and temporal bandwidths, lower cross talk independent of data rates, higher interconnect densities, better signal integrity at high frequencies, lower signal attenuation, and lower power requirements at higher bit rates [2,3,10–14], all of which could potentially achieve the much desired high bit rates data communication at a much reduced power level at the board-to-board distances of 0.1–1 m.

We previously proposed a reconfigurable all-photonic interconnect for distributed and parallel systems (RAPID)[15]. RAPID topology maximizes bandwidth availability and lowers network latency. As every node in RAPID requires two sets of transmitters for intraboard and interboard communication, we developed alternate versions of RAPID. In modified (M)-RAPID, the processors have electrical links for intraboard communication and optical links for interboard. In electrical (E)-RAPID, electrical onboard communication is used for both intraboard and interboard (up to optical transmitters and receivers) [16]. This allowed us to reduce the cost of the network without excessive optical signaling. E-RAPID optimizes interconnect cost (based on the number of wavelengths) and incorporates novel features such as dynamically reconfiguring the interconnect. The key features of E-RAPID include:

1. **High Bandwidth:** E-RAPID utilizes the high capacity of optical interconnects by partitioning the huge bandwidth available in optical fibers into multiple nonoverlapping, manageable, high-speed channels through a combination of wavelength-division multiplexing (WDM), space-division multiplexing (SDM), and time-division multiplexing (TDM).
2. **Low Latency:** E-RAPID has a constant node degree and low network diameter enabling lower queueing and routing delays for interprocessor communication.
3. **Reasonable Optical Complexity:** The cost complexity of the proposed optical interconnect is maintained at a reasonable level by a combination of several approaches: (a) Wavelength reuse is adopted throughout the network design. Wavelength reuse achieved at each level of the interconnect magnifies the usefulness of WDM channels and reduces the number of WDM channels required. (b) Optical passive components are used for the design of the transfer medium in E-RAPID without any active switches, thereby reducing power dissipation, minimizing cost, and accelerating communication. (c) E-RAPID design delineates electronic processing units from the optical components and devices, and bridges the gap by using electrical interconnects. This eliminates the need for every processor to have transmitters, thereby allowing groups of processors to share common transmitters and receivers. This decoupling further reduces the cost as there is no dependence between the processing and the communication units.
4. **Scalable Design:** E-RAPID provides several scalable features including the incremental addition of wavelengths, nodes, and boards. Various design choices that enable high bandwidth, low latency, and reasonable cost in E-RAPID have cumulatively resulted in providing maximum flexibility for scaling [16].
5. **Dynamic Reconfiguration:** In HPC applications a high degree of temporal and spatial locality exists between communicating processors. E-RAPID exploits this locality by dynamically reconfiguring the network based on traffic patterns. While static routing and wavelength allocation (RWA) can provide improved performance that is due to increased bandwidth availability, the significant feature of E-RAPID is being able to dynamically reallocate bandwidth to system boards. Reconfigurability is realized by interconnecting arrays of optical transmitters, and adaptivity is implemented by the dynamic bandwidth reallocation (DBR) technique that balances the load on each communication channel. We propose a DBR technique, Lockstep (LS) protocol that monitors traffic intensities, reallocates bandwidth, and adapts to changes in communication patterns. This

dynamic reallocation results in reduced communication bottlenecks and optimized resource utilization leading to balanced–improved system architecture design.

In [17,18] we discussed the ability to reconfigure E-RAPID and dynamically reallocate the bandwidth. The significant contributions of this paper are as follows:

- We extend and enumerate clearly the working of the E-RAPID architecture. We exhaustively explain the onboard electrical network and the offboard optical network. This clarifies and explains the E-RAPID network clearly.
- We exhaustively develop the DBR technique, which includes the proposed technology for reconfiguration of the system, the proposed statistics collection, and dissemination and system resynchronization. This work clarifies and builds over previously developed LS protocol. We also develop an algorithm that explains the working of the proposed reconfiguration succinctly.
- We compare E-RAPID with previously proposed RAPID versions and electrical networks for all traffic traces including Uniform, Bit Reversal, Butterfly, Matrix Transpose, Perfect Shuffle, and Complement traffic. This covers all synthetic traffic traces for interconnection network simulation.

The goal of this paper is to evaluate E-RAPID system architecture that can be reconfigured, that can adapt to shifts in traffic patterns and at the same time deliver scalable bandwidth with low communication latency and reduced power consumption. In Section 2 we discuss E-RAPID architecture and RWA. We explain the reconfiguration of E-RAPID In Section 3. In Section 4 we analyze the performance evaluation of E-RAPID, and we conclude the paper in section 5.

2. E-RAPID: Baseline Optoelectronic Architecture

An E-RAPID network [15] is defined by a 3-tuple (C,B,D), where C is the total number of clusters, B is the total number of boards per cluster, and D is the total number of nodes per board. The total number of nodes in E-RAPID is the multiplicative factor $N = C \times D \times B$. Figure 1 shows the conceptual E-RAPID architecture for a single cluster. In Fig. 1(a), 0 to $D-1$ nodes are connected together to form a board; 0 to $B-1$ boards are connected to form a single cluster. All the nodes are connected to two subnetworks; a scalable (electrical) intraboard interconnection (IBI) and an (optical) scalable remote superhighway (SRS) via passive couplers. We separated intraboard and interboard (remote) communications from one another to provide a more efficient implementation for both communications. Figure 1(b) shows the conceptual diagram of the E-RAPID network. All the interconnections on the board are implemented using an electronic crossbar

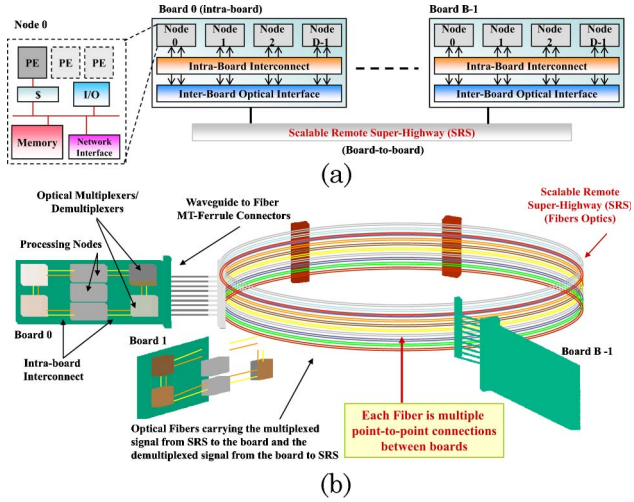

Fig. 1. (Color online) (a) Schematic of the E-RAPID architecture and (b) conceptual diagram of the E-RAPID network.

as explained in Subsection 2.A, and the interconnections from the board to the SRS are implemented using optical fiber, optical multiplexers, and demultiplexers as explained by the RWA in Subsection 2.B. Although the architecture is shown as a ring system, this is only done for clarity of the illustration. E-RAPID is actually implemented as a point-to-point topology as explained next.

A. Intraboard Electronic Switch

Intraboard communication is implemented by use of the crossbar switch design as shown in Fig. 2. The network interface at every node is composed of send and receive ports, which are connected to the optical transmitter and receiver ports through a bidirectional switch. For D nodes with W wavelengths per board, a $2D \times 2W$ crossbar is needed for complete connectivity. Although crossbar implementations have an $O(N^2)$ requirement, other interconnection networks can also be implemented for onboard communication. Here we consider a crossbar

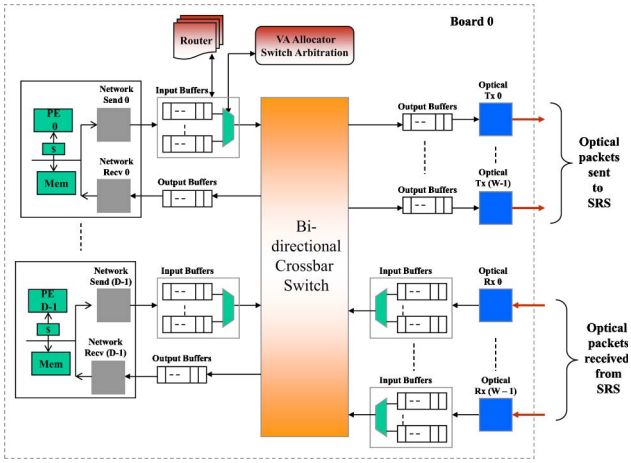

Fig. 2. Onboard interconnect in E-RAPID; board 0 is shown as an example, which consists of network send and receive pairs along with optical transmitters and receivers.

implementation for intraboard communication, since the number of nodes and transmitters is reasonable. The send port of a given node can communicate with the receive port of other nodes for intraboard communication and can communicate with the optical transmitters for interboard communication. Each optical transmitter is associated with a single wavelength λ_k, where $k = 0, 1, \ldots W - 1$ for interboard communication. The RWA for interboard communication is explained in Subsection 2.B. The multiplexed signal received at the board is demultiplexed when each optical receiver detects a single wavelength λ_k. This packet is then rerouted through crossbar switches to the appropriate node receiver ports.

Each packet that arrives in the physical input buffer progresses through various stages in the router before it is delivered to the appropriate output port. These input buffers are typically separated into several virtual channels that can be used to prevent deadlocks, implement fair scheduling, and increase the throughput by allowing block packets to progress. The input buffers and other router resources are allocated in fixed-size units called *flits* and each packet is broken into several flits. The progression of a packet is separated into *per-packet* and *per-flit* steps. As soon as *header flit*, the first flit of the packet arrives, the per-packet actions are initiated that includes (1) route computation: based on the information stored in the header, the output port of the packet is selected; and (2) virtual-channel allocation (VA): a packet must gain exclusive access to an output virtual channel associated with the output port. After the per-packet scheduling is completed, the per-flit scheduling begins that includes (3) switch allocation (SA): if there is a free buffer in its output virtual channel, flit can compete for access; and (4) switch traversal (ST): the flit is now transferred from the input buffer to the output buffer. Steps (3) and (4) are repeated for each flit of the packet. After transmission of the tail flit, the last flit of the packet, the virtual channel is freed and is available for another packet. We do not present the detailed implementation of the virtual channel allocator and switch arbitration in this paper and the readers can refer to [19] for more explanation.

B. Interboard Routing and Wavelength Assignment

The RWA for interboard communication for a 4 board, 4 nodes/board, and 1 cluster is shown in Fig. 3. For remote communication, different wavelengths from various boards are selectively merged to separate channels to provide high connectivity. Remote wavelengths are indicated by $\lambda_i^{(s,c)}$, where i is the wavelength, s is the source board number, and c is the cluster number from which the wavelength originates. To simplify, c is dropped as a single cluster working is explained. The wavelength assigned for a given source board s and destination board d is given by $\lambda_{B-(d-s)}^{(s)}$ if $d > s$ and $\lambda_{(d-s)}^{(s)}$ if $s > d$, where B is the total number of boards in the system, the superscript (in parentheses) indicates the source board and the subscript indicates the wavelength to be transmitted on. For example, if any node on board 1 needs to communicate with any node on board 2, the wavelength to be used is $\lambda_3^{(1)}$ and, for reverse communication, the wavelength required is $\lambda_1^{(2)}$. To illustrate with an example, consider a board 0 transmitter set. All the nodes on board 0 have an array of transmitters such that they can transmit on any wavelength $\lambda_i^{(0)}$, $i = 0, 1, 2, 3$. Any node on board 0 can communicate with itself on $\lambda_0^{(0)}$, with board 1 on $\lambda_3^{(0)}$ and with board 2 on $\lambda_2^{(0)}$. The physical fiber channel on which λ_0 is transmitted is called the home channel for that particular board (shown as a dotted line for board 0). All the signals that originate from a particular board are demultiplexed and then selectively multiplexed with different home board channels. For board 0, the multiplexed signal on home channel $\lambda_0^{(0)} + \lambda_1^{(1)} + \lambda_2^{(2)} + \lambda_3^{(3)}$ is then demultiplexed at the board 2 receiver. As the receivers are fixed, λ_i, $i = 1, 2, 3$ are received by node $i - 1$. For remote traffic, the number of wavelengths required to obtain the connectivity mentioned above is B, i.e., $(B - 1)$ wavelengths are required to communicate with every other board and one more wavelength λ_0 for multicast communication. The wavelengths used in each home channel are the same, thereby reusing the same set of wavelengths. Multicast and broadcast implementation have previously been described in [15].

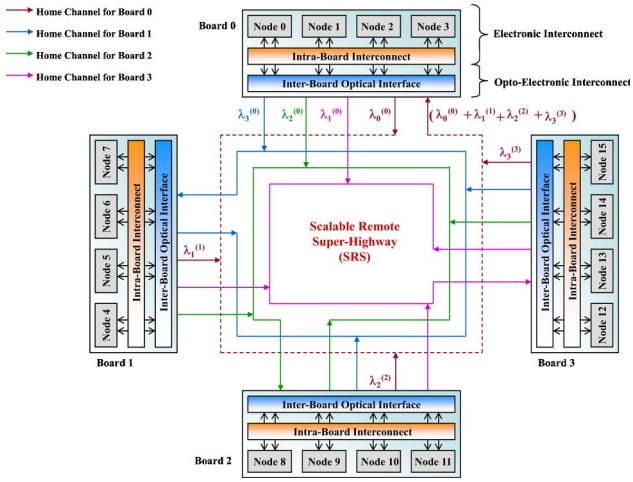

Fig. 3. Static RWA in E-RAPID for interboard communication. The example shows four boards, each consisting of four nodes connected within a cluster.

As the propagation of packets in E-RAPID is across multiple technologies, one significant distinction should be made. Flits from different nodes are interleaved in the electrical domain using virtual channels, whereas packets from different boards are interleaved in the optical domain. Although flit transmission in the optical domain is feasible, flit management across multiple domains is extremely complicated. Therefore, each packet transmitted by the node's send port or optical receivers is transmitted using flits, whereas the optical transmitters

queue the individual flits and transmit the entire packet in the optical domain.

3. Optical Reconfigurable Architecture

In E-RAPID, the RWA allocated bandwidth statically between various communicating boards using different wavelengths, fibers, and time slots. Static allocation of channels offers every node with equal opportunity for communication regardless of the network loads. Although static allocation ensures fairness and is suitable for uniform traffic patterns, it can lead to network congestion for nonuniform communication patterns. On the other hand, dynamic reallocation of channels in response to actual network load could lead to improved performance for most communication patterns.

To achieve dynamic reconfiguration of system architecture, E-RAPID can be extended by adding arrays of transmitters, link-level controllers (LCs) and reconfiguration controllers (RCs) to the intraboard interconnect as shown in Fig. 4. Dynamically reconfigurable E-RAPID has several advantages: (1) E-RAPID will not require any active optical switching for reconfiguration. It will rely only on passive optical components such as couplers. While several reconfigurable architectures use electromechanical or electro-optical switching elements [20–22], the proposed E-RAPID does not have any such element with the bandwidth switching controlled at the source. (2) E-RAPID could reallocate maximum link and even system bandwidth between boards. This is extremely useful for hot-spot or bursty traffic patterns, where extreme load is placed for a very short duration of time. (3) The reconfiguration mechanism in E-RAPID is completely decentralized and could happen between any boards without affecting the ongoing communication in the overall system.

A. Reconfiguration Mechanisms

The proposed reconfiguration mechanism in E-RAPID is explained with Fig. 5. Consider two source nodes $S1$ and $S2$ located on the same transmitting board (source board) and two destination nodes $D1$ and $D2$ located on the same receiving board (destination board). We begin with the assumption that $S1$ communicates with $D1$ and $S2$ communicates with $D2$. Figure 5(a) shows the nonreconfigured E-RAPID communication mechanism. Both packets from $S1$ and $S2$ enter the same transmitter queue and are multiplexed in time by virtual channels and fair arbitration by the switch allocator. The transmitter Tx_0 waits for the entire packet from each of the sources to be received and then transmits one after the other. These packets are received by the receiver Rx_0, which then starts transmitting to the destination nodes $D2$ and $D1$. Both the transmitter and the receiver are active for any packet propagation from any node at the source board to any node at the destination board. All packets are time multiplexed on the channel from the transmitter to the receiver. As the network load increases or more nodes from the same source board communicate with the destination board, more packets are queued in the transmitter queue, leading to an increase in the network latency.

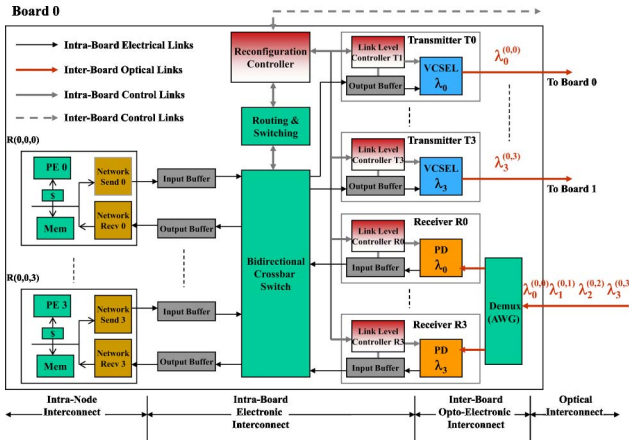

Fig. 4. Proposed E-RAPID architecture with a reconfiguration controller and a LC.

For example, from Fig. 3 consider the source board to be 1, the destination board to be 0, and the transmitter wavelength to be $\lambda_1^{(1)}$. Now suppose the wavelength $\lambda_2^{(2)}$ that board 2 is allocated to communicate with board 0 is not being used. As the load between boards 1 and 0 increases, this idle wavelength $\lambda_2^{(2)}$ could be better utilized by board 1 instead of board 2. Let us assume that we had this additional wavelength for communication from the same source board 1 to the destination board 0. Figure 5(b) shows 2 transmitters associated with Tx_0, and they are connected to the same transmitter queue. Packets from the two sources, $S1$ and $S2$, are time multiplexed through the switch, and the transmitter queue receives both packets. These packets enter different transmitters and are transmitted at different wavelengths. From Fig. 4 we note that all the wavelengths received by a destination board are separated into different receivers. These receivers Rx_0 and Rx_1 transmit the packets to the appropriate destinations $D1$ and $D2$. This reconfiguration mechanism spreads the communication traffic in the optical domain, thereby reducing the network latency. Moreover, by using additional wavelengths from other source boards, better resource utilization is achieved. While this reconfiguration mechanism has the potential to show better performance than that in Fig. 5(a) for small sized networks, it does not scale well with the number of nodes. As the number of nodes per board increases, there will be more contention for the transmitter queue from different sources. Even with an increase in the number of virtual channels, the improvements obtained are marginal. Another scheme shown in Fig. 5(c) is to spread the communication traffic completely, all the way from the source nodes to the destination nodes. This is based on the principle of sending the communication traffic to separate transmitter queues on the source board. This methodology provides a scalable solution along with a simpler technology implementation. In the

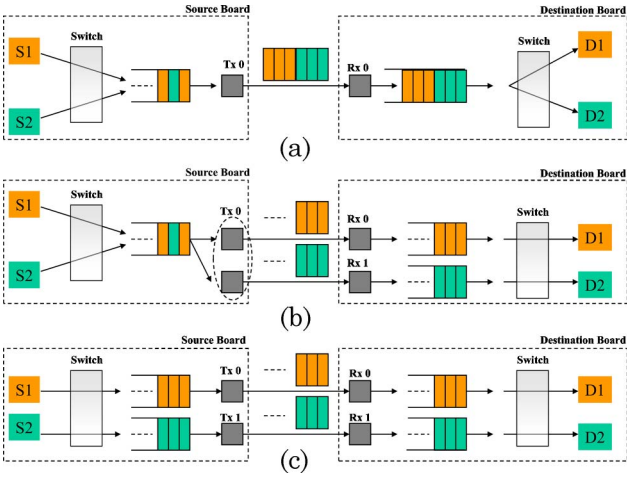

Fig. 5. (a) Nonreconfigured communication in E-RAPID, (b) reconfigured communication in E-RAPID based on the same optical transmitter queue, (c) a reconfigured communication in E-RAPID based on different optical transmitter queues.

following subsections, we discuss the technology and the algorithm for the proposed reconfiguration based on Fig. 5(c).

B. Technology for Reconfiguration

The enabling technology for E-RAPID is shown in Fig. 6. This configuration is based on the example in Fig. 3, with $D = 4$, $B = 4$, and $C = 1$. Figure 6 shows the same proposed E-RAPID design as in Fig. 4, except that there are multiple lasers emitting at the same wavelength per transmitter. Each transmitter is associated with four output ports (a, b, c, and d) as there are four boards in the system. The notation $\lambda_x^{(y)}$ is used here to indicate wavelength x originating from port y for a given transmitter. The statically assigned wavelength as per the communication requirements from Fig. 3 is highlighted.

The ability to dynamically switch multiple wavelengths through different ports of a given transmitter simultaneously to different system boards using passive couplers forms the basis for system reconfigurability in E-RAPID. This provides the flexibility to E-RAPID where more than one wavelength can be used for board-to-board communications in the case of increased traffic loads. The basis of reconfiguration is to combine different wavelengths at a given coupler from similar numbered ports but from different transmitters to facilitate the communication as explained in Fig. 5(c). Referring to Fig. 6, the multiplexed signal appearing at coupler 1 is composed of all the signals inserted by the same numbered b ports [$\lambda_0^{(b)}$, $\lambda_1^{(b)}$, $\lambda_2^{(b)}$, and $\lambda_3^{(b)}$] but from different transmitters. Now, when needed, different destination boards can be reached by more than one static wavelength, thereby enabling the dynamic reconfigurability of the proposed architecture. For example, from Figs. 3 and 4 assume that the traffic intensity from board 0 to 2 is high.

The static wavelength being used to communicate between boards 0 and 2 is $\lambda_2^{(c)}$ at coupler 2. The other

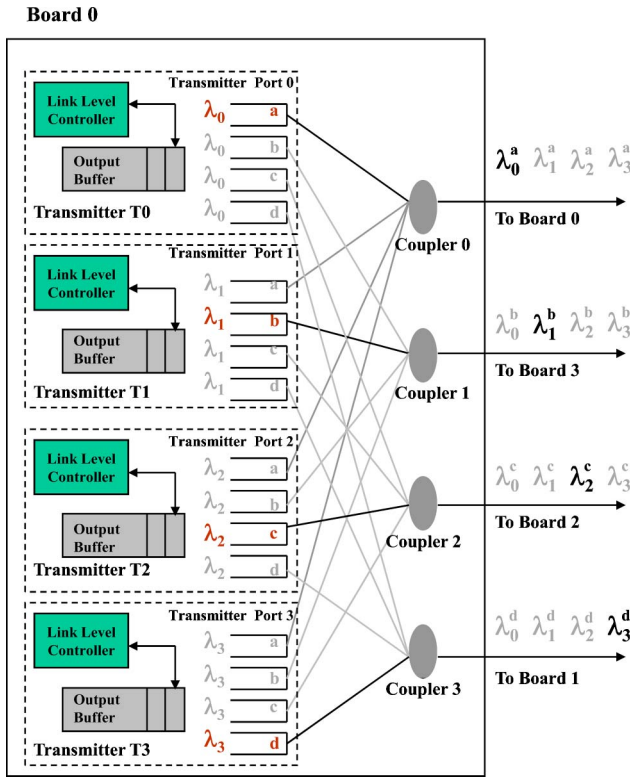

Fig. 6. Proposed technology for reconfiguration using passive couplers and an array of transmitters.

wavelengths, $\lambda_0^{(c)}$, $\lambda_1^{(c)}$, and $\lambda_3^{(c)}$ that appear at the same coupler 2, could be used if other boards (board 1, 2, or 3) release their statically allocated wavelengths (with which they can communicate with board 2) to board 0. If board 1 releases wavelength λ_1 to board 0, then board 0 can start using port c at transmitter 1 [$\lambda_1^{(c)}$] in addition to port c at transmitter 2 [$\lambda_2^{(c)}$], thereby doubling the bandwidth and reducing communication latency. The physical link over which both wavelengths, λ_1^c and λ_2^c, propagate is the same, whereas the different channels are formed between different transmitters 1 and 2 at board 0 with different receivers at board 2. In Subsection 3.C the LS protocol that implements this reconfiguration is explained.

C. Dynamic Reconfiguration Algorithm: Lockstep Protocol

To achieve DBR, the reconfiguration algorithm should at minimum (a) detect load/traffic imbalance in the system, (b) decide how to reconfigure, (c) reallocate the system bandwidth, and (d) resynchronize the system. To implement all the above requirements for dynamic reconfiguration, we propose a lightweight DBR technique called the LS protocol. The LS protocol is a history-based reconfiguration algorithm that triggers reconfiguration phase, disseminates state information, reallocates system bandwidth, and resynchronizes the system periodically with low control overhead to achieve optimized resource utilization, thereby improving the performance of the system. The LS protocol reallocates

wavelengths associated with idle channels to busy channels based on historical information. In the LS protocol, each reconfiguration phase works in several circular stages, each stage is implemented either as a request or a response stage between board-level RCs and LCs. Each RC triggers the reconfiguration phase, communicates with the local LCs and other RCs to determine the network load based on state information collected during the previous phase. The LS protocol works in the background and does not affect the ongoing communication, thereby minimizing the impact of reconfiguration latency on the overall network latency.

Reconfiguration Statistics: Historical statistics are collected with the hardware counters located at each LC. Each LC is associated with an optical transmitter to measure link statistics, and to turn the laser on/off. The link utilization $\text{link}_{\text{util}}$ tracks the percentage of router clock cycles when a packet is being transmitted in the optical domain from the transmitter queue. The buffer utilization $\text{buffer}_{\text{util}}$ determines the percentage of buffers being utilized before the packet is transmitted. At low-to-medium network loads, $\text{link}_{\text{util}}$ provides accurate information with regard to whether a link is being used at all, whereas $\text{buffer}_{\text{util}}$ provides accurate information with regard to network congestion at a medium-to-high network load. Another statistic used in congestion control is packet age, $\text{packet}_{\text{age}}$, that determines how long the packet has queued in the input buffers [23]. All these statistics are measured over a sampling time window called reconfiguration window or phase R_w. This sampling window impacts performance, as reconfiguring at low granularity incurs a latency penalty and reconfiguring coarsely might not adapt in time for traffic fluctuations. We utilize network simulations to determine an optimum R_w of 2000 simulation cycles.

Each RC_i, $i = 0, 1, \ldots B-1$, is connected to all the LC_j, $j = 0, 1, \ldots D-1$, on the board. In addition, each RC_i is also connected to $(\text{RC}_{i+1})\text{modulo}B$ in a simple electrical ring topology separated from the optical SRS. A ring topology with unidirectional flow of control ensures that what information is sent in one direction is always received in another. Electrical ring topology is used as we expect short bursts of communication between closely spaced boards (<1 m). Figures 7(a) and 7(b) show the two communication stages, RC–LC and RC–RC of the reconfiguration implementation. Each LC associated with a transmitter has a link utilization counter, a buffer utilization counter, and an on/off binary value for each wavelength λ_0, λ_1, λ_2 … on a given system board.

The symmetry of E-RAPID with respect to the number of wavelengths provides insight into the reconfiguration algorithm. For example, if $\Lambda = \lambda_0, \lambda_1, \lambda_2 \ldots \lambda_{B-1}$ is the total number of wavelengths associated with the system, we can see that this is exactly the same number of wavelengths transmitted/received from every system board. In other words, the number of outgoing or incoming wavelengths

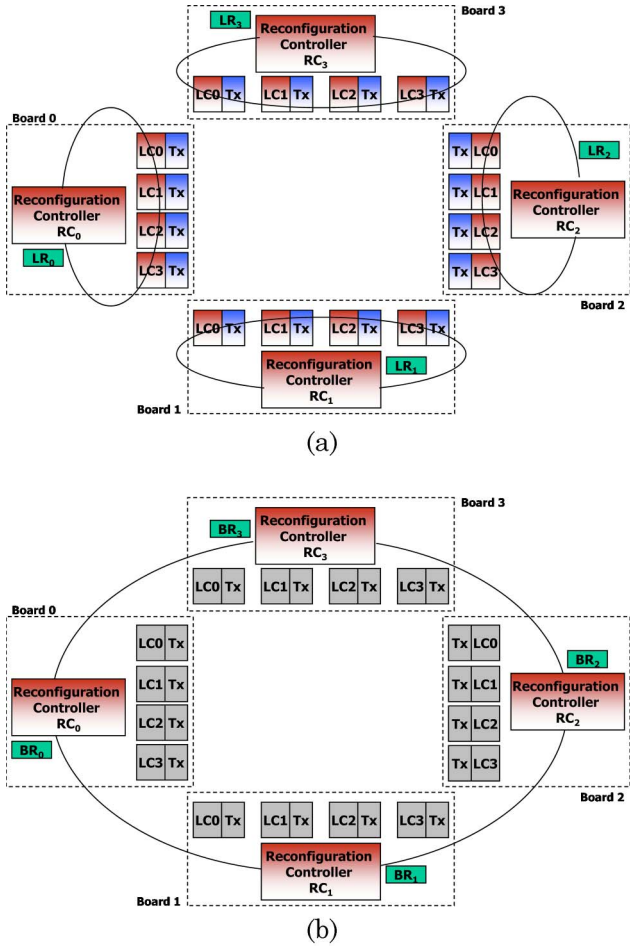

Fig. 7. Reconfiguration algorithm implementation: (a) the LR (link request/response) packet sent to all the nodes within a board and (b) the BR (board request/response) packet sent to all the boards within a cluster.

per system board is the same. Therefore, to balance the load and reallocate wavelengths on a given link, the system board needs all the link statistics on its incoming links. This is achieved by coordination between the LCs and the RCs as explained in the LS algorithm.

Reconfiguration Algorithm: To implement the LS protocol, RCs evaluate the state information and reallocate the bandwidth for the current R_w based on previous R_w. After RCs have decided which links to reallocate, this information is disseminated back to the RCs on other boards. The pseudocode of the LS algorithm is shown in Table 1. After R_w, in Step 2, RC_i sends out $\text{link}_{\text{request}}$ packets to LC_i as shown in Fig. 7(a). When this packet is received by RC_i, it updates all the outgoing link statistics. In Step 3, each RC_i sends $\text{board}_{\text{request}}$ packets to obtain all the link statistics for its incoming links as shown in Fig. 7(b). As it sends out, due to the symmetry of the ring architecture, it receives $\text{board}_{\text{request}}$ packets from other RC_i. For example, when board 1 receives BR_0 from say board 0, it will update the field for the wavelength at which board 1 communicates with board 0, i.e., λ_1 using the data stored in its outgoing

Table 1. LS Algorithm for DBR Implementation

Step 1: Wait for reconfiguration window, R_w
Step 2: Each RC_i sends the $link_{request}$ control packet to all its outgoing LC_i
Step 2a: Each LC_i computes the $link_{util}$ and $buffer_{util}$ for the previous R_w and updates the field in the $link_{request}$ packet and forwards to the next LC_{i+1} and finally to RC_i
Step 3: Each RC_i sends the $board_{request}$ control packet to all RC_j, $i \neq j$
Step 3a: RC_i updates the $link_{util}$ and $buffer_{util}$ for the link (wavelength) with which it communicates with RC_j when it receives the $board_{request}$ packet from RC_j
Step 4: RC_i receives its $board_{request}$ packet containing utilization information for all its incoming links
Step 4a: RC_i classifies every $B-1$ incoming links for DBR as
If $link_{util} \leq L_{min}$ => underutilized
If $link_{util} \geq L_{min}$ and $buffer_{util} < B_{con}$ => normal utilized
If $buffer_{util} > B_{con}$ => overutilized
Reallocates underuilized links to overutilized links
Step 5: Each RC_i sends the $board_{response}$ control packet with updated link information to RC_j, $i \neq j$
Step 5a: RC_i updates the wavelength reallocation for the link with which it communicates with RC_j when it receives the $board_{response}$ packet from RC_j
Step 6: Each RC_i sends the $link_{response}$ control packet to all its incoming LC_i with updates link reallocation information
Step 7: In response to DBR, each LC_i turns the lasers off/on for wavelength reallocation
Step 8: Go to step 1

link statistic. When board RC_i receives its own $board_{request}$ packet, it updates all the incoming link statistics.

In step 4, DBR is implemented. Now, each RC_i computes if reconfiguration is necessary based on buffer congestion, B_{con} and minimum link utilization L_{min}. While profiling traffic traces can provide more accurate information with regard to when the network is actually congested, setting the B_{con} to 0.5 is fairly reasonable for most traffic scenarios. This implies that, on an average, 50% of our buffers are occupied by packets for the given reconfiguration window R_w. We set L_{min} to 0.0, which indicates no packets are being transmitted on the link. Each incoming link statistic is classified into three categories as underutilized if $link_{util}$ is less than L_{min} (implying that this wavelength can be reallocated), normal utilized if $buffer_{util}$ less than B_{con} and $link_{util}$ is greater than L_{min} (implying the wavelength is well utilized) and overutilized if $buffer_{util}$ is greater than B_{con} (implying that additional wavelengths are needed). RC would allocate the underutilized links to the overutilized links.

In Step 5 and from Fig. 7(b), each RC_i now sends out $board_{response}$ packets to all the remaining board RCs to update their outgoing link statistics. As in board request stage, RC_i updates the information received from other RCs for the transmitters with which RC_i communicates with those boards into its outgoing link statistics. In Step 6 and from Fig. 7(a), each board RC_i sends out $link_{response}$ packets using the data received from its outgoing link statistics to each of the LC_i. In Step 7, each LC_i updates the state information received, thereby turning the lasers either on/off.

The LS protocol detects load imbalance by using reconfiguration statistics, which is implemented by the switching mechanism proposed in Section 3. B and the bandwidth are reallocated according to the algorithm explained in Table 1. In addition, the transmitter and the receiver need to be resynchronized after reconfiguration to prevent any collision. This is implemented by delaying the new transmitter from transmitting packets until the old transmitter has cleared all the packets.

4. Performance Evaluation

The performance of E-RAPID is evaluated using YACSIM and NETSIM [24] discrete-event simulators and is compared to various electrical interconnects for both uniform and nonuniform traffic traces [25]. The electrical networks chosen for comparison were 2-D torus, hypercube, and fat-tree topologies. These topologies are the most common clustering interconnects, for example, the 2-D torus is used in the Alpha 21364 network [26]; the hypercube is used in the SGI Spider chip used for SGI Origin machines [27]; and the fat-tree topology is the basis of most Mellanox switches used in Infiniband architectures [28] as well as in Elan 2 used in QsNet [25]. In addition, we compared E-RAPID with two other RAPID topologies, RAPID and M-RAPID architectures [16]. While RAPID is an all-photonic network, M-RAPID replaces optics with electronics for the intraboard interconnect for local communication only.

A. Simulation Methodology and Architectural Assumptions

YACSIM is a discrete-event simulation engine and NETSIM is an electrical network component library. These two can be combined to construct a wide range of direct and indirect electrical interconnects. We modified the baseline wormhole routed NETSIM with virtual channels to decouple the allocation of channel bandwidth from channel state to achieve substantially higher throughput. Due to the lack of optical simulators at the system level, we augmented the NETSIM component library by adding several optical components such as couplers, fibers, waveguides, demultiplexers, and splitters and developed a simulation environment called OPTISIM [29]. In OPTISIM the functional modeling of each of these components at the system level was implemented to

determine three parameters of interest: (1) length to determine the propagation latency, (2) attenuation to determine the signal loss due to components, and (3) wavelength to determine the routing within a component (demultiplexer). The components were then connected to design various WDM-routed RAPID configurations. For M-RAPID and E-RAPID, we designed the electrical intraboard interconnect using a crossbar switch.

We use cycle accurate simulations to evaluate the performance of RAPID configurations and other electrical interconnects. Packets were injected according to the Bernoulli process based on the network load for a given simulation run. The network load is varied from 0.1 to 0.9 of the network capacity. The network capacity was determined from the expression N_c (packets/node/cycle), which is defined as the maximum sustainable throughput when a network is loaded with uniform random traffic [19]. The simulator was warmed up under load without taking measurements until a steady state was reached (up to 1000 cycles). Then a sample of injected packets was labeled during the measurement interval (1000–10000). The simulation was allowed to run until all the labeled packets reached their destinations.

For the router model designed, the channel width is 16 bits and the speed is 400 MHz, resulting in a unidirectional bandwidth of 6.4 Gbits/s and a per-port bidirectional bandwidth of 12.8 Gbits/s. Credit-based flow control is implemented for a single flit buffer with credits incurring a single cycle channel delay. For the optical network, we assume a channel speed of 10 GHz, based on current optical technology. At 10 Gbits/s data rates, the transmission of an 8 byte flit takes around 6.4 ns$(= (8 \times 8)/(10 \times 10^9))$. For most of the runs we maintained a constant packet size of 64 bytes, resulting in an 8 flit packet size.

Experimental Setup: Network workloads that accurately reflect the high temporal and spatial traffic variance of many parallel numerical algorithms usually employed by scientific applications are most useful for evaluating the performance of HPC systems. Here we present three sets of traces:

a. *Uniform Traffic:* In this pattern, each node randomly selects its destinations with equal probability.

b. *Permutation Patterns:* In these static communication patterns, each node selects a fixed destination for all its transactions. The permutation patterns tested were (1) bit-reversal (node with binary coordinates $a_{n-1}, a_{n-2}, \ldots, a_1, a_0$ communicates with node $a_0, a_1, \ldots, a_{n-2}, a_{n-1}$), (2) butterfly (node with binary coordinates $a_{n-1}, a_{n-2}, \ldots a_1, a_0$ communicates with $a_0, a_{n-2}, \ldots, a_1, a_{n-1}$), (3) matrix transpose (node with binary coordinates $a_{n-1}, a_{n-2}, \ldots, a_1, a_0$ communicates with node $a_{n/2-1}, \ldots, a_0, a_{n-1}, a_{n/2}$), (4) complement (node with binary coordinates $a_{n-1}, a_{n-2}, \ldots, a_1, a_0$ communicates with $\overline{a_{n-1}, a_{n-2}, \ldots, a_1, a_0}$), (5) perfect shuffle (node with binary coordinates $a_{n-1}, a_{n-2}, \ldots, a_1, a_0$ communicates with node $a_{n-2}, a_{n-3}, \ldots, a_0, a_{n-1}$), and (6) neighbor (nodes are divided into pairs of adjacent nodes, for example, nodes 0 and 1, 2, and 3, n and $(n+1)$ with n even number, and each node communicates with its buddy).

B. Results and Discussion

Figures 8 and 9 show the throughput and average latency plots for uniform and permutation traffic

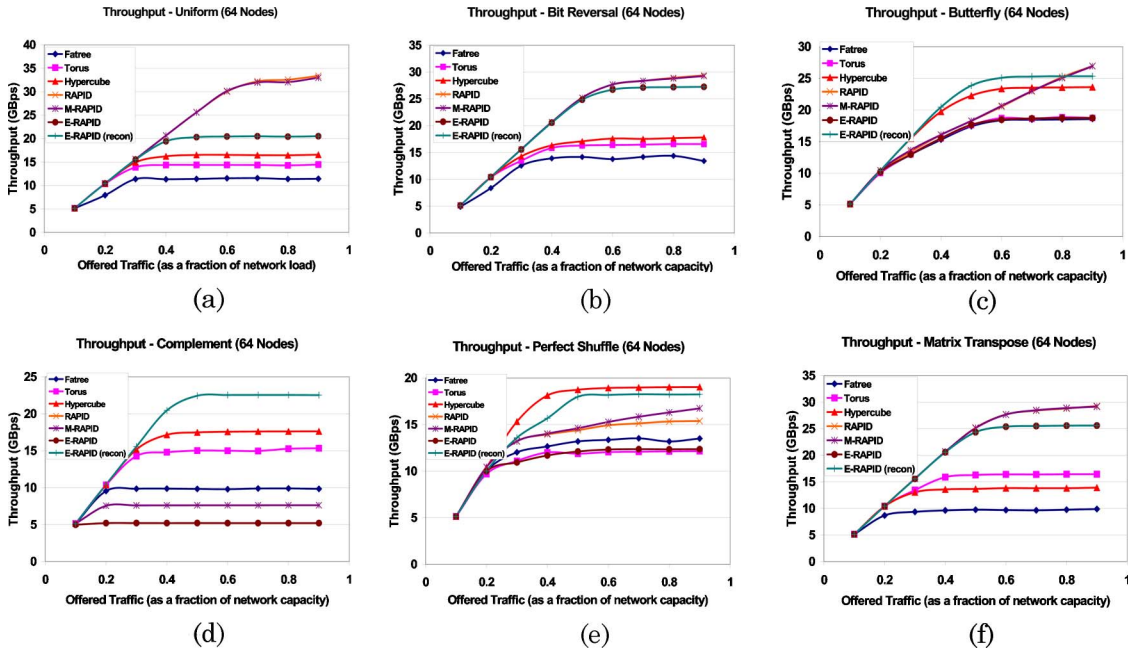

Fig. 8. Throughput for 64 nodes with reconfiguration for Uniform, Bit-Reversal, Butterfly, Complement, Matrix Transpose, and Perfect Shuffle traffic patterns. The networks compared are Torus, Fatree, Hypercube, and RAPID variations: RAPID, M-RAPID, E-RAPID, and E-RAPID (recon).

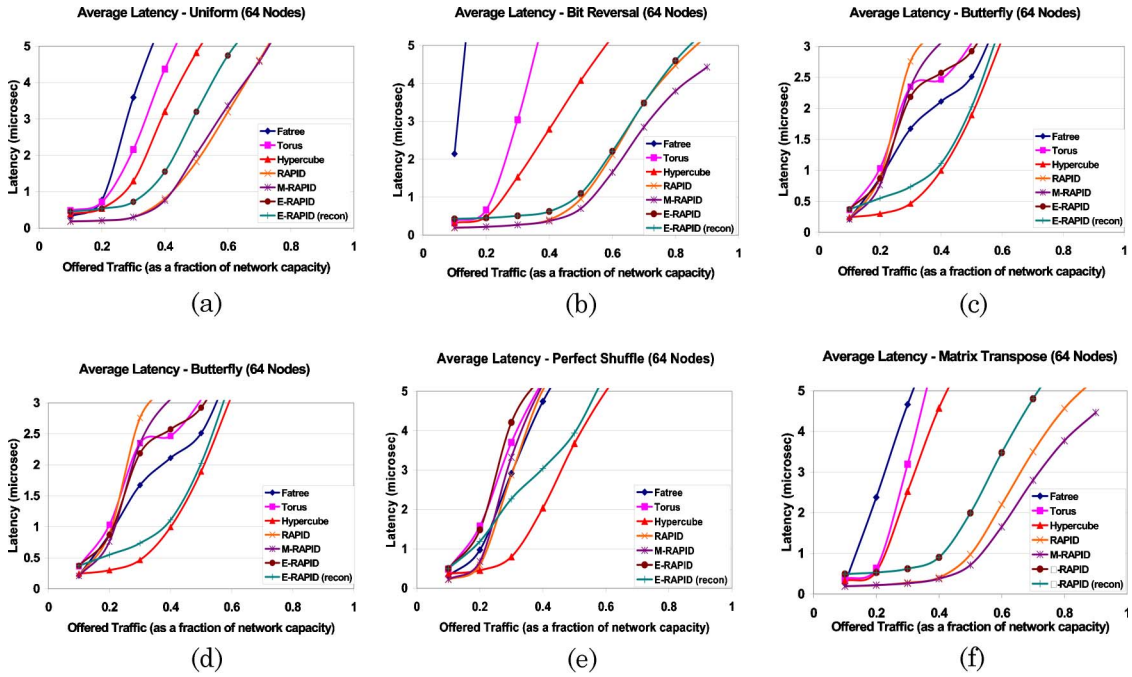

Fig. 9. Latency for 64 nodes with reconfiguration for Uniform, Bit-Reversal, Butterfly, Complement, Matrix Transpose, and Perfect Shuffle traffic patterns. The networks compared are Torus, Fatree, Hypercube, and RAPID variations: RAPID, M-RAPID, E-RAPID, and E-RAPID (recon).

for 64 nodes with eight nodes per board. Under uniform traffic conditions, the load is well balanced among all the system boards. In such a scenario, the proposed E-RAPID network without reconfiguration works as well as the E-RAPID with reconfiguration. The plot for throughput shows that E-RAPID outperforms the closest networks by 20% and is saturated later than electrical networks. More significantly, with reconfiguration, there is no excess latency/throughput penalty that is seen. This implies that the LS protocol independently evaluates if reconfiguration is necessary; if it cannot reconfigure the network, it does not hinder the ongoing communication. It should be noted that the best performance is obtained in RAPID and M-RAPID architectures that are purely optical architectures as opposed to E-RAPID which is an optoelectronic architecture. In E-RAPID, processors can communicate with the optical transceivers using the bidirectional crossbar as explained before. However, RAPID and M-RAPID have optical transmitters connected directly to the processor itself, which increases the bandwidth available in these architectures leading to an improvement in performance.

For bit-reversal and matrix transpose, the traffic is well balanced to begin with that the performance of E-RAPID with/without reconfiguration is the same. For butterfly, the reconfigured E-RAPID improves performance over the nonreconfigured by almost 38%, whereas for perfect shuffle the improvement is almost 50%. In perfect shuffle for board 0, node 0 communicates with itself, node 1 with node 8 (board 2), node 2 with node 4 (board 1), and node 3 with node 6 (board 1). Now when the reconfiguration algorithm is applied, the most advantage can be obtained if the communication can be spread between nodes 2 and 3.

The best case performance is observed for complement traffic. In complement traffic, node 0, node 1, node 2, and node 3 on board 0 communicate with nodes 15, 14, 13, and 12 on board 3. Therefore, in the nonreconfigured E-RAPID, the network is saturated even for low load. The LS algorithm reallocates the entire board bandwidth for communicating between boards 0 and 3. As board 3 receives data from only board 0, the entire incoming bandwidth is allocated to board 0. This results in an almost 300% improvement in performance in terms of throughput. This enormous improvement results in better performance than electrical networks with reconfiguration.

5. Conclusion

This research is focused on developing reconfigurable and scalable high-performance computing (HPC) systems using optical technology. The proposed LS protocol reallocates system bandwidth dynamically based on past link and buffer utilizations. The LS protocol does not incur any excess reconfiguration latency because it works in the background without affecting the ongoing communication. As the simulation results have shown, the proposed E-RAPID architecture provides significant performance advantages by making use of the enormous bandwidth of optical technology. In cases when the communication pattern imposes constraints, the reconfiguration algorithm then steps into the communication and alleviates the performance as seen in

butterfly, perfect shuffle, and most importantly for complement traffic patterns.

This research is supported by National Science Foundation (NSF) grants CCR-0538945 and ECCS-0725765.

References

1. D. E. Culler, J. P. Singh, and A. Gupta, *Parallel Computer Architecture: a Hardware/Software Approach* (Morgan Kaufmann, 1999).
2. J. Kash, C. Baks, S. Gowda, L. Graham, A. Hajimiri, C. Haymes, J. Jewell, D. Kucharski, D. Kuchta, Y. Kwark, P. Pepeljugoski, J. Schaub, C. Schuster, J. Tierno, and H. Wu, "Bringing optics inside the box: recent progress and future trends," presented at the 16th Annual Meeting of the IEEE/LEOS (2003), p. 23.
3. E. Mohammed, A. Alduino, T. Thomas, H. Braunisch, D. Lu, J. Heck, A. Liu, I. Young, B. Barnett, G. Vandentop, and R. Mooney, "Optical interconnect system integration for ultra-short-reach applications," Intel Technol. J. **8**, 114–127 (2004).
4. A. F. Benner, M. Ignatowski, J. A. Kash, D. M. Kutcha, and M. B. Ritter, "Exploitation of optical interconnects in future server architectures," IBM J. Res. Dev. **49**, 755–775 (2005).
5. T. S. D. Huang, A. Landin, R. Lytel, and H. L. Davidson, "Optical interconnects: out of the box forever?," IEEE J. Sel. Top. Quantum Electron. **9**, 614–623 (2003).
6. N. Kirman, M. Kirman, R. Dokania, J. Martínez, A. Apsel, M. Watkins, and D. Albonesi, "Leveraging optical technology in future bus-based chip multiprocessors," in *Proceedings of the 39th International Symposium on Microarchitecture* (IEEE, 2006).
7. A. Shacham, B. Small, O. Liboiron-Ladouceur, and K. Bergman, "A fully implemented 12 × 12 data vortex optical packet switching interconnection network," J. Lightwave Technol. **23**, 3066–3075 (2005).
8. "Closing the gap between peak and achievable performance in high performance computing," Tech. Rep. WP-0020404, CRAY Incorporated, Seattle, Washington (2004).
9. D. E. Lenoski and W.-D. Weber, *Scalable Shared-Memory Multiprocessing* (Morgan Kaufmann, 1995).
10. D. A. B. Miller, "Rationale and challenges for optical interconnects to electronic chips," Proc. IEEE **88**, 728–749 (2000).
11. J. Collet, D. Litaize, J. V. Campenhut, C. Jesshope, M. Desmulliez, H. Thienpont, J. Goodman, and A. Louri, "Architectural approach to the role of optics in monoprocessor and multiprocessor machines," Appl. Opt. **39**, 671–682 (2000).
12. A. V. Krishnamoorthy and K. W. Goossen, "Optoelectronic-VLSI: photonics integrated with VLSI circuits," IEEE J. Sel. Top. Quantum Electron. **4**, 899–912 (1998).
13. B. Lemoff, M. E. Ali, G. Panotopoulos, G. M. Flower, B. Madhavan, A. F. J. Levi, and D. W. Dolfi, "MAUI: enabling fiber-to-the-processor with parallel multiwavelength optical interconnects," J. Lightwave Technol. **22**, 2043–2054 (2004).
14. A. V. Krishnamoorthy, K. W. Goossen, L. M. F. Chirovsky, R. G. Rozier, P. Chandramani, S. P. Hui, J. Lopata, J. A. Walker, and L. A. D'Asaro, "16 × 16 VCSEL array flip-chip bonded to CMOS VLSI circuit," IEEE Photon. Technol. Lett. **12**, 1073–1075 (2000).
15. A. K. Kodi and A. Louri, "RAPID: reconfigurable and scalable all-photonic interconnect for distributed shared memory multiprocessors," J. Lightwave Technol. **22**, 2101–2110 (2004).
16. A. K. Kodi and A. Louri, "RAPID for high-performance computing systems: architecture and performance evaluation," Appl. Opt. **45**, 6326–6334 (2006).
17. A. K. Kodi and A. Louri, "A new technique for dynamic bandwidth re-allocation in optically high-performance computing systems," in *Proceedings of the 14th Annual IEEE Symposium on Hot Interconnects* (IEEE, 2006), pp. 31–36..
18. A. K. Kodi and A. Louri, "Power aware bandwidth reconfigurable optical interconnects for HPC systems," in *Proceedings of the 21st IEEE International Parallel and Distributed Symposium (IPDPS'07)* (IEEE, 2007), p. 81.
19. W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks* (Morgan Kaufmann, 2004).
20. P. Dowd, J. Perreault, J. Chu, D. Hoffmeister, R. Minnich, D. Burns, F. Hady, Y.-J. Chen, M. Dagenais, and D. Stone, "LIGHTNING: network and systems architecture," J. Lightwave Technol. **14**, 1371–1387 (1996).
21. P. Krishnamurthy, R. Chamberlain, and M. Franklin, "Dynamic reconfiguration of an optical interconnect," presented at 36th Annual Simulation Symposium (Society for Modeling and Simulation International, 2003).
22. C. M. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, "Dynamic reconfiguration of optically interconnected networks with time-division multiplexing," J. Parallel Distrib. Comput. **22**, 268–278 (1994).
23. X. Chen, L.-S. Peh, G.-Y. Wei, Y.-K. Huang, and P. Pruncal, "Exploring the design space of power-aware opto-electronic networked systems," in *Proceedings of the 11th International Symposium on High-Performance Computer Architecture (HPCA-11)* (IEEE, 2005), pp. 120–131.
24. J. R. Jump, "Yacsim reference manual," Rice University; available at http://www-ece.rice.edu/ rppt.html (1993).
25. F. Petrini, E. Frachtenberg, A. Hoisie, and S. Coll, "Performance evaluation of the quadrics interconnection network," Cluster Comput. **6**, 125–142 (2003).
26. S. S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb, "The Alpha 21364 network architecture," IEEE Micro **22**, 26–35 (2002).
27. M. Galles, "Spider: a high-speed network interconnect," IEEE Micro **17**, 34–39 (1997).
28. Mellanox Technologies, http://www.mellanox.com/.
29. A. Kodi and A. Louri, "Optisim: a system simulation methodology in optically interconnected HPC systems," IEEE Micro **28**, 22–36 (2008).