

# HIGH-FIDELITY MODELLING AND SIMULATION OF MYRINET SYSTEM AREA NETWORKS

A.D. George and R.A. VanLoon

*High-performance Computing and Simulation (HCS) Research Laboratory*  
Department of Electrical and Computer Engineering, University of Florida  
P.O. Box 116200, Gainesville, FL 32611-6200  
{george,vanloon}@hcs.ufl.edu

## Abstract

This paper presents a high-fidelity, event-driven model for performance analysis of various applications on Myrinet system area networks (SANs). The model is designed for the Block-Oriented Network Simulator (BONeS) commercial CAD environment for discrete-event simulation, is accurate down to the Myrinet character level, and supports a wide variety of topologies and parameter permutations. An analytical model is also presented by which the simulation model is verified, and validation of the simulation model is achieved through comparisons with experimental testbed results on both networking and parallel computing tests. In addition, a case study on performance analysis of Myrinet-based networks and clustered computers is included, where the impact of Myrinet data rate and buffer size on communication and computing performance is examined, further demonstrating the flexibility and versatility of the new simulation model.

## Key Words

BONeS, discrete-event simulation, high-fidelity system modelling, high-performance networks, Myrinet, system area networks

## 1. Introduction

Developments in processor technology and growth of the computer industry are increasing the need for faster and more powerful high-performance interconnection networks. One such network is Myrinet, a 1.28 gigabit-per-second (Gb/s) system area network (SAN) developed by Myricom, Inc. and standardized under ANSI VITA 26-

1998 [1, 2]. Developed with support from DARPA, Myrinet is designed to provide a high-speed and low-cost solution for communication in distributed-memory parallel computers and other message-passing architectures designed from commercial workstations and personal computers on a system-level scale. Although much has been reported in the literature on experimental research with Myrinet, simulative research on performance analysis, enhancement, and projection of Myrinet-based systems has received little attention despite the potential provided by new and emerging modelling and simulation theory, techniques and CAD tools for rapid virtual prototyping. This paper builds a foundation for such simulative research.

Simulative research on conventional networks and interconnects has received significant attention in the literature. Typically this research focuses on either the performance of the networks under study or the performance of the simulators themselves. For example, simulative analysis on selected characteristics of Ethernet and Token Bus was analyzed by Obaidat using event-driven simulation models, where verification was achieved through a comparison with previous analytical results [3]. Multistage bus networks have also been modelled. For instance, Bhuyan et al. employ execution-driven simulation models to analyze multistage bus networks and compare them with equivalent bi-directional multistage interconnection networks (MIN) [4]. Modelling and simulation has also been used to study the basic attributes of a few of the more recent high-performance interconnects. Both HiPPI (High Performance Parallel Interface) and SUCN (Storage Unit Control Network) have been modelled and simulated by Menascé et al. using the event-driven simulation language CSIM, both to verify an analytical model and determine their capability to serve as the networks for a hierarchical mass storage system [5]. Agrawal and Varshney present and analyze a protocol for a multimedia local ATM (Asynchronous Transfer Mode) network in their paper, as well as validate their protocol through discrete-event network simulation [6]. Other ATM research includes the integration of a mathematical model and a computer simulation model by Ani and Halsall to attain reasonable simulation time for modelling cell loss rate in ATM networks [7]. Examples of research that emphasizes the performance of the simulator itself include Ayani et al. on a gigabit-per-second LAN (SUPERLAN) [8], and Östman et al. on large ATM switch fabrics [9]. Both of these papers focus not on performance of the high-performance networks under study, but instead target the reduction of simulation time for these models. More specifically, parallel discrete-event simulation (PDES) is highlighted with discussion on conservative and optimistic modelling techniques.

The high-fidelity model of Myrinet presented in this paper is one of the few but not the only simulative research on the subject. Kleinrock et al. built a metropolitan area network (MAN), the Supercomputer SuperNet, consisting of an optical backbone linked with Myrinet [10]. Subsequently, Bagrodia et al. developed a model of the Supercomputer SuperNet, which includes both a Myrinet network model and an optical backbone model [11]. Their Myrinet model emulates first-generation Myrinet in the Maisie PDES language and is reportedly accurate to the discrete-byte level. Maisie is a C-based parallel language that allows processes to model network nodes as well as the communication between the Maisie processes to model network communication. Unfortunately, their Myrinet model includes only a 2x2-crossbar switch, and neither performance measurements nor verification and validation characteristics were presented. Instead, the main focus of the Bagrodia et al. work relates to a comparison of conservative and optimistic parallel simulation execution approaches.

This paper presents a new Myrinet model adhering to the second-generation Myrinet standard [1, 12]. The model was constructed with the BONEs (Block Oriented Network Simulator) platform, a commercial UNIX-based CAD tool from Cadence Design Systems [13, 14]. This Myrinet model is accurate to the discrete-byte (i.e. Myrinet character) level and allows various network topologies to be constructed, simulated, and evaluated in terms of networking and distributed computing characteristics. The Myrinet model permits non-intrusive collection of measurements at any level within the model, for pinpointing bottlenecks in complex networking topologies or gathering any desired performance measurements. In addition to networking simulation, the model can also be employed with the Integrated Simulation Environment (ISE), an extension to BONEs developed by the University of Florida, to simulate the performance of real parallel applications using the Message Passing Interface (MPI) over a Myrinet-connected multicomputer [15].

Three sets of experiments are conducted with the Myrinet model, providing verification and validation and demonstrating the model's versatility. The first set of experiments is driven by traffic delivered from a number of uniform traffic generators. In these fundamental experiments, the results from the discrete-event model are compared to data from a simple analytical model for verification. A realistic network traffic generator, in the form of a parallel MPI matrix-multiplication application, drives the second set of experiments. The execution of this application is accomplished by connecting the Myrinet model to the ISE. A comparison is then made between the Myrinet virtual prototype and a Myrinet testbed system for validation. The third set of experiments includes tests driven by both uniform traffic generators and an MPI matrix-multiplication application. These experiments vary

Myrinet system parameters to examine the effect of different link data rates and buffer sizes. In all three sets of experiments the high-fidelity Myrinet model is shown to perform accurately, and provide versatility and flexibility with the ability to support any number of parameter and configuration permutations, with reasonable simulation times.

The characteristics of the Myrinet model are described in Section 2. The three sets of experiments are detailed in Section 3. Section 4 examines the results from these experiments. Finally, concluding remarks and future work are discussed in Section 5.

## 2. Model Specifications

Myrinet is a switch-based, high-performance interconnect that boasts a link data rate of 1.28 Gb/s. The Myrinet standard is defined for use in SANs or multicomputer environments because of its limitation to work over "a few tens of meters" [1]. Myrinet links are composed of bi-directional, full-duplex channels. These bi-directional links allow control information to be inserted into the channels as necessary. Myrinet links are present in both network interface cards and crossbar-switch ports. This implementation allows the simple scenario where two Myrinet network interface cards (NICs) are directly connected, but more importantly it supports an unlimited number of more complex topologies constructed from switches, NICs, and their hosts. Myrinet employs source routing for packets, where the source NIC specifies the complete route a packet will traverse from source to destination.

Two generations of Myrinet specifications have been released. The first-generation specifications were released by Myricom [12], and the second-generation was standardized by the VITA Standards Organization [1]. The model developed for this work adheres to the second-generation Myrinet standard. The main difference between the specifications relates to the method of handling faulty links. In the first-generation protocol for Myrinet, a reset character is issued when a port has been in a non-sending state for a given period of time. The second-generation protocol, however, uses a constant frequency transmission mechanism to monitor the continuity of links and drops any packets that have not exited the Myrinet system in a given timeout period.

There are four types of Myrinet characters in the second-generation specification: control, data, packet-type, and source-route characters [1]. The main types of control characters include **BEAT**, **CRC** (Cyclic Redundancy Check), **GAP**, **GO**, and **STOP**. Based on a registry maintained by Myricom, packet-type characters are used to identify the upper-layer protocol assigned the handling of the packet once received at the destination, and consist of

a two-byte primary type and a two-byte secondary type. If there is a packet to be sent, a number of source-route characters (i.e. one per switch traversed from source to destination) are first transmitted to guide the packet through the crossbar-based network in a wormhole-routed fashion of pipelined communication. A source-route character is stripped off inside each switch encountered in the path and used to route the trailing characters. Following the source-route character(s) are four packet-type characters, an arbitrary number of data characters forming the payload of the packet, and a CRC character. At least one GAP character always separates packets. BEAT, GO, and STOP characters can be sent at any time, if necessary even interrupting current packet transmissions. BEAT characters are transmitted at a constant frequency to allow the continuity of links to be monitored. Small queues called slack buffers are employed at the input and output ports of a Myrinet system. The slack buffers use high-water and low-water marks to determine when a queue has become ‘excessively full’ or ‘sufficiently empty.’ These slack buffers allow GO and STOP characters to be used as simple flow-control mechanisms causing transmission to continue and halt, respectively. As mentioned previously, the second-generation protocol for Myrinet implements a timeout mechanism to terminate, or sink, any packet that has been traveling within the Myrinet system for a period of time greater than the timeout interval.

## **2.1 Discrete-Event Simulation Model**

The discrete-event simulation model implements the behavior described above at the Myrinet character level. It consists of over 90 model blocks, developed with numerous BONEs core modules, which span as deep as 8 hierarchical levels. Several parameters are provided for accurate simulation of the desired Myrinet architecture, including: *Link Data Rate*, *Low-water Mark*, *High-water Mark*, *Slack Buffer Size*, *BEAT Time*, *Timeout Interval*, *Length of Link*, and *Propagation Velocity*. These values can be varied to simulate a variety of Myrinet network configurations. The *Link Data Rate* of a Myrinet architecture can be set to simulate older or current Myrinet implementations, as well as to measure the performance of projected future Myrinet systems. Similarly, the speed of each traffic source may be increased or decreased to determine the performance when the I/O bus and network interface card of a host inserts data into the Myrinet system at different rates. The *Low-water Mark*, *High-water Mark*, and *Slack Buffer Size* can all be varied to study the impact and find the optimal queue settings for a given application. When analyzing Myrinet systems with fault injection, the *BEAT Time* can be varied to provide faster or slower fault detection. Similarly, the *Timeout Interval* may be varied in simulations to examine the relationship

between latency and reliable packet delivery. Finally, *Length of Link* and *Propagation Velocity* allow physical media of different length and composition such as copper or fiber to be simulated.

Fig. 1 shows the high-level operation of the model. The model accepts data from the traffic sources, encapsulates it into a Myrinet packet, and inserts the packet into the host interface module. The Myrinet packet is then segmented into Myrinet characters and stored in the outgoing slack buffer, as long as space is available. Once the Myrinet packet has been fully converted into Myrinet characters, subsequent data may enter the model and proceed in a similar manner. Myrinet characters exit the outgoing slack buffer sequentially. Characters may not be transmitted when the channel is in a STOP state or when Myrinet control characters are waiting to be transmitted.

Myrinet characters are transmitted out of the host interface module through one or more crossbar switch modules in the network fabric (e.g. a system with a single switch is depicted in fig. 1), or directly into another host interface module in the simple, switchless case. When Myrinet characters flow into a crossbar switch, they are stored in a slack buffer. As soon as the source-route character enters the switch, a request is made to the corresponding output port. Once the output port becomes free, the characters are removed from the slack buffer and transmitted to another crossbar switch or host interface. The main duties performed by the Myrinet crossbar switch models are handled by two types of modules. The port controller modules handle all port traffic and incorporate control characters as necessary. The store and request modules contain a slack buffer and request service from port controller modules.

Myrinet characters are stored in a slack buffer upon entering a host interface. Once the previous Myrinet packet has exited the Myrinet system, the Myrinet characters are removed from the slack buffer and begin forming another Myrinet packet. When the packet is fully reassembled, the destination host is notified that an incoming packet is ready. After approval is received, the data is stripped out of the Myrinet packet and sent to the traffic sink. The next characters and packet are handled similarly.

The model generates and transmits BEAT, CRC, GAP, GO, and STOP characters as mandated by the standard. The timeout mechanism is also enforced. Regardless of the number of slack buffers or transmission links that contain Myrinet characters from a specific Myrinet packet, the Myrinet packet is removed in full from the system when the packet's lifetime within the Myrinet system exceeds the timeout interval.

An extensive number of Myrinet switches and switch fabrics are available with the model. Several crossbar switches have been constructed including 2x2, 4x4, and 8x8 switches. Moreover, a crossbar switch of arbitrary size

can be assembled using the components in these basic switch modules. Larger systems combining multiple crossbar switches have also been built. For instance, an 18-node system is available that employs three 8x8-crossbar switches in a triangular configuration. In addition, a 96-node Myrinet system has been constructed that connects six 16-node clusters using an 8x8 crossbar-switch, where each 16-node cluster module is derived from the 18-node system. As evidenced, the model provides a versatile and flexible means by which to study any Myrinet topology.

## 2.2 Analytical Model

In order to predict basic throughput and latency performance for the purpose of verifying the simulative model of an  $N$ -node Myrinet network, an analytical model is needed. This analytical network model is comprised of  $N$  traffic units for generating and receiving data, where each is connected to a Myrinet host interface. The  $N$  Myrinet host interfaces are then interconnected via an  $N \times N$  Myrinet crossbar switch. Table 1 defines the parameters used in the analytical model.

Crossbar occupancy describes the contention experienced by an input packet in its travel through a crossbar switch from input port to desired output port. Bhandarkar derives  $C$  for an  $N \times N$  crossbar switch [16]. First, at any particular input of the crossbar, the probability that a request is directed to any particular one of  $N$  outputs is  $1/N$ . The probability that a particular output port is not requested by any of the  $N$  inputs is  $(1-1/N)^N$ . Thus, crossbar occupancy is defined as follows, where  $C = 1$  is the ideal case (i.e.  $N = 1 \Rightarrow C = 1$ , since there is no contention) and  $C$  decreases as  $N$  increases:

$$C = 1 - (1 - 1/N)^N \quad (1)$$

Consider the transfer of a packet from a source node to a destination node in a Myrinet network connected by a single  $N \times N$  crossbar switch. The first equation comprising the analytical model for this scenario describes the *average peak effective throughput (APET)* delivered to any one of the  $N$  traffic sinks. This throughput is found by multiplying the network data rate by the crossbar occupancy by the ratio of data bytes per packet to total bytes per packet. Total bytes per packet is calculated by summing one source-route byte, four packet-type bytes, one CRC byte, one GAP byte, and  $P$  data bytes. The equation follows:

$$APET = \frac{R \cdot C \cdot P}{P + 7} \quad [\text{bits/second}] \quad (2)$$

The second equation that makes up the analytical model describes the *one-way average latency (OWAL)* for data packets from their traffic source to their traffic sink. At time  $t = 0$  the first Myrinet character begins transmission from the source host interface. This character does not traverse the entire network, but is absorbed by the switch for routing, and therefore the second character (i.e. the first packet-type character) must be examined. The second character, which is the first one bound for the final destination, is sent after the first character has completed transmission at time  $t = 8/(R \cdot C)$ . The second character is received at the switch after both the transmission and propagation times have elapsed. Since the switch is below saturation and the first character has prepared the switch for transmission, once the second character has been fully received it may begin transmission from switch to destination node without delay. Thus, after the propagation delay on the link from the switch to the destination, the second character is received at the destination host interface. With the addition of the time for the three remaining packet-type characters,  $P$  data characters, and the CRC character, and the use of a factor of 8 to convert bytes to bits, the final formula is:

$$OWAL = \frac{8(1 + 3 + P + 1)}{R \cdot C} + \frac{2L}{V} \quad \text{[seconds]} \quad (3)$$

### 3. Experiment Specifications

Three sets of experiments used to evaluate the Myrinet model are detailed in this section. The first set is executed over both the simulative and analytical models. These experiments are driven by traffic with uniform interarrival rates and include two topologies. This first set of experiments verifies the basic operation of the simulative model via comparison to the analytical model. The second set of experiments consists of a matrix-multiplication algorithm implemented by MPI parallel code using matrices of various sizes over four network topologies. These experiments validate the simulative model by comparison to a physical testbed. Finally, the third set of experiments is a case study examining the effect of varying Myrinet link data rate and buffer size on network and parallel processing performance.

### 3.1 Verification Experiments

The topologies of the first set of experiments consist of either two or eight traffic generators, each of which driving a host interface and functioning with uniformly distributed interarrival rates and destination addresses. The host interfaces are connected to one another via either a 2x2 or an 8x8 crossbar switch. The general form of these topologies is shown in fig. 2.

This set of experiments collects latency and throughput results from both the simulation and analytical models. The simulation and analytical model parameters have been matched to a testbed in the laboratory to allow fair comparison and provide realistic results. This testbed consists of SBus-based Myrinet adapters in 170 MHz UltraSPARC workstations connected via a Myricom M2F SW8 8x8-crossbar switch. The parameter values are given in table 2.

Since in these verification experiments a node does not send packets to itself, *Crossbar Occupancy* has been calculated with  $N=1$  and  $N=7$  output ports occupied for the 2- and 8-node topologies, respectively. *Offered Load* for the traffic generators has been set to 1.44 Gb/s for throughput measurements and to 160 Mb/s for latency measurements. Throughput measurements are taken when the network is saturated (i.e. above the 1.28 Gb/s Myrinet data rate). Conversely, latency measurements are taken below saturation. Through iterative simulation, it has been determined that 160 Mb/s is sufficiently below saturation to yield good latency results. *Packet Payload Size* for the traffic generators varies from 4 bytes to 8 kB in powers of two to test a wide variety of payload lengths. *Propagation Velocity* has been set to the propagation speed through copper Myrinet cables, which is approximately 0.6 multiplied by the speed of light (3e8 m/s).

### 3.2 Validation Experiments

The second set of experiments consists of tests performed with the discrete-event simulation model and a physical testbed over the topologies shown in fig. 3. The systems all employ UltraSPARC workstations to execute MPI parallel matrix-multiplication processes, which act as traffic generators. This data traffic is fed to the simulation model via the BONEs/MPI Runtime Interface, which is an ISE mechanism [15]. Conversely, the physical testbed makes use of the Myrinet/MPI Interface, which is the Myrinet MPI on BDM MCP (“BullDog” Myrinet Control Program) software developed at Mississippi State University [17].

These experiments execute a parallel matrix-multiply application written in MPI-C. For  $A \times B = C$ , let  $C$  be computed as follows. The master processor evenly distributes rows of the  $A$  matrix to itself and the slave processors across the Myrinet network. The master processor then transmits  $B^T$ , the transpose of matrix  $B$ , to each of the slave processors. The purpose of the transposed form is to minimize cache misses during the dot-product calculations. Once a slave processor calculates its assigned rows of the solution, the respective elements of the  $C$  matrix are returned to the master processor. The application is complete once the master processor computes its  $C$  matrix rows and receives the corresponding  $C$  matrix rows from all of the slave processors.

All the parameters used in these experiments match those in the previous set of experiments. The physical testbed consists of eight UltraSPARC workstations with Myricom M2F-SBus32B network interface cards connected to each other via a Myricom M2F SW8 8x8-crossbar switch. The experiments were conducted with  $N \times N$  matrices where  $N$  is 32, 48, 64, 80, 96, 112, 128, and 144. Total execution time is measured, which allows both speedup (i.e. the ratio of parallel to sequential performance) and parallel efficiency (i.e. the percentage of perfect or linear speedup achieved) to be calculated.

### 3.3 Case Study

The third set of experiments consists of both networking and parallel computing tests performed on the discrete-event simulation model with the 8-node topologies explained previously in this section. For the networking experiment, the model is driven by a uniform interarrival traffic source at an *Offered Load* of 160 Mb/s using a *Packet Payload Size* of 64 bytes. For the computing experiment, traffic is generated by the parallel matrix-multiplication application executed on 64x64 integer matrices.

These two experiments investigate and demonstrate the versatility of the model. For both experiments the Myrinet link data rates are varied from 640 Mb/s to 5.12 Gb/s by powers of two. Also varied are the Myrinet buffer sizes controlled by the low- and high-water marks, which vary in doubled increments from 20 and 28 bytes to 160 and 224 bytes, respectively. Average latency and total execution time are measured for the networking and computing experiments, respectively.

## 4. Results

The results collected from the three sets of experiments are detailed below. The basic latency and throughput performance experiments are more fundamental and offer verification as well as demonstrate peak Myrinet

performance without higher-level overhead. The matrix-multiplication experiments represent the simulation of an entire multicomputer system and provide validation with higher-level overhead, demonstrating the capability of Myrinet to serve as the communication medium for an MPI-based, parallel-processing application. The case study experiments demonstrate the flexibility of the model and its ability to prototype and explore the effect on system performance of variations in the Myrinet network parameters.

Although experiment simulation times are not the primary focus of this paper, it is important to demonstrate that the model's simulation times are reasonable. To offer an estimate of the required simulation times, the experiments requiring the longest and shortest amount of time from each experiment set are reported along with the machine used for execution. The verification experiments required between 2 minutes and 17 hours on a 200 MHz UltraSPARC-2 workstation with 256MB of memory. The simulation times for the validation experiments ranged from 1 to 20 hours on an Ultra-2. When driven by a uniform traffic source, the case study experiments took between 12 minutes and 13 hours to complete on an Ultra-2, while the experiments driven by a matrix-multiplication application required 3 to 4 hours on a 300 MHz UltraSPARC-30 workstation with 256MB of memory. Moreover, using the distributed simulation mechanisms provided in BONEs, multiple iterations of a simulation (i.e. multiple instances of a permutation in parameters on a given model) can be simulated concurrently on multiple workstations. Thus, case studies like the one described in Section 3.3, where 16 iterations are required, can be conducted on 16 workstations in the same period of simulation time as a single permutation on a single workstation.

#### **4.1 Verification Experiments**

Throughput results are displayed in fig. 4. The 2-node throughputs are examined in fig. 4(a). The simulative results have a peak throughput of nearly 1.28 Gb/s. The analytical results are on average approximately 1.02 Mb/s or just 0.09% higher. Since the 2-node system is a relatively simple configuration with no contention in its 2x2 switch, we can expect the analytical and simulative results to track one another so closely. The results from the 8-node experiments are examined in fig. 4(b). Due to the switch contention associated with the uniformly distributed destinations used in the traffic patterns, the simulative results have a peak throughput of 790 Mb/s and average approximately 34.4 Mb/s lower than the analytical results. This difference of 4.80% demonstrates how the analytical approach loses accuracy in describing the behavior of a more complex system.

Fig. 5 contains the latency results. The 2-node simulation results in fig. 5(a) virtually match the analytical results with a relative difference of only 0.0004%, with a minimum latency of less than 0.17  $\mu$ s. Fig. 5(b) shows the

latency measurements for the 8-node topology. In this case, the minimum latency is just over 0.19  $\mu$ s, and the simulative results are on average approximately 0.73  $\mu$ s or 5.58% smaller than the analytical results.

In these experiments the simulative throughput and latency results from the Myrinet model yield results that are comparable to the analytical estimates. Of course, as the Myrinet systems and the application traffic patterns under study inevitably become more sophisticated, the simulative model's accuracy persists while the analytical model's accuracy quickly diminishes. The validity and usefulness of the Myrinet simulative model is further demonstrated and reinforced by the experiments that follow.

## 4.2 Validation Experiments

Execution-time results from all of the matrix-multiplication experiments are displayed in fig. 6. Fig. 6(a) examines the 2-node results. The two curves track each other closely, with the physical testbed results averaging approximately 0.87 ms or 0.90% less than the simulative results. The 4-node results in fig. 6(b) also follow each other closely where the simulative results average approximately 0.37 ms or just 0.77% lower than the testbed results. In fig. 6(c), the 8-node curves track almost as closely, with the simulative results averaging approximately 0.57 ms or 2.27% less than those from the testbed. Finally, the results for the 16-node experiment on the simulative model are shown in fig. 6(d). Since a physical testbed of this size was not available in the laboratory, no testbed results are included in this figure. However, the previous results justify a high degree of confidence in these simulative results.

Fig. 7 displays both the speedup and parallel efficiency results from the matrix-multiplication experiments. The 2- and 4-node experiments demonstrate nearly linear speedup over all matrix sizes. The 8- and 16-node experiments show improvement as the problem size increases. As seen in fig. 7(b), the 2-node system maintains a parallel efficiency above 95%. The 4-node system achieves 95% efficiency and above for matrix sizes of 80x80 integers and larger. The 8-node system reaches a 95% efficiency level for the 128x128 matrices and larger. Finally, the 16-node system reaches a maximum efficiency of only 85% with the largest matrices tested in this case, although higher efficiency is expected for matrices larger than 144x144. In general, systems with fewer nodes achieve higher efficiency for variations in problem size more quickly, while systems with more nodes require larger problem sizes to overcome the increased overhead of a larger system and perform well.

### 4.3 Case Study

With this final set of experiments the goal is to explore and demonstrate the potential of the simulative Myrinet model to support sensitivity studies, where changes in system or application parameters are injected and networking or computing performance is measured. Fig. 8 contains both the average-latency and execution-time results for the third set of experiments.

In fig. 8(a), significant decreases in latency are achieved in the networking experiment as the link data rate increases. By contrast, only slight decreases in latency from reduced flow-control overhead are achieved as buffer size increases. The results from the parallel computing experiment are shown in fig. 8(b). The trend is a decrease in execution time as the tests range from the smallest buffer and slowest link data rate to the largest and fastest, respectively. The largest reduction in execution time occurs as the link speed increases from 640 Mb/s to 1.28 Gb/s, after which the execution time then sharply levels off. This effect is due to the computation portion of the application dominating the communication portion once the link data rate reaches 1.28 Gb/s. Therefore, the data indicates that network data rates greater than 1.28 Gb/s would not significantly improve the performance of this parallel computing application on this system. These results demonstrate the ability of the model to support “what if” studies with any number of permutations in both network and application parameters.

This case study illustrates the potential of this modelling and simulation approach for the rapid virtual prototyping of Myrinet-based networks and distributed parallel computers. With the ability to support variations in any of the variables associated with the network, the system, and the applications, an unlimited number of design permutations can be posed and evaluated to support the most effective mapping between application and network architecture. Sensitivity studies like the one above will prove invaluable in the ability to predict performance strengths and weaknesses of candidate system configurations.

## 5. Conclusions

This paper has presented, examined, and demonstrated a precise and versatile Myrinet model. This model is believed to be the first CAD-based, high-fidelity, discrete-event simulation model for the latest generation of Myrinet ever designed, verified, and validated. With it, any desired Myrinet topology can be constructed with the modular switch and network interface components in the model. In addition, reasonable simulation times ranging

from 2 minutes to 20 hours are achieved with the model's event-driven scheme, since no time-consuming idle clock cycles are simulated, and multiple iterations are simulated concurrently on multiple workstations.

The Myrinet simulative model was shown to accurately depict the second-generation Myrinet standard through the development of two sets of experiments. The first set of experiments verified the simulative model with a simple analytical model in terms of effective throughput and latency characteristics. The second set of experiments focused on the execution of a parallel matrix-multiplication application in MPI over both the Myrinet simulative model and a real testbed in order to provide validation. Finally, a third set of experiments was conducted in the form of a case study for sensitivity analysis. In this study, the impact of changes in basic network parameter settings on networking and parallel computing performance was evaluated. The results illustrated the limitations inherent in network-based computing, where increases in network speed and buffer size may lead to diminishing returns in application performance unless and until other bottlenecks are identified and overcome. Moreover, this case study illustrates how any number of design permutations may be posed and evaluated with the rapid virtual prototyping mechanisms associated with this high-fidelity simulation model for Myrinet.

Several extensions to this Myrinet simulative research are planned for the future. Taking advantage of the flexible interface between model and application, further experiments with a broader range of application types may be pursued in order to quantify the performance characteristics of selected Myrinet configurations for other algorithms and applications of interest. Moreover, further simulative analyses are planned to compare the performance characteristics of Myrinet with those of other high-speed networks (e.g. SCI, ATM, Gigabit Ethernet, and Fibre Channel) through high-fidelity modeling and simulation. Potential enhancements to the Myrinet protocol will also be investigated in the future using these models, including new extensions to support logical or physical multicast and broadcast for collective communication. Finally, extensions to the simulative model itself are also planned, including the incorporation of fault-injection mechanisms and additional fidelity in the modules for the adapters and their I/O buses where the network meets the hosts.

## Acknowledgements

This work was sponsored in part by the National Security Agency and Sandia National Laboratories.

## References

- [1] VITA Standards Organization, *Myrinet-on-VME Protocol Specification Draft Standard*, [HTTP://www.vita.com/vso/draftstd/5v2.pdf](http://www.vita.com/vso/draftstd/5v2.pdf), 1998.
- [2] N.J. Boden, D. Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, & W. Su, Myrinet: A Gigabit-per-Second Local Area Network, *IEEE Micro*, 15(1), 1995, 26-36.
- [3] M.S. Obaidat, A Performance Simulation Study of the Ethernet and Token Bus Local Computer Networks, *Simulation*, 64(6), 1995, 381-396.
- [4] L.N. Bhuyan, R.R. Iyer, T. Askar, A.K. Nanda, & M. Kumar, Performance of Multistage Bus Networks for a Distributed Shared Memory Multiprocessor, *IEEE Transactions on Parallel and Distributed Systems*, 8(1), 1997, 82-95.
- [5] D.A. Menascé, O.I. Pentakalos, & Y. Yesha, An Analytic Model of Hierarchical Mass Storage Systems with Network-Attached Storage Devices, *Performance Evaluation Review*, 24(1), 1996, 180-189.
- [6] J.P. Agrawal & U. Varshney, Architecture and Performance of MLAN: A Multimedia Local ATM Network, *Simulation*, 64(1), 1995, 15-26.
- [7] C.I. Ani & F. Halsall, Simulation Technique for Evaluating Cell-Loss Rate in ATM Networks, *Simulation*, 64(5), 1995, 320-329.
- [8] R. Ayani, Y. Ismailov, M. Liljenstam, A. Popescu, H. Rajaei, & R. Rönngren, Modeling and Simulation of a High Speed LAN, *Simulation*, 64(1), 1995, 7-14.
- [9] F. Östman, T. Lazraq, & R. Ayani, Modeling and Parallel Simulation of Large ATM Switch Fabrics, *Simulation*, 70(1), 1998, 35-40.
- [10] L. Kleinrock, M. Gerla, N. Bambos, J. Cong, E. Gafni, L. Bergman, J. Bannister, S.P. Monacos, T. Bujewski, P. Hu, B. Kannan, B. Kwan, E. Leonardi, J. Peck, P. Palnati, & S. Walton, The Supercomputer Supernet Testbed: A WDM-Based Supercomputer Interconnect, *Journal of Lightwave Technology*, 14(6), 1996, 1388-1399.
- [11] R. Bagrodia, Y. Chen, M. Gerla, B. Kwan, J. Martin, P. Palnati, & S. Walton, Parallel Simulation of a High-Speed Wormhole Routing Network, *Proc. Parallel and Distributed Simulation Workshop*, Los Alamitos, CA, 1996, 47-56.
- [12] Myricom Inc., *Myrinet link specification*, [HTTP://www.myri.com/scs/documentation/link/index.html](http://www.myri.com/scs/documentation/link/index.html), 1995.
- [13] K.S. Shanmugan, V.S. Frost, & W. LaRue, A Block-Oriented Network Simulator (BONeS), *Simulation*, 58(2), 1992, 83-94.
- [14] Alta Group Inc., *BONeS DESIGNER User's Guide* (Foster City: Alta Group, 1994).
- [15] A.D. George, R.B. Fogarty, J.S. Markwell, & M.D. Miars, An Integrated Simulation Environment for Parallel and Distributed System Prototyping, *Simulation*, 75(5), 1999, 283-294.
- [16] D.P. Bhandarkar, Analysis of Memory Interference in Multiprocessors, *IEEE Transactions on Computers*, 24(9), 1975, 897-908.
- [17] T. McMahon, G. Henley, S. Hebert, B. Protopopov, R. Dimitrov, & A. Skjellum, *MCP and MPI for Myrinet on Sun and Windows NT*, [HTTP://www.erc.msstate.edu/labs/icdcr/myrimpi/multi\\_myrinet\\_html](http://www.erc.msstate.edu/labs/icdcr/myrimpi/multi_myrinet_html), 1997.

## **Biographies**

*Alan D. George* is an Associate Professor of Electrical and Computer Engineering at the University of Florida, and Director & Founder of the HCS Research Lab. He received the BS degree in Computer Science and the MS in Computer-Electrical Engineering from the Univ. of Central Florida, and the Ph.D. in Computer Science from Florida State University. Dr. George's research interests are in high-performance networks and architectures for parallel, distributed, and fault-tolerant systems and applications.

*Roger A. VanLoon* received a double-major BS degree in Computer Science and Electrical Engineering from Florida State University in 1997 and an MS degree in Electrical and Computer Engineering from the University of Florida in 1999. His research interests include high-performance computer networks, modelling and simulation, and system administration. He is currently working for the Exxon Upstream Technical Computing Company in Houston, TX.

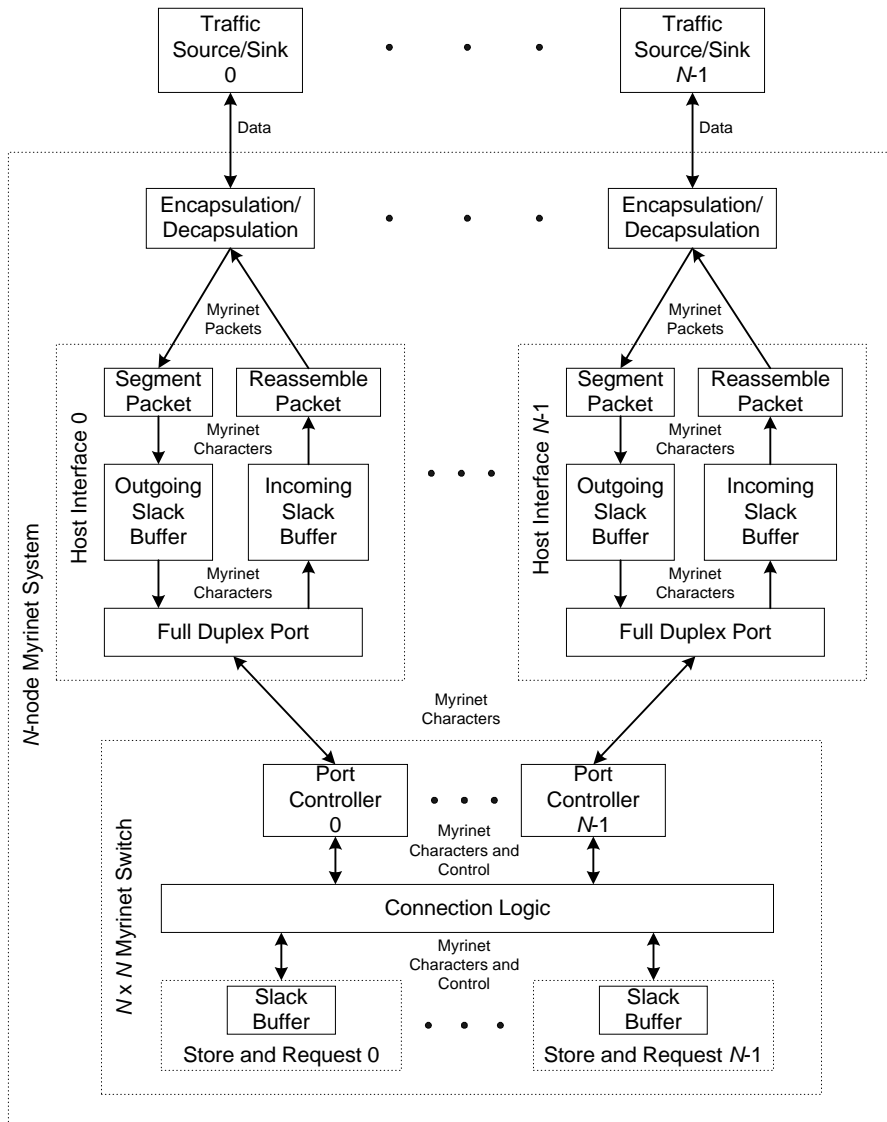


Figure 1. High-level flow diagram for the Myrinet discrete-event simulation model.

Table 1  
Parameter definitions for the Myrinet analytical model

Parameter	Definition
Crossbar Occupancy (C)	Ratio of available crossbar bandwidth
Length of Link (L)	Length of Myrinet cables in meters
Link Data Rate (R)	Network data rate in bits per second (b/s)
Packet Payload Size (P)	Number of data bytes in a packet (excluding overhead bytes)
Propagation Velocity (V)	Velocity of signal propagation through Myrinet cables in m/s

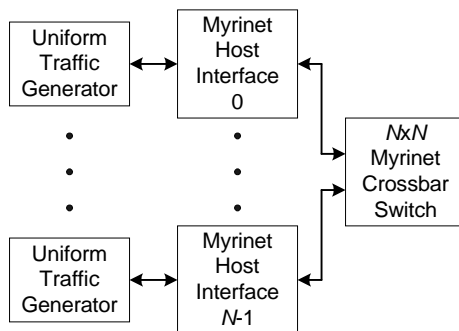


Figure 2. Topology for the verification experiments.

Table 2  
Parameter values for the verification experiments

Parameter	2-Node Value	8-Node Value
<i>BEAT Time</i>	1 second	1 second
<i>Crossbar Occupancy</i>	1.0	0.66
<i>High-water Mark</i>	56 bytes	56 bytes
<i>Length of Link</i>	10 meters	10 meters
<i>Link Data Rate</i>	1.28 Gb/s	1.28 Gb/s
<i>Low-water Mark</i>	40 bytes	40 bytes
<i>Offered Load</i>	160 Mb/s, 1.44 Gb/s	160 Mb/s, 1.44 Gb/s
<i>Packet Payload Size</i>	4 bytes - 8 kB	4 bytes - 8 kB
<i>Propagation Velocity</i>	1.8e8 m/s	1.8e8 m/s
<i>Slack Buffer Size</i>	96 bytes	96 bytes
<i>Timeout Interval</i>	1 second	1 second

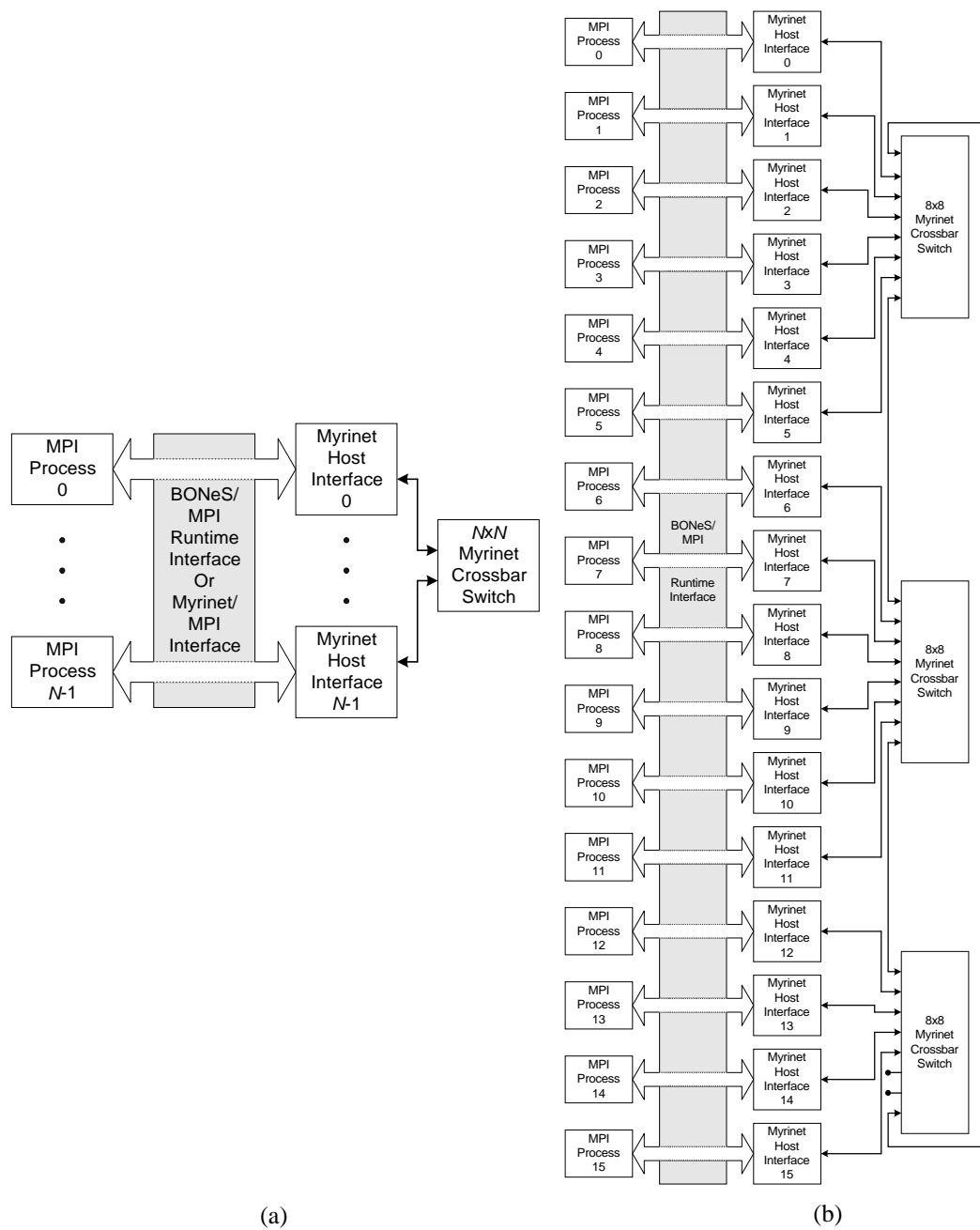
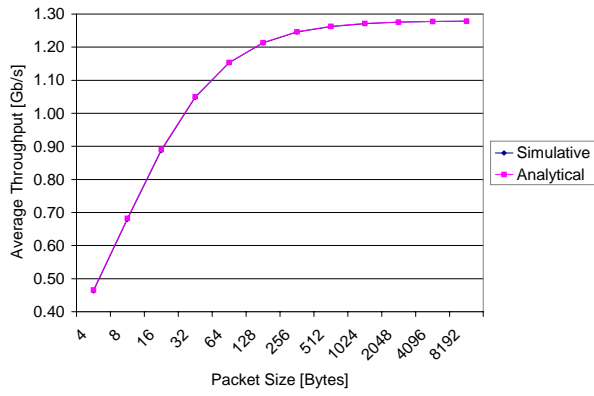
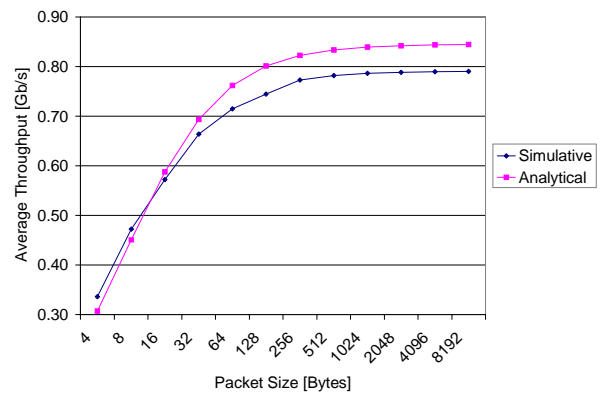


Figure 3. Topologies for the validation experiments: (a) 2-, 4-, or 8-node; (b) 16-node.

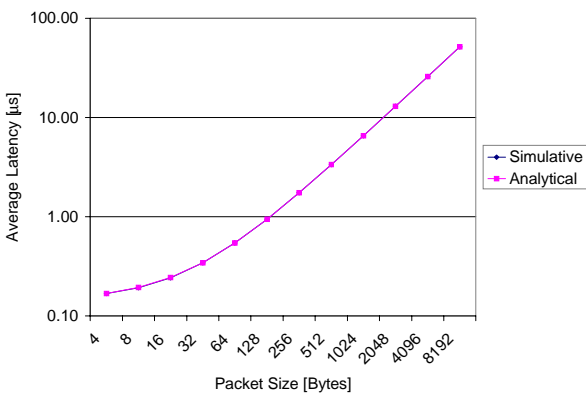


(a)

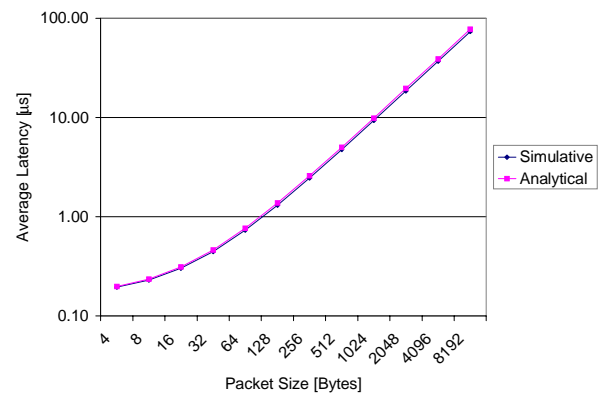


(b)

Figure 4. Throughput results from analytical and simulative models: (a) 2-node; (b) 8-node.

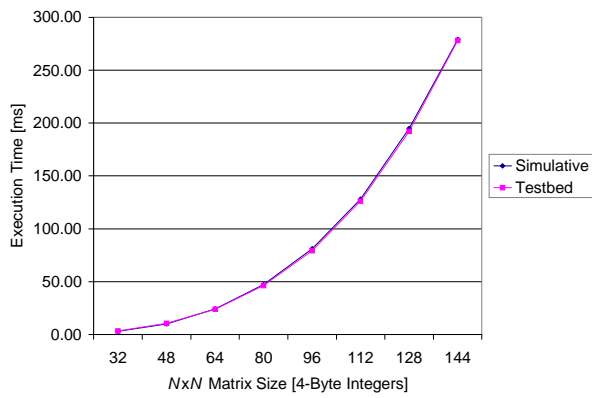


(a)

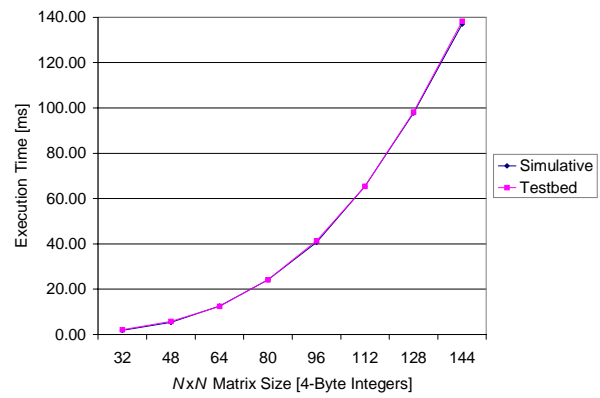


(b)

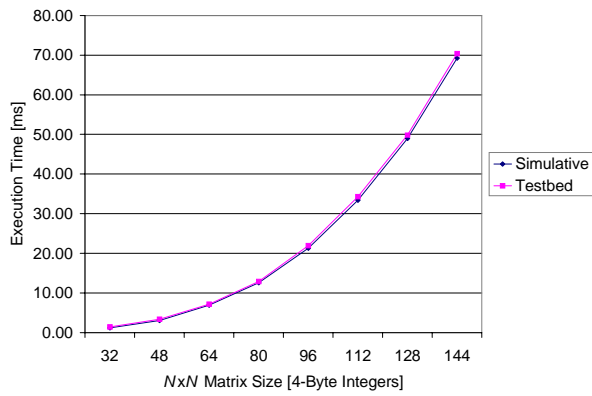
Figure 5. Latency results from analytical and simulative models: (a) 2-node; (b) 8-node.



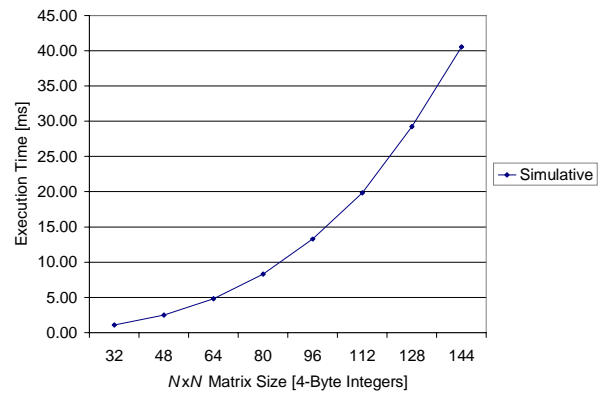
(a)



(b)



(c)



(d)

Figure 6. Execution time results from simulative model and testbed: (a) 2-node; (b) 4-node; (c) 8-node; and (d) 16-node.

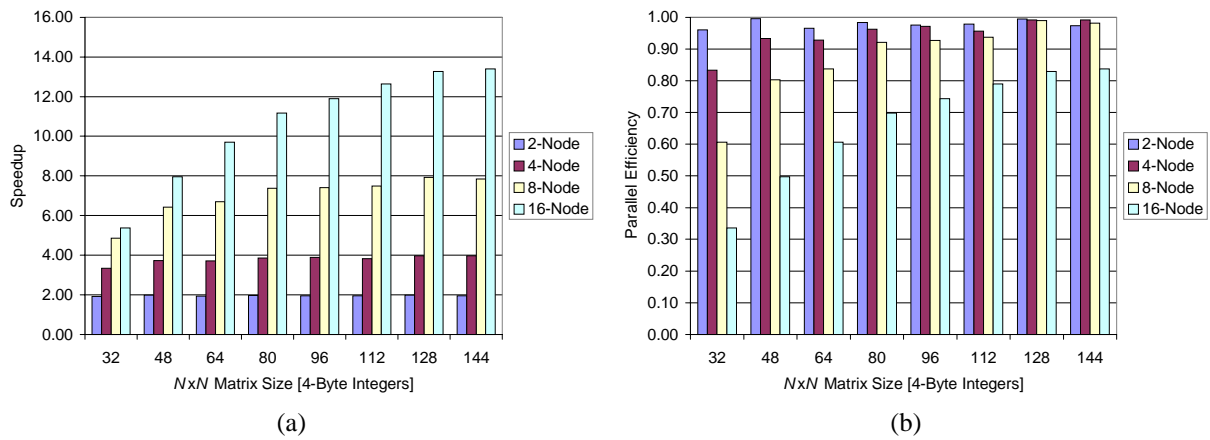


Figure 7. Speedup and parallel efficiency results from simulative model: (a) speedup; (b) parallel efficiency.

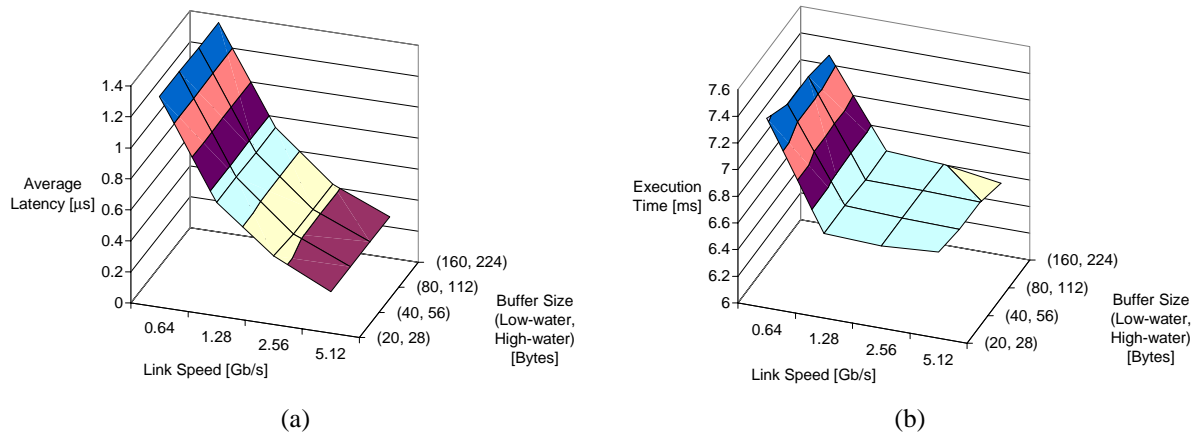


Figure 8. Latency and execution time results from simulative model: (a) latency; (b) execution time.