# ANALYTICAL MODEL AND PERFORMANCE ANALYSIS OF A NETWORK INTERFACE CARD

Naveen Cherukuri[1], Gokul B. Kandiraju[2], Natarajan Gautam[3],♣ and Anand Sivasubramaniam[4]

## Abstract

One of the key concerns for practitioners and academicians is that there are almost no platforms based on analytical models for testing the impact of various architectural and design modifications for intelligent Network Interface Cards (NICs). Simulations are typically time-consuming, especially for experimenting different scenarios and what-if analysis. In this research, we study the performance of a NIC called Myrinet developed by Myricom. We develop an open queueing network model to predict its performance. We compare the analytical results with the simulations.

The reason there are very few analytical models is because of the enormous complexity posed by the performance-analysis problem. In particular, the problem is a combination of: (a) multi-class queueing network with class switching, (b) polling system with limited service discipline, and (c) finite-capacity queues with blocking. The above three issues have been treated only in isolation in the literature. However the problem becomes much harder when all three issues are simultaneously present.

One of the key contributions of this paper is an analytical approximation of this complex system. From an analytical modeling standpoint, we observe that making simplifying assumptions to analyze nodes that are not bottlenecks does not impact performance greatly. The main findings of this research are the bottlenecks of the queueing network, utilizations of the various nodes and performance measures such as the expected delay. The model as well as findings can be used to test the performance impact of various enhancements to the operation of NICs.

**Keywords**: Network Interface Cards, Performance Analysis, Queueing Model, Simulation

[1] 82 Devonshire Street, #V7B, Boston MA 02109, naveen_ch@hotmail.com

[2] Dept. of Computer Science and Engineering, Penn State Univ., University Park, PA 16802 kandiraj@cse.psu.edu

[3] Dept. of Industrial Engr., 310 Leonhard Bldg, Penn State Univ., University Park, PA 16802 ngautam@psu.edu

♣ Corresponding Author

[4] Dept. of Computer Science and Engineering, Penn State Univ., University Park, PA 16802 anand@cse.psu.edu

## 1. Introduction

Distributed applications require rapid and reliable exchange of information across a network to synchronize operations and/or to share data. The performance and scalability of these applications depend upon an efficient communication facility. In order to connect computers to a network for facilitating communication, a network interface card (NIC) is necessary. The NIC is a computer circuit board or card that is installed in a computer. Personal computers and workstations on a local area network (LAN) typically contain a NIC specifically designed for the LAN transmission technology, such as Ethernet or token ring. NICs are also used to interconnect clusters of computers or workstations such that the cluster can be used for high performance or massively parallel computations. Although clusters are slowly replacing large supercomputers due to their low cost, one of the biggest stumble blocks for clusters to reach the performance of supercomputers is that their NICs are inefficient.

To address the inefficiencies of the NICs, three developments have been considered: (i) Using a processor on the NIC and thereby making the NIC more 'intelligent'. These second-generation intelligent NICs (including Myrinet [8], Fore Systems SA-200 [18], Giganet's cLan [19], etc.) outperform traditional NICs (conventional Ethernet NICs). Myrinet is one of the most popular second generation NICs in use today. (ii) Moving the network interface much closer to the application (called Virtual Interface Architecture (VIA) [1]). The VIA uses a virtual interface mechanism to transfer the most common messages directly between memory and the NIC. The result is a substantial reduction in processing overhead along the communication paths that are critical to performance. (iii) Removing the operating system from the critical path of communication, User-Level Networking (ULN) provides the user with the direct access to the NIC. The operating system is used only to setup the protected channels which can be accessed later during communication without the costs of crossing protection boundaries.

In this paper we study such intelligent VIA NICs that use ULN. Due to the rapid growth of ULN as a high-performance cost-effective solution with clusters, industry has also taken note of ULNs potential, and attempted to standardize it in the form of a VIA specification [1]. This specification was released by a group of companies together called the Virtual Interface Architecture Consortium (which includes Microsoft, Compaq, Intel). Hardware and software [20, 21, 22] implementations of VIA have also been developed and VIA is receiving a lot of attention both from the industry front as well as academia. Hence, research is under way to further improve the efficiency of the NICs. However there is a need for analytical models for NICs that the researchers could use to quickly test design alternatives. In this paper, analytical models are derived for the Myrinet [8] NIC, keeping in mind that, similar models can be derived for other network interfaces as well.

The literature on performance modeling for intelligent NICs is fairly limited. In [23], the NIC throughput is computed using an average case analysis by modeling the entire system at a much macro-granularity level than what is considered here. In fact [23] does not consider any details of the Myrinet NIC. The Myrinet NIC and software system in [24] is modeled at a meso-granularity level (i.e., in between the macro-granularity level in [23] and the micro-granularity level we have considered here) using a system of 2 queues such that the processor polls between queues at the software and the hardware layers, thereby incurring time during the context switch. However, since the entire NIC is modeled as a single-server queue in [24], it is not clear how various designs inside the NIC can be evaluated. The motivation for considering a micro-granularity level for modeling the Myrinet NIC in this paper is to test various NIC designs and evaluate the performance improvement across the NIC. In particular, a multi-station and multi-class open queueing network model is considered to capture the multitude of operations and queues inside the NIC. The complexity of the problem is due to the fact that it is a combination of (a) multi-class queueing network with class switching, (b) polling system with limited service discipline, and (c) finite-capacity queues with blocking. However by identifying the bottleneck node and modeling it accurately, and using approximations for the rest of the nodes, we are able to obtain system performance.

Networks of queues have proven to be useful models to analyze the performance of complex systems such as computer systems, switches, routers and communications networks [2-5]. This method has contributed to significant design decisions for performance improvements in various computer and communication systems. Analytical models are fast in obtaining the results and cost effective in implementation when compared to a simulation model or an experiment. Another important advantage of analysis using a network of queues is the flexibility to choose a wide range of operating parameters and obtain the performance measures with little effort. Simulations are developed for benchmarking and the results are compared with the performance measures obtained in the analytical model.

The rest of this paper is structured as follows. Section 2 deals with some preliminaries including a description of the VIA NICs as well as some results from queueing networks. Section 3 provides a detailed explanation of the various modeling aspects and the simplifications used in obtaining the analytical model. In Section 4, two simulation studies performed using a commercial package ARENA [6] are presented to justify the approximations. In Section 5, performance of the analytical model is compared against simulation studies. This paper concludes with highlights of the summary of the study, contributions from this research, and recommendations for future work in Section 6.

## 2. Preliminaries

We first present a description of the Myrinet NIC and then describe some known results from queueing networks. They constitute some of the preliminaries that are necessary in order to describe this research.

### 2.1 The Myrinet NIC

Figure 1 [7] shows a Myrinet NIC. Myrinet is popular for deploying clusters because it provides high hardware transmission rates. The several hardware features that it provides make VIA implementations more efficient.

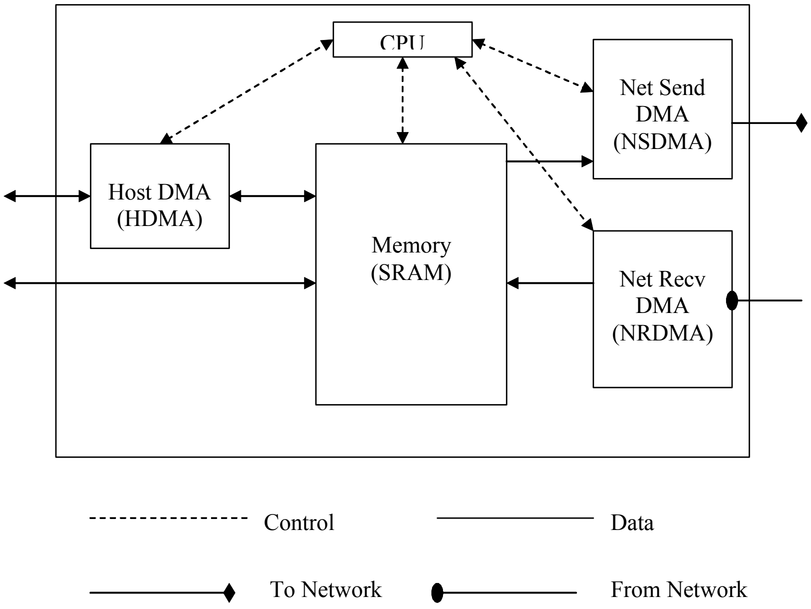

Figure 1: Network Interface Card

The NIC contains a processor called LANai, a Direct Memory Access (DMA) engine (represented as HDMA) which is used to transfer the data between the host memory and card buffer (SRAM), a DMA engine (represented as NSDMA) to transfer the data from SRAM onto the network, and another DMA engine (represented as RSDMA) to transfer data on to the SRAM from the network. From a modeling point of view, sending can be translated to appending a message to a queue in the card buffer and receiving can be translated to removing a message from the card buffer. A NIC provides an electro-mechanical attachment of a computer to a network. Under program control, a NIC copies data from memory to the network medium, transmission, and from the medium to memory, reception, and implements a unique destination for messages traversing the network. Myrinet NIC interface comprises of various physical components. A description of these components along with the critical data movements is provided in detail below.

**Doorbells**: Send or receive notification to NIC by application processes is done in VIA by a mechanism called *doorbells.* It is through a doorbell by which the NIC knows that there is work placed in the work queue. There are two sets of doorbells, one each for send and receive. When an application wants to send or receive a message, it creates a header for it (called a *descriptor*), makes it accessible to the NIC, and then rings a doorbell.

**Descriptors**: A Descriptor is a data structure recognized by the NIC that describes a data movement request. It is organized as a list of segments. A Descriptor is comprised of a control segment followed by an optional address segment and an arbitrary number of data segments. The data segments describe a communication buffer gather or scatter list for a NIC data movement operation. Descriptors contain all the information needed to process a request, such as the type of transfer to make, the status of the transfer and the queue information.

**Direct Memory Access**: There are situations in which data must be moved very rapidly to or from a device. Interrupt processing of each data transfer would be awkward and slow. With all of the bookkeeping involved in handling and interrupt, data would probably be lost. Direct Memory Access, or DMA, solves this problem. It is a method for direct communication from peripheral to memory with no programming involved. DMA reduces CPU overhead by providing a mechanism for data transfers that do not require monitoring by the CPU.  The data is moved to memory via the bus, without program intervention.

**LANai and SRAM**: A Myrinet host interface consists of two major components: the LANai chip and its associated SRAM memory. The LANai is a processor chip that controls the data transfer between the host and the network. Besides controlling the data transfer, the LANai is also responsible for automatic network mapping and monitoring the networks status. SRAM is a precious resource that hosts many queues. The size of the onboard SRAM ranges from 512 KB to 4 MB. The LANai communicates with the host's device drivers or user-level libraries through work queues residing in the SRAM.

**HDMA**: This is one of the most important entities of a NIC. Once a doorbell is detected, LANai processes the descriptor and then the corresponding data buffer. Allocating space for too many descriptors can be a waste of precious NIC buffer SRAM. The descriptors are thus kept on the host memory, and HDMA transfers them to the card buffer. Then LANai examines the descriptor and in the case of send, the HDMA transfers the corresponding data on to SRAM. In the case of receive, the direction of transfer is reversed, i.e., the data is transferred from SRAM into the host memory.

**NSDMA**: NSDMA (Network Send DMA engine) is a DMA engine in the Myrinet NIC that facilitates the data transfer from SRAM on to the network. If NSDMA is idle and there is data on SRAM queued up to be sent on to the network, LANai programs NSDMA to pick it up from the queue (FIFO basis). Then the data goes through the network bus to the peripheral destination in the network.

**NRDMA**: NRDMA (Network Receive DMA engine) is a DMA engine in the Myrinet NIC that facilitates the data transfer from network on to SRAM. LANai programs NRDMA to pick up a packet from the network. The destination id of an incoming message is extracted using this DMA engine and the receive descriptors are checked for a match. If there is a matching descriptor, then the data transfer up to the host can be initiated using HDMA (depending on the availability). Else, a receive descriptor needs to be brought down before the data can be transferred.

**Myrinet Control Program (MCP)**: The MCP (Myrinet Control Program) is the program that runs on the LANai chip on the host interface board. It is the MCP's job to transfer messages between the host and the network. LANai initiates the following operations that are needed to be performed by MCP.

- Poll doorbell queues for an application's send/receive notification
- Transfer the descriptor associated with the doorbell from the host memory down onto the SRAM using HDMA
- Transfer the data associated with a send descriptor from host memory to SRAM using HDMA.
- Transfer the packet out onto the network using NSDMA.
- Pick up packet from the network using NRDMA.
- Transfer data from SRAM to the host memory using HDMA.
- Transfer completion information (of send/receive) to host memory using HDMA.

**Sequence of Operations**: LANai goes through these operations cyclically: Polling the doorbell queue, polling the descriptor queue on SRAM and polling the data queue. In addition, it programs NSDMA and NRDMA to send and receive the data to and from the network respectively. LANai polls the doorbell queue and makes them available for HDMA to obtain the corresponding descriptors. Polled doorbells wait in a queue at HDMA to get serviced on a FCFS basis. They are processed by HDMA and the corresponding descriptors are stored in the descriptor queue on SRAM. The descriptors in this queue are polled by LANai and it makes them available for HDMA to obtain the corresponding data. In the case of a send descriptor, LANai initiates the transfer of data from the host memory on to the data queue on SRAM using HDMA. In the case of a receive descriptor, LANai initiates the transfer of data (if any) from the network queue at NRDMA to the data queue on SRAM using NRDMA. LANai polls the data queue and if the polled data is of type "send", it checks whether NSDMA is busy. If not, it initiates the transfer of send data from SRAM data queue to NSDMA. If the polled data is of type "receive", it initiates the transfer of data from SRAM data queue to host memory using HDMA.

## 2.2 The Queueing Network Analyzer

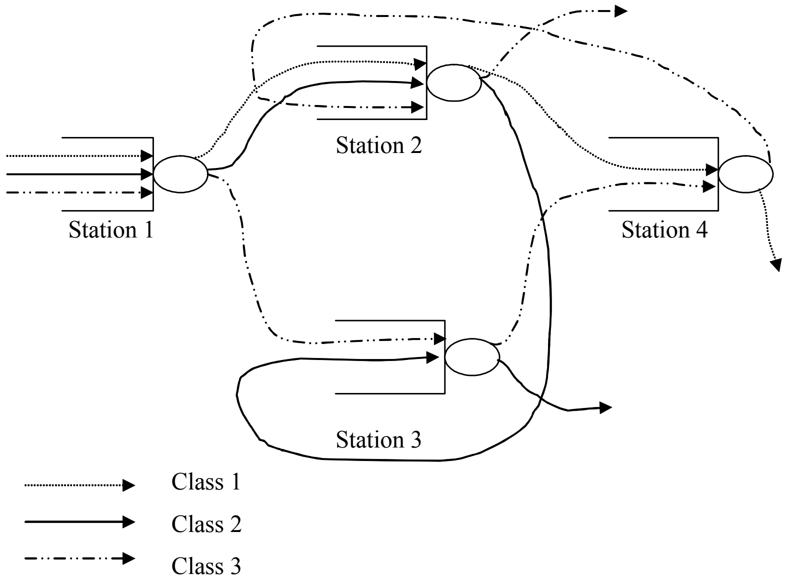

Figure 2: An example of a Multi-Station and Multi-Class Open Queueing Network

We now recapitulate some of the results from the multi-station and multi-class open queueing network using the QNA approach by Whitt [9,10]. Figure 2 is an example of a multi-station and multi-class open queueing network with three stations and four classes of traffic. It is provided to give a pictorial display for a multi-station and multi-class open queueing network. The following description forms the problem setting for the algorithm.

(1) There are N service stations (nodes) in the open queueing network. The outside world is denoted by node 0 and the others 1,2…N.

(2) There are $m_i$ servers at node i ($1 \leq m_i \leq \infty$), $1 \leq i \leq N$.

(3) The network has multiple classes of traffic, and class switching is not allowed.

(4) Service times of class r customers at node i are independent and identically distributed (iid) with mean $1/\mu_{i,r}$ and squared coefficient of variation (SCOV) $C^2_{S_{i,r}}$ .

(5) The service discipline is First Come First Served (FCFS).

(6) There is infinite waiting room at each node.

(7) Externally, customers of class r arrive at node *i* according to a general inter-arrival time distribution with mean $1 / \lambda_{0i,r}$ and SCOV $C^2_{A_{i,r}}$ .

(8) When a customer of class r completes service at node i, he or she or it joins the queue at node j ($j \in [0, N]$) with probability $p_{ij,r}$.

(9) Utilization of node i is the ratio of mean arrival rate at node i to the maximum possible service rate at node i. Being a ratio, it has no units.

### 2.2.1 Notation

The notation that is given here follows [11] and will be utilized for the decomposition algorithm to be presented later in this section.

R : Total number of classes.

$\lambda_{ij,r}$ : Mean arrival rate from node i to node j of class r.

$\lambda_{0i,r}$ : Mean arrival rate to node i of class r (or mean departure rate from node of class r)

$p_{ij,r}$ : Fraction of traffic of class r that exit node i and join node j.

$\lambda_i$ : Mean arrival rate to node i.

$\rho_{i,r}$ : Utilization of node i due to customers of class r.

$\rho_i$ : Utilization of node i.

$\mu_i$ : Mean service rate of node i.

The next five symbols are used to denote the squared coefficients of variation (SCOV) of different parameters.

$C^2_{A_{i,r}}$ : SCOV of class r inter arrival times into node i.

$C^2_{A_i}$ : SCOV of arrival times into node i.

$C^2_{D_i}$ : SCOV of inter departure times from node i.

$C^2_{S_i}$ : SCOV of service time of node i.

$C^2_{ij,r}$ : SCOV of time between two customers going from node i to node j.

### 2.2.2 Decomposition Algorithm

The network is broken down into individual nodes, and analysis is performed on each node as an independent GI/G/$m_i$ queue with $m_i$ servers in station i and with multiple classes. The required parameters are mean arrival and service rates as well as SCOV of the interarrival and service times. Obtaining these will be hard when multiple streams are merged (superposition) or when traffic flows through a node (flow) or when a single stream is forked into multiple streams (splitting). The algorithm supposes that just before entering a queue, superposition takes place which results in one stream. Likewise, it assumes that there is only one stream that gets split into multiple streams [12, 13].

There are 3 basic steps in the decomposition algorithm.

*Step 1:* Mean arrival rates, utilizations and aggregate service rate parameters are calculated using the given data in the following way.

$$\lambda_{ij,r} = \lambda_{i,r}\, p_{ij,r} \tag{1}$$

$$\lambda_{i,r} = \lambda_{0i,r} + \sum_{j=1}^{N} \lambda_{j,r}\, p_{ji,r} \tag{2}$$

$$\lambda_i = \sum_{r=1}^{R} \lambda_{i,r} \tag{3}$$

$$\rho_{i,r} = \frac{\lambda_{i,r}}{m_i \mu_{i,r}} \tag{4}$$

$$\rho_i = \sum_{r=1}^{R} \rho_{i,r} \quad \text{The condition for stability is } \rho_i < 1 \ \forall\, i \tag{5}$$

$$\mu_i = \frac{1}{\sum_{r=1}^{R} \frac{\lambda_{i,r}}{\lambda_i} \frac{1}{m_i \mu_{i,r}}} = \frac{\lambda_i}{\rho_i} \tag{6}$$

$$C^2_{S_i} = -1 + \sum_{r=1}^{R} \frac{\lambda_{i,r}}{\lambda_i} \left( \frac{\mu_i}{m_i \mu_{i,r}} \right)^2 \left( C^2_{S_{i,r}} + 1 \right) \tag{7}$$

*Step 2:* The coefficient of variation of inter-arrival times at each node is calculated iteratively by initializing $C^2_{ij,r}$ and performing superposition, flow and splitting cyclically.

(i) Superposition:

$$\mathrm{C}^2_{A_{i,r}} = \frac{1}{\lambda_{i,r}} \sum_{j=0}^{N} C^2_{ji,r} \lambda_{j,r} p_{ji,r} \tag{8}$$

$$\mathrm{C}^2_{A_i} = \frac{1}{\lambda_i} \sum_{r=1}^{R} C^2_{A_{i,r}} \lambda_{i,r} \tag{9}$$

(ii) Flow

$$\mathrm{C}^2_{D_i} = 1 + \frac{\rho_i^2 \left( C^2_{S_i} - 1 \right)}{\sqrt{m_i}} + \left( 1 - \rho_i^2 \right)\left( C^2_{A_i} - 1 \right) \tag{10}$$

(iii) Splitting

$$\mathrm{C}^2_{ij,r} = 1 + p_{ij,r}(\mathrm{C}^2_{D_i} - 1) \tag{11}$$

The splitting formula is exact if the departure process is a renewal process. The expressions for flow and superposition are approximations.

*Step 3:* Treating each queue independently, performance measures are obtained as follows

Choose $\alpha_{m_i}$ such that $\alpha_{m_i} = \frac{\rho_i^{m_i} + \rho_i}{2}$ if $\rho_i > 0.7$ or $\rho_i^{\frac{m_i+1}{2}}$ if $\rho_i < 0.7$

The mean waiting time for class r customers in the queue (not including service time) at node i is approximately

$$W_{iq} \approx \frac{\alpha_{m_i}}{\mu_i}\left(\frac{1}{1-\rho_i}\right)\left(\frac{C_{A_i}^2 + C_{S_i}^2}{2m_i}\right) \tag{12}$$

The mean waiting time for class r customers at node i (including service time) is given by

$$W_i = W_{iq} + \frac{1}{\mu_i} \tag{13}$$

The mean queue length for class r customers at node i (without customers in service) is given by

$$L_q = W_{iq} * \lambda_i \tag{14}$$

The mean number of customers at node i (including the customers in service) is given by

$$L = W_i * \lambda_i \tag{15}$$

The performance measures presented in Section 3 are mean queue length ($L_q$) and utilization $(\rho_i)$ at each node. Note that both are numbers and do not have units.

## 3. Analytical Model

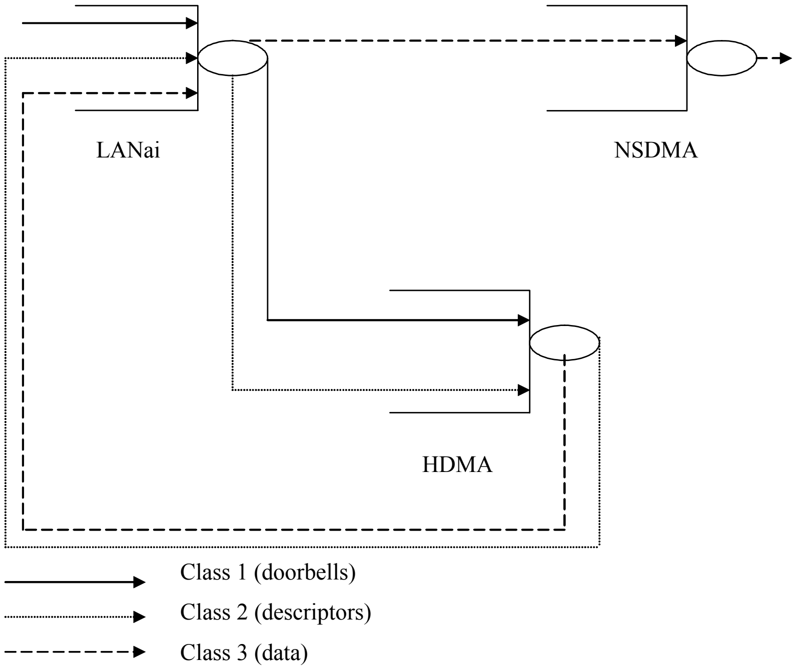

Figure 3: Three-station and three-class open queueing network model for NIC

We model the performance of the NIC of a send station, where the data flow is from the station to the network. In essence, the hardware features of NIC consisting of only NSDMA are modeled in an open queueing network. The analysis provided for multi-station and multi-class open queueing network in Section 2.2 is used to obtain various performance measures. This simplified version of NIC consists of send doorbell, send descriptor and send data as three classes of traffic in the network. The servers in the network correspond to LANai, HDMA and NSDMA and henceforth will be referred to as nodes. Figure 3 shows a three-station and three-class open queueing network model for the performance analysis of NIC.

### 3.1 Description and Analysis

The most critical assumption that is required for the analysis in Section 2.2 is that class switching is not allowed. Strictly speaking, there is a class switching here in the form of Class1 to Class2 and Class2 to Class3. Doorbells are processed by HDMA to produce descriptors, and descriptors in turn are processed by HDMA to obtain data. Both the processes involve class switching: Doorbells to Descriptors and Descriptors to Data. If the performance results can be confirmed with simulations, it is a great gain in terms of applicability of the analytical model. Deviations can be investigated and one of the factors to which they can be attributed will be class switching.

However the other assumptions made for the queueing network analysis in Section 2.2 are valid, viz.

(1) Service times of class r customers are independent and identically distributed (iid).

(2) The service discipline is FCFS, which is the same for the NIC queues.

(3) There is infinite waiting room at each node.

The waiting room at each node can be translated to the memory space available for each type of message on the NIC, and in reality there is a limitation. One design consideration is that there are always sufficient doorbells to accommodate the data transfer to make sure that no data is practically dropped off. It is thus safe to assume that there is infinite waiting room at each node, which will serve the purpose the application of this generic design methodology to obtain the performance measures in a quicker and convenient way as opposed to lengthy and costly simulations. We now describe the nodes in detail.

Node 1 (LANai) serves all the classes of traffic, namely doorbells, descriptors and data. The service time for each class of traffic is given in the Table 2. All three classes are queued up in different queues in the NIC. LANai polls these queues in an order determined by the Myrinet Control Program (MCP). For the purpose of mathematical analysis, the physical location of the queue does not matter as long as the ordering of polling is consistent with the MCP. A survey of the existing literature on the polling models for any applicable models for deterministic times of service [14-16] shows that an analytical analysis would be extremely difficult for a polling model when the service times are deterministic. Moreover, it is an asymmetric polling system with feedback, infinite buffers and a finite switchover time, gated service and cyclic service order. All these conditions make the problem computationally intractable and shift the focus from the main aim of getting the performance measures of the NIC and validating them through extensive simulations. Instead, a different way of analytical modeling for node LANai is adopted to which the existing methods of analysis can be applied. The performance of LANai is mathematically approximated as follows. LANai is a node that serves a single queue, which contains three different classes of traffic. The traffic corresponding to different classes is thus pooled up into a single queue, and the queue is served on a First Come First Served (FCFS) basis. Here lies the basic assumption of the modeling: for the purpose of analytical analysis to obtain the performance measures under the existing parameters of operation, pooling up the class traffic is immaterial. This assumption can be verified with simulations. If all the other design is same, the following two simulation models can be compared to obtain the effect of pooling up the traffic of all classes into a single queue at node LANai.

(1) Simulation model 1 in which all the classes are queued up for LANai in the same queue.

(2) Simulation model 2 that has three separate queues for the three classes of traffic at node LANai.

The service times for different classes are given in the Table 2.

Node 2 (HDMA) serves the traffic comprising of Classes 1 and 2, namely, doorbells and descriptors. It models the operation of HDMA. Classes 1 and 2 arrive into an FCFS queue according to a general arrival process that is equivalent to the departure process from Node 1. HDMA service effectively transforms the class of messages. For example, HDMA service of a doorbell assigns a descriptor and places it in the SRAM, which in this model is Node 1. It is modeled as changing a Class 1 message in to a Class 2 message upon departure from Node 2. Thus, after service completion, Class 1 becomes a Class

2 and goes into SRAM (node 1). Likewise, Class 2 becomes Class 3 and resides in SRAM to be picked up by NSDMA. The service times are given in the Table 2. The service time for Class 2 messages is more than double the service time for Class 1 messages. Servicing messages of Class 1 in the hardware terms is equivalent to obtaining the descriptor contained in the doorbell. This descriptor provides the information about the data that is picked by the NIC and put on the network. The descriptor is processed by HDMA to get the memory location details, and the data is transferred onto the temporary memory to be picked up by NSDMA. This is the hardware level explanation for the analytical class switching from 2 to 3.

Node 3 (NSDMA) has a queue that receives Class 3 messages, namely, data. This node models the operation of the NSDMA. Class 2 messages after getting served at Node 2, which models the operation of HDMA, are put on the SRAM of the NIC. In case there are multiple data, they are queued up in the SRAM, and LANai processes them in FCFS basis. The hardware design of Myrinet NIC does not allow queueing of messages at NSDMA, i.e., the node is effectively a G/D/1/1 queue with blocking. As part of the Myrinet Control Program (MCP), LANai checks for the status of the NSDMA and the SRAM queue for any data available for transfer. The entire operation of putting the data on the network depends on the status of the NSDMA and the availability of the data to be transferred on to the network. In all of the scenarios explained below, data resides in the SRAM queue, and the data that is to be transferred onto the network is the first data that is in the queue.

Scenario 1: Data to be transferred on to the network is present in SRAM, and NSDMA is busy

Result of MCP: There are no arrivals to NSDMA while a service is in progress since LANai does not pass a message to it if the latter is still busy.

Scenario 2: Data to be transferred on to the network is present in SRAM, and NSDMA is idle.

Result of MCP: LANai passes a message to NSDMA by programming it to pick up the message from SRAM.

Scenario 3: Data to be transferred on to the network is not present in SRAM, and NSDMA is busy.

Result of MCP: Lanai does not pass any message and continues with the next operation under MCP.

Scenario 4: Data to be transferred on to the network is not present in SRAM, and NSDMA is idle.

Result of MCP: Nothing happens in this scenario and LANai proceeds to next operation under MCP. Table 1 summarizes all of the scenarios and the outcome for each of them.

| Scenario | Availability of Data | Status of NSDMA | Programming of NSDMA by LANai |
|---|---|---|---|
| 1 | Available | Busy | No |
| 2 | Available | Idle | Yes |
| 3 | Not Available | Busy | No |
| 4 | Not Available | Idle | No |

Table 1 Four different scenarios for programming of NSDMA by LANai

These scenarios have to be captured analytically, to be incorporated in our model. It is essentially a status checking process by LANai. The modeling of NSDMA thus depends on some approximations that are stated here. One of the important conditions that needs to be satisfied for any network to model it as a multi-station and multi-class open queueing network is the availability of infinite waiting space in the queues at each of its nodes. For analytical purposes, an attempt is made here to model NSDMA as a node that has infinite space.

The network is designed such that the service time for messages of Class 3 at Node 1 is an average value that takes into consideration of all the four scenarios explained above. In reality, LANai can program NSDMA only in Scenario 2, and the time that it takes to program NSDMA to pick up the data from SRAM is 10 microseconds. A way of getting around this problem is estimating the probability of the occurrence of Scenario 2 and multiplying it with the time that is required to program NSDMA (10 microseconds) and using the result as the service time for messages of Class 3 at Node 1. It will provide an average service time for servicing the traffic corresponding to Class 3 (data) by LANai. Each time LANai services data with an approximated service time, which will reflect the probability of programming of NSDMA by LANai. The next paragraph will propose and explain in detail the approximation that is used to achieve the result stated above.

Let $t$ be the service time for a data message at node NSDMA. Let $\lambda$ be the arrival rate of data traffic at NSDMA. These two values are available from Table 2. From the model, $\lambda$ is the arrival rate chosen for the doorbells. The service time $t$ (52.6887 microseconds) can be obtained from the Table 2. Therefore, probability that NSDMA is busy $\approx \lambda * t$ and probability that NSDMA is idle $\approx 1 - \lambda * t$. When NSDMA is idle, a message in the data queue on the SRAM might be present or might not be present. Assuming equal probability, $p$ = Probability that NSDMA is idle and a message is present in SRAM $\approx 0.5*(1 - \lambda * t)$

Note that this assumption is made only for the analytical model. This does not affect the simulations. Table 2 uses $p$ in the calculation of estimated service time of data messages at node LANai. The average service time for the data messages at LANai is approximately the product of probability that NSDMA is idle and a message is present in SRAM and the time that takes for LANai to program NSDMA. Therefore, service time for messages of type Class 3 at Node 1 $\approx 10*0.5*(1 - \lambda * t)$ microseconds and service rate for messages of type Class 3 at Node 1 $\approx 1 / (10*0.5*(1 - \lambda * t))$ microseconds. This value is used to compute the aggregate service rate at Node 1. Mean arrival rates and aggregate service rates are available from Table 2 at all of the nodes for the multi-station and multi-class open queueing network algorithm to calculate the utilizations at each node. Each class has a deterministic service time in each queue namely LANai, HDMA and NSDMA in accordance with the following table.

| Class | Node 1 (LANai) | Node 2 (HDMA) | Node 3 (NSDMA) |
|---|---|---|---|
| 1 (doorbells) | 22 | 21 | N/A |
| 2 (descriptors) | 0.12 | 68.3154 | N/A |
| 3 (data) | 10*$p$ | N/A | 52.6887 |

Table 2 Service times in microseconds measured on a Myrinet NIC

In Table 2, if a node does not serve traffic of a class, the term N/A (Not Applicable) is used to indicate so. All the values are in microseconds. The MCP program running on the NIC, cycles through all of the destinations, copies the message data to the NIC buffer, and sends the data in packets to all destinations in turn. In order to deal efficiently with variable size fragments, packets do not correspond to fragments, but are fixed size blocks containing segments of fragments. Fragments may span block boundaries, and a block can contain one or more segments, depending on fragment sizes. The receiving NIC in the cluster network reassembles these segments back into fragments and adds them to the message queue. To maintain the synchronization, the sender NIC's always emit packets, also when there is not enough data available for a particular destination, in which case, the block contains empty segments. Thus, each packet is a fixed size block containing segments of fragments. The values in Table 2 assume a block size of 4 Kbytes, and corresponding transfer times at the DMA engines are computed [17].

Table 2 provides details about service times of various classes of traffic at each node. Arrival rates for descriptors and data are the same as the arrival rate for doorbells. Various numerical values are chosen for the arrival rate of doorbells. The squared coefficient of variation (SCOV) for the service times at each node for each class is initialized in conjunction with the type of the service. In the case of deterministic service times as the present analytical model for NIC, all the values $C^2_{S_{i,r}}$ for i =1, 2 and 3 and r = 1, 2 and 3 are initialized to zero. All these values are used in the decomposition algorithm in Section 2.2 and performance measures at each node are obtained.

### 3.2 Summary of the analytical model

Section 3.1 contained detailed description of the analytical model proposed for obtaining the performance of NIC. Polling by LANai, programming of NSDMA by LANai, class switching from doorbells to descriptors and from descriptors to data make it impossible to use the decomposition algorithm described in Section 2.2. There are several assumptions and simplifications made in arriving at the analytical model for modeling the performance of the NIC. All of the simplifications are predicted to be observed and need to be verified with extensive simulations. The most notable simplifications are:

(1) Mathematical approximation for polling by LANai: In a NIC, LANai polls three queues, namely, doorbells, descriptors, and data. In the analytical model, LANai is a node with a single queue with multiple class traffic corresponding to doorbells, descriptors, and data. This approximation is required to be able to apply the decomposition algorithm described in Section 2.2.

(2) Mathematical approximation for programming of NSDMA by LANai: In a NIC, NSDMA is a node with zero waiting space and the service of data traffic at NSDMA depend on four scenarios outlined in Section 3.1. Analytically, NSDMA is modeled as a node with a single queue and infinite waiting space. This approximation is required to be able to apply the decomposition algorithm described in Section 2.2.

(3) Estimated probability for Scenario 2: An approximation is used in estimating the probability of Scenario 2, i.e., that there is data available in SRAM when NSDMA is idle. This approximation is needed to obtain an estimated service time for data traffic at node LANai and model NSDMA station as having a single queue with infinite waiting space. Analytical results will be compared with the simulation results, where in the analytical results we use $p$ = 0.5, which is not done in simulation.

(4) Class switching: There is a class switching involved in the network. After getting served at node HDMA, doorbells are converted to descriptors. Similarly, after getting served at node HDMA, descriptors are converted to data. The decomposition algorithm explained in Section 2.2 can be applied only when class switching is not present in the queueing network.

Section 3.3 discusses the numerical results to test the above simplifications and abstractions.

### 3.3. Numerical Results

In this section, a detailed description of numerical results using the proposed multi-station, multi-class open queueing network model are presented for the performance analysis of a VIA NIC. The results are computed for six different test cases. For each case, an arrival rate for the doorbells is chosen in such a way that the six cases comprise a range of traffic intensities. Once the arrival rate of doorbells is chosen, the arrival rate of descriptors and data will be the same as the arrival rate of doorbells because Classes 2 and 3 are essentially the transformed versions of Class 1 at Node 2. Let this arrival rate be λ. As explained in the previous section, an estimated probability of 0.5 is proposed for the probability of data being available when NSDMA is idle. Since the service times are deterministic, $C^2_{S_{i,r}}$ for i = 1,2 and 3 and r = 1,2 and 3 is zero. Figure 4 shows the utilizations of all the three nodes under various arrival rates. Note that the utilization values do not have units. Utilization of node i is the ratio of mean arrival rate at node i to the maximum possible service rate at node i. Utilizations are computed using (6) in Section 2.2. Utilization results provide valuable information about the message loads that each node is subjected to and are very useful in finding the bottlenecks in the system, which forms a crucial part of queueing network analysis. Utilizations of the nodes increase with the increase in arrival rate. From Figure 4, at any given arrival rate, HDMA is the node with the highest utilization, and LANai is the node with the lowest utilization. The utilization of HDMA linearly increases with the increase in arrival rate and will be 1 when the mean arrival rate at node HDMA is equal to the mean service rate of node HDMA. Compared to it, the utilizations of the other two nodes are considerably less. Thus, HDMA is the bottleneck node in the NIC. Hence significant improvements in the performance of NIC can be obtained by improving the performance of HDMA rather than by improving the performance of the other nodes. For the purpose of analytical approximations, it may be necessary to accurately model only the bottleneck node.

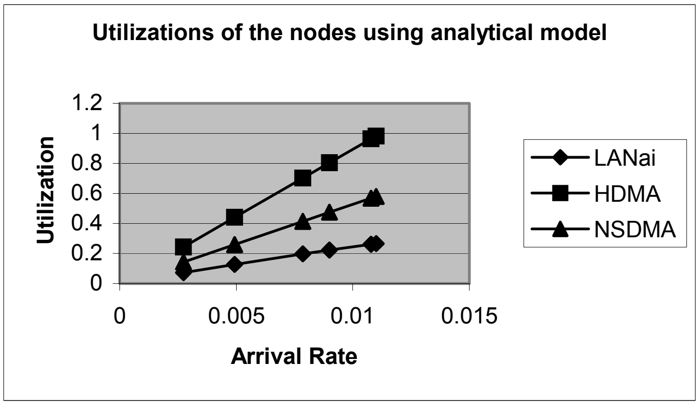

Figure 4: Utilizations of the nodes using the analytical model

| Arrival Rate λ | Queue Length at LANai | Queue Length at HDMA | Queue Length at NSDMA |
|---|---|---|---|
| 0.00273 | 0.0059 | 0.0480 | 0.0133 |
| 0.00493 | 0.0191 | 0.1922 | 0.0378 |
| 0.00786 | 0.0486 | 0.8007 | 0.1006 |
| 0.00900 | 0.0642 | 1.5285 | 0.1384 |
| 0.01079 | 0.0940 | 11.2929 | 0.2250 |
| 0.01100 | 0.0980 | 24.1981 | 0.2383 |

Table 3 Mean queue lengths at the nodes using the analytical model

Based on the multi-station and multi-class open queueing network analysis in Section 2.2, Table 3 provides the mean queue lengths at the three nodes for different arrival rates. Mean queue lengths are computed using (14) in Section 2.2. Note that mean queue length is a number and does not have units. Mean queue lengths of the nodes increase with the increase in the arrival rate. From Table 3, the length of the queue for HDMA is increasing rapidly with the increase in the arrival rate and would approach infinity as the queue becomes unstable at HDMA with the increasing arrival rate. The percentage of increase in the queue length of HDMA is very large compared to the other nodes since it is the bottleneck node in the network. It makes sense because HDMA is the node with the highest utilization in the network. The queue lengths at LANai and NSDMA are comparatively insignificant. Mean queue lengths depend upon the mean service times and the mean arrival rate at each node and the assumptions summarized in Section 3.2 are not the reasons for the insignificance of mean queue lengths of LANai and NSDMA.

## 4. Simulations

This section deals with the simulation models that are developed to validate the analytical model. The simulations are performed using a commercial-off-the-shelf simulation software package ARENA [6]. It is crucial to note that two different simulation models are designed to mimic the performance of a NIC. Both simulation models are designed with zero waiting space at the NSDMA server (i.e., they permit blocking and LANai checks). The first simulation model depicts the architecture of a NIC in which all the messages are queued up in a single queue for LANai and it serves the messages in the order of arrival (FCFS). This is identical to the scenario in the analytical model with respect to the operation of LANai. The main purpose is to verify the accuracy of the analytical model. The second simulation model depicts the architecture of a NIC in which LANai polls different queues corresponding to various components of NIC. This is a software simulation of the exact NIC operation, so that the analytical model performance can use this as a benchmark for comparison. The performance measures (mean queue length and utilization at each node) for both the simulation models are obtained and compared for accuracy. The assumption involved in the analytical model that the pooling-up of the messages in a single queue for LANai is a simplification for the actual Myrinet NIC is checked with the simulations. We first present the two simulation models (Sections 4.1 and 4.2) and then compare the models (Section 4.3).

## 4.1 First Simulation Model

The model has 3 nodes, one corresponding to each of the LANai, HDMA and NSDMA. The three classes of traffic are doorbells, descriptors, and data. LANai is modeled as a node that serves all the three different types of messages pooled up in to a single queue with infinite waiting space. HDMA is modeled as a node that serves a queue that consists of two types of messages: doorbells and descriptors. The queue has infinite waiting space. The performance of NSDMA is modeled in the exact way as in the NIC. The NSDMA node processes messages of type data but does not serve them from a queue. If NSDMA is free and if there is a message of type data available, LANai programs NSDMA to pick up the data and the service for that particular message will be started at node NSDMA. A check for the availability of the resource NSDMA is placed in the simulation model to detect this scenario. If NSDMA is busy, the message will wait in the queue for LANai. This first simulation model consists of one mathematical approximation that is used in the analytical model, i.e., LANai serves messages from a single queue into which all three types of messages are pooled-up. Referring to Section 3.3 regarding the assumptions made in the analytical model, modeling the performance of NSDMA in exact way as in NIC removes the necessity of using the estimated probability 0.5.

## 4.2 Second Simulation Model

This simulation model captures the exact behavior of the NIC. In addition to the first simulation model discussed in Section 4.1, it removes the mathematical approximation for LANai that it serves a single queue, which contains multiple class traffic. Instead it uses three different queues: one for each of doorbells, descriptors, and data. Node LANai now serves these queues in cyclic order and within each queue; the order of service is FCFS. The service times are same as that are given in Table 2. The second simulation model is the only model among the three that provides information about the lengths of queues on SRAM corresponding to doorbells, descriptors, and data (see Table 4).

| Arrival Rate $\lambda$ | Doorbell Queue Length | Descriptor Queue Length | Data Queue Length |
|---|---|---|---|
| 0.00273 | 0.0024 | 0.0022 | 0.0019 |
| 0.00493 | 0.0084 | 0.0076 | 0.0062 |
| 0.00786 | 0.0244 | 0.0216 | 0.0166 |
| 0.00900 | 0.0337 | 0.0293 | 0.0224 |
| 0.01079 | 0.0530 | 0.0444 | 0.0344 |
| 0.01100 | 0.0554 | 0.0464 | 0.0360 |

Table 4 Mean queue lengths at the nodes using the second simulation model

### 4.3 Comparison of the Two Simulation Models

The difference between both the simulation models is the modeling of performance of LANai. The first simulation model has LANai serving all types of messages pooled-up in to a single queue on FCFS basis. The second simulation model consists of three separate queues, one for each of the doorbells, descriptors and data. The two simulation models are compared in order to verify the assumption of pooling different SRAM queues in to a single queue. If the results show a match, modeling NIC analytically using a multi-station and multi-class open queueing network is justified. Table 5 compares the two simulation models with respect to the utilizations of the different nodes. The utilizations of HDMA and NSDMA are exactly the same in both simulation models. In the case of node LANai (the only node where the two simulation models differ), the utilizations are nearly same with the maximum error equal to 1.38% at the largest arrival rate among the six numerical values chosen. Mathematical approximation of node LANai practically did not affect the utilization values of the nodes in the network. To obtain the performance measures of the network, it is a very useful result that supports the assumption that for the given parameters of Myrinet NIC, LANai serving a single queue which has all the types of messages pooled up is equivalent to LANai serving different queue which are present in SRAM.

| Arrival Rate λ | Utilization of LANai in model 1 | Utilization of LANai in model 2 | Utilization of HDMA in model 1 | Utilization of HDMA in model 2 | Utilization of NSDMA in model 1 | Utilization of NSDMA in model 2 |
|---|---|---|---|---|---|---|
| 0.00273 | 0.0875 | 0.0875 | 0.2430 | 0.2430 | 0.1433 | 0.1433 |
| 0.00493 | 0.1588 | 0.1586 | 0.4393 | 0.4393 | 0.2591 | 0.2591 |
| 0.00786 | 0.2561 | 0.2551 | 0.7015 | 0.7015 | 0.4138 | 0.4138 |
| 0.00900 | 0.2954 | 0.2935 | 0.8032 | 0.8032 | 0.4738 | 0.4738 |
| 0.01079 | 0.3609 | 0.3564 | 0.9635 | 0.9637 | 0.5683 | 0.5683 |
| 0.01100 | 0.3690 | 0.3640 | 0.9822 | 0.9822 | 0.5794 | 0.5794 |

Table 5 Comparison of the two simulation models for the utilizations of nodes

| Arrival Rate λ | Length of HDMA Queue in model 1 | Length of HDMA Queue in model 2 |
|---|---|---|
| 0.00273 | 0.0464 | 0.0465 |
| 0.00493 | 0.1999 | 0.2002 |
| 0.00786 | 0.9433 | 0.9438 |
| 0.00900 | 1.8653 | 1.8653 |
| 0.01079 | 14.58 | 14.576 |
| 0.01100 | 30.506 | 30.499 |

Table 6 Comparison of the two simulation models for mean queue lengths of HDMA

| Arrival Rate λ | Queue Length at node LANai in model 1 | Sum of the lengths of three individual queues which LANai polls in model 2 |
|---|---|---|
| 0.00273 | 0.0067 | 0.0064 |
| 0.00493 | 0.0238 | 0.0222 |
| 0.00786 | 0.0680 | 0.0626 |
| 0.00900 | 0.0925 | 0.0854 |
| 0.01079 | 0.1413 | 0.1317 |
| 0.01100 | 0.1478 | 0.1378 |

Table 7 Comparison of mean queue lengths at LANai in the two simulation models

Table 6 and 7 compare the mean queue lengths of node HDMA and LANai respectively in the two simulation models. Table 7 shows a maximum difference of 6.76%. Strictly speaking, these values cannot be compared because the second simulation model does not have a single LANai queue. They are presented here for the completeness of comparison of the both simulation models. The most notable result that is obtained by comparing the two simulation models is that for the given parameters of NIC, LANai can be mathematically abstracted to be a node that polls a single queue in FCFS manner in to which all of the different kinds of messages (doorbells, descriptors and data) are pooled up.

## 5. Comparison of Analytical and Simulation Models

| Arrival Rate λ | Analytical Utilization of LANai | Simulated Utilization of LANai | Analytical Utilization of HDMA | Simulated Utilization of HDMA | Analytical Utilization of NSDMA | Simulated Utilization of NSDMA |
|---|---|---|---|---|---|---|
| 0.00273 | 0.0721 | 0.0875 | 0.2438 | 0.2430 | 0.1438 | 0.1433 |
| 0.00493 | 0.1273 | 0.1586 | 0.4403 | 0.4393 | 0.2597 | 0.2591 |
| 0.00786 | 0.1969 | 0.2551 | 0.7020 | 0.7015 | 0.4141 | 0.4138 |
| 0.00900 | 0.2227 | 0.2935 | 0.8039 | 0.8032 | 0.4742 | 0.4738 |
| 0.01079 | 0.2620 | 0.3564 | 0.9637 | 0.9637 | 0.5685 | 0.5683 |
| 0.01100 | 0.2664 | 0.3640 | 0.9825 | 0.9822 | 0.5796 | 0.5794 |

Table 8 Comparison of utilizations of nodes in analytical and simulation models

This section presents a comparison between analytical and simulation results. There are two types of simulation results available, and the second simulation model results are used to check the accuracy of the analytical model. The second simulation model is chosen because it represents the true behavior of NIC, and in this section it will be henceforth referred to as the ‘simulation model’. The effects of all the assumptions that are made in the analytical model are observed. Table 8 shows the comparison between the analytical and the simulation models for the utilization of different nodes.

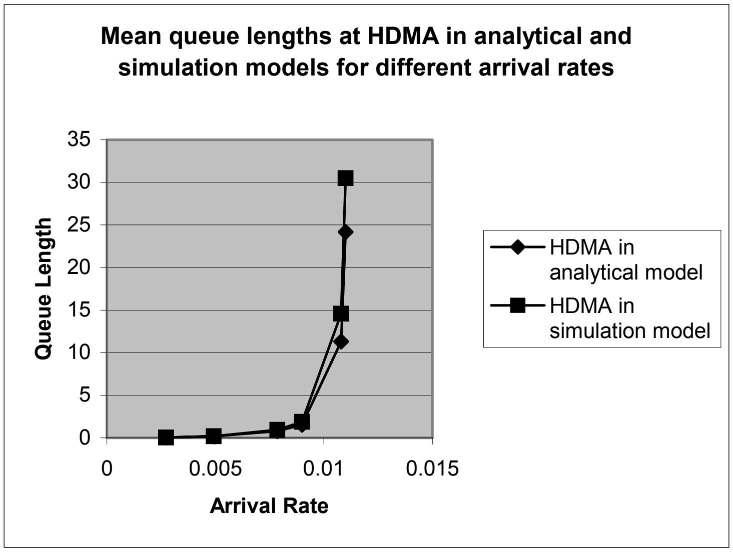

Figure 5: Mean queue lengths at HDMA in analytical and simulation models

The utilization results match very closely for HDMA and NSDMA nodes, with the maximum error percentage being 0.35. The deviation in the utilization values for the node LANai (maximum error percentage is –26.81) is due to the approximation involved in the estimated probability of 0.5 that is chosen for the probability of messages being present in the data queue when NSDMA is idle. The deviation in the utilization of node LANai is not a critical result for two reasons. Firstly, the LANai node is a mathematical abstraction for the performance of LANai, and secondly HDMA is the bottleneck node in the network. Nonetheless, the approximation involved in the estimated probability value is bound to influence the mean queue length of HDMA, which is the other performance measure of interest in the present study. Figure 5 compares the mean queue lengths of HDMA in the two models for different arrival rates. The analytical queue length of queue that the NSDMA serves cannot be compared with the simulation because in simulation, the NSDMA does not serve any queue. The queue length of node LANai can be compared by summing up the lengths of the queues that are polled by LANai in the simulation model (see Table 9).

| Arrival Rate λ | Analytical queue length of node LANai | Sum of lengths of three individual queues that LANai polls (simulation) |
|---|---|---|
| 0.00273 | 0.0059 | 0.0064 |
| 0.00493 | 0.0191 | 0.0222 |
| 0.00786 | 0.0486 | 0.0626 |
| 0.00900 | 0.0642 | 0.0854 |
| 0.01079 | 0.0940 | 0.1317 |
| 0.01100 | 0.0980 | 0.1378 |

Table 9 Mean queue lengths at LANai in analytical and simulation models

Summarizing the comparison between the analytical and simulation performance measures,

(1) In most cases, analytical performance measures are less than simulated performance measures.
(2) Utilization values of HDMA and NSDMA match in both models.
(3) Utilization values of LANai in analytical model are lower than the values in simulation model.
(4) The analytical model predicts the mean queue length of the bottleneck node in the network (HDMA) within an average error of 14% and a peak-utilization error of 20-25%, which are fairly good estimates. At lower utilizations, the model predicts the mean queue length of HDMA with higher accuracy.

The estimated probability for Scenario 2 explained in Section 3.1 affects the utilization value for LANai in the analytical model. For all other performance values, the deviations are attributed to various simplifying assumptions under which analytical model is obtained: class-switching, mathematical abstraction for programming NSDMA and the estimated probability for Scenario 2 in Section 3.1.

## 6. Concluding Remarks and Future Work

This section summarizes this research work and provides pointers to future developments possible in this area. Section 6.1 provides a short synopsis of modeling, simulation and the comparison between both of them to check the validity. Section 6.2 highlights the research accomplishments, and Section 6.3 presents concluding remarks of the work with possible enhancements that can be made and possible other research opportunities that lie ahead.

### 6.1 Summary of the Research Work

A multi-station and multi-class queueing network model is applied to study the performance of Myrinet NIC. The stations correspond to LANai and the DMA engines. Different messages in the system, which are doorbells, descriptors and data, are modeled as different classes of traffic in the queueing network. Various simplifying assumptions are made which are essential for applying the proposed analytical model. From the point of view of design, two mathematical abstractions (modeling LANai as a node where as in reality it is a processor which visits certain number of queues in a predetermined order, and modeling the programming of NSDMA by LANai to pick the available data messages on SRAM when it is idle) are needed to develop a mathematically tractable model. Important performance measures for the design of NIC are obtained analytically. The bottleneck node of the system is HDMA. Two simulation models are presented as part of the study. They differ in the design of node LANai. LANai in the first simulation model services on a FCFS basis a single queue in which all the three classes of messages are pooled up. LANai in the second model polls cyclically three queues corresponding to doorbells, descriptors and data, which reside on SRAM. Results of the two simulation models are almost identical, and the basic assumption for the validity of analytical model that pooling up of the three classes into a single queue is not going to affect the performance is verified. The major difference between the analytical and the simulation models is the design of programming NSDMA by LANai. The analytical model uses an estimated probability for the case of NSDMA being idle and a data message being available on SRAM. NSDMA serves a queue with infinite waiting space in the analytical model. The simulations designed the programming by LANai in exactly the same way as in a NIC. NSDMA does not serve any queue and a resource status check is used for NSDMA to know whether it is busy or idle. The analytical model predicts the mean queue length of the HDMA reasonably well.

## 6.2 Contributions from this Research Work

The contributions from this research work are as follows.

(1) Applying the existing queueing modeling principles to obtain the performance measures of a NIC. It provides an alternative to obtaining performance measures by testing or by simulations, which may be expensive and computationally time intensive.

(2) Ability to study various design alternatives for the components of NIC quickly. This ability is a very important feature of analytical modeling. It provides the ability to evaluate alternate designs without going for simulations or experiments that are expensive and time-consuming. For example, changing the order of service for LANai is not going to affect the performance. The analysis presented in Section 2.2 will hold good for any distribution and for any kind of discipline as long as it is work conserving.

(3) Providing a general framework for analysis of NIC that can be referred for future work.

(4) Identifying the bottleneck in the system to be HDMA, the main direct memory access DMA engine for processing doorbells and descriptors.

(5) Reducing the effort in computation to obtain the performance measures approximately in a much easier and faster way. Once the required parameter values are put in the algorithm for a NIC, the analytical results are obtained in less than a second. Table 10 provides the time taken for the polling simulation model at various arrival rates.

| Arrival Rate λ | Time taken in minutes for simulation model |
|---|---|
| 0.00273 | 3.55 |
| 0.00493 | 7.55 |
| 0.00786 | 16.12 |
| 0.00900 | 20.93 |
| 0.01079 | 35.07 |
| 0.01100 | 37.58 |

Table 10 Time taken in minutes for the second simulation model to run

### 6.3 Recommendations for Future Work

Interesting opportunities are possible for enhancing the present research work.

(1) Analytical design of a send-and-receive NIC. Additional approximations need to be made for modeling the behavior of NRDMA that increases the complexity.

(2) Proposing a model for developing the estimated probability in an analytical manner. Present research work proposed an estimated value based on various scenarios for states of LANai and NSDMA.

(3) Different policies for the operation of LANai. Changing the order of operations is not going to affect the performance measures. Other changes in operations for LANai might be of interest for the performance improvements. An example would be to think of operations where LANai, at a time, polls more than one message at specified queues.

## Acknowledgements

The authors thank the anonymous referees for their comments and suggestions that led to considerable improvements in the content and presentation of this paper. This research was partially supported by NSF Awards No. ANI-0219747, 9900701, 9988164, 0097998, and 0130143.

## REFERENCES

[1] Compaq Corp., Intel Corp., and Microsoft Corp. Virtual Interface Architecture Specification, Version 1.0. Available at http://www.viarch.org.

[2] L. Kleinrock, *Queueing Systems, Volume 2: Computer Applications*, New York: John Wiley and Sons, 1976.

[3] M. Schwartz, *Computer-Communications Network Design and Analysis*, Englewood Cliffs: Prentice-Hall, 1977.

[4] C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*, Englewood Cliffs: Prentice-Hall, 1981.

[5] Ricardo F. Garzia and Mario R. Garzia, *Network Modeling, Simulation and Analysis*, New York: Marcel Dekker, Inc., 1990.

[6] Rockwell Software, Sewickley, PA, USA. http://www.arenasimulation.com.

[7] Nagar, S., Liu, C., Kandiraju, G., Sivasubramaniam, A., and Gautam, N., "Incorporating Quality-of-Service in the Virtual Interface Architecture", Proceedings of IPDPS, April 2002.

[8] Myricom, Inc., Arcadia, CA, USA. http://www.myri.com.

[9] W. Whitt, "The Queueing Network Analyzer", The Bell System Technical Journal, Vol. 62, No. 9, November 1983.

[10] W. Whitt, "Performance of the Queueing Network Analyzer", The Bell System Technical Journal, Vol. 62, No. 9, November 1983.

[11] Natarajan Gautam. Course Lecture Notes: Queueing Theory. Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, 2001.

[12] John A. Buzacott, J. George Shanthikumar, *Stochastic models of manufacturing systems*, Englewood Cliffs, N.J. : Prentice Hall, c1993.

[13] Gunter Bolch, Stefan Greiner, Hermann de Meer, Kishor S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, WileyEurope, October 1998.

[14] Takagi, H. "Queueing Analysis of Polling Models", ACM Computing Surveys, Vol. 20, No. 1, March 1988.

[15] Takagi, H. 1986. “Analysis of Polling Systems”. The MIT Press, Cambridge, Mass.

[16] Takagi, H., and Kleinrock, L. “Analysis of polling systems”. JSI Res. Rep. TR87-0002. IBM Japan Science Institute, Tokyo, Jan. 1985.

[17] Antchev, G. et al, “Evaluation of Myrinet for Event Builder of the CMS experiment”. Source: NEC Research Index.

[18] Thorsten von Eicken, Anindya Basu, Vineet Buch, and Werner Vogels, "U-Net: A user-level network interface for parallel and distributed computing," in Proceedings of the 15th ACM Symposium on Operating Systems Principles, Dec. 1995, pp. 40 53, Available: http://www.cs.cornell.edu/tve/u-net/papers/sosp.pd.

[19] Communication performance on Windows 2000 clusters connected by Fast Ethernet, Gigabit Ethernet , Giganet VIA and SCI networks. http://grappew2k.imag.fr/evalRezo.htm

[20] M. Bazikazemi,V. Moorthy, L. Herger, D. K. Panda, and B. Abali. Efficient Virtual Interface Architecture Support for IBM SP Switch-Connected NT Clusters. In Proceedings of International Parallel and Distributed Processing Sympo-sium, May 2000.

[21] F. Berry, E. Deleganes, and A. M. Merritt. The Virtual Interface Architecture Proof-of-Concept Performance Results. IntelCorp. Available at ftp://download.intel.com/design/servers/vi/viaproof.pdf.

[22] P. Buonadonna, A. Geweke, and D. E. Culler. An Implementation and Analysis of the Virtual Interface Architecture. In Proceedings of Supercomputing 98, November 1998.

[23] Loïc Prylli, Bernard Tourancheau, and Roland Westrelin. Modeling of a high speed network to maximize throughput performance: the experience of BIP over Myrinet. In H.R. Arabnia, editor, *Parallel and Distributed Processing Techniques and Applications (PDPTA '98)*, volume II, pages 341-349, Las Vegas, USA, 1998. CSREA Press.

[24] Dmitry Ponomarev, Kanad Ghose, Eugeny Saksonov, "Optimal Polling for Latency-Throughput Tradeoffs in Queue-Based Network Interfaces for Clusters", *7th ACM Euro-Par Conference, Manchester, UK. Published as LNCS 2150, Springer-Verlag, August 2001, pp.86-95.*