

1-1-1999

# A Hierarchical Analysis Approach for High Performance Computing and Communication Applications

Salim Hariri  
*Syracuse University*

Pramod Varshney  
*Syracuse University, varshney@cat.syr.edu*

Luying Zhou  
*Syracuse University, lzhou@cat.syr.edu*

Vinod V. Menon  
*Syracuse University, vinod@cat.syr.edu*

Shihab Ghaya  
*Syracuse University, ghaya@cat.syr.edu*

Follow this and additional works at: <http://surface.syr.edu/eecs>

 Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Hariri, Salim; Varshney, Pramod; Zhou, Luying; Menon, Vinod V.; and Ghaya, Shihab, "A Hierarchical Analysis Approach for High Performance Computing and Communication Applications" (1999). *Electrical Engineering and Computer Science*. Paper 136.  
<http://surface.syr.edu/eecs/136>

This Article is brought to you for free and open access by the L.C. Smith College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Electrical Engineering and Computer Science by an authorized administrator of SURFACE. For more information, please contact [surface@syr.edu](mailto:surface@syr.edu).

# A Hierarchical Analysis Approach for High Performance Computing and Communication Applications

Salim Hariri,  
(Corresponding Author),  
ATM HPDC Lab., Syracuse University,  
Syracuse, NY – 13244.  
[hariri@cat.syr.edu](mailto:hariri@cat.syr.edu)  
Phone: +1 315-443-428  
Fax: +1 315-443-1122

Pramod Varshney,  
Dept. of Electrical Engineering and  
Computer Science,  
Syracuse University, Syracuse, NY- 13244,  
[varshney@cat.syr.edu](mailto:varshney@cat.syr.edu)  
Phone: +1 315-443-4013

Luying Zhou,  
ATM HPDC Lab., Syracuse  
University, Syracuse, NY-13244  
[lzhou@cat.syr.edu](mailto:lzhou@cat.syr.edu)  
Phone: +1 315-443-1117

Vinod V. Menon,  
ATM HPDC Lab., Syracuse  
University, Syracuse, NY-13244.  
[vinod@cat.syr.edu](mailto:vinod@cat.syr.edu)  
Phone: +1 315-443-1117

Shihab Ghaya,  
Dept. of Electrical Engineering and  
Computer Science,  
Syracuse University, Syracuse, NY- 13244,  
[ghaya@cat.syr.edu](mailto:ghaya@cat.syr.edu)  
Phone: +1 315-443-4401

## *Abstract*

The proliferation of high performance computers and high-speed networks has made parallel and distributed computing feasible and cost-effective on High Performance Computing and Communication Systems (HPCC). However, the design, analysis and development of parallel and distributed applications on such computing systems are still very challenging tasks. Therefore, there is a great need for an integrated multilevel analysis methodology to assist in designing and analyzing the performance of both existing and proposed systems. Currently, there are no comprehensive analysis methods that address such diverse needs. This paper presents a three-level hierarchical modeling approach for analyzing the end-to-end performance of an application running on an HPCC system. The overall system is partitioned into application level, protocol level and network level. Functions at each level are modeled using queueing networks. Norton Equivalence for queueing networks and equivalent-queue representations of complex sections of the network are employed to simplify the analysis process. This approach enables the designer to study system performance for different types of networks and protocols and different design strategies. In this paper we use video-on-demand as an application example to show how our approach can be used to analyze the performance of such an application.

*Key Words:* Hierarchical Modeling and analysis, queueing networks, ATM networks, Norton equivalence, Video-on-Demand.

*Area:* Performance analysis and modeling

## 1. Introduction

The evolution and proliferation of heterogeneous computing systems such as workstations, high performance servers, specialized parallel computers, and supercomputers, interconnected by high-speed communication networks, has led to an increased interest in studying High Performance Computing and Communication (HPCC) systems and parallel and distributed applications running on HPCC resources. There has been an increased interest in developing computer-aided engineering tools to assist in modeling, analysis, and design of HPCC systems [1-4]. Most of the current modeling and analysis techniques focus on performance issues related to the lower layers of the OSI architecture. Very few address the end-to-end performance analysis that takes into account application, protocols, and network impact on the performance. The main objective of the research presented in this paper is to remedy this problem and develop a convenient hierarchical analysis approach to analyze the performance of HPCC systems and their applications. The three general approaches used for network modeling, analysis, and design, are measurement techniques (monitoring), analytical techniques, and simulation techniques. We concentrate on analytical techniques in this paper.

Analytical techniques use mathematical models to represent the functions of different layers of an HPCC system. Modeling includes queueing network models, Markov chains, and stochastic Petri nets. The first models of computer networks were developed in the early 1960's [5,6]. A serious effort in network modeling began in the late 1960's when the United States Department of Defense Advanced Research Projects Agency (DARPA) funded the development of the ARPANET. Kurose [7] provides a detailed overview on tools developed by 1988. Also, there are several existing network modeling and simulation tools such as QNA from Bell Labs [1], Netmod from the University of Michigan [2], Netmodeler from IBM [3], OPNET from MIL3 Inc., COMNET from CACI Inc., and a few others. Most of these packages stop at the system level. Most of the tools do not address the end-to-end performance analysis that takes into account the effect of application, protocols, and network on the system performance. Queueing network analytic models concerning standard data and communication networks can be found in numerous references [8-11].

In this paper we present a generic three-level hierarchical approach for analyzing the end-to-end performance of an application running on an HPCC system. The overall system is partitioned into application level, protocol level and network level. Functions at each level are modeled using queueing networks. Norton Equivalence for queueing networks [12] and equivalent-queue representations of complex sections of a network are employed to simplify the analysis process. This approach enables the designer to study system performance for different types of networks and protocols and different design strategies. In this paper we use a video-on-demand as an application example to show how our analysis approach can be used for the performance analysis of an HPCC system. This paper is organized as follows: Section 2 describes the three-level hierarchical approach, Section 3 describes analysis and performance evaluation methodology using the approach of the previous section, Section 4 is a video-on-demand case study, and Section 5 is the conclusion.

## **2. Hierarchical Modeling of HPCC Systems**

Our approach to analyze large-scale HPCC systems is to employ a hierarchical decomposition strategy. This strategy is used as a means to decompose the system into smaller interacting functional units that can be analyzed more easily. We decompose the system into three levels – network, protocol, and application- and accordingly follow a three-layered analysis approach. This is the primary level of decomposition. Each level can further be sub-divided into finer levels of granularity until the desired accuracy in representation is achieved. Each level is represented as a network of service nodes and queues. The overall interconnected queueing-networks of the three levels represent the overall system.

This approach aims to provide a good approximation of the performance of each level of the system and finally of the whole system. A complex HPCC system is made up of diverse units that perform a wide variety of functions to realize the application. So we need to find a common method that represents all these diverse subsections and interconnects them. Studies so far have dealt with some small section of the system that was represented using a model appropriate to it, and then it was analyzed in depth. So when it comes to the overall system, we need to look for unifying threads so that each unit of the system, and the system as a whole, can be represented by the same framework. We employ queueing models where each service center is represented as a node with exponential service and a FCFS queue of finite capacity. We also identify the end-to-end traffic in the resulting queueing network. The message units that traverse the system form the traffic in the queueing networks where they are queued and serviced by the nodes within each of the three levels. For ease of analysis, the nature of traffic is assumed Poisson.

After the primary decomposition of the system into the three levels mentioned, the task is to identify the functions and components within each level and represent them as service nodes with queues. The system parameters used in the analysis are service rate and queueing capacity at each node, branching probabilities of traffic to service centers, and the arrival rates. They are obtained by calculations from individual system parameters and also from empirical data. Care needs to be taken where the message units can undergo segmentation and reassembly that causes changes in the traffic nature.

## **3. Layered Analysis and Performance Evaluation**

We propose a layered analysis approach in which after decomposition into the three levels and subsequent modeling using queueing networks, each level is analyzed in isolation with the contributions of the other levels denoted using their equivalent models. The goal of such an analysis approach is to perform the analysis in a more efficient manner, omitting all the unnecessary details while concentrating on a subsection, while at the same time capturing all the essential functions of the entity.

When product form solutions exist and local balance is satisfied, one could employ Norton equivalence in the analysis of queueing networks. Norton's Theorem, which is applied to electrical circuit theory, can be applied to the analysis of open and closed queueing networks [12]. Consider a closed queueing network (Figure.1(a)) with  $N$  customers, where we need to study the behavior of a subsystem,  $\sigma$ , of the network between two terminals, 1 and 2. An equivalent network (Figure. 1(b)) can be constructed in which all queues except those in the subsystem are replaced by a single composite queue. The service rate of the composite queue has a state-dependent value,  $T(n)$ , where  $n$  is the number of customers awaiting service in this queue ( $n = 0, 1, \dots, N$ ) and  $T(n)$  is the throughput between those terminals when there are  $n$  customers in the network and the subsystem of interest is replaced by a short. For an open queueing network (Figure.2(a)), the equivalent network (Figure.2(b)) consists of the subsystem,  $\sigma$ , and a composite Poisson source. This composite source generates customers at a rate,  $T$ , equal to the throughput of the subsystem considered, when the service times of all other queues are reduced to zero.

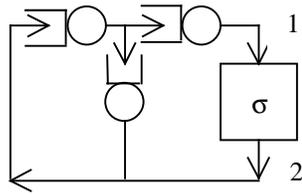


Figure 1(a) Closed Queueing Network

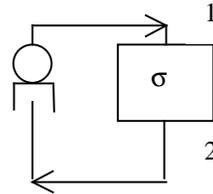


Figure 1(b) Equivalent Network

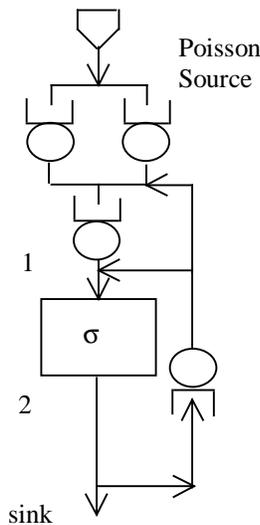


Figure 2(a) Open Queueing network

Equivalent Poisson Source

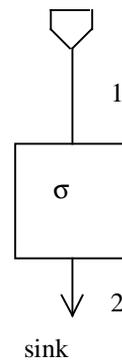


Figure 2(b) Equivalent open queueing network

We use Norton equivalence to single out a particular level, or even a subsection of it, from the overall end-to-end queueing network for analysis and replace the remainder of the system with its equivalent simplified contribution – either in the form of a composite Poisson source or a composite queue. Consider the analysis of the protocol level distributed at both ends of the communication channel. For this, we first perform detailed

analysis on the network level and obtain the delay faced by a traffic unit that traverses the network (for example, a message packet). Next, we approximate the intervening network with an equivalent queueing node; this node has its service rate equal to the delay obtained from the network analysis mentioned above. This approximation of the network's contribution helps us to concentrate on the details of the protocol level without the complications of the network level. Now, we can isolate the protocol level (with a connecting network node) for detailed study, replacing the contribution of the application layer with a composite Poisson source that supplies the traffic in terms of packets. This is shown in Figure.6 in Section 3. We can extend this method to further isolate and concentrate on a portion of the protocol level, say for instance, flow control. This method is applicable to any level of the system and further details of the analysis are discussed in the next subsections.

The advantages of this analysis methodology are manifold. First and foremost is the simplification in analysis brought forth by decomposition. Isolating a level and concentrating on its details provides for a thorough analysis. Alternate functional implementations can be tried out within the smaller unit and what-if scenarios can be explored. This gives a quick picture of variations in the system performance for different design strategies. System parameters of interest are now available at a finer level and their cumulative contribution provides the end-to-end parameters; this can also help to identify potential bottlenecks. Queueing networks provide a uniform framework to represent all levels of the system. And finally, this methodology provides a stepwise and fairly accurate framework for an HPCC system analysis and design tool.

### **3.1 Network level Analysis**

Here, a detailed analysis of the network level is made to identify the parameters and performance issues of interest to the analysis at the network level. In analyzing a network, one is interested in determining the packet transfer time through that network, network utilization, and use of equivalent queueing networks to simplify the analysis. In what follows, we describe the performance issues associated with network level analysis.

#### **3.1.1 Packet Transfer Time**

The packet transfer time through the network represents the delay imposed by the network on system response time. It includes four components:

- (1) Transmission delay, which is defined as the time of transmitting a packet by using the link rate and is determined by the packet length and the link rate (medium),
- (2) Propagation delay, which is the signal propagation time from source to destination host and is determined by the end-to-end distance (topology),
- (3) Switch delay, which is the packet processing time in a switch or a multiplexer and is determined from the characteristics of the network device.
- (4) Queueing delay, which is the waiting time of a packet in the network device. This component depends on the network transfer mechanism, network traffic load, traffic pattern, and network device.

The first three components are deterministic, while the last one is stochastic in nature. A lot of work on this aspect of network analysis has been reported in the literature [13-16]. We adopt these results to our analysis approach and use them when appropriate to analyze the queueing models.

### 3.1.2 Network Utilization

The possibility of full utilization in the network under a given traffic load is an important criterion in network design. Exceeding the capacity results in an infinite delay and affects system performance and basically crashes the network. For varying traffic, we determine the corresponding utilization factor for alternate network designs. This plot (Figure.11) describes the load handling capability of the network. This together with cost-performance consideration influences the design choice. We demonstrate this under different traffic intensity (number of customers).

### 3.1.3 Equivalent Network-Level Queueing Model

To illustrate the concept of equivalent nodes, we use the distributed system example shown in Figure. 3. A set of servers and client machines are connected together by three networks. Server S is connected to both an Ethernet and an ATM switch, and Client C is connected to an FDDI network. Suppose we need to analyze the performance between Server S and Client C as in a database access or video file transfer. To simplify the analysis, we apply equivalent queueing nodes as discussed before.

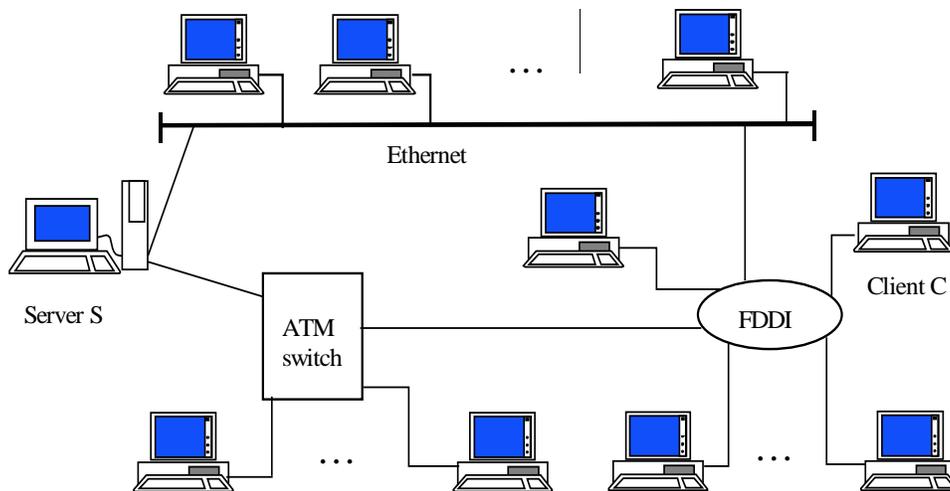


Figure 3. Distributed network application

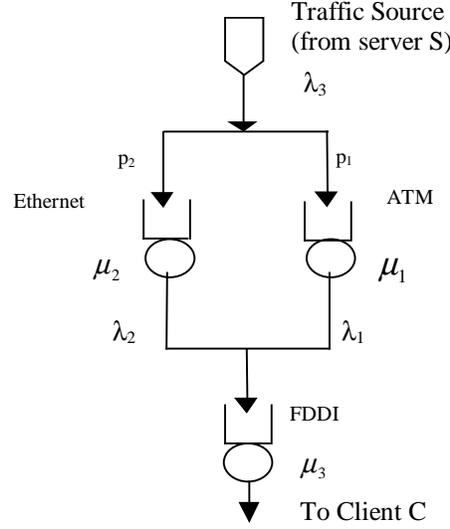


Figure 4. Queueing network model.

This can be achieved by first approximating each network by a queueing node, such that the service time of each queue is set equal to the packet transfer time through the network considering appropriate background network load [17]. The resulting approximate queueing network connecting Server S and Client C is presented in Figure 4.  $p_1$  and  $p_2$  are the branch probabilities of the traffic from server S. To find the traffic  $\lambda_3$  entering the network level, we use the open equivalent queueing node approach. This input traffic is set to be equal to the throughput of the shortened link connecting the output of the Server and the input of the Client. The throughput is calculated based on upper level traffic condition. Based on the derived network traffic the network performance can be analyzed accordingly.

In Figure 4,  $\mu_1$ ,  $\mu_2$  and  $\mu_3$  are the packet transfer rates through the respective networks and are obtained as follows:

*Calculating  $\mu_1$  : Transfer Rate over ATM network*

Assume that the ATM switch has an  $N \times N$  Space-Division architecture and with nonblocking and output buffer features, the service time (switch time for one cell) is slotted and is equal to  $\Delta T$  for one slot. Assume that the traffic at the  $N$  links are identical. Let  $p$  denote the probability that a slot or input link  $i$  contains a cell. The probability of a cell arriving at an incoming link destined for a particular outgoing link is assumed to be equal to  $p/N$  [13]. The cell waiting time at the ATM switch is approximated as

$$D_{cell} = \frac{N-1}{N} \times \frac{p}{2(1-p)} \times \Delta T \quad (3.1)$$

The packet transmission time through the ATM network,  $D_p$  and the equivalent transfer rate,  $\mu_1$ , are given as

$$D_p = D_t + D_{pr} + D_{cell} \quad (3.2)$$

$$\mu_1 = 1/D_p. \quad (3.3)$$

where  $D_t$  is the packet transmission delay,

$$D_t = \left\lceil \frac{L}{48} \right\rceil \times \Delta T \quad (3.4)$$

where  $L$  is the packet length,  $\Delta T$  is the switch speed

$D_{pr}$ , the propagation delay is given by,

$$D_{pr} = 5 \times H \quad (3.5)$$

$H$  is the link length. The propagation delay is assumed to be  $5\mu\text{s}/\text{km-cable-length}$ .

### *Calculating $\mu_2$ : Transfer Rate over Ethernet*

The average packet waiting time is given by

$$E(w) = 0.5\lambda \times \frac{E(b^2) + (4e+2)E(b)\tau_p + 5\tau_p^2 + 4e(2e-1)\tau_p^2}{1 - \lambda(E(b) + \tau_p + 2e\tau_p)} + 2\tau_p e^{-\frac{(1-e^{-2\lambda\tau_p})(\frac{1}{\lambda} + \frac{\tau_p}{e} - 3\tau_p)}{F_p^*(\lambda)e^{-\lambda\tau_p-1} + e^{-2\lambda\tau_p}}} \quad (3.6)$$

where  $\lambda$  is the total packet arrival rate to the network,  $E(b)$  and  $E(b^2)$  are the first and second moments of the packet service time,  $\tau_p$  is the maximum end-to-end propagation delay and  $F_p^*$  is the Laplace Transform of the probability density function of the packet service time. [14]

The packet transmission time through the Ethernet is given as

$$D_{pe} = E(w) + D_{tp} \quad (3.7)$$

and the transfer rate,  $\mu_2$ , is

$$\mu_2 = 1/D_{pe}. \quad (3.8)$$

where

$D_{tp}$  is the mean packet transmission and propagation time and is given by

$$D_{tp} = \frac{L}{C} + \frac{\tau_p}{2} \quad (3.9)$$

where

$C$  is the link speed,  $L$  is the packet length, and  $\tau_p$  is the end-to-end propagation delay.

### *Calculating $\mu_3$ : Transfer Rate over FDDI network*

Maximum station access delay is given by [15]

$$D_{\max} = (M - 1) \times TTRT + 2R \quad (3.10)$$

where  $M$  is the number of stations,  $TTRT$  is the target token rotation time, and  $R$  is the ring latency.

The maximum packet transmission time and transfer rate through the FDDI network are given by

$$D_{mp} = D_{\max} + D_{tp} \quad (3.11)$$

$$\mu_3 = 1/D_{mp}. \quad (3.12)$$

where  $D_p$  is the packet transmission time and propagation time

$$D_p = \frac{L}{C} + \frac{1}{2} \times 5 \times H \quad (3.13)$$

where  $C$  is the link speed,  $L$  is the packet length and  $H$  is the link length.

### 3.2 Protocol Level Analysis

In the protocol level analysis, we analyze in detail, the communication protocol running on the designed HPCC system. The protocol could be a standard protocol like the TCP/IP, UDP/IP, IP/ATM, or a synthesized protocol, where the designer specifies the implementation of each protocol mechanism or function. Some of the main issues of interest in protocol level analysis are throughput, protocol processing overhead and latency, reliability, and suitability of a certain mechanism of flow control or error control for a particular application.

In the discussion below, we describe a generalized protocol model, wherein we represent it as a set of functions performed at the protocol level and not in terms of layers as in the OSI model. These functions include connection management, flow control, error control, and acknowledgements. Connection management could encompass functions like data stream segmentation and reassembly, framing, addressing and routing, signaling, priority, and preemption. The flow-control mechanism could be window based, rate based, or credit based. The error control scheme could be Stop and Wait, selective retransmission, go-back-N, etc. What-if scenarios under alternate design implementations could be analyzed and this would aid the designer in making the appropriate choice for a given application.

A generalized protocol model could be represented by a flow diagram as shown in the example given in Figure 5, where data flow is assumed to be duplex and Regions A and B are shown in a symmetric manner to represent protocols at both ends of the system. The contents of the boxes are determined by the actual protocol. Let us consider a specific case. Let region A represent a server and let B represent a client. The message packets are assumed to arrive as a Poisson stream from the application level. We assume a controller that coordinates the protocol level functions. The packets initially pass through the connection management unit where functions like addressing and routing are performed. Acknowledgment unit represents the initiation of feedback from the receiver end and this could trigger the appropriate flow-control and error-control functions. The packets are finally consumed at the application level at the receiver end (sink). This can be represented as an open queueing network with some closed chains due to feedback.

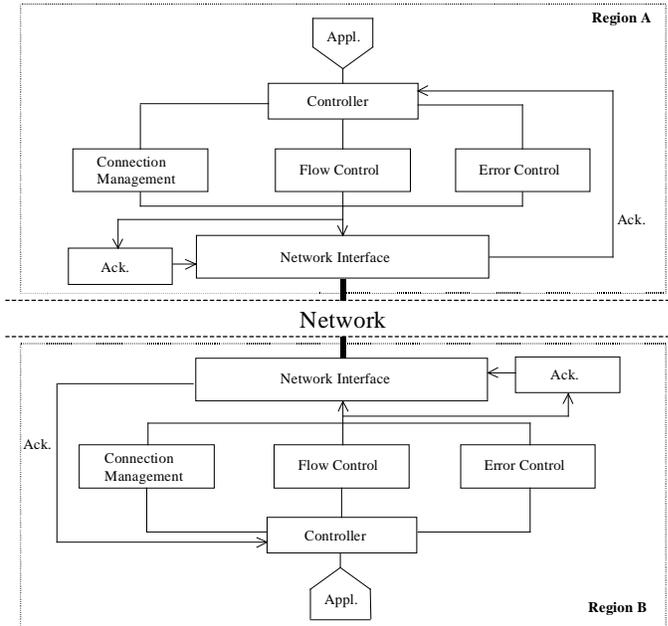


Figure 5. Example of a Flow Diagram representing Protocol Level Functions

Let us look at some of the options available to the designer to implement two main protocol level functions, namely flow control and error control.

*Flow control:*

Two main schemes are window based and rate based flow control schemes.

- (1) Window Based Flow Control: In this case, we need the following parameters: the window size  $W$ , average packet transmission time  $X$  and the round trip delay  $D$ . The maximum rate of transmission is calculated using the following relation [18]

$$r = \min \{1/X, W/D\} \quad (3.14)$$

- (2) Rate Based Flow Control: An important scheme that provides rate based flow control is the leaky bucket scheme. In this scheme, packets cannot be transmitted before they obtain a permit, where the flow of permits controls the flow of packets. The average delay for a packet to obtain a permit is given by [18]

$$T = \frac{1}{r} \sum_{j=W+1}^{\infty} p_j (j - W) \quad (3.15)$$

where the  $p$ 's are the steady state probabilities in the Markov model used to model the availability of permits,  $r$  is the rate at which the permits are allowed into the system, and  $W$  is the maximum number of permits that can be allowed at any instant by the scheme. Once the permit is obtained, the packet joins a queue for transmission; the queueing delay

at this queue is determined by the distribution of the packet sizes and standard models are used to calculate those queuing delays.

*Error Control:*

There are many different available schemes such as Stop-and-wait, go-back-N, selective repeat, etc. The performance models of some error control schemes are given below. [19]

Table 3.1. Performance models of error control schemes

Error Control Scheme	Maximum Throughput	Expected Number of retransmissions
Stop & Wait	$(1-p)/t_T$	$1/(1-p)$
Selective Repeat	$(1-p)T * l / (l+l')$	$1/(1-p)$
Go back N	$(1-p)/[T+(t_T-T)p]$	$[1+(t_T/T - 1)p]/(1-p)$

In this table,  $p$  is the probability of receiving a frame in error,  $T$  is the time required to transmit a frame,  $t_T$  is the minimum time between successive frames and is the sum of  $T$  and timeout interval.  $l$  is the length of the packet (data) field in the frame and  $l'$  is the total number bits in the control fields.

We can use the Norton equivalence for open queuing networks to isolate the protocol section for analysis. The network can be approximated with an equivalent node whose service rate is equal to the delay faced by a single message unit (or packet) passing through the network as shown in Figure 6. The application level that generates the message packets is replaced by a composite Poisson source.

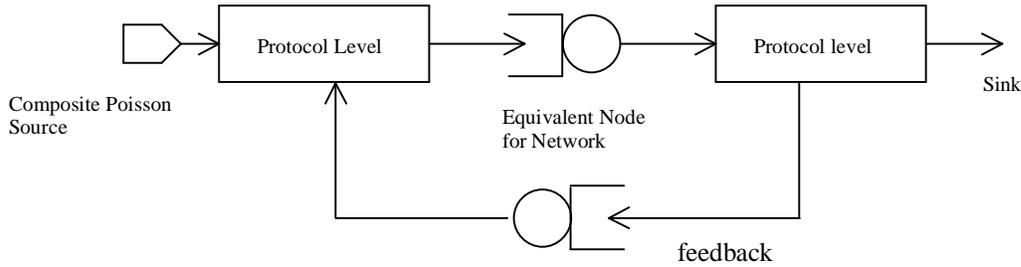


Figure 6. Model for hierarchical analysis involving Protocol and Network Levels.

### 3.3 Application level Analysis

At this level, we perform analysis of the HPCC application under consideration or a combination of HPCC applications (workloads). We develop analytical models to characterize an HPCC computation on its chosen computing platforms that take into consideration their loads, problem size, and memory requirements. Issues that need to be considered in the analysis are application specific, but one needs to consider factors like processor scheduling, parallel or distributed processing, storage strategy, type of storage such as disk or tape (for archives), type of data distribution on disks such as striping and duplicating, use of caches, etc. Analysis is then performed to obtain the end-to-end

performance metrics of interest to the application level analysis such as application response time, application utilization on each computing platform involved in its execution, and application execution cost. In section 4, when we deal with the case study, we show in more detail the performance analysis at the application level.

#### 4. Case Study: A Video-on-Demand System

In this Section, we illustrate the concepts and approaches used in the analysis and design by presenting a case study example of a VoD system. Interactive video-on-demand services enable a viewer to choose a video file from a database on a video server, specify the video's start time and have that video file transmitted over the network. Figure.7 shows a 2-tiered architecture for providing video-on-demand services.

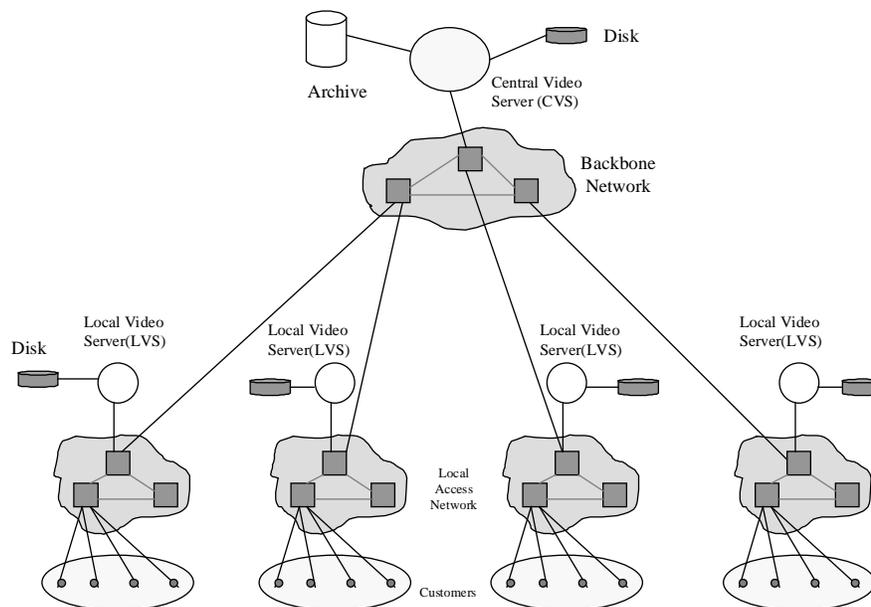


Figure 7. Layout of a VoD system

An area accessed by a VoD system is divided into several identical customer groups. Each customer group is served by a Local Video Server (LVS) connected through a local area network. Several such LVS's within a geographical area are connected to a Central Video Server (CVS) by a backbone network. Each LVS has a local copy of all the popular videos in its disk storage. The CVS keeps a copy of the popular videos in its disk storage and archives the non-popular ones in a magnetic tape storage system. A client request arrives first at the respective LVS. If the request is for a popular movie and the LVS is not congested, then the request is handled at the LVS itself. However, if the LVS is congested or if the request is for a non-popular video, then it will be directed to the CVS. Since there are only two tiers in the system under study, the request either is served at the CVS or is blocked. In the latter scenario, the client will be notified of the inability of the system to handle its request.

The system described here is analyzed using the hierarchical analysis approach presented in Section 3. The goal of our analysis is to demonstrate how this hierarchical approach can be used to analyze the performance of different configurations of network or computing technologies at each level. In this paper, we concentrate on the application and network level analysis. Protocol level analysis will be addressed in a companion research paper.

Table 4.1

Symbols used	Description
$\lambda_p$	Request Arrival Rate for popular movies to the LVS
$\lambda_{np}$	Request Arrival Rate for non-popular movies to the LVS
N	Number of customers in a customer group
L	Number of customer groups connected to a CVS
$P_{B(L)}$	Blocking probability of requests at the LVS
$P_{B(C)}$	Blocking probability of requests at the CVS
$n_l$	Number of customers accepted at a LVS
$n_c$	Number of customers accepted at the CVS
$R_{dr}$	Retrieval rate from a disk head
$R_{pl}$	Mean required playback rate

#### 4.1 Application Level Analysis

The objective of this application level analysis is to determine some parameters and issues of interest with respect to a VoD application. This first involves determining the features of the servers such as disk storage, presence of archives, scheduling of file retrievals, queuing of requests, QoS guarantees, and client-side buffer for minimum build-up. The request arrival for videos from customers is assumed to be a Poisson process. The analysis here involves determining the load handling capacity at the LVS and the CVS which indicates the maximum number of customers that the system can support, and blocking probability for customer requests (admission control).

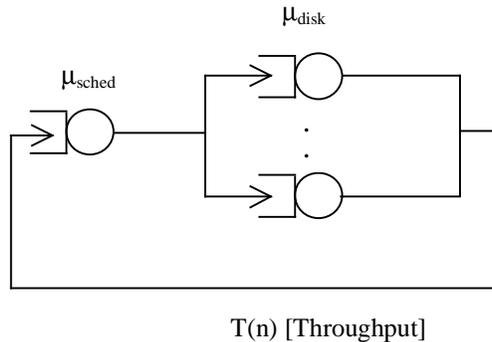


Figure.8 Closed queuing network representation of frame retrieval at LVS

### 4.1.1 Application level analysis at the local server

The number of customers that can be handled at the LVS depends primarily on the retrieval capability of the disk subsystem. Once admitted, we assume a Round Robin scheduling strategy for the customers, where each one is allotted a service time for frame retrieval. A service round is completed when all the admitted customers have been served once and we get back to the first customer. The service rate is proportional to the retrieval capability. For a disk, retrieval capacity is given by  $kR_d$  Megabits/sec(Mbps), where  $R_d$  is the retrieval rate from a disk head, and  $k$  is a factor to account for retrieval latencies. Also, there are  $d$  disks in an array.

Assuming the average size of a frame to be  $F$  Mbits, we convert the retrieval rate to frames per second. We assume in a service round, only 1 frame is retrieved for a customer; this could mean increased number of schedulings and henceforth, delays. But if the scheduling rate is taken to be very high, this delay is very small. In addition, scheduling and queueing delays can further reduce the throughput. In the QoS negotiation phase, we agree to guarantee a minimum playback rate to the customers, equal to  $R_{pl}$  (we assume 24 frames/sec). We model the frame retrieval from disks as a closed queueing network (see Figure.8). So, for a state-dependent throughput of  $T(n_i)$ , the number of customers that can be admitted will be

$$n_i = T(n_i)/R_{pl} \quad (4.1)$$

This is obtained by iteration till we arrive at the maximum supportable number of customers. The throughput is calculated based on the following iterative formula [20],

$$g(n, m) = g(n, m-1) + \rho_m g(n-1, m), \rho_m = \frac{\lambda_m}{\mu_m}$$

$$g(n, 1) = \rho_1^n, n = 0, 1, \dots, N$$

$$g(0, m) = 1, m = 1, \dots, M \quad (4.2)$$

The throughput  $\gamma_i$  of queue  $i$  is given by

$$\gamma_i = \mu_i \times p(n_i \geq 1) = \lambda_i \times g(N-1, M) / g(N, M) \quad (4.3)$$

Thus

$$T(n_i) = \mu_{sched} \times p(n_i \geq 1) \quad (4.4)$$

This throughput is the rate at which frames would enter the protocol-processing phase. Given a set of system parameters in LVS: mean frame size of 28000Bytes/frame (we use the value obtained from the statistics of a *Star Wars* MPEG1 trace [21]), disk retrieval rate of 66.96 frames/s (per disk), scheduling rate of 60000/s, we obtain the capacity of the LVS using equations 4.2 - 4.4 and the results are shown in Table 4.2.

Table 4.2 LVS parameters and capacity

# disk-heads	5	10	15	20
Throughput (frames/s)	200	385	561	747
Max customer number	8	16	23	31

For 15 disks, the throughput is  $T(n_l)=561$  f/s, and the maximum number of requests the LVS can handle is  $n_l=23$ . This number is used for estimating traffic through local network.

Thus for a given server, with a given number of disks, we can find the limit on the number of customers who can be guaranteed a quality of service. Since the server is now simultaneously handling  $n_L$  requests, we can consider the LVS to be composed of  $n_L$  virtual servers and thus model the LVS as an  $M/M/n_L/n_L$  system [22] where queueing of requests is not allowed. This assumes the service times to be exponentially distributed with a mean service time of  $T_m$ , which is equal to the duration of the movie, taking into account the interaction periods with the client. We assume  $T_m=3600$ s and ( $\mu=1/T_m$ ). From this  $M/M/n_L/n_L$  model, we can compute the blocking probability  $P_{B(L)}$  for a request whose arrival rate is  $\lambda_p$  (popular video requests). This is given by the following equation:

$$P_B = \frac{\rho^{n_L}}{n_L!} p_0 \quad (4.5)$$

where  $\rho = \lambda_p/\mu$  and,

$$p_0 = \left[ \sum_{c=0}^n \frac{\rho^c}{c!} \right]^{-1} \quad (4.6)$$

The plot of request blocking probability on LVS against  $N$ , the maximum number of supported customers, is shown in Figure 9(a).

#### 4.1.2 Application level analysis at the central server

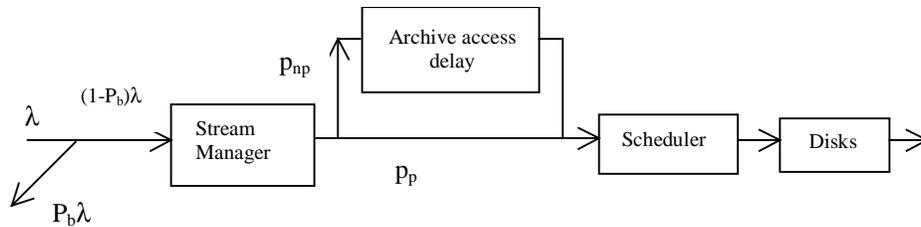


Figure.10 Video file retrieval process at the CVS

Figure 10 shows the model of the video retrieval process at the CVS. All blocked requests as well as requests for non-popular movies are handled at the CVS. The arrival rate of requests to the CVS from an LVS would then be  $(\lambda_p P_{B(L)} + \lambda_{np})$ . Assuming there are  $L$  LVS's under a CVS, the total arrival rate at the CVS would be  $L\lambda_{np}$  for the non-popular movies and  $L\lambda_p P_{B(L)}$  for the popular ones. This is shown in Figure.10 as  $\lambda=L(P_{B(L)}\lambda_p+\lambda_{np})$ . The acceptance rate for popular movies is given by  $\lambda_p^*=L(1-P_{B(C)})P_{B(L)}\lambda_p$  and the acceptance rate for non-popular movies  $\lambda_{np}^*=L(1-P_{B(C)})\lambda_{np}$ . These requests for popular and non-popular movies are handled separately at the CVS; the

former by the archival subsystem and the latter by the disk storage subsystem (service rate,  $\mu_d$ ). The branch probability to the archive  $P_a = \lambda_{np}^*/(\lambda_p^* + \lambda_{np}^*)$  and the branch probability to the disks,  $P_d = \lambda_p^*/(\lambda_p^* + \lambda_{np}^*)$ . That is, we assume the relative access probabilities to the disk and archive to be in the proportion of the relative accepted request rate for popular and non-popular movies, respectively. The non-popular movies are first accessed from the archives (which could be a magnetic-tape storage system) and are then laid out on the disk caches using efficient algorithms for striping and placement [23]. Now, the scheduler generates disk reads to retrieve the video files.

As in the case of LVS's, the blocking probability for requests is derived from the M/M/n/n model for which we need to determine the number of customers accepted by the CVS, with guaranteed QoS. The closed queueing network model for video retrieval in CVS is similar to that for LVS as shown in Figure 8.

Given the system parameters for the CVS: Disk retrieval rate=66.96 frames/s(per disk), Scheduling rate=60000 /s, Number of LVS L=10. The maximum number of customers that can be supported for number of disks ranging from 30 to 60 is shown in Table 4.3 along with the corresponding frame throughput.

Table 4.3 CVS parameters and capacity

# of Disk-heads	30	40	50	60
Total Throughput (frames/s)	1108	1470	1832	2203
Max N (number of customers accepted)	46	61	76	92

Thus, from these computations, with the system capacity  $n_c$ , and modeling the server as an M/M/ $n_c$ / $n_c$  model, we derive the blocking probabilities of the admission control phase (see Figure.9(b)).

## 4.2 Network Level Design and Analysis

In this level of analysis, we concentrate on the network design and performance analysis such as exploring different network technologies such as ATM, FDDI, etc. The related performance metrics are network transfer delay (switch or multiplexer transfer delay, transmission delay, propagation delay), cell loss probability, throughput, error rates, etc. This analysis is performed for the central and local networks. For this video application system, we analyze the network transmission delay performance, computing the average packet delay. We consider a video frame of mean size 28000bytes to be segmented into packets of size 2200bytes (including overheads) for transmission. We also determine the amount of traffic that can be handled by various networks, and suggest the ideal network technology for the given application. In the VOD case study ATM networks are used as local access networks and central distribution network.

### 4.2.1 Performance Analysis of Local Access Network

Based on the analysis performed at the application level, we obtain the traffic load and its characteristics, and can thus proceed with the network level analysis. We investigate the packet transfer time over Ethernet and ATM networks. Using the Equations (3.6), (3.7) with regard to VOD streams, we found that Ethernet can handle only two customers and is thus ruled out. Next we study the performance of ATM switches operating on 155 and 622 Mbps.

#### 1) Analysis of 155 ATM local access network

In equation (3.1) for the ATM delay performance, when  $N$  approaches infinity, the analytic model tends to be an M/D/1 queue, as shown in (4.7)

$$D_{cell} = \frac{P}{2(1-p)} \times \Delta T \quad (4.7)$$

We thus approximate the ATM switch delay as an M/D/1 queue delay. We assume the cell arrival distribution at the input link of an ATM switch to be Poisson and the switch time for each cell to be fixed. Based on the packet arrival rate to the network level from the application level and protocol level analysis, we approximate the cell arrival rate to be equal to  $\lambda \lceil L/48 \rceil$ , where  $\lambda$  is the mixed packet rate from local and central video servers and  $L$  is the mean packet length. Thus, the cell delay,  $D_{cell}$ , is given by

$$D_{cell} = \frac{\lambda \lceil L/48 \rceil \Delta T}{2(1 - \lambda \lceil L/48 \rceil \Delta T)} \times \Delta T \quad (4.8)$$

The packet transmission time through the ATM network is given by Equation (3.2), i.e.,

$$D_p = D_t + D_{pr} + D_{cell} \quad (4.9)$$

Figure 11(a) shows the packet transmission time as a function of the number of customers. In the analysis, we know that one link of 155 Mbps ATM can handle about 20 video customers. Given the condition that 23 customers in a customer group are served by LVS and 9 customers from the same group are served by CVS, the local access network needs to handle about 32 customers. Consequently, we do need to use two 155 ATM links to handle the video traffic from the LVS.

### 4.3.2 Performance Analysis of CVS Backbone Network

In order for the CVS to support 92 customers, the number of disks is 60 and the total throughput is 2203 frames/s (see Table 4.3). We consider using ATM network with two speeds (155 and 622 Mbps) to support the video traffic.

#### a) Using 155 Mbps ATM link

Based on analysis shown in Figure 11(a), we do need about 5 ATM links between the CVS and backbone network in order to handle the required customers.

b) *Using 622 Mbps ATM link*

Figure 11(b) shows the packet transfer time as the function of number of video customers. It is clear from the Figure that one 622 Mbps ATM link can not handle the required number of customers. Consequently, we do need two 622 ATM links to handle the traffic.

In summary, we have either to use five 155 Mbps ATM links or two 622 Mbps ATM links in order to support the required number of customers. Based on cost-effective analysis, one can choose the optimum solution.

### 4.3 End-to-End Performance Evaluation

Besides analysis at each level, we can also do end-to-end performance evaluation using this method. End-to-end application response time is an important parameter in system analysis. To calculate the application response time, we need to find the time interval between the request leaving the client and the commencement of playback. The sum of delays that a packet encounters in the different stages is computed. To calculate the average delay, we determine the probabilities that the request is served at the LVS and CVS. Thus the delay in the path to customer end from CVS and that in the path from LVS are weighted by these probabilities to get the mean end-to-end delay. The protocol-processing delays and the network delays are obtained from the analysis at these respective levels.

A minimum number of frames (a few Megabytes for a moderate memory) should be available in the buffer at the client-end before playback can commence. This is to act as a cushion against possible starvation due to intervening delays. So based on the minimum number of frames needed and packet arrival rate, this buildup delay can be computed.

For the requests arriving at one specific LVS, the ratio of requests processed by LVS and by the CVS is taken as the ratio of the number of customers in a customer group (that accesses the same LVS) that get serviced at the LVS and CVS respectively. For example, in our analysis, we take one instance of the number of customers served at the LVS and CVS as 23 and 92 respectively, from considerations of the server and network capacities. So, the probability that the service was at LVS is  $P_{lvs}=0.714$ , and that the service was at CVS is  $P_{cvs}=0.286$ . To calculate the response delay, the delay faced in each system element by the first retrieved packet till it reaches the client-side buffer is required. The other packets follow in a pipeline. At the client-side, a delay to account for the minimum build-up of frames is to be provided. In addition, decoding of the MPEG coded frames, and some latency in initiation of playback, contribute further to the total delays. The response delay,  $D$ , from server to client after the request has been accepted for service is composed of the following:

$$D = (D_{lvs} + D_{lp}) * P_{lvs} + (D_{cvs} + D_{cp} + D_{cnet}) * P_{cvs} + D_{lnet} + D_{client} \quad (4.10)$$

where  $D_{lvs}$  and  $D_{cvs}$  are the retrieval delays at LVS and CVS,  $D_{lp}$  and  $D_{cp}$  are the protocol processing times related to LVS and CVS,  $D_{lnet}$  and  $D_{cnet}$  are the packet transmission times in local and central network and they depend on network used and network traffic

loads (this was obtained from network level analysis).  $D_{\text{client}}$  is the delay at the client end, and it includes the minimum frame storage time,  $D_{\text{store}}$ , and decoding time,  $D_{\text{dec}}$ .

## 5. Conclusion

In this paper, we presented a three-level hierarchical analysis approach to analyze the performance of a High Performance Computing and Communication system and its applications. We used queueing network models to represent the system in each of its three levels. Norton Equivalence for queueing networks and equivalent queueing-node models were used to isolate and analyze each of the three levels, thereby making the analysis detailed at each level and also simpler. This helped us to concentrate on a certain level while the other levels were replaced by their equivalent contributions. This approach also facilitated exploring alternate design implementations. We also described a simple 2-tiered VoD system and analyzed its performance at the application and network levels. A proof-of-concept design and analysis tool, NETHPCC, has been implemented in the lines of our approach at the HPDC laboratory at Syracuse University. It can be accessed at <http://ebeowulf.cat.syr.edu:5000>.

## References

- [1] W. Whitt, "The Queueing Network Analyzer", *Bell System Technical journal*, Vol.62, No.9, Nov.1983, pp.2779-2815.
- [2] D. W. Bachmann, M. E. Segal, M. Srinivasan , and T.J.Teorey, "Netmod: A DesignTool for large Scale Heterogeneous Campus networks", *IEEE Journal on Selected Areas in Communications*, Vol.9, No.1, Jan 1991.
- [3] A. Rindos, W. Metz, R. Draper, A. Zaghloul, "Netmodeler: An Object Oriented Performance Analysis Tool", *IBM, RTP, NC, USA*
- [4] W. C. Lau, and S. Q. Li, "Traffic Analysis in Large-Scale High-Speed Integrated Networks: Validation of Nodal Decomposition Approach", *Proc. IEEE INFOCOM'93*, pp. 1320-1329, 1993
- [5] L. Kleinrock, *Communication Nets; Stochastic Message Flow and Delay*, New York: McGraw-Hill, 1964. Reprinted by Dover Publications, 1972.
- [6] L. Kleinrock, "On the Modeling and Analysis of Computer Networks", *Proc. of the IEEE*, Vol. 8, NO. 8, August 1993.
- [7] J. F. Kurose and H. T. Mouftah, "Computer-Aided Modeling, Analysis, and Design of Communication Networks", *IEEE Journal on Selected Areas in Communication*, Vol. 6, NO. 1, January 1988.
- [8] L. Kleinrock, *Queueing Systems, Vol. 2: ComputerApplications*, Wiley, New York 1976.
- [9] M. Schwartz, *Telecommunication Networks: Protocols Modeling and Analysis*, Reading, MA, Addison-Wesley, 1987.
- [10] R.W. Wolff, *Stochastic Modeling and the Theory of Queues, International Series in Industrial and System Engineering*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.
- [11] T. G. Robertazzi, *Computer Networks and Systems*, Springer Verlag, New York, 1994.

- [12] K. M. Chandy, U. Herzog and L. Woo, "Parametric Analysis of Queuing Networks", *IBM J. RES. DEVELOP.* January 1975, pp36-42
- [13] M. J. Karol et al, "Input versus Output Queueing on a Space-Division Packet Switch", *IEEE Trans. Comm.* V35, N2, pp1347-1356, 1987
- [14] E. Drakopoulos, M. J. Merges, "Performance Analysis of Client-Server Storage Systems", *IEEE Trans. On Computers*, Vol.41, No.11, November 1992.
- [15] S. Mirchandani, R. Khanna, *FDDI Technology and Application*, 1993
- [16] S. Lam, "Carrier Sense Multiple Access Protocol for Local Area Networks", *Computer Networks* Vol 4, 1980, pp21-32.
- [17] L. C. Mitchell, D. A. Lide, "End-To-End Performance Modeling of Local Area Networks", *IEEE Journal on Selected Areas in Communications*, Vol.SAC-4, No.6, September 1986.
- [18] D. Bertsekas, and R. Gallager, *Data Networks*, Prentice Hall, 1991
- [19] M. Schwartz, *Telecommunication Networks Protocols, Modeling and Analysis*, Addison Wesley.
- [20] J. P. Buzen, "Computational Algorithms for Closed Queuing Networks with Exponential Servers", *Comm. ACM*, vol 16, No 9, 1973, pp527-531
- [21] M. Garret, "starwars.frames": available via anonymous ftp from thumper.bellcore.com.
- [22] V. O. K. Li, W. Liao, X. Qiu, E. W. M. Wong, "Performance Model of Interactive Video-on-Demand Systems", *IEEE Journal on Selected Areas in Communications*, Vol.14, No.6, August 1996.
- [23] Y. H. Chang, D. Coggins, D. Pitt, D. Skellern, M. Thapar, C. Venkataraman, "An Open-Systems Approach to Video on Demand", *IEEE Communications magazine*, May 1994.

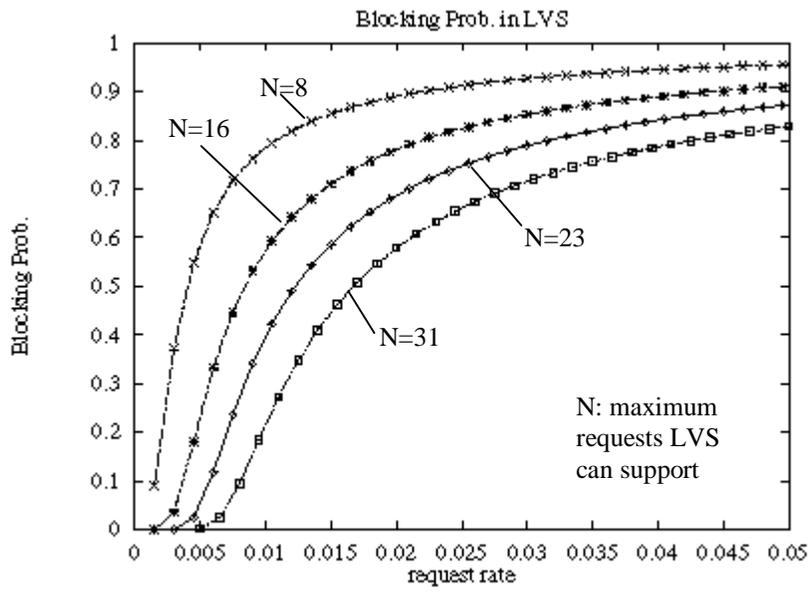


Figure 9(a). Request Blocking Probability at LVS

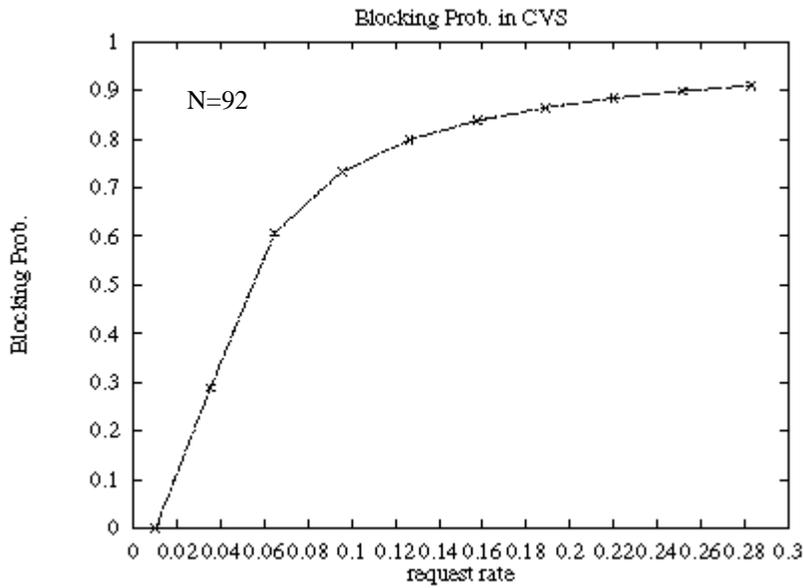


Figure 9(b) Request Blocking Probability at CVS

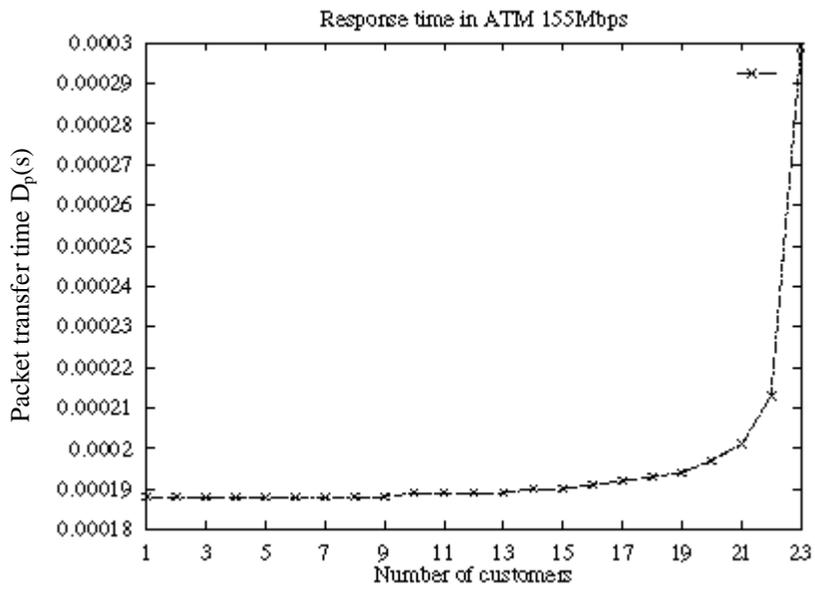


Figure 11(a) Transfer delay vs. number of video customers

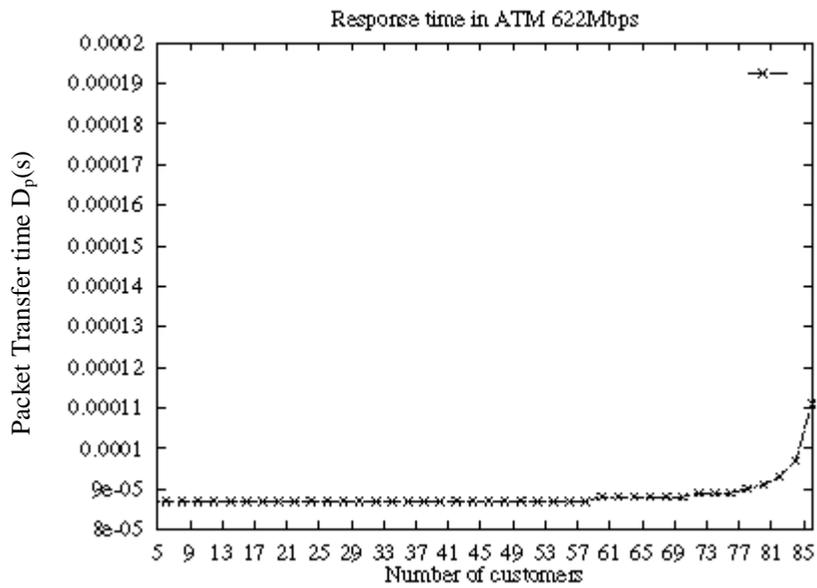


Figure 11(b) Transfer delay vs. number of video customers