Journal of the Islamic University of Gaza, V.10 No. 2 P-P, 2002

DESIGN OF A SCALABLE ARCHITECTURE OF A MULTIMEDIA-ON-DEMAND SERVER

Mohammad A. Mikki[*]
Islamic University of Gaza
Gaza, Palestine
mmikki@mail.iugaza.edu
P.O. Box108, Gaza, Palestine

Kangbin Yim[**]
Ajou University
Suwon, Korea 442-749
hotspot@csl.ajou.ac.kr

تصميم معمار توسعي لخادم "وسائط متعددة حسب الحاجة"

ABSTRACT **in this paper we develop a Vhdlbased scalable architecture of a multimedia on demand server model. Te three server subsystems are:the control subsystem, the communication subsystem and the storage subsystem are modeled. The communication subsystem is designed as an interconnection network with meshed-pipe structure and delivers packets to their destination through the sub-optimal shortest path. A model for the video traffic generated at the server side for modeling the data sent to the users is also presented**

We use simulation experiments to validate our modeling approach and compare the results with those obtained using OPNET simulation tool. The comparison results show that our approach is accurate compared to the simulation results. Our modeling approach also shows that using VHDL (Very high speed integrated circuits Hardware Description Language) in modeling digital systems is powerful and accurate.

ملخص **في هذا البحث قمنا بتطوير نموذج معمار توسعي لخادم "وسائط متعددة حسب الحاجة" مبني على اساس لغة VHDL . ان الانظمة الفرعية لهذا الخادم و هي: نظام التحكم الفرعي، نظام الاتصال الفرعي، و نظام التخزين الفرعي قد تم تصميم نماذج لها. لقد تم تصميم نظام الاتصال الفرعي كشبكة ترابطية على شكل تركيب انبوب شبكي، و يقوم هذا النظام الفرعي بتوصيل رزم البيانات الى غاياتها باستخدام المسار الاقصر الامثل.**

لقد تم أيضا في هذا البحث عرض نموذج لحركة مرور الفيديو التي يتم توليدها في جانب الخادم لنمذجة حركة مرور البيانات المرسلة الى المستخدمين.

لقد قمنا باستخدام تجارب محاكاة لتدعيم و تأييد طريقة النمذجة التي قمنا بتصميمها و مقارنة النتائج بتلك التي تم الحصول عليها باستخدام برنامجOPNET للمحاكاة. ان نتائج المقارنة تبين أن طريقتنا هي طريقة دقيقة مقارنة بنتائج المحاكاة. كما أن طريقة النمذجة المصممة في هذا البحث تثبت أيضا أن استخدام لغة VHDL في عمل نماذج للانظمة الرقمية هي طريقة فعالة و دقيقة.

1. INTRODUCTION

Servers are an integral part of multimedia environments such as digital libraries, video on demand, in-house training, digital universities, etc.

---

[*] Assistant prof. At the electrical and engineering department .
[**] Assistant pro. At Ajou University .

DESIGN OF A SCALABLE ARCHITECTURE

Multimedia-on-demand (MOD) servers are required to store, manage, and retrieve the multimedia data streams isochronously on request. The most important parameters in a multimedia server are its I/O bandwidth, seamless playback with very hard real-time constraints, and storage requirements. The I/O bandwidth designates how many clients can be simultaneously served, and the available amount of storage determines the number of video streams that can be stored in the server [Serpanos 1998]. Another important requirement of a multimedia server is scalability. A scalable multimedia server consists of a cluster of processors (storage subsystem nodes) that share a pool of disks [Krikelis 1998].

To increase the bandwidth, many ideas have been proposed. The ideas include utilizing the RAID and striping data throughout the components attached to a single bus [Chen 1996]. But this architecture is not suitable to efficiently handle the so-called "hot movie contention problem". The problem occurs when extremely many requests of very few popular movies are issued in a short period of time. The major reason behind this problem is that each popular movie is located entirely in one processing element. Regardless of the number of processing elements that compose a single scalable server, requests are congested on a specific processing element which contains the hot movie while other processing elements are idle. Considering the request congestion phenomenon, it is required for a high performance scalable MOD server to have a somewhat different architecture to resolve the hot movie contention problem. One simple technique to solve this problem is copying several frequently accessed movies and locating them in multiple processing elements. Another popular technique is utilizing hierarchical memory structure. Frequently used programs are located in the fast storage while less popular programs are placed in slow storage. We have designed and implemented a new ATM-based scalable Multimedia-on-Demand Server (SMoDS) model which uses the inter-unit striping technique for movie storage.

The rest of the paper is organized as follows. Section two presents the related work. Section three presents an overview of our approach. Section four presents how we use VHDL to model the SMoDS in detail. Section five presents some simulation and experimental results. And finally, section six concludes the paper.

2. RELATED WORK

In this section we present some of the relevent work in the field of Video-on-Demand server design, modeling, and analysis.

[Wright 2001] presents a one-way elevator with interleaving and delayed start VoD model that uses the elevator algorithm for disk scheduling. The

design includes the scheduling algorithm, buffer policy, file storage layout, and a mechanism for dealing with the system's saturation. [Cao 1999] proposes an architecture with dividing the media servers into multiple groups with a specific size. Within each group, there are a registration agent and an index agent, which take care of the resourse reservation, membership management, media indexing search, and load balancing. [Fei 1997] presents a distributed multimedia on demand system based on client/server architecture. It focuses their attention on how to provide intra and inter media synchronization in presentation, and on providing QoS standard in order to make a trade-off between overall performance of the system and synchronization presentation of each multimedia application. [Fabbrocino 1998] presents a large-scale, multi-user multimedia server that supports both current and next generation multimedia applications. The server utilizes randomized data allocation with a dynamic load-balancing scheme that provides a statistically guaranteed delay bound for I/O performance. [Sonah 1995] presents a multimedia architecture and a data retrieval model for supporting simultaneously multiple clients requesting files of different playback rates. It analyzes the architecture's performance using a circular SCAN disk scheduling policy in terms of the maximum number of concurrent video streams it can support. It uses a technique named the maximum operation set and the Rate-base Buffer Weighting Scheme to relieve the processing load on the server.

3. VHDL-BASED MODELING AND ANALYSIS

VHDL is the newest standard language to address the rapidly growing complexity and sophistication of digital system design. VHDL is rapidly gaining acceptance and is influencing advancements in design methodologies and design automation technology [Dewey 1997],and [Skahill 1996]. It plays an increasing part in digital systems design and modeling. VHDL is currently being used as a common modeling language. Once written, a VHDL model can be executed by a software program called a simulator. A simulator runs a VHDL description and computes the outputs of the modeled digital system in response to a series of inputs applied over time [Dewey 1997],and [Skahill 1996]. Currently, there is no comprehensive method to use VHDL in modeling and analysis of Multimedia servers. Our approach extends the capability of VHDL to be used as a modeling language. We chose VHDL for performance modeling and analysis because it has the following important features:

- It provides a standard, portable and flexible design representation for complex digital hardware.

DESIGN OF A SCALABLE ARCHITECTURE

- It provides support of concurrency, program constructs and behavioral completion.
- It is an executable representation in that a simulator can be used to verify functionality and timing specifications.
- It enables models to be refined as design details become available.
- It supports top-down hierarchical modeling that supports accurate modeling and simplifies the design process.
- It is an abstract modeling language that could describe the temporal behavior and structure of a system from the overall block diagram level down to the gate level.

Figure 1 shows the main components of our VHDL-based modeling and analysis approach. It is based on a three-level hierarchy that analyzes the MOD server's performance by partitioning it into network level, protocol level and application level. The functions and operations of each level are modeled using VHDL. This approach enables the designer to study different architectures of MOD servers and different traffic models. The approach consists of the following modules: Traffic Module, Controller Module and System (component) Module. The Traffic Module can model different types of traffic models (e.g., Poisson, Bernoulli). The System Datapath Module models the main functions in any system or component (e.g., channel, processor, switch, interconnection network, memory). The Controller Module defines how these functional units interact with each other during the system operations (normal or abnormal). It also models the protocols and the routing algorithm. Each system (component) is divided into two parts: a controller and a datapath. The datapath manipulates data according to the commands from the controller. The controller is modeled as a finite state machine (FSM). We chose finite state machines because they are commonly implemented in programmable logic devices [Dewey 1997],[Skahill 1996]. Writing a behavioral state machine description in VHDL is simply a matter of translating a state flow diagram to **case-when** and/or **if-then-else** VHDL statements. Furthermore, the controller module can also model the communication protocols and routing algorithms. Once the VHDL models that represent the traffic, the controller and the datapath associated with the system (component) are modeled, the next step is to invoke the VHDL simulator and obtain the performance metrics as specified in the VHDL models. In the next section, we will discuss in further detail how to use our VHDL-based modeling to analyze the performance metrics (packet delay, packet loss probability, deflection ratio and switch link utilization) of the SMoDS.

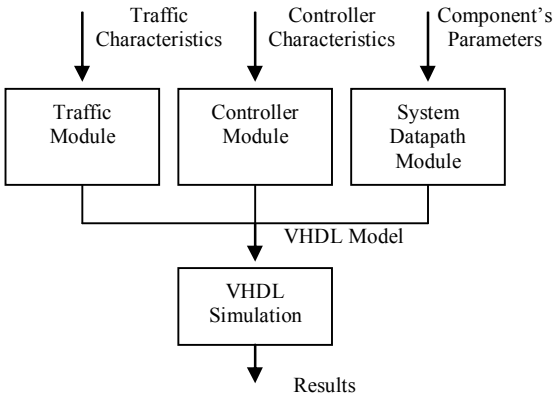

Figure 1: Procedure of the VHDL-based modeling approach

4. DESIGN AND IMPLEMENTATION OF THE SERVER ARCHITECTURE

A MOD server consists of three subsystems (see Figure 2). The control subsystem for managing the client requests, the communication subsystem moves the data from the server to the clients, and the storage subsystem manages the storage and retrieval of data from storage disks [Krikelis 1998]. The users and the server are connected through an ATM network. The SMoDS subsystems are explained in detail in the next three subsections.

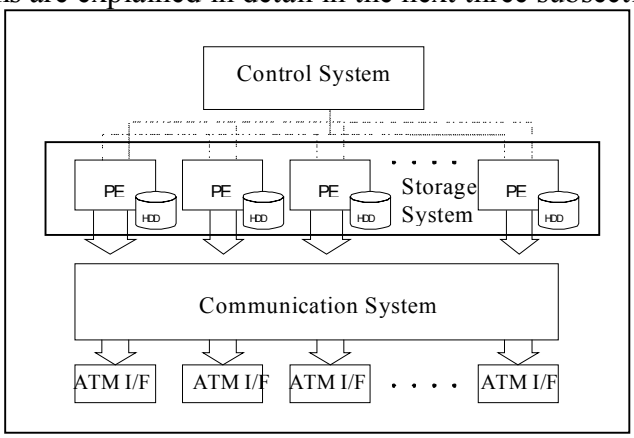

Figure 2: SMoDS structure

4.1 CONTROL SYSTEM

The control system responds to user requests. It contains all information needed for managing and controlling the server and clients. The major information includes the locations of video streams, a valid client list, system configuration, and database for payment. The admission controller is located in the control system. Whenever a new client requests service, the

admission controller decides whether it admits the request or not, depending on system resources such as available network bandwidth and I/O bandwidth. A real-time communicator is also included in the control system. To maintain the sequence of data streams and serve the streams continuously without violating the hard real-time requirements, the real-time communicator has to provide the starting times and the related signals to the appropriate data pumps in the storage system at exact times.

4.2 STORAGE SYSTEM

The storage system manages the storage and retrieval of multimedia data from storage disks. It consists of a cluster of processors with each processor physically attached to local disks. We call a processor and the disks physically attached to it a data pump. The proposed SMoDS distributes all video streams to all attached data pumps instead of storing a whole movie in a single processing element. With this structure all processing elements share the load, and the case that a few processing elements are heavily loaded while other processing elements are idle is eliminated. This also leads to the elimination of the hot movie contention problem. In the storage system two processors are employed: The main processor takes care of communicating with the server manager and scheduling the commands issued by the manager. And the I/O processor, performs scheduling I/O tasks issued by the main processor, accesses the requested data streams from massive storage disks, and delivers them to the network. Since the target network is ATM, the I/O processor generates ATM cells from the accessed video streams.

In the inter-unit striping architecture two problems have to be solved: first, the synchronization between consecutive processing elements to guarantee seamless playback of the transferred data on the user side. Second, the delivering of data streams from different processing elements to a specific established communication channel with the user. This paper proposes a new switch architecture to solve the second problem. The switch needs to handle very huge amount of data to support a lossless data transfer and to control data flow. To accomplish the requirement, the proposed switch hires a parallel interconnection network with a meshed-pipe architecture.

4.3 COMMURICATION SYSTEM

The communication system moves data through the network from the server to the user. Data pumps contain striped data stream of programs. Upon receiving a service command from the server manager, a data pump schedules the command with real-time operating system. According to the scheduled result, the pump accesses the requested video stream from its

massive storage and delivers it to the client through the communication system at a precise time. The communication system is designed as a local switch. Even though a movie is divided into many small stripes and distributed to all data pumps, regardless of the locations of data pumps, the stripes from any data pump have to be fed into a specific network port. The switch is implemented as the meshed-pipe interconnection network shown in Figure 3. The deflection routing algorithm [Greenberg 1993] is used by the control system to route the ATM cells through the interconnection network from the storage system to the clients.

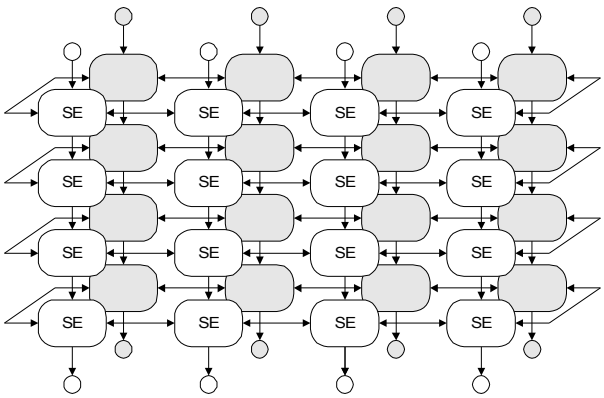

Figure 3: Interconnection network structure

In many existing mesh networks, the store-and-forward [Robinson 1994] or the wormhole routing algorithms [Park 1996],[Robinson 1994] are popularly used to solve the problem of competing packets. These algorithms are known to be efficient for communicating short and non-periodic messages. But they do not look feasible algorithms for delivering massive periodic data streams of MOD server since they use large buffers which cause excessive delay that may not be tolerable in the hard real-time MOD server. Also, buffers would be filled up too fast which makes them useless. Many studies show that the deflection algorithm outperforms the other algorithms [Greenberg 1993].

A switching element (labeled as SE in Figure 3) is the basic element of the network. The horizontal links are bi-directional and the vertical links are unidirectional. Each SE has three input ports and three output ports. The message arriving from one input port is routed to one output port. Each output port has a message-size buffer for communication synchronization purpose. The switching element has a unique control block for deciding the routing information of incoming messages. New message detecting circuit,

preference decision circuit and switch element addressing circuit are also included. The ATM interface controls establishing channels between the server and clients. Upon receiving the requests from clients, it exchanges protocol and signals with ATM network and delivers the requests to the server manager through the Ethernet. As soon as the ATM interface receives ATM cells that contain the requested data streams from the interconnection network, it supplies them to ATM network.

4.4 TRAFFIC MODEL

In this section we develop a model for the video traffic generated at the server side to send multimedia data to the users upon request. This model is used to estimate the steady state server performance. The simulation was done at the cell level, assuming fixed-length packets transmitted in equal-length time slots, where a slot is the unit of time necessary for transmitting one packet. We model all the traffic generated by each processor and destined to interconnection network as a bursty process. There are different models of traffic burstiness that were developed in the literature. [Feng 1996] overviews some of them. Most of these models include ON/OFF bursty model for VBR traffic and bursty traffic with long-range dependence. We use the bursty model similar to the model used in [Eliazov 1990]. Each processor's traffic alternates between silent periods (S) and active periods (A) from the perspective of the interconnection network with the following parameters: m(A), m(S), c^2(A), c^2(S), k(A) and ρ where:

m(A) is the mean of the active period
m(S) is the mean of the silent period
c^2(A) is the squared coefficient of variation of active period
c^2(S) is the squared coefficient of variation of silent period
k(A) is constant cell inter-arrival time during active periods.
ρ is the total load.

k(A) is assumed constant for the sake of incorporating periodicity which is inherent in most MOD applications. The load λ destined from one processor to a specific switch input port is given by Equation (1).

$$\lambda = \left[\frac{m(A)}{m(A) + m(S)} \right] \frac{1}{k(A)} \qquad (1)$$

We assume that the lengths of the active and silent periods to be independent and identically distributed random variables and their distributions are mixtures of geometric distributions. The procedure to obtain their distributions is given below.

The distribution function $p(n)$ of a mixture of two geometric distributions with parameters p1 and p2 respectively and mixing ratios of X and 1 - X is given by Equation (2).

$$p(n) = X(1 - p_1)\, p_1^{\,n-1} + (1 - X)(1 - p_2)\, p_2^{\,n-1} \quad (2)$$

$$n \geq 1,\ p_1 \quad and \quad p_2 > 0$$

$$X = 0.5\left[1 + \sqrt{\frac{(c^2 - 1)m + 1}{(c^2 + 1)m + 1}}\right] \quad (3)$$

$$p_1 = (m - 2X) / m \quad (4)$$

$$p_2 = [m - 2(1 - X)] / m \quad (5)$$

4.5 PERFORMANCE ANALYSIS

In this section we analyze the performance of the switch in the SMoDS. The packet transmission delay is determined based on the following factors:

- Average link utilization
- Packet conflict probability
- Packet deflection probability

In steady state, all links are equally utilized. The probability μ that a packet utilizes a randomly selected link is shown in Equation (6).

$\mu = u^3 + u^2(1-u) + u(1-u)^2$ (6)

where u denotes the link utilization.

In Equation (6) u^3 represents the case when three packets have arrived at the switching element, $u^2(1-u)$ represents the case when two packets have arrived and $u(1-u)^2$ represents the case when one packet has arrived. When more than one packet arrive at a switching element then a conflict occurs and the packet conflict probability η is given by Equation (7).

$\eta = u^3 + u^2(1-u)$ (7)

At any switching element, the probabilities that a packet travels in the horizontal direction and vertical direction of the shortest path are α and β respectively. The probability that a packet travels through the link that represents the non-shortest path is represented as γ which is the deflection probability. The packet deflection probability depends on the number of packets at a switching element.

The packet selecting already assigned link causes a conflict. Figures 4 and 5 show all cases of two and three competing packets at a switching element

respectively. The circles represent the switching element and the arrows represent the packets. The direction of the arrow represents the preferred shortest path direction. Some cases do not generate a conflict. In Figure 4, only cases 7 and 8 represent conflicts. In Figure 5 cases 1, 2 and 3 do not represent conflicts while all other cases represent conflicts. In cases 7 and 8 of Figure 5 the conflicting packets can be redirected to their other preferred links. So only cases 4, 5, 6 and 9 contribute to the probability of deflection. From analyzing these cases the deflection probability γ is obtained as listed in Equation (8).

$$\gamma = (\alpha^2\beta+\alpha\beta^2/2)\,\mu^3 \qquad (8)$$

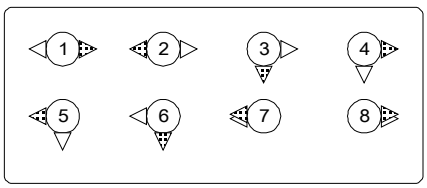

Figure 4: Two competing packets at a switch node

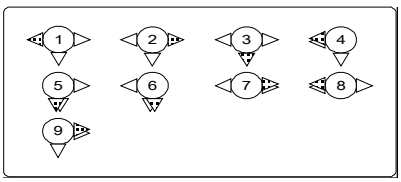

Figure 5: Three competing packets

To get the steady state deflection probability we need to determine average α and β. The average α and β denoted as α_avg and β_avg are calculated easily by evaluating all preferences over all source-destination pairs in the switch. Because the number of all possible source-destination pairs is dependent on the dimensions M and N of the switch, α_avg and β_avg are functions of M and N. α_avg is represented by the ratio of the sum of all αs to the total number of source-destination pairs after preference evaluation. β_avg is defined similarly. Let the source-destination pair be noted as L(S,D), where S is the source and D is the distenation, and the result of preference evaluation as α(S,D) or β(S,D) then α_avg and β_avg are as listed in Equations (9) and (10) respectively.

α_avg = { total # of α(S,D) } / { total # of L(S,D) } (9)

β_avg = { total # of β(S,D) } / { total # of L(S,D) } (10)

D is one of the nodes in row N-1 of the switch. L(S,D) is equal to M^2N. From the above, the values of α and β then become as listed in Equations (11) and (12) respectively.

$$\alpha = \frac{\sum_{n=0}^{N-1}(M - 2n - 1)M + \frac{1}{2}\sum_{n=0}^{N-2} 2M}{M^2 N} \quad (11)$$

$$\beta = \frac{\sum_{n=0}^{N-2}(2n + 1)M + \frac{1}{2}\sum_{n=0}^{N-2} 2M + M}{M^2 N} \quad (12)$$

In Equations (11) ans (12) M and N are the dimensions of the switch.

5. EXPERIMENTAL RESULTS

Test-benches were created to simulate the model by instantiating the traffic model and the SMoDS. The simulation results are compared with the simulation results using OPNET tool.

In the first experiment we studied the effect of varying the packet generation rate on the link utilization. The results are shown in Figure 6. As the packet generation rate increases the link utilization increases. These results agree with our expectations.

In the second experiment we studied the relation between the packet loss ratio and the link utilization. Figure 7 shows this relation. As shown in the figure, as long as the links are not highly utilized, there is almost no packet loss. Packet loss starts to be introduced when link utilization increases. A significant packet loss starts to be introduced when the links are utilized beyond 60%. The results of the second experiment suggest that the performance of the SmoDS will be acceptable as long as the links of the interconnection network structure are not very highly utilized.

In the third experiment we studied the effect of the link utilization on the packet delay for the 8X3 and 16X4 switch configurations. The results are shown in Figure 8.

In the last experiment we studied the effect of the link utilization on the deflection ratio of packets. We used an 8X3 switch. Figure 9 shows the results. By varying the link utilization from 0% to 40% the deflection ratio increases exponentially. The results of the OPNET model are close to those of the VHDL model.

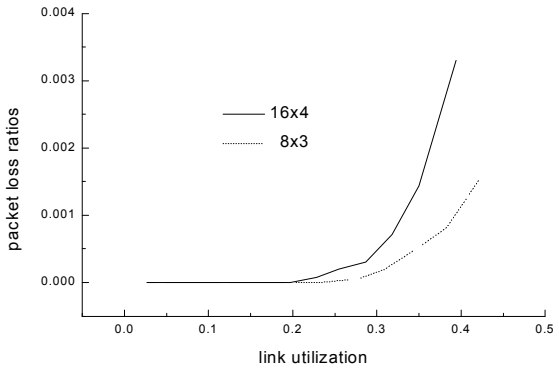

Figure 6: Link utilization vs. Packet generation rate

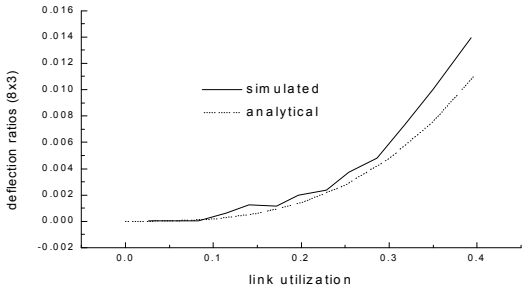

Figure 7: Packet loss ratio vs. link utilization

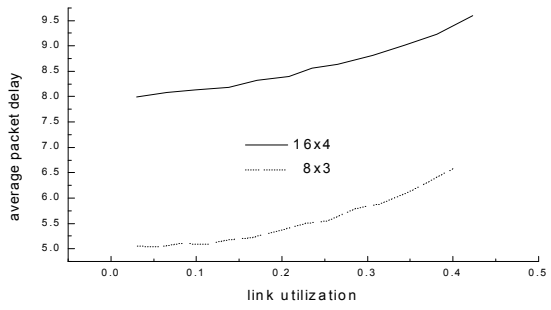

Figure 8: Average packet delay vs. link utilization

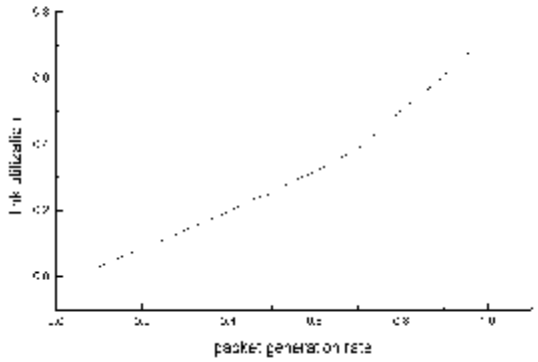

Figure 9: Deflection ratio vs. link utilization for the 8X3 switch

6. CONCLUSION

In this paper we develop a VHDL-based scalable multimedia on demand server model. The three server subsystems: The control subsystem, the communication subsystem and the storage subsystem are modeled.

The control subsystem contained all information needed for managing and controlling the server and clients. The admission controller that controls the admission of clients to the system is included as part of the control subsystem. Whenever a new client requests service, the admission controller decides whether it admits the request or not, depending on system resources such as available network bandwidth and I/O bandwidth. A real-time communicator is also included in the control system.

The communication subsystem is designed as an interconnection network with meshed-pipe structure and delivers packets to their destination through the sub-optimal shortest path. A model for the video traffic generated at the server side for modeling the data sent to the users is also presented.

The storage subsystem consists of a cluster of processors with each processor physically attached to local disks. The proposed SMoDS distributes all video streams to all attached data pumps instead of storing a whole movie in a single processing element.

To validate our model we compared simulation results from our model with simulation results obtained using OPNET simulation tool. Test-benches were created to simulate the model by instantiating the traffic model and the SMoDS.

Three experiments were designed to study the effect of varying the packet generation rate on the link utilization, the effect of the packet loss ratio on the link utilization, and the effect of the link utilization on the packet delay for the 8X3 and 16X4 switch configurations. As the packet generation rate

increases the link utilization increases. As long as the links are not highly utilized, there is almost no packet loss. Packet loss starts to be introduced when link utilization increases. A significant packet loss starts to be introduced when the links are utilized beyond 60%. The results of the experiments suggest that the performance of the SmoDS will be acceptable as long as the links of the interconnection network structure are not very highly utilized. By varying the link utilization from 0% to 40% the deflection ratio increases exponentially.
The results of the OPNET model are close to those of the VHDL model. The comparison results show that the proposed architecture is accurate (within 5% of accuracy). Our modeling approach also shows that using VHDL in modeling and analysis is accurate; this is true because we are using the same VHDL models used to build the systems or components in our modeling and analysis.

REFERENCES

-[Chen 1996] S. Chen, and D. Towsley, "A Performance Evaluation of RAID Architectures", IEEE Transactions on Computers, Vol. 45, No. 10, Oct. 1996, pp. 1116-1130.
-[Dewey 1997] Dewey A. M., "Analysis and Design of Digital Systems", International Thompson Publishing Inc., 1997.
[Eliazov 1990] T. Eliazov, V. Ramaswami, W. Willinger, and G. Latouche, "Performance of an ATM Switch: Simulation Study", In the Proceedings of the Ninth Annual Joint Conference of the IEEE Computer and Communication Societies, INFOCOM 90, June 3-7 1990, pp. 644-659.
-[Feng 1996] F. Feng, C. Li, A. Raha, S. Yu, and W. Zhao, "Modeling and Regulation of Host Traffic in ATM Networks", In the Proceedings of the 21st IEEE Conference on Local Area Networks, Oct. 13-16 1996, pp. 45 8-467.
-[Greenberg 1993] A. Greenberg, and Jonathan Goodman, "Sharp approximate Models of Deflection Routing in Mesh Networks", IEEE Transactions on Communications, Vol. 41, No. 1, Jan. 1993.
-[Krikelis 1998] A. Krikelis, "Scalable Multimedia Servers", IEEE Parallel and Distributed Technology, Vol. 64, Oct-Dec. 1998, pp. 8-10.
-[Lin 1993] X. Lin, P. McKinley, and A. Esfahanian. "Adaptive Multicast Wormhole Routing in 2D Mesh Multiprocessors", In the Proceedings of 1993 Parallel Architectures and Languages Europe Conference (PARLE'93), Munich, Germany, June 1993, pp. 228-241.
-[Park 1996] D. Park, J. Shim and et. al., "A real Time Micro Kernel Implemented on Transputer", In the Proceedings of the International

Conference on Parallel and Distributed Processing Techniques and Applications, Aug. 1996, pp. 1106-1117.
-[Robinson 1994] D. F. Robinson, P. K. McKinley, and B. H. Cheng, "Optimal Multicast Communication in Wormhole-Routed Torus Networks", In the Proceedings of the 1994 International Conference on Parallel Processing, St. Charles, Illinois, August 1994.
-[Serpanos 1998] D. N. Serpanos, L. Georgiadis, and T. Bouloutas, "MMPacking: A Load Balancing Algorithm for Distributed Multimedia Servers", IEEE transactions on Circuits and Systems for Video Technology. Feb. 1998, pp. 13-17.
-[Skahill 1996] Skahill, K., "VHDL for Programmable Logic", Addison-Wesley Publishing Inc., 1996.
-[Wright 2001] Wright, W, "An Efficient Video-on-Demand Model", IEEE Computer, May 2001, pp. 64-70
-[Fei 1997] Fei X. and Shi P., "Construction of Multimedia Server in a Distributed Multimedia Systems", 1997 IEEE, pp. 248-252
-[Cao 1999] Cao F., Smith J., and Takahashi K., "An architecture of Distributed Media Servers for Supporting Guaranteed QoS and Media Indexing", 1999 IEEE International Conference on Multimedia Computing and Systems, 1999 Vol. 2, pp. 1-5
-[Fabbrocino 1998] Fabbrocino F, Santos J., and Muntz R, "An Implicitly Scalable, Fully Interactive Multimedia Storage Server", Second International Workshop on Distributed Interactive Simulation and Real Time Applications, July 1998, pp. 92-101
-[Sonah 1995] Sonah B, Ito M., and Neufeld G., "The Design and Performance of a Multimedia Server for High-Speed Networks", 1995 IEEE, pp. 15-22