PERFORMANCE MODELING AND ANALYSIS OF A
LAN USING VHDL

Praveen Chawla, Wenying Zhou and Herbert L. Hirsch
MTL Systems, Inc.
3481 Dayton-Xenia Road
Dayton, OH 45432-2796
pchawla@mtl.com, wendy@mtl.com, hhirsch@mtl.com

ABSTRACT

An ability to model and simulate a Local Area Network (LAN), consisting of components that interact frequently and execute concurrently, is critical in predicting its dynamic behavior. An accurate prediction of LAN behavior is required in order to assess capacity requirements and implications for additions to the LAN. VHDL is well-suited for this purpose and has been successfully employed to model and analyze LAN performance for a commercial organization. We describe generic VHDL models that can be utilized to construct a LAN model. In addition, we demonstrate VHDL's effectiveness by comparing the results of a baseline LAN simulation with simulations of four LANs obtained through modifications to the operational load and configuration of the baseline.

1.0 INTRODUCTION

With the advent of Client-Server computing, LANs have become an essential part of the computing environment. LANs typically consist of the following:

A. Several clients operating through workstations, personal computers and terminals,

B. A few servers which provide computing horsepower and storage capacity by employing very powerful computers and large disk drives, and

C. An interconnection network which connects clients and servers, allowing clients to utilize servers for their computing and storage needs.

There are a variety of LAN configurations possible depending on the number and function of clients and servers and the manner and media by which clients and servers are connected. The performance of a LAN will vary based on the configuration employed. In addition, operational loads will have a great impact on LAN performance. To efficiently operate a LAN and plan future additions to it, it is important to have an ability to model and analyze it accurately. Such an ability will lead to the following:

A. A determination of how well needs and expectations of LAN users are met.

B. Appropriate selection, design, upgrade, tuning and analysis.

C. A measurement of responsiveness, effectiveness and failure-related behavior.

Although performance evaluation can be done through various means [8,9], this paper will only address the modeling and analysis scheme based on discrete event simulation. Under this scheme, a performance model is created for each system
component. These components are interconnected to create a performance model of the system. Simulation of the system model, driven by an appropriate abstraction of the workload, leads to generation of performance statistics, such as throughput, latency and utilization. Statistical techniques are then employed to analyze the data and make appropriate decisions.

There have been several performance modeling and simulation efforts in the recent past which have employed VHDL for a variety of applications [1,2,3,4,5,6,10,11]. This paper differs from previous efforts in the application it deals with and the level of abstraction utilized for modeling. In this paper, we report on a tool we call Simulator for Predicting Performance of a LAN (SimPLAN). SimPLAN constructs a LAN model from an interconnection of the following generic component models:

A. Stochastic models of clients,
B. Stochastic models of servers,
C. A model of simulation coordinator, and
D. Behavioral models of interconnection network components.

We describe generic components employed by SimPLAN and demonstrate their ease-of-use and flexibility through an example. In the example, we demonstrate effectiveness of SimPLAN through its use for the analysis of a baseline LAN under four possible modifications to its configuration and operational load. We analyze the simulation results to assess capacity requirements and implications of different scenarios on LAN performance.

We have organized the rest of the paper into four sections. Section 2 provides a high-level overview of SimPLAN. Section 3 describes generic components employed by SimPLAN. Section 4 illustrates SimPLAN's effectiveness through an example. Section 5 presents our anticipated future work.

2.0 SimPLAN OVERVIEW

SimPLAN exists today only at a conceptual level. However, the methodology underlying the tool has been tested through its application to real life applications. In this section, we first provide a brief description of our methodology. Then, we present organization of SimPLAN as we envision it.

2.1 METHODOLOGY

Figure 1 shows the top level view of a LAN. As shown in the figure, a LAN consists of several network stimulants, which communicate with each other by sending and receiving packets of information. The stimulants are of two kind: clients and servers. The stimulants are interconnected by hardware components like cable segments, switches, and routers.

The basic unit of information is a packet. There is a finite delay in sending a packet from a source stimulant to a destination stimulant because it takes a finite amount of time to travel on the network hardware. The delay will vary based on the distance between source and destination stimulants, the speed of hardware components and the network protocols employed to exchange information. A packet can collide with another packet while traveling towards its destination. A collision results in loss of colliding packets. Therefore, a re-transmission of colliding packets may be required in
order to exchange information correctly. However, the way collision is handled in a LAN will depend on the protocol utilized for exchanging information.

Our methodology, stated in simple terms, is to first construct performance models of primitive elements and then hierarchically compose a LAN model from component models. The generic LAN model constructed in this manner is then instantiated with specific values to obtain the desired simulation-ready model. Instantiation of a LAN model requires values to define behavior of stimulants and hardware components.

![Diagram of LAN model](image)

**Figure 1. Top-Level View of LAN**

Figure 2 shows structure of a LAN model under SimPLAN. In such a structure, a model is created for each network stimulant. In addition, models are created for hardware components collectively termed as interconnection network model. All the component models created are then connected to a simulation coordinator shown in the figure. The simulation coordinator is generic in nature and therefore, can be reused for each LAN model. However, upon instantiation, it will differ in terms of physical locations of the stimulants and the protocols utilized for exchanging information.
Our methodology promotes reuse of already existing component models since the models are constructed to be generic in nature. Therefore, if a previously defined model is utilized, it only needs to be instantiated with the right values in order to create a simulation-ready model.

**Figure 2. SimPLAN Model Structure**

### 22 ORGANIZATION

Figure 3 shows the SimPLAN organization. The organization shown in this figure is mainly conceptual and an implementation of the tool has just begun. As shown in the figure, SimPLAN in turn utilizes five tools, namely:

A. Definition and Composition Tool  
B. Instantiation Tool  
C. VHDL Analyzer and Simulator  
D. Statistical Analysis Tool  
E. Visualization Tool

**Figure 3. SimPLAN Organization**

A user can select one or more of these tools through a Window, Icon, Menu, and Pointing Device (WIMP) based Graphical User Interface (GUI). Each of the constituent tools, in turn, also utilize a WIMP-based GUI. The following provides a brief description of each tool.

**Definition and Composition Tool:** This tool is utilized to define performance models of primitive components. In addition, it is used to hierarchically compose system components models and a LAN model from already existing models.
**Instantiation Tool:** This tool is used to provide specific values pertaining to a LAN model created using the Definition and Composition Tool. Values for parameters such as mean, type of distribution, protocol definition, and physical location of stimulants, is provided during instantiation.

**VHDL Analyzer and Simulator:** A VHDL analyzer is used to check syntactic and semantic correctness of the LAN model and a simulator is utilized to simulate the LAN model.

**Statistical Analysis Tool:** This tool is used to statistically analyze the simulation results.

**Visualization Tool:** This tool is utilized to visualize the performance data.

A typical sequence of actions consists of the following:

A. Select Definition and Composition Tool,
B. Define primitive component models,
C. Compose system component models using primitive component definitions,
D. Compose a LAN model using system component models,
E. Close Definition and Composition Tool,
F. Select Instantiation Tool,
G. Instantiate LAN model with specific values,
H. Close Instantiation Tool,
I. Select VHDL Analyzer and Simulator Tool,
J. Analyze and Simulate LAN Model,
K. Close VHDL Analyzer and Simulator Tool,
L. Select Statistical Analysis Tool,
M. Analyze simulation results,
N. Close Statistical Analysis Tool,
O. Select Visualization Tool,
P. Visualize performance data, and
Q. Close Visualization Tool.

### 3.0 COMPONENT DESCRIPTIONS

As shown in Figure 2, SimPLAN employs three kinds of components:

A. Network stimulants
B. Interconnection network components
C. Simulation coordinator

These components are created in a generic manner. Therefore, their models can be configured for a particular network under analysis by providing the physical locations of stimulants, the characteristics of stimulants, the protocols utilized for exchanging information, and the types of hardware components employed by the interconnection network. In the following subsections, we describe each kind of component models. Subsection 3.1 describes the network stimulants. Subsection 3.2 provides the behavioral models of interconnection network hardware components such as a cable segment model, and a switch model. Subsection 3.3 presents the description of a simulation coordinator.
3.1 NETWORK STIMULANTS

The network stimulants provide a realistic and accurate representation of network loading to properly represent the operational environment under simulation. Network stimulants are of two kind: clients and servers. Clients send request packets to servers via the interconnection network. Packets are put in FIFO request queues when they reach their designated servers. Servers monitor their request queues continuously and respond to packets in their queues by sending a set of packets back to the corresponding clients. A packet from a network stimulant consists of at least the following:

A. Sender Id, B. Receiver Id,
C. Packet Number, D. Creation Time,
E. Size, F. Collision Flag, and
G. Received Flag

Collision flag is set when a packet collides with another packet. Received Flag is set when a packet has been received by the destination network stimulant. Clients and servers are responsible for distinct tasks as detailed below.

Client - A client performs two tasks, namely sending packets and receiving packets. In the process of sending packets, a client first sends a ready signal to the simulation coordinator indicating that it is ready to put out a packet. Then the client waits until it receives the network-ready signal from the simulation coordinator. The client then sends a packet of random size determined by an exponential distribution to a server chosen randomly through a probability distribution. The client will return to the previous step again after waiting for a random interval based on an exponential distribution. While sending packets to the network, clients also collect statistics on number of packets sent, and average wait time for a client between two consecutive send operations. To receive packets, a client first waits for the packets from the servers to arrive. Then, it collects statistics on the number of packets received and the response time.

Server - Servers only send packets in response to request packets from clients. Upon arrival of a request packet at the server, it is stored in server’s FIFO request queue. Statistical data on number of packets received is collected during this process. If the request queue is non-empty, the server processes the earliest request from its FIFO queue. Processing of a request starts with sending of a ready signal to the simulation coordinator after a random amount of time determined by an exponential distribution. The server then waits until the network-ready signal is received. Upon receipt of a network-ready signal, the server sends a random number of packets based on an exponential distribution to the requesting client. Servers collect statistical data on the average service time and the number of packets received from various clients.

3.2 INTERCONNECTION NETWORK

The interconnection network model is based on the interconnection protocol and the physical media. It includes a structural model of the network and behavioral models of constituent hardware components. From an operational viewpoint, the network keeps status record of each interconnection segment in terms of busy or idle and provides the status of segments to the simulation coordinator upon request. When packets are received by the network, they are either transmitted to their designated network stimulants in a delay based on the interconnection protocol, or are reported lost due to packet collision. The SimPLAN currently employs two generic hardware
component models: a cable segment model, and a switch model. Each of these component models is discussed in detail below.

**The Cable Segment Model** - A cable segment is essentially a wire segment which transmits information in the form of packets from one end to the other after a delay. Each segment has a status flag associated with both its ends. When a network packet is received by a segment, the busy flag is raised on its corresponding end to prevent other packets being placed on the segment before it finishes the current transmission. The packet on the segment is then transmitted out to the opposite side after a delay based on the type of physical media and the size of the packet. When a packet is received by the idle side of the segment before current transmission is completed, a collision is assumed to have occurred and the packet collision flag is raised for colliding packets. Each of the colliding packets is either re-transmitted or lost, based on the protocol used for packet transmission. For example, if ethernet protocol is used, packets are re-transmitted after a random amount of time determined using an exponential distribution. Under ethernet protocol, if a packet on the segment collides over sixteen times, the packet is assumed to be lost. A cable segment also collects statistical data such as the number of collisions, the total traffic on the segment and the number of successfully transmitted packets.

**The Switch Model** - A network switch performs two tasks:

A. Receiving packets from any of the configured ports which are connected to cable segments and putting them in the corresponding request queue.

B. Checking the designated segment status continuously for each of the packets in its request queue until the destination segment is ready and sending the packets to their designated segments when the corresponding segment is found idle.

The switch model assumes that there is zero delay in transmission of a packet through the switch unless the designated segments are busy. In addition, it is assumed that no collisions occur in the switch.

### 3.3 Simulation Coordinator

The simulation coordinator is the overall executive element for SimPLAN. It includes the physical locations of network stimulants, as well as definitions of network protocols. The tasks performed by the simulation coordinator are as follows:

A. Monitor the ready-to-send signals from network stimulants. When a ready signal is received, convert the corresponding ID of the sending network stimulant into a physical address on the network. Check the status of segment corresponding to the physical address obtained. If the corresponding status is idle, send network-ready signal to the appropriate network stimulant. If the status is busy, monitor segment status till it becomes idle.

B. Intercept packets sent by network stimulants. Translate IDs of stimulants into physical locations and transmit them to corresponding segments.

C. Intercept packets sent by network components. Translate physical addresses of stimulants into IDs and transmit them to corresponding stimulants.

Figure 4 shows the data and control flow for various network models as described in this section.
4.0 AN EXAMPLE CASE

In this section, we demonstrate SimPLAN’s utility and its underlying methodology by modeling, simulating and analyzing performance of a LAN and its four variations. Subsection 4.1 describes the example case and its four variations. Subsection 4.2 summarizes the results of the performance analysis.

4.1 DESCRIPTION
As shown in Figure 5, we modeled an example LAN that resides in a building with two floors. Floor 0 has one client, which in turn consists of several workstations, and three servers: a home server, a specialized server, and a binary server. Floor 1 includes a client, a home server, and a binary server. Binary server on floor 0 is assumed to be three times faster as the binary server on floor 1. A switch and some cable segments interconnect the clients and servers as shown. We hierarchically composed a LAN model for the example case with the generic models.

Figure 5. An Example LAN Configuration
provided by SimPLAN. Four variations to the baseline configuration, as described above, were conceived for demonstration. The four variations are described below.

**SERVER DISTRIBUTED (SD):** It is assumed that there is a 25% extra network traffic due to introduction of an additional application. Furthermore, it is assumed that users of the application send request packets to binary servers, which, in turn, cause responsive computations at the servers. The computation results are sent back to the users, who in turn save it on the home server on their floor. As a consequence, mean service times at the binary servers would increase. A 50% increase in the service times for the binary servers is assumed due to the new application. In addition, it is assumed that 90% of the additional traffic is directed towards binary servers and 10% of the additional traffic is directed towards home servers.

**SERVER CENTRALIZED (SC):** It is assumed that there is a 25% extra network traffic due to introduction of an additional application. Furthermore, it is assumed that users of the application send request packets only to Binary 0 server, which causes the computation of desired results at Binary 0. The results are sent back to the users, who in turn save it on the home server on their floor. As a consequence, mean service time at Binary 0 would be expected to increase. A 150% service time increase is assumed for the Binary 0 server due to additional traffic. In addition, it is assumed that 90% of the additional traffic is directed towards Binary 0 and 10% of the additional traffic is directed towards home servers.

**WORKSTATION DISTRIBUTED (WD):** This case is similar to Server Distributed except that the computation caused by the new application is handled by the workstations themselves and only application binary is obtained from the server. Therefore, mean service times at the binary servers do not increase.

**WORKSTATION CENTRALIZED (WC):** This case is similar to Server Centralized model except that the computation caused by the new application is handled by the workstations themselves and only application binary is obtained from the Binary 0 server. Therefore, mean service time at the Binary 0 servers do not increase.

### 4.2 ANALYSIS

The four variations to the example LAN configuration were analyzed relative to the baseline configuration. The simulation data obtained was used to calculate percentage changes in average client response time, total segment traffic, and number of packet collisions. Table 1 lists the percentage changes for each of the test cases as compare to the baseline model.

<table>
<thead>
<tr>
<th>Description</th>
<th>SD</th>
<th>SC</th>
<th>WD</th>
<th>WC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Time</td>
<td>554%</td>
<td>168%</td>
<td>380%</td>
<td>-9%</td>
</tr>
<tr>
<td>Segment Traffic</td>
<td>-5%</td>
<td>6%</td>
<td>7%</td>
<td>9%</td>
</tr>
<tr>
<td>Collision</td>
<td>18%</td>
<td>-16%</td>
<td>-15%</td>
<td>66%</td>
</tr>
</tbody>
</table>

With the exception of Workstation Centralized, a general increase in response times and a re-distribution of response times over the network components is observed which would be an expected result of increased loading upon the network. The Workstation Centralized case shows a better response time because Binary 0 server is much faster as compared to Binary 1 server. With the exception of the Server Distributed case, segment traffic shows a general increase. The Server Distributed case shows less traffic due to
an increase in the service times for the binary servers. Such an increase causes accumulation of packets from clients in the server request queues and a net reduction in the traffic due to the slower response from servers. Server Centralized case shows an increase though because Binary 0 server is much faster than Binary 1 server. Average number of collisions varies depending on the traffic distribution over the timeline. Collisions show the highest increase in the Workstation Centralized case.

With the example above, we have attempted to demonstrate SimPLAN's effectiveness. We have shown that SimPLAN could be used to do what-if kind of analysis given a LAN model. Furthermore, we have shown that SimPLAN could be utilized to predict dynamic behavior of a LAN as the operating conditions and configurations are varied.

5.0 FUTURE WORK

In the future, we will continue to implement SimPLAN. Specifically, we intend to implement an X-based GUI, select a commercial VHDL analyzer and simulator, select a commercial visualization tool, select a statistical analysis tool, and completely implement the definition and composition tool and the instantiation tool.

ACKNOWLEDGMENTS

We acknowledge and thank Dr. Harold W. Carter, of the University of Cincinnati, for providing helpful comments and suggestions.

REFERENCES


