3-1996

Performance Study of Shared Versus Nonshared Bandwidth on a Packet-Switched Network

John P. Stevens

PERFORMANCE STUDY OF SHARED VERSUS NONSHARED BANDWIDTH ON A PACKET-SWITCHED NETWORK

THESIS

John P. Stevens
Captain, USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENG/96M-01

PERFORMANCE STUDY OF SHARED VERSUS NONSHARED BANDWIDTH ON A PACKET-SWITCHED NETWORK

THESIS

John P. Stevens
Captain, USAF

AFIT/GOR/ENG/96M-01

Approved for public release; distribution unlimited.

AFIT/GOR/ENG/96M-01

PERFORMANCE STUDY OF SHARED VERSUS NONSHARED BANDWIDTH ON A PACKET-SWITCHED NETWORK

THESIS

Presented to the faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science (Operations Research)

John P. Stevens, B. S.

Captain, USAF

March, 1996

Acknowledgments

This research investigated performance characteristics of a wide-area network using shared bandwidth versus nonshared bandwidth. The methodologies developed were the result of six months of research into network communications. I would like to thank my thesis advisor, Dr. Richard Raines, for his technical expertise and guidance. I would also like to thank my readers, Dr. Yupo Chan, and Dr. Robert F. Mills for their insights and suggestions. Finally, I want to thank my wife, Leonie, and my three children, Jennie, Joseph, and Joshua for their support and understanding during these 18 months.

Table of Contents

List of Figures

List of Tables

Abstract

In wide-area computer data communications, many networks have evolved by satisfying increased user demands in the most expedient manner. In some cases, new users' demands are satisfied by installing a new link, rather than sharing the links that are already in place. This research investigates the differences in performance between using a dedicated link for each source-destination pair (nonshared bandwidth) and using a single link to be used by all source destination pairs (shared bandwidth). Simulation models are developed for a wide-area network using shared bandwidth, and a wide-area network using nonshared bandwidth. The quality of service offered by each network is based on its responsiveness and productivity. Responsiveness will be measured in terms of average end-to-end delay of packet transmission, and productivity will be measured in terms of percent bandwidth utilization. The networks are modeled under a common set of operating assumptions and system environment. This allows for accurate comparison of packet delay and bandwidth utilization. Two variable input parameters are used in the simulation: intensity of input traffic load, and amount of link capacity. Provided that the intensity of the input traffic load remains below the network saturation level, it is shown that the shared system clearly outperforms the nonshared system. This result occurs for both a uniform and a nonuniform traffic load destination distribution.

Performance Study of Shared Versus Nonshared Bandwidth on a Packet-Switched Network

1 Introduction

This chapter provides a clear understanding of the problem and the approach which will be taken to solve the problem. Before the problem can be defined in a clear manner, some background material must be presented, and is done so in Section 1.1. Next, the problem definition, Section 1.2, attempts to shed light on the problem by use of an illustration and an explanation of what nonshared and shared bandwidth mean in the context of this research. Following the problem definition is the scope of the research. In regards to the approach to be taken to solve the problem, a brief plan of attack is presented in Section 1.4. And finally, prior to closing out this chapter, a summary will be presented.

1.1 Background

The Department of Defense (DoD) has a global communications network which consists of several intermediate switching nodes interconnected by various links, such as satellite, fiber, and microwave. This network enables world-wide transmission of video, facsimile, voice, images, and all types of computer data. Over the years, the network has evolved, and many upgrades have been implemented to satisfy increased user demands. Specifically, the DoD has installed additional higher capacity links, replaced older switching nodes with high-speed switching nodes and has implemented new technologies, such as Fiber Distributed Data Interface (FDDI). As a result, the network has grown in complexity at a dramatic rate. The concern that now faces the DoD is whether or not the network is operating efficiently. Specifically, the DoD suspects that bandwidth (the capacity available in the links) is not being utilized efficiently and that some of the links which have been purchased/leased are possibly unnecessary. The DoD feels that the network can be reconfigured in a way which will reduce the amount of links currently being leased, and at the same time, meet on-going user demands.

1.2 Problem Definition

The links used in the DoD's Global Area Network (GAN) can be configured two different ways: Dedicated or Circuit Switched. In the dedicated mode of operation, a link is dedicated to a specific source-destination pair. No other user can use this link (bandwidth), even if it is sitting idle (not in use by the designated source-destination pair). In the circuit switching mode, the links are configured to allow other source-destination pairs to use the link when it is not in use. It thus can be said that the link (bandwidth) can be shared by the users, although not simultaneously. However, another very important consideration must be taken into account when determining whether or not a link is shared or nonshared. If the source and/or the destination is a Local Area Network (LAN) or multi-user computer system, then the dedicated link can be considered to be shared by those users specific to the LAN and/or multi-user computer system. In fact, in data communications, it is very common for several source-destination pairs to share a common link. These type of networks normally use packet-switching technology, in which blocks of data called packets are transmitted from a source to a destination.

The small sample network shown in Figure 1.1 represents a high level view of the DoD's GAN configuration. Some of the links shown are configured dedicated point to point, others such as those carrying voice traffic may be operating in a circuit switched mode, similar to commercial telephone networks. As previously mentioned though, some of the dedicated links carrying computer data traffic have a source/destination consisting of a LAN and/or multi-user system.

The major thrust of this research will focus on computer data traffic. In the nonshared mode, the links between the Packet Switching (PS) nodes (via Circuit Switched (CS) nodes) can only pass traffic from a designated source/destination pair. For example, a LAN at site A can communicate with a LAN at site B, but it could not communicate with a LAN at site C. In order to do so, the PS node at site B would have to be able to route the data, or a separate link from site A to site C would have to be installed. Each link, whether dedicated or circuit switched, has a capacity (bandwidth) assigned to it which specifies the maximum rate at which traffic can flow across it. The DoD would like a methodology developed which they can employ on their GAN to see if the bandwidth within the links (as well as bandwidth available by sometimes idle links) can be more efficiently utilized if all links operated in a shared mode.

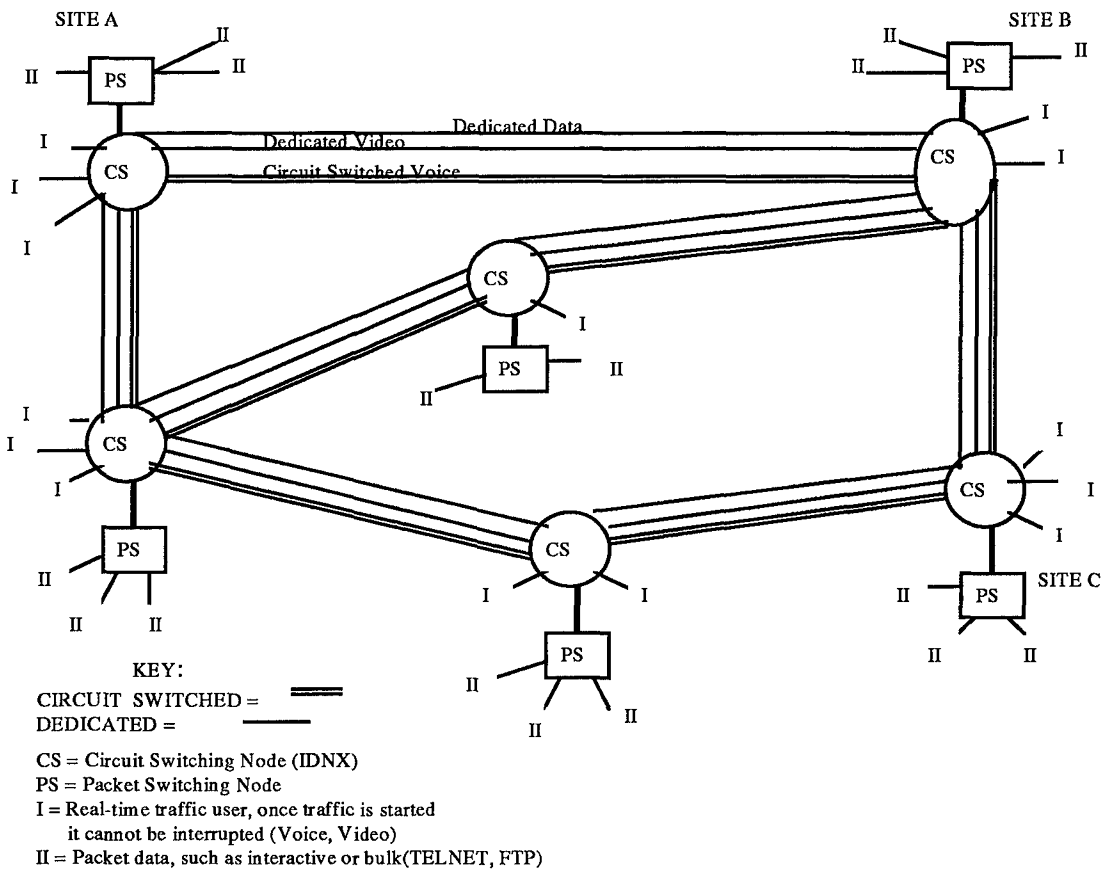

Figure 1.1 High-level view of DoD network topology

This investigation will explore possible methods which can be used to increase bandwidth utilization in a wide area network. The overall goal can be stated as follows: analyze and compare the difference between a shared versus nonshared system.

1.3 Scope

The major thrust of this research will focus on data traffic using dedicated links for nonshared bandwidth and a packet-switched network for shared bandwidth. Real-time traffic will be briefly discussed in the literature review under different methods of sharing bandwidth, but will not be discussed any further. Since the Transmission Control Protocol / Internet Protocol (TCP/IP) is the most predominantly used protocol in the DoD GAN, it will be the only protocol covered in this investigation. Integrated Services Digital Network (ISDN) and newer technologies, such as Broadband ISDN, Asynchronous Transfer Mode (ATM), and Synchronous Optical Network (SONET) will not be covered, except for a brief discussion in the literature review on different methods for sharing bandwidth.

1.4 Approach

The research effort will begin with a literature review. In the literature review, different methods currently being used to share bandwidth will be examined. Some of the major technical issues pertaining to packet-switched networks will be discussed. How networks can be modeled, and traffic patterns particular to packet-switched data with emphasis on TCP/IP protocol applications will be explored.

The next step will consist of the methodology. The approach used in the methodology is to first simulate a two node network with two Local Area Networks (LANs) located at each node, and then to simulate a five node network with seven Local Area Networks connected arbitrarily to various nodes.

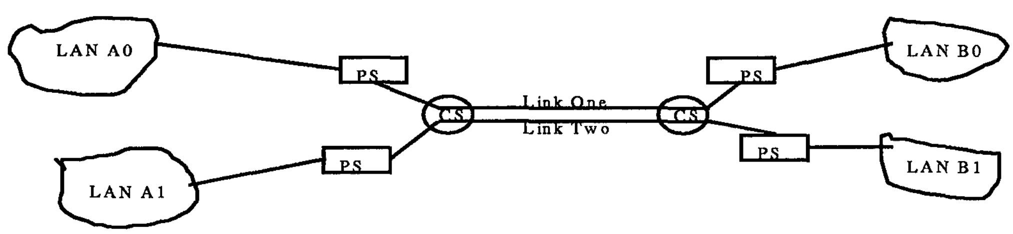

Figure 1.2 Two Node Nonshared System

The two node nonshared system, shown in Figure 1.2, will consist of two separate links, one for each LAN to LAN connection (i.e., LAN A0 to LAN B0 uses link one and LAN A1 to LAN B1 uses link two). In the shared system, shown in Figure 1.3, all four LANs communicate across only one link.

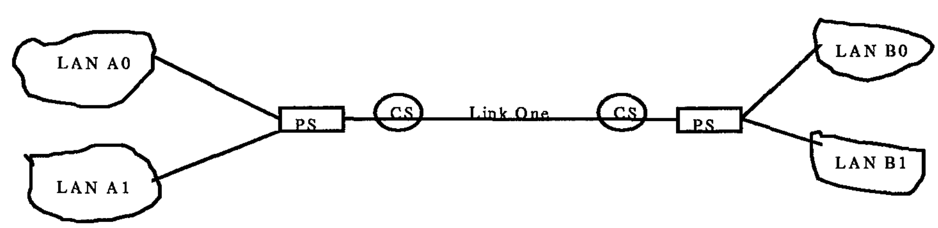

Figure 1.3 Two Node Shared System.

The five node nonshared system, shown in Figure 1.4, will consist of point-to-point links which may travel across store and forward switching nodes and are restricted to a single source-destination pair. In the shared system, shown in Figure 1.5, the links will be configured to allow all source-destination pairs to communicate across them, and routing will be implemented at the nodes. Only packet-switching (PS) nodes are included, as the CS nodes perform no function other than establishing a permanent connection to the links. The topology has been chosen somewhat arbitrarily, but it does, however, represent how a possible series of ongoing demands for data access may have evolved over time. The topology is considered fixed for both the nonshared and shared system. The variable parameters will be peak traffic loads and the capacities of the links. A performance analysis will be conducted to compare shared versus nonshared systems for both the two node and the five node systems. Chapter 4 will discuss the results, and Chapter 5 will present conclusions and future recommendations for further work in this area.

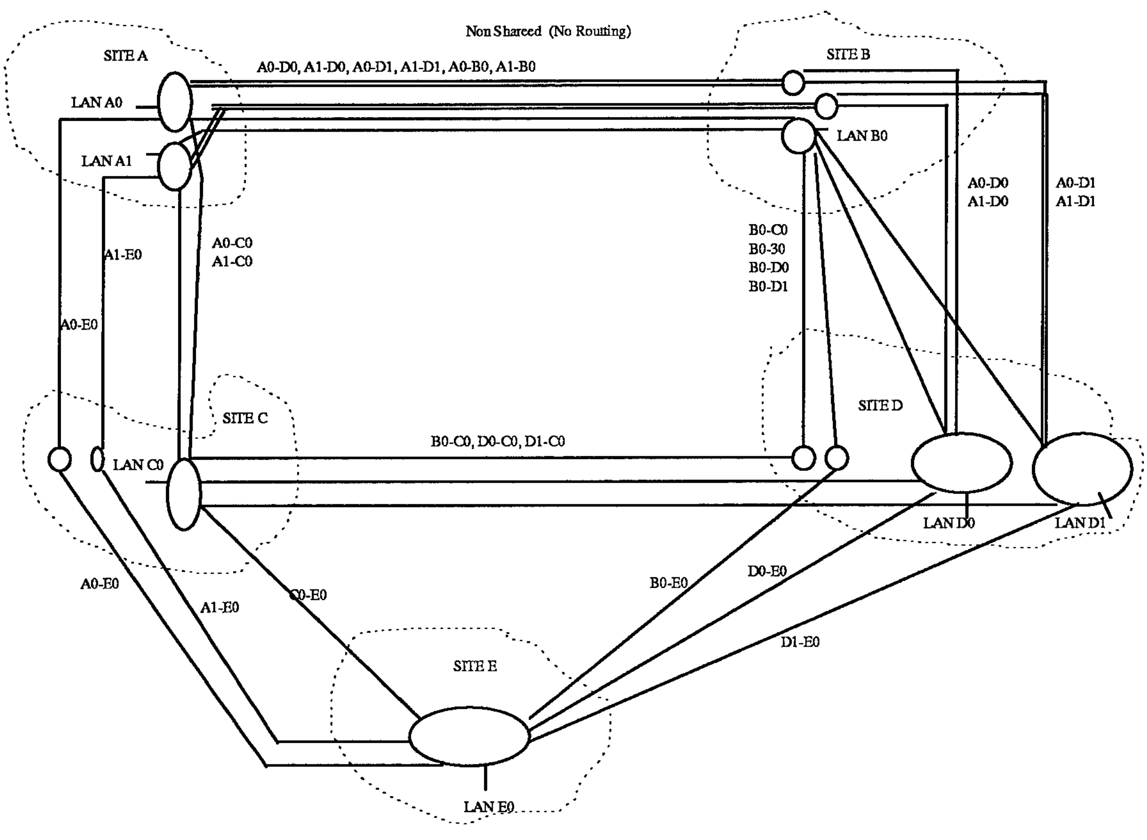

Figure 1.4 Five Node Nonshared System

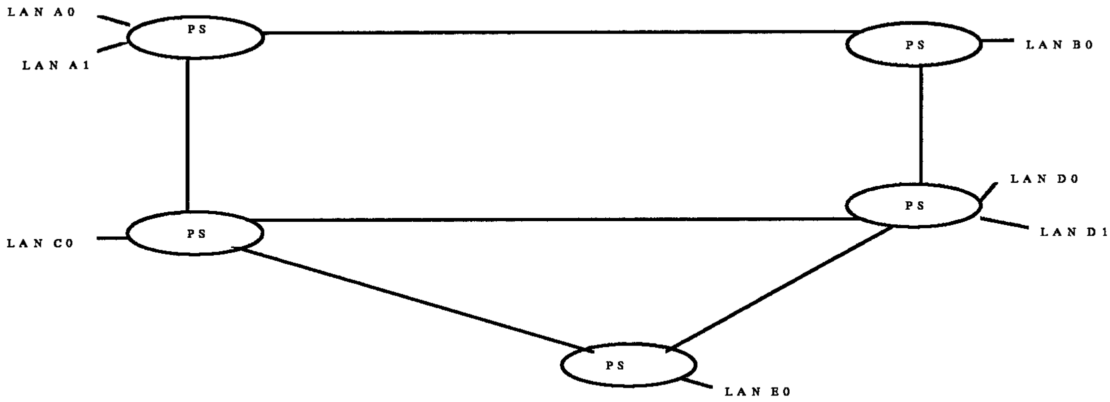

Figure 1.5 Five Node Shared System.

1.5 Summary

The DoD has experienced a rapid growth in their global communications network. They are concerned that bandwidth can be more efficiently utilized if shared bandwidth links are used rather than nonshared links. The scope of the research will focus on the links carrying data traffic only. The plan of attack will be to construct networks of similar topology with one being configured with nonshared bandwidth and the other having shared bandwidth links. Then the shared and nonshared systems performance characteristics will be analyzed and compared.

2 *Literature Review*

2.1 Introduction

This chapter reviews the literature pertinent to modeling and analyzing wide-area networks. To begin with, it may have been noticed in the problem definition that the term 'shared bandwidth' (bandwidth on demand) had been used rather loosely. For this reason, the first section of this chapter has been devoted to clarify the issue of shared vs. nonshared bandwidth in the context of this research.

The next topic to be discussed deals with the technical details of data communications. In order to understand how a wide-area network operates, major technical issues pertaining to packet-switching have to be explored. Section 2.3 describes how the network can be represented. Section 2.4 discusses the major technical issues, such as flow control and routing. Section 2.5 presents issues pertaining to modeling and performance analysis of wide-area networks.

2.2 Shared and Nonshared Bandwidth

Since the introduction of telephone networks in the 1890s, a large variety of dedicated (dedicated to a single telecommunications service) networks have been developed and deployed around the world [SaA94]. The DoD global area network currently employs a dedicated circuit-switched network for voice traffic, dedicated lines for video traffic, and dedicated lines for data traffic [RoK95]. Under the current mode of operation, the bandwidth is nonshared. In order to clarify the distinction between shared and nonshared bandwidth, it is necessary to provide a short discussion on various methods of sharing the bandwidth. The following methods will be described: Integrated Services Digital Network (ISDN), Packet-Switching (not only data, but also real-time traffic), and various multiplexing techniques. These methods are not all inclusive but have the most relevance to this research.

2.2.1 Integrated Services Digital Network (ISDN)

One method which can be used in sharing bandwidth is ISDN. This method allows both voice and data to be sent over a single connection to the network [SaA94]. An example of how the bandwidth is shared can be explained as follows. Assume a user wants to send voice (via digital telephone) and data simultaneously. Further assume that the computer and voice equipment are connected to a Basic Rate Interface (BRI) ISDN connection. A BRI connection allows bi-directional transmission over two independent B channels (user information at 64 kbps each channel) and a separate D channel (signaling at 16 kbps), which are time division multiplexed over a four wire interface. During the connection the user will use 64 kbps of the available 128 kbps for voice, and the other 64 kbps for data. Now, if the user hangs up while sending data, the 64 kbps bandwidth which was used for voice will transparently be allocated to the data connection. Likewise, if the user picks up the phone while transmitting data, the bandwidth allocated to data transmission will be reduced by 64 kbps. ISDN also provides a Primary Rate Interface (PRI) which has a capacity of 1.5 Mbps (23 B channels and one D channel). At the networking layer (Layer 3), ISDN may use packet-switching, circuit-switching or a combination thereof in transmitting information (voice and data) across a network.

A shortcoming of ISDN (also referred to as narrowband ISDN [N-ISDN])is that its capacity is based on the 64 kbps digital rate. Such rates can support a wide range of services; however high-bit-rate services such as image and video services cannot be provided at a rate of 64 kbps. This led to the development of Broadband ISDN (B-ISDN). Examples of services expected to be provided by B-ISDN according to [SaA94] are "full-motion video and high definition television, image, videotelephony, video conferencing, videotex, video surveillance, data, electronic mail, data transactions, voice, video and voice mail, LAN interconnection, and high-speed data communications." All of these services demand high-speed transmission and switching within the network well beyond that provided by current N-ISDN [SaA94]. Emerging technologies, such as the Synchronous Optical Network (SONET) and Asynchronous Transfer Mode (ATM) provide a high-bandwidth, low-delay, packet-like switching and multiplexing which should support these high-bit rate services [SaA94, Sta95,Spo93].

The DoD does not plan to implement these technologies in the near future and is more concerned with ways to save bandwidth using existing networking components [RoK95]. In this research, N-ISDN and B-ISDN will not be used as the method for sharing bandwidth. More information on these protocols as well as on emerging technologies, such as SONET, ATM, and others can be found in [Spo93,Sta95,SaA94,JeL95,HuZ94].

2.2.2 Packet Switching for Data and Real-time Traffic

Another method of sharing bandwidth is to provide both data service and real-time services in packet-switched wide-area networks. A representative study in this area was performed by Ferrari and Verma [FeV90]. In their paper, they assume that a real-time connection is established as a virtual circuit with performance guarantees. Thus, in order to provide real-time service (digitized video/audio), they require that the clients declare the quality of service required (i.e., maximal allowable delay) at the time of channel establishment. The virtual circuit is established by performing tests at each node along the route. The tests check to see if there is sufficient bandwidth available in the links, and to see if the node's processing power and buffer space are adequate to allow a newly requested real-time channel to go through without jeopardizing the performance guarantees given to the already established channels passing through the same node. If any of the tests fail, a message will be sent back to the user, who may decide to wait or try another output link. The authors admit that fast packet switching, such as ATM, is better suited to the type of high-bit rate traffic they envision. The DoD does not plan to implement this technology in the near future and is more concerned with ways to save bandwidth using existing networking components [RoK95]. As such, this method of bandwidth sharing will also not be covered in this research. More information in this area can be found in [FeV90, JeL95,ArK94].

2.2.3 Multiplexing Techniques

Various multiplexed configurations provide other methods of achieving shared bandwidth. One type of multiplexed configuration can support both real-time and non real-time traffic. For instance, Research Triangle Institute [Tay90] set up a T1 system to carry 16 voice circuits (64 kbps each) and a single 448 kbps LAN interconnection between its North Carolina and Washington D.C. locations. This method offers the transparency of circuit switching for real-time traffic and packet switching for data traffic. A certain portion of the link capacity can be allocated to real-time traffic, and the remaining capacity is then assigned to data traffic. Fischer and Harris [FiH76] examined mixing digitized voice with data to allow data traffic to use residual voice capacity momentarily available due to statistical variations in voice traffic. This process is called the movable boundary case. Another such case is one where the boundary is fixed (i.e., data traffic is not allowed to use residual voice capacity) which is equivalent to two separate systems [FiH76]. In summary, multiplexers can be configured to send digital video, voice, fax, and data traffic onto a single leased line or across a wide-area network [MaS81,Kie90,Tay90,Joh93,Tay95].

2.2.4 Definition of Shared and Nonshared Bandwidth in Thesis Context

In this research effort, multiplexing with fixed boundaries will be used in sharing bandwidth. As such, according to [FiH76], this is equivalent to two systems. One system which carries the real-time traffic (i.e., digitized video/voice) using circuit-switching, and the other system will carry data traffic using packet-switching. Further, this separation allows the research to be divided into two areas: 1) incorporating dedicated video, circuit-switched voice, and other possible real-time traffic into a single circuit-switched system; and 2) employ packet-switching with routing for data traffic. Throughout the remainder of this research effort, only data traffic is considered. Comparing the performance of real-time traffic (shared versus nonshared) using circuit-switching techniques will not be covered due to time limitations, and is a possible candidate for follow-on work.

The way in which data traffic is classified as either shared or nonshared is as follows. In the nonshared mode of operation, it is assumed that dedicated links are set up to allow only a single source-destination pair per link. The intermediate nodes have only store and forwarding capability. A source/destination may be either a LAN, a multi-user system, a single terminal, or basically any other network accessible device (only LANs are used in this research). In the shared mode of operation, bandwidth will be shared by all sources and destinations and routing will take place at the packet-switching nodes.

2.3 Packet-Switched Network Representation

A simplified packet switching network is shown in Figure 2.1. In packet switching, the communication systems break a message into variable sized packets. Each packet is transmitted from source to destination over a network containing links and intermediate switching nodes [Kie90]. The nodes in the network act as a gateway between the local area network and the wide-area network. While the arcs may represent communications links between the nodes. Each arc has an associated capacity which is the maximum flow the arc can carry. The flow between the nodes represents the amount of information transmitted in bits per second (BPS).

The intermediate switching nodes can be configured to look at the address fields of each arriving packet so that appropriate routing decisions (link selection) can be made [Kie90]. The computer data traffic flowing across the nonshared WAN system uses packet switching, but routing is not performed. When routing is not performed, the arc acts as a dedicated link which can be used by only one source-destination pair (i.e., LAN A - LAN B).

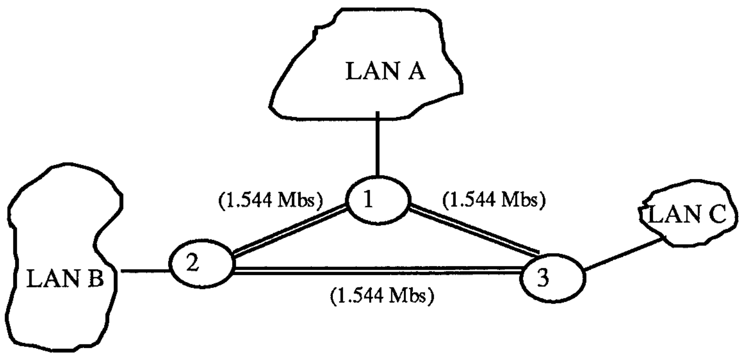

Figure 2.1 Three node packet switching network.

In the shared WAN system, routing is performed. To better understand the operation of the network (assuming shared bandwidth in this case), suppose a user on LAN A wants to send a message to a user on LAN B (refer to Figure 2.1). The message is first broken up into packets and then put onto the network (LAN A) one packet at a time. This step is accomplished by the end-user system's network software and network interface card. The packets travel across the network from the LAN A end-user system to the packet switching node (node 1). When a packet is received by node 1, it's destination address is examined, an appropriate link is selected (in this case the link from node 1 to node 2), and the packet is subsequently forwarded to node 2. After node 2 looks at the packet's destination address, it routes the packet to the end-user system on LAN B. The end-user system on LAN B will reassemble the incoming packets into the original message and store this message so that the user may view it at his/her convenience. The end-user systems and the networking components must all be in agreement on the how the message is transmitted. In fact, a specifically defined protocol must be implemented as discussed in the next section.

2.3.1 Network Protocol

The protocol used for computer data traffic throughout this investigation is a subset of the Transmission Control/Internet Protocol (TCP/IP) suite. Figure 2.2 illustrates how the TCP/IP protocol is layered. The three basic layers of TCP/IP are as follows: application, transmission (TCP), and network (IP). The application is the top layer and consists of services such as file transfer protocol (FTP), simple mail transfer protocol (SMTP) and virtual terminal access (TELNET) [Fei93]. The user can access the applications directly (via operating system) or by use of a graphical user interface (GUI). The application service passes data downwards to the transmission (TCP) layer through a service access point. The TCP layer establishes connections, breaks a message into packets, sends acknowledgments, handles duplicate packets, regulates the flow of data (i.e., implements flow control), detects errors, and terminates connections [Fei93]. TCP is a full-duplex protocol; that is it can send and receive at the same time. According to Feit, "TCP can play a sender role and receiver role simultaneously." The layer below the TCP layer is the network (IP) layer. At this layer, the destination and source network addresses are attached to the packet. From the IP layer, the packets are passed downwards to the network access layer. This layer provides access to the physical media and attaches a physical source and destination address. The network access protocol (NAP) is dependent upon the type of physical media the host is connected to. Additionally, if the size of the maximum transmission unit (MTU) of network 1 is different from network 2's MTU, then the IP layer will fragment the packet to appropriately sized sub packets.

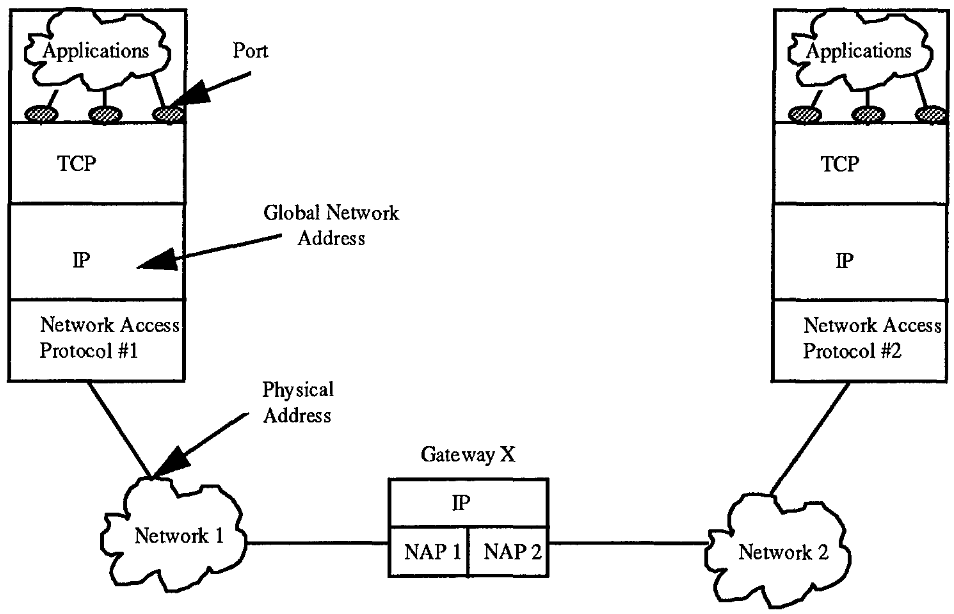

Figure 2.2 Communications using the TCP/IP Protocol [Sta88]

The most popular network access layers that TCP/IP can run over in the LAN environment are High-Level Data Link Control (HDLC), IEEE 802.3, Ethernet, and Token Ring [Fei93]. In the wide-area network (WAN) environment, X.25, T1, and fractional T1 are commonly used. The application, TCP and IP layers as well as the network access layer reside on the end-user systems. The intermediate networking nodes must implement the lower layers, and if routing is employed, the nodes will include the IP layer (i.e., Gateway X). The PSN in Figure 2.2 (Gateway X) acts as a gateway between two local area networks (LANs) where NAP1 may be Ethernet, and NAP 2 may possibly be Token Ring. In the case where the PSN (Gateway) connects a LAN to a WAN (not shown), if the data rate of the incoming packets received from the LAN side exceeds the bandwidth capacity available on the outgoing link of the WAN, then the packets will be stored in a queue until bandwidth becomes available (i.e., the previous packet has finished transmitting). This is the case in the scenario described above under network representation (Section 2.3) where the LAN, for instance, implements Ethernet, which allows a data rate of 10 Mbps, but the WAN T1 bandwidth available may only be 1.544 Mbps. If the packet switching node runs out of buffer space, additional incoming packets will be dropped. More detailed information on the operation of the node is presented in the next section.

2.3.2 Packet-switching Node (Gateway)

Now that the packet-switching network operation has been discussed at the network level, it is now time to look at the functions specific to the packet-switching node. In the literature, packet-switching nodes are sometimes referred to as gateways, and if routing has been implemented, they are sometimes referred to as routers. The packet switching node discussed in this section is a typical node which may be found on the Internet [CoS91]. As explained in the previous section, the packet switching node (router) implements two layers: the network access layer, and the IP layer. These two layers, as well as a general description of the node operation are discussed below.

2.3.2.1 The Network Interface Layer

The network interface layer consists of a device and a device driver (software). There is a separate device and device driver for each link connected to the node. The layer's main function is to transfer incoming packets from the network link to the switching node's memory and inform the processor that a packet has arrived. Other functions include performing mappings from IP addresses to hardware addresses, encapsulating (appending a physical header and possibly a tail), and transmitting outgoing datagrams.

2.3.2.2 The IP layer (shared system)

IP is the central switching point in the protocol software. It uses a routing table to choose a next hop for outgoing datagrams. Other functions include verifying that the datagram checksum is correct. Fragmentation can also occur in the IP layer. If the datagram length exceeds the physical network MTU, then IP will fragment the datagram into two independent datagrams. And finally, IP will generate error messages. If the gateway does not have a route to the specified destination, it must generate an ICMP 'destination unreachable' message. If the routing table specifies that the datagram should be sent to a destination on the network over which it arrived, IP must generate an ICMP 'redirect' message.

Before moving on, an additional note about fragmentation must be addressed. According to Feit [Fei93], the biggest datagram size for a type of network is called its maximum transmission unit or MTU.

For example, Ethernet has an MTU of 1500 bytes, and FDDI has an MTU of 4352 bytes. According to Comer [CoS91], TCP should use a default maximum segment size of 536 bytes when communicating with destinations that do not lie on a directly connected network. Shankar [ShA92] used a MTU of 512 bytes in a study of routing protocols used on the Internet. Like in Shankar's study, the MTU size chosen in this research effort is set to 512 bytes. As such, fragmentation does not need to be implemented.

Routing is the main function of the IP layer. According to Comer [CoS91], routing software can be divided into two groups. One group includes procedures used to determine the correct route for a datagram. The other group includes procedures used to add, change, or delete routes. Because a router must determine a route for each datagram it processes, the route lookup code determines the overall performance of the gateway. Thus, the lookup code is usually optimized for highest speed. Programs that compute new routes communicated with other nodes to establish reachability can take an arbitrarily long time before changing routes. Thus, route update procedures need not be as optimized as lookup operations. Because routing plays a very important role in packet-switched networks, it will be covered in more detail in Section 2.4.3.3.

2.3.2.3 Node Operation

A description of the node's operation is best explained by Comer [CoS91]. When a packet arrives, the network device driver enqueues the packet and notifies the IP process that a datagram has arrived. When the IP process has no packets to handle, it remains in a wait state. As shown in Figure 2.3, there is an input queue associated with each input device, and a single IP process extracts datagrams from all queues and processes them.

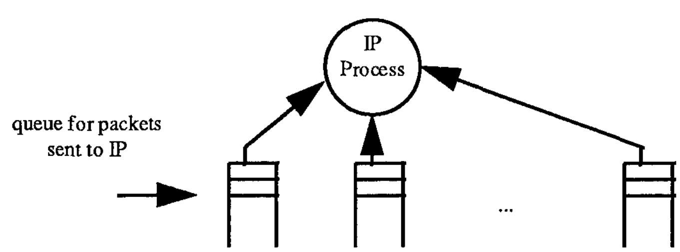

Figure 2.3 Communication between the network devices and the IP process [CoS91]

IP repeatedly extracts a datagram from one of the queues, uses a routing table to choose a next hop for the datagram and sends the datagram to the appropriate network output queue for transmission. If multiple datagrams are waiting in the input queues, the IP process must select one of them to process. The choice of which queue to select determines the behavior of the system. IP can be configured to give a priority to a specified queue, or it can assign priority fairly and allow incoming traffic to be routed with equal priority. One implementation achieves fairness by selecting datagrams in a 'round-robin' manner [CoS91]. This is the policy which will be assumed in this research. That is, the processor selects one datagram from a queue, and then moves on to check the next queue. If N queues contain datagrams waiting to be routed, IP will process one datagram from each of the N queues before processing a second datagram from any of them.

After retrieving a packet from the appropriate queue, IP calls a procedure to compute the next hop address, and then deposits the datagram on a queue associated with the network interface over which the datagram must be sent. This concept is illustrated in Figure 2.4. The network access layer then processes the datagrams for transmission.

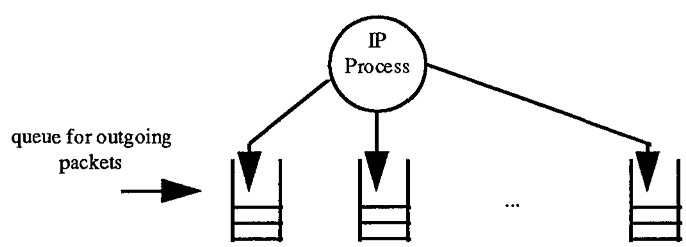

Figure 2.4 Output process showing the path of a datagram sent from the IP layer to the network layer

2.3.3 Connection Oriented vs. Connectionless

A very important characteristic of a packet-switched network is whether it uses datagrams or virtual circuits. According to Stallings, there are basically four modes of operation [Sta95]:

1) External virtual circuit, internal virtual circuit - This is a connection-oriented service in which the end-user systems set up a connection and all data packets transmitted are acknowledged. Furthermore, a dedicated route is set up and all packets travel that same route.

2) External virtual circuit, internal datagram - This is a connection-oriented service in which the end-user systems set up a connection and all data packets transmitted are acknowledged. In this scenario, a dedicated route is not set up. As such, the packets may travel across the network on different routes. A good example of this type of service is TCP/IP, which is a connection-oriented service, but packets are sent using a datagram service (IP).

3) External datagram, internal datagram - Each packet is treated independently from both a user's and the network's point of view. No connection is set up and packets do not get acknowledged. An example of this type of service is User Datagram Protocol (UDP/IP).

4) External datagram, internal virtual circuit - According to Stallings [Sta95], this combination "makes little sense, since one incurs the cost of virtual circuit implementation, but gets none of the benefits."

2.4 Major Technical Issues

According to [Sun90], there are four major technical issues common to any packet-switching network. They are as follows: 1) stepwise versus endpoint services, 2) level of interconnection, 3) naming, addressing, and routing, and 4) congestion control.

2.4.1 Stepwise Versus Endpoint Services

This issue arises when there is a different set of protocols being used between networks. For instance, TCP/IP may be implemented on network A, while Novel Netware is used on network B. Stepwise service implies that the interconnecting gateway include an additional layer which converts the TCP/IP to Novell or vice versa. Endpoint services implies a common network (i.e., TCP/IP) and that the conversion takes place at the end-user systems. Sunshine [Sun90] brings out the argument that functionality mismatches are inevitable when translation is accomplished at a gateway (stepwise approach). He also argues that the endpoint approach guarantees a full service with common attributes at both ends by requiring implementation of a common protocol in the two endpoint services. He states that the endpoint approach "makes use of simpler services on the individual networks along the way, and hence allows use of simpler gateways...[and] there are fewer failure points." This will not be an issue in this research effort, since all LANs are assumed to implement TCP/IP when communicating across the WAN.

2.4.2 Level of Interconnect

Another main issue is to determine at what level in the protocol hierarchy are the networks interconnected. Alternatives exist all the way from the lowest level (i.e., repeaters) to the highest (application) level. As shown previously in Figure 2.2, networks used in this research effort will be interconnected at the network level (IP layer).

2.4.3 Naming, Addressing, and Routing

Stallings [Sta88] states a distinction which is generally made among names, addresses, and routes: "A name specifies what an object is, an address specifies where it is, and a route indicates how to get there."

2.4.3.1 Naming

Sunshine [Sun90] states that the name serves to identify the host logically, and which may be independent of which network the host is located on. An address identifies a point of attachment for purposes of delivering data to the host. The process of sending data to a destination generally involves first determining its address from its name using a directory service. TCP/IP allows the user to bypass the directory system and deliver data directly using the address.

2.4.3.2 Addressing

According to Sunshine [Sun90], a method must be found for uniquely identifying all network interfaces in an internet system. In TCP/IP, each address is 32 bits in length. The first set of bits identify the network and the remaining bits identify the host. A router (i.e., a packet-switching node with routing capabilities) uses the network identifier in determining which outgoing link to send the packet on.

2.4.3.3 Routing

Routing is normally one of the basic services provided by packet switched networks. Two types of routing services may be used in the network: virtual circuit and datagram. In virtual circuit routing (i.e., X.25) , a path is chosen at circuit setup time and used throughout the connection lifetime. On the other hand, when datagram routing is used (i.e., IP), a routing decision may be made for each packet at each intermediate switching node.

The operation of a typical packet switching node (PSN) as described by [SaA94] goes as follows. Packets enter and leave a node via a set of incoming and outgoing links as shown in Figure 2.5. As packets enter the node, they are processed by the node's central processing unit (CPU). If a packet is currently being processed when a new packet arrives, then the newly arrived packet will be stored in a queue. The processor will check the packet for errors, examine the packet's destination network identifier, possibly consult a routing table, and subsequently schedule the packet for transmission on the appropriate outgoing link by placing it in the link's corresponding queue.

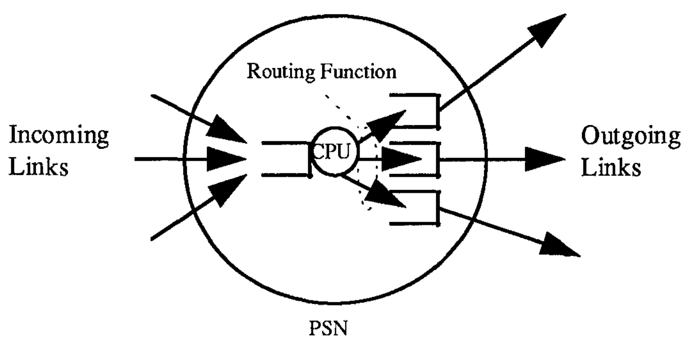

Figure 2.5 A typical Switching Node [SaA94]

2.4.3.3.1 Approaches to Routing

Two basic approaches are used to implement the routing function: table-driven, and table-free. Table-driven is the most popular approach [SaA94] and requires each node to store and maintain a routing table, which contains an association between a packet's identification and an outgoing link. The packet's ID can be the packet's destination address or an indication of a virtual circuit to which the packet belongs. Determination of the appropriate outgoing link involves the examination of a packet's header to extract the packet ID followed by a search of the routing table to determine the outgoing link. Routing tables need to be initialized and updated and may require extensive storage capacity in large networks.

In high-speed networks where the link capacity is high (i.e.,100 Mbps FDDI, 155 Mbps ATM) the processing time can become a bottleneck. In these cases, table-free routing might be used. This reduces the processing time considerably. For example, routing tables do not have to be consulted, nor do they have to be maintained. Examples of table-free routing, such as random routing, source routing, computed routing and flooding are referenced in [SaA94].

2.4.3.3.2 Routing Algorithms

Most routing algorithms attempt to route a packet over the best guess of the shortest path from source to the destination. This is done by assigning fixed or variable costs to links in the network and performing a shortest-path calculation. The results of this calculation are reflected in the routing tables at each node. Different approaches have been used in determining a link's cost. A straightforward approach is always to select a path that will carry a packet to its destination in the least amount of time. However, queuing and processing times can only be estimated, because they are strongly dependent on traffic conditions in the network and tend to vary over time. Another approach which is actually used quite often is the minimum hop path count. A variety of other approaches such as shortest backward path, and distributed procedures exist as well and are referenced in [SaA94].

2.4.3.3.3 Classification

Routing can be classified as static (deterministic) or dynamic. The distinction between the two types is usually defined in how often the routing tables get updated. According to [SaA94], if the shortest path calculation is performed often (e.g., 10 times an hour) and is based on some real-time measurement of network conditions, the routing procedure is said to be dynamic. Otherwise, it is static. Saadawi states that it must be emphasized that routing tables may change even when the static procedure is used. This, however, happens less frequently, (e.g., once a week) and typically is based on long-term averages of network conditions.

Routing can also be classified as either centralized or decentralized. In a centralized routing procedure, a central site is in charge of computing the shortest paths in the networks. If the procedure is dynamic, each node needs to report periodically the status of its links to the central site, which, in turn, needs to periodically provide new routing tables to all the nodes. In a decentralized (distributed) routing procedure, all network nodes are involved in shortest-path calculations. As a node typically possesses direct knowledge of only its local links, a distributed procedure needs to somehow pool the information available at each node to perform the distributed computation.

2.4.3.3.4 Routing in TCP/IP

The Internet model for routing separates a large network into many separate autonomous routing regions. These are called autonomous systems. For example, a campus network might control its own autonomous system. A wide-area network can also be an autonomous system. The shared WAN system to be investigated in this research is also an example of an autonomous system. A routing protocol used within an autonomous system is called an Interior Gateway Protocol (IGP). A popular IGP which is in use on the Internet today is the Routing Information Protocol (RIP) [Fei93]. The new Open Shortest Path First (OSPF) is rapidly gaining acceptance, and has been implemented in parts of the Internet [Fei93].

RIP computes routes using a simple distance vector routing algorithm [ZaA91, Fei93]. Every hop in the network is assigned a cost. The total metric for a path is the sum of the hop costs. RIP chooses the next hop so that datagrams will follow a least cost path. The costs for each path can take the form of least number of hops, amount of link capacity, or a combination thereof (other costs can be used as described above as well). RIP has to send routing updates, receive routing updates, and recompute routes. A RIP router sends information to its neighbor routers every thirty seconds. If no errors occur in the network (i.e., no link or node failures), then the tables will pretty much remain static, although RIP messages will still be sent out every 30 seconds. This is one of the reasons why RIP is sometimes considered inefficient.

There are other shortcomings associated with RIP. The maximum metric for any route is 15. For this reason, RIP usually is configured with a cost of one for each hop. After a disruption in the network, RIP is slow to find optimal routes. In fact, according to Feit [Fei93], datagrams may run around in a loop for a while. RIP cannot respond to changes in delay or load across links (i.e., it is not considered dynamic). Normally, it cannot split traffic to balance the load, although some router vendors have added facilities to do this in special cases.

In 1988, the Internet Engineering Task Force started work on a new protocol to replace RIP. The result is the Open Shortest Path First (OSPF) protocol. OSPF supports traffic splitting across multiple paths, routing based on type of service, and authentication for routing update messages. An OSPF router keeps a table of up-to-date information on the entire network topology [ZaA91]. Whenever a change occurs (such as a link failure) the information is propagated throughout the network.

Version two of OSPF was published in mid-1991. Because of the complexities of the OSPF protocol, minor revisions and bug fixes will continue for some period of time [Fei93].

2.4.3.3.5 Buffer Allocation Schemes

Another technical issue which must be considered in the operation of a packet-switching node is buffer allocation [IIM85]. The bottom line in buffer allocation is that no user (i.e., specific source/destination pair) should occupy all the buffers, nor should any user be starved of buffers because both cases degrade the performance. These undesirable situations can be avoided by limiting the maximum number of buffers a user can occupy at a time or by allowing each user to have a minimum number of buffers at his disposal (to avoid starvation). Wahida and Ahmed [WaA92] studied the buffer management problem and came up with interesting results. In their paper, three buffer management schemes were analyzed and compared. These schemes are complete partitioning, complete sharing, and square-root sharing.

In the complete partitioning scheme, a fixed number of buffers are permanently allocated to each queue. In other words, the node's total buffer space is partitioned into sub buffer spaces called queues. In this scheme, each queue's buffer space is finite and fixed in size. A packet may enter the system only if the portion of the buffer space associated with its queue is not filled. The opposite of the complete partitioning scheme is the complete sharing scheme. The complete sharing scheme permits an unrestricted sharing of total buffer space among all the queues at the node.

Both the complete partitioning (CP) and the complete sharing (CS) schemes may lead to undesirable behavior for the system. Under the CP policy, the buffers allocated to an almost empty queue are wasted (i.e., they are not used by their process and cannot be used by others). On the other hand, it has been found that CS succeeds in achieving a better performances (less losses) than CP under normal traffic conditions and for fairly balanced input systems. However, for highly unbalanced load, CS tends to heavily favor queues with higher input, which leads to the monopolization of most of the storage space by one of the queues. The above considerations suggest that, in order to reduce the impact of such circumstances, contention for space must be limited. This is incorporated by the third scheme: square-root sharing. This scheme imposes a limit on the maximum number of buffers to be allocated to any queue. The results show the square-root sharing scheme has the ability to avoid the deficiencies of both CP and CS schemes, and as such, outperforms them [WaA92].

2.4.4 Flow/Congestion Control

In a packet switched network, resources (links, nodes, buffer space, etc.) are shared among all the hosts (end-user systems). Because speed mismatches often times occur between LANs (i.e., 10 Mbps Ethernet) and slower speed wide-area networks (i.e., 64 kbps links), the WAN switching nodes can become potential bottlenecks that cause congestion in the network. For an illustration of how network performance may be affected, refer to Figure 2.6. Yang and Reddy [YaR95] state that if the load continues to increase up to the capacity of the network, the queues on switching nodes will build up, potentially resulting in packets being dropped, and throughput will eventually arrive at its maximum (see top diagram) and then decrease sharply to a low value (possibly zero). End-to-end delay (middle diagram), on the other hand, will begin to increase at a dramatic rate, and a point will eventually be reached where the connection will be broken. The power of the network is defined as the ratio of throughput to delay. The lower diagram shows that as the load is increased from zero, the power continuously increases until congestion begins to occur. This is considered the optimal load value.

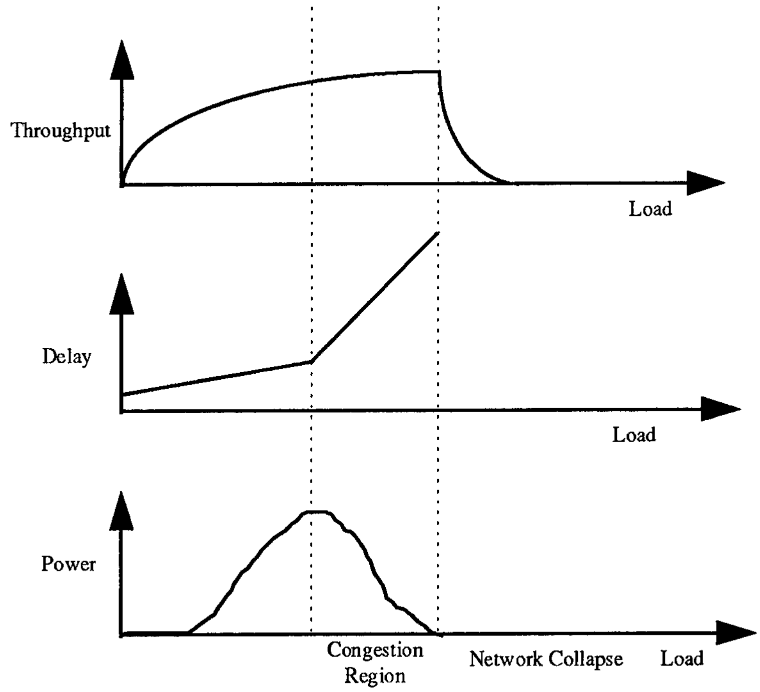

Figure 2.6 Network performance versus offered traffic load [YaR95]

There are numerous existing approaches for network congestion control which cover a broad range of techniques, including window (buffer) flow control, source quench, slow start, scheduled based control, binary feedback, rated based control, and others. Congestion control remains a high priority in network design due to ever-growing network bandwidth and intensive network applications. Many congestion control methods have been proposed, and more are forthcoming. Information concerning the operation and performance analysis of the above mentioned flow control strategies are referenced in [Kes91,Cha92,San93,YaR95]. A flow control scheme which warrants further investigation is TCP/IP's slow start implementation.

2.4.4.1 TCP/IP Flow/Congestion Control

Yang and Reddy [YaR95] point out that although a number of survey papers on a variety of congestion control algorithms have appeared in the literature, there still is not a systematic way for classification and comparison of so many diverse congestion control algorithms. The TCP/IP flow control algorithm employs a sliding window scheme for controlling the transmission rate from source end-user system to destination end-user system. Specifically, the sender and receiver will negotiate a window size upon initial connection. During the connection, the window size may vary (i.e., the receiving host may signal the sending host to slow down if it cannot handle the incoming data rate due to a shortage of available buffers). If a certain amount of time expires between the time a sender transmits a packet until an acknowledgment is received (i.e., dropped packet or excessive queuing delay encountered), a sender will assume congestion and go into a 'slow start' state. When in the 'slow start' state, the sender decreases its window size to one segment. For instance, the sender will send one packet and will not send another until an acknowledgment is received. The window size will begin to grow gradually as acknowledgments arrive [Fei93]. The sender will also retransmit any packets for which it did not receive an acknowledgment.

However, this scheme by itself is not effective in preventing congestion from occurring from within the network. When the network traffic becomes abnormally high, some "hot-spots" may occur on the network. Floyd states that most current routers in TCP/IP networks "have no provision for the detection of incipient congestion" [Flo94]. In other words, TCP/IP waits until a congestion problem occurs and then implements congestion control (via slow start).

There is also a congestion control algorithm called "source quench" employed by TCP/IP that explicitly notifies the source of the traffic that congestion is occurring, but according to Floyd [Flo94], "it is rarely used on the Internet." An intermediate node, such as a router, or a host would send a source quench message when it receives datagrams at a rate that is too fast to be processed. Floyd states that Request For Comments (RFC) 1009 (RFCs are used for establishing TCP/IP networking standards) requires routers to generate source quenches when they run out of buffers, but the current draft on Requirements for IP routers specifies that a router should not originate source quench messages, and a router that does originate source quench messages must be able to limit the rate at which they are generated. Floyd [Flo94] states that according to the draft, "Source Quenches are criticized as consuming bandwidth, and as being both ineffective and unfair." According to RFC 1009 guidelines, hosts should respond to a source quench by "triggering a slow start, as if a retransmission had occurred."

According to Feit [Fei93], "new connections that immediately start to transfer bulk data across a network can cause stress." The 'slow start' incorporated by TCP/IP prevents this by initializing an end-user's congestion control window to one segment, and allow the window size to increase as ACKS (acknowledgments) arrive, just as is done in congestion recovery. The current TCP standard requires conformant implementations to adhere to 'slow start' when initiating a connection and for controlling congestion.

The current TCP standard also states that implementations use the algorithms of Karn and Jacobson to estimate the timeout period. In order to understand these algorithms, a brief description on TCP's retransmission policy is needed. After sending a packet, TCP sets a timer and listens for an acknowledgment (ACK). If the ACK does not arrive within the timeout period, TCP retransmits the segment. According to Feit [Fei93] , if the retransmission timeout is too short, the sender will clutter the network with unnecessary packets, and burden the receiver with extraneous duplicates. On the other hand, if the time-out is too long, a brisk recovery will be prevented when a packet really has been lost, and will decrease throughput. The algorithms of Karn and Jacobson "enable TCP to adapt to changing conditions, and are now mandated for TCP implementation" [Fei93]. These algorithms are defined next beginning with Jacobson's algorithm.

Jacobson's algorithm is used in initializing the TIMEOUT value. First, round trip time (RTT) is calculated as the time that elapses between the transmission of data and the arrival of matching acknowledgments. Jacobson actually takes the running average of RTTs called smoothed RTT (SRTT) and weights the last RTT to have a greater effect on the smoothed average. The variable, α, is used as the weigting factor and its value lies between 0 and 1. A typical value of α would be 1/8 [Fei93]. The equation used in Jacobson's algorithm is:

$$\text{New SRTT} = (1 - \alpha) \times \text{Old SRTT} + \alpha \times \text{Latest RTT}. \tag{1}$$

He also defines another variable, DEV (short for deviation) as

$$\text{DEV} = |\text{Latest RTT - Old SRTT}|. \quad (2)$$

The TIMEOUT value is then calculated as

$$\text{TIMEOUT} = \text{SRTT} + 2 \times \text{SDEV}. \quad (3)$$

where,

$$\text{New SDEV} = 3/4 \times (\text{Old SDEV}) + 1/4 \times \text{DEV}. \quad (4)$$

As can be seen, the TIMEOUT is a dynamic value and is constantly changing during packet transmission. This will go on until an actual time-out occurs. When a retransmission occurs, the system immediately switches over to Karn's Algorithm.

Karn's algorithm is based on the assumption that the expiration of retransmission timer (a timeout occurs) probably indicates a condition of congestion in the network. As such, Karn's algorithm calls for an increase to the retransmission timer. According to Feit [Fei93], this is usually done via a multiplicative factor as follows:

$$\text{New TIMEOUT} = \text{Factor} \times \text{Old TIMEOUT} \quad \text{(Usually Factor is set to 2)} \quad (5)$$

If the new time-out expires, it gets increased again. Time-outs will increase up to a prespecified maximum, and then stay there. There will be a limit on the total number of unacknowledged retransmission attempts. According to Feit [Fei93], when this limit is reached, the connection will be aborted.

Growth in round trip times (RTT) or the arrival of a Source Quench message are indicators of congestion in the network. TCP employs an additional measure whenever a retransmission timer expires or upon receipt of a Source Quench message. Feit [Fei93] says that the mechanism for doing this is to "define a congestion window and to restrict a sender to transmitting data that lies within the congestion window." During normal transmission, the congestion window is set to the same size as the send window. When a retransmission occurs, or a Source Quench is received, then the congestion window is resized to:

$$\text{maximum}\{[1/2 \times (\text{current congestion window size})], [\text{single segment size}]\} \quad (6)$$

For efficiency, it is not required that the receiver acknowledge each segment received. A cumulative acknowledgment is permitted [Sta88].

2.5 Modeling and Analysis

2.5.1 Network Model

The last issue to be discussed in this section is to summarize some of the methods being used to model and analyze packet switched wide-area networks. Packet switching networks are normally modeled using queuing analysis techniques. In many instances, a wide-area network is modeled as a collection of several queues and servers. The customers are called data packets. When the customer finishes one service, it may request service from another server, and so on, until it is ready to leave the system. Such a model is also called a network of queues [AgJ90]. Deterministic methods have been used to analyze the performance of such networks. In doing so, many simplifying assumptions are usually made. For instance, Agrawla and Jain [AgJ90] provide a deterministic analysis of a communications network. First, they describe the operation:

> *Over a virtual circuit, the source sends a sequence of packets to a destination. All packets follow the same path. Each packet is processed as it moves from node to node in a store and forward manner. It sustains possible queuing delays at intermediate processing nodes and transmission delays in moving from one node to the next.*

They assume that all quantities (i.e. processing time, packet length) are deterministic and known. They also assume that all packets between the source-destination host pair move through the network along the same path, and that no cross traffic takes place (refer to Figure 2.7). Packets relating to other source-destination pairs, but which are processed by one or more intermediate network processors along the path, are considered as cross traffic (refer to Figure 2.8).

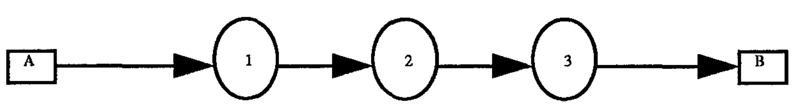

Figure 2.7 Flow of traffic between a pair of hosts with no cross traffic [Jai94]

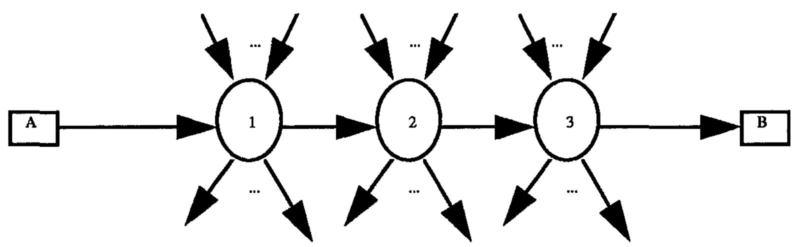

Figure 2.8 Flow of traffic between a pair of hosts with cross traffic [Jai94]

Many other studies have been conducted using analytical methods, but they have usually been used for a specific function, such as comparing flow control strategies [San93,Kes91], comparing buffer allocation schemes [YaO94, WaA92], or comparing specific routing algorithms [Los92]. In many cases, closed form solutions have not been possible, and approximation techniques have been implemented. Matta [MaS94] and others have analyzed models by numerically solving differential equations. However, these calculations become rapidly unmanageable for realistic models [Bol93,San93,LaM94]. Kiemele states that "the numerous internodal conditions and variables preclude an exact analytic solution" [Kie90].

Discrete event simulation has also been extensively used [AgJ90]. In such models, the exact interaction among the components of the system are duplicated in the same time sequence order, while carrying out actions corresponding to every event. However, Agrawala does point out that some discrete event simulation models tend to become large programs which are difficult to write and debug. Agrawala further states that they are even more difficult to verify and validate. Despite these pitfalls, simulation will be the method used in this research effort. Of the two simulation packages (SLAM and DESIGNER) available at the Air Force Institute of Technology (AFIT), DESIGNER has been chosen, because it is more geared towards simulation of communication networks.

2.5.2 Simulated Traffic

An important concern in modeling a wide-area network and analyzing its performance is the simulation of network traffic. The amount and pattern of simulated traffic should represent the network's true traffic as close as possible. Traditionally, packet arrivals have often been assumed to be Poisson processes because such processes have attractive theoretical properties [PaF94]. However, Paxson and Floyd's research [PaF94] and numerous other studies [CaD91,FrM94,BrC93,Zha90] have shown that wide-area traffic is much burstier than Poisson models predict. Jain [Jai90] found that a 'bursty' Poisson arrival pattern matched the actual measured traffic generated at a Warehouse Inventory Control customer site. Upon observation of the true measured traffic, he found that when a station transmits, it generally transmits not one frame, but a burst of frames. Other studies in traffic modeling have recommended a similar traffic model, called a 'packet train,' over the standard Poisson process [Zha90,JaR86,CaD91].

Caceres and Danzig [CaD91] describe the packet-train as a "handshake followed by a big burst." Zhang [Zha90] uses the packet train model for generating data in order to analyze queuing delays and packet losses across a wide-area network. According to [CaD91], the 'packet train' model has proven useful in the design of packet routers. Because the packet train model will be used in simulating bulk traffic, it warrants further discussion.

The generation of a packet train model is described by three parameters: train length, inter-train gap, and inter-packet gap. An example of the packet train used in Zhang's simulation is shown in Figure 2.9 [Zha90]. The train length (number of packets in the train) is modeled as a geometrically distributed random variable. The inter-train gap is modeled as an exponentially distributed random variable. The interpacket gap can be set as a constant or as an exponentially distributed random variable with a mean interarrival time significantly less than the inter-train gap [JaR86]. Zhang uses an interpacket gap equal to 1/(2 x average rate). However, Zhang does not specify what he means by average rate, although it appears as though he is using the mean inter-train gap as the average rate. Further, Zhang does not specify whether he uses a constant or an exponential distribution for interpacket arrival times. All data packets are assumed a constant size of 250 bytes.

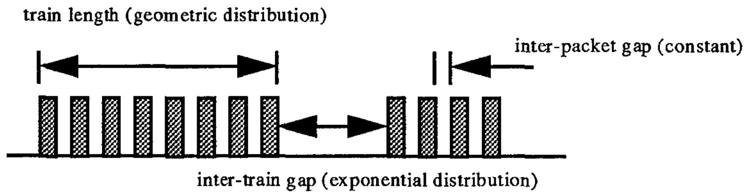

Figure 2.9 Packet Train example

2.5.2.1 Internet Traffic Analysis

Caceres and Danzig's [CaD91] paper presents an analysis of wide-area TCP/IP traffic patterns collected on two campus networks and one industrial research site (these sites are interconnected via Internet). They point out that numerous studies have either used a continuous bulk transfer or an arbitrary mix of bulk and interactive traffic as a traffic model. In their results, they characterize the traffic into five areas: 1) traffic breakdown - is it interactive or bulk traffic; 2) amount of bulk - how bulky is the data; 3) characteristics of the interactive applications; 4) traffic flow - is the traffic unidirectional or bi-directional; 5) Network-pair locality preference - does there exist a network pair or network pairs where a majority of the conversations take place.

2.5.2.1.1 Traffic breakdown results

The results of their traffic measurements show that TCP traffic consists of bulk and interactive traffic as commonly assumed. Approximately 25-45 % of the packets were interactive.

2.5.2.1.2 Bulk data transfer

Caceres and Danzig found that 75-90% of the bulk transfer conversations transfer less than 10K bytes. They believe this occurs because most files are small. They additionally state that in most sessions, data transfer will complete before any feedback to a flow control type mechanism is received.

2.5.2.1.3 Interactive Applications

Caceres and Danzig's results show that about 90% of TELNET and RLOGIN conversations send less than 10K bytes over a duration of 1.5 to 50 minutes. Furthermore, about 90% of TELNET and RLOGIN packets carry less than 10 bytes of user data, which is much smaller than the maximum transmission unit (MTU). Because of this, they say that interactive applications are more or less unaffected by flow control and MTU size. Their analysis further reveals that interactive conversations can be modeled by a constant plus exponential random time.

2.5.2.1.4 Traffic Flow

The results show that a large percentage of traffic, both interactive and bulk, is bi-directional. In contrast, they point out that most previous simulations show only a one way data flow.

2.5.2.1.5 WAN Locality Preference

In Local Area Networks (LANs), certain hosts communicate more with one another than with other hosts. Caceres and Danzig investigated to see whether or not the same locality of preference exists between host pairs or network pairs in wide-area internetworks. Their results show that half of the TELNET conversations from one of the two campus networks are directed to just 10 sites. While the other half referenced over a 100 sites. The results further show little evidence in host to host locality preference, except in a few NNTP (Net News Transfer Protocol) exchanges.

2.5.2.1.6 Conclusion

Caceres and Danzig have described a way to generate wide-area traffic from a stub network. They break down the traffic to be either bulk or interactive. They state that interarrival times for bulk data exhibit the packet train phenomenon, and that interarrival times for interactive applications should be modeled by a constant plus exponential random time. During bulk transfer, packets are at their MTU size (512 bytes). They further found that traffic is bi-directional, and that some network locality preference exists. Since 6 of the 35 applications they identify in their experiment account for 96% of the bytes transmitted, they modeled only those applications. They are FTP, SMTP, NNTP, VMNET, TELNET, and RLOGIN.

2.5.3 Flow Control

In this research effort, focus is on the wide-area network performance issues (i.e., WAN end-to-end delay associated with shared links vs. nonshared links). As such, implementation details within the LAN environment are avoided as much as possible. One of these details is flow control. In order to implement flow control mechanisms, many LAN issues have to be investigated. Some of the LAN details are as follows: 1) what type of LAN the host is connected to (i.e., LAN access mechanisms such as collision detection, or token circulation time must be taken into account); 2) window sizes associated with different types of hosts (i.e., PCs, Sun Workstations, Multiuser Systems, etc.); 3) client-server relationships (i.e., mail servers, file servers, etc.); and 4) other possible issues such as the need to generate local traffic (host to host within the LAN) as well as wide-area traffic to get a true simulation of the real system traffic going across the WAN.

As mentioned in Section 2.5.2.1, Caceres and Danzig investigated how TCP/IP traffic might be simulated from a 'stub' network. In their study [CaD91], live TCP/IP data was captured and measured as it traveled from the LAN environment into the WAN environment. They focused their attention mainly on TCP layer generated traffic, and in doing so they filtered out retransmissions. They did however, estimate a total of only .3% to a little below 3% of all packets transmitted were retransmissions. In their conclusions, they stated that the traffic patterns (see Section 2.5.2.1 Internet Traffic Analysis) could be represented as that generated by a local area network. Before concluding, however, Caceres and Danzig point out that if robust testing is to be done on specific algorithms, they do not suggest that their traffic model be used in place of worst case scenarios. They merely describe it as a 'realistic internetwork traffic model' which can possibly be simulated without flow control mechanisms in place. Based on the above observations, flow control will not be explicitly modeled into the system used in this research.

To further justify obviating the need for flow control, the operating region where performance measurements will be taken will be in the non congested region (i.e., to the left side of the congestion region shown in Figure 2.6). Numerous studies of wide-area networks which require operation in the congested region are usually focused on analyzing flow control algorithms (see section 2.4.4). The point where the congestion region begins will be determined by a steady state analysis of the design model.

In regards to acknowledgments, as was described earlier, TCP/IP is a connection-oriented protocol. However, in order to simplify the model design, acknowledgments will be assumed to occur (i.e. piggybacked on packets returning in the opposite direction), and will not be specifically modeled into the system. Other wide-area network performance studies with connection oriented traffic (i.e., voice traffic) have been performed similarly (i.e., without flow control mechanisms and acknowledgments) [YaT93].

2.5.4 Performance Analysis

Performance evaluation criteria for a communications network may vary from one individual to another, depending upon how someone is associated with the network under consideration [IlM85]. Ilyas organizes the performance evaluation criteria into three categories: 1) user-oriented criteria, 2) manager-oriented criteria, and 3) designer-oriented criteria. In describing user oriented criteria, the authors state that the user is mostly concerned with how quickly the information is transferred from one point to another (that is the delay per message). In the manager oriented criteria, they say that the manager is mostly concerned with the best utilization of resources (i.e., high bandwidth utilization). However, the manager (at the same time) likes to keep network users happy by keeping the network delay to an acceptable minimum value. A problem lies in the fact that as the throughput increases, so does delay. For this reason, the authors present a more compact performance criterion called 'power.' Power is defined as the ratio of a networks throughput to its delay. A designer has a somewhat different perspective of the network. The designer is mainly concerned with tuning the network parameters so as to achieve desired objectives. Buffer efficiency, protocol efficiency, flow control effectiveness, and adaptability are some of the main concerns of the designer. In all three categories, a cost is also of concern.

Practically every paper encountered in the literature review echoed the same thoughts regarding the measurements of effectiveness of computer-communications networks: responsiveness and productivity [IlM85,Jai90,San93,LaM94,KaS95]. Productivity is normally measured in terms of throughput, which is a measure of how many bits are sent from source to destination over a specific period of time. Throughput is proportional to the load and is restricted by the amount of bandwidth available. Responsiveness is measured in terms delay, normally average end-to-end delay of the packets. Other concerns addressed in some of the literature also include number of lost packets, delays from point A to B, costs, node buffer utilization, and various other queue related measurements.

2.6 Summary

In the first section shared bandwidth versus nonshared bandwidth was discussed The way in which data traffic is classified as either shared or nonshared had been determined as follows. In the nonshared mode of operation, it has been assumed that dedicated links will be set up to allow only a single source-destination pair per link. In the shared mode of operation, bandwidth will be shared by all sources and destinations, and routing will take place at the intermediate packet-switching nodes. In Section 2.2, it was found that a wide-area network can be represented by a series of nodes and interconnecting links. Section 2.3 dealt with major technical issues such as flow control and routing. Specifically, the 'slow start' windows based flow control mechanism used in TCP/IP was elaborated upon. In the last section, under modeling, simulation versus analytical methods were explained. Different traffic patterns were examined. Specifically, the 'pulse train' for bulk traffic, and the exponential plus constant for interactive traffic were presented in detail. The most common measurements of effectiveness were found to be in terms of average end-to-end delay and bandwidth utilization.

3 *Methodology*

3.1 Introduction

The purpose of this chapter is to present a method (via simulation) which can be used to compare the performance between a shared bandwidth system and a nonshared system. First an explanation of the performance metrics is provided in Section 3.2. As determined from the literature review, responsiveness and productivity have been found to be good measures of effectiveness. They have, therefore, been chosen as the main performance metrics to be used in this research effort. Section 3.3 discusses traffic load patterns. Although the focus is on TCP/IP traffic, the traffic load patterns discussed are similar in other types of networks as well. The remainder of the chapter is divided into two sections: 1) the development of a two node experiment in Section 3.4, and 2) the development of a five node experiment in Section 3.5. In both the two node and five node experiments, a shared system will be compared to a nonshared system. Operating assumptions, model development, steady state analysis, and verification and validation of the each of the models will be included. And finally, verification and validation of the two node and five node systems are presented.

3.2 Performance Metrics

Previous research suggests that quality of service of a packet switched network can be determined by its responsiveness and productivity [Jai90,San93,KaS95]. Productivity is normally measured in terms of throughput, and responsiveness is normally measured in terms of average end-to-end delay of the packets. Each of these performance metrics are now discussed in turn.

3.2.1 Throughput

Throughput is a measure of how many bits are sent from source to destination over a specific period of time. The throughput is proportional to the input traffic load (number of packets per unit time transmitted from source to destination) and is restricted by the amount of bandwidth available. In this experiment, throughput is measured in terms of percent bandwidth utilization. Percent bandwidth utilization is a measure of actual number of bits per second flowing across a link divided by the maximal possible number of bits per second (link capacity). Percent bandwidth utilization is calculated as follows:

$$\frac{TotalNumberOfBits}{LinkCapacity * WindowPeriod} \tag{1}$$

where 'TotalNumberOfBits' equals the total number of bits going across the link, and 'WindowPeriod' equals the simulation run time minus the warm up period (the warm up period prevents initial bias). 'LinkCapacity' equals the maximal possible flow that the link can withstand and is expressed in terms of bits per second (bps).

3.2.2 End-to-End Delay

End-to-end delay is an accumulation of delays a packet encounters as it travels from the source to the destination. It can be calculated as follows:

$$TransmissionDelay + PropagationDelay + ProcessingDelay + QueueDelay = TotalDelay \tag{2}$$

$$TransmissionDelay = \frac{L}{C} \tag{3}$$

where 'L' equals the length of the packet in bits and 'C' is the capacity of the channel in bits per second. The amount of transmission delay will vary because the packet size varies.

$$PropagationDelay = \frac{d}{c} \tag{4}$$

where 'd' equals the distance between the nodes in meters and 'c' is the speed of light (3 X 10^8 meters per second). This delay will remain fixed based on the distance between the nodes.

$$ProcessingDelay \approx N(\mu, \sigma^2) \tag{5}$$

where $\mu = 100 \times 10^{-6}$s and $\sigma = 10 \times 10^{-6}$s. This is an approximate time of how long it takes a node to process an incoming packet and send it to an appropriate outgoing channel. Clark and Van Jacobson [ClJ89] measured the time to process a TCP packet at 440μs, but this was done in 1989 on a Sun 3/60, based on a 20 MHz processor. With today's faster processors (100 Mhz range), the processing delays are sometimes considered negligible [Cha92,YaK93]. According to Spohn [Spo93], packet switches can process packets anywhere from 300 packets per second (PPS) at the low end to 10,000 PPS at the high end.

Queue Delay is the most variable delay parameter. It covers the amount of time that a packet spends in the queues as it travels from the source to destination.

3.3 Traffic Load

According to Jain [Jai90], the response times depend not only on the load of the input traffic, but also on the arrival pattern of network traffic. Unfortunately, as was pointed out in Chapter 2, workload is probably the most controversial part of every performance evaluation project [Jai90,CaD91,Zha90,PaF94]. The traffic load pattern to be used in this simulation is similar to the patterns used in previous research efforts in simulating traffic for packet switching networks [Jai90,Zha90,CaD91,YaK93]. The pattern consists of a pulse train for bulk traffic and a constant plus exponential for interactive traffic.

3.4 Two Node Experiment

A two node packet switching network is constructed and analyzed in this section. The packet switching nodes are modeled as intermediate switching nodes. On one side of the node are two Local Area Networks (LANs), and on the other side is the node-to-node interconnecting link. Two configurations will be examined: 1) shared bandwidth, and 2) nonshared bandwidth. In the shared mode, one T1 link will be used, and in the nonshared mode two T1 links will be used. The first part of this section contains the objective of the experiment. The operating assumptions are explained in part two. In part three, the principles of operation for the network models are explained along with illustrations of the network model. The statistical aspects of simulation are covered last, including steady state analysis and results. Verification and validation of the models are presented in Sections 3.6 and 3.7.

3.4.1 Objective

The purpose of this experiment is to present a method (via simulation) which can be used to compare the performance between a shared bandwidth system and a nonshared system. Both systems will be examined in terms of percent bandwidth utilization (productivity) and average end-to-end delay (responsiveness). In this scenario, the nonshared system has been configured with two T1 links, whereas the shared system contains only one T1 link. These systems were configured this way so that a determination can be made whether or not a single shared T1 link could be used in place of two nonshared T1 links. The performance parameters (% bandwidth utilization and average end-to-end delay) will be observed to see how they are affected by varying the input traffic load and link capacity.

3.4.2 Operating Assumptions

(a) Each Local Area Network (LAN) generates the same load (see Section 3.4.4.2).

(b) Approximately half of the traffic is interactive, while the other half is bulk traffic.

(c) Traffic is bi-directional.

(d) Traffic generated at each site is independent from the other. However, it is assumed that a portion of the packets being generated (although independently) are, in fact, responses to packets being received and include acknowledgments.

(e) Number of 'bulk' packets generated during each burst follows a geometric distribution with mean equal to 10. The interarrival time between packets during a burst is 10 μs [Zha90].

(f) Nodal Buffers are assumed to have infinite capacity.

(g) The Local Area Networks (LANs) connected have a high bandwidth capacity (i.e. 100 Mbps, FDDI) and transmission and access delay to the local network are negligible.

(h) LAN to LAN intercommunication is as follows: LAN A1 intercommunicates with LAN B1, and LAN A2 intercommunicates with LAN B2. No other cross communication takes place.

(i) Node processing time is assumed $N(\mu,\sigma^2)$, with $\mu = 100 \times 10^{-6}$ s and $\sigma = 10 \times 10^{-6}$ s.

(j) Errors due to channel noise are negligible.

(k) All links and components are 100 % reliable.

(l) There is no priority traffic.

(m) The sites are separated by 500 miles and connected by full-duplex links.

(n) The nodes use a 'large buffer' memory management scheme (explained in Section 3.5.3).

3.4.3 Network Model

A brief explanation on how a packet-switching communication network is modeled follows. Generated packets are represented as customers requesting service; the service times correspond to the processing and transmission delays; and the amount of time packets spend waiting for each service corresponds to queuing delays. Both the shared and nonshared bandwidth have been modeled. The packet structure applies to both systems and is discussed first.

3.4.3.1 Packet Structure

The fields in the packet structure are shown in Figure 3.1. The size field is equal to the Maximum Transmission Unit (MTU) of 4096 bits for bulk traffic, and is exponentially distributed with mean equal to 450 bits for interactive traffic. The minimum size of a packet is 360 bits [Fei93]. The origin is set as follows: '0' if originating from LAN A1, '1' if originating from LAN A2, '2' if originating from LAN B1, and '3' if originating from LAN B2. Time Created is a real number which represents the time the packet is created in the traffic source. Time Finished represents the time the packet reaches its destination and exits the system. The type field (not used in this experiment) is used to determine whether the packet contains data or an acknowledgment.

Name: LAN packet [thebones]
Date: Monday, 11/20/95 03:40:02 pm EST

Name	Type	Subrange	Default Value
origin	INTEGER	[0, +Infinity)	0
destination	INTEGER	[0, +Infinity)	0
time created	REAL	[0, +Infinity)	0.0
type	Packet Type	...	Data
time finished	REAL	[0, +Infinity)	0.0
size	INTEGER	[0, +Infinity)	1024

Figure 3.1 Packet Structure

3.4.3.2 Shared Bandwidth Configuration

The first system to be discussed uses shared bandwidth (Figure 3.2). At the system level, the network includes two LANs at each site connected to a packet switch node. The nodes are interconnected by a single full-duplex T1 link [Kar94]. Hence, during site-to-site LAN communication, a common channel is shared by all four LANs. A 'data collection' block (shown in Appendix A, Figure A 5) is used to record end-to-end delay of all packets transmitted. The LAN traffic generators are constructed identically and an illustration of one of them is shown in Appendix A, Figure A 1. Figure A 2 shows the submodule within the traffic generator which implements an exponential plus constant distribution.

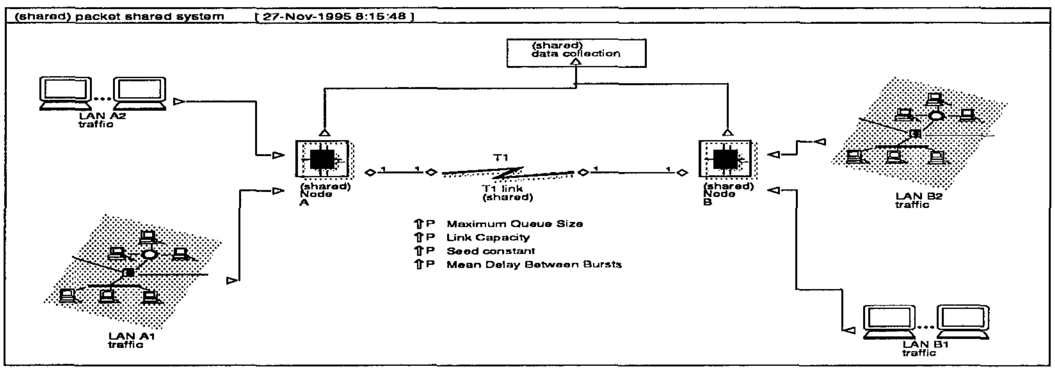

Figure 3.2 Two Node Shared Network System

The modules representing the Site A and Site B nodes (Figure 3.3) are constructed identically. Within the node is a network layer block, and a separate data link layer block for each link connected (in this case two of the links are connected to LANs, and the other is connected to the WAN). When a packet arrives, it first enters the data link layer. The data link layer (Figure 3.5) performs no functions on an incoming packet. It merely passes the packet onto the network layer. In the network layer (shown in Figure 3.4), a packet undergoes a processing delay and is subsequently routed to the appropriate link. The node's central processing unit (CPU) can process only one packet at a time. If it is busy, an arriving packet enters a queue on a First In First Out (FIFO) basis. After processing, the packet is routed via the 4-way switch to the appropriate data link layer block. In the data link layer (Figure 3.5), an outgoing packet will undergo transmission processing. If there is another packet currently being serviced, the outgoing packet will be stored in the queue on a First Come First Serve (FCFS) basis. The transmission processing time is proportional to the size of the packet and inversely proportional to the capacity of the link. The capacity of the link is configured as a variable input parameter named 'Link Capacity.' For illustrations of the Processing Delay and Transmission Delay Modules refer to Appendix A, Figures A 3 and A 4 respectively.

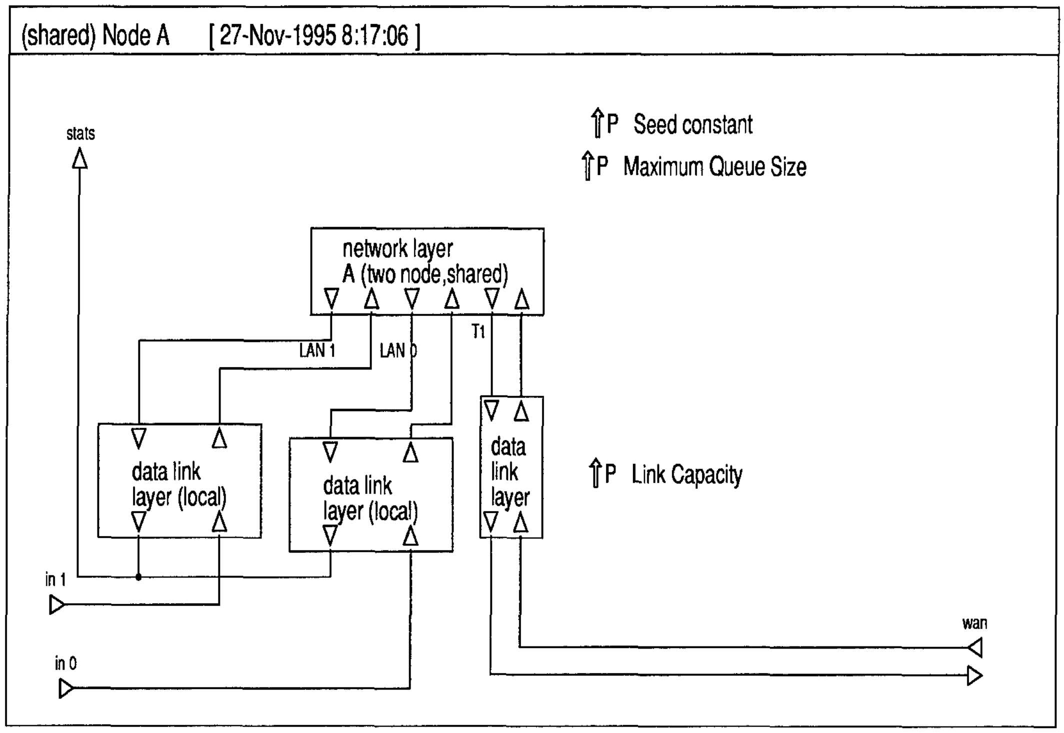

Figure 3.3 Node Block

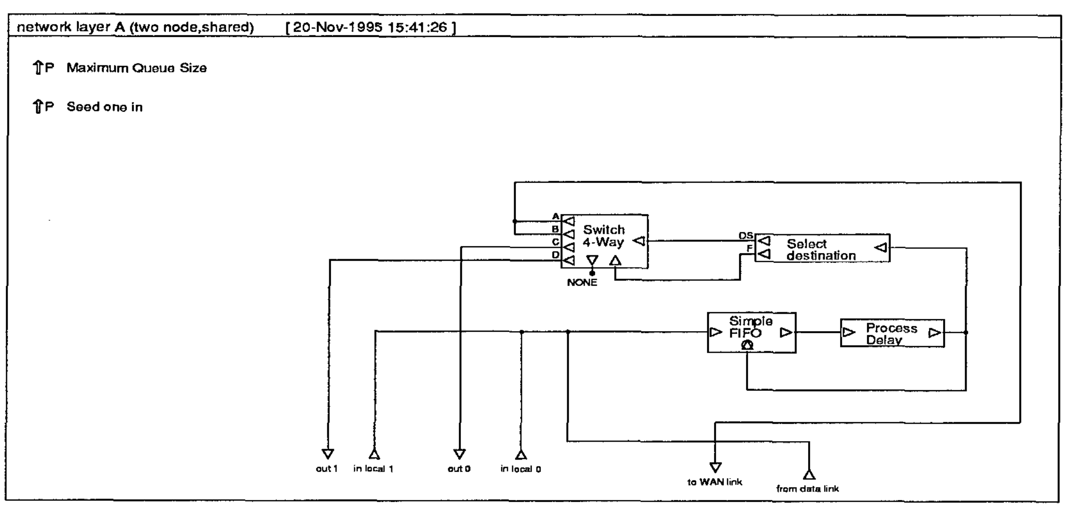

Figure 3.4 Network Layer

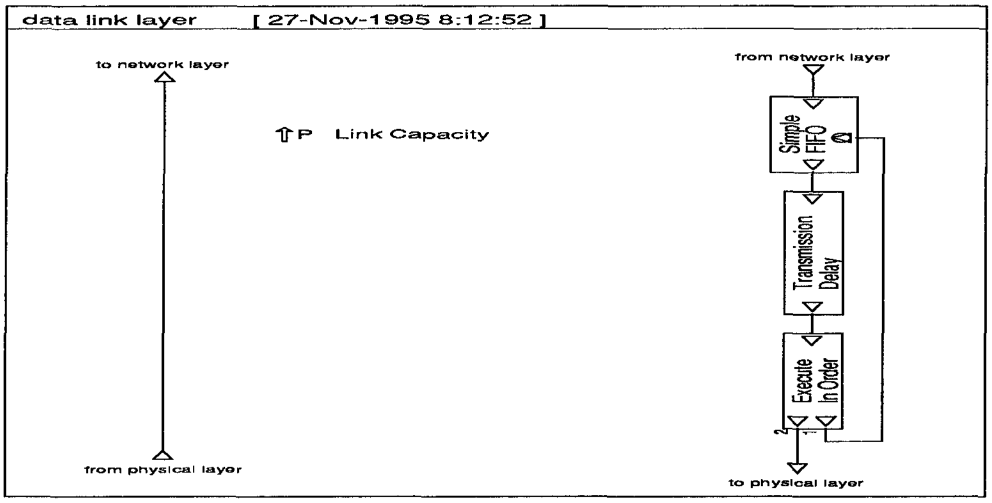

Figure 3.5 Data Link Layer

Within the T1 link module (Figure 3.6), the 'Fixed Abs Delay' blocks implement the propagation delay. Each packet encounters a fixed delay as it passes through this block. The delay is based on the distance between the links. A sink is included for the purposes of collecting bandwidth utilization statistics.

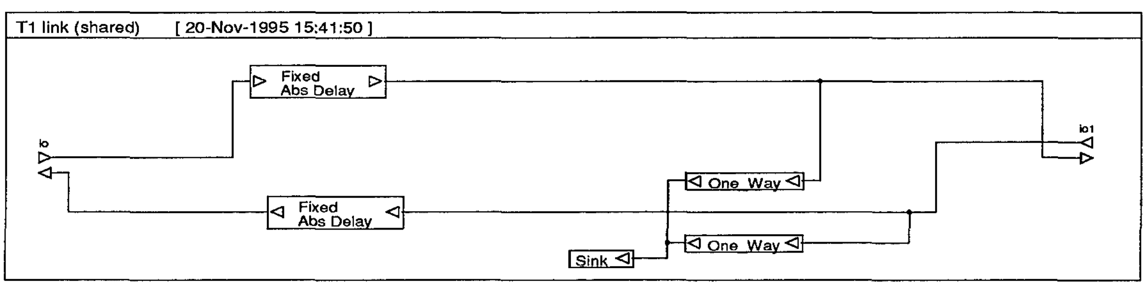

Figure 3.6 T1 Link

3.4.3.3 Non Shared Bandwidth Configuration

The nonshared configuration is very similar in construction to the shared configuration. The main differences occur at the system level (Figure 3.7). Whereas the shared system has only one interconnecting link, the nonshared system has two links interconnecting the nodes. Further, within the node (Figure 3.8), there is a separate node processor (network layer) for each LAN connected. In fact, it can be said that there are actually two independently operating nodes within the node. Notice, however, that there is no link between the site's LANs (i.e., LAN A1 is not linked to LAN A2). As such, there can be no communication between the LANs local to the site. For an illustration of the T1 lines and the network layer refer to Appendix A, Figures A 6 and A 7.

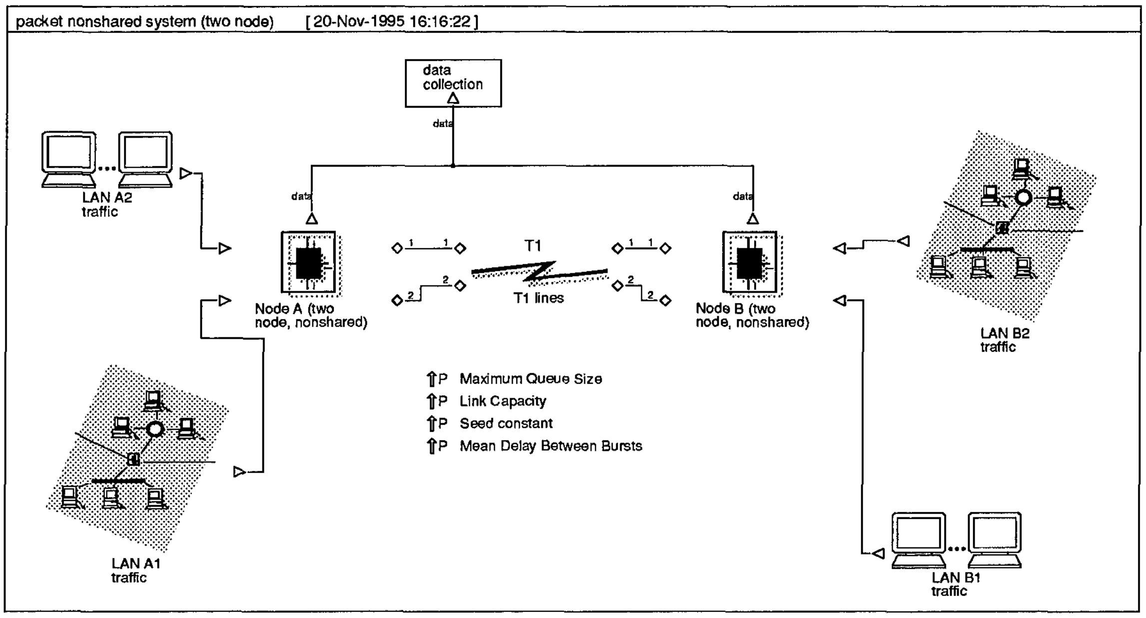

Figure 3.7 Two Node Non Shared System

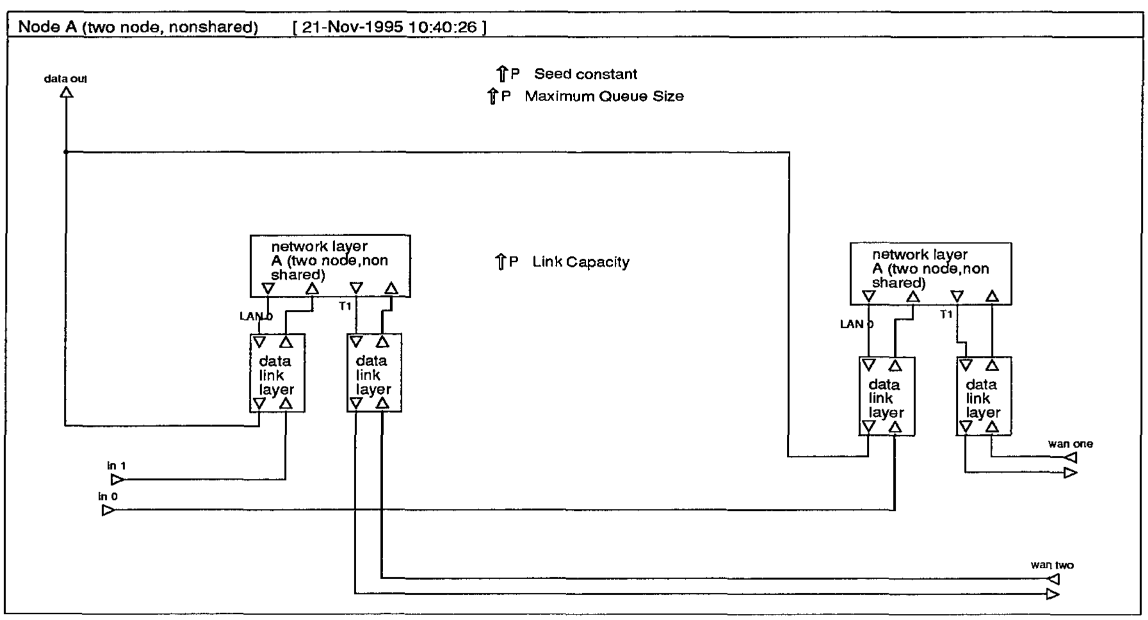

Figure 3.8 Node Block

3.4.4 Parameters

Several different simulations are performed to produce the required performance metrics. There are fixed input parameters, variable input parameters, and output metrics. Each of these parameters are discussed below.

3.4.4.1 Fixed Input Parameters

1. Number of nodes = 2
2. Topology: One link interconnecting two nodes
3. Distance between nodes = 500 miles
4. Node processing delay ~ N(100μs, 10μs^2)

3.4.4.2 Variable Input Parameters

1. Bandwidth = (shared, nonshared)
2. Link Capacity = (768, 1152,1544, 2305) * 1000 bits per second (bps)
3. Traffic Load = (4,6,8) bursts per second (an example of a burst may be a file transfer)

In order to clarify the parameter, 'Traffic Load,' the following example is provided. Assume "'Traffic Load' = 6." For bulk traffic, the number of bursts follows a Poisson process with a mean equal to **six** bursts (packet trains) per second. The number of packets generated by each burst (packet train) follows a geometric distribution with mean equal to 10 (fixed parameter). Interpacket time during the burst is equal to 0.1 ms (also a fixed parameter). Each packet is MTU size (4096 bits). For interactive traffic, the interarrival time between packets follows an exponential plus constant distribution with a mean equal to 1/('Load'*Constant), where 'Constant' is set to a value of 100 (so that the amount of interactive traffic is approximately equal to the amount of bulk traffic). In following the example, mean interarrival time for interactive packets will equal 1/600 seconds. The size of interactive packets follows an exponential distribution with a mean equal to 450 bits (fixed parameter). To summarize, a rough approximation of the loads in bps is illustrated below in Table 3.1. Note: pps = packets per second, and bps = bits per second, I/A = interactive traffic.

Table 3.1 Load expressed in bits per second (bps) (two node network).

Load	bulk pps (10 * Load)	bulk bps (bulk PPS * size)	I/A pps (100 * Load)	I/A bps (I/A PPS * size)	total bps (bulk bps + I/A bps)
4	40	163,480	400	180,000	343,480
6	60	245,760	600	270,000	515,760
8	80	327,680	800	360,000	687,680
9	90	368,640	900	405,000	773,640
10	100	409,600	1000	450,000	859,600

3.4.4.3 Output Metrics

1. Mean number of packets accumulated in the node's processing and transmission queues
2. Average end-to-end delay
3. Percent bandwidth utilization

3.4.5 Statistical Precision

In performing the simulation, certain choices have to be made, such as the length of the runs, the number of independent runs, and the length of the warm up period [LaM94]. In this section, the first task pertains to determining what is a fair representation of a heavy, medium, and light traffic load. In order to produce statistically precise performance estimates, it is necessary to vary the load in a range which will not overload the system. An initial step in determining the heavy load is accomplished by varying the load, while keeping the other parameters fixed (Link Capacity = 1.544 Mbps, Shared Bandwidth) as shown in Figure 3.9.

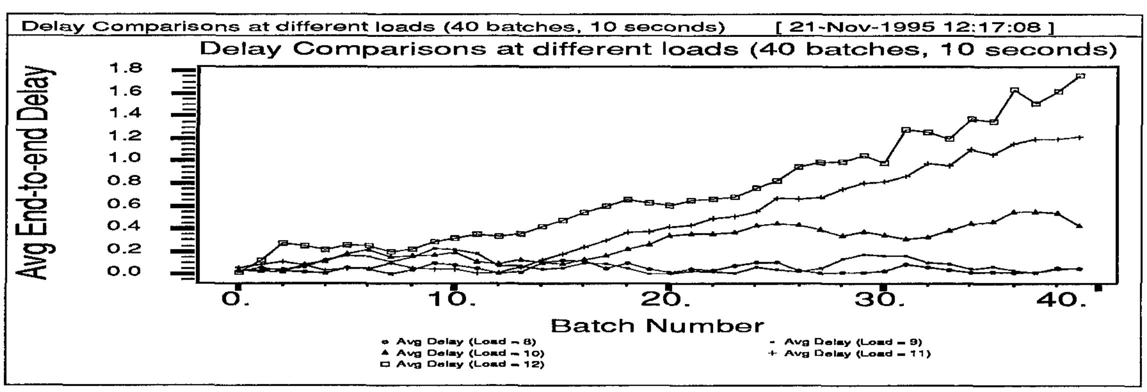

Figure 3.9 Delay Comparisons at Different Loads

The results depicted from Figure 3.9 show that stabilization seems to occur with loads equal to eight (687,680 bps) and nine (773,640 bps), and possibly with a load equal to ten (859,600). In order to further investigate ten as a candidate, another simulation which runs 10 times longer than the previous simulation has been accomplished. The results (Figure 3.10) show that steady state has still not been achieved. Since a steady state is not achieved with load equal to ten, the best candidate appears to be equal to nine.

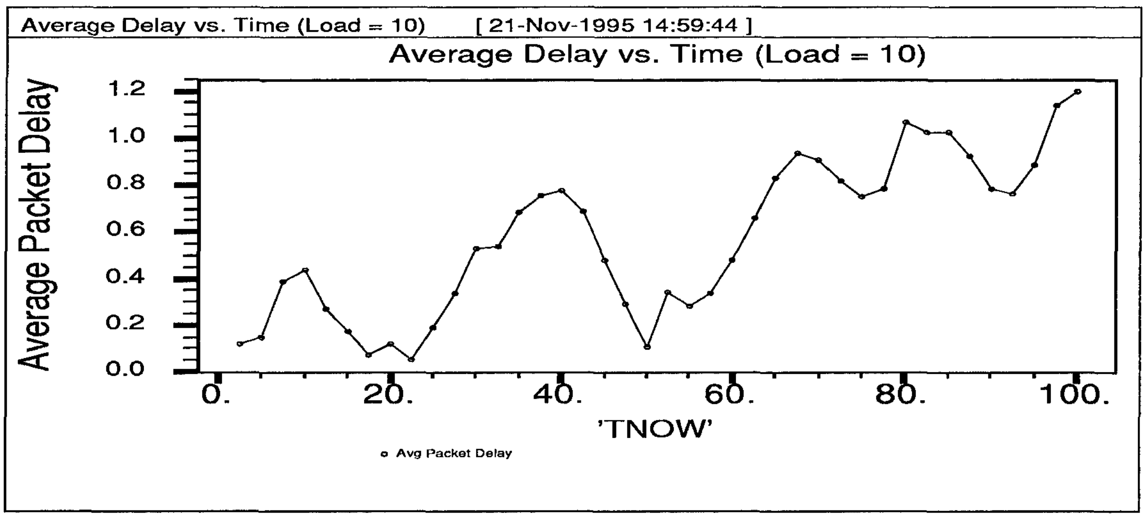

Figure 3.10 Average Delay vs. Time (Load = 859,600 bps)

The next decision to make is to determine the warm up time. A warm up time is required to eliminate bias caused by initial transients. Unfortunately, there are not too many methods that perform well in determining a warm up time [LaK91]. However, Law and Kelton, have developed an algorithm which works very well on a variety of stochastic models. Welch [LaK91] came up with a graphical procedure, which implements the algorithm. This procedure essentially allows you to select the warm up time by observing where the curve smoothes out. Welch's procedure is also called Welch's Algorithm. The algorithm is implemented in this experiment by taking 120 observations of the average end-to-end delay, each occurring at equally spaced time intervals (every .25 seconds) over 30 seconds simulation time. Ten independent trials are run and the values at each observation point are averaged together to produce a vector of 120 average values corresponding to each of the observed points in time. A window is moved through time in order to get a moving average. The window size is increased until a "reasonably smooth" curve is obtained. Law and Kelton [LaK91] use the term "reasonably smooth" to indicate that the curve on the graph should initially increase in magnitude and then level off and appear to have little jaggedness. The results obtained from applying this algorithm are shown in Figure 3.11. Notice that the curve never levels out, and as such, indicates a steady state has not been achieved. Therefore, a load equal to nine is ruled out.

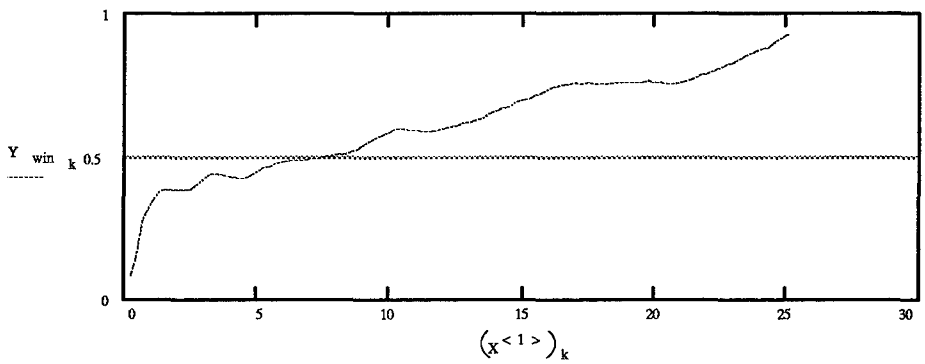

Figure 3.11 Attempt to establish warm up time using Welch's Algorithm

Welch's Algorithm is again applied with the load reduced to 8. The results shown in Figure 3.12 indicate that the curve levels off when 'TNOW' equals two seconds. As such, the warm up period to be used in the simulation runs is set to two seconds.

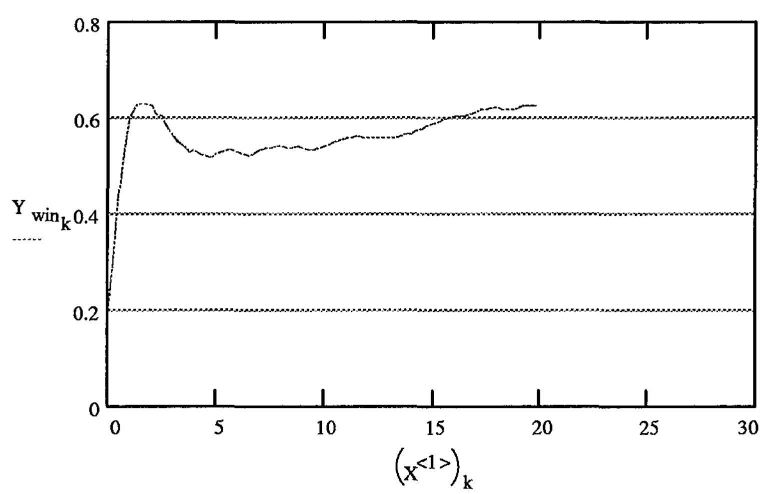

Figure 3.12 Warm Up Time Determination (Welch's Algorithm)

To further confirm a load equal to eight as a heavy load, it is desirable to see if the bandwidth utilization is high. The results (Figure 3.13) show an approximate 85% bandwidth utilization. This indicates a reasonably heavy amount of traffic which does not saturate the system.

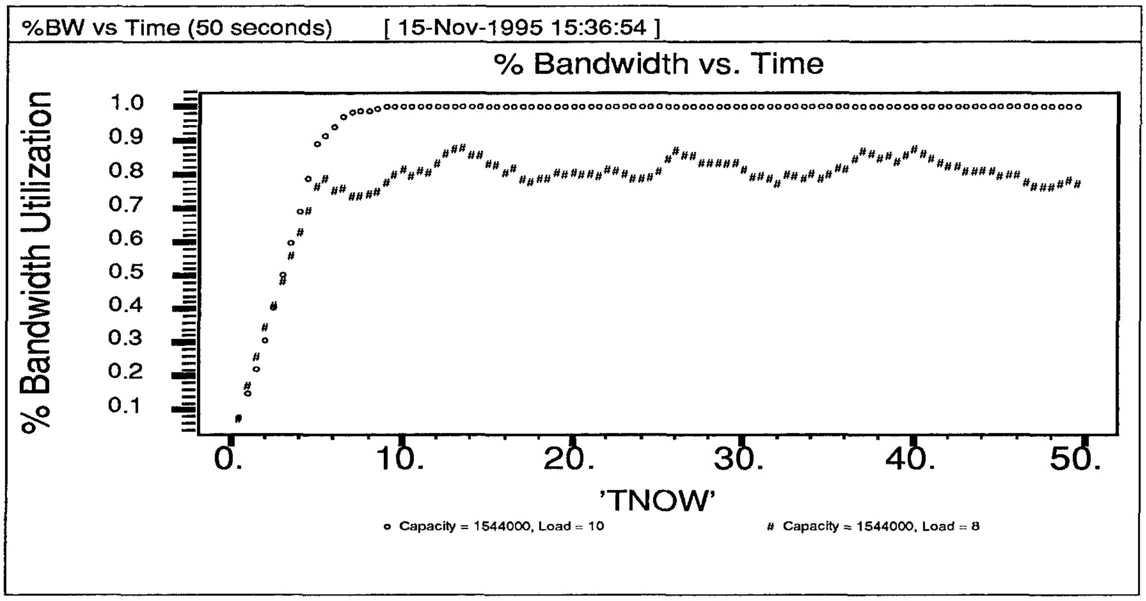

Figure 3.13 Bandwidth Utilization vs. Time

Notice in Figure 3.13 that 100% bandwidth utilization (load = 10) occurs for a steady period of time. This is slightly misleading. In actuality, when flow control mechanisms (see Chapter 2, Section 2.4.4) are implemented, time-outs (due to excessive delay times and dropped packets from buffer overflow) will be experienced by the sources. The sources of the traffic will assume the time-outs are due to congestion and may simultaneously stop sending packets, causing a rapid decline in bandwidth utilization (this scenario is depicted in the top diagram in Figure 2.6). In TCP/IP, as soon as a certain amount of time expires between the time a sender transmits a packet until an acknowledgment is received, a sender will assume congestion and go into a 'slow start' state. When in the 'slow start' state the sender decreases its window size to one segment. For instance, the sender will send one packet and will not send another until an acknowledgment is received. The window size will begin to grow gradually as acknowledgments arrive [Fei93]. The sender will also retransmit any packets for which it did not receive an acknowledgment. Thus, with flow control mechanisms in place, a 100% bandwidth utilization might not ever be achieved. It is assumed in this experiment that flow control mechanisms are in place and that a few packets will be retransmitted due to excessive delay and buffer overflow, but this traffic is negligible providing system is not subject to overload [CaD91]. In other words, the system is assumed to be operating in the noncongested region. The noncongested region is considered to be the area to the left of the congested region as shown in Figure 2.6.

From the above results, the heavy load is chosen to be equal to eight, and a light load is chosen somewhat arbitrarily to equal half of the heavy load (four) and a medium load is chosen to equal six. Further, the warm up time is chosen to equal two seconds. Law and Kelton [LaK91] recommend a simulation run time in the steady state to be significantly greater than the warm up time, and therefore, the simulation run time is set to twenty seconds. And finally, the number of runs in the following simulations will be a minimum of three [LaM94,ShD94].

3.5 Five Node Experiment

In the two node network simulations (described in the previous section), the nonshared system has twice the link capacity of the shared system in all runs. In this experiment, both the nonshared and the shared system have the same link capacities. In fact, all fixed parameters (i.e., topology, node processing speed, etc.) are set to the same values in both systems to the maximum extent possible. In the nonshared system the links are dedicated to a single source-destination (i.e., LAN A0 to LAN B0). Moreover, the nonshared system is configured to allow full connectivity to all the sites, and are set up using the same topology as the shared system. Further, the nodes perform store and forwarding only. As for the shared system, the links are shared by all source-destination pairs. Additionally, routing is performed at the packet-switching nodes.

Since routing is incorporated, additional details about TCP/IP are presented. More details on how the host communicates across the network and on the buffer management scheme implemented by the nodes are also covered. After the objective of the experiment is stated, these network issues are described in more detail. The other topics to be covered are as follows: 1) operating assumptions, 2) network model, and 3) steady state analysis. Verification and validation are discussed in Sections 3.6 and 3.7.

3.5.1 Objective

The purpose of this experiment is to present a method (via simulation) which can be used compare the performance between a shared bandwidth system and a nonshared system. Both systems are examined in terms of percent bandwidth utilization (productivity) and average end-to-end delay (responsiveness). The combined channel capacities of each link in the nonshared system are equal to the single channel capacity used in the shared system. This allows a fair performance comparison to be made. Another objective is to determine how the performance parameters (% bandwidth utilization and average end-to-end delay) are affected by changing the load (with fixed capacity) and changing the capacity (with fixed load).

3.5.2 TCP/IP on the Host

An example on how TCP/IP is implemented on the host of a network is shown in Figure 3.14. At each layer, a header, which contains control information, is attached to the data. The combined data plus header is called a protocol data unit (PDU). As the PDU is passed down from the TCP layer to the network (IP) layer, it is referred to as a segment. In the network (IP) layer (sometimes called the internet layer), the source and destination network address are appended to the segment. These addresses (32 bits in length) are globally unique and are used to identify the source and destination host across a wide-area network. The resulting PDU, referred to as a datagram, is then passed down to the network access layer. In this layer, a physical header, which contains a physical source and destination address, is appended to the datagram. These addresses identify the local source/destination addresses which are unique to the network. Other information included in the physical header is dependent upon the type of physical network the host is connected to. Moreover, the media access control mechanism is also very dependent on the type of network the host is attached to (i.e. Ethernet, FDDI). When the PDU is passed onto the physical medium, it is referred to as a packet. Other types of information (i.e., checksum) are included in the header at each of the other layers as well.

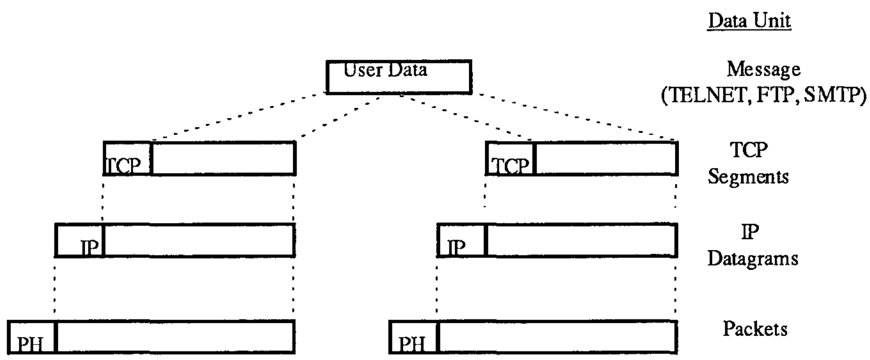

Figure 3.14 Layers on the Host [Sta88]

The TCP layer is normally implemented only on the host systems, and not on the packet-switching nodes. This layer provides reliable, flow-controlled, end-to-end, stream service between two hosts of arbitrary processing speed using the unreliable IP service for communication. According to Comer [CoS91], TCP is carefully constructed to handle delayed, duplicated, lost, delivered out of order, corrupted, and truncated packets. Thus, concerning this research effort, it is the function of the host to provide this type of service, and it will not be explicitly modeled into the system. It is assumed that the traffic generated from the LANs come from host-to-host connections which implement the above services. Further information on what takes place at this layer can be found in [Sta88,CoS91,Fei93].

3.5.3 Buffer Management

Another important issue is the memory management scheme. Comer [CoS91] describes a large buffer management scheme which allocates buffers that are capable of storing the largest possible packet size (in this case 512 bytes). For the purposes of this research, it is assumed that the large buffer management scheme will be used and that fixed size partitions of memory (number of buffers) will be allocated to each queue (i.e., each incoming and outgoing queue is allocated 100 buffers). Other buffer management schemes can be found in Chapter 2 and in [CoS91].

3.5.4 Operating Assumptions

TRAFFIC

(a) The number of packets generated for bulk and interactive traffic are approximately the same [CaD91].

(b) The traffic is bi-directional. Traffic loads between source-destination pairs will be equal and generated in opposite directions and independently. It is assumed that a portion of the packets being sent are responses to received data packets and include acknowledgments.

(c) Interactive traffic will be generated independently from the bulk traffic, and the interarrival time between packets will follow an exponential plus constant distribution.

(d) Bulk traffic will be generated by pulse train. The interarrival time between bursts will follow an exponential distribution.

(e) Number of packets generated during each burst follows a geometric distribution with mean equal to eight. In order to generate more realistic traffic with flow control implemented, the trains will not exceed 16 packets in length [IlM85,Sha92].

NODES

(f) The node buffer scheme allocates a fixed number of buffers to each queue. Each buffer is set to 512 bytes in length.

(g) Node processing time is assumed $N(\mu,\sigma^2)$, with $\mu = 100 \text{ X } 10^{-6}$ s and $\sigma = 10 \text{ X } 10^{-6}$ s.

(h) Bandwidth is fairly shared among communicating entities without any prioritization scheme.

NETWORK

(i) Errors due to channel noise are negligible.

(j) A new variable parameter was added so that the distance between the links could be set up individually. In order to simplify the model, the distance was set to 300 miles, somewhat arbitrarily.

(k) The Local Area Networks (LANs) connected have high bandwidth and transmission and access delay from the node to the local networks is negligible.

3.5.5 Network Model

3.5.5.1 Packet Structure

The fields in the packet structure for the nonshared system are shown in Figure 3.15. The size field is equal to the Maximum Transmission Unit (MTU) of 4096 bits for bulk traffic, and is exponentially distributed with mean equal to 450 bits for interactive traffic. The minimum size of a packet is 360 bits [Fei93]. The origin is set as follows: '0' if originating from LAN A0, '1' if originating from LAN A1, '2' if originating from LAN B0, '3' if originating from LAN C0, '4' if originating from LAN D0, '5' if originating from LAN D1, and '6' if originating from LAN E0. Time Created is a real number which represents the time the packet is created in the traffic source. Time Finished represents the time the packet reaches its destination and exits the system.

Name: packet [A five node network]

Date: Tuesday, 1/16/96 04:40:33 pm EST

Name	Type	Subrange	Default Value
origin	INTEGER	[0, +Infinity)	0
destination	INTEGER	[0, +Infinity)	0
time created	REAL	[0, +Infinity)	0.0
type	INTEGER	[0, +Infinity)	0
time finished	REAL	[0, +Infinity)	0.0
size	INTEGER	[0, +Infinity)	360

Figure 3.15 Packet Structure for Nonshared System

The packet structure for the shared system shown in Figure 3.16 is very similar to the nonshared system's, with the exception of two additional fields, 'source node' and 'destination node.' Before discussing these fields, the addressing scheme must be addressed.

Name: WAN packet [five_node_shared]
Date: Tuesday, 1/16/96 04:37:34 pm EST

Name	Type	Subrange	Default Value
origin	INTEGER	[0, +Infinity)	0
destination	INTEGER	[0, +Infinity)	0
source node	INTEGER	[0, +Infinity)	0
destination node	INTEGER	[0, +Infinity)	0
size	INTEGER	[0, +Infinity)	360
Time Created	REAL	(-Infinity, +Infinity)	0.0

Figure 3.16 Packet Structure for Shared System

Since there are only eight LANs and five nodes, the IP address scheme has been simplified. Table 3.2 shows the addressing scheme used in the model. Because the model is only concerned with LAN to LAN traffic, and not host-to-host, a host identifier is unnecessary. The 'origin' and 'destination' fields will contain the 'Subnetwork Identifier' and the 'source node' and 'destination node' fields will contain the 'Network Identifier.' For instance if LAN A0 sends a bulk packet to LAN D1, the packet would look like that shown in Figure 3.17.

Table 3.2 Addressing scheme used in the model

NAME	Network Identifier	Subnetwork Identifier	Host Identifier
LAN A0	0	0	x
LAN A1	0	1	x
LAN B0	1	2	x
LAN C0	2	3	x
LAN D0	3	4	x
LAN D1	3	5	x
LAN E0	4	6	x

origin	destination	source node	destination node	size	time created
0	5	0	3	4096	xxxx

Figure 3.17 Example of a Packet being transmitted from A0 to D1

3.5.5.2 Shared System

The first system to be discussed uses shared bandwidth (Figure 3.18). At the system level, the diagram includes seven LANs (denoted as LAN traffic) and five nodes. The interconnecting links operate as full-duplex and the value of each link's capacity can be varied (in 64 kbps increments) by the six parameters shown (i.e., 'Link A to B Capacity'). Additionally, there is a module which initializes the traffic, costs, and routing matrices and is shown in the upper right corner. And finally, a compute statistics block calculates the performance measures of the system. Each of these modules will now be discussed in turn, beginning with the 'Init Traffic and Cost Matrices' module.

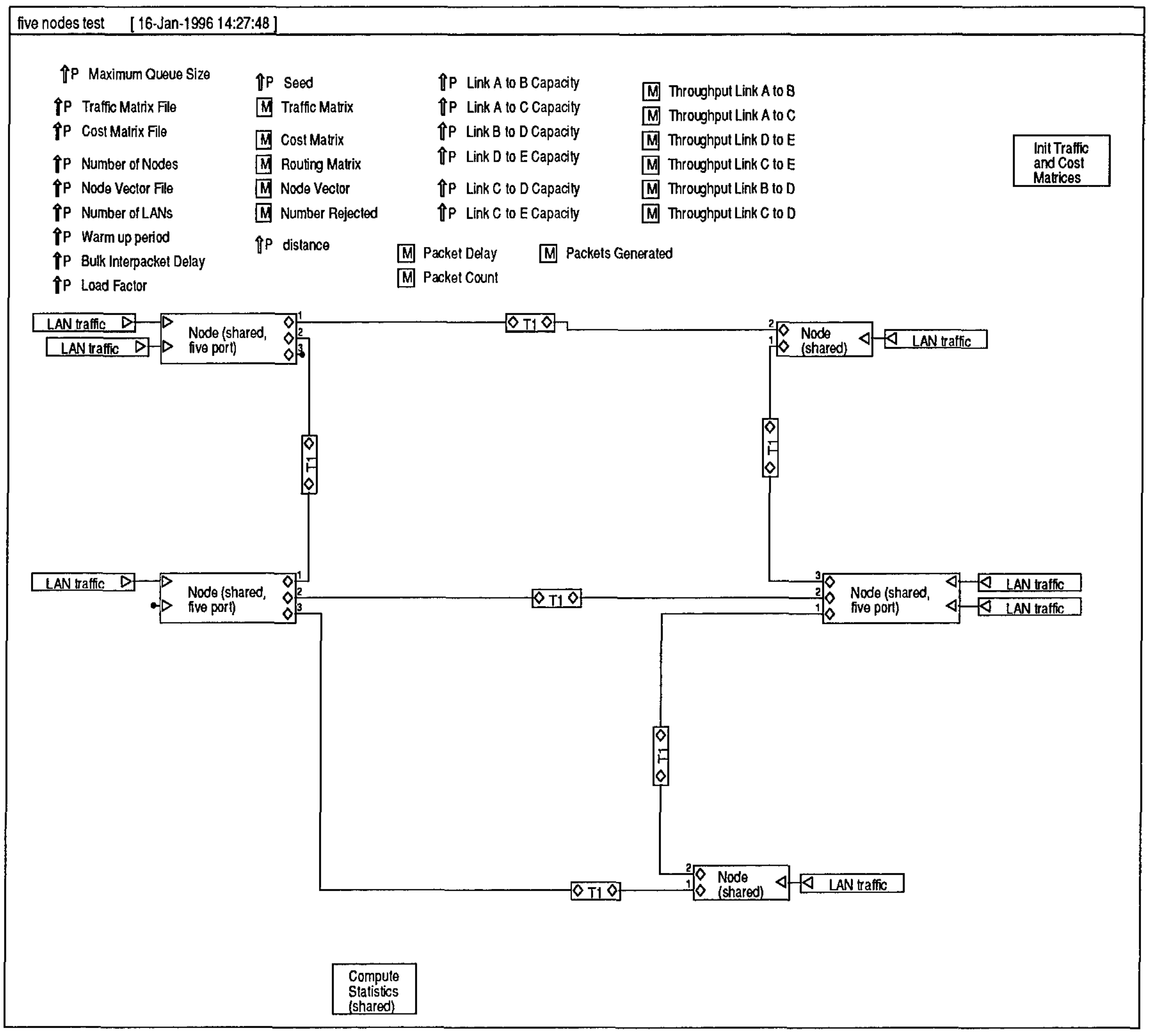

Figure 3.18 Five Node Shared System

The 'Init Traffic and Cost Matrices' module shown in Figure 3.19 reads in the 'Traffic Matrix' and the 'Cost Matrix' from text files and stores them in memory as a matrix structure. The 'Compute Route Matrix' module uses Dijkstra's Algorithm to compute the least cost paths. Instead of providing a separate routing table for each node, the routing matrix, which specifies the routing information for the entire network, is set as global memory, so that it can be accessed by all the nodes. A node's routing table corresponds to a row in the routing matrix. The 'Compute Route Matrix' module and its inner modules are shown in the Appendix, Figures A 8, A 9, and A 10.

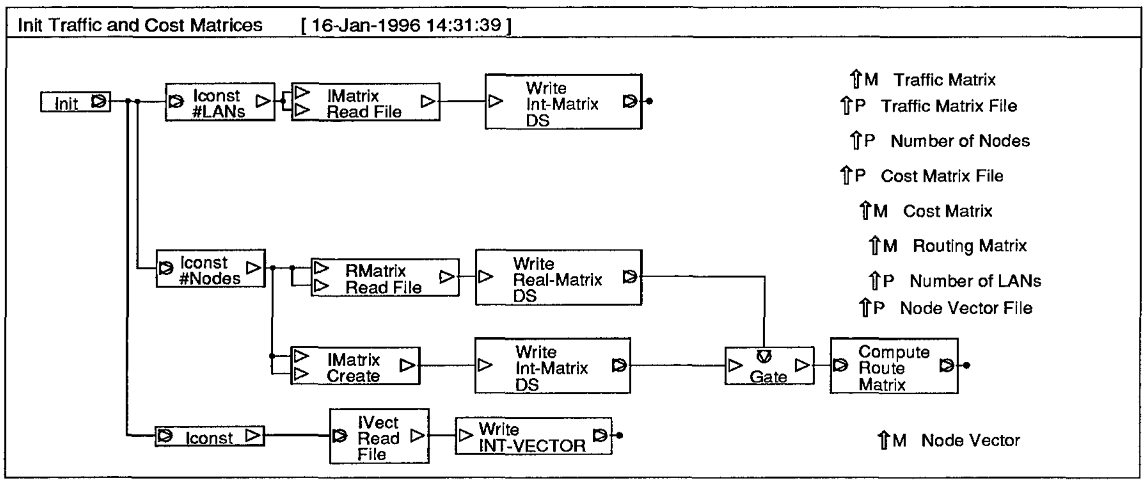

Figure 3.19 Init Traffic and Cost Matrices Module

An illustration of the 'Traffic Matrix' is shown in Figure 3.20. Each row and column correspond to the source and destination respectively. Each element within the matrix is a factor used to generate the traffic load. The traffic load is calculated differently than it was for the two node experiment. In order to vary the traffic load, an external parameter, called 'Load' has been created. The actual traffic load is based on the mean arrival rate of pulses (generated from an exponential distribution) and is calculated by multiplying the 'Load' variable to the 'Traffic Matrix' entry. As discussed previously, two types of traffic are generated: bulk and interactive (i/a). In generating bulk traffic, each pulse triggers a train of packets (generated from a geometric distribution with mean of eight). In the two node network, the amount of interactive traffic generated was set to approximately the same amount of bulk traffic generated. In Chapter 2, some of the performance studies have been based on bulk traffic alone

[Jai90,Sha92,Zha90], while yet others have been based on various mixtures of bulk and interactive traffic [CaD91]. In order to stay more consistent with Caceres data [CaD91], an approximately equal number of interactive packets and bulk traffic packets will be generated. As such, a constant set equal to eight is used as a factor and is multiplied by the 'Load' and the 'Traffic Matrix entry' for generating interactive traffic. Subsequently, this is the arrival rate of packets (generated from an exponential plus a constant distribution). Table 3.3 illustrates the loads in terms of bits per second (bps) for the corresponding 'Load.'

```
0 0 8 8 8 8 8
0 0 8 8 8 8 8
8 8 0 8 8 8 8
8 8 8 0 8 8 8
8 8 8 8 0 0 8
8 8 8 8 0 0 8
8 8 8 8 8 8 0
```

Figure 3.20 Traffic Matrix (each site generates an equal amount of traffic)

Table 3.3 Load in terms of bits per second (L=Load, TM = Traffic Matrix Entry)

Load	Bulk PPS (L*TM*8)	Bulk BPS (PPS*4096)	I/A PPS L*TM*8	I/A BPS (PPS*450)	Tot BPS (B.BPS+I/A.BPS)
3	192	786,432	192	86,400	872,838
2	128	524,288	128	57,600	581,888
1.5	96	393,296	96	43,200	436,416
1	64	262,144	64	28,800	290,944
0.66667	42.667	174,764	42.667	19,200	193,964
0.33337	21.333	87,380	21.333	9,600	96,981
0.16667	10.667	43,692	10.667	4,800	48,492
0.1	6.4	26,189	6.4	2,880	29,069

The cost matrix shown in Figure 3.21 is used to find the least cost path from one node to the other. Each entry indicates the cost of traveling from the node represented by the row number to the node represented by the column number. A value of 1.0e6 indicates there is no connectivity. The value of '2' is used to indicate connectivity at one hop. A cost of '3' is also used to indicate one hop connectivity, just as '2' does, but in order to keep the traffic flowing the same way in both the shared and the nonshared system, a higher cost had been added to this link. A node vector is also read into this module and indicates which node the LANs are connected to as shown in Figure 3.22

1.0e6	2	3	1.0e6	1.0e6
2	1.0e6	1.0e6	2	1.0e6
3	1.0e6	1.0e6	2	2
1.0e6	2	2	1.0e6	2
1.0e6	1.0e6	2	2	1.0e6

Figure 3.21 Cost Matrix

LAN	Node
0	0
1	0
2	1
3	2
4	3
5	3
6	4

Figure 3.22 Node Vector

The next module to be discussed is the 'LAN traffic' module shown in Figure 3.23. Each 'LAN traffic' module is constructed identically. Inside of this module, the 'Start Traffic' sub module sets the packet's 'origin' field to the parameter 'Node Identity.' Each column of the 'Traffic Matrix' is then read along a single row corresponding to the 'origin' field. If the entry is nonzero, then the column number (destination) will be outputted. The destination values received from the 'Start Traffic' module are submitted to both the upper and lower half of the 'LAN traffic' module. The upper half implements the bulk traffic using the pulse train, and the lower half implements the interactive traffic using an exponential plus constant interarrival time distribution. The mean to be used for the time between train pulses is equal to the 'Load Factor' (= 1/'Load') divided by the 'Traffic Matrix' entry. Sub module 'Build Bulk Packet' will generate a train of packets each time it receives an input pulse. The input pulse has a value associated with it: the 'destination'. This value will be inserted into the 'destination' field of each packet generated. The output from 'Build Bulk Packet' will be a train of MTU sized packets, and additionally, the destination value will be sent back to the random generator, so that it can be recycled. The mean interarrival time between packets in the lower half is equal to the 'Load Factor'(= 1/'Load') divided by the product given by 'Traffic Matrix' entry times eight (eight is used to generate approximately the same number of interactive packets as there are bulk). The sub modules, 'Start Traffic,' 'Build Bulk

Packet' and 'Build Interactive Packet,' can be found in the Appendix, Figures A 11, A 12, and A 13. And finally, the sub module 'Record packets generated' sets the variable 'Packets Generated' to the total number of packets which have been generated. This variable is used for computing statistics. See Figure A 14 in the Appendix for an illustration of this module.

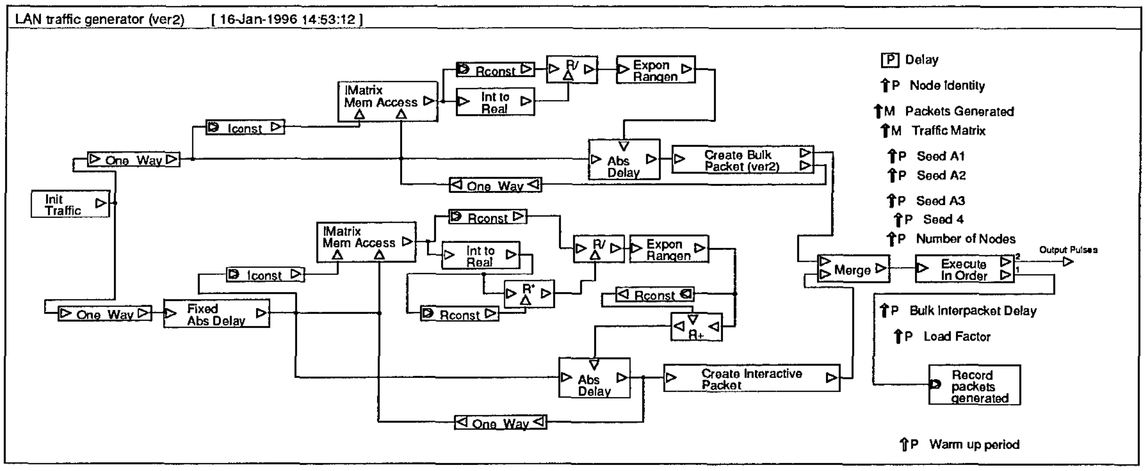

Figure 3.23 LAN Traffic Module

Each interconnecting link consists of a 'T1' module as shown in Figure 3.24. Inside this module, the packets encounter a propagation delay as determined by the 'distance' between nodes divided by the speed of light. The 'Compute Throughput' sub module begins calculating throughput by adding up all the bits that pass through. It does not initiate counting until after a 'Warm up period' has expired to prevent any initial bias. The 'Compute Throughput' module is shown in Figure A 15.

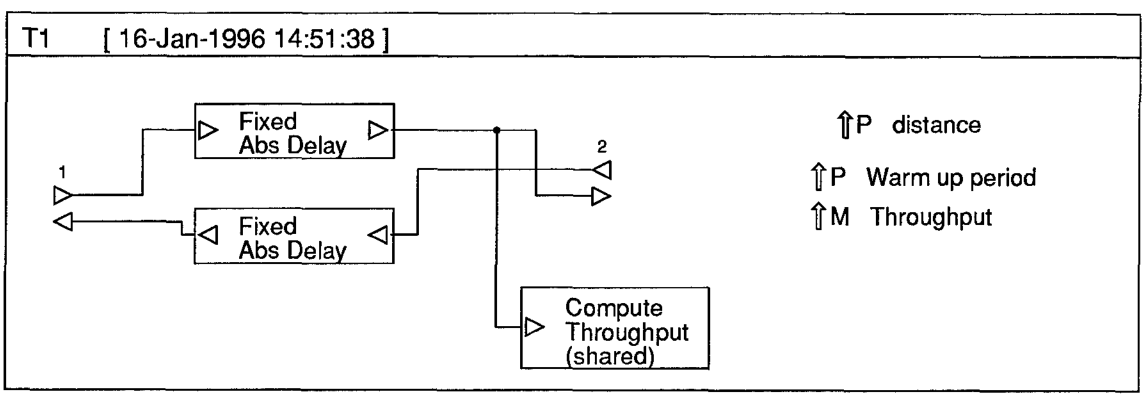

Figure 3.24 T1 Module

There are two types of nodes used in the design, a five port and a three port. Since they are both similar in construction, only the five port node is discussed. The five port node, shown in Figure 3.25 basically consists of two layers: the IP layer and the network access layer. Additionally, there is a 'Compute End-to-End Delay' module used for statistics collection.

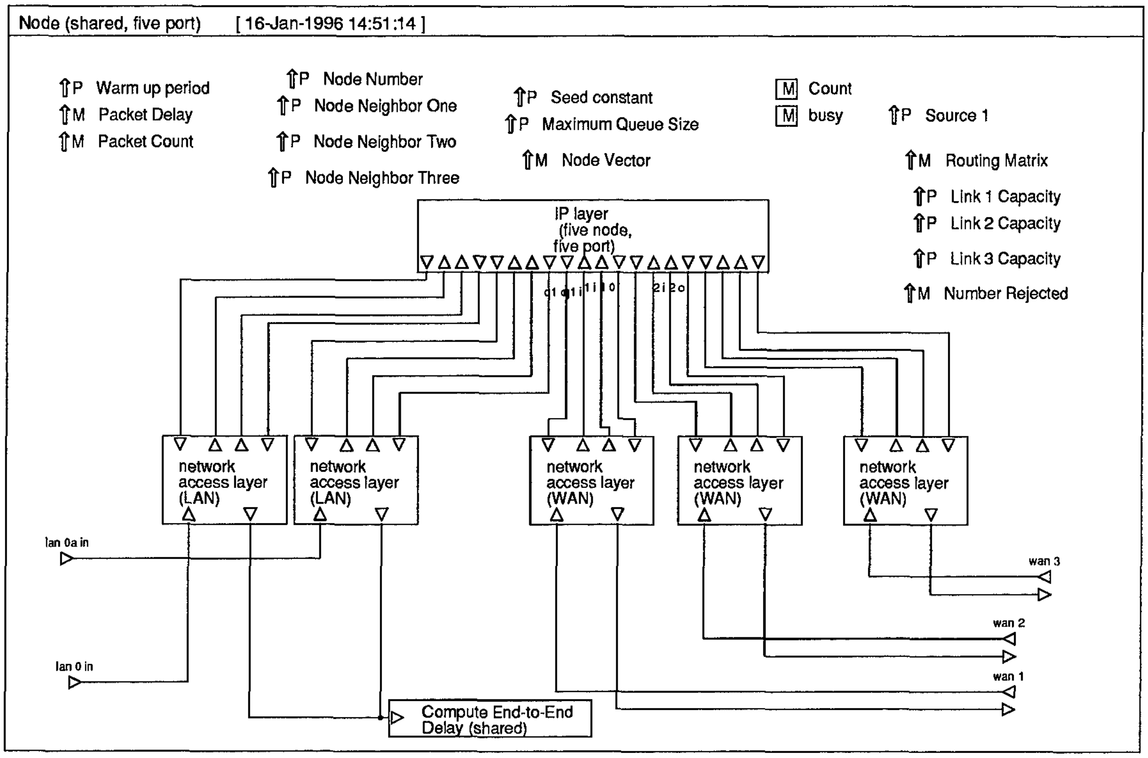

Figure 3.25 Node Module.

The 'IP layer' module is shown in Figure 3.26. When a packet enters this module, the first action taken is that the queue number from which the packet came from is written to a local memory variable named 'Queue Number.' Immediately before the packet begins processing, the variable 'Count' (used in the round robin scheme) is set to zero, and the variable 'busy' is set to 1 (true), indicating that the processor is busy. When processing of a packet has completed, two things happen: 1) the packet is sent to the 'Route' module, and 2) the packet is sent to 'Get next packet.' In the 'Route' module, the 'Routing Matrix' is consulted, the appropriate link is selected, and the packet exits the appropriate output port. When the packet arrives at 'Get next packet,' it acts as a trigger to get the next packet. 'Get next packet' implements the round robin scheme. It does so by first incrementing 'Queue Number' by one, so that the queues can be examined in sequence. If there is no packet in the next queue, then the process will repeat, each time examining the queues in sequence. If there are no packets in any of the queues, the procedure will quit when the 'Count' variable (which gets incremented by one each time a queue is examined) reaches the number of queues indicated by the 'Number of Links' variable. If this happens, the variable 'busy' will be set to zero, indicating that the server is not busy. For illustrations of the 'Get next packet module' and its inner modules, refer to the Appendix, Figures A 16, A 17, and A 18. Illustrations of 'Route' and its inner modules can be found in Figures A 19, A 20, and A 21. And finally, an illustration of the 'Process Delay' module is the same as that for the two node network and is shown in Figure A 3.

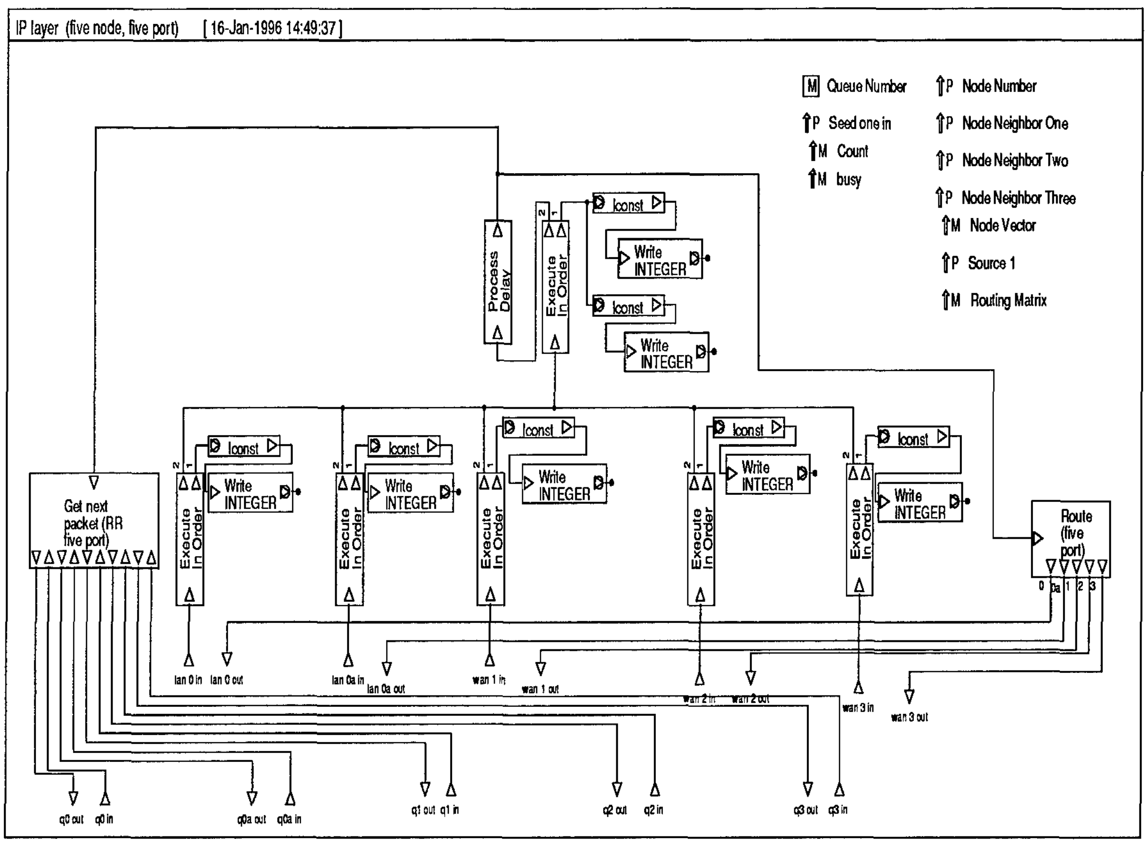

Figure 3.26 IP layer Module.

The 'network access layer (WAN)' module is shown if Figure 3.27. When a packet enters this module from the physical link, it first checks to see if the server is 'busy.' If the server is not busy, the packet will immediately flow through the queue and into the 'IP layer.' Otherwise, the packet will be stored in a FIFO queue. If a signal is received from the 'IP layer' to request a packet, the incoming queue 'General Queue 2.0' is checked to see if there are any packets waiting to be served. If so, the queue is triggered to release the leading packet. When a packet arrives from the 'IP layer' module, the network access layer will send it to the transmission process. The transmission processing time is determined by the size of the packet divided by the 'Link Capacity' variable. If another packet is currently being transmitted, then the newly arrived packet will be stored in the 'FIFO' (First In First Out) queue. The 'Record packets rejected' module keeps a count of the number of packets rejected. Each time a packet is rejected the global variable, 'Number Rejected' gets updated. The 'Transmission Process' module is identical to the one used for the two node network and is shown in Figure A 4. For an illustration of module 'Record packets rejected,' refer to the Appendix, Figure A 22. The network access layer (LAN) module is similar in construction and is shown in Figure A 23.

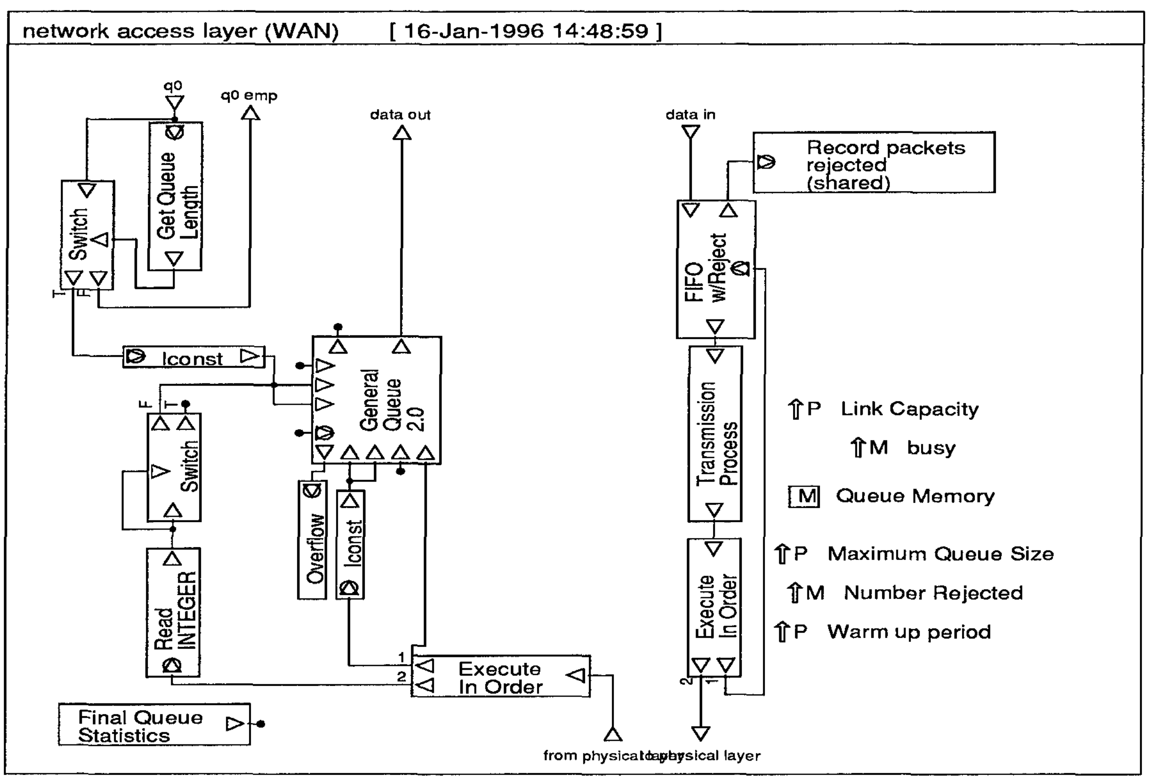

Figure 3.27 network access layer (WAN).

Lastly, the 'Compute End-to-End Delay' module (shown in Figure A 24) is used to continuously update the global end-to-end delay variable 'Packet Delay,' which is used in the next main module to be discussed.

The last main module to be discussed in the shared system is the 'Compute Statistics' module. This module is shown in Figure 3.28. The main purpose of this module is to compute global statistics such as average end-to-end delay of all packets generated, total number of packets generated and rejected, and to compute percent bandwidth utilization of each of the links. In order to calculate percent bandwidth utilization, the capacity and the throughput of each link (i.e., 'Link A to B Capacity' and 'Throughput Link A to B') are passed in as parameters. For an illustration of the 'Compute % BW utilization' and 'Compute Average Delay' modules, refer to Figures A 25 and A 26.

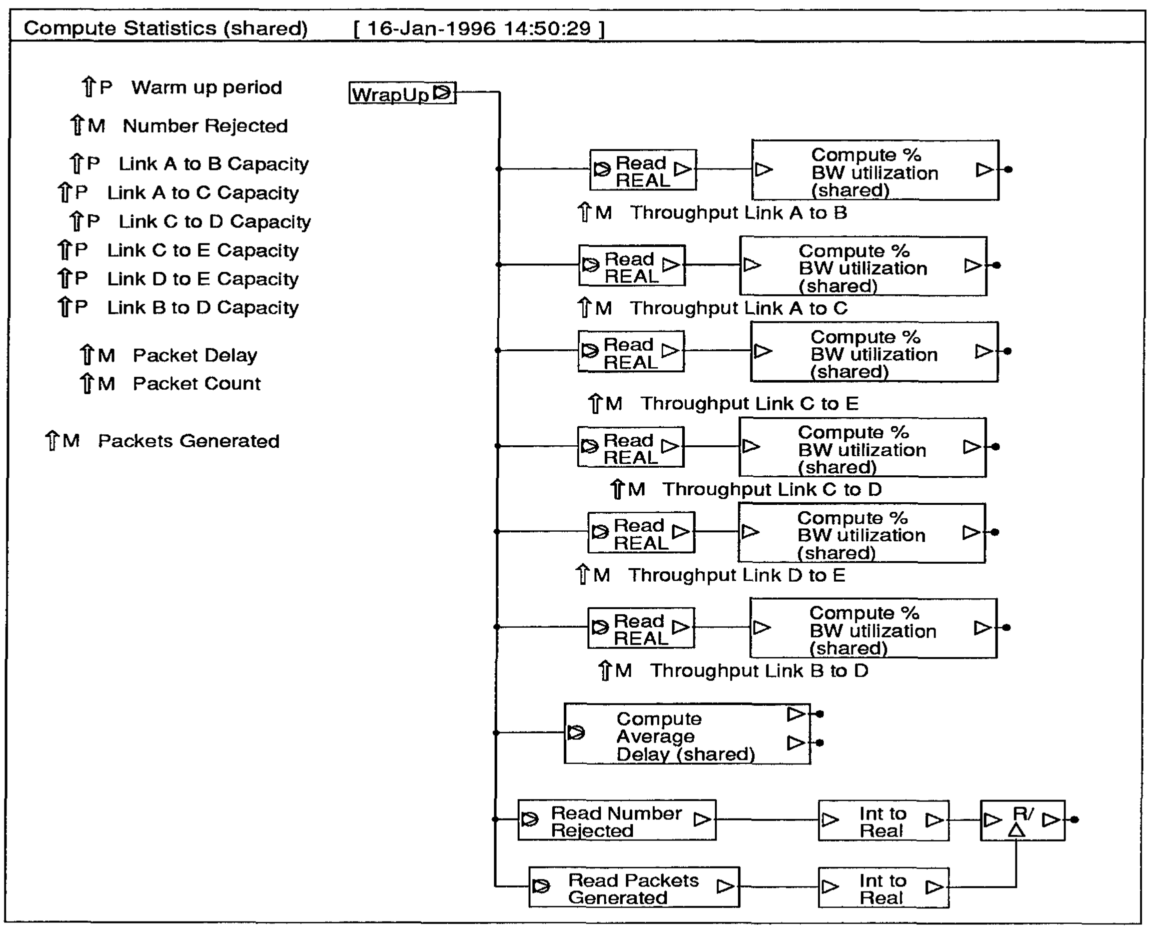

Figure 3.28 Compute Statistics Module

3.5.6 Nonshared System

The nonshared system is shown in Figure 3.29. The 'Initialize Traffic Matrix' and 'Compute Statistics' modules basically operate the same as in the shared system, except that no routing matrix is created. The major difference between this system and the shared system is that the interconnecting links are now dedicated to a single source-destination pair. Moreover, because there are now between one to five nodes at each of the sites, the topology shows (for the sake of simplicity) the sites rather than the nodes.

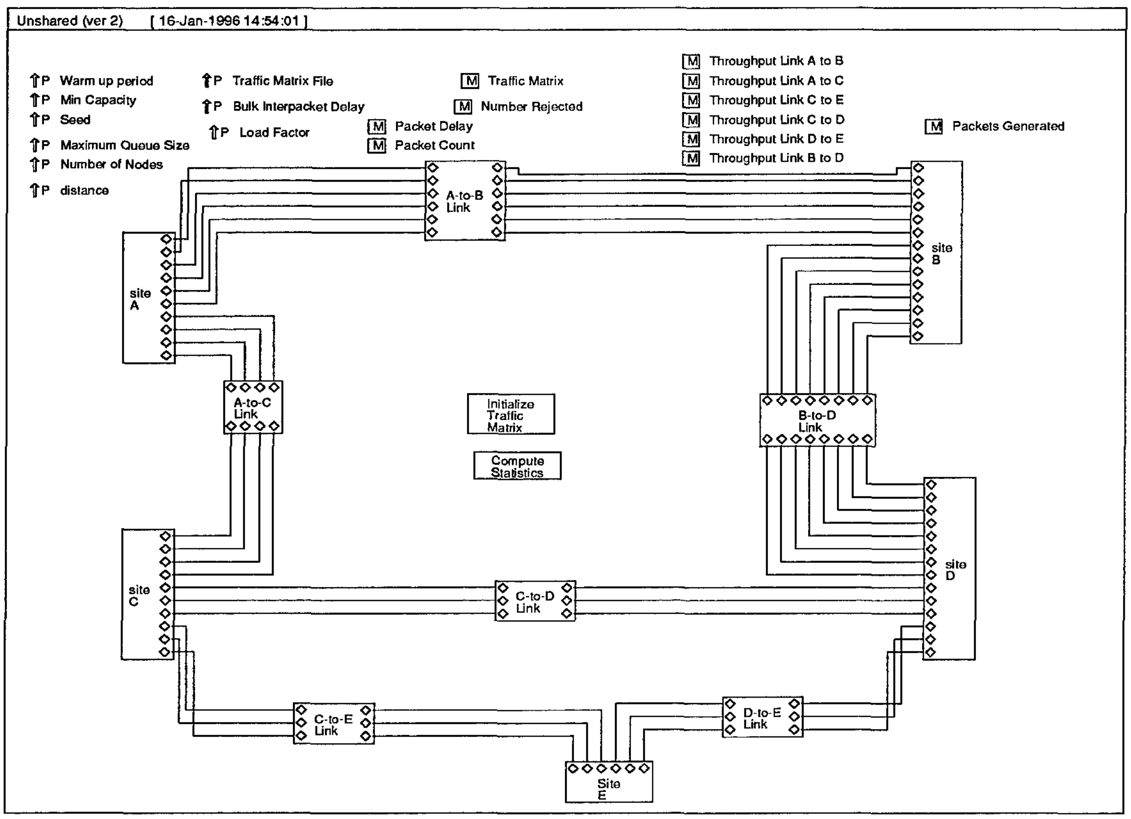

Figure 3.29 Nonshared System

The interconnecting links are similar in construction, and therefore only one will be illustrated. The 'A to B Link' module is shown in Figure 3.30. Each link is full-duplex, and although dedicated to a single source-destination pair, the throughput is calculated as if the links were combined into a single link. Since the traffic is equal in both directions, only one line in each of the links is monitored.

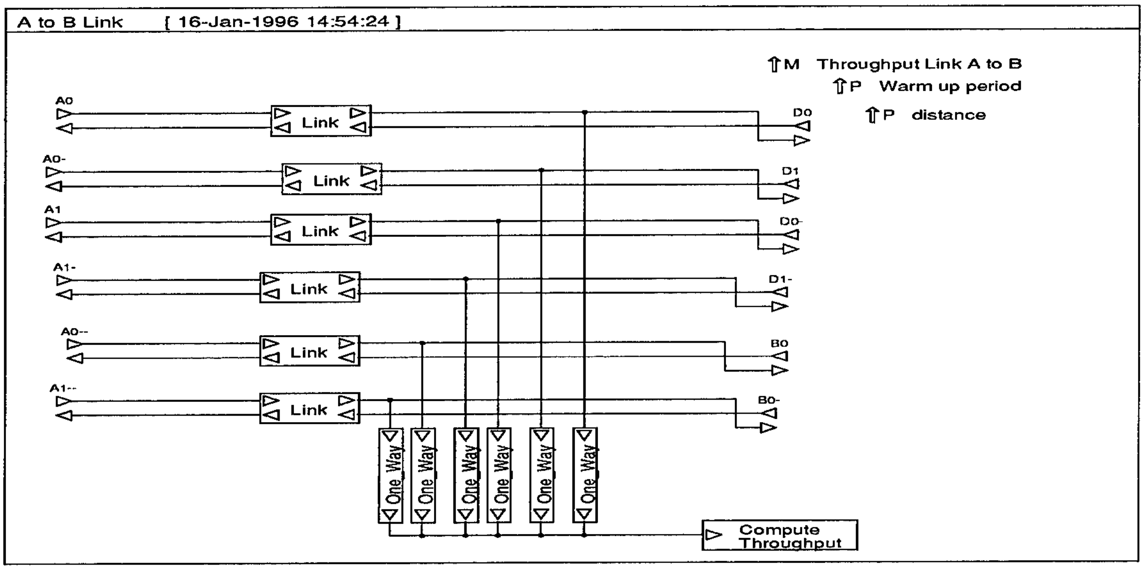

Figure 3.30 A to B Link Module

The site modules are constructed similarly, and therefore only one site is shown. The 'Site C' module is shown in Figure 3.31. The other site modules can be found in the Appendix, Figures A 27 - A 30. This module contains a LAN traffic generator (same as the one used for the shared system) and two types of nodes.

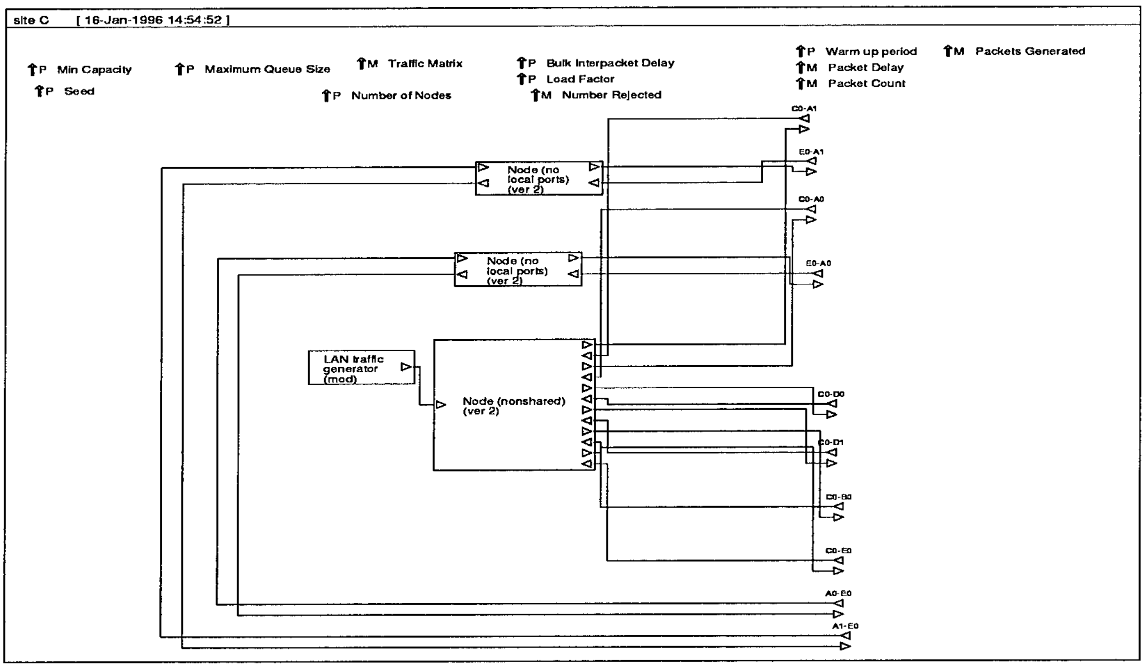

Figure 3.31 Site C Module.

The node 'Node (no local ports) (ver 2)' is simply a store and forward type node and is shown in Figure 3.32. The 'network access layer (Link)' modules are constructed identically to the shared system's 'network access layer (WAN)' module shown in Figure 3.27. In the 'Switching Layer' module, a processing delay is incurred, and after processing, both the incoming and outgoing queues are checked for another packet waiting to be processed in round robin fashion. Figure A 31 in the Appendix shows the 'Switching Layer' Module.

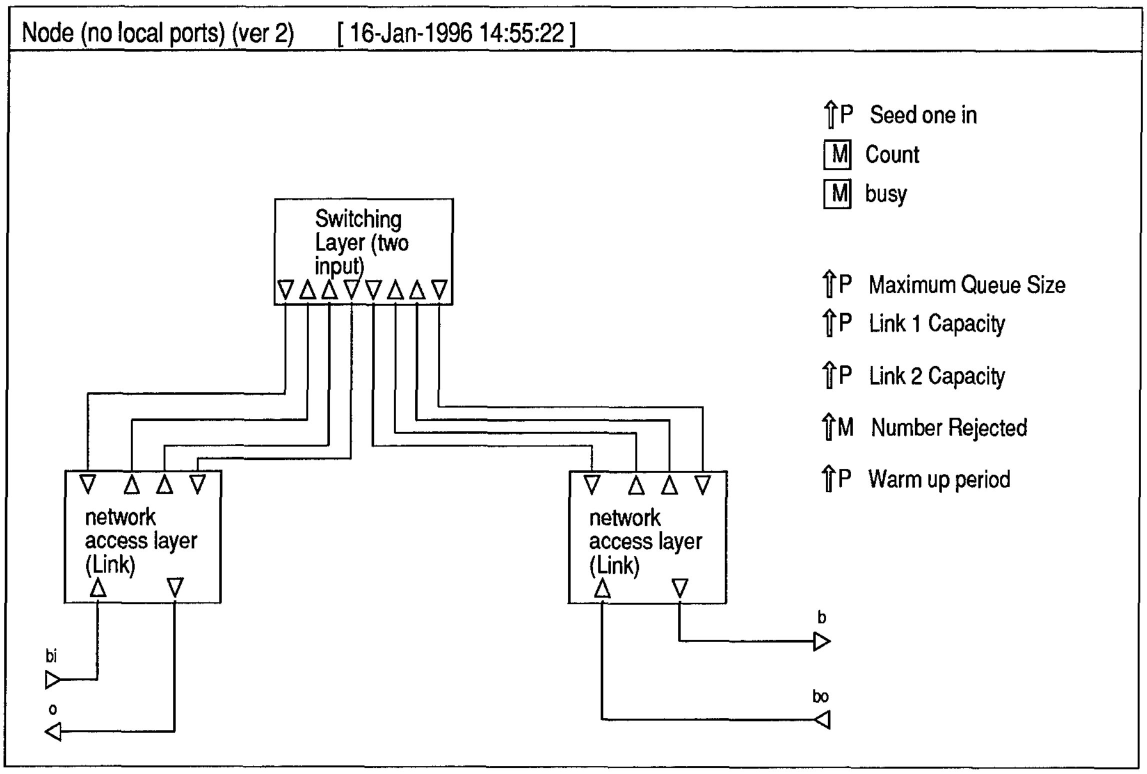

Figure 3.32 Node (no local ports) Module

The node labeled 'Node (nonshared) (ver2)' is constructed as shown in Figure 3.33. This node basically operates the same as 'Node (no local ports) (ver 2)' which was just previously described. It does however, contain a 'network access layer (local)' and a 'Compute End-to-End Delay' module. The 'network access layer (local) allows a LAN traffic generator to be connected. These modules operate identically to the modules used in the shared system.

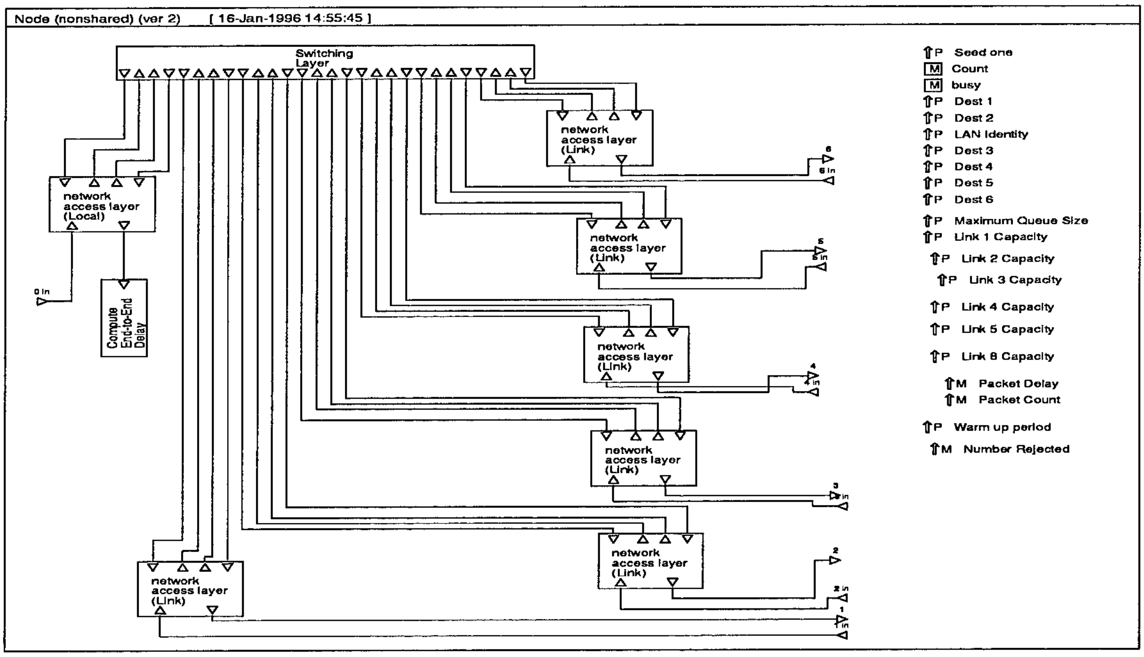

Figure 3.33 Node (nonshared) (ver 2) Module

3.5.7 Steady State Analysis

A steady state analysis is used to determine how much of a load the system can sustain without becoming overloaded, and to determine the warm up period in order to prevent initial bias. In this experiment, the maximum load will be designated as the peak load, and it will further be used to determine the amount of buffer allocation assigned to the queues.

After running a series of pilot tests in which 'Load' was varied from 0.1 to 3.0 (29,069 bps to 872,838 bps - see Table 3.3), a load equal to approximately 193,964 bps ('Load ' = 2/3) had been selected as a reasonable candidate. Since a similar set of tests have been done for the two node network shown in Figure 3.9, the results of these tests are not shown. An additional five runs (using different seed values each run) have been made at the candidate's load to gather data needed to perform the following tasks: 1) verify no overload occurs, 2) determine queue buffer allocation, and 3) establish warm up time.

In order to verify that no overload condition exist, the average end-to-end delay and bandwidth utilization (one of the links only) have been monitored. Average end-to-end delay is shown in Figure

3.34, and the bandwidth utilization is shown in Figure 3.35. Both appear to be stable (i.e., not steadily increasing as time increases).

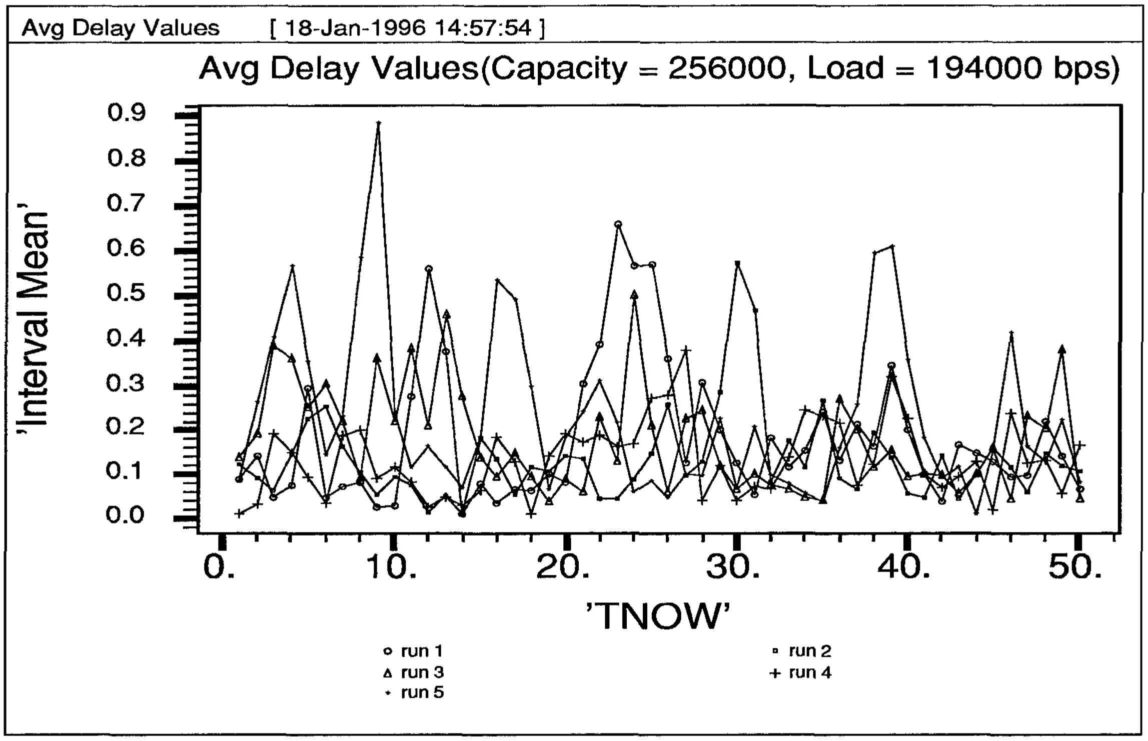

Figure 3.34 Average End-to-End Delay Plot

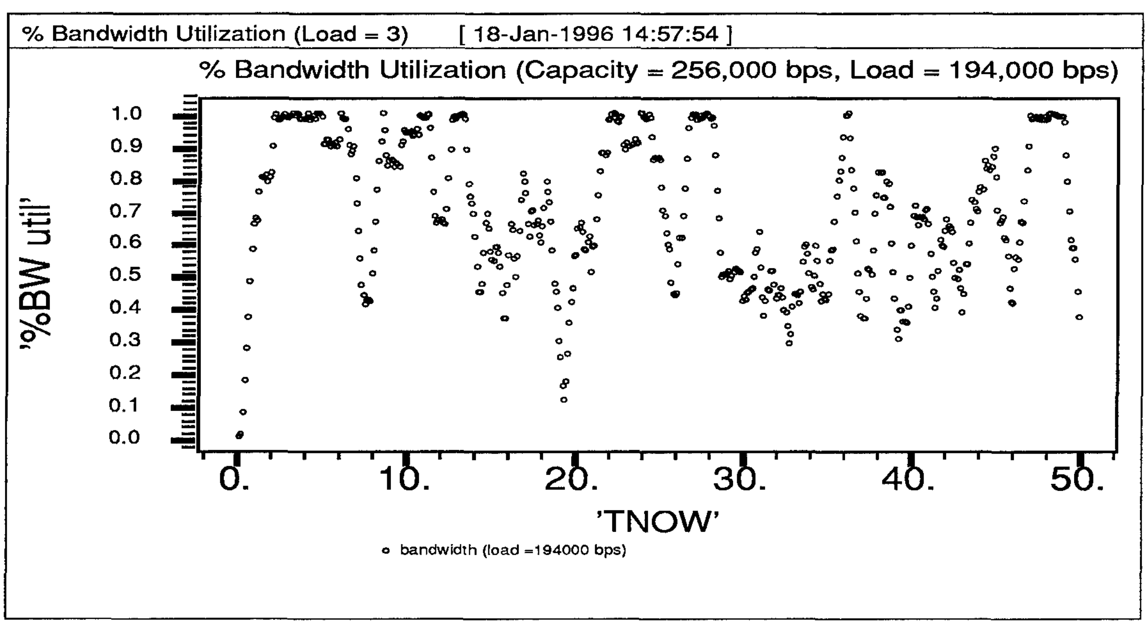

Figure 3.35 Bandwidth Utilization Plot

In determining queue buffer allocation, a transmission and a processing queue had been monitored during the runs. Figure 3.36 shows the transmission queue's average length (five runs) and the maximum length (three runs only) over a duration of 50 seconds. The processing queue had only a maximum of two entities during all five runs (not shown). As shown in Figure 3.36, the maximum number of packets observed in the transmission queue is 100, and therefore this number seems reasonable to use for buffer allocation. This will allow a peak traffic of approximately 193,964 bps with a minimal number of packets being rejected.

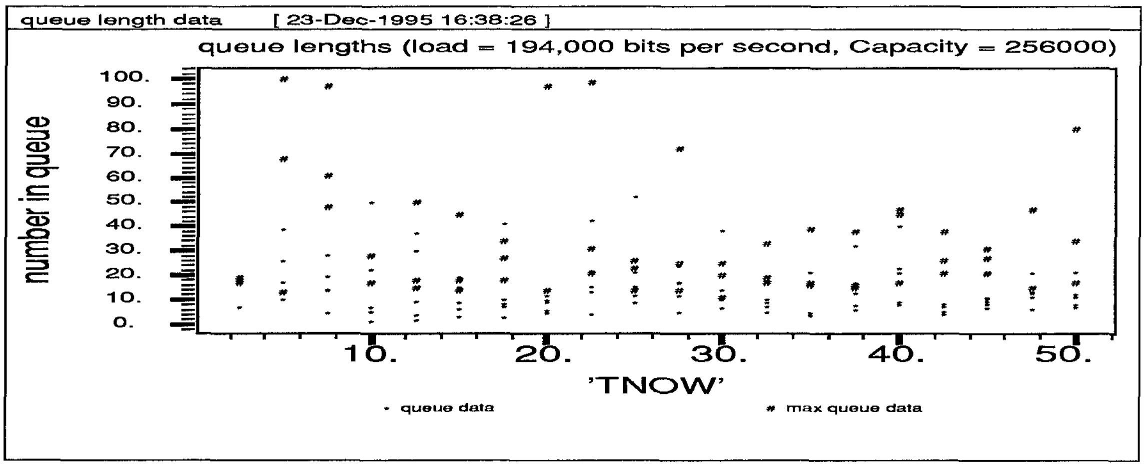

Figure 3.36 Queue Length Data

Welch's algorithm is applied to establish the warm up period. Upon observation of the results shown in Figure 3.37, it appears a warm up time of 3 seconds seems reasonable.

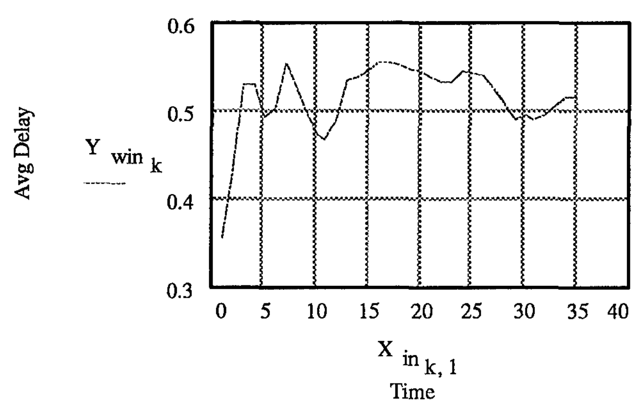

Figure 3.37 Results of Welch's Algorithm

3.5.8 Statistical Precision

From the above results, the peak load is chosen to be equal to approximately 193,964 bps ('Load'= 2/3). Further, the warm up time is chosen to equal three seconds. Law and Kelton [LaK91] recommend a simulation run time in the steady state to be significantly greater than the warm up time, and therefore, the simulation run time is set to thirty seconds. And finally, the number of runs in the following simulations will be a minimum of three [LaM94,ShD94], and each run will have unique seed values for each of the random variate generators.

3.5.9 CPU Time Required

The amount of time it took to run the simulations was proportional to the traffic load. Table 3.4 shows the average amount of time it took for 30 seconds of simulation time at each of the respective loads. The values provided are the average of three runs rounded off to the nearest 1/2 minute. The simulations were run on a Sun Sparcstation 20, using Designer software.

Table 3.4 Amount of time required for 30 seconds simulation time

Load	Load in BPS	avg. time nonshared (minutes:seconds)	avg. time shared (minutes:seconds)
.1	29,094	2:00	2:30
1	290,944	25:00	22:00
1.5	436,416	30:00	27:00
2.5	727,360	37:00	33:00

3.5.10 Parameters

Several different simulations are run to produce the required performance metrics. There are fixed input parameters, variable input parameters, and output metrics. Each of these parameters are now discussed.

3.5.10.1 Fixed Input Parameters

1. Number of sites = 5
2. Topology: see Figure 3.18
3. Distance between nodes = 300 miles
4. Node processing delay ~ N(100μs, 10μs^2)
5. Traffic Matrix: see Figure 3.20
6. Cost Matrix: see Figure 3.21
7. Queue buffer size: 100 buffers (see Figure 3.36)
8. LAN to Node mapping file (Node Vector File): shown in Figure 3.22.
9. Number of LANs = 7
10. Warm up period = 3 seconds
11. Bulk interpacket delay (during bursts) = .002 seconds.

3.5.10.2 Variable Input Parameters

1. Bandwidth = (shared, nonshared)
2. Link Capacity

Nonshared System: Each channel within the link is set to same amount with variable named 'Min Capacity.' Range: (64, 128, 256, 512, 1024, 1544) * 1000 bps.

Shared System: Each link capacity is set equal to sum of link channel capacities of nonshared system. In order to achieve equality in link capacities between the systems, a new variable has been created for each link (i.e., 'Link A to B Capacity').

3. Load = (.1, .125, .1667, .333, .667, 1, 1.25, 1.5, 1.75, 2.0, 2.5)

Table 3.5 calculates a rough approximation of the loads in terms of bits per second (BPS).

Bulk PPS = Load * ('Traffic Matrix' entry)*(mean number of pulses per burst).

= Load * 8 * 8.

Interactive PPS = Load*('Traffic Matrix' entry)*constant

= Load *8*8

Table 3.5 Load in terms of bits per second (bps)

Load	Bulk PPS	Bulk BPS (PPS*4096)	I/A PPS	I/A BPS (PPS*450)	Tot BPS (B.BPS+I/A.BPS)
.1	6.4	26,214	6.4	2,880	29,094
.125	8	32,768	8	3,600	36,368
.1667	10.667	43,692	10.667	4,800	48,492
.333	21.333	87,380	21.333	9,600	96,980
.667	42.667	174,764	42.667	19,200	193,964
1	64	262,144	64	28,800	290,944
1.25	80	327,680	80	36,000	363,380
1.5	96	393,216	96	43,200	436,416
1.75	112	458,752	112	50,400	509,152
2.0	128	524,288	128	57,600	581,888
2.5	160	655,360	160	72,000	727,360

3.6 Verification

Designer's interactive simulator was used extensively in verifying that correct paths were taken within the modules as well as throughout the system. In both the two node systems and the five node systems, the verification was accomplished by modular block testing using a bottom-up approach. Since the five node system was more complex, more extensive verification was performed on various activities, such as the round-robin algorithm and the computation involved in the creation of the routing tables. The routing tables and delay values had been verified by comparing their output to analytical values. Additionally, the traffic generators had been checked for correct traffic patterns and uniform destination distributions. Each of these are discussed below.

3.6.1 Round Robin Scheduling

This check pertains to the five node system only. The two node systems used a fair queuing system (all packets received went into a single queue and were served in FIFO order) and it was verified by simply monitoring the queue occupancy (refer to Figure 4.1). In order to check for correct operation of round-robin scheduling in the five node system, the algorithm was tested with the inbound queues empty, and again, with the inbound queues nonempty. The queues' empty condition was simulated by inputting a very light traffic load equal to one packet per second. This enabled the processor to completely process a packet before the next packet arrived. Upon completion of the processing of the packet, each queue was checked in the proper sequence, ending with a final check on the queue in which the last packet had been received. The round robin process did not activate again until the arrival of another inbound packet. The process then repeated itself. Meanwhile, the packets, after completion of processing were sent to the routing module in the shared system, or to the appropriate output queue in the nonshared system. In the queues' nonempty condition, the queue next in sequence was chosen. This step was accomplished by setting up breakpoints and following the sequence of events using the interactive simulator to ensure that the next queue in sequence was selected.

3.6.2 Routing

Routing tables were established only in the five node shared system. Because the two node shared system had only one interconnecting link, it was not necessary to create a routing table. Routing in the two node system could be accomplished using a four-way switch module, and the correct operation was verified by observing the packet's destination field, and ensuring the appropriate path was taken using the interactive simulator. In the five node shared system, a data file containing the costs of each link was loaded into the system and used along with Dijkstra's algorithm to compute the routing matrix. The values computed and inputted into the routing matrix by Designer, corresponded to the analytical values. Then, using Designer's interactive simulator, and setting up external displays at each of the node's input and output ports, the packets were traced to ensure they took the appropriate path. This was accomplished by simulating a very light traffic load of approximately one packet per second, and then following the packets as they traversed through the system. This was done for each possible source-destination pair.

3.6.3 Delays

In order to check for correct end-to-end delay values, a single packet was generated. By the use of breakpoints and external displays set up at the input and output ports of the nodes, the packet was monitored as it traveled from source to destination. By subtracting the packet's 'time created' field from the current simulation time, TNOW, the processing delays, transmission delays, and propagation delays each matched with hand calculated values, and the sum of delays equaled the corresponding value outputted by the 'Compute Statistics' module.

3.6.4 Traffic Distribution

The traffic generator output was monitored to ensure that a packet train pattern was generated and that the destinations of the packets were uniformly distributed. In checking for the correct traffic pattern, a probe was placed on the traffic generator output and filtered on a single destination. Figure 3.38 shows that the bulk packets generated do indeed represent a packet train (packets shown at the top of the diagram with 'size' equal to 4096 bits). The interactive packets generated appear to be exponentially distributed in size with the minimum size being 360 bits.

In order to check for a uniform destination distribution, an additional probe was placed on the traffic generater output of LAN 0. Figure 3.39 shows the number of packets that LAN 0 sends to each each of its destinations. Upon visual examination, there appears to be a near equal number of packets sent to LANs 2 through 6 from LAN 0, as expected.

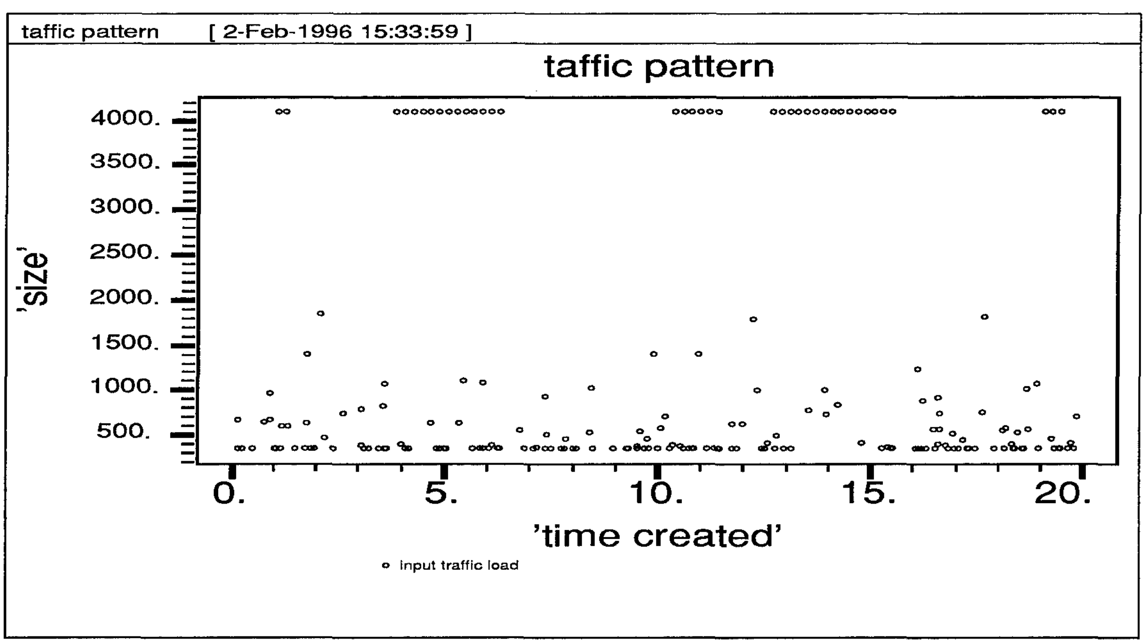

Figure 3.38 Traffic Generator Output (size is scaled in bits)

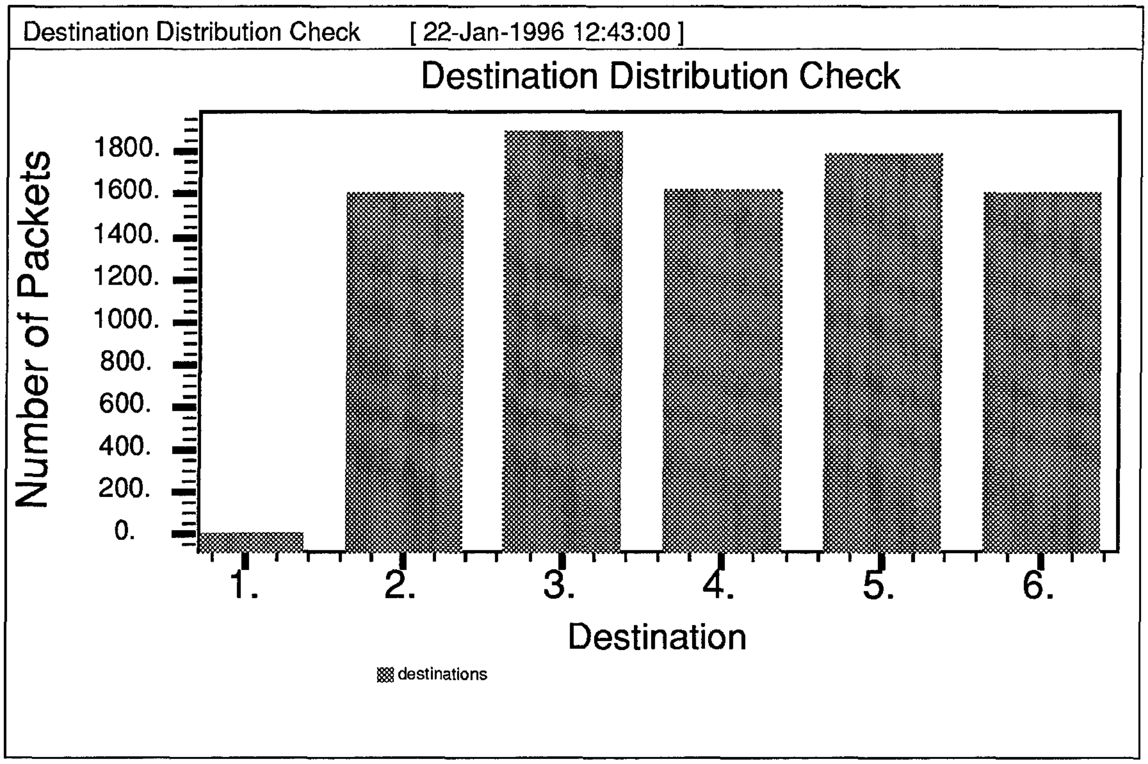

Figure 3.39 Destination Distribution for LAN A0

3.6.5 Designer Module Block Verification

The verification of the designer blocks was accomplished by testing modules at the lowest possible level and building upward. Encapsulation of verified lower level modules allowed for the testing of modules at the next highest level. This process was continued until the system level was reached.

The testing of block modules was accomplished using a combination of interactive simulation and probe modules supplied within Designer. The routing function (shared system only) and general packet flow through the system were accomplished by the use of a single packet with user-specified source and destination address. Textual probes and external displays were placed throughout the network to monitor the packet's progression through the network. The data collected by these probes and external displays were then analyzed to ensure that the packet was routed correctly. These probes and external displays also allowed for the verification of the delay incurred by the packet as it moved from source to destination.

3.7 Validation

The validation of the simulation models consisted of validating the operating assumptions, input parameter values and distributions, and the output values and conclusions associated with the models. Validity tests on these three model aspects can be accomplished by a combination of expert intuition, measurement from real systems, and/or comparison with theoretical results. For certain applications, all three comparative processes can apply. In other cases, only one may apply. In this scenario, comparison with measurements from real systems applies only in the sense of general system behavior. An example of a system's general behavior would be indicated by the queuing delays following a classic response relative to delay versus load characteristics. To further validate the models by comparing them to a known real system cannot be accomplished, since no known real system with a configuration exactly the same as the model's exists. Also the systems being modeled do not fit classic queuing models. This is because of the 'packet train' traffic model being used, opposed to the standard Poisson model, and to the use of the round-robin queue selection algorithm. As a consequence of the above factors mentioned, the determination of the model validity followed a step-wise approach for the operation assumptions, input parameters, and output results.

3.7.1 Validation of Operating Assumptions

The overall operating environment of the systems being modeled closely match those of systems found in previously published related works [Cha92,Jai90,San93,KaS95]. Most of these performance studies were based on flow control methods, routing comparisons, and buffer allocation schemes. But, because these studies took place on packet-switched networks, their operating assumptions could be readily carried into this research.

3.7.2 Validation of Input Parameters

The input traffic load for the system was established as a variable input parameter. The traffic load pattern (distribution) used was the 'packet train' model which is consistent with that used in previously published related works [JaR86,Zha90]. The topology chosen was a scaled down version of a seven node model also seen during the literature review[ChA92], and it represents how a network may have evolved over time. The other remaining input parameters are also consistent with current literature [Gru81,IIM85,Jai90,San93,KaS95]. The distance between the nodes was set to 500 miles in the two node case and is representative of a long distance terrestrial link. In the five node system, the distance was changed to be a variable input parameter, so that the distance of each link can be set individually. For the sake of simplifying the system, the distance was set to 300 miles for all of the links. The capacity of the links were also set up as variable parameters. In the five node case, the link capacities and distances, as well as other parameters were consistent in both the shared and the nonshared system. This consistency allowed for a fair comparison in system performance to be made. For the two node case, only the bandwidth parameter (named 'Link Capacity') differed. The two node nonshared system had twice the amount of bandwidth as the shared system. This was done to see how much the performance would degrade if the nodes were to communicate across just one link operating in the shared mode. In this way, a determination could be made as to whether the other link was needed. Simulation warm up time, run time, and the number of independent trials are consistent with Law, Kelton, and McComas's published works [LaK91,LaM94].

3.7.3 Validation of Output Results

The validation of the output results followed a similar approach used in the verification of the model. A bottom-up approach to validation was used for the system models. The output results of interest were the delay encountered by a packet as it traverses through the network, and link utilization rates. Each of these outputs are discussed below.

3.7.3.1 Validation of Packet Delay

Packet delay is defined as the difference in time between the arrival of the first bit of a given packet at the originating LAN and the receipt of the last bit of the packet by the destination LAN. The delay encountered by a packet as it travels through the network is a function of several factors. These factors include node processing delays, transmission delays, propagation delays, and queuing delays. To validate the packet delay portion of the model, it was necessary to control the environment of the system. A single packet was chosen to transmit from LAN A0 to LAN D0. By transmitting only one packet, the queuing delays at the nodes were eliminated. Therefore the packet delay was a function of the processing time, transmission time, and propagation time. Each delay factor was implemented using absolute delay module for processing time and transmission time, and a fixed delay module for propagation delay. The validation of these factors come from known physical and engineering laws (transmission and propagation delays). The processing delay was chosen to be consistent with recent literature published [ClJ89,Cha92,YaK93,Spo93].

The queuing delays at the nodes were validated by performing a sensitivity analysis. This was done by incrementally increasing the arrival rate of packets from a given source setting probes on inputs and outputs of the nodes and then analyzing the resultant delays. The effects of queuing at the nodes follow a classic response relative to delay versus loading characteristics. As the load was increased, the delays associated with queuing also increased. In the two node systems, this type of response continued indefinitely (infinite queue lengths were assumed). In the five node system, the delays began to decrease as the network became saturated. This occurred due to the fact that most of the packets generated during network saturation had been dropped, and the excessive delay times that would have occurred had they been kept in the queue were not taken into account in the calculation of average end-to-end delays.

3.7.3.2 Verification and Validation of Link Utilization

The verification and validation of the usage of the links followed an approach similar to that described above. Verification of link usage was accomplished by placing throughput versus time probes onto the link module's ports. The probe's output was a time-based average which divided the input traffic flow by the link capacity over one second time intervals. The resulting output follows a classic response relative to throughput versus loading characteristics. As the load was increased, the throughput increases, unless, of course, the network has become saturated. For an example of the output from the two node system, refer to Figure 3.13.

3.8 Summary

This chapter has presented a methodology which can be used to compare the performance (in terms of productivity and responsiveness) between using shared versus nonshared bandwidth in a packet-switched network. Two experiments have been conducted: a two node network experiment and a five node network experiment. In both experiments, a nonshared bandwidth configuration has been compared to a shared bandwidth configuration. The models created include LAN traffic generators to simulate the input load (the traffic generated can be interpreted as that coming from a 'stub' network). Traffic patterns generated have been based on a pulse train for bulk traffic and a constant plus exponential distribution for interactive traffic. The traffic intensity 'Load' and the 'Capacity' of the links have been varied to see how they affect the performance of the systems. All other input variables remain fixed. The network models have been designed in modular fashion using the Designer software package, and have been verified mainly by use of Designer's interactive simulator. The assumptions made and agreed upon by the sponsor have been checked for consistency in the model.

4 *Results*

4.1 Introduction

This chapter shows the results of the study examining shared bandwidth versus nonshared bandwidth systems. First, the results of the two node networks are presented, and then the results of the five node networks. In both of the configurations, the system traffic load and the link capacities are varied to see how they affect performance. The plots presented point out possible bottlenecks in the systems and compare the performance in terms of percent bandwidth utilization and mean end-to-end delay between the shared and nonshared systems. In the two node network, plots showing the average number of packets in the transmission queue are provided to give a measure of how serious the bottleneck points are in the two node system. In the five node system, a fixed queue length has been established, and the percent of packets dropped at the bottleneck points (transmission queues) are examined.

4.2 Two Node Network Analysis

4.2.1 Mean Queue Lengths

Three resources which are shared by all hosts communicating across a packet-switched network are the packet-switching node's processor, the packet switching node's buffer space, and the communication link. According to Yang and Reddy [YaR95], these three resources are potential bottlenecks that cause congestion in a network. In this scenario, infinite buffer space is assumed, and is therefore ruled out as a potential bottleneck. There are two objectives in this section. First, find out where the bottleneck is. Second, provide an estimate on the amount of buffer space required at each node. By observing the average number of packets awaiting to be processed by the node's processor, and comparing this to the average number of packets awaiting to be transmitted onto the link, an assessment can be made as to whether the communication link (capacity) and/or the node's processor is a bottleneck. Thus the first output metric which is examined is the average number of packets in each of these queues.

The results show that the lengths of the queues are affected by the traffic load and whether or not the system is using shared or nonshared bandwidth. Since the nodes are identical and all LANs are inducing the same load, the processing and transmission queues are monitored on node A only. In the shared configuration, a single T1 link (1.544 Mbps) is used, whereas in the nonshared system two T1 links (each 1.544 Mbps) are used. The load induced by each LAN is varied from 4 to 8 bursts per second (343 kbps, 516 kbps, and 688 kbps - see Table 3.1). The average number of packets versus time in the processing queue (shared system only) and transmission queues (shared and nonshared system) is shown in Figures 4.1 through 4.3. In order to get an idea of the average number of packets at certain time intervals (instead of a single overall average), batch means are used and are taken at one second intervals. Using batch means also provides some insight on the burstiness of the traffic.

4.2.1.1 Bottlenecks

Notice in Figure 4.1 that the processing queue basically remains empty. This is as expected since the packets are processed at a mean rate of 10,000 PPS (mean service time = 100×10^{-6}s), whereas they arrive at a rate of only 880 PPS. As a consequence, a packet will have completely finished processing before another packet arrival occurs. It is thus obvious that the node processor is not a bottleneck. On the other hand, the shared system's transmission queue appears to accumulate many packets. This occurs because the transmission service time is proportional to the size of the packet and inversely proportional to the communication link's capacity (Equation 3, Chapter 3, Section 3.2.2). For instance a packet of size 4096 bits would take 0.0027 (4096/1,544,000) seconds to process, whereas the next packet arrival may arrive in 0.0011 (1/880) seconds. Thus the newly arrived packet would have to wait 0.0016 (0.0027 - 0.0011) seconds in the queue before being processed. Further, the waiting times for packets are autocorrelated. This means that if the i'th packet had to wait, then the (i + 1)th packet would have a near equal waiting time in the queue. Due to the burstiness of the traffic, some packets may spend a great deal of time waiting in the transmission queue, while others may not. This waiting time has a significant impact on end-to-end delay. The main message that these graphs point out is that the bottleneck in the system is the communication link. Thus, one way to increase the performance (decrease end-to-end delay) is to increase the amount of bandwidth (capacity) of the communication link.

According to the data, no packets had to wait in the node's processor queue for both the shared system and nonshared system. Further, the transmission queue in the nonshared system had accumulated a maximum of 30 packets, but had an overall average of less than 5. This occurs because the nonshared system has twice the amount of link capacity as the shared system. Thus for the given load, the nonshared system's transmission queue did not present itself to be a serious bottleneck. It is expected however, that if the load intensity is doubled, then like in the shared system, the communication link (link capacity) will become a serious bottleneck. Although the node processing delay is negligible in this scenario, it can become significant when larger capacity links are used (i.e., SONET).

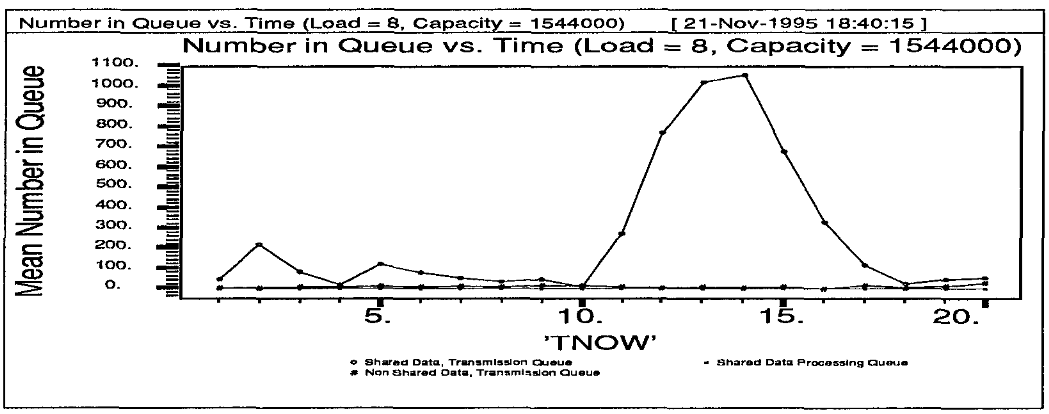

Figure 4.1 Average Queue Length (Load = 8, Capacity = 1.544 Mbps)

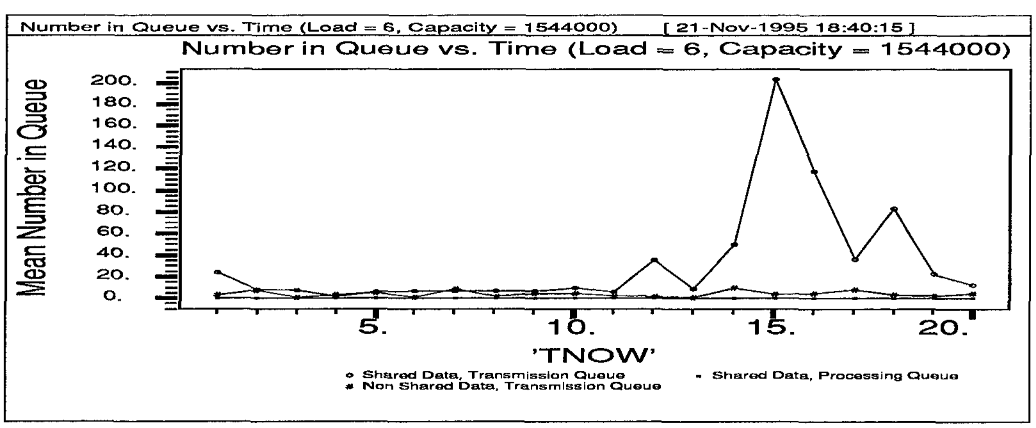

Figure 4.2 Average Queue Length (Load = 6, Capacity = 1.544 Mbps)

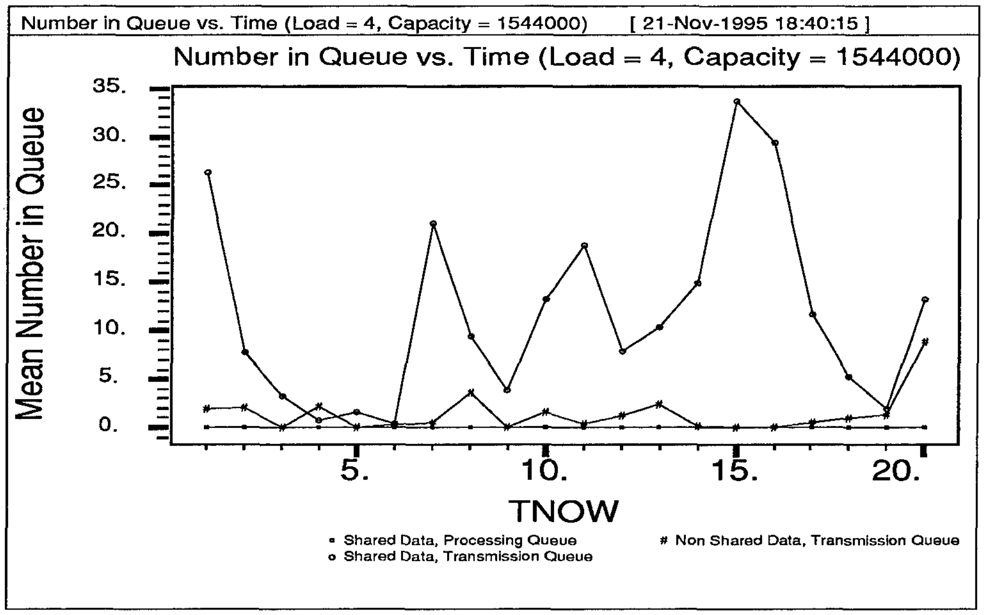

Figure 4.3 Mean Number in Queue (Load = 4)

4.2.1.2 Load Effect on Queue Length

As expected, when the load decreases from eight bursts per second (shown in Figure 4.1) down to four bursts per second (shown in Figure 4.3.), there is a corresponding decrease in the average number of packets in the queue. When the load is equal to eight, an average up to 1100 packets had accumulated in the queue, whereas, when the load is equal to four, only an average of up to 35 packets had accumulated. Additionally, as mentioned previously, notice, that in the nonshared system, the average transmission queue length is much less than the shared system's. As explained earlier, this is accounted for by the fact that in the shared configuration, the traffic of both of the node's LANs are transmitted across a single link (1.544 Mbps) instead of two separate links (each at 1.544 Mbps).

4.2.1.3 Burstiness Effect on Queue Length

The burstiness of the traffic can be seen with all three different loads applied (Figures 4.1, 4.2, 4.3), but in Figures 4.1 and 4.2, there appears to be a hump when the simulation time reaches approximately 15 seconds. In order to see why this occurs, the bulk traffic generated from the LANs at site A was monitored. Only bulk traffic was monitored because they are much larger in size (4096 bits per packet) than the interactive packets (mean of 450 bits per packet), and, therefore, more likely to have a greater impact on the size of the queues. The results are shown in Figure 4.4. Observe the increase in the number of bulk packets generated as the simulation time approaches 15 seconds.

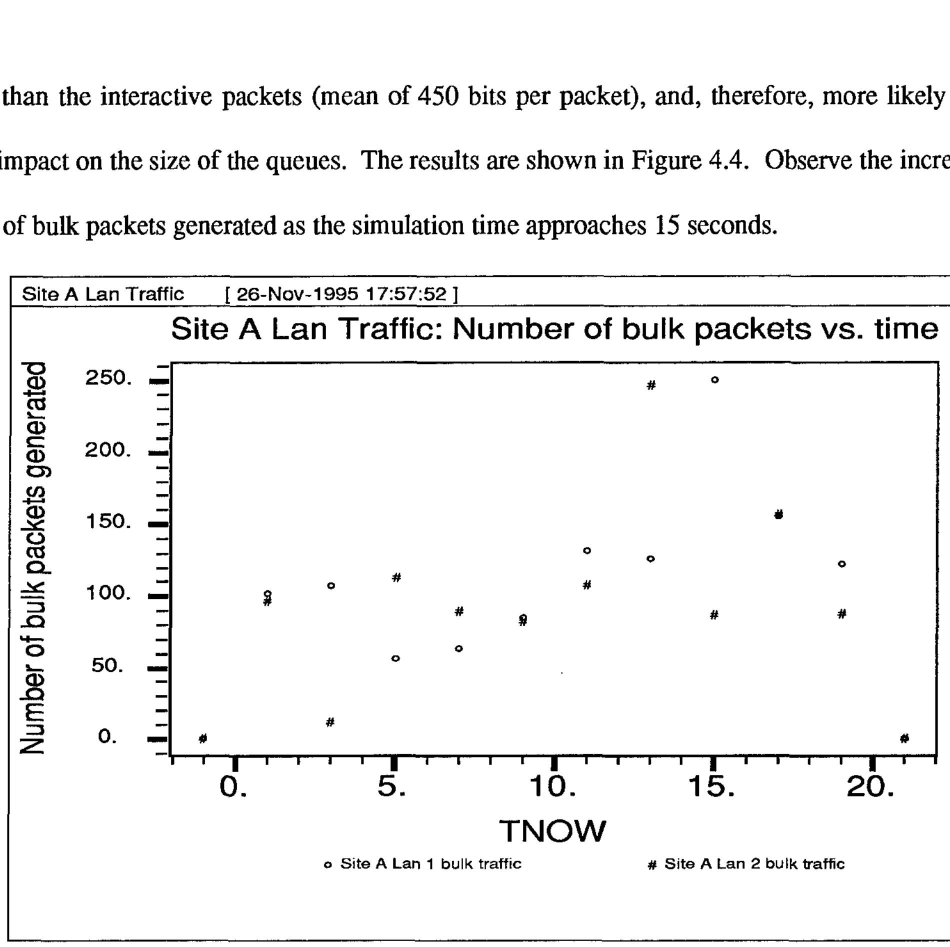

Figure 4.4 Number of Bulk Packets Generated vs. Time (Load =6)

4.2.1.4 Determination of Buffer Size Required

If the assumption of infinite buffer size is to be removed, the buffer size must be estimated. One way of determining buffer size requirements would be to input a peak traffic load (in this case 8), and calculate the mean number of packets in the queue over total simulation time (minus warm-up time). However, if the peak traffic load is extremely bursty, a better method may be to use batch means so that the queues can be monitored throughout the simulation. For instance, if the peak traffic expected is equal to eight (reference Figure 4.1), a maximum average of 1100 packets is indicated. Thus, a safe estimate would be to choose the queue size to be equal to 1100 buffers. Since a 'large buffer' buffer memory management scheme (see Section 3.5.3) was assumed, each buffer is set equal to 512 bytes (this would be the size of the largest packet expected). Thus, total memory required, in terms of bytes, would be 563,200 bytes (563,200 = 512 x 1100). However, 1100 packets is an average, thus, a few packets may still be lost.

Due to other memory requirements which must be satisfied in the node, such as memory needed for the node's processor queues and routing tables, it may not be possible to allocate 1100 buffers to the transmission queue. If for example, a smaller value of 300 buffers is chosen, then it would be expected that significant packet loss would occur 25% of the time (25 = [16-11]/20). Upon making the decision on how much buffer space is required, consideration must be given to the type of traffic supported. If, for instance, real-time traffic is to be supported, the following loss rates described by Aras and Kurose [ArK94] should be taken into account. Aras and Kurose [ArK94] pointed out that in their study of real-time traffic in packet-switched networks, that packet loss in short audio segments have been cited to be as high as 50%. They say that high quality audio can tolerate a loss of only 5% and music 10%. Further, video (dependent upon the coding scheme) can tolerate a loss of only 1%.

4.2.2 Capacity and Load Effect on Delay and Bandwidth Utilization

In this scenario, end-to-end delay and percent bandwidth utilization are plotted against link capacity. This is done for three traffic load intensities: 8, 6, and 4 (i.e., each LAN generates 688 kbps, 516 kbps, and 343 kbps). Both the shared and nonshared system are plotted in each graph, and a separate graph is used for each load intensity. How end-to-end delay is affected by the link capacity can identify minimum bandwidth (capacity) requirements between nodes. For instance, for voice packets, the maximum average delay allowed is 200 milliseconds [IIM85]. A study done by Braun and Chinoy in 1993 [BrC93] showed that average end-to-end delays across the Internet's National Science Foundation's NSFNET backbone did not exceed 100 ms. The study further showed that round trip times (RTT) of packets from California to Japan were between 600 ms and 1600 ms.

Assuming a peak traffic load of 8 (each LAN generates 688 kbps), Figure 4.5 shows that end-to-end delay increases as link capacity decreases. Notice, as the link capacity drops below 1.5 Mbps, the end-to-end delay begins to increase at a dramatic rate for the shared system. However, at around 1.6 Mbps, the difference between the shared and nonshared system is small. This results from the fact that the shared system becomes saturated when the link capacity drops below 1.544 Mbps. When saturation occurs, the packets accumulate in the transmission queue, and as a result a dramatic increase in queuing delay occurs. In the nonshared system, the total link capacity is twice that of the shared system, and, therefore, does not become saturated throughout the range of the link capacities.

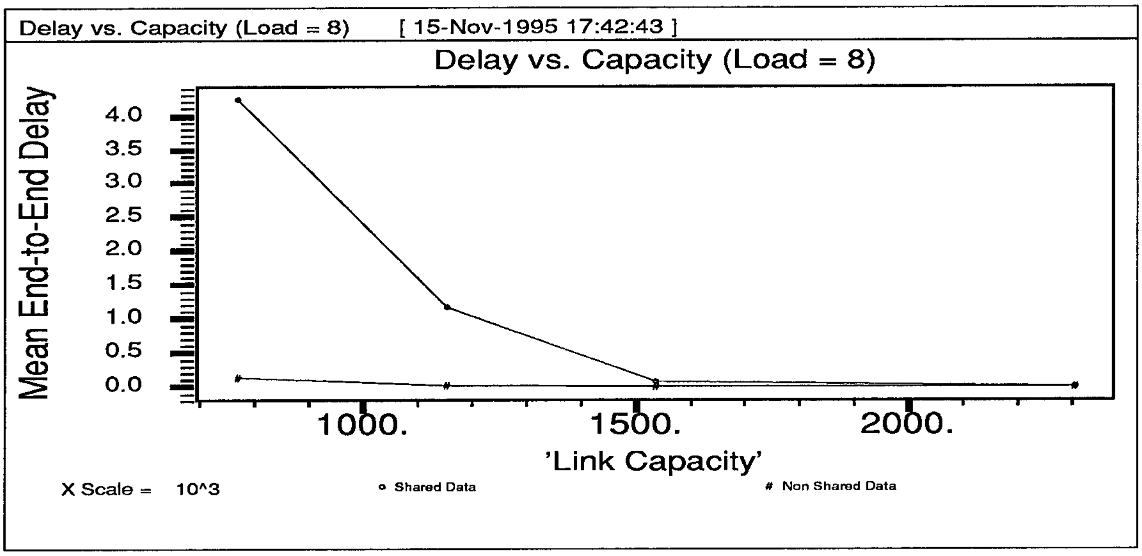

Figure 4.5 End-to-end Delay vs. Capacity (mean end-to-end delay scaled in seconds)

The percent bandwidth utilization (Figure 4.6) appears to increase as link capacity decreases. This gives a good measure of the throughput of the system. For instance, at a capacity equal to 1.544 Mbps, the percent bandwidth utilization of the shared system is approximately 80%, while the percent utilization of the nonshared system is only 40%. This implies that the link is idle approximately 20% of the time in the shared system and 60% of the time in the nonshared system.

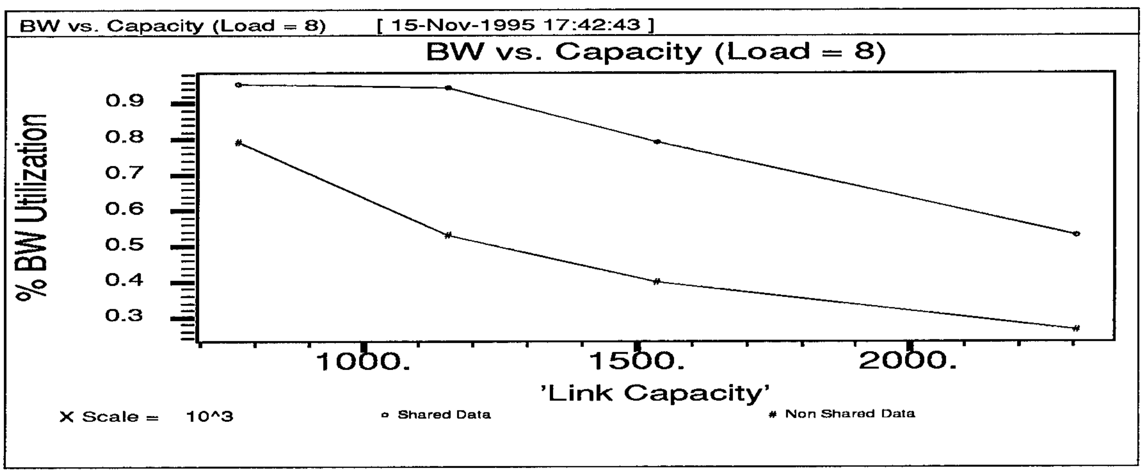

Figure 4.6 Percent BW Utilization (x 100) Versus Capacity

Figure 4.7 shows that as the traffic load intensity is reduced to six (each LAN generates 516 kbps), there is a corresponding decrease in the average end-to-end delay. For example, Figure 4.7 shows that for link capacities greater than 1152 kbps, delay is minimal. Figure 4.8 zooms in on the capacity range between 1,152 kbps and 1,544 kbps and shows the average delay to be less than 100 ms. Having a specified upper bound on average end-to-end delay, such as 200 ms will enable real-time packetized voice communication. Thus, if peak traffic load is six, then a link capacity equal to or above 1152 kbps should be sufficient. Figure 4.9 shows the shared system to be more productive than the nonshared system. Specifically, there is between a 20 to 25% increase in percent bandwidth utilization in the shared system over the nonshared system.

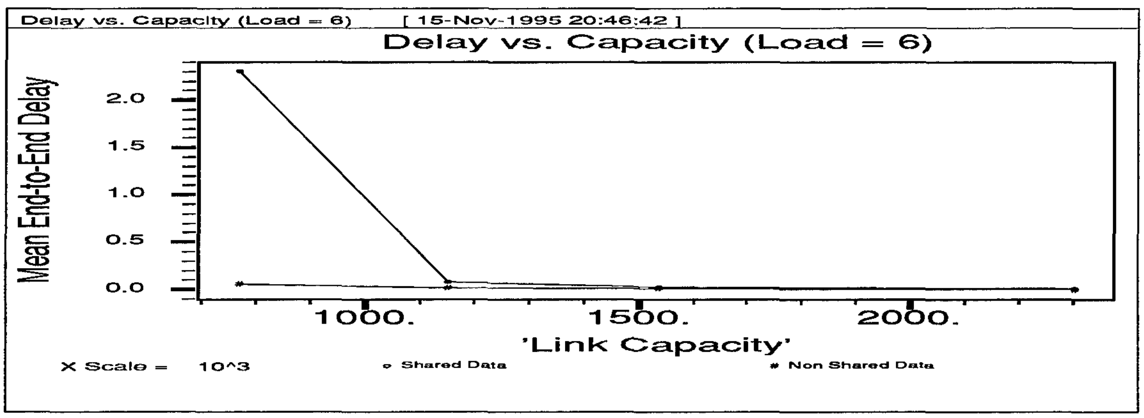

Figure 4.7 End-to-end Delay vs. Capacity (mean end-to-end delay scaled in seconds)

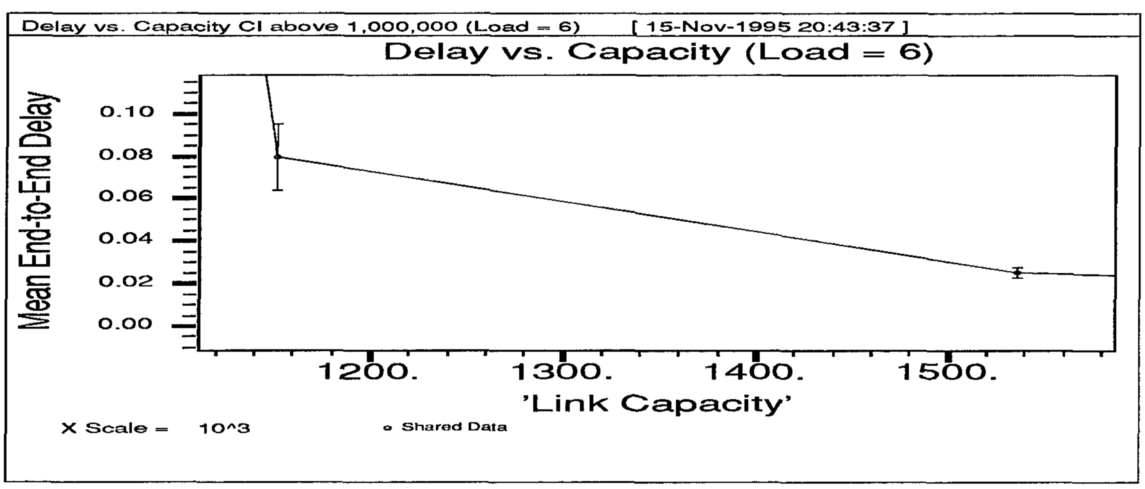

Figure 4.8 End-to-End Delay vs. Capacity - With Zoom (95% Confidence Intervals, Shared Bandwidth)

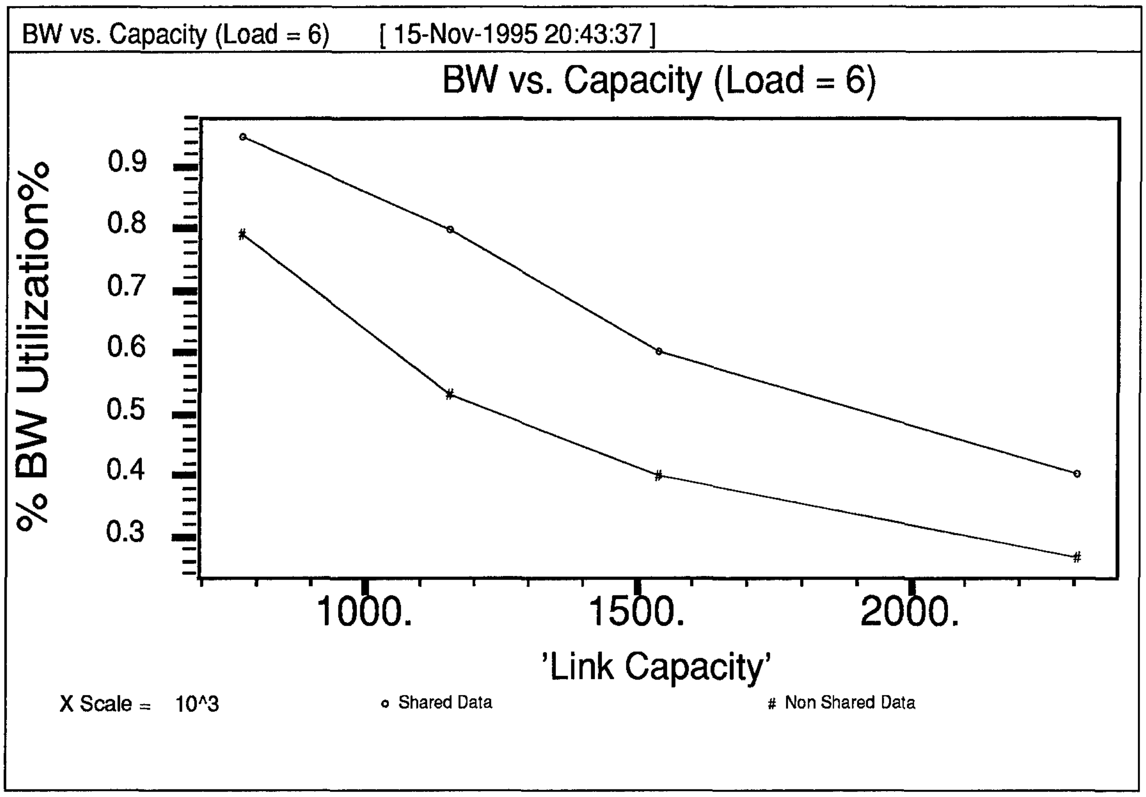

Figure 4.9 Percent BW (x 100) Versus Capacity (Load = 6)

As the traffic load intensity is reduced even further to four (each LAN generates 343 kbps), a similar result takes place in that the average end-to-end delay and percent bandwidth utilization decrease by a proportional amount. Figure 4.10 shows the average end-to-end delay in both the shared and nonshared systems can satisfy real-time voice requirements at a link capacity value above 768 kbps. However, the results indicate that the average end-to-end delay is less in the nonshared system. This occurs primarily because the nonshared system has twice the capacity of the shared system. This does not really allow for a fair comparison between end-to-end delay between the shared and the nonshared system. In the five node systems, the link capacities will be set equal to each other so that a more fair comparison can be made. It must be acknowledged at this point though that the nonshared system is more responsive than the shared system. On the other hand, in regards to productivity, Figure 4.11 shows that the percent bandwidth utilization of the shared system is approximately 10 to 15% larger than the nonshared system throughout the range of the link capacity.

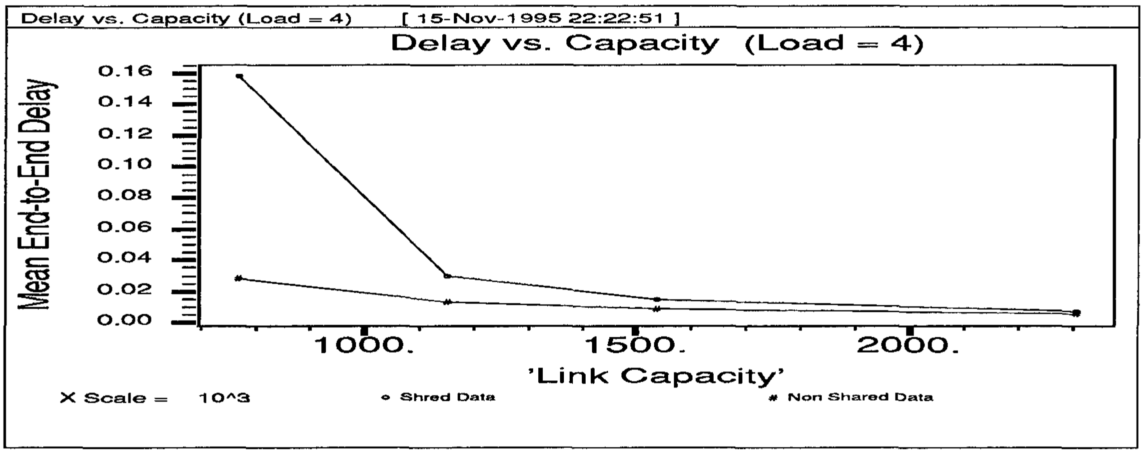

Figure 4.10 End-to-end Delay vs. Capacity (mean end-to-end delay scaled in seconds)

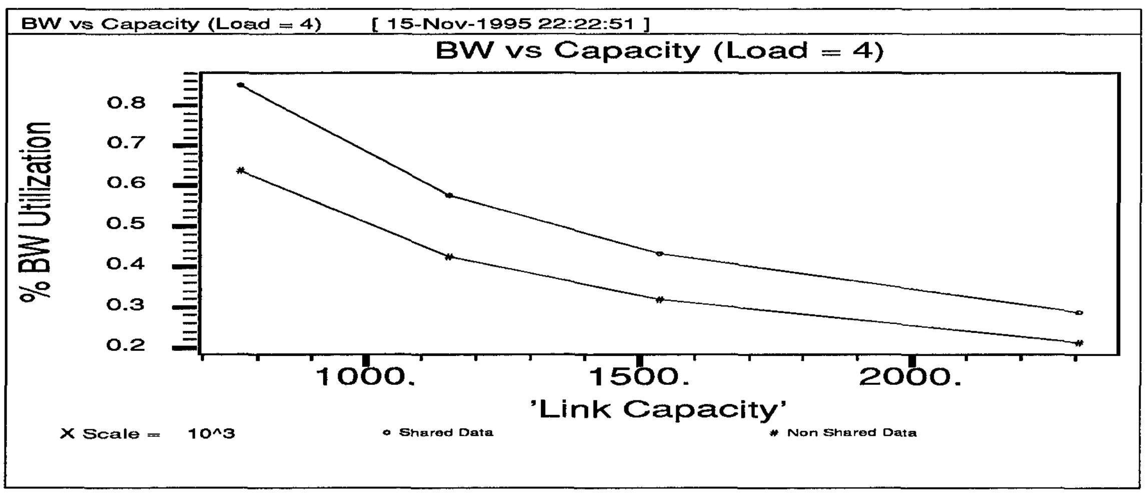

Figure 4.11 Percent BW (x 100) Vs Capacity (Load = 4)

The above graphs provide a good measure on how much bandwidth is required in order to achieve a specific amount of responsiveness and productivity. It has been shown that the amount of link capacity needed is dependent upon the traffic load intensity. As the traffic load intensity increases, there is a corresponding increase in the amount of capacity needed to achieve a specified upper bound on average end-to-end delay. For instance, if the peak traffic is equal to six (each LAN generates 516 kbps), and the upper bound on average end-to-end delay is 100 ms, then a link capacity equal to or above 1152 kbps should be chosen. However, if the peak traffic load is increased to eight (each LAN generates 688 kbps), then the link capacity chosen should be equal to or be above 1,544.

Having a specified upper bound on end-to-end delay allows the realization of transmitting real-time traffic. For instance, as mentioned earlier, packetized voice traffic is possible if average delay does not exceed 200 ms. Other time-sensitive applications, such as TELNET and RLOGIN can be improved so that they are more responsive to the user. According to Floyd [Flo94], a round trip delay greater than 100 ms is likely to be noticeable to a TELNET user.

Overall, the results showed that the shared system was more productive than the nonshared system. For example, Figure 4.8 showed that approximately 60 % of the bandwidth would be utilized in the shared case, whereas in the nonshared case only 40% of the bandwidth would be utilized. In terms of responsiveness, the nonshared system had less average end-to-end delay. But it must be remembered, that in this scenario, the objective was to find out whether a single shared T1 link could be used in place of two nonshared T1 links. The results showed that a single shared T1 link could be used, provided the peak traffic load did not exceed eight (each LAN generated approximately 688 kbps at this load), and assuming a specified upper bound of 200 ms. Thus, by operating in the shared mode, the cost of an additional T1 link could be saved, provided the above conditions are met.

4.3 Five Node Network Analysis

The results in this experiment are based on the following three performance measures: 1) percent bandwidth utilization, 2) percent of packets dropped due to buffer overflow, and 3) average end-to-end delay. Each plot will show the shared versus the nonshared system. First, these performance measures are checked to see how they are affected by varying the input traffic load (packet arrival rate). Second, a similar performance analysis is undertaken to compare the systems as the capacity of the links varies. In each section, the first metric examined is the percentage of packets dropped. Next, the average end-to-end delay and percent bandwidth utilization parameters are compared. After examining how the shared system performs against the nonshared system under various loads and capacities, a similar performance analysis is undertaken to compare these systems under nonuniform traffic loading. In this scenario, nonuniform traffic loading occurs when a LAN's mean transmission rate is dependent upon the destination, whereas uniform loading occurs when the LAN's mean transmission rate is the same for all destinations.

4.3.1 Comparing a Shared Versus Nonshared System as the Load Varies

The percentage of packets lost is a good indicator of system performance. Each node's transmission queue size is fixed to hold up to 100 packets, and if packets continue to arrive when the queue is filled, they are dropped. When packets are dropped in an actual TCP/IP network, flow control mechanisms will stifle the source's transmission rate. As a consequence, end-to-end delay increases and throughput decreases. Since flow control mechanisms are not explicitly modeled in the system, a valid range for comparing the performance between the systems takes place when less than one-half of one percent (0.5%) of the packets generated are lost. This loss rate was chosen because it would probably not affect the traffic flow significantly in a real TCP/IP network, since only a small percentage of the hosts would reduce their transmission rate. Thus, when less than approximately 0.5% packets are dropped, the traffic flow in the model should closely resemble the traffic flow in a real system. Further, if real-time traffic is a future possibility, Aras and Kurose's study [ArK94] on real-time traffic stated that high quality audio can tolerate a loss of up to 5% and music 10%. While, video (dependent upon the coding scheme) can tolerate a loss of only 1%.

Figure 4.12 shows that when the traffic load intensity increases above 2/3 (approximately 194 kbps), the percentage of lost packets increases as the load increases. When the load equals 194 kbps, the number of lost packets is most likely due to the burstiness of the traffic. Whereas when the load exceeds 1.0 (approximately 291 kbps), the network has become saturated, and the percentage of dropped packets increases at a dramatic rate. This result occurs because no flow control mechanisms have been explicitly modeled into the system. Under real operating conditions, the hosts on the LANs will have implemented flow control mechanisms, resulting in retransmissions and reduced transmission rates. As a consequence, a decrease in overall throughput would occur. Since flow control mechanisms are not implemented, the comparison in performance between the nonshared and shared system is not really valid when the system becomes saturated (i.e., the load increases above 256,000 bps). Another noteworthy observation about

Figure 4.12, is that the percentage of dropped packets in the nonshared system is approximately equal to the percentage of dropped packets in the shared system. In fact, the number of packets dropped equal zero in both systems when the load intensity is less than 1/3 (97 kbps) and both systems drop less than 0.3% when the load intensity is equal to 2/3 (194 kbps). This is as expected since neither system is saturated.

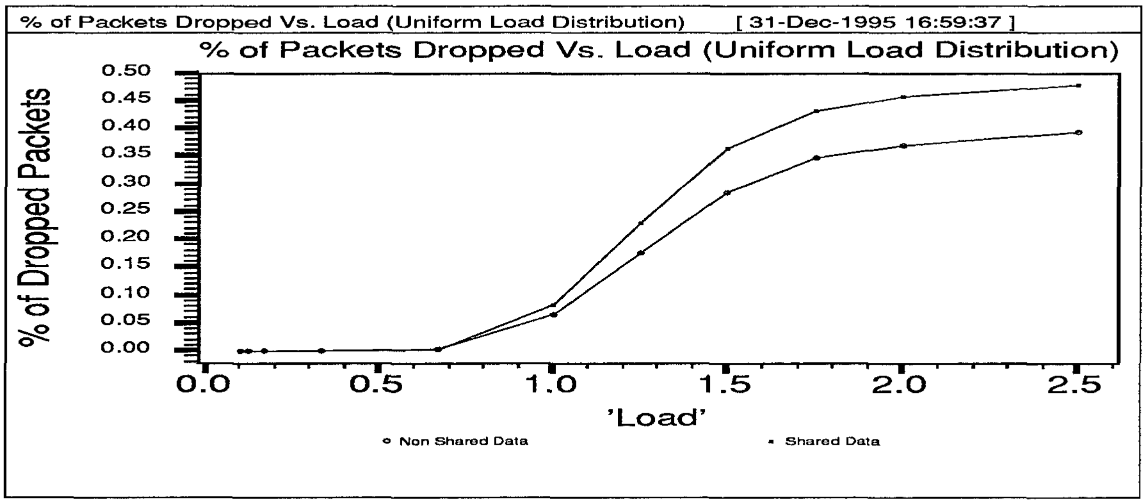

Figure 4.12 % of Dropped Packets(x 100) Versus Load

Figure 4.13 shows the bandwidth utilization versus load. As expected, the bandwidth utilization increases as the load (packet arrival rate) increases. In regards to productivity, it is desirable to have a high bandwidth utilization (i.e., maximize resource utilization). Originally all six links of both the shared system and nonshared system were plotted. But it was noticed, and can be seen in Figure 4.13, that the links had approximately the same amount of bandwidth utilization. Examination of the data at each of the load values revealed less than a 4% difference between the two systems and less than 5% difference between all of the links. Thus for clarity, only links A to B and C to E of both the nonshared and shared system are shown. This near equal bandwidth utilization occurs because the traffic load is distributed equally, and because the total link capacity of the nonshared system equals the total link capacity of the shared system. Another noticeable feature occurs when the load increases above 1.0 (approximately 291 kbps). As expected, the systems become saturated. When saturation occurs, the transmission queues build up, and there is always an accumulation of packets awaiting to be transmitted. This agrees with the results obtained earlier (Section 4.2.1.1) in that the communication link is indeed a bottleneck point.

Although, in this case it is a bottleneck point in both the shared and nonshared systems. As previously discussed, since no flow control mechanisms have been implemented into the model, the readings above the load of 1.0 are unreliable. Thus, so far, there does not appear to be much of difference between the shared and the nonshared system. The average end-to-end delay is discussed next.

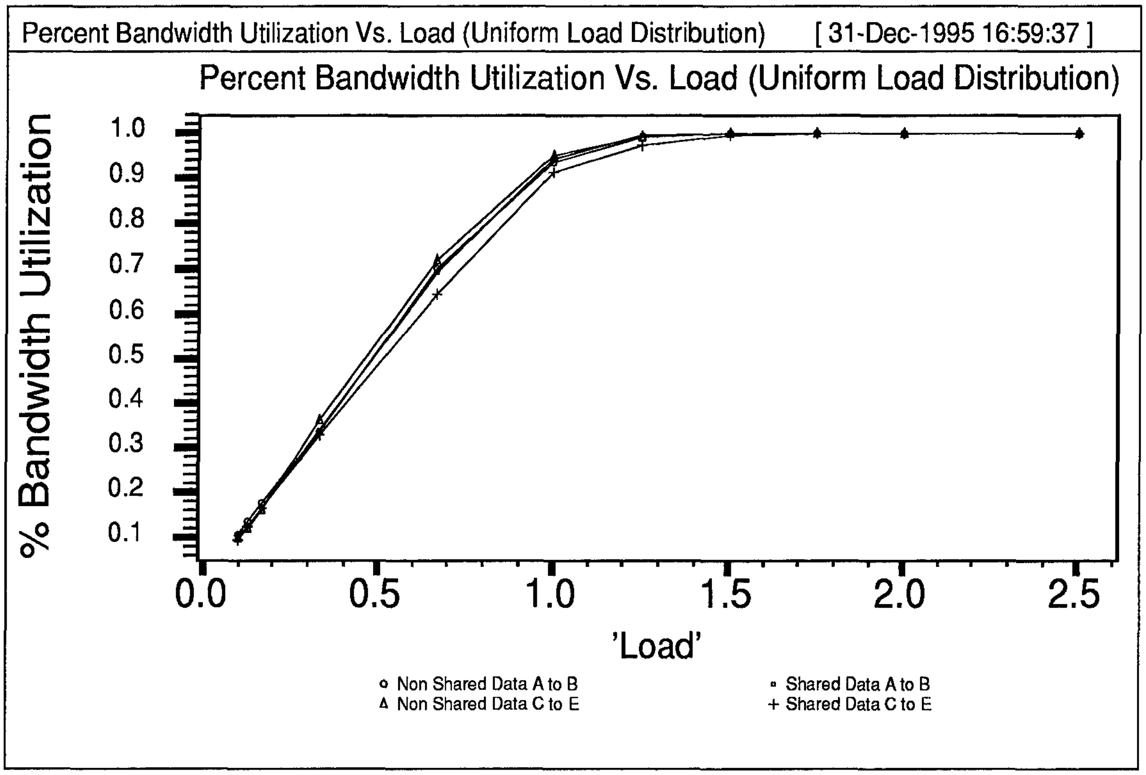

Figure 4.13 % Bandwidth Utilization (x 100) Versus Load

Figure 4.14 shows the mean end-to-end delay versus load. Notice as the load increases, so does the delay in both the shared and the nonshared system. The most striking result is that the average end-to-end delay is considerably less in the shared system than in the nonshared system. For instance at a traffic load equal to .1666 (48 kbps), the difference in mean end-to-end delay between the nonshared system and the shared system is approximately 50 ms. Dividing the difference of 50 ms by the nonshared mean end-to-end delay value, 63 ms, results in a 79.4% improvement. At an increased load equal to 2/3 (194 kbps), the difference between the systems is approximately 180 ms. Again, by dividing the difference by the nonshared system's mean end-to-end delay value results in a 78.6% improvement.

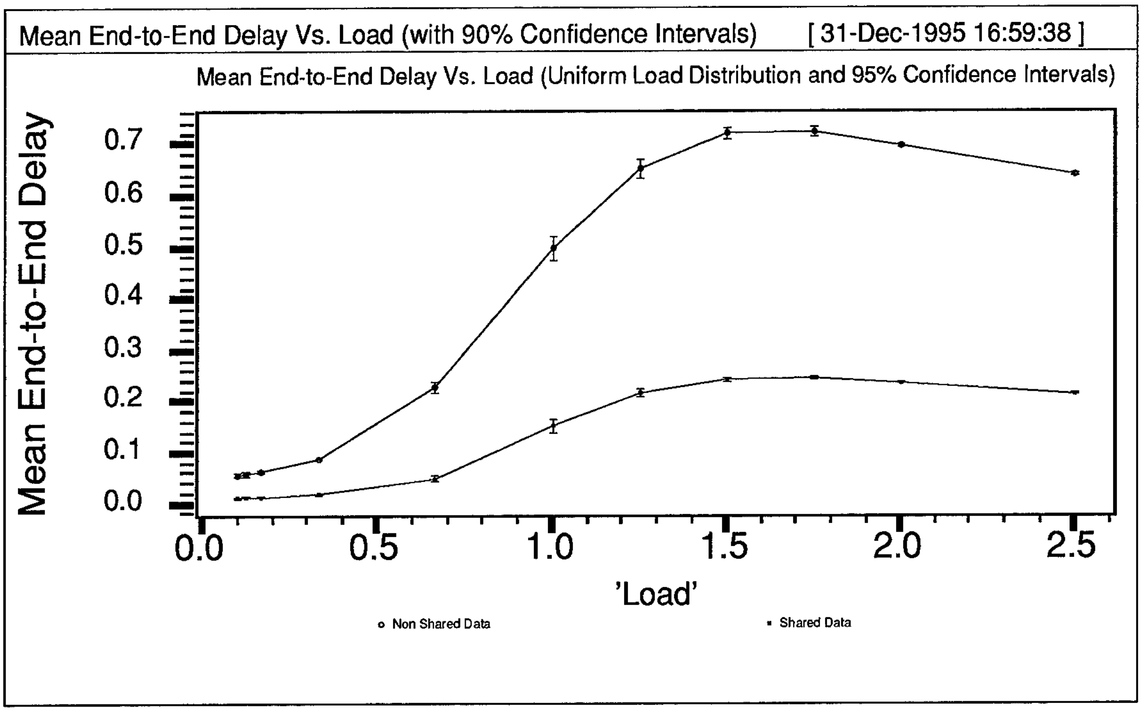

Figure 4.14 Mean End-to-End Delay Versus Load (Delay measured in seconds)

In order to explain why the average end-to-end delay is so much less in the shared system than in the nonshared system, it must be remembered that in order to compare the nonshared to the shared system, the total capacity in each system's links were set to equal to each other. For instance, the Site A to Site B link in the nonshared system had six channels each with 256 kbps capacity for a total of 1,536 kbps link capacity. Thus, the shared system's A to B link was set up with a single channel having 1,536 kbps capacity. Therefore, the transmission delay, according to Equation 3 (Section 3.2.2), will be much smaller (six times smaller for the A to B link) for the shared system than for each of the nonshared system's individual channels. For illustrative purposes, the transmission delay (the time it takes the node to place a packet onto the transmission media) calculations for the Site A to Site B link are given below.

Nonshared System (6 channels):

$$TransmissionDelay = \frac{sizeofpacket}{256{,}000} \tag{6}$$

Shared System (1 channel):

$$TransmissionDelay = \frac{sizeofpacket}{1{,}536{,}000} \tag{7}$$

As can be seen from the above equations, packets will be processed (sent on to the transmission media) six times faster in the shared system. Further, since the traffic intensities for both systems are the same, it is expected that queuing delays will be about the same for both systems. For example, on the Site A to Site B link with an input traffic load of .666 (approximately 194 kbps per LAN to LAN connection), the traffic intensity for the nonshared system is approximately 194 kbps / 256 kbps, whereas in the shared system it will be (6 x 194 kbps) / (6 x 256 kbps), which of course equals 194 kbps/ 256 kbps.

When the input traffic load exceeds the capacity of the channels, system saturation occurs. When this happens, an increased number of packets will be dropped (see Figure 4.12). Figure 4.14 shows that end-to-end delay decreases when the load exceeds 1.5 (436 kbps). This occurs because a large number (over 25%) of packets generated are dropped and are not taken into account in calculating average end-to-end delay. As explained earlier, in an actual TCP/IP network, flow control mechanisms would force the senders to slow down their transmission rates prior to reaching this point. However, if flow control mechanisms were explicitly modeled into both systems, it is expected that the shared system would still have a significant less average end-to-end delay over the nonshared system in these load ranges as well.

4.3.2 Comparing a Shared Versus Nonshared System with Varied Link Capacities

As was done in the previous section, the same three performance metrics will be used: 1) percent of packets dropped, 2) percent bandwidth utilization, and 3) mean end-to-end delay. The load is fixed at 2/3 (approximately 194, kbps) and the amount of traffic generated by each possible source-destination LAN pair is equally distributed. Furthermore, as stated earlier, the transmission queue's buffer size is fixed at 100 packets maximum. Figure 4.15 shows the percentage of dropped packets versus the capacity. Notice as the capacity drops below 256 kbps, the percentage of packets rejected due to queue overflow increases at a dramatic rate. This occurs because the network becomes saturated whenever the capacity is of the link is less than the incoming traffic flow, which in this case is 194 kbps. Further, notice that the number of packets dropped is approximately equal in both the shared and the nonshared system. In fact, when the capacity values are greater than 256 kbps, zero packets are dropped in both systems. This is as expected because both systems operate in a steady state mode when link capacity exceeds the traffic flow.

At a capacity value equal to 256 kbps, there is approximately 0.3 percent of the packets dropped, and this is most likely due to the burstiness of the traffic load.

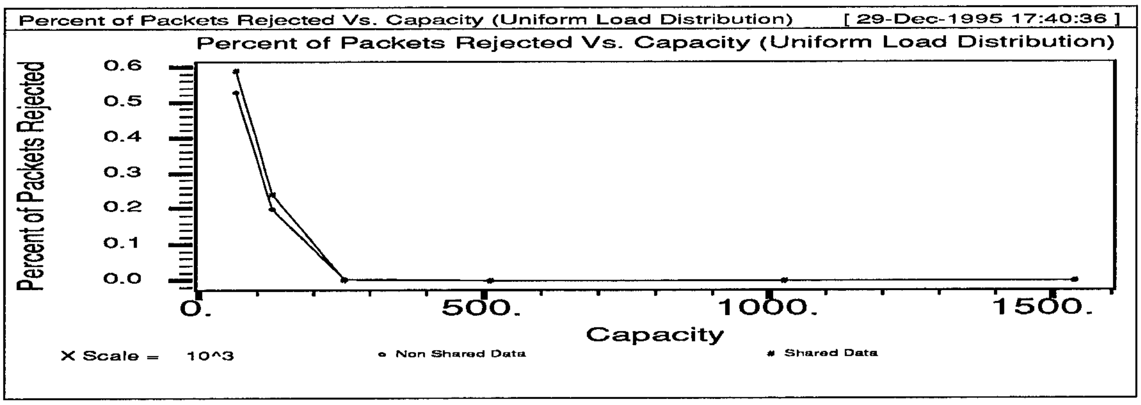

Figure 4.15 Percentage of Packets Dropped (x 100) Versus Capacity

Figure 4.16 shows the percent bandwidth utilization versus capacity. As was done in the previous case, only two links of each system have been plotted. Notice that there is very little difference between bandwidth utilization. Examination of the data at each capacity value revealed that less than a 4.4% difference exists between the two systems and less than 5.5% difference exists between all of the links. These results were obtained by dividing the difference by the value of the higher percent bandwidth utilization value. Further, notice that as the capacity decreases below 256 kbps, that both systems become saturated, as expected. As in the previous case, this occurs because the incoming traffic rate exceeds the capacity available. Thus far, there appears to be no difference between the shared and the nonshared system's performance. This is about to change.

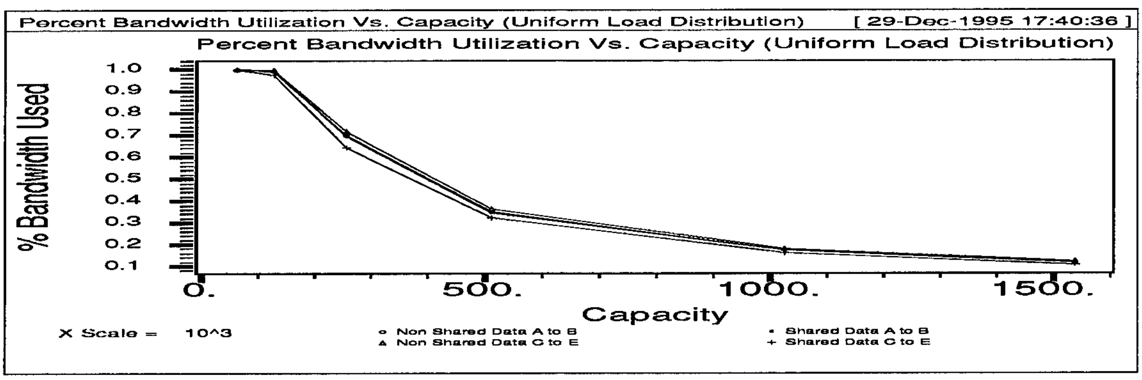

Figure 4.16 Percent Bandwidth Utilization (x 100) Versus Capacity

As just previously mentioned, not much difference in performance between the nonshared and the shared system has occurred in this section yet. Figure 4.17 indicates that the mean end-to-end delay is remarkably better in the shared system than in the nonshared system. This can be explained by the same reasoning used in the previous section. Due to the shared system's increased channel capacity, the packets are transmitted much more quickly (six times faster on the A to B link) resulting in the better overall average end-to-end delay. Notice, especially, in the range from 256 kbps to 1024 kbps where the difference ranges from 200 ms down to 10 ms. Analysis of the data revealed a 79.2% improvement at 256 kbps and a 73.2% improvement at 1024 kbps. The mean end-to-end delay values at a capacity below 256 kbps are not shown since the system was saturated, and as such (no flow control mechanisms), the mean end-to-end values in that range are somewhat meaningless.

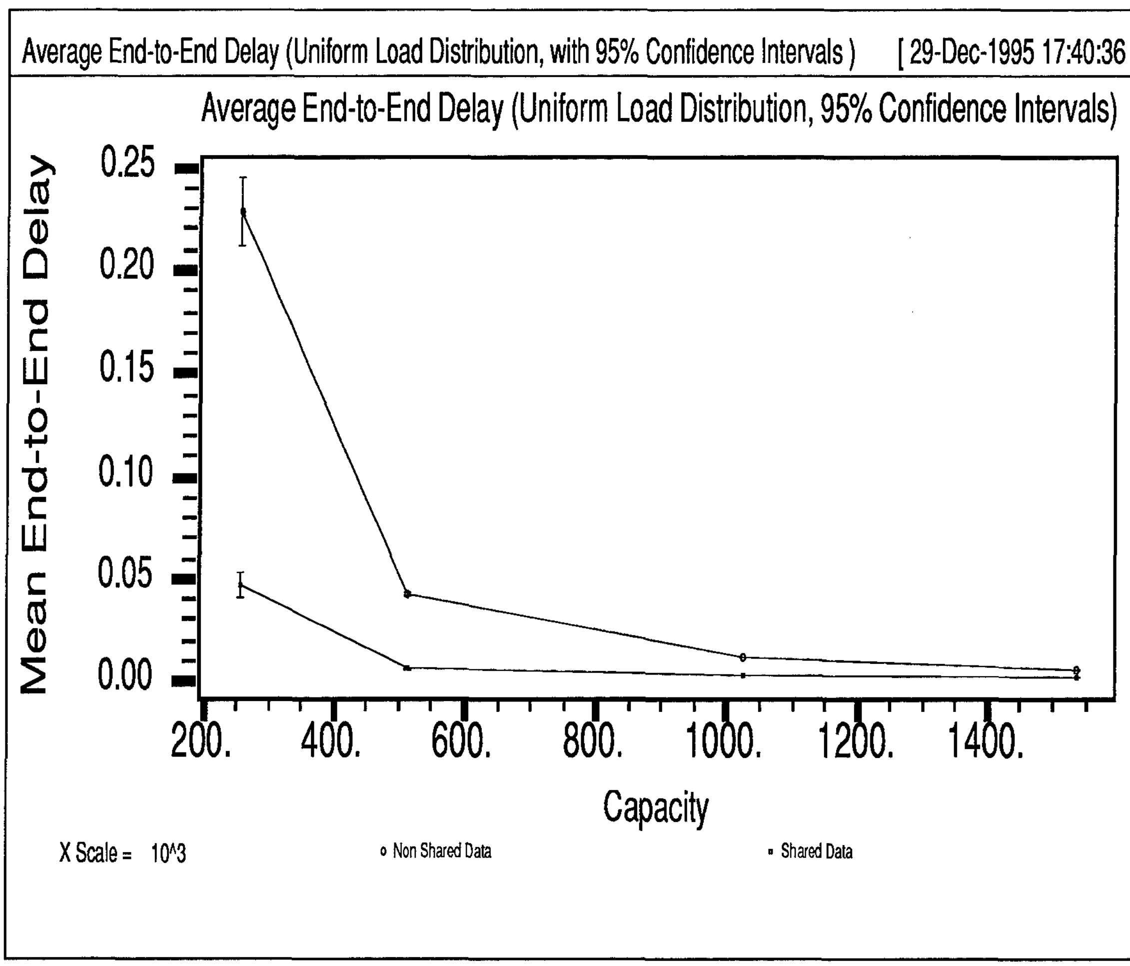

Figure 4.17 Mean End-to-End Delay Versus Capacity (Delay scaled in seconds)

4.3.3 Shared Versus Nonshared Using a Nonuniform Destination Distribution

A question that may arise at this point is what happens to the performance metrics if the load is not equally distributed. This is a legitimate question. Up to this point, each LAN has generated a traffic load to a given destination using the same load distributions (packet train and exponential plus constant) with the same mean interarrival times. These traffic loads were distributed to each destination equally by use of the traffic matrix shown in Figure 3.20. In order to vary the source to destination's traffic load rate, (i.e., using the same load distributions but with mean packet interarrival times dependent upon the destination), the traffic matrix was modified to that shown in Figure 4.18

0	0	4	4	8	8	4
0	0	4	4	8	8	4
4	4	0	4	8	8	4
4	4	4	0	8	8	4
8	8	8	8	0	0	8
8	8	8	8	0	0	8
4	4	4	4	8	8	0

Figure 4.18 Traffic Matrix used in Unequal Load Distribution

The traffic matrix values determine the amount of traffic each LAN sends to a given destination and were chosen so that the following scenario occurs. The scenario created is one in which Site D acts as a main processing center. As such, each of the sites communicate heavily (approximately 194 kbps) with Site D, and the other LAN to LAN communications have been set to one-half the rate and is assumed to consist mostly of email and one-half the amount of interactive traffic. It is expected that if alternative values had been chosen, or a different scenario for that matter, similar results would be achieved. This is true of course, provided there are no LAN to LAN traffic loads which exceed a rate of 256 kbps, causing a portion of the system to be saturated.

Table 4.1 provides a summary comparing the previous mean interarrival times to the new mean interarrival times and clarifies how the load is distributed throughout both the nonshared and shared systems. The variable 'Load' is an input parameter which allows the traffic intensity to increase, yet keep the proportion of the traffic generated by each source-destination pair in tact. The entries in the table have been calculated as follows: 1) bulk Traffic - mean interarrival time between bursts (each burst triggers a train of packets generated by a geometric distribution with a mean equal to 8) = 1/(Traffic Matrix Entry x Load); 2) interactive traffic - mean interarrival time between packets = 1/(Traffic Matrix Entry x Load x 8). As a reminder, the interactive traffic load was multiplied by eight to get an approximately equal number of interactive packets as there are bulk packets.

Table 4.1 Mean Interarrival times between source-destination pairs - equal load distribution versus unequal load distribution (Site D is the main processing center).

	Equal Loads ('Load' = 2/3)		Site D is main processing center ('Load' = 2/3)	
Source-Dest pair	bulk traffic mean burst interarrival time (seconds)	I/A traffic mean packet interarrival time (seconds)	bulk traffic mean burst interarrival time (seconds)	I/A traffic mean packet interarrival time (seconds)
A0-B0	.1875	.0234	.375	.0468
A0-C0	.1875	.0234	.375	.0468
A0-D0	.1875	.0234	.1875	.0234
A0-D1	.1875	.0234	.1875	.0234
A0-E0	.1875	.0234	.375	.0468
A1-B0	.1875	.0234	.375	.0468
A1-C0	.1875	.0234	.375	.0468
A1-D0	.1875	.0234	.1875	.0234
A1-D1	.1875	.0234	.1875	.0234
A1-E0	.1875	.0234	.375	.0468
B0-C0	.1875	.0234	.375	.0468
B0-D0	.1875	.0234	.1875	.0234
B0-D1	.1875	.0234	.1875	.0234
B0-E0	.1875	.0234	.375	.0468
C0-D0	.1875	.0234	.1875	.0234
C0-D1	.1875	.0234	.1875	.0234
C0-E0	.1875	.0234	.375	.0468
D0-E0	.1875	.0234	.1875	.0234
D1-E0	.1875	.0234	.1875	.0234

Figure 4.19 shows that the percent bandwidth utilization values are basically the same in the shared system as those in the nonshared system. Examination of the data at each capacity value revealed that less than a 4.4% difference exists between the two systems. Notice, however, that the A to B link in both systems has a higher utilization rate than the C to E link. This occurs because the A to B link includes site A's traffic going to site D. The mean end-to-end delay versus capacity plot is shown in Figure 4.20, and again the shared system has a significantly less end-to-end delay (approximately 80% reduction). This can be explained the same way as it was for the 'equal' traffic load case in the previous section.

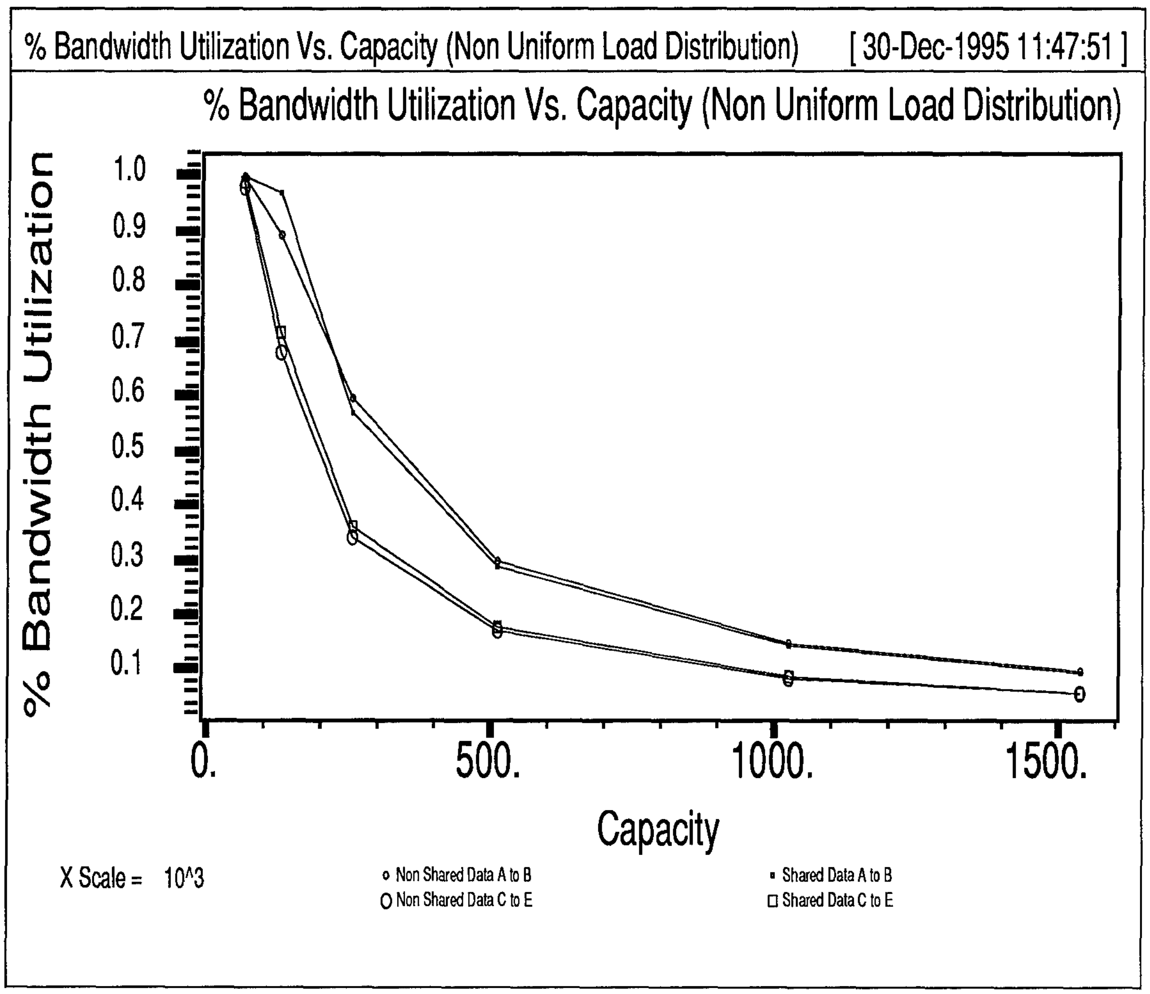

Figure 4.19 Percent Bandwidth Utilization (x 100) Versus Capacity (Non Uniform Load)

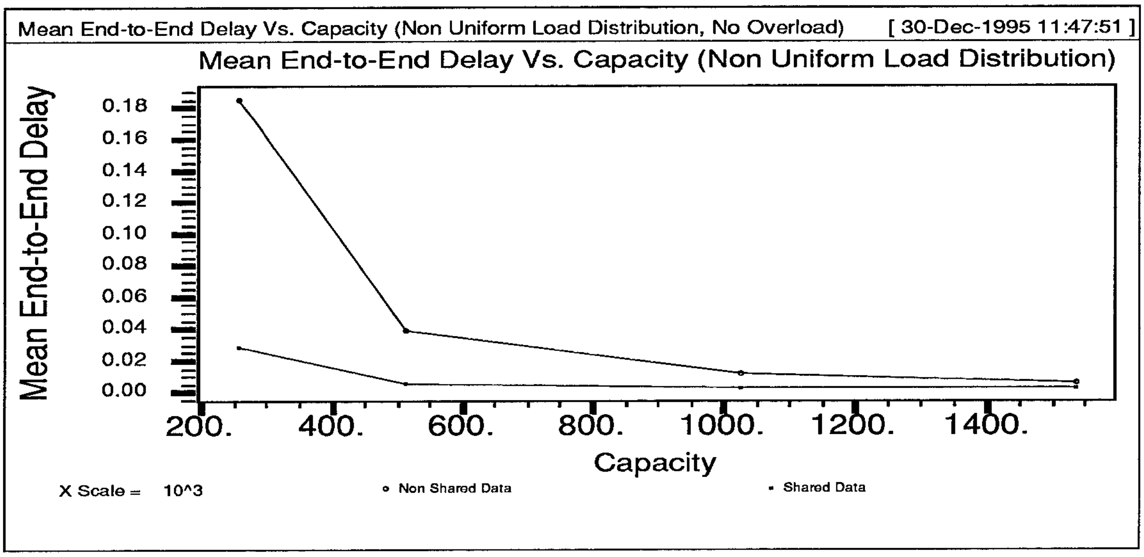

Figure 4.20 Mean End-to-End Delay Versus Capacity (Non Uniform Load)

Figure 4.21 shows that when the capacity is kept constant, and the load is varied, the shared system's mean end-to-end delay remains significantly less than the nonshared system's system. An examination of the mean end-to-end delay revealed an 84% reduction when the load equals 2/3 (approximately 194 kbps) and an 80.1% improvement when the load equals 1/6 (approximately 48 kbps). This is as expected and agrees with the results obtained using an equal load distribution.

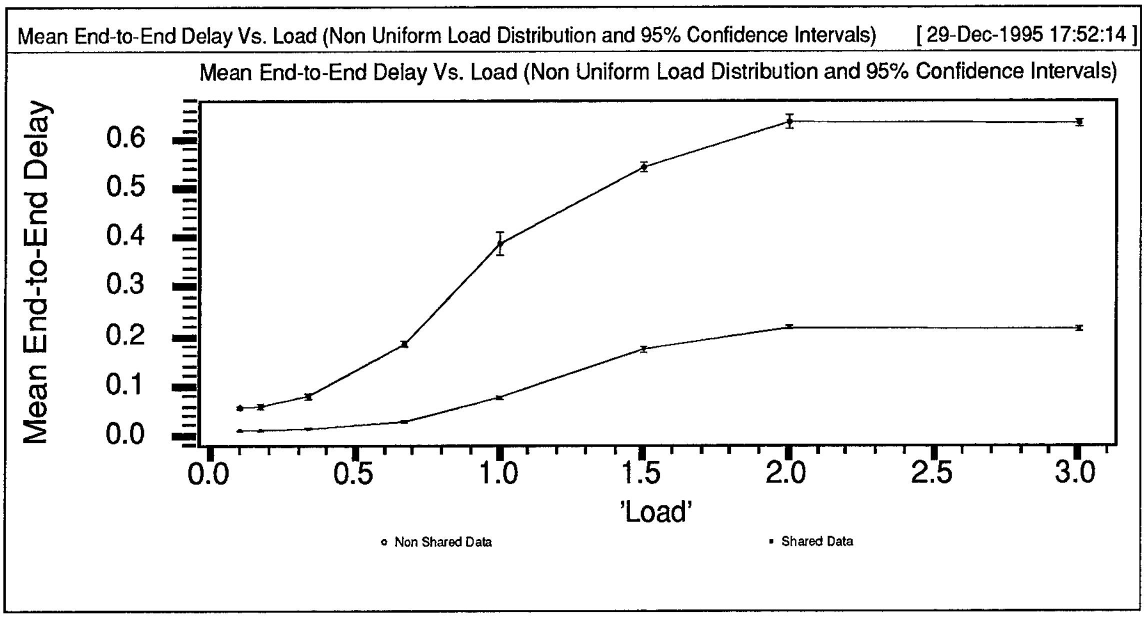

Figure 4.21 Mean End-to-End Delay Vs. Load (Non Uniform Load)

4.4 Summary

This chapter has shown the results of the shared bandwidth system versus the nonshared bandwidth systems. In both of the two node and five node configurations, the system traffic load and the link capacities have been varied to see how they affect performance. In the two node network, the plots showing the average number of packets in the two different types of queues (processing, transmission) revealed that an extensive amount of packets accumulated in the transmission queue, while a negligible amount (less than two packets total) accumulated in the process queue. This showed that the transmission processing time (which is a function of the link capacity) is indeed a bottleneck point. Experiments were also performed to determine whether or not a single shared T1 link could be used in place of two nonshared links. The results showed this depended on the input traffic load and on the specified upper bound for average end-to-end delay. It was found that provided the peak traffic load did not exceed eight (each LAN generated approximately 688 kbps at this load), and assuming a specified upper bound of 200 ms, a single T1 link would be sufficient. Moreover, the results pointed out that if shared bandwidth is incorporated into the DoD network, some of the nonshared dedicated links may be found to be unnecessary. This may save the DoD a lot of money in terms of leasing costs.

In the five node system, a fixed queue length has been established to be equal to 100, and the percent of packets dropped at the bottleneck points (transmission queues) showed a negligible amount until the load exceeded the peak traffic load (194 kbps). Examination of the data for both the shared and the nonshared system revealed that zero packets were dropped with load input values less than 194 kbps, and less than 0.3% when the load value equaled 194 kbps. In terms of bandwidth utilization the two systems were less than 4.4 % different from each other. The most striking distinction between the nonshared and the shared system was in the average end-to-end delay. The results showed that there was an approximate 80% improvement in the shared system's mean end-to-end delay. Overall, the results provided a convincing argument that incorporating shared bandwidth will increase the performance of the network.

5 *Conclusions and Future Recommendations*

5.1 Introduction

This research had the following objective: determine if bandwidth on a packet-switched network could be used more efficiently if it were to be shared versus nonshared. The first topic covered in this chapter, is a brief overview of what had been accomplished. In short, the research consisted of two parts. First, two node shared and nonshared systems were created to see if a shared single channel T1 link could replace two nonshared T1 links. Second, five node shared and nonshared systems were constructed, using equal bandwidth in each link and the same topology, thus allowing a performance comparison to be made. After the overview, the conclusions reached on the two experiments are presented. The two node network system conclusions are presented first and tell whether or not a single shared channel can be used in place of two nonshared channels. Subsequently, the conclusions reached on the five node systems address which system outperforms the other. Following the conclusions, recommendations for further research are presented. And finally, before closing out the chapter, an overall summary is provided.

5.2 Overview

In Chapter 1, the problem was defined, and a generalized plan of attack was presented. The DoD wanted to see if bandwidth could be used more efficiently if it was shared versus nonshared. The scope of the research was narrowed down to a performance study of the data traffic using packet switching technology. Specifically, a system using shared bandwidth was to be compared to a system using nonshared bandwidth. Because there were some vagueness in what defines shared bandwidth, some research had to be done on different ways bandwidth can be shared. This step was accomplished in Chapter 2. The remainder of Chapter 2 consisted of reviewing current literature pertaining to the major technical issues of packet-switching, and on the modeling and analysis techniques of wide-area networks.

The methodology used in solving this investigation was the topic of Chapter 3. In this chapter, the performance metrics were stated, and the input parameters were specified. Average end-to-end delay and percent bandwidth utilization were used as the main performance indicators. The input traffic load consists of both bulk data and interactive traffic. A packet train model was used to input the bulk traffic, and a constant plus exponential was used to generate interactive traffic. The modules were constructed, described, and illustrated. A two node experiment and a five node experiment were conducted. In each experiment, a shared system and a nonshared system were constructed. Steady state analysis followed. In this section, simulation warm up time, and run time were determined. Lastly, the models were verified and validated.

In Chapter 4, the results were presented. Specifically, average end-to-end delay and percent bandwidth utilization performance parameters were displayed in a manner which allowed a comparison to be made between the shared system and the nonshared system. For the two node systems, infinite queue capacities were assumed, and therefore the average length of the queues were also displayed. In the five node case, the percentage of packets dropped was also displayed, and used as an indicator of network saturation.

5.3 Conclusions

Before discussing the conclusions, a quick summary of the differences between the shared and the nonshared systems are given. This is done for both the two node configuration and the five node configuration.

5.3.1 Shared Versus Nonshared in the Two Node Systems

In the two node case, the nonshared bandwidth system's link contained two channels: one channel for each LAN to LAN communication. Whereas in the shared bandwidth system the link was reduced to a single channel in which all four LANs had to share. Thus in the shared system, the bandwidth available was half that available in the nonshared system. The main objective of the experiment was to determine whether or not a single shared T1 link could be used in place of two nonshared T1 links. The systems' performance metrics were recorded and plotted in both configurations.

This step showed how average end-to-end delay and percent bandwidth utilization were affected by switching from the two channel nonshared system to a single channel shared system.

5.3.2 Two Node System Conclusion

The results showed the shared system to be more productive than the nonshared system, provided the input traffic load did not exceed the peak traffic load (i.e., cause the shared system to be saturated). It was found that the percent bandwidth utilization was 20% higher for a light traffic load (343 kbps or 440 packets per second) and 40% higher for the peak traffic load (688 kbps or 880 packets per second). In terms of responsiveness, the nonshared system was found to have a smaller average end-to-end delay. However, the main concern was to determine whether or not a single shared T1 link could be used in place of two nonshared T1 links. The results showed that this determination depended upon the peak traffic load and on a specified upper bound for average end-to-end delay. Provided the peak traffic load did not exceed eight (688 kbps or 880 packets per second), and an assumed upper bound of 200 ms on average end-to-end delay, the single shared T1 link was found to be sufficient (200 ms was chosen because it would allow real-time packetized voice transmission). Thus, by sharing the bandwidth, the cost of an additional T1 link could be saved. The main impact of this experiment is that it showed that costs of leasing communication links could be reduced if shared bandwidth were used in place of nonshared bandwidth.

5.3.3 Shared Versus Nonshared in the Five Node Systems

In the five node systems, both the nonshared and the shared system had the same link capacities. This approach was different than the two node system. In the two node system, one channel was removed when switching from the nonshared to the shared system, resulting in a reduction of one half the bandwidth. In the five node system, the modeling approach allowed the channels to combine to form a single channel when switching from the nonshared system to the shared system, resulting in the same amount of total bandwidth available in both systems. The topology of the links was the same for both systems. When the links operated in the nonshared mode, a separate channel within the link was dedicated to a single source-destination LAN pair. In the shared mode, the links consisted of a single channel, shared by all source-destination LAN pairs.

5.3.4 Five Node System Conclusion

The results show that the shared system clearly outperforms the nonshared system. In this case the productivity (percent bandwidth utilization) basically remained the same, however, the responsiveness (average end-to-end delay) showed a remarkable improvement over the nonshared system. Specifically, the percent bandwidth utilization between the shared and the nonshared system differed by less than 4.4%. It was also found that the percentage of packets dropped by each system was approximately equal as well. When the traffic load was below the peak traffic rate (194 kbps), not a single packet was dropped by either system. With a traffic load equal to 194 kbps, less than 0.3% packets were dropped by the shared system and less than 0.2% were dropped by the nonshared system. In regards to responsiveness, the shared system dominated the nonshared system by showing an approximate 80% decrease in average end-to-end delay. The main impact of this experiment is that it showed that shared bandwidth clearly outperforms nonshared bandwidth given that each system has the same amount of capacity in each link and that the same topology is used.

5.4 Future Recommendations

This investigation provided a comparison between a shared and nonshared bandwidth system under a common set of system operating assumptions which had not been previously performed. Due to the diversity, complexity, and time restraints of this investigation, certain enhancements to the simulation could not be implemented. These enhancements to the simulation form a base for future research in the area of comparing performance of shared versus nonshared bandwidth. These enhancements are as follows:

1. According to Floyd [Flo94], current routers generally have a single queue for each output port. Floyd further states that future routers could have separate queues for separate classes of traffic. Thus, one enhancement would be to modify the shared system model to include priority queues and incorporate packetized real-time traffic. Then compare the performance to that of sending real-time traffic across a circuit-

switched network.

2. Incorporate failures into the system, and compare the performance between the shared and the nonshared system.

3. Using Designer, construct Local Area Network models to include hosts which implement all the layers of the protocol (i.e., flow control, media access control, etc.) and attach them to the input ports of the five node network model's nodes (gateways) and verify that the shared bandwidth still outperforms the nonshared bandwidth system.

5.5 Summary

This chapter closes out the thesis effort. After the overview of the research effort was provided, the conclusions were given. This investigation revealed a method which could be used to make a wide-area network operate more efficient. In this case 'more efficient' means higher resource utilization and increased responsiveness. It has been shown that using shared bandwidth versus nonshared bandwidth can result in a savings in terms of leased line costs. Given a peak traffic load, and a specified upper bound on average end-to-end delay, a determination can be made whether or not a single shared channel can be used in place of multiple nonshared channels. Further, given that links might already be owned, the investigation revealed that using shared bandwidth over nonshared bandwidth results in a 80% improvement in responsiveness. And finally, given the assumptions of the operating environment, the research proved beyond a shadow of doubt that the shared system outperforms the nonshared system.

Appendix

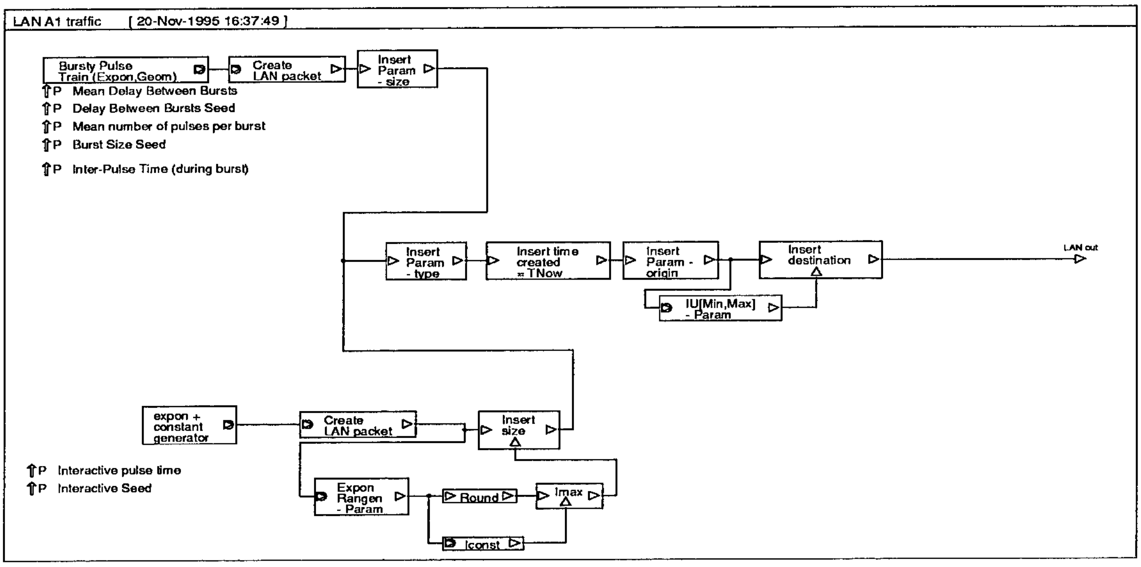

Figure A 1 Traffic Generator

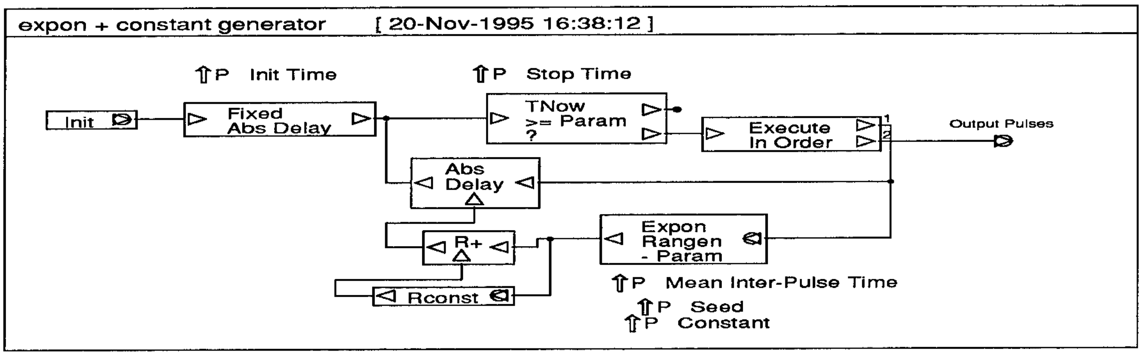

Figure A 2 Exponential Plus Constant Generator

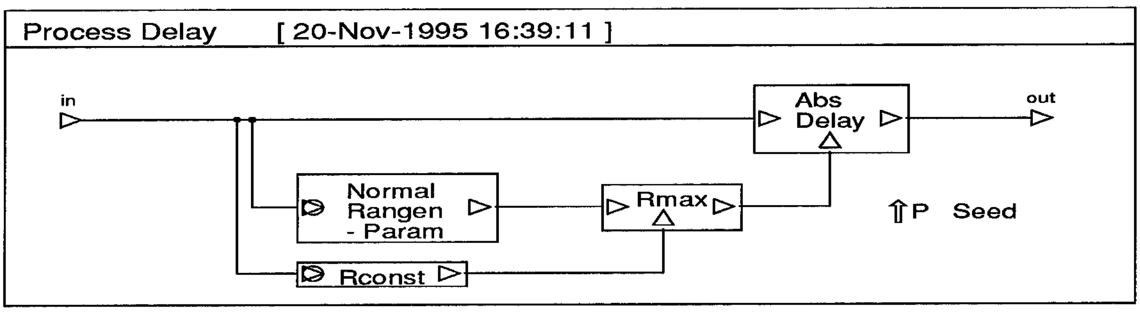

Figure A 3 Process Delay Module

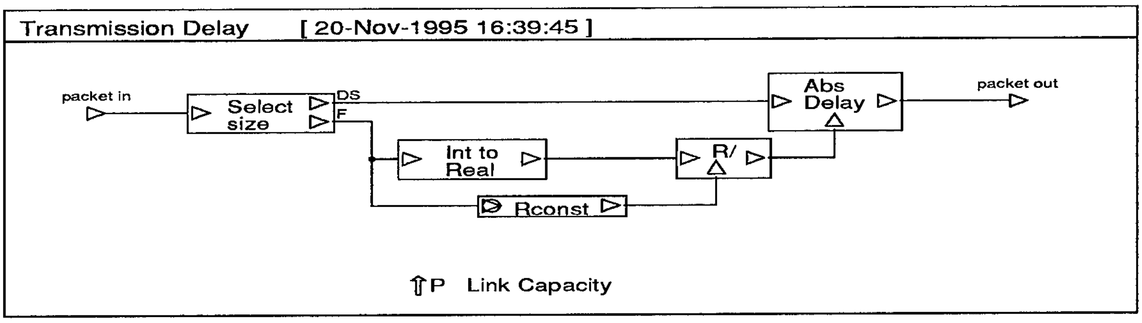

Figure A 4 Transmission Delay Module

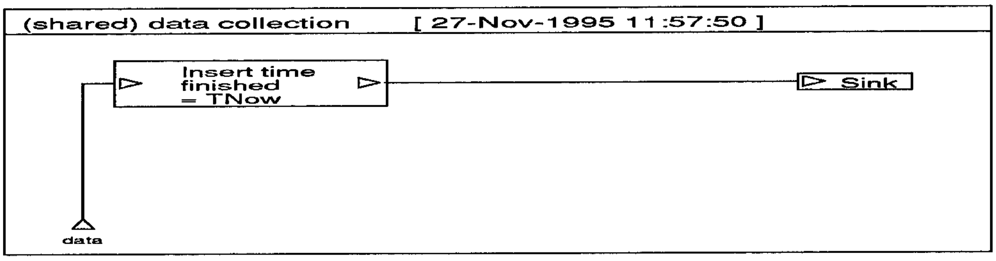

Figure A 5 Data Collection Module

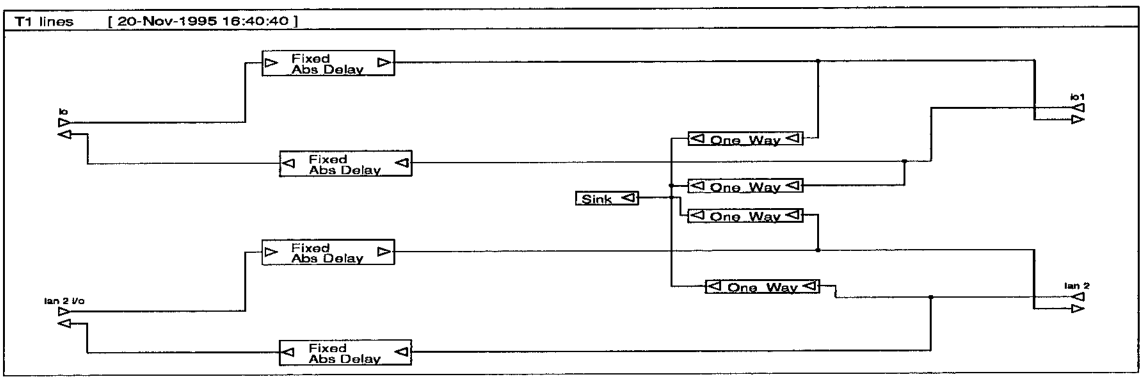

Figure A 6 T1 Link Non Shared Bandwidth

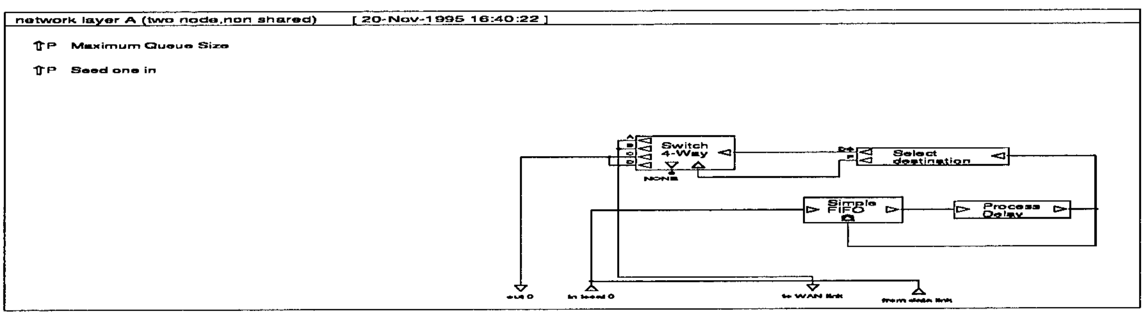

Figure A 7 Network Layer for the Non Shared Bandwidth Configuration

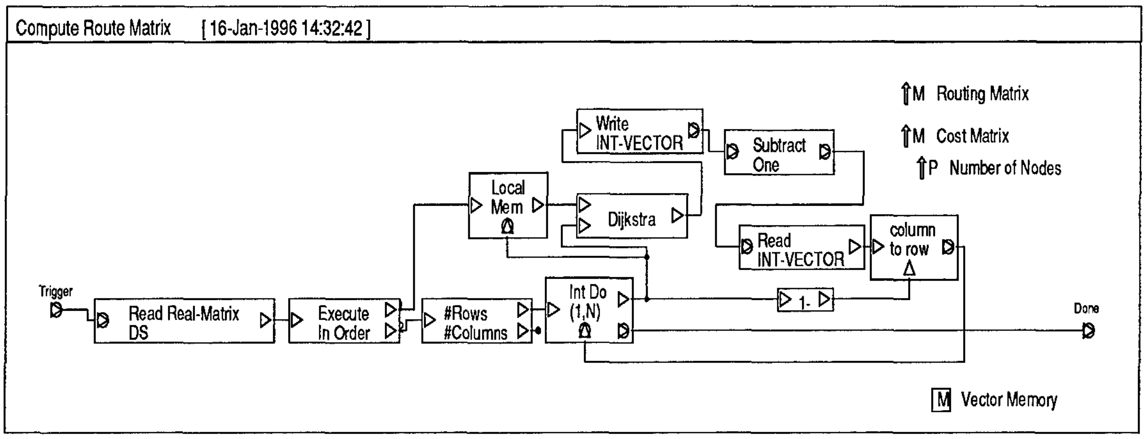

Figure A 8 Compute Route Matrix

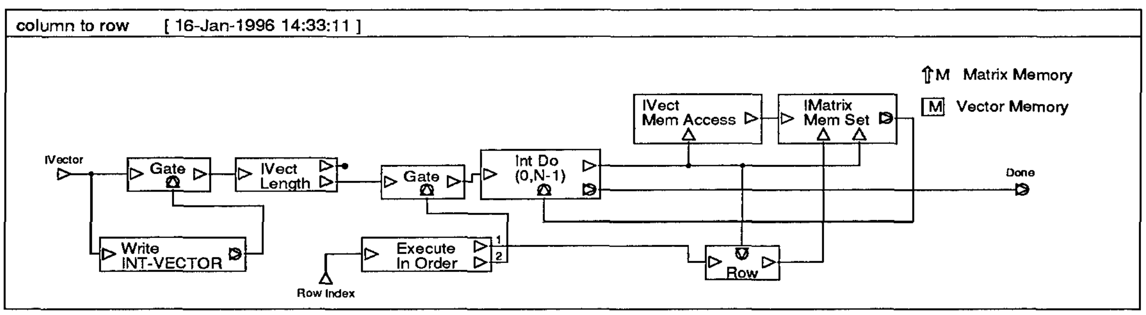

Figure A 9 column to row

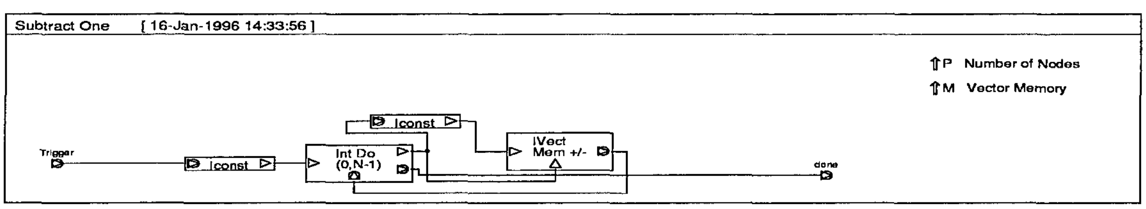

Figure A 10 Subtract One

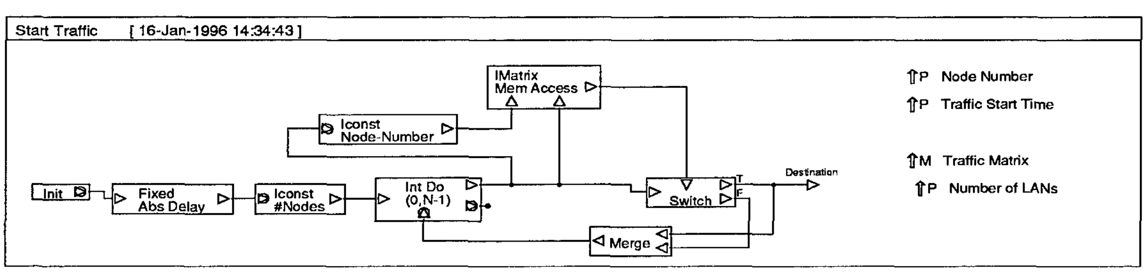

Figure A 11 Start Traffic

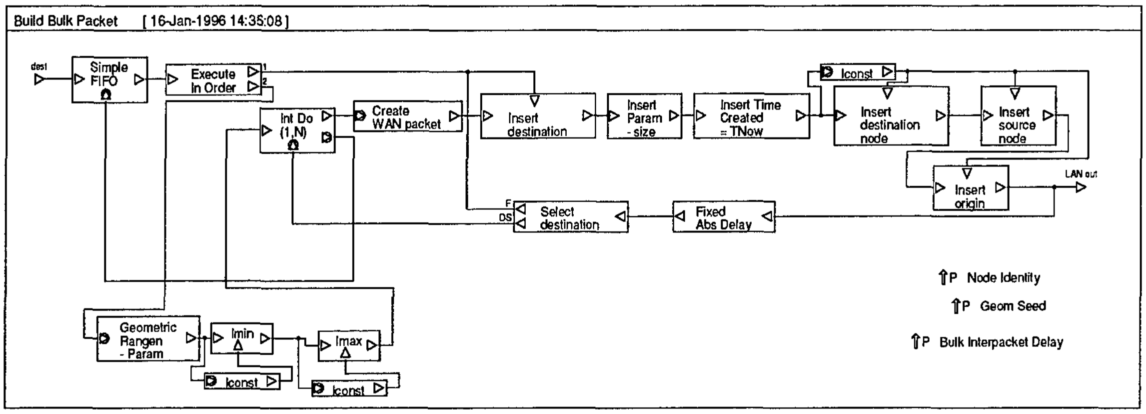

Figure A 12 Build Bulk Packet

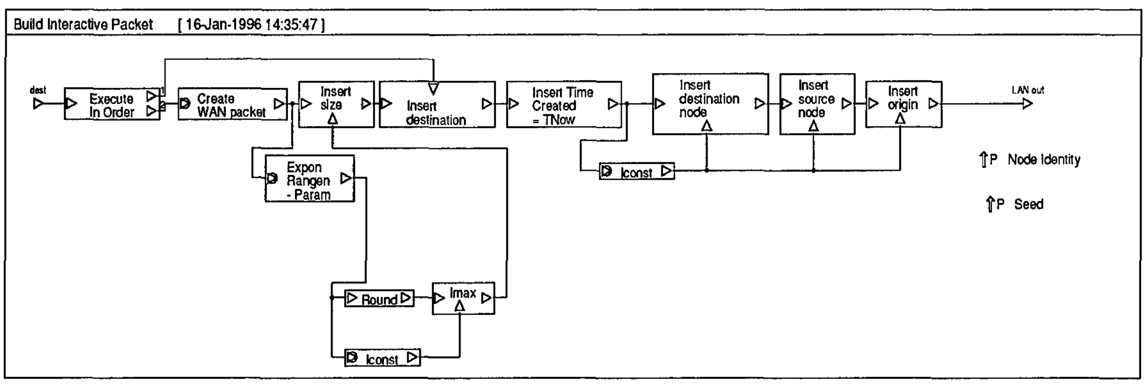

Figure A 13 Build Interactive Packet

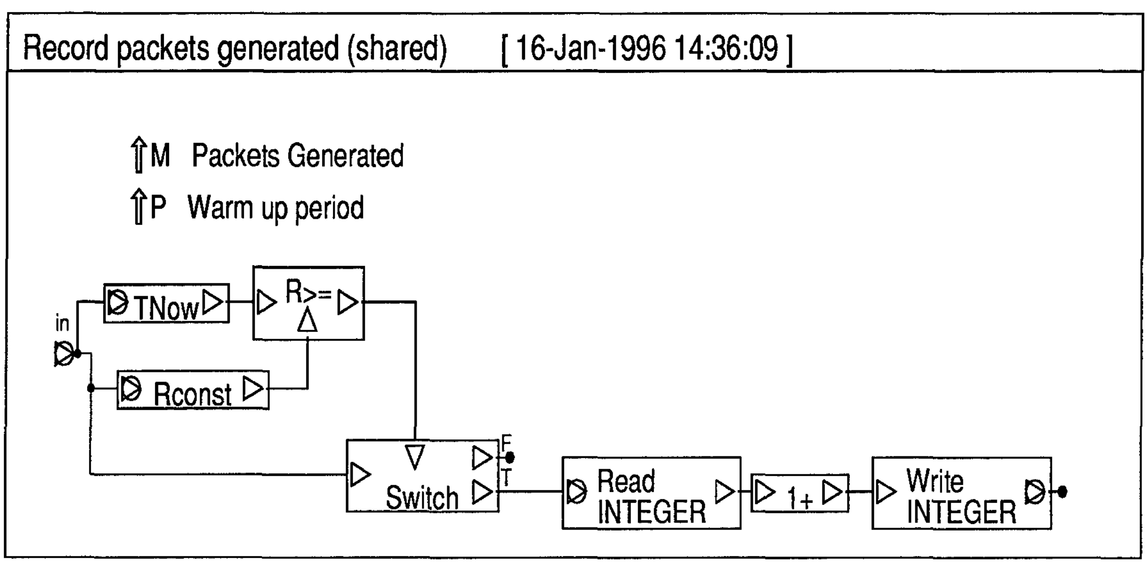

Figure A 14 Record Packets Generated

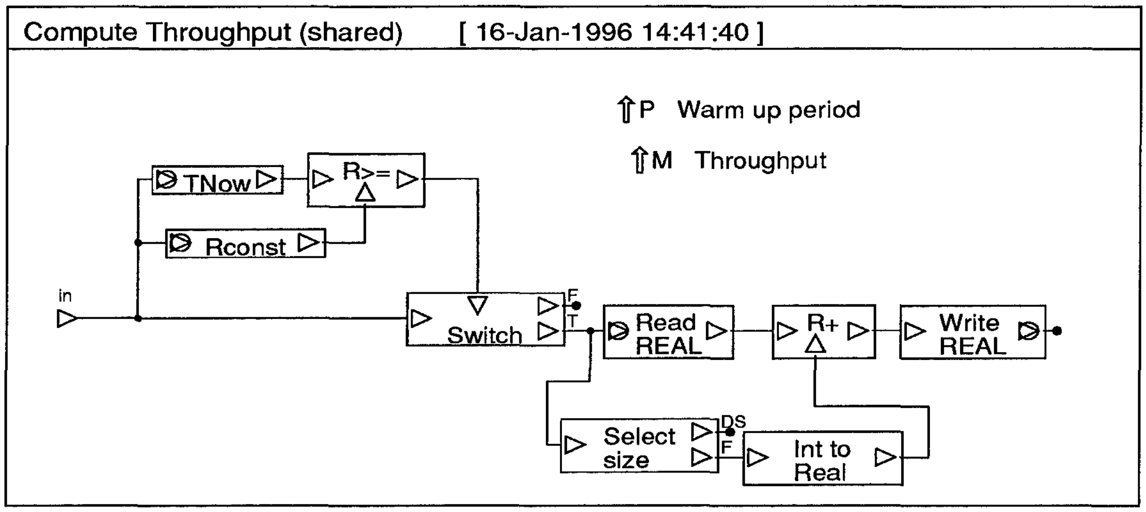

Figure A 15 Compute Throughput

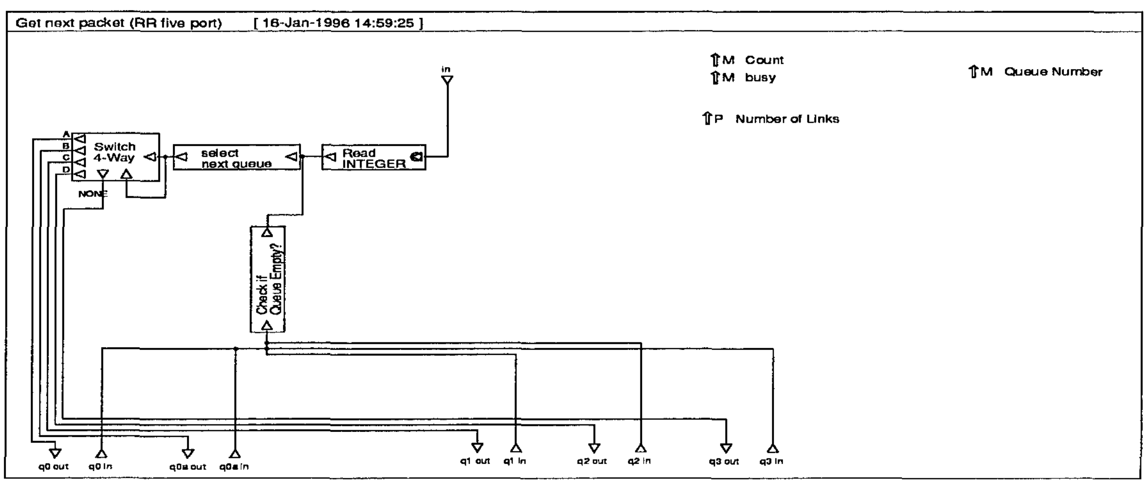

Figure A 16 Get Next Packet

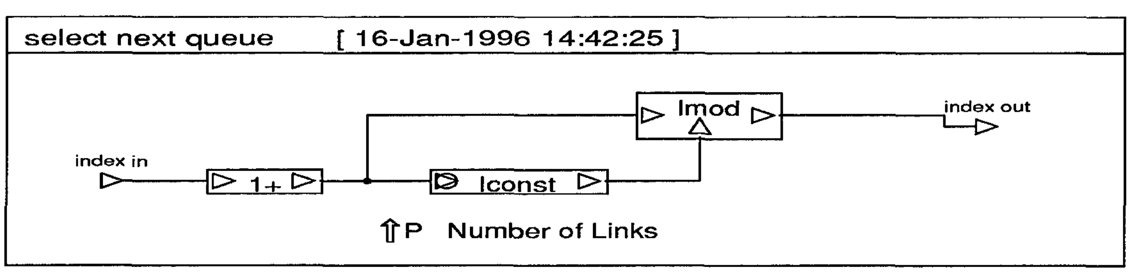

Figure A 17 Select Next Queue

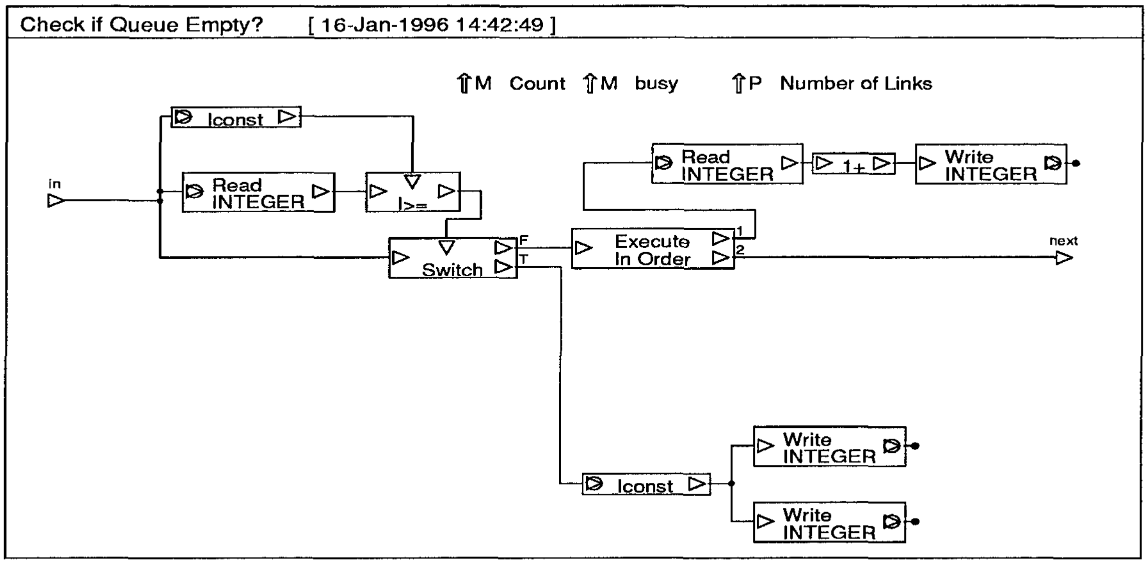

Figure A 18 Check if Queue Empty?

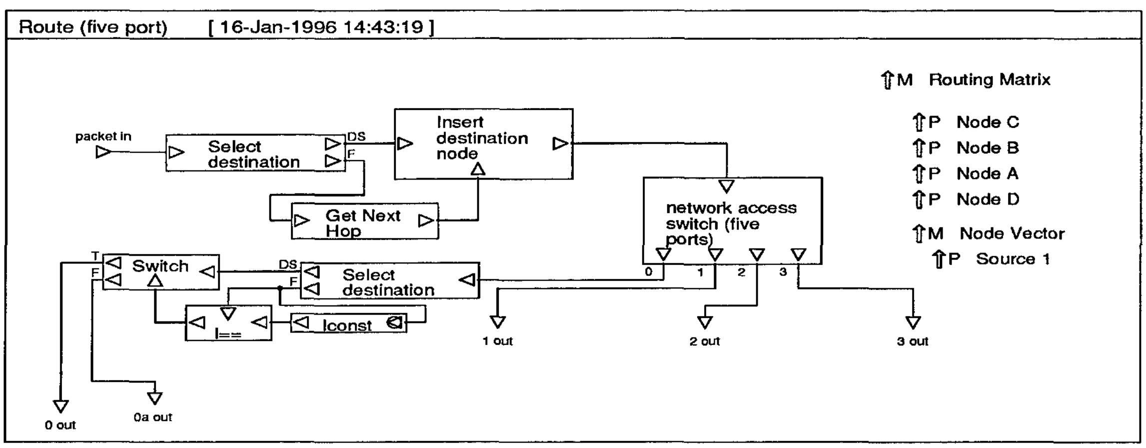

Figure A 19 Route (five port)

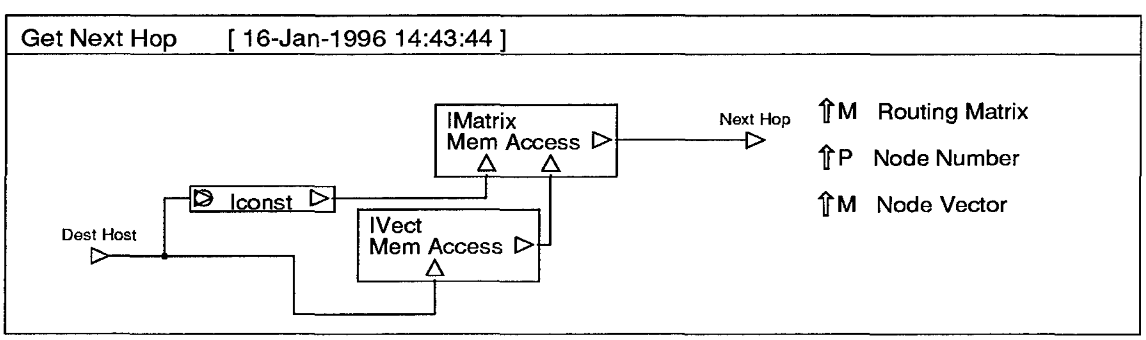

Figure A 20 Get Next Hop

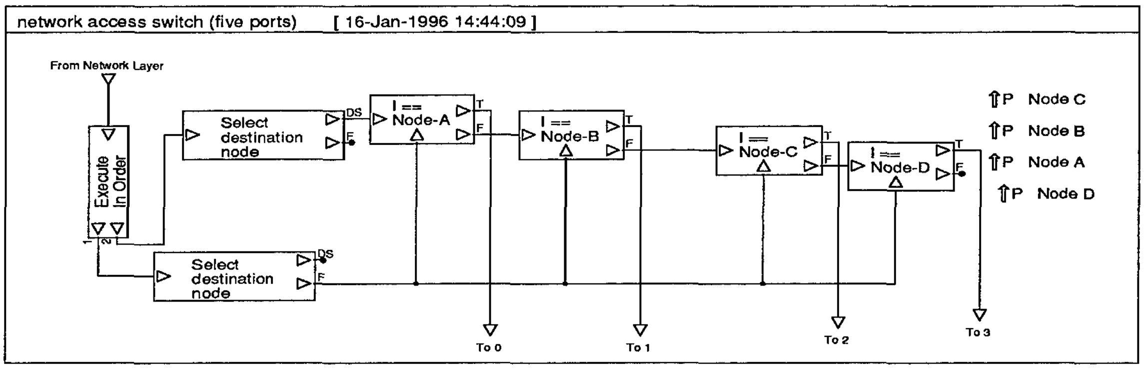

Figure A 21 network access switch

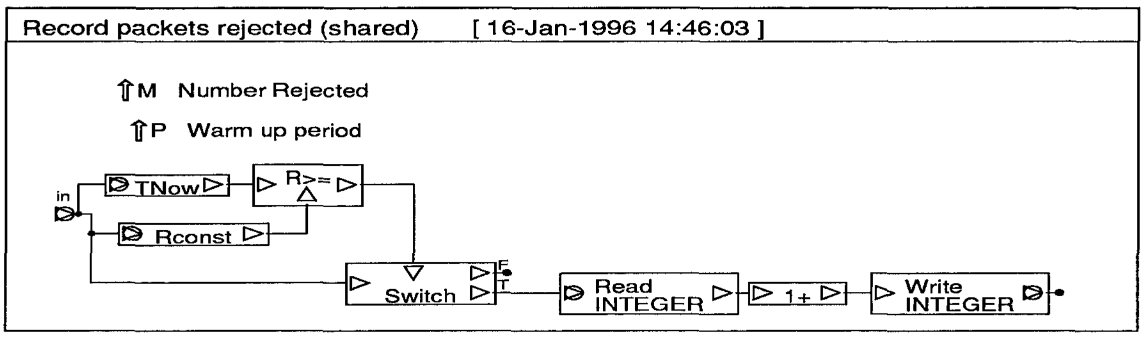

Figure A 22 Record packets rejected

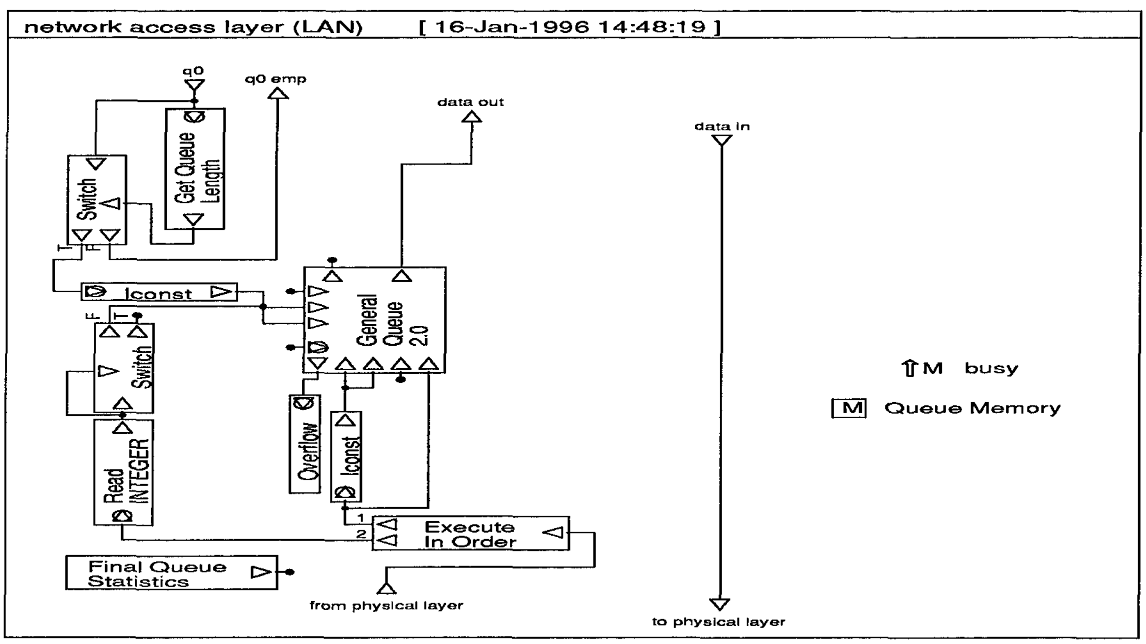

Figure A 23 network access layer (LAN)

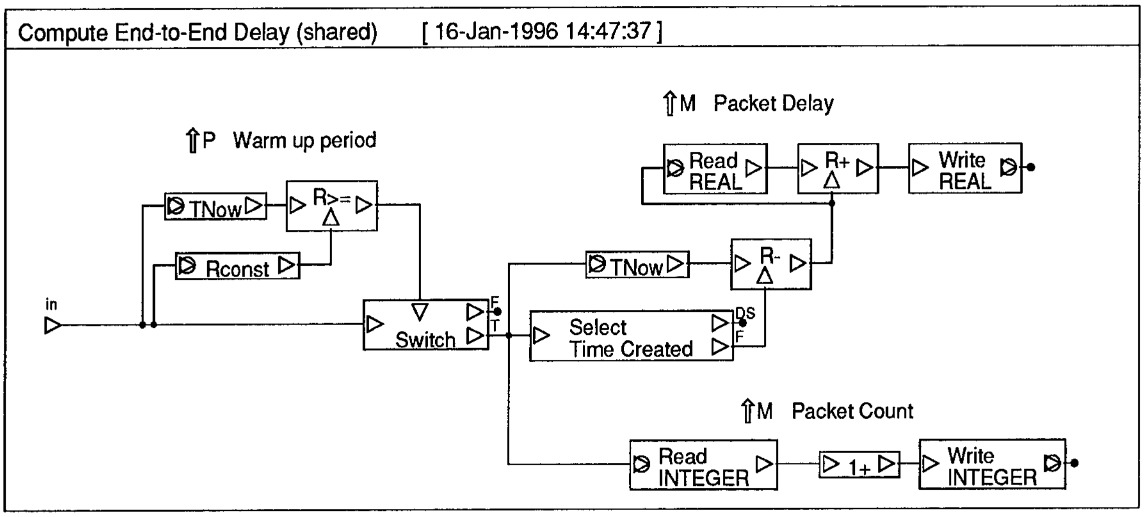

Figure A 24 Compute End-to-End Delay

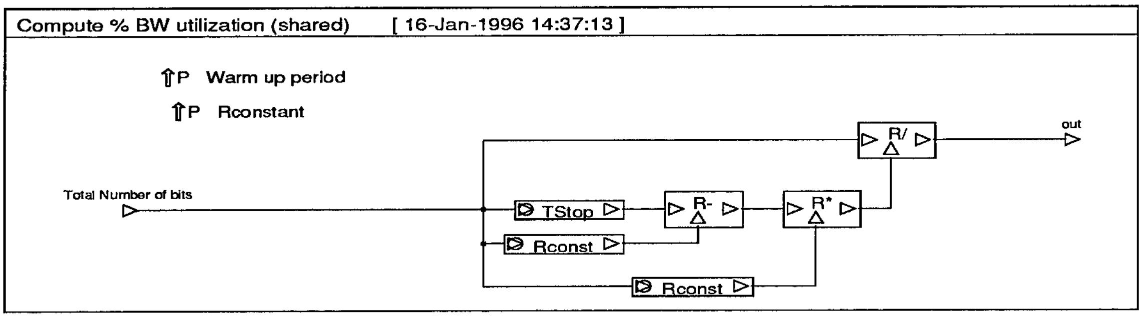

Figure A 25 Compute % BW utilization

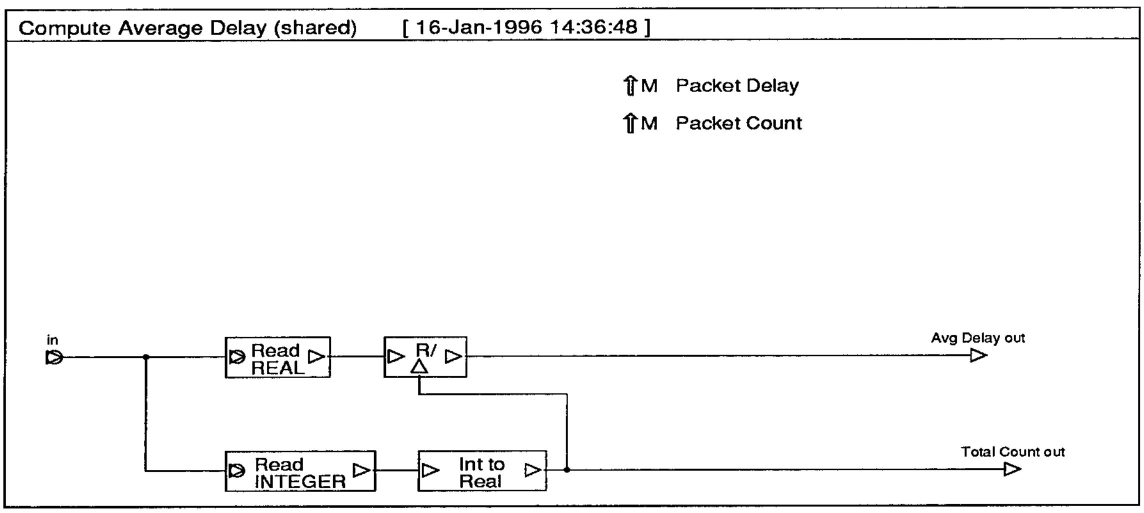

Figure A 26 Compute Average Delay

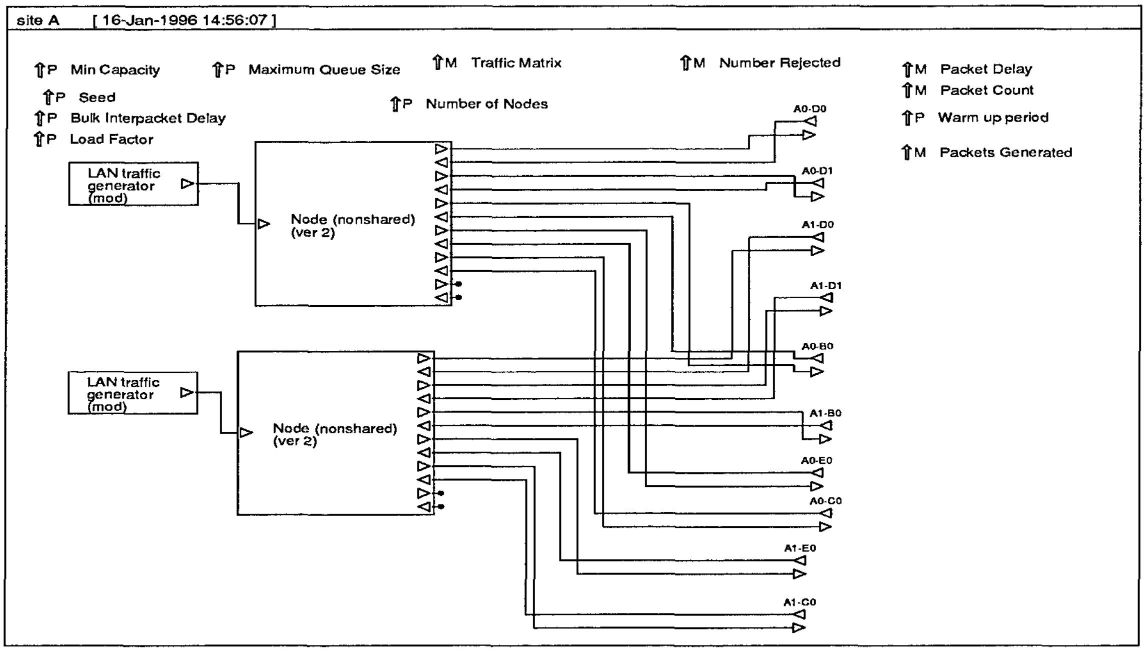

Figure A 27 Site A

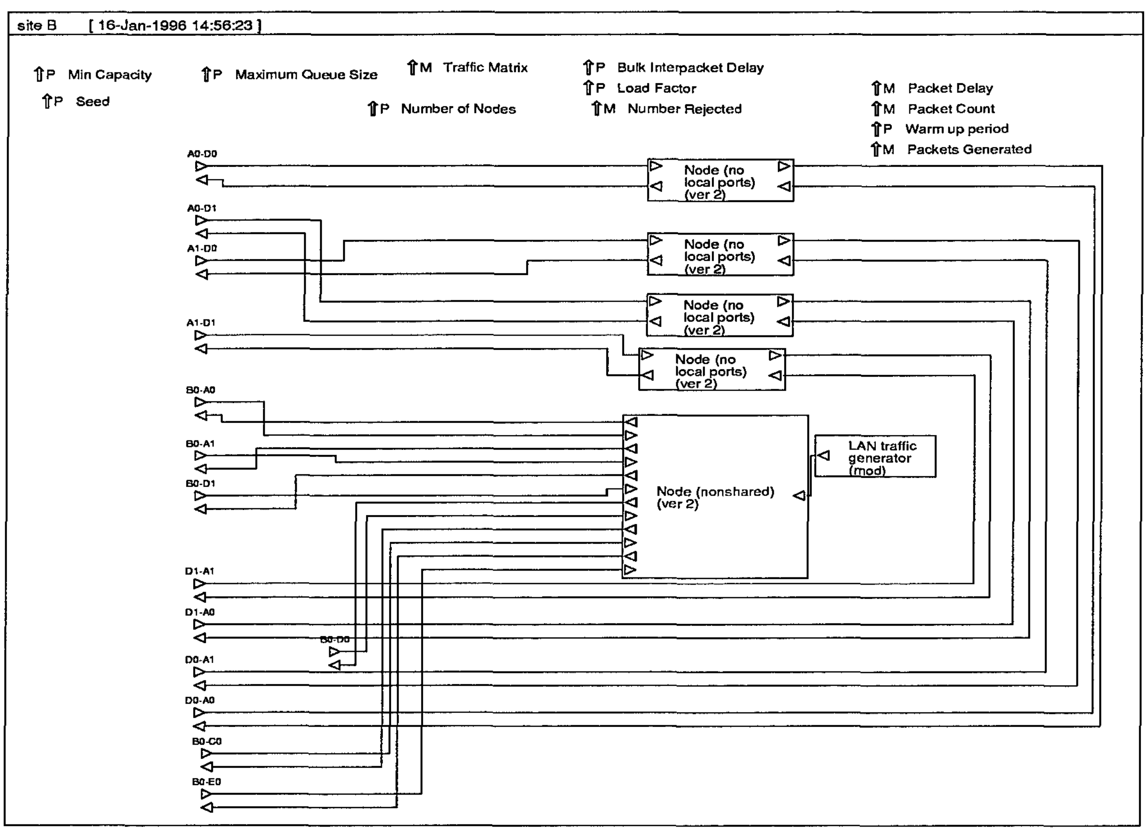

Figure A 28 Site B

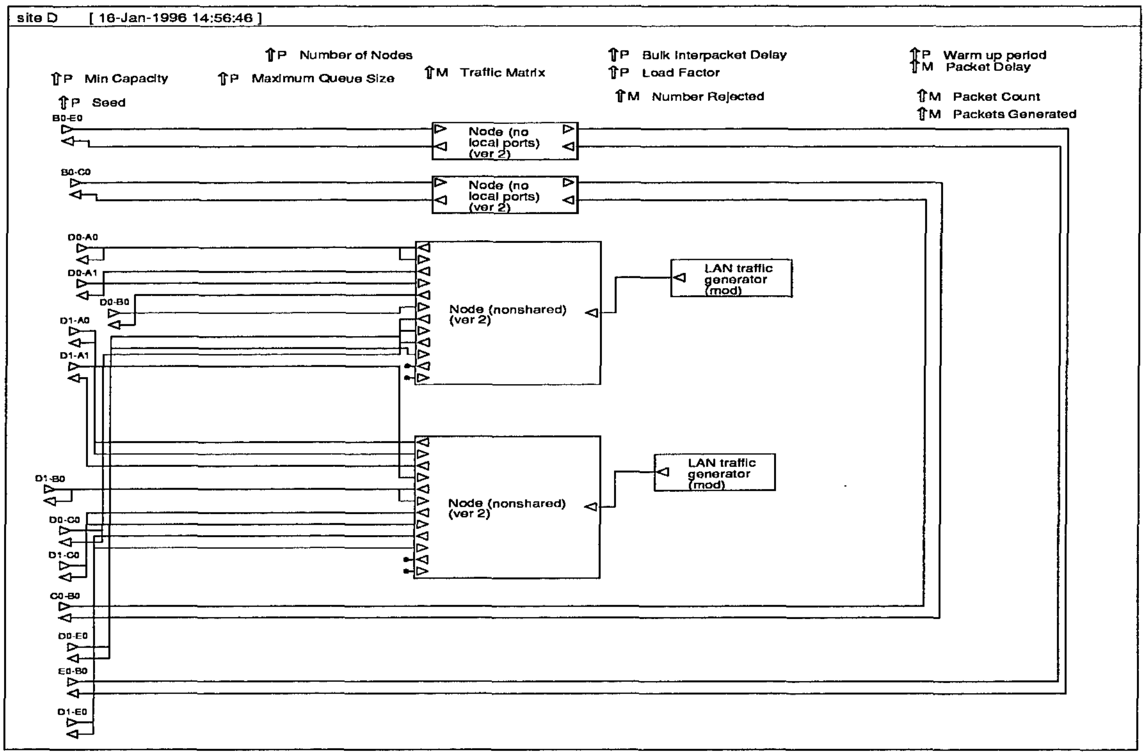

Figure A 29 Site D

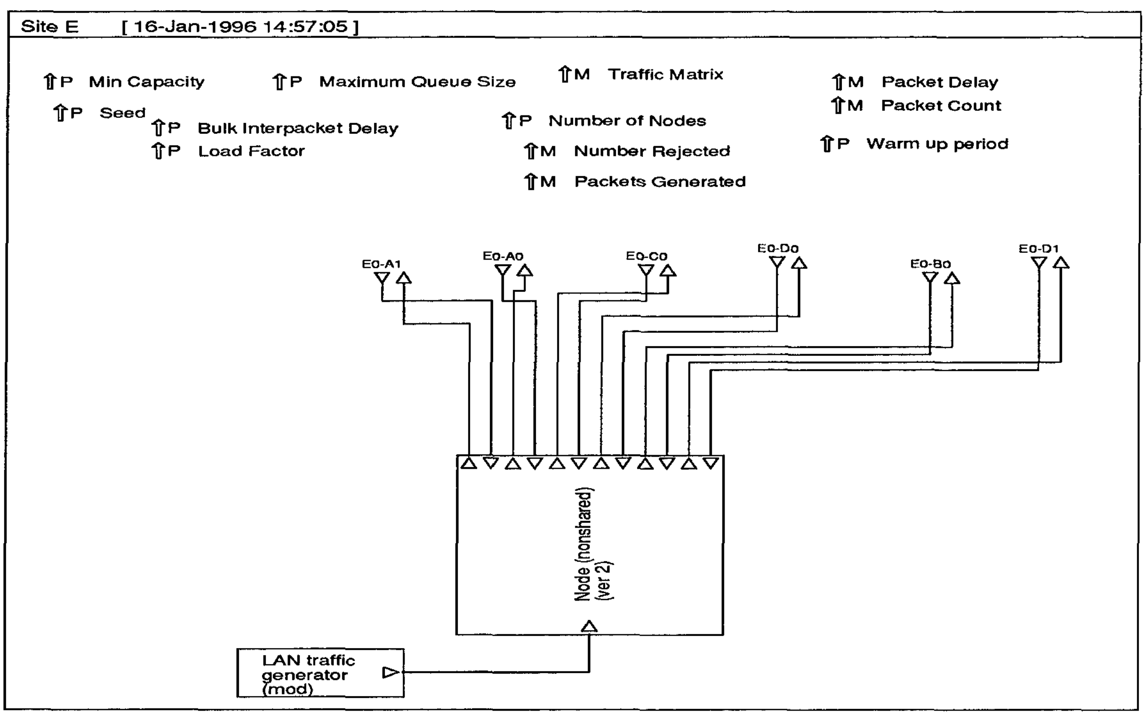

Figure A 30 Site E

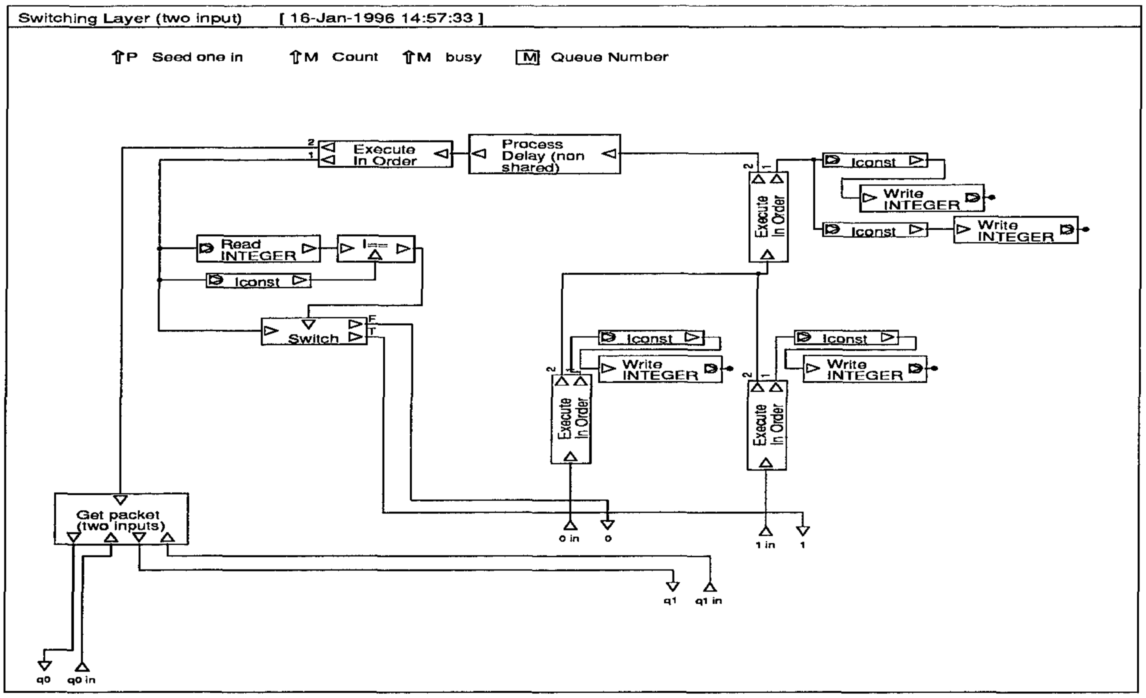

Figure A 31 Switching Layer (two input)

Bibliography

[AgJ90] Agrawala, Ashok K. Jain, Bijendra N. "Deterministic Analysis of Queues in Tandem with Application to Computer Networks." Technical Report, CS-TR-2511, University of Maryland, July 1990, pp. 1-26.

[ArK94] Aras, Caglan M. Kurose, James F. Reeves, Douglas S. Schulzrinne, Henning. "Real-Time Communications in Packet-Switched Networks." Proceedings of the IEEE, Vol. 82, No. 1, January 1994, pp. 122-139.

[Bol93] Bolot, Jean-Chrysostome. "End-to-End Packet Delay and Loss Behavior in the Internet." SIGCOMM'93, pp.289-298.

[BrC93] Braun, Hans-Werner. Chinoy, Bilal. Claffy, Kimberly. Polyzos, George. "Analysis & Modeling of Wide-Area Networks: Annual Status Report." Applied Network Research, San Diego Supercomputer Center and University of California, San Diego, GA-A21224. February 1993.

[CaD91] Caceres, Ramon. Danzig, Peter B. Jamin, Sugih. Mitzel, Danny J. "Characteristics of Wide-Area TCP/IP Conversations." ACM, Vol #21, #4, September 1991.

[Cha92] Chan, H.W. "Simulation of Flow Control Schemes - Backpressure and Window Control." Simulation, February, 1992, pp. 95-106.

[ChA92] Cheng Sheng-Tzong. Agrawala Ashok. "Modeling of a Virtual Circuit Communications Network." Rome Laboratory Technical Report, RL-TR-94-37, November 1992, pp. 4-29.

[CoS91] Comer, Douglas E. Stevens, David L. Internetworking With TCP/IP Vol II: Design, Implementation, and Internals. Prentice Hall, 1991.

[Fei93] Feit, Sidnie. TCP/IP Architecture, Protocols, and Implementation. Published by McGrawl-Hill, Inc., 1993

[FeV90] Ferrari, Domenico. Verma, Dinesh C. "A Scheme for Real-Time Channel Establishment in Wide-Area Networks." IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, April 1990, pp 368-379.

[FiH76] Fischer, M. J. Harris, T. C. "A Model for Evaluating the Performance of an Integrated Circuit- and Packet-Switched Multiplex Structure." IEEE Transactions on Communications, Vol. Com-24, No. 2, February 1976, pp 195-202.

[Flo94] Floyd, Sally. "TCP and Explicit Congestion Notification." ACM SIGCOMM Computer Communication Review, October 1994, pp. 8-22.

[FrM94] Frost, Victor S. and Melamed, Benjamin. "Traffic Modeling for Telecommunications Networks." IEEE Communications Magazine, March 1994, pp. 70-80.

[Gru81] Gruber, John G. "Delay Related Issues in Integrated Voice and Data Networks." IEEE Transactions On Communications, vol com-29, no 6, June 1981, pp. 786-797.

[HuZ94] Hui, Joseph Y. Zhang, Junbiao. "GRAMS: A Distributed Multimedia Service System." Proceedings SPIE - The International Society for Optical Engineering , Volume 2188, February 1994, pp 189-199.

[IlM85] Ilyas, M. Mouftah, H.T. "Performance Evaluation of Computer Communications Networks." IEEE Communications Magazine, April 1985, Vol.23, No.4, pp. 18-29

[Jai90] Jain, Raj. "Performance Analysis of FDDI Token Ring Networks: Effect of Parameters and guidelines for setting TTRT.", ACM, August 1990, pp. 264-268.

[Jai94] Jain Bijendra, N. "Model and Analysis of a Virtual Circuit with Cross Traffic." Rome Laboratory Technical Report, RL-TR-94-37, April 1994, pp. 4-29.

[JaR86] Jain, R. Routhier, S. "Packet Trains - Measurements an a New Model for Computer Network Traffic." IEEE Journal on Selected Areas in Communications, SAC-4(6): September 1986, pp. 986-955

[JeL95] Jeng, Hou-Wa J. Liu, Ming T. "Protocol Conversion in Multimedia Networks: Simulation and Algorithm." Simulation, January 1995, pp 51-60.

[Joh93] Johnson, Johna Till. "Bandwidth On Demand at a Bargain Price." Data Communications, June 1993, pp. 35-36.

[Kar94] Karve, Anita. "Chicago's Ritzy WAN," Case Study, LAN, The Network Solutions Magazine, August 1994, pp 141-144.

[KaS95] Kalyoncu, Hakan. Sankur, Bulent. "Design and Performance Evaluation of Picture Archival and Communication Systems." Simulation, February 1995, pp. 77-89.

[Kes91] Keshav, Srinivasan. "A Control-Theoretic Approach to Flow Control." ACM, September 1991, pp. 3-15.

[Kie90] Kiemele, Mark J. "Mixed Voice/Data Networks." Network Modeling, Simulation, and Analysis, edited by Ricardo F. Garzia and Mario R. Garzia, Bell Telephone Laboratories, Incorporated, 1990.

[LaK91] Law, Averill M. Kelton, W. Kelton. Simulation Modeling and Analysis. Published by McGraw-Hill, Inc. 1991

[LaM94] Law, Averill M. McComas, Michael G. "Simulation in Communications Networks." Proceedings of the 1994 Winter Simulation Conference, December 1994, pp. 166-169.

[LoS92] Lorentzen, Ralph. Skogsrud, Harald. "Routing and Grouping of Circuits by Integer Programming." Networks '92, May 1992, pp.49-54.

[MaS81] Maglaris, Basil. Schwartz, Mischa. "Performance Evaluation of a Variable Frame Multiplier for Integrated Switched Networks." IEEE Transactions on Communications, Vol. Com-29, No. 6, June 1981, pp 800-807.

[PaF94] Paxson, Vern. Floyd, Sally. "Wide-Area Traffic: The Failure of Poisson Modeling." ACM, April 1994. pp. 257-268.

[RoK95] Robinson, Al. Krause, Jeff. Discussions held with the sponsors Captain Al Robinson and Jeff Krause in early June and mid September 1995.

[SaA94] Saadawi, Tarid N. Ammar, Mostafa H. with Hakeem, Ahmed El. Fundamentals of Telecommunication Networks. John Wiley and Sons, Inc. 1994

[San93] Sanghi, Dheeraj. "Flow Control in Computer Networks." Dissertation, University of Maryland, 1993, pp. 1-62.

[ShA92] Shankar, Udaya, A. Alaettinoglu, Cengiz. Dussa-Zieger, Klaudia. Matta, Ibrahim. "Transient and Steady-State Performance of Routing Protocols: Distance-Vector versus Link-State," RL-TR-94-37, Part II of (two), Final Technical Report, April 1994, pp. 454-493.

[ShD94] Shaw, Wade H. Davis, Nathaniel J. Raines, Richard A. "The Application of Metamodeling to Interconnection Network Analysis," ORSA Journal on Computing, Vol 6,No. 4, Fall 1994, pp. 369-375.

[Spo93] Spohn, Darren L. Data Network Design. McGraw-Hill, Inc., 1993

[Sta88] Stallings, William. Handbook of Computer-Communications Standards (Volume 3), Macmillan Publishing Company, 1988.

[Sta95] Stallings, William. ISDN and Broadband ISDN with Frame Relay and ATM (Third Edition). Prentice-Hall, Inc., 1995.

[Sun90] Sunshine, Carl A. "Network Interconnection and Gateways." IEEE Journal on Selected Areas of Communications, Vol. 8, No.1, January 1990, pp.4-11.

[Tay90] Taylor, Steven. "T1 Muxes Meet the Networking Challenge." Datamation, November 1, 1990, pp 67-70.

[Tay95] Taylor, Kieran. "Subrate Voice/Data Muxes Keep WAN Costs in Check." Data Communications, July 1995, pp. 91-98.

[WaA92] Wahida, C. Ahmed, N.U. "Congestion Control using Dynamic Routing and Flow Control." Stochastic Analysis and Applications, 1992, pp. 123-141.

[YaK93] Yates, David. Kurose, James. Towsley, Don. Hluchyj, Michael G. "On per-session end-to-end delay distributions and the call admission problem for real-time applications with QOS requirements," SIGCOMM '93, ACM, September 1993, pp 2-9.

[YaO94] Yamashita, Hideaki. Onvural, Raif O. "Allocation of Buffer Capacities in Queuing Networks with Arbitrary Topologies." Annals of Operations Research, 48(1994), pp. 313-332.

[YaR95] Yang, Cui-Qing, Reddy, Alapati V. S. "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks." IEEE Network Magazine, July-August 1995, pp. 34-45.

[Zha90] Zhang, Lixia. "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks." ACM, August 1990, pp. 19-29.

Vita

Captain John Patrick Stevens [redacted]. He holds a Bachelor of Science in Computer Science from Wright State University (1990). He was commissioned in the United States Air Force in 1990 and entered the Air Force Institute of Technology in August 1994.

P[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March, 1996

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
PERFORMANCE STUDY OF SHARED VERSUS NONSHARED BANDWIDTH ON A PACKET-SWITCHED

5. FUNDING NUMBERS

6. AUTHOR(S)
John P. Stevens, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2750 P Street
WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENG/96M-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of Defense (R55)
9800 Savage Road
Fort George Meade, MD 20755-6000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
In wide-area computer data communications, many networks have evolved by satisfying increased user demands in the most expedient manner. In some cases, new users' demands are satisfied by installing a new link, rather than sharing the links that are already in place. This research investigates the differences in performance between using a dedicated link for each source-destination pair (nonshared bandwidth) and using a single link to be used by all source destination pairs (shared bandwidth). Simulation models are developed for a wide-area network using shared bandwidth, and a wide-area network using nonshared bandwidth. The quality of service offered by each network is based on its responsiveness and productivity. Responsiveness will be measured in terms of average end-to-end delay of packet transmission, and productivity will be measured in terms of percent bandwidth utilization. The networks are modeled under a common set of operating assumptions and system environment. This allows for accurate comparison of packet delay and bandwidth utilization. Two variable input parameters are used in the simulation: intensity of input traffic load, and amount of link capacity. Provided that the intensity of the input traffic load remains below the network saturation level, it is shown that the shared system clearly outperforms the nonshared system. This result occurs for both a uniform and a nonuniform traffic load destination distribution.

14. SUBJECT TERMS
Shared Bandwidth, Network Performance, Packet-Switching
Wide-area Communications Network

15. NUMBER OF PAGES
136

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.