PERFORMANCE EVALUATION OF PACKET VIDEO TRANSFER OVER LOCAL AREA NETWORKS

by

Jie Lu

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

Scott F. Midkiff, Ph.D., Chairman

Nathaniel J. Davis, Ph.D.

Ira Jacobs, ~~Ph.D.~~

October 1993

Blacksburg, Virginia

PERFORMANCE EVALUATION OF PACKET VIDEO TRANSFER OVER LOCAL AREA NETWORKS

by

Jie Lu

Scott F. Midkiff, Ph.D., Chairman

Electrical Engineering

(ABSTRACT)

This research investigates the implementation and performance of packet video transfer over local area networks. A network architecture is defined for packet video such that most of the processing is performed by the higher layers of the Open Systems Interconnection (OSI) reference model, while the lower layers provide real-time services. Implementation methods are discussed for coding schemes, including data compression, the network interface unit, and the underlying local area network (LAN), Ethernet or the Fiber Distributed Data Interface (FDDI).

Performance evaluation is presented using simulation results and analyses for different video sources, implementation models, and LANs. The simulation experiments are performed for systems where video images are retrieved from databases at one or more servers and delivered over the local area network.

Acknowledgments

I deeply thank Dr. Scott F. Midkiff for his counsel and encouragement throughout the course of this research and also for the time he took to correct the language errors of this thesis. With his knowledge and talent, he guided and helped me overcome many obstacles. I am also eternally indebted to him for giving me the opportunity to work in the area where I could not before. I would also like to express my gratitude to Dr. Nathaniel J. Davis and Dr. Ira Jacobs for consenting to be on my advisory committee.

I wish to thank Mr. Sharath Manjunath and Mr. Robert William Ford for their partial proof reading this thesis draft. They helped me when they themselves were very busy.

Words can not express my gratitude to my parents in China and brother in Bolivia, S. A. for their boundless love and endless support. Although far away in space, they are always close to me in heart. Dad and Mom's monthly call gave me courage to cope with difficulties. Brother's financial support enabled me, a spoiled girl, to smile at tomorrow. Without their love and support, this thesis would not have been possible.

I thank God for all the arrangements in my life so far and thank the people who have ever given me more or less help. Finally, I dedicate this thesis to my grandmother who is leaving us soon as a kind of consolation.

Table of Contents

List of Illustrations

List of Tables

Chapter 1. Introduction

In recent years, interest in video and image-based telecommunication services has increased significantly due to video-on-demand and video teleconference services. Today's heterogeneous networks are mixtures of circuit-switched and packet-switched networks dedicated to synchronous and asynchronous applications, respectively. In the future, however, networks will likely be based on a common fast packet-switched technology [3]. Therefore, real-time transmission of packetized video, although relatively new, is an important area of research. This research investigates the implementation and performance of packet video over local area networks (LANs). Section 1.1 presents an overview of packet video. Section 1.2 describes the objectives of this research. Section 1.3 describes the organization of the thesis.

1.1. Packet Video

Video transmission over computer networks is not currently popular since image transfer consumes a significant amount of network bandwidth. However, the need for video services, such as picturephone, teleconferencing and broadcast TV, is increasing. The interest in packet video is then motivated by the use of packet-switched communication networks to support these video services. This is due to the fact that many currently available or emerging networks are based on packet transmission mechanisms that provide benefits such as dynamically variable capacity, statistical multiplexing of traffic, and multipoint-to-multipoint operation [9]. The emergence of high speed integrated networks gives packet video a bright future.

Issues related to video coding standards and high speed networks are currently being considered by the International Consultative Committee for Telephone and Telegraph (CCITT) study group XV and XVIII [5]. The overall system performance of packet video is affected by both the video source and network properties [1, 3]. Video source coding has to be somewhat network dependent to obtain an overall performance improvement. Some coding schemes, in fact, are strongly dependent on network properties, e.g. two-layer coding [6]. Hence, how different networks support packet video transfer is an important issue in terms of the system performance.

Many coding schemes have been proposed for packet video [4, 6, 7, 8], with the most successful ones to date being variable bit rate (VBR) transform coding-based algorithms. High speed networks are also being developed to support packet video applications, e.g. asynchronous transfer mode (ATM) networks. Some simulation studies have been done for packet video transfer over local area networks such as carrier sense multiple access with collision detection (CSMA/CD) and token ring [9, 10] and over a satellite channel [8]. In work reported in [8, 9, 10], video data for a picture are encoded by the hybrid differential pulse code modulation with discrete cosine transform (DPCM/DCT) algorithm and then separated into many blocks of small sizes to be transferred over the underlying network. This scheme has at least two weaknesses. First, the packet delay does not reflect the entire picture delay. Many packets with variable lengths are needed to replenish the screen at the receiving side when a scene changes quickly and it increases the end-to-end delay due to the overhead for each packet. Secondly, one lost packet may garble the entire picture.

In this research, video images are transferred picture-by-picture to reduce the above weaknesses. Therefore, a better understanding of the special properties of packet video and different LANs can be obtained.

1.2. Objective of This Research

The objective of this research is to study and evaluate the system performance of packet video applications over local area networks (LANs). The sensitivity of system performance to different factors, including implementation models, source models, and LANs is studied through simulation. Also, a basic understanding of methods of system implementation, the functionality of components in the system, and properties of video source models and LANs are developed to compare the system performance of different cases. A clear view of the performance of packet video over two LANs, Ethernet [24] and the Fiber Distributed Data Interface (FDDI), are obtained by simulation results and analyses.

To achieve the above objective, this research includes the study of video source models, the implementation of hardware and software models for simulation, and analyses of results. Also, various techniques described in the literature to improve the performance of packet video over local area networks are surveyed.

1.3. Organization of the Thesis

Chapter 2 defines the functionality of each layer of the Open Systems Interconnection (OSI) reference model for packet video applications. Chapter 3 discusses the implementation methods, including video source coding schemes, network interface unit (NIU), and underlying networks. Also, the coordination among entities in the system is discussed. Chapter 4 describes the simulation models, including hardware components and software components for both a single source system and a multiple source system. Chapter 5 presents simulation results and analyses. A detailed comparison of system performance is given. Finally, Chapter 6 gives concluding remarks and suggestions for future work. Appendices provide details of simulation parameters and results.

Chapter 2. Network Architecture for Packet Video

This research investigates packet video from a system perspective. The most important issues in video transmission are identified and studied in the context of a layered network architecture model [2]. To illustrate the interaction between network and video processing, this chapter defines a layered network architecture for packet video. Section 2.1 discusses the overall objectives of packet video. Section 2.2 details the functionality of each layer corresponding to the OSI model.

2.1. System Objectives for Packet Video

Three objectives must be meet for effective packet video transmission: good image quality, tolerable transmission delay, and low implementation complexity. Image quality is strongly related to packet loss. Transmission delay involves both packet delay and jitter in packet delivery. Implementation complexity depends on the hardware and software components needed.

Packet loss is an important issue in video services. Since recursive schemes are usually used in video source coding, one lost picture will impair the image quality of subsequent pictures. The effect of packet loss is more serious in systems where a picture is sent as a number of packets, e.g. 53-byte cells in ATM networks. As mentioned in Chapter 1, one lost packet may garble the whole picture and consequently impair image quality. In fact, under a given packet loss rate, video services are more prone to packet loss than standard 64 kits per second (Kbits) telephony due to much higher bit rate involved. In other words, the average period between two losses is much shorter in video services than in telephony under a given packet loss rate.

Another critical issue is packet delay and jitter. A human's vision is more sensitive than his or her hearing in terms of the delay of information. Hence, intolerable packet delay and jitter will seriously affect the quality of a video service. In fact, packet loss is an extreme example of intolerable packet delay. Since pictures must be updated on the screen at the same rate as they are encoded, proper synchronization is needed between the encoder and the decoder. The decoder has to derive a stable clock. Other issues such as jitter, time reference, and elastic buffers at the decoder must also be considered [4]. These issues are also related to the type of encoding scheme adopted in a specific system.

Implementation complexity is closely related to system cost. This issue is more important in a multiple source system than in a single source system since more components are involved. Obviously, different implementation methods lead to different costs. However, there is usually a tradeoff between the quality of service and cost. In this research, two implementation schemes are compared in a multiple source system. One is a single server with multiple sources and the other uses multiple servers.

In real-time packet video applications, the three criteria discussed above are related issues. Thus, consideration of all objectives is needed in the design of a packet video system.

2.2. Functionality of Layers in the OSI Reference Model for Packet Video

To realize the overall system objectives needed for packet video addressed in Section 2.1, it is necessary to define the functionality of layers in the network architecture. By using the OSI reference model [13] as a starting point for discussion, terminology for packet video architecture is provided in [2]. The OSI reference model was not developed with real-time video transmission in mind. Therefore, most of the specific video issues are considered at the application-oriented layers, while the network-oriented layers provide general real-time service.

The OSI reference model subdivides the functions of computer communications and networking into two functional groups: the application-oriented layers and the network-oriented layers. As shown in Figure 2.1, application, presentation, session, and transport layers belong to the application-oriented group, while network, data link control and physical layers belong to the network-oriented group. The function of each layer of the OSI reference model is described below for packet video applications.

The application layer forms the user interface [13]. For packet video applications, the layer handles signal conversions such as analog-to-digital and digital-to-analog, according to the standard of the user's choice. This layer is then dependent on both the analog and digital video formats. The allowed set of formats must include all possible analog video formats, while a highly reduced set of digital video formats could be used [2].

The presentation layer is concerned with the syntax of the data being exchanged between two user application processes [13]. For packet video applications, possible functions performed at this layer include source signal separation and compression, encryption and

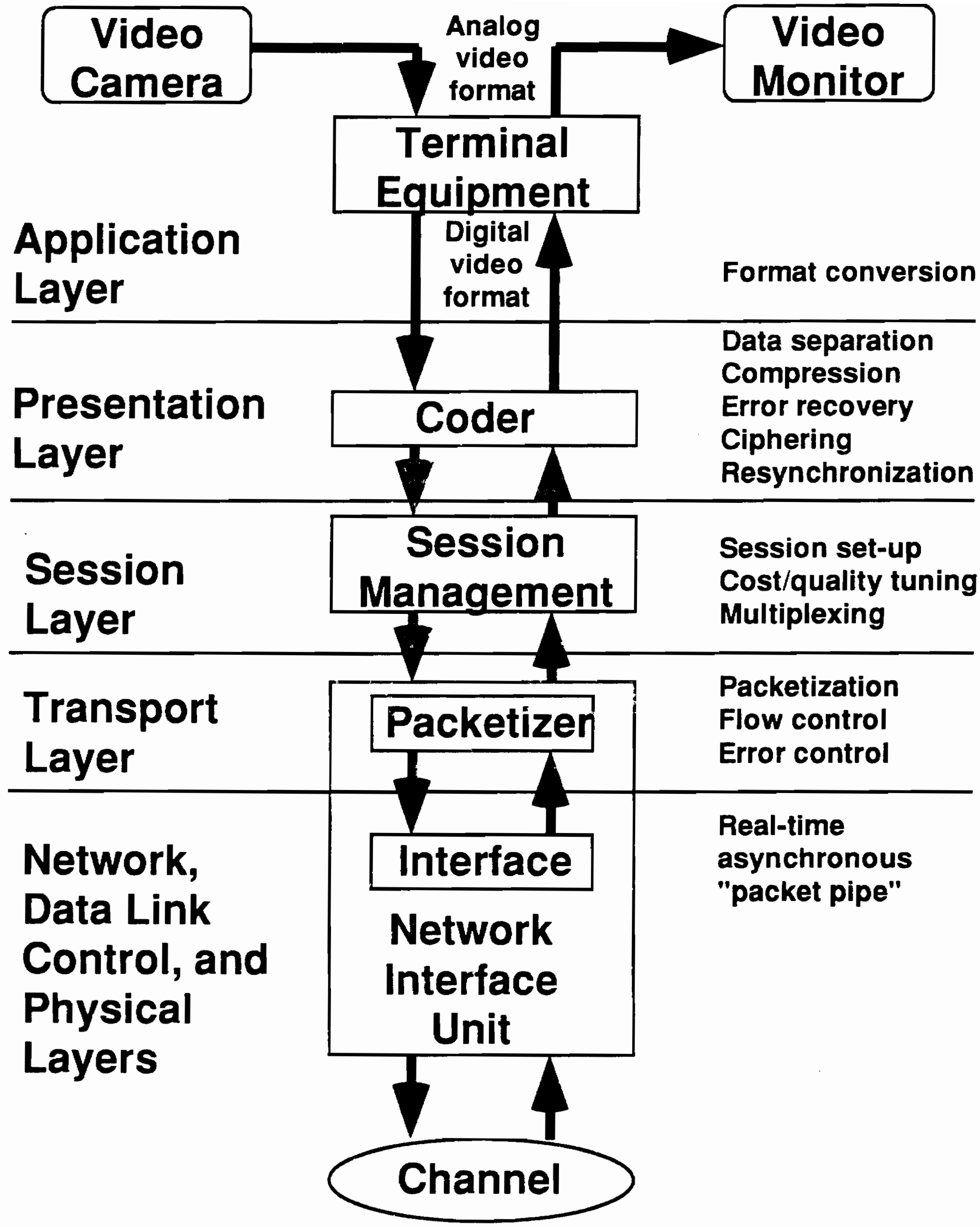


Figure 2.1. Packet video network architecture model (adapted from [2]).

decryption, error concealment, and video resynchronization. Coding and compression algorithms should balance coding effectiveness and complexity. They also must be efficient enough to limit error propagation in order to perform error concealment. Related to timing problems, resynchronization should be able to minimize packet delay jitter.

The session layer provides the means for two cooperating application processes running on different network nodes to organize, synchronize and regulate the orderly exchange of data [13]. For packet video applications, the session layer is mainly responsible for session set-up and tear-down. It should provide not only different types of sessions, but also flexibility in the quality of the sessions. Its functions are invoked only a limited number of times over an entire session. These functions should also be completely independent of the format of the video signal. Thus, a complete session of integrated real-time services, including video, voice and data, can be created at this layer due to the independence of signal format.

The transport layer provides the session layer above it with a reliable message transfer facility that is error-free and without replication. It is independent of the underlying network being used. Functions associated with this layer in packet video systems are packetization and depacketization, flow control, and error control. This layer should serve all data emanating from the video coding at the presentation layer and other associated data that have added to the session at the session layer. Each signal should be segmented independently, while the packetized signals have to be multiplexed onto the network layer. Schemes for flow control and error control should be fast enough to meet the end-to-end delay requirement.

The network layer provides end-to-end communication and shields the transport layer from the physical aspects of the transfer medium [29]. Functions at this layer include routing, congestion control and packet duplication for broadcast and multicast sessions. Through the use of congestion control, the network layer should maximize the probability of successful and timely delivery. This is especially important since the real-time requirements of packet video preclude retransmission.

The data link control layer (DLC) provides the means of transmitting data, i.e. network layer data units, over the underlying physical connection [13]. It is in charge of bit clocking, frame synchronization, and error detection. It should be noted that automatic repeat request (ARQ), a common error handling technique for the DLC layer, is unsuitable for real-time video services since it exacerbates packet delay variations. Hence, functions associated with this layer should be simplified to deal only with link-management issues.

The physical layer establishes the physical connection between the computer and network termination equipment and is concerned with the electrical and mechanical properties of this connection [13]. The requirements for this layer for packet video applications are adequate capacity and a low bit-error rate.

2.3. Summary

This chapter defined the functionality of layers in the OSI reference model for packet video applications as well as the system objectives. Generally, the lower layers, i.e. the network-oriented layers, are required to be as simple as possible to support real-time services, while the higher layers, where the video data's format and their importance to the picture's quality are known, deal with sophisticated processing of video information.

Chapter 3. Implementation Methods

Once the required functions of each layer in the OSI model have been defined for a packet video system, the question of implementation arises. There is no unique method to implement the system since various networks and coding schemes are already in use. An end-to-end design is needed for a particular system to realize specific goals. However, there are many important underlying issues common to all packet video systems regardless of their implementation. Section 3.1 presents an overview of the packet video system structure. Section 3.2 investigates various coding schemes. Sections 3.3 and 3.4 address the implementation of the network interface unit and the underlying LAN. Finally, Section 3.5 assesses the cooperation among entities in the system.

3.1. System Overview

Figure 3.1 shows the generic structure of a packet video system. The server and user each represent the application, presentation and session layers in the OSI model. The network interface unit (NIU) comprises functions of the transport, network and data link control layers. The underlying network is required to have enough bandwidth for supporting packet video applications.

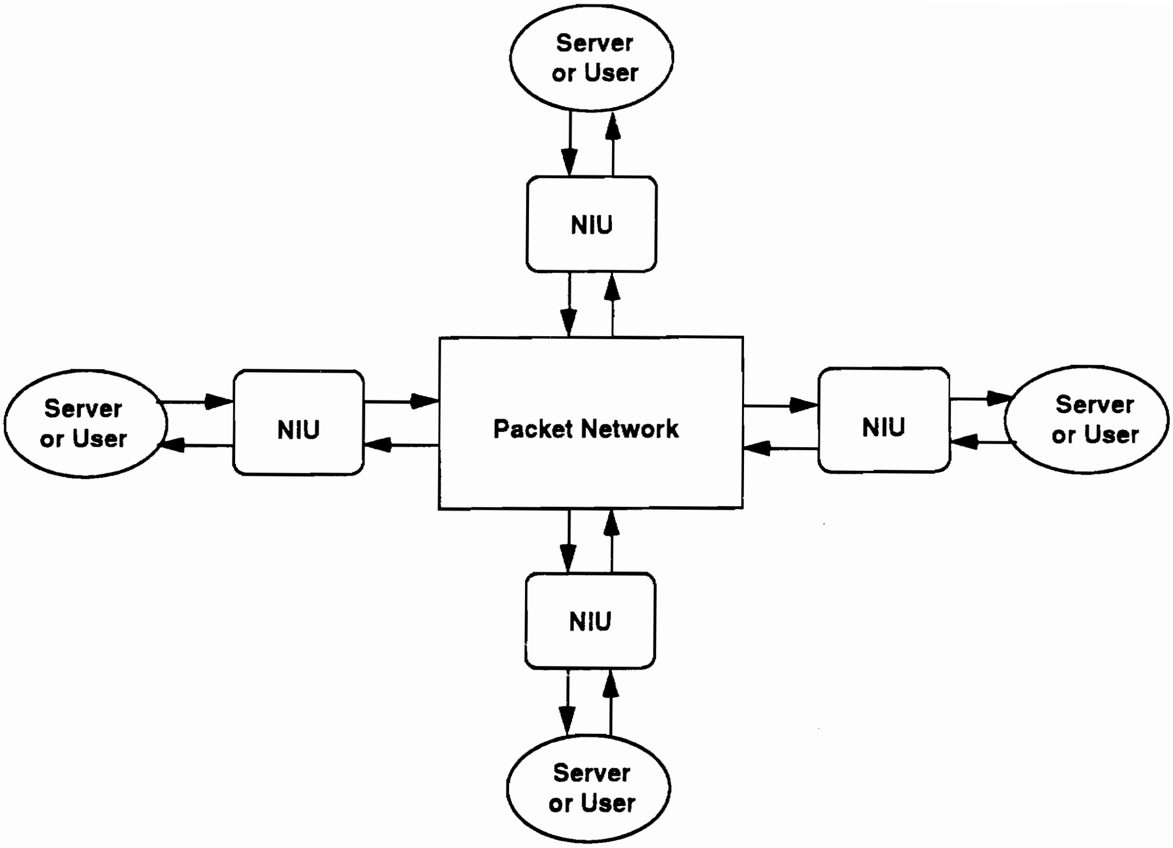


Figure 3.1. Schematic representation of a packet video system [8,10].

In this system, the server is in charge of video source coding and related data compression. The user is responsible for data decompression and decoding. The coding scheme should be designed to transmit the least amount of data with no loss in quality. For expressly this purpose, the variable bit-rate (VBR) coding scheme is widely accepted. A number of algorithms have been proposed [6, 18, 20], and some of them such as hybrid DPCM coding and sub-band coding have proven to be fairly effective.

The NIU connects the video source encoder (server) or decoder (user) to the link-level interface of the underlying LAN. Its function is consistent with the class of services normally associated with transport protocols and with that of an interface between the encoder/decoder and the LAN. One NIU can be connected to multiple video encoders to form a single server with multiple sources, if desired. The simulation model of this scheme is described in Chapter 4 and an evaluation of its performance is given in Chapter 5.

The underlying network is less flexible than the other components in the system. It is more realistic to implement packet video applications using existing networks rather than trying to design a whole new network to meet the specific requirements of packet video. The LAN's bandwidth is a key factor that will affect the quality of the video services. The general properties of a LAN will also determine the best type of coding scheme to be adopted.

3.2. VBR Coding

As mentioned in Chapter 1, it is variable bit-rate (VBR) coding that makes packet video services feasible over networks in which bandwidth is constrained. By using a VBR coding scheme, the transmission bandwidth requirement can be reduced to a range of 0.384 to 2.048 megabits per second (Mbps) for teleconference video and 15 to 30 Mbps for broadcast video, while uncompressed pulse coding modulation (PCM) video requires a transmission capacity of 75 to 100 Mbps [8]. This fact leads to a compression ratio of 2.5 to 260.417. In contrast to conventional constant bit-rate (CBR) coding where codecs transmit information at a fixed bit rate, VBR coding allows the bit-rate to vary according to the amount of information contained in an image [17]. Image quality in a CBR system must suffer in areas of high activity, while an unnecessary amount of information is often transmitted during non-active periods. In a VBR coding system, the bit rate will be reduced when only a small amount of information is needed and more information is transmitted during periods of increased activity. As a result, VBR coding produces more consistent image quality and more bursty traffic than CBR coding, although the average bit rate may be the same.

There are many VBR coding schemes in use today, and most are variations of hybrid transform algorithms [18]. They combine a mathematical transformation with minimum redundancy coding to produce near-optimum statistical compression. A general VBR coding diagram from [17] is shown in Figure 3.2. The original spatial image is divided into smaller sub-blocks of pixels to obtain equal-sized blocks of transform coefficients in the frequency domain. A threshold is applied and the coefficients are quantized to remove much of the subjectively redundant data from the image. The blocks of remaining coefficients are scanned and coded into binary form as output. A detailed description of this algorithm is given in [19]. The following sections describe three VBR coding schemes that are currently popular.

3.2.1. Hybrid DPCM/DCT Coding

Based on preliminary reports of CCITT study Group XV, the hybrid DPCM/DCT coding system is widely accepted as a practical coder [6, 9, 20, 21]. The algorithm used for video source encoding is based on a motion compensated intra/interframe discrete cosine transform (DCT). A detailed description can be found in [20]. Figure 3.3 shows a simplified flow diagram of a hybrid DPCM/DCT encoder. The incoming image is partitioned into non-overlapping blocks of $N \times N$ pixels. Two DCTs, a forward one and an inverse one, are located in the coding loop. In a similar fashion as the DCTs, two uniform quantizers are used to quantize the coefficients of each image block that are scanned later in a zigzag pattern. As a generic structure of the configuration, a simple differential pulse coding modulation (DPCM) loop is applied in the temporal dimension. A data unit memory is included in the loop to contain the previous reconstructed data unit. The selection between interframe and intraframe coding is on the basis of the expected entropy of the output data stream.

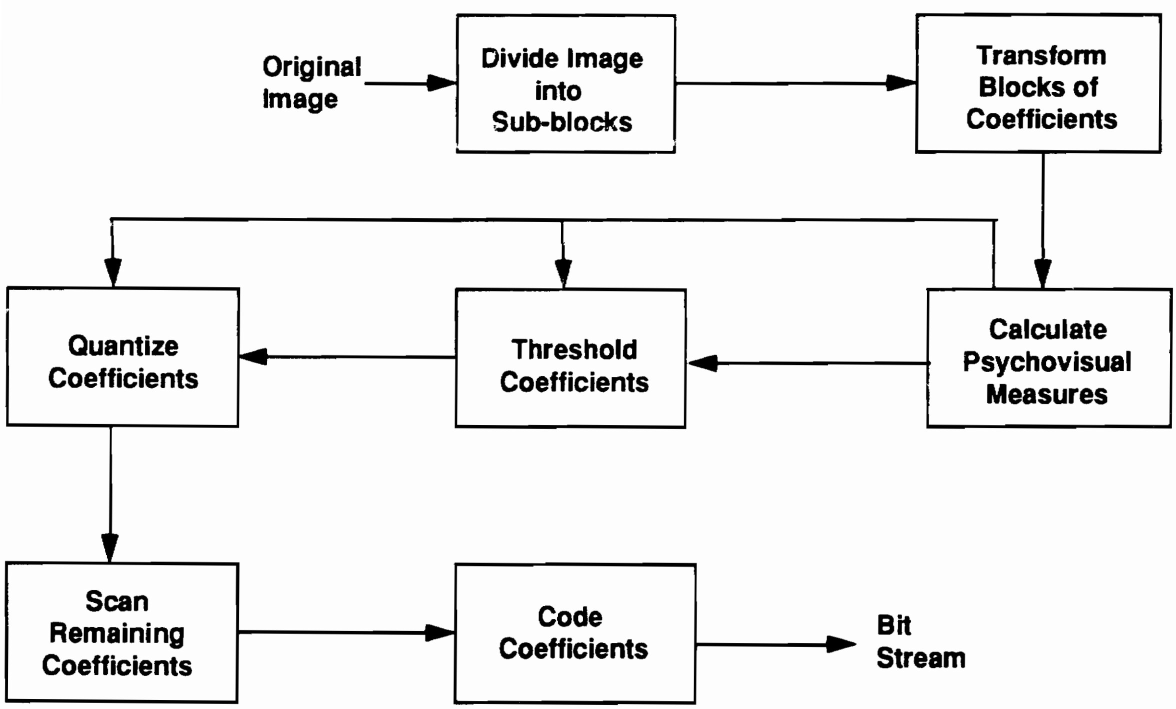


Figure 3.2. General VBR coding diagram [17].

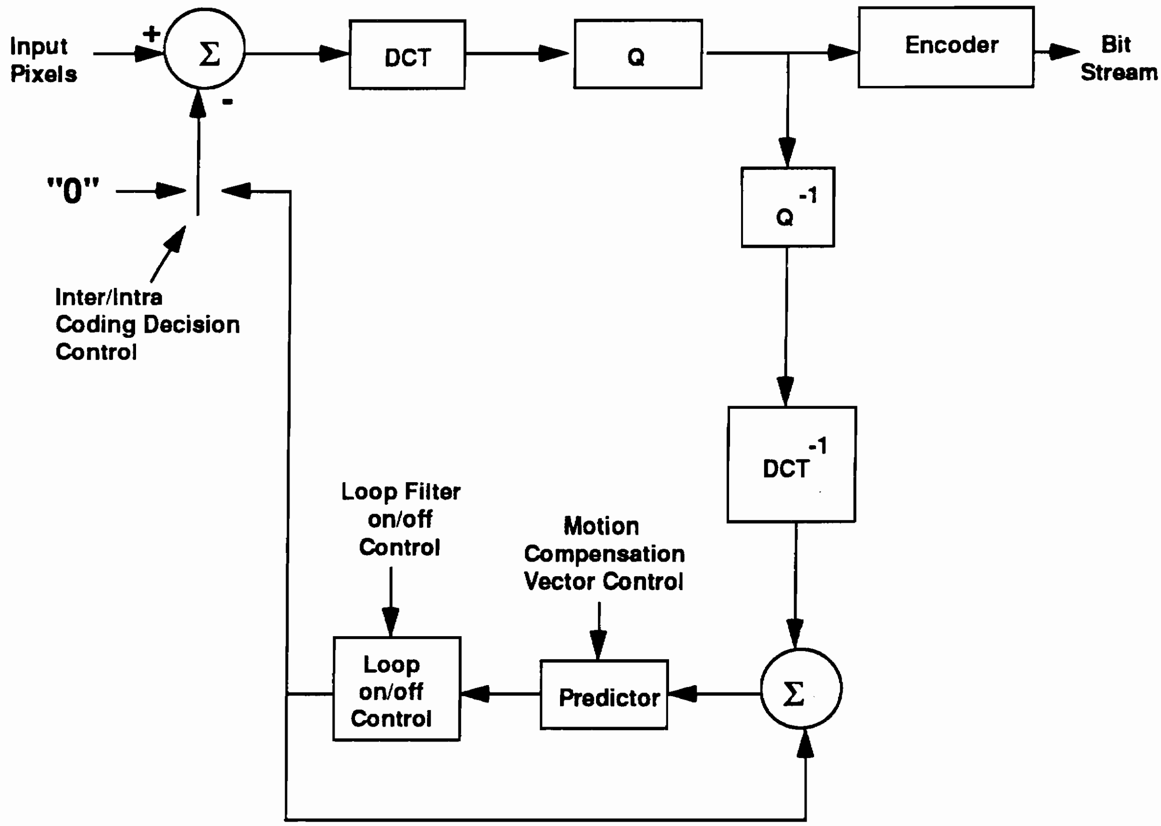


Figure 3.3. Simplified diagram of the hybrid DPCM/DCT coding scheme [9].

3.2.2. Subband Coding

Subband coding is a hierarchical coding scheme. As shown in Figure 3.4, the image data are separated into sub-data of various importance. These sub-data may be coded and transmitted independently of each other. After reception and decoding, the sub-data are recombined to form the output signal [2]. The general requirements for this method are that the error propagation is strictly limited and the locations of the lost values are known. Ideally, data separation and recombination should be lossless and should not increase the amount of data that needs to be coded. The coding method used for subband data should meet these objectives.

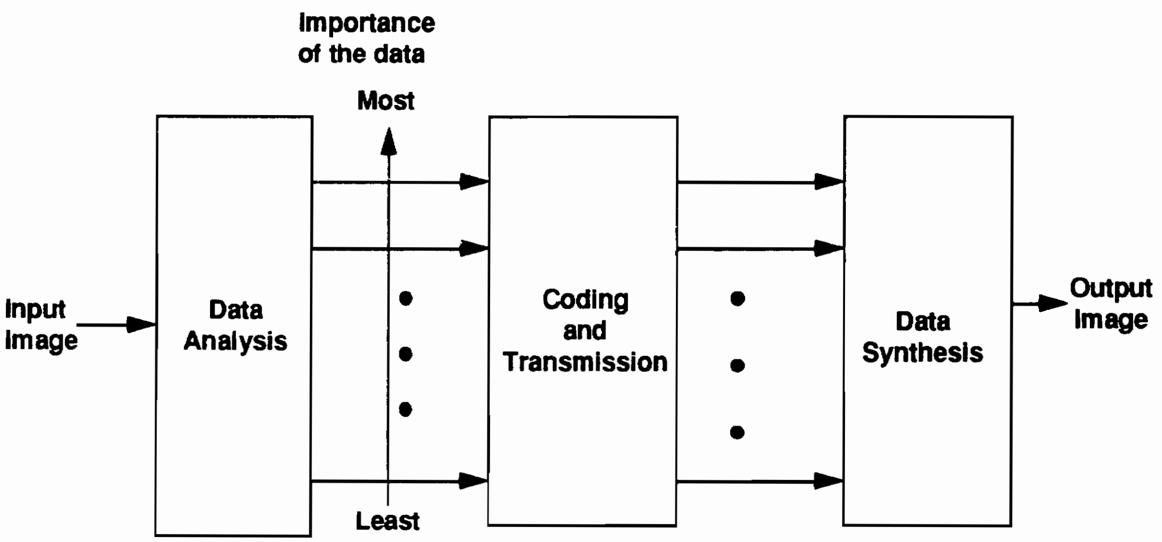


Figure 3.4. Block diagram of subband coding [2].

A sample implementation of subband coding is addressed in [2]. As illustrated in Figure 3.5, the frequency spectrum in all three dimensions, i.e. temporal, vertical and horizontal, is split along the midpoints of each frequency axis into 11 three-dimensional regions. The one that contains both low temporal and low spatial frequencies is split into four spatial frequency regions. The subbands are obtained by sub-sampling the data in each dimension at the new Nyquist frequencies. In [2], sub-band 1 is encoded with first-order one-dimensional DPCM since it retains a high variance of its intensity distribution after low-pass filtering in all three dimensions. All the other bands are encoded by PCM due to their greatly reduced variance.

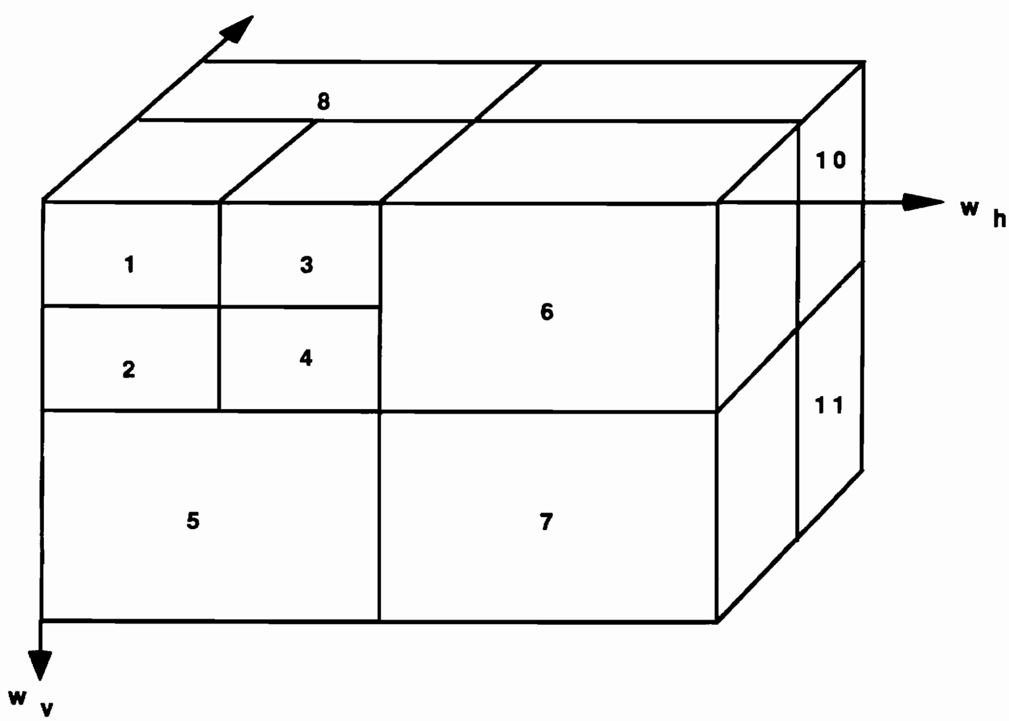


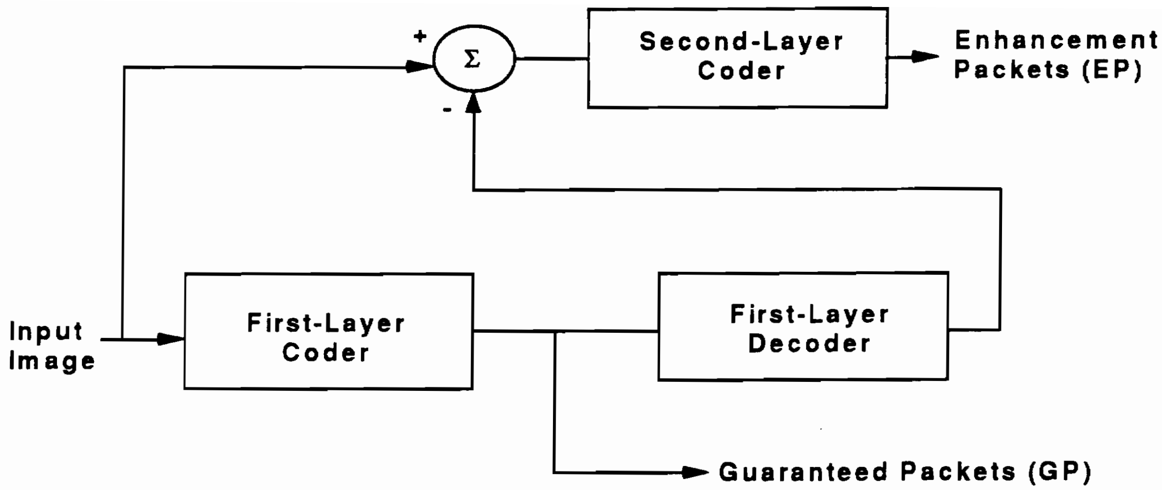
Figure 3.5. The 11 frequency regions of the subband analysis [2].

3.2.3. Two-layer Coding

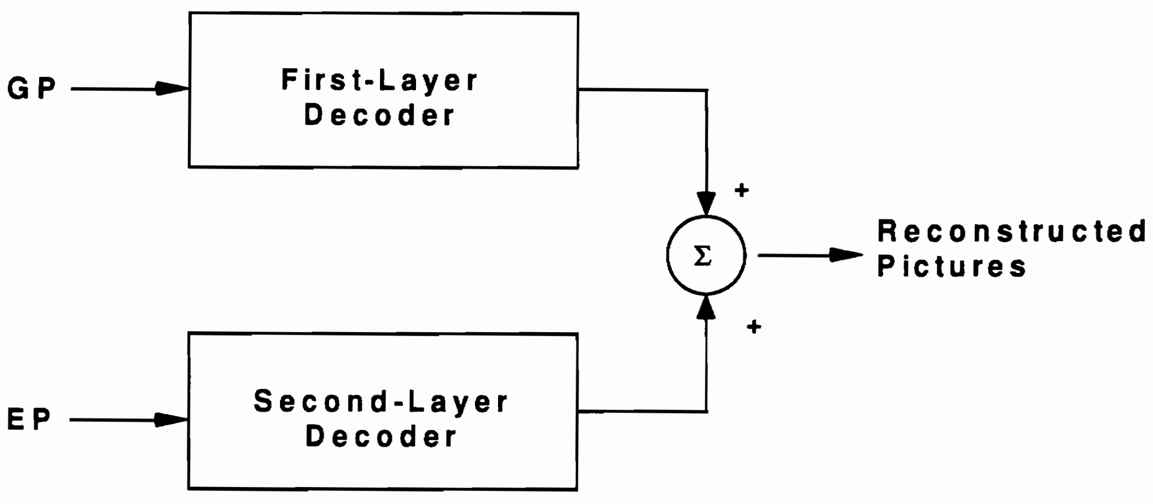
The two-layer coding is designed for use in networks that can carry integrated services, e.g. the "Orwell" ring [6] and asynchronous transfer mode (ATM) networks. The two-layer codec produces two output bit streams. The first bit stream contains all the important structural information in the image and is accommodated in the guaranteed capacity of the network, while the second one adds information to enhance image quality.

Figure 3.6 shows the block diagram of a two-layer coder and decoder. The first coding layer generates guaranteed packets that contain mainly vital information such as synchronization pulses and changed addresses, together with essential video data. This is due to the fact that the receiver has to reconstruct the picture by relying primarily on these packets, even if with low quality. A hybrid DPCM/DCT coder is used as the first layer coder in [6] where implementation issues such as motion detection, motion compensation, coding loop, and overhead information are discussed in detail.

To obtain good image quality, the second coding layer produces enhancement packets that contain the difference between the input and the decoded output of the first layer. These packets are labeled "low priority" when transferred over a network which allows a priority assignment. Their loss should not affect the tracking of the reconstructed picture. However, the second coding layer handles the fine picture detail. The DPCM algorithm is employed in the second layer coder to deal with slow scene changes. Issues such as overhead information and protection from error accumulation are also addressed in detail in [2].



(a) Encoder.



(b) Decoder.

Figure 3.6. Block diagram of a two-layer coding [6].

There have been many other VBR coding schemes proposed to improve image quality or reduce the amount of data needed without affecting the image quality [1, 7, 22, 23]. To achieve good overall performance, the decision to adopt a certain coding scheme must be influenced by the properties of the underlying LAN.

3.3. Network Interface Unit (NIU)

Usually, an existing network is used for packet video applications. Its link level access protocol does not automatically support the special requirements of compressed packet video. Therefore, the transport level of a NIU has to take certain measures. Its protocol has to be customized to meet the requirements of the video encoder/decoder and underlying network. Figure 3.7 shows a functional implementation of the transport level, link level and physical level of the NIU [8, 9, 10]. The NIU on the transmission side is called the server NIU (SNIU) and the one on the receiving side is called the user NIU (UNIU). The link level and physical level of the NIU merely function as an interface to the underlying LAN, while the transport level of the NIU plays an important role in determining the performance of a packet video application. However, in this research, the output buffer of the UNIU in Figure 3.7 is moved to the user (the video decoder) as an input buffer. It is more convenient and much safer for the user to read data from local memory rather than from the UNIU by an internal bus. Thus, the user takes care of resynchronization to reduce the burden on the UNIU.

Basic transport level functions of the NIU include packetization, flow control, error control and concealment, and packet resequencing, if necessary [8,9,10]. In this research, since a whole data unit is used as a transmit packet, packet reseqencing may not be a critical issue. Note that each packet may be broken into many blocks suitable for transmission over the underlying network. Also, packet delay compensation is assumed to be handled by the high layers at the receiving side. Some of other basic functions of the NIU are discussed briefly in the following three sections.

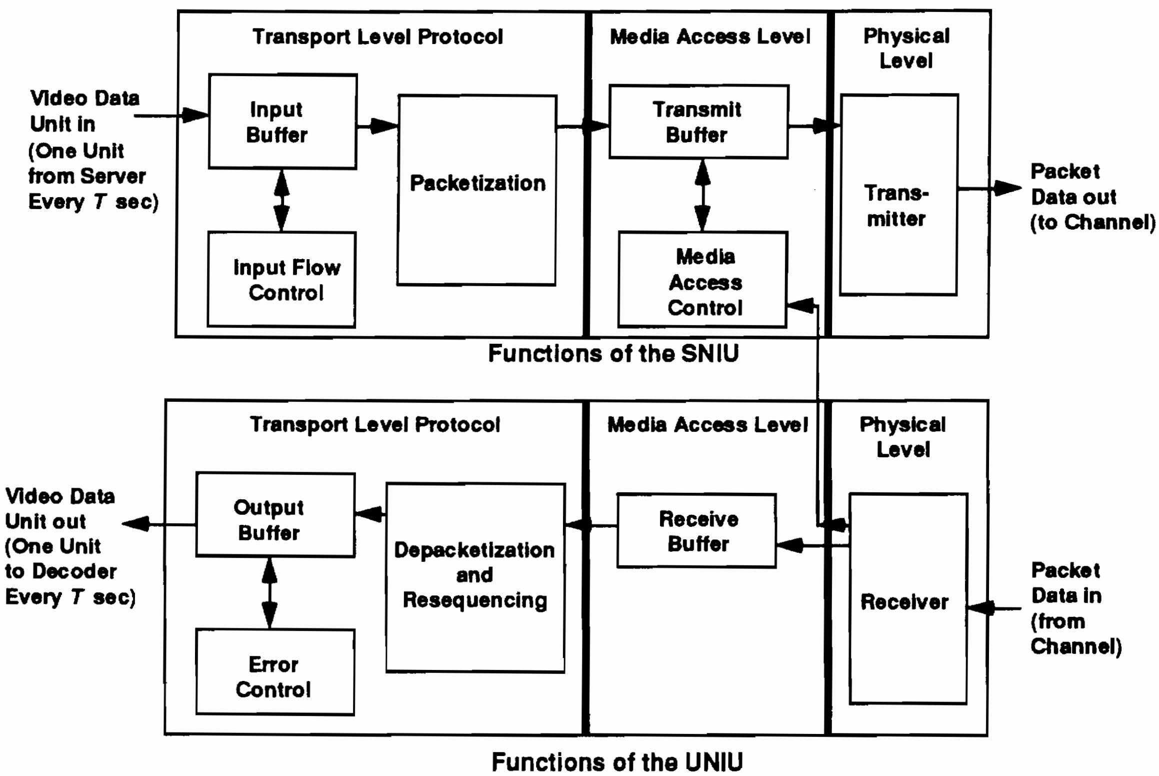


Figure 3.7. Functional implementation of the NIU (adapted from [9]).

3.3.1. Packetization

The SNIU accepts variable length units of video information from a single server or multiple servers in the case of a single server with multiple sources, every T seconds. Data units used in the work reported in [8, 9, 10] are lines or small blocks of a picture. In this research, a whole frame, a video picture, is used as a data unit due to the problems of small data units addressed in Chapter 1. Before the video data enters the underlying LAN, it is necessary to form transport packets that are suitable for transmission. The determination of transport packet size is a tradeoff between packet delay and packet loss in a given network. Large packets result in less delay for the whole frame due to reduced overhead for each individual packet, but may result in inferior quality due to packet loss. Small packet size has an opposite effect. However, frame delay is more critical than packet loss, since a large delay will also affect picture quality. In this research, the User Datagram Protocol (UDP) [32] is used as the transport protocol. Providing connectionless services, UDP requires the least amount of overhead for each packet and can deliver packets quickly in a local area network. UDP does not offer guaranteed packet delivery as does the Transport Control Protocol (TCP) [32]. Thus, 1472 bytes is a proper size for transport level packets in Ethernet [12], while 4452 bytes is a good size in FDDI. This is due to the fact that the maximum frame size in Ethernet is 1518 bytes of which 46 bytes are reserved for overhead, including a 20-byte Internet Protocol (IP) header. For FDDI, the maximum frame size is 4500 bytes on the LAN of which 48 bytes are reserved for overhead.

3.3.2. Flow and Congestion Control

Under a traffic overload condition, video packets will encounter unacceptable delay if they do not have a high transmission priority. Thus, flow control and congestion control are important functions of the transport layer in this system. Common data networks usually employ an input buffer limit (IBL) scheme. When the input buffer is full, no new data packets from the source are accepted. However, for packet video applications it is better to discard the oldest data packet in the input buffer of SNIU rather than reject the new one. It might be too late for the oldest data to reach the receiving end due to the strict delay requirement.

A better strategy is to have the encoder keep track of the input buffer of the SNIU and adjust its encoding speed slightly when the buffer is full. This strategy avoids error propagation due to dropping packets if interframe coding is used.

With a two-layer coding scheme, flow control and congestion control become much simpler. When the input buffer of the SNIU is full, enhancement packets generated by the second coding layer can be discarded without seriously affecting picture quality.

3.3.3. Error Control and Concealment

There are several scenarios that can cause packet loss, e.g. channel errors and transport buffer overflow [9]. Channel errors are detected by a cyclic redundancy check (CRC) at the link level. The common automatic repeat request (ARQ) mechanism might not work well with real-time packet video applications due to the delay problem associated with retransmission. This condition may be treated the same as packet loss and error concealment will be needed at the receiving end. Although not shown in Figure 3.7, ARQ logic is still needed at the UNIU to coordinate with other nodes that transmit non-video information. Transport buffer overflow is known at the transmission end. The encoder may attempt recovery action, if possible. Otherwise, error concealment will also be needed at the receiving end.

If the locations of lost data in a picture are known, error concealment can be realized by interpolation within the video codec, for example inserting null-bit units temporally or holding the previous picture until the next refresh. [2, 9]. The best strategy depends on the compression algorithm used for the video source coding. Error propagation due to packet loss has to be strictly limited when recursive coding methods are used.

3.4. Local Area Networks

Although the underlying network in Figure 3.1 can be any packet network with sufficient bandwidth, two existing local area networks, Ethernet and FDDI, are considered in this research. As mentioned previously, the specific properties of the network control the overall performance of packet video. Therefore, it is necessary to discuss the two LANs in order to understand system performance.

3.4.1. Ethernet

Ethernet is a 10 megabit per second (Mbps) branching broadcast communication system for carrying digital data packets among locally distributed computing stations [24]. It became the IEEE 802.3 standard in 1983 and was adopted as ANSI standard 802.3 in

1984. As shown in Figure 3.8, IEEE 802.3 spans the media access control (MAC) layer and physical layer. The MAC layer, which is not in the OSI model, has the functions of controlling access to the medium, framing, address recognition, and error checking [25].

Carrier-sense multiple-access with collision detection (CSMA/CD), which is a multi-access reservation scheme suitable for networks with relatively small propagation delay, is used in Ethernet. Figure 3.9 gives an example of the operation of a CSMA/CD bus. Each node on the Ethernet listens to the medium to find out whether any transmission is in progress. If idle is detected, transmission starts and the medium is reserved for the remainder of the frame transmission time, assuming no collision. However, if a collision occurs due to more than one node trying to transmit a frame simultaneously, the reservation fails and another attempt must be made at a later time. The retransmission interval is dynamically adjusted based on the actual traffic load [24,25].

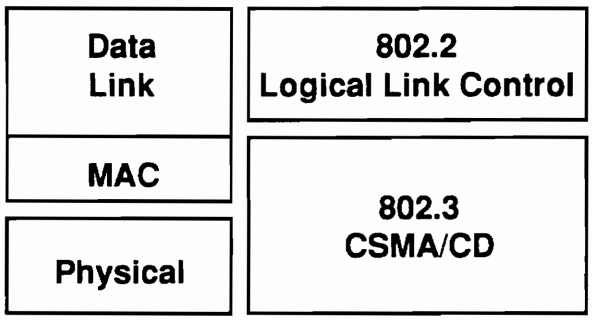


Figure 3.8. IEEE 802.3 standard layers [25].

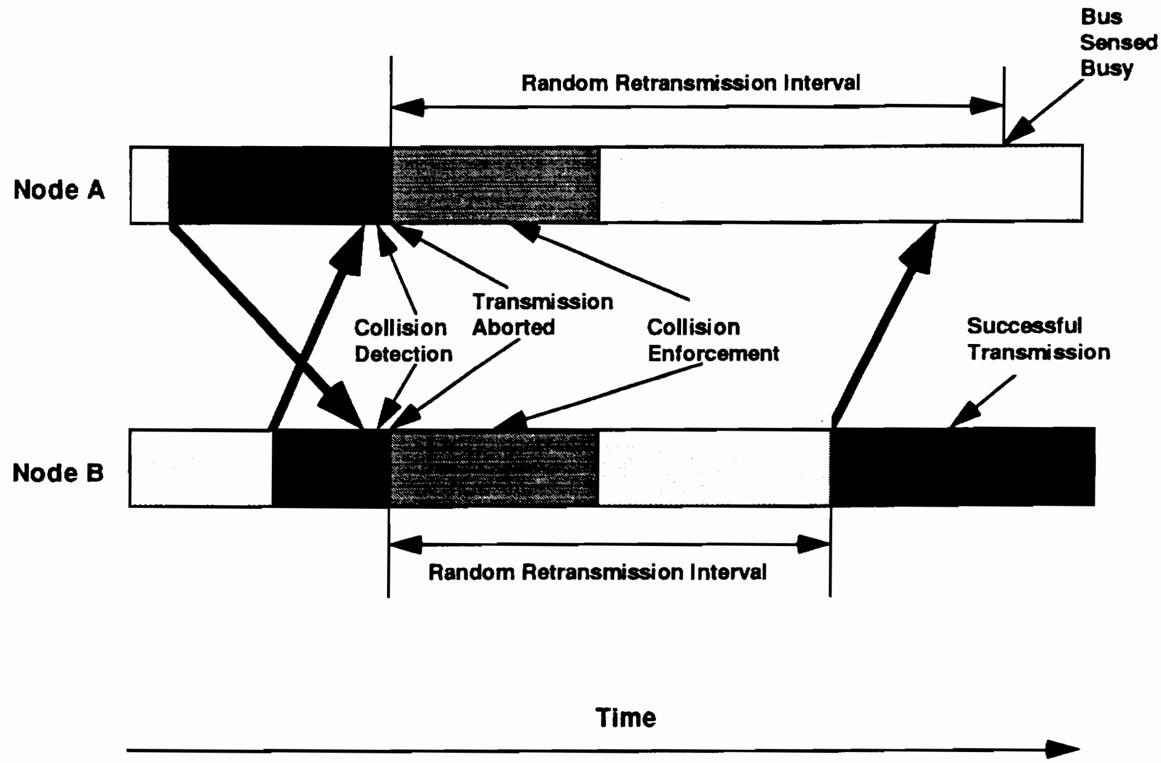


Figure 3.9. Operation of CSMA/CD bus (adapted from [24]).

There is no priority assignment in Ethernet. Frames carrying video packets must compete with other traffic. This may delay the delivery of video packets in the event of heavy traffic such that the delay requirement may not be met. This is a disadvantage of Ethernet for packet video applications. One of the objectives of this research is to determine a range of background traffic for packet video.

3.4.2. FDDI

The fiber distributed data interface (FDDI) is a 100 Mbps timed token ring network based on fiber optics [31]. As shown in Figure 3.10, the FDDI standard defines layers including media access control (MAC), physical (PHY), physical media dependent (PMD), and station management (SMT). The MAC layer is in charge of frame construction, addressing, token handling, and frame error detection. The PHY layer takes care of encoding and decoding, clocking, and framing. The PMD layer defines physical connectors, cable types, the electro-optic interface, and bit error rates. Finally, the SMT layer is responsible for connection management, ring management, and frame services [25].

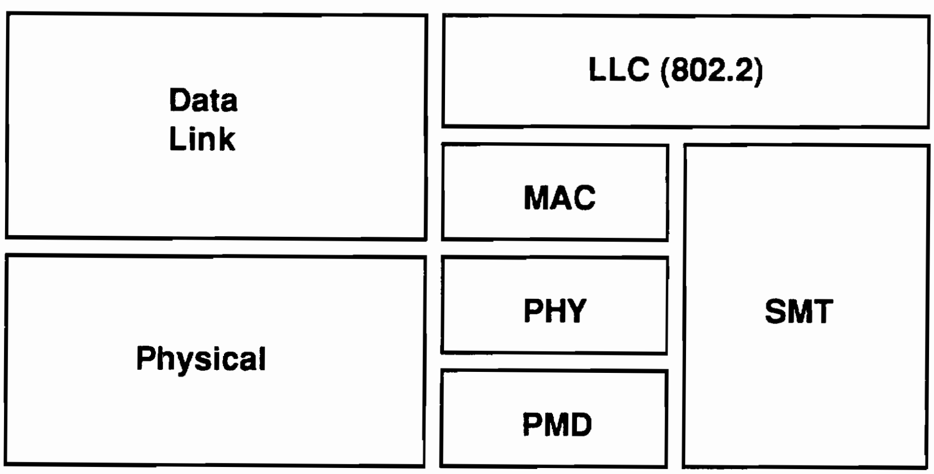


Figure 3.10. FDDI standard layers [25].

A priority token ring protocol is employed in FDDI. Access to the transmission channel is controlled by passing a permission token around the ring. When the system is initialized, a free token that is generated by a designated node travels around the ring until a node is ready to put its frame onto the ring. The sending node is responsible for removing its own frame from the ring. At the end of its transmission, the sending node passes the free token to the next node. Figure 3.11 illustrates this operation for a ring with four nodes. It shows how nodes 1 and 3 and again node 1 subsequently access the ring to transmit frames.

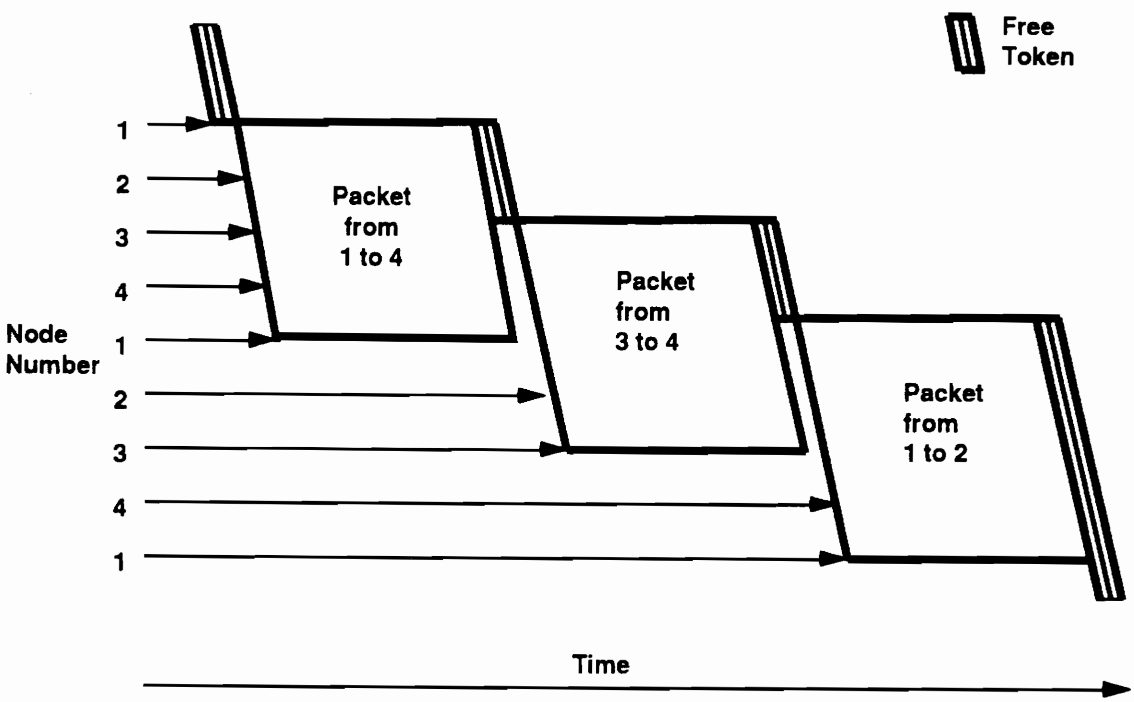


Figure 3.11. An example of token ring operation with four stations (adapted from [24]).

FDDI supports low-priority, or asynchronous, traffic and high-priority, or synchronous, traffic. The capacity allocation is provided by a timed token scheme [25]. Each node measures the time between token arrivals so that low-priority frames can be sent only if the inter-token time is sufficiently small, while high-priority frames can be sent anytime the token arrives. For packet video applications, frames carrying video packets can be designated as high-priority traffic to ensure they are transmitted whenever the token arrives, while non-video traffic can be designated as low-priority traffic if its delay is not critical.

3.5. Coordination Among Entities in the Packet Video System

As discussed above, there are a variety of existing coding schemes and networks. To design an effective packet video system, coordination among entities is critical. Coordination between peer entities, between the coding scheme and the underlying network, and between the encoder and the SNIU are discussed below.

As shown in Figure 3.1, the server (encoder) and the user (decoder) are peer entities. The SNIU and the UNIU are also peer entities. From their functionality addressed in the previous sections, it is clear that their coordination is essential to system performance and complexity.

The adoption of a coding scheme according to the type of available network is also an important issue with respect to the overall performance. For example, a coding scheme that requires the least amount of data to keep a desired quality should be chosen when using Ethernet, since Ethernet has limited bandwidth. However, a system based on FDDI has more flexibility due to its higher bandwidth and priority scheme. Sub-band coding and two-layer coding may be used in FDDI. To take advantage of the coding scheme, frames containing the most important video data packets can be given high-priority and frames carrying enhancement packets can be given low-priority. Therefore, pictures can still be reconstructed in the case of delay or loss of enhancement packets.

Coordination may also be needed between flow control and error control and between the encoder and the SNIU. As mentioned in Sections 3.3.2 and 3.3.3, the encoder can adjust its processing speed or retransmit the frame which is in error or is dropped, according to information feedback by the SNIU.

3.6. Summary

This chapter described the various methods of implementing components in the packet video system as well as the coordination needed among these components. A tradeoff exists for each entity in the system. The tradeoff for a coding scheme is using less bandwidth versus having the best picture quality. The tradeoff for the underlying network is quick packet delivery versus packet loss or error. Finally the NIU has a tradeoff between effectiveness and lower complexity. Therefore, careful consideration of components and their interaction is important to the overall system performance and simplicity of implementation .

Chapter 4. Simulation Models

A network simulator, CACI Products Company's NETWORK II.5 [11], is used to simulate the packet video systems. This chapter describes the hardware and software models and the video source models. Hardware and software components of networks are defined by parameters required by the simulator. Both single source and multiple source systems are considered.

Section 4.1 briefly describes NETWORK II.5. Section 4.2 discusses the two video source models used in this research for teleconference video and broadcast video. Section 4.3 describes hardware and software components in a single source system. Section 4.4 addresses hardware and software components in multiple source systems.

4.1. NETWORK II.5

As mentioned in [12] and detailed in [11], NETWORK II.5 is a simulation tool for evaluating computer networks. It defines hardware and software components for a network. Hardware components include processing elements, transfer devices and storage devices, while software components include modules, instruction mixes, and statistical distributions. Each entity is characterized by a number of parameters. Appendix A gives parameter values and explanations. A full description of the parameters is given in [11]. Statistical results, such as link-level delay, end-to-end delay, system throughput and packet buffering time, can be obtained from summary reports produced by NETWORK II.5.

4.2. Video Source Models

An accurate video source model is important to system performance evaluation. As mentioned in Chapter 1, different coding schemes can be used for different LANs to achieve the best performance. However, this research uses the same coding scheme in both Ethernet and FDDI for the purposes of comparison. The following two sections describe video source models for teleconference video and broadcast video applications.

4.2.1. Teleconference Video Source Model

The teleconference video source model differs among several papers [5, 3, 14]. Different coding schemes cause different statistical distributions for the length of video data units from the encoder, even though their inter-unit rates are the same. Typical statistical distributions for the length of video data units are normal, exponential and gamma. The teleconference video source model derived in [5] is adopted in this research. It is supported by a fairly long (30-minute) sequence of real data, while most other models use short sequences of data, e.g. a few seconds long.

PAL standard data, coded using a hybrid DCT/DPCM coding scheme with no motion compensation, are used in [5]. The inter-unit period is 40 ms, i.e. a video data unit is generated every 40 ms by a server. Figure 4.1 shows the histogram for the length of data units. Since the measured data from [5] is in an Asynchronous Transfer Mode (ATM) network environment, all parameters are in units of cells that are given as 64 bytes in length. In units of bits, the mean and standard deviation of the data length are 66,712 bits and 38,098 bits, respectively. The lower bound is 12,800 bits and the upper bound is 320,000 bits.

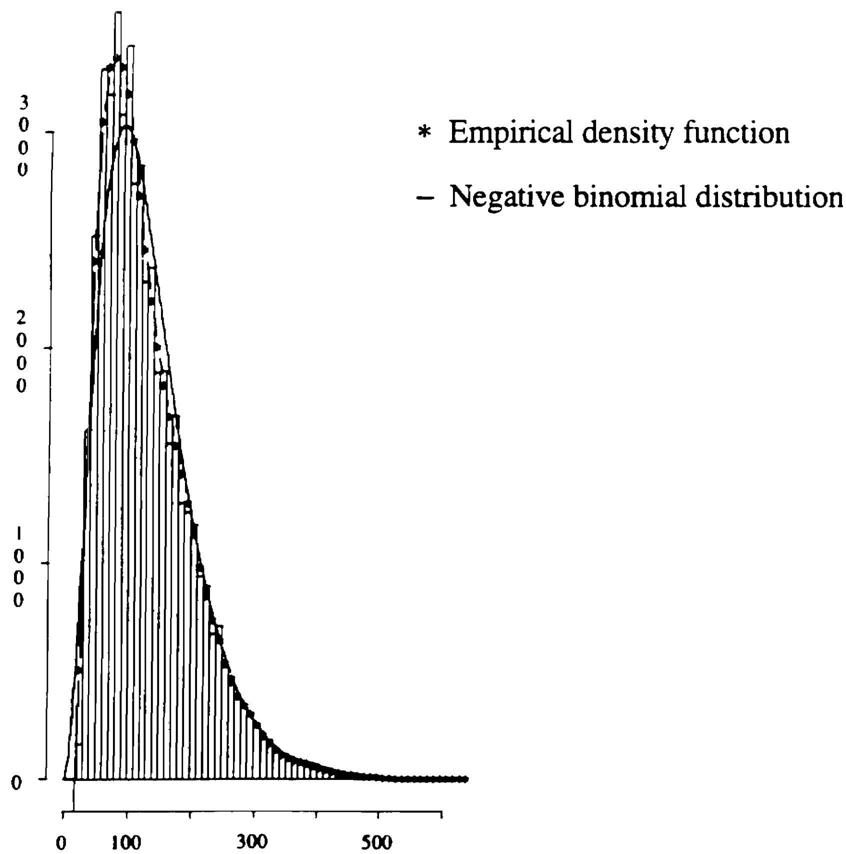


Figure 4.1. Histogram of the length of data units (in cells) for teleconference video [5].

As discussed in [5], the length of data units for teleconference video applications follows a statistical distribution that is a combination of gamma and exponential. It follows a gamma distribution when data units are less than 204,800 bits in length and fits an exponential distribution well when data units are greater than 204,800 bits in length. Statistically, with 99.4 percent probability, the length of video data units follows a gamma distribution and fits an exponential distribution with the remaining 0.6 percent probability. In both cases, the length of data units has a mean of 66,712 bits and a standard deviation of 38,098 bits. Based on the mean length of data units and the transmission rate of 25 data units per second, the average bandwidth requirement of the traffic is 1.67 Mbps.

In the simulation using NETWORK II.5, an instruction mix is used to simulate the source model addressed above. Instruction mixes are "pseudo-instructions" included in the instruction list of a module with a percentage associated with each. Two instructions are listed in this instruction mix both of which send video data units at a fixed iteration period of 40 ms. 99.4 percent probability is assigned to the one that sends video data with the length of data units following the gamma distribution, while 0.6 percent probability is assigned to the other that sends video data with the length of data units fitting the exponential distribution. Whenever the module that sends video data is invoked, one of the two instructions is randomly chosen to be executed. The number of executions corresponds to the probability assignment.

4.2.2. Broadcast Video Source Model

The source model for broadcast video is consistent among several papers [4, 6, 16]. It is commonly acknowledged that the length of video data units from the encoder for broadcast video applications follows a normal distribution. Figure 4.2 shows the bit rate probability density distribution for three hours of video broadcast captured from a CATV network [4]. The inter-unit period is 40 ms, as in teleconference video. The average bit rate is 16.8 Mbps. The peak bit rate is 44.7 Mbps. Therefore, in units of bits, the mean length of data units is 672,000 bits, while the standard deviation of the length is approximately 973,030 bits. This is derived from the bit rate probability density function shown in Figure 4.2. The lower bound of the length is assumed to be 12,800 bits, which is the same as that of teleconference video, and its upper bound is 1,788,000 bits [4].

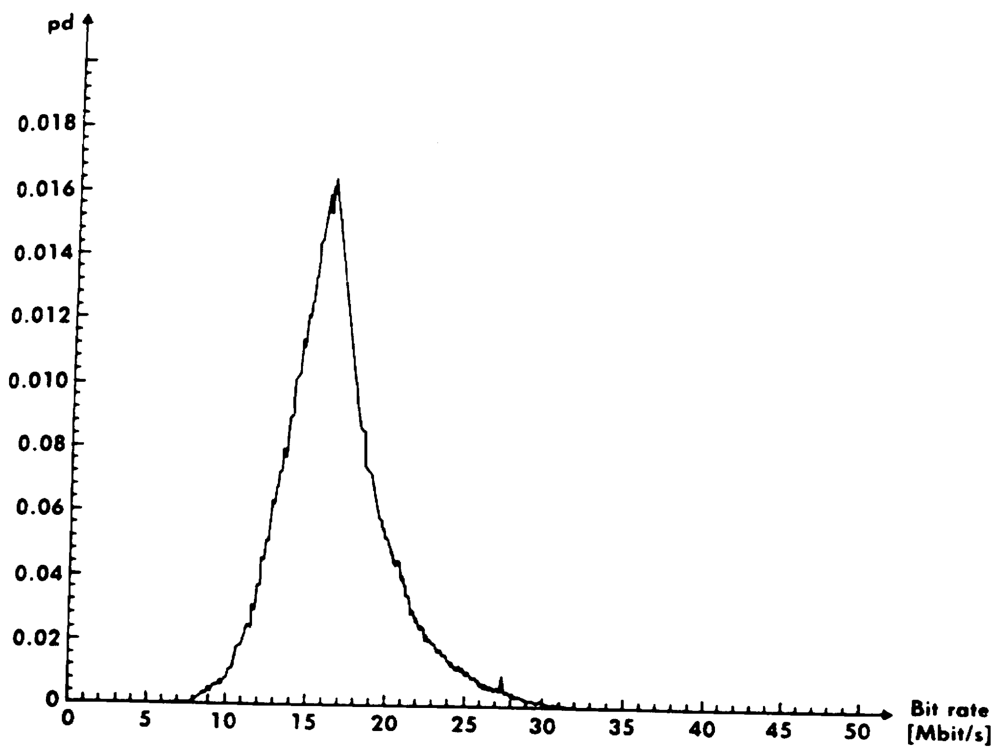


Figure 4.2. Bit rate probability density function of video broadcast scenes [4].

The length of data units in the broadcast video source model has a much higher mean and peak rate than in the case of the teleconference video. This is due to the fact that there is more movement and the scene movement is more unpredictable for broadcasting video than for teleconference video, resulting in more random data. Therefore, a normal distribution is a reasonable model for broadcasting video.

In the simulation using NETWORK II.5, a message type instruction is used to simulate the source model for broadcast video. Like teleconference video, the iteration period of this instruction is 40 ms since the module that sends video data units is invoked every 40 ms. The message length that represents the length of video data units differs from teleconference video and follows the normal distribution as indicated above.

4.3. System Models

Besides video source models, network system models are also important to packet video systems. There are two system models used in this research: single source and multiple source. The single source model is concerned with an underlying network through which video packets from a single server can be properly transferred without seriously affecting other traffic. Multiple source models also involve competition among video sources, as well as with non-video traffic.

4.3.1. Single Source System

A single source system has only a single video source. However, if two or more users request video services simultaneously, multiple connections may be set up at a given time by broadcasting. The server works on a first-come first-served (FCFS) basis. In some cases, certain users may have higher priority than others when accessing the video service. Since users rarely request the video service simultaneously, this study assumes that only one video service connection is invoked at any time, with one server and one user at each end of the LAN. Hardware and software components of the single source system model are discussed below.

4.3.1.1. Hardware Components

Figure 4.3 shows the structure of the single source system model used in this research. There are four processing elements (nodes) involved in packet video applications, two elements represent the video source server and the other two represent the user. At the server side, the SA represents the application-oriented layers in the OSI reference model, while the SNIU is the network interface unit for the server. At the user side, the UA represents the application-oriented layers in the OSI reference model, while the UNIU is the network interface unit for the user. The remaining N processing elements that are directly connected to the LAN are used to generate background traffic, e.g. packet voice and data. The number of background nodes is chosen as six based on trials that show that enough background traffic can be generated.

Transfer devices used in the system are internal buses and the LAN. Internal buses are between the SA and the SNIU and between the UA and the UNIU, while the LAN connects the SNIU, the UNIU, and the six background loads. The FCFS protocol is used for the internal buses. As for the LAN, the CSMA/CD protocol is used for Ethernet, while the priority-token-ring protocol is employed for FDDI. The specific parameter values are given in Appendix A.

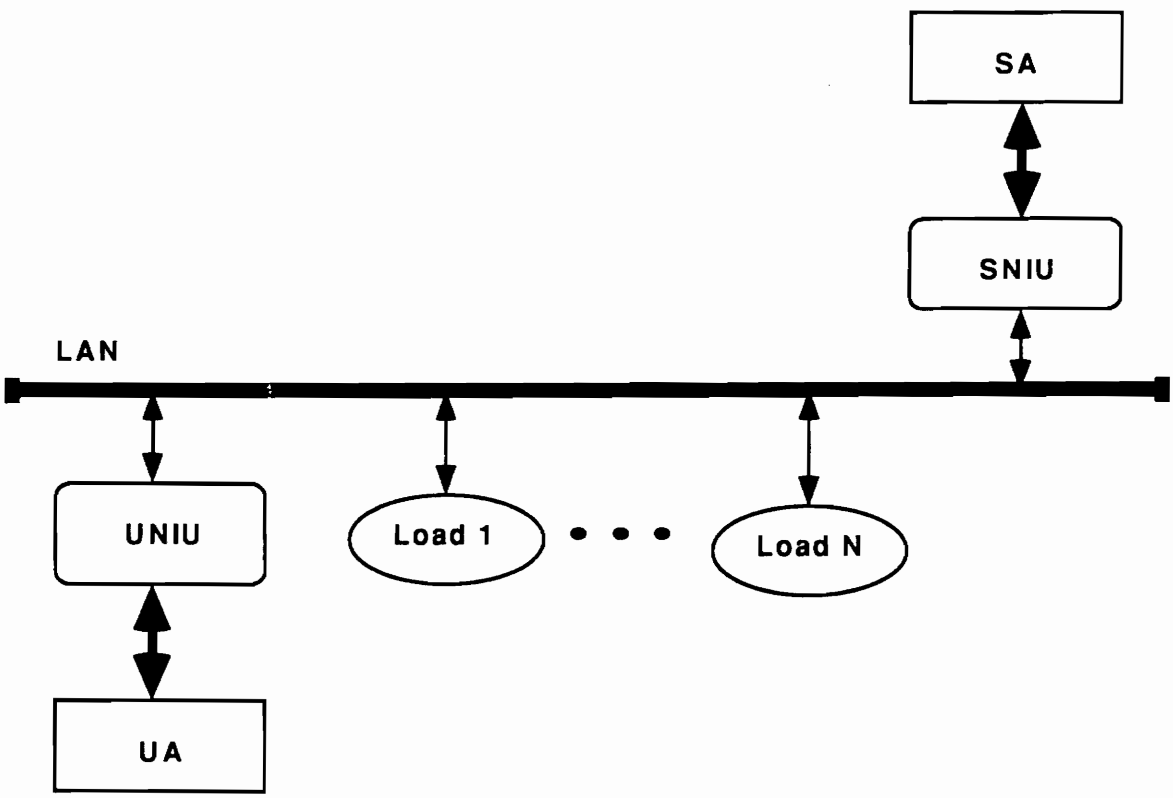


Figure 4.3. Hardware components for the single source system model.

Unlike Ethernet, FDDI assigns a priority to each entity connected to it. The priority is assigned such that video packets have high priority with a synchronous time allocation to ensure that most of them are transferred as synchronous traffic. The background traffic is assigned to be asynchronous traffic, i.e. to have low priority. This strategy can reduce packet delay for video traffic so that a better video quality can be achieved. Ethernet, however, cannot provide this property since all nodes have the same priority.

4.3.1.2. Software Components

Software components used in the simulation are modules, an instruction mix, semaphores, and statistical distributions. Video source images are assumed to be stored in the database of the SA. A connection between the SA and UA is assumed to be established. Thus, the modules are fairly simple. They are *Send Data, Packetize Data, Transfer Data, Depacketize and Relay Data,* and *Read Data.* As shown in Figure 4.4, the video data units are retrieved from the database of the SA and sent to the SNIU where packetization is done. The packetized data are transferred from the SNIU to the UNIU over the underlying LAN. After receiving a frame, the UNIU depacketizes the data and relays it to the UA where data is decoded. This procedure is repeated every 40 ms. Also, N-1 other nodes send messages periodically to another arbitrarily chosen node to generate background traffic. The rate and the packet length of background traffic are assumed to follow normal distributions with means and standard deviations chosen so that the traffic occupies the desired percentage of the LAN utilization, assuming no video traffic.

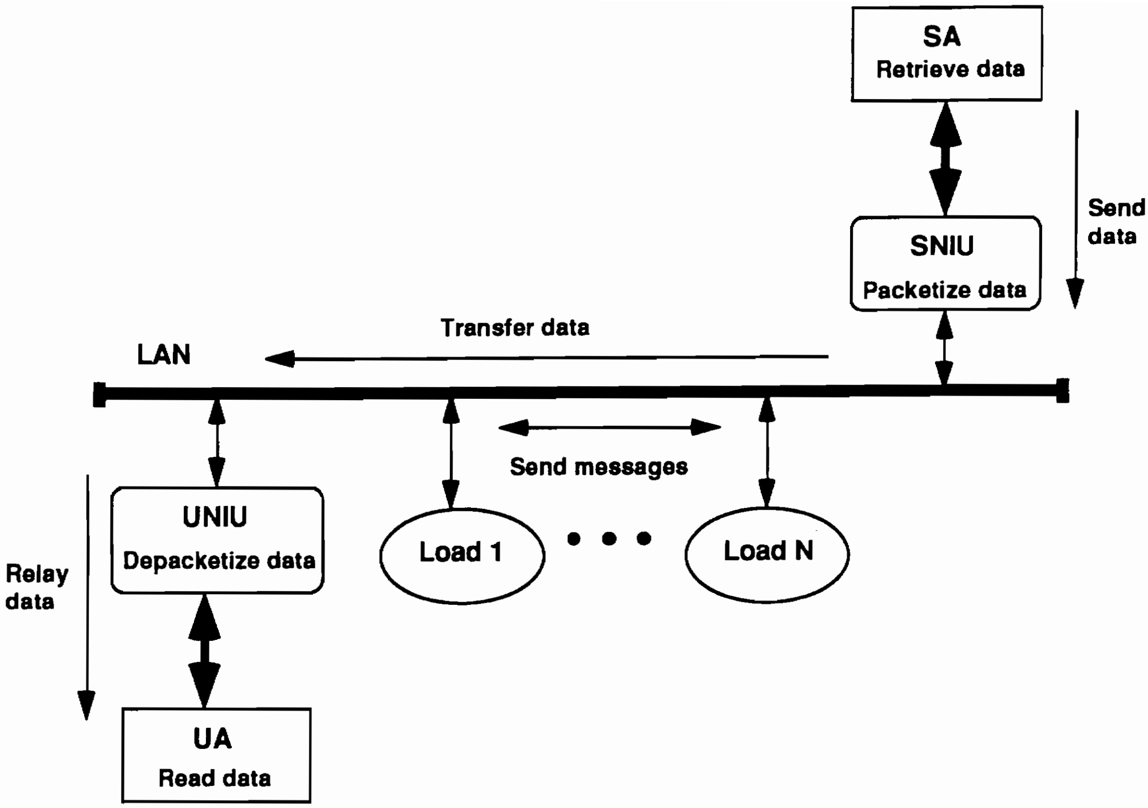


Figure 4.4. Software components for the single source system model.

The instruction mix has been discussed previously. A semaphore is used as a counter to terminate the simulation. Thus the semaphore controls the simulation length. As for statistical distributions, a uniform distribution is used for the initial time condition. Gamma and exponential distributions are employed for the length of video data units coming from the server. Two normal distributions are used to represent the length and iteration period of messages transferred between two background nodes, respectively. "IEEE Backoff" that uses a uniform distribution is used for the collision protocol in Ethernet.

4.3.2. Multiple Source System Models

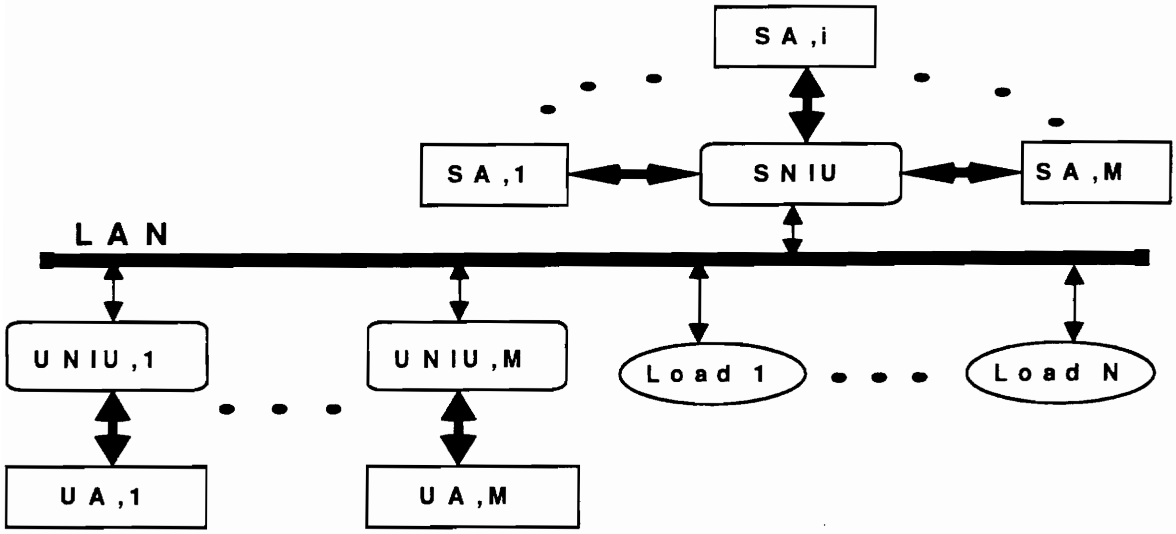
Multiple source system models support more than one video connection at a time. Thus, they are more attractive than a single source system. Due to the same reason as in the single source system, broadcasting is not considered. Therefore, an M-source system has M servers and M users. The following sections describe the hardware and software components of multiple source systems.

4.3.2.1 Hardware Components

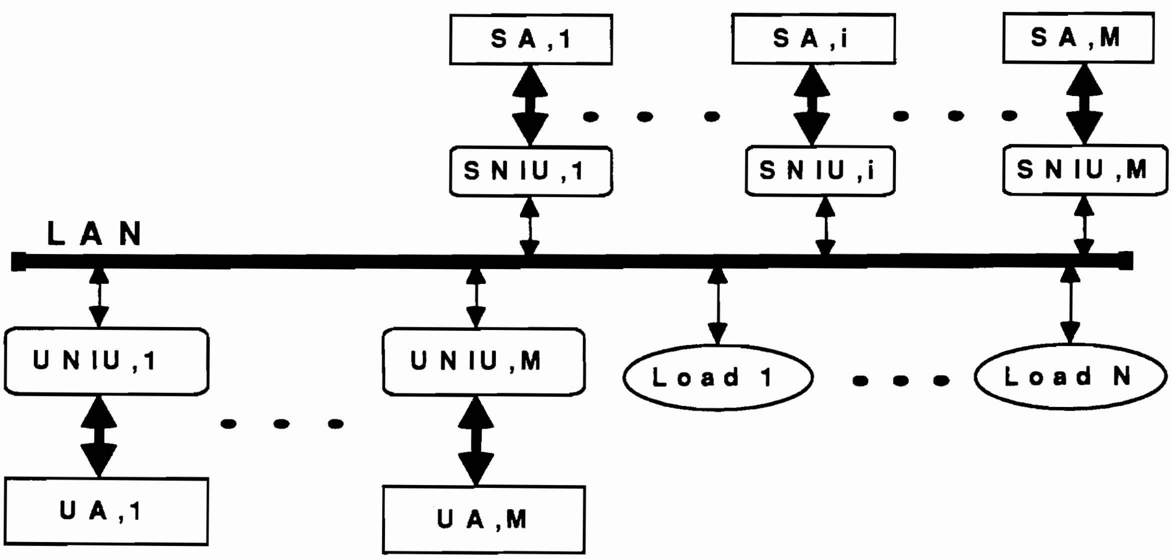
There are two possible hardware structures, as shown in Figure 4.5. The scheme in Figure 4.5 (a) uses a single server (SS) with multiple sources. M SAs are connected to a single SNIU by internal buses. Figure 4.5 (b) shows a hardware structure with multiple servers (MS). Each SA has its own SNIU that is connected to the LAN. In this structure, the servers act independently. The remaining components of the hardware structures are similar to those in the single source system. It is clear that the SS system is less complex than the MS system.

4.3.2.2 Software Components

Like the single source system, all video service connections are assumed to be fixed. Thus each of them works in exactly the same way as the single connection in the single source system. Figure 4.6 (a) gives the software components of a SS system. All M SAs send video data units to the common SNIU every 40 ms, but the time of the initial transmission is a random variable with a uniform distribution. The SNIU packetizes the data units from

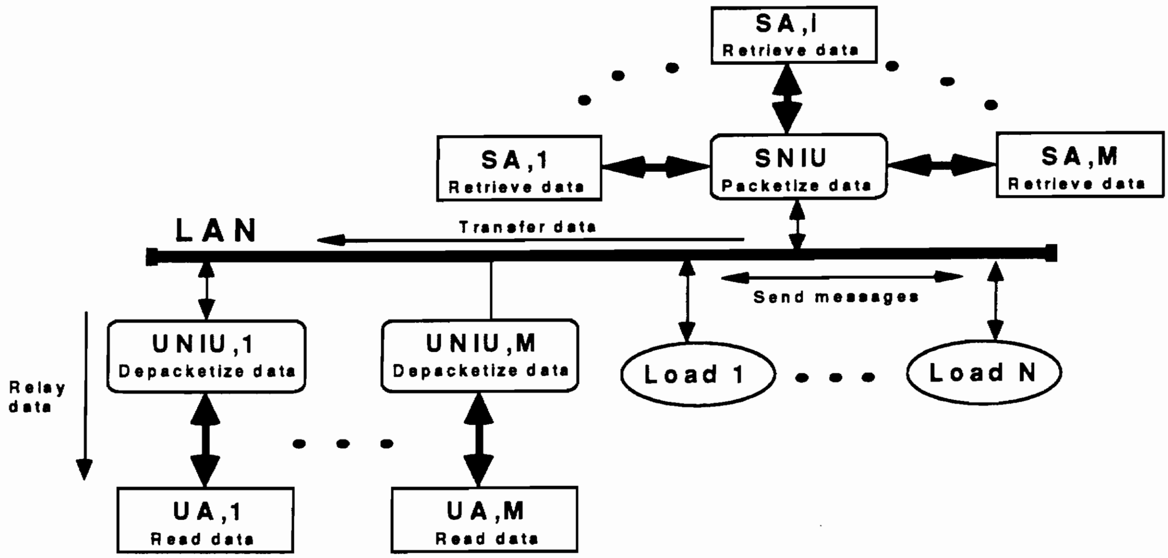


(a) Single server with multiple sources.

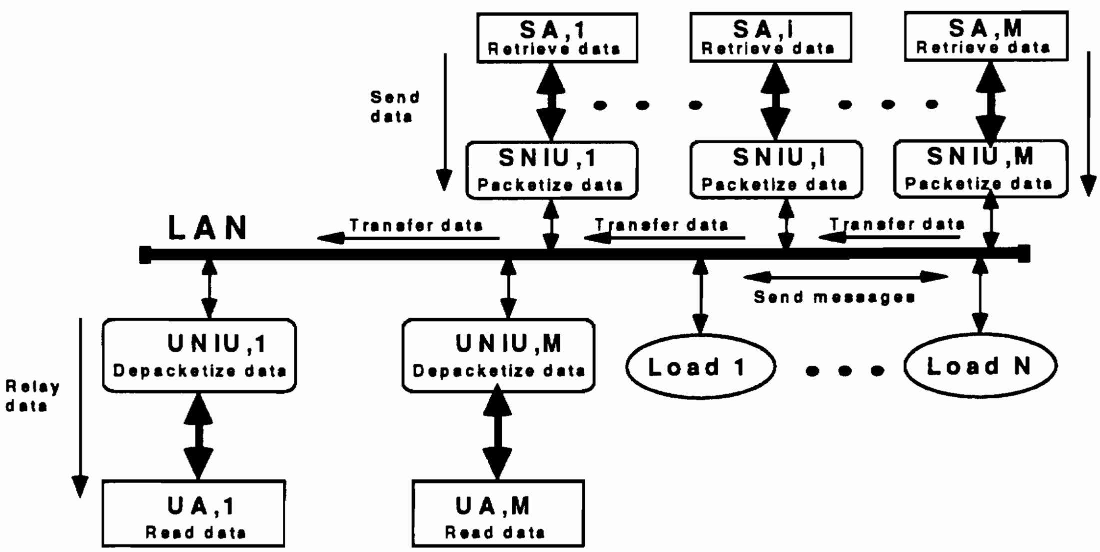


(b) Multiple servers.

Figure 4.5. Hardware components for the multiple source system model.



(a) Single server with multiple sources.



(b) Multiple servers.

Figure 4.6. Software components for the multiple source system model.

all M sources and also functions as a multiplexer. It uses the first-come first-served (FCFS) service discipline.

As shown in Figure 4.6 (b), each SA periodically sends video data units to its own SNIU. The burden on the SNIU in an SS system is relieved since each SNIU takes care of only one source. Thus, the MS structure is superior to the SS structure in terms of software complexity.

The remaining software components of both the SS and the MS systems are similar to those in the single source system. However, the background traffic generated by N nodes must be much lower than in the single video source system due to the limited LAN bandwidth. Hence, the background traffic is fixed at 5 percent.

4.4. Summary

This chapter presented the simulation models for packet video applications over Ethernet and FDDI. Video data are assumed to be digitized and stored at a server or multiple servers and retrieved for transfer to the user. Video source models are adopted from studies of measured data. The video server or servers send data units with a fixed iteration period of 40 ms. However, the length of data units varies for different source models. These source models are applied to both single source and multiple source systems whose hardware and software components were addressed. A sample NETWORK II.5 input file is in Appendix B.

Chapter 5. Simulation Results and Analyses

This chapter presents simulation results and corresponding analyses. A performance comparison is given for Ethernet versus FDDI, the teleconference video source model versus the broadcast video source model, and the single server scheme versus the multiple server scheme for a multiple source system. Parameters that are considered in the comparison are packet delay, including link level and end-to-end delay, system throughput, LAN utilization, packet buffering time at the SNIU and the UNIU, and packet loss probability. Each of these varies with factors such as the type of LAN, video source, and system structure. Section 5.1 describes the simulation environment. Sections 5.2 and 5.3 present the simulation results and analyses for the single source system and the multiple source system, respectively.

5.1. Simulation Environment

For the simulation models described in Chapter 4, it is assumed that a Sun SPARCstation 370 and a Sun SPARCstation 1 are used as a packet video server and a user node, respectively, due to the availability of parameters from [12]. The SPARCstation 370 is rated at approximately 16 million instructions per second (MIPS) and the SPARCstation 1 is rated at approximately 12.5 MIPS. A VMEbus that is specified at 16 MHz [26] is used as the internal bus between the SA and the SNIU, while the internal bus between the UA and the UNIU is an SBus rated at 16.67 MHz [27]. The six background loads are workstations rated at 10 MIPS.

The process of sending video data is started by the server at an arbitrary time with an iteration period of 40 ms. A uniform distribution between 0 ms and 40 ms is employed to choose the starting time. Only the modules that need to send data over the LAN will wait if the LAN is busy. Other active modules are not allowed to be interrupted unless their time slice expires. For the purpose of comparison, 5 percent of Ethernet capacity is used as background traffic for both Ethernet and FDDI in a multiple source system.

It is difficult to determine an appropriate simulation length. The longer the simulation runs, the greater the confidence in the results, but the higher the cost in computer time. Therefore, the length of the simulation run is a tradeoff between confidence and cost. Based on a number of observations, a one-minute simulation was found to be sufficient in that the statistical outcomes for entities and modules in the system converge.

5.2. Single Source System

In the single source system, the interaction between frames carrying video packets and the background traffic is given close attention. The addition of packet video will inevitably affect the existing background traffic and vice versa. Thus, an investigation into their interaction is important. It is observed that Ethernet does not support broadcast video due to its limited bandwidth. When broadcast video is applied to an Ethernet, the network becomes saturated even if operating with only 5 percent background traffic. Therefore, simulation results for broadcast video are shown only for FDDI. Results are obtained from the statistics which are given in Appendix C.

5.2.1. Simulation Results

Tables C.1 to C.3 in Appendix C shows time statistics for modules in the single source system. The modules in Table C.1 (a) are *Send Data*, *Packetize Data*, and *Depacketize and Relay Data*. Each of these modules is processed without a wait after being invoked, i.e. they do not experience any queuing delay. Also, the time consumed by the module *Send Data* is solely machine dependent. It does not depend on the properties of the underlying LAN. The small difference between Ethernet and FDDI for the module *Send Data* in Table A.1 (a) is due to random value generation. However, the times spent by modules *Packetize Data*, and *Depacketize and Relay Data* is LAN dependent. As shown in Table C.2, the time statistics of the module *Transfer Data* are more complicated than the others in Table C.1 since it has a queuing delay in addition to an execution time.

In Tables C.2 and C.3, the background traffic is based on the utilization of Ethernet; the same absolute amount of traffic is used in FDDI for the purpose of comparison. However, the background traffic does not consume the same percentage of FDDI's capacity as it does in Ethernet. In fact, 95 percent background traffic in Ethernet corresponds to only about 9.5 percent in FDDI. Moreover, the maximum background load simulated for teleconference video traffic in Ethernet and broadcast video traffic in FDDI is 60 percent since the network saturates for higher than 60 percent background traffic. This is consistent with results in [9] for teleconference video traffic in Ethernet.

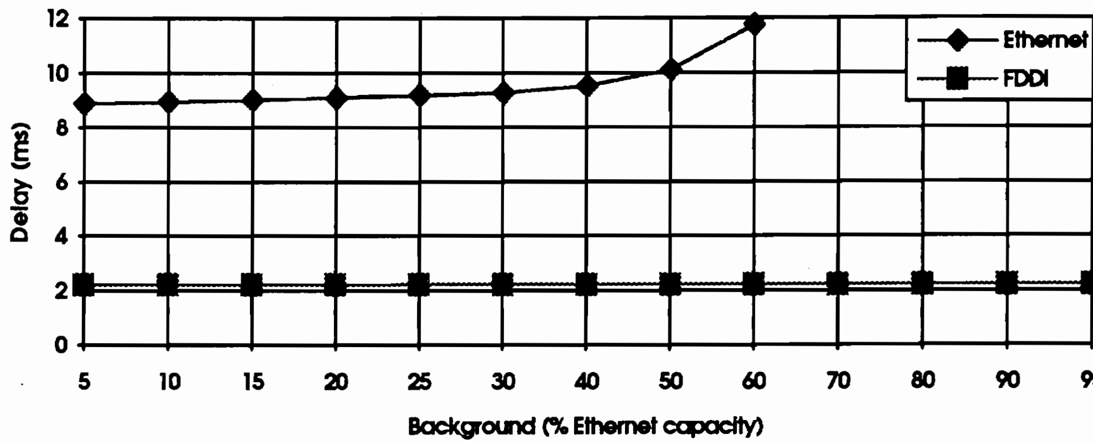
The delay for the background traffic is also important. The performance of the background traffic should not be severely degraded due to the addition of video traffic. It is found that the background traffic from different hosts experiences different degradation in performance. The overall average and the maximum delays are listed in Table C.2 of Appendix C.

Simulation results also show that there is some buffering time at the UNIU for the broadcast video application. This means that an appropriately sized buffer should be provided at the UNIU for broadcast video. The buffer size is discussed along with the multiple source system in section 5.3.2.6.

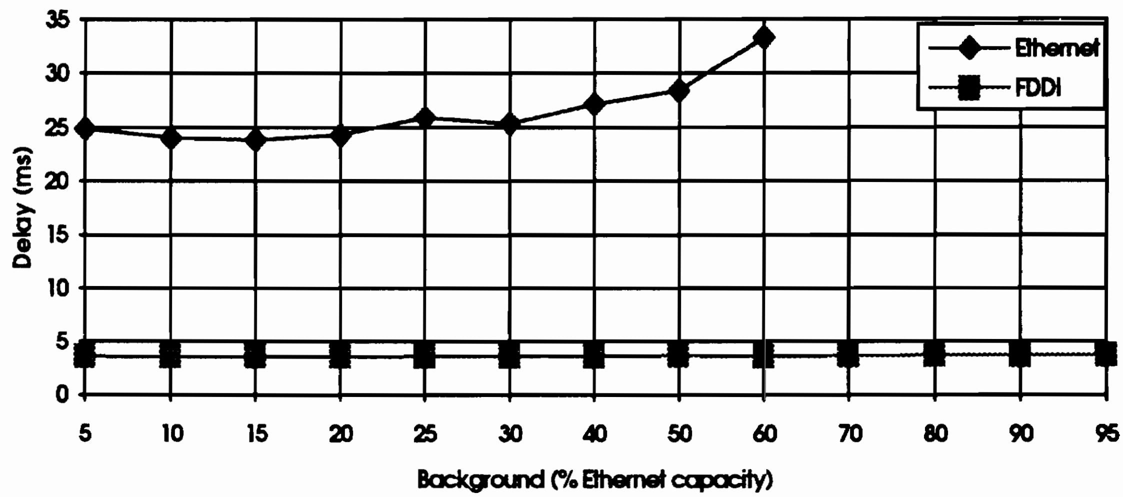
5.2.1.1. Packet Delay

Packet delay is measured for both video traffic and background traffic. Link-level delay and end-to-end delay are considered for the video traffic, while only link-level delay is measured for background traffic. The average and the upper bound of observed packet delay are given close attention since both influence system performance. Note that the "average" and "upper bound" are the average and upper bound, respectively, observed in the simulation runs.

Figure 5.1 shows the average and the upper bound link-level delay for teleconference video traffic for varying background traffic in Ethernet and FDDI. This delay is the time consumed by the *Transfer Data* module. The average link-level delay is the sum of the average queuing time and execution time, while the upper bound of link-level delay is the sum of the maximum values of queuing time and execution time. It is clear that the link-level delay for video packets is much higher in Ethernet than in FDDI. Furthermore, the link-level delay increases much more rapidly in Ethernet when the background traffic increases. The link-level delay curve for FDDI increases slowly.



(a) Average delay.



(b) Upper bound of delay.

Figure 5.1. Link-level delay for teleconference video traffic in the single source system.

A comparison of the source models in terms of the link-level delay in FDDI is plotted in Figure 5.2. Note that results for Ethernet are not shown since it cannot support broadcast video. All the curves increase slowly, although the link-level delay for broadcast video traffic is much higher than that for teleconference video traffic.

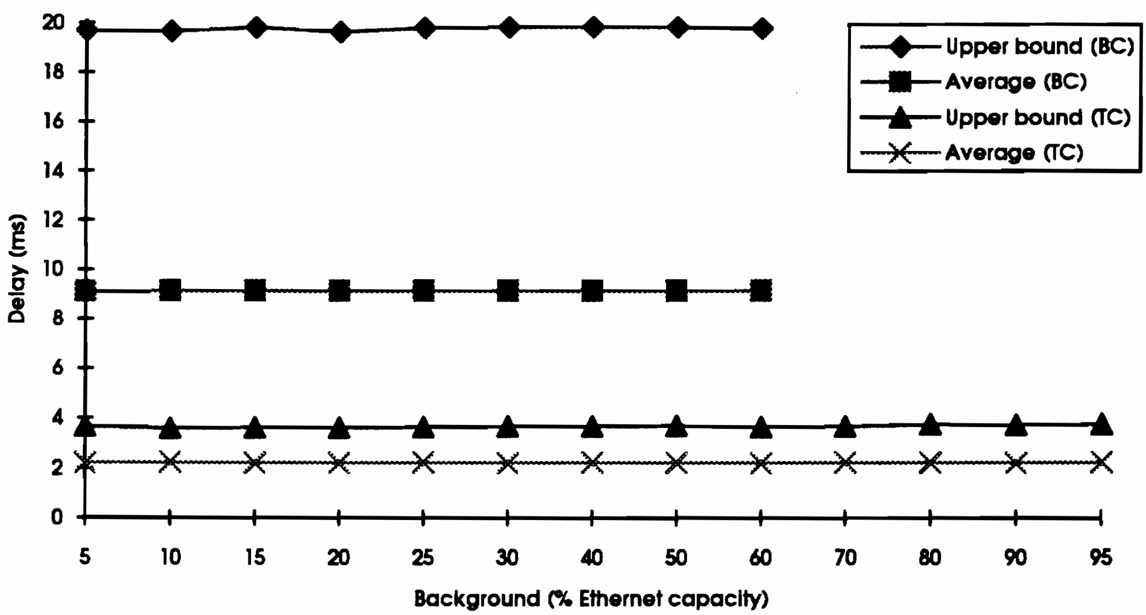


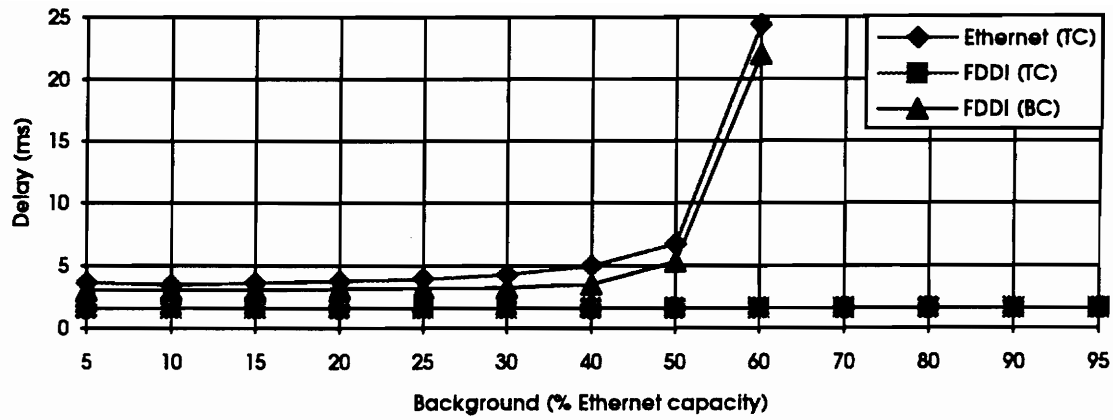
Figure 5.2. Link-level delay for video traffic in the single source system using FDDI for broadcast (BC) and teleconference (TC) traffic.

The average and the upper bound of link-level delay for background traffic is presented in Table C.3 in Appendix C and shown in Figure 5.3. In Figure 5.3 (a), the average delay for broadcast (BC) video traffic in FDDI and for teleconference (TC) video traffic in Ethernet have similar properties, i.e. they start to increase after the background load reaches 25 percent, and increase more rapidly from 50 percent to 60 percent. However, in Figure 5.3 (b), the upper bound of delay for broadcast video traffic in FDDI increases rapidly after 30 percent background traffic is reached, while the upper bound for teleconference video traffic in Ethernet keeps the same properties as average delay.

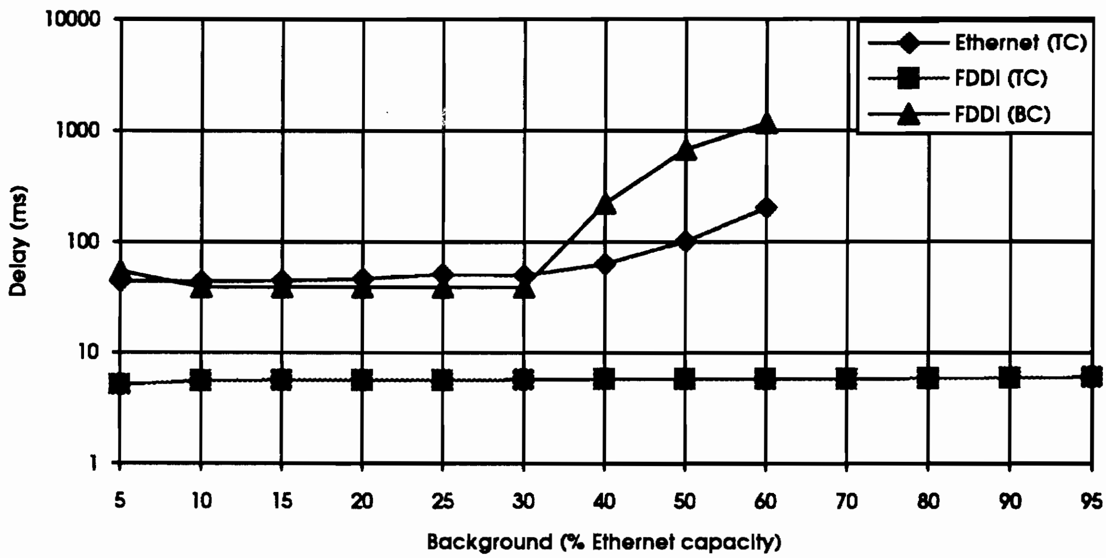
The end-to-end delay for video traffic, one of the overall system performance parameters, is obtained by summing the time spent in each individual module. Figure 5.4 shows the end-to-end delay versus the background traffic for Ethernet and FDDI. It is clear that each of the three curves has properties similar to the link-level delay for video traffic.

5.2.1.2. System Throughput

The system throughput for packet video is another important performance parameter. It is defined as the amount of video information successfully transferred in a time unit in terms of megabits per second (Mbps). Only the average system throughput is considered in this research. As plotted in Figure 5.5, it is the quotient of the average video data length over the average end-to-end delay. It is clear that the system throughput is affected less by background traffic in FDDI than in Ethernet. Also, the throughput for broadcast video traffic is higher than that for teleconference video traffic, although packet delay is greater for broadcast video traffic as shown in Figure 5.2 and 5.3.

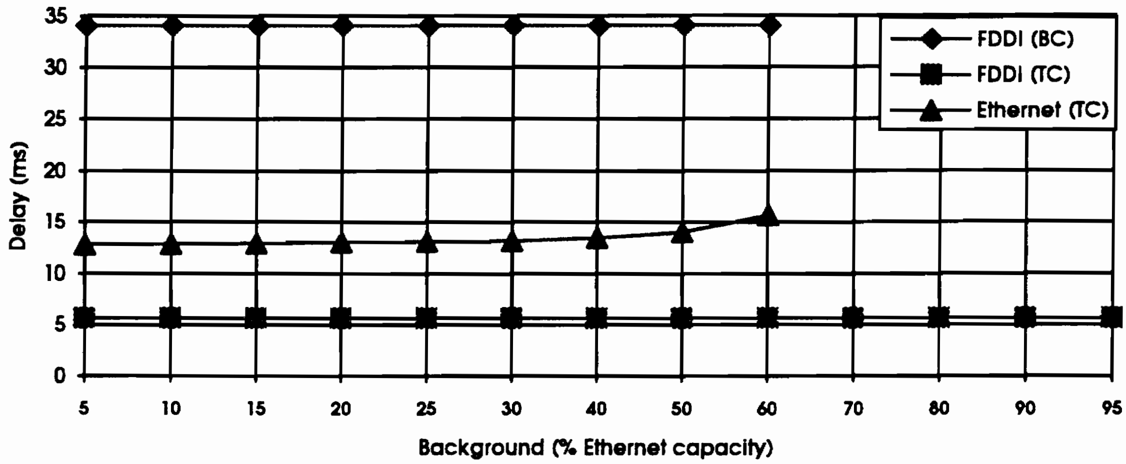


(a) Average delay.

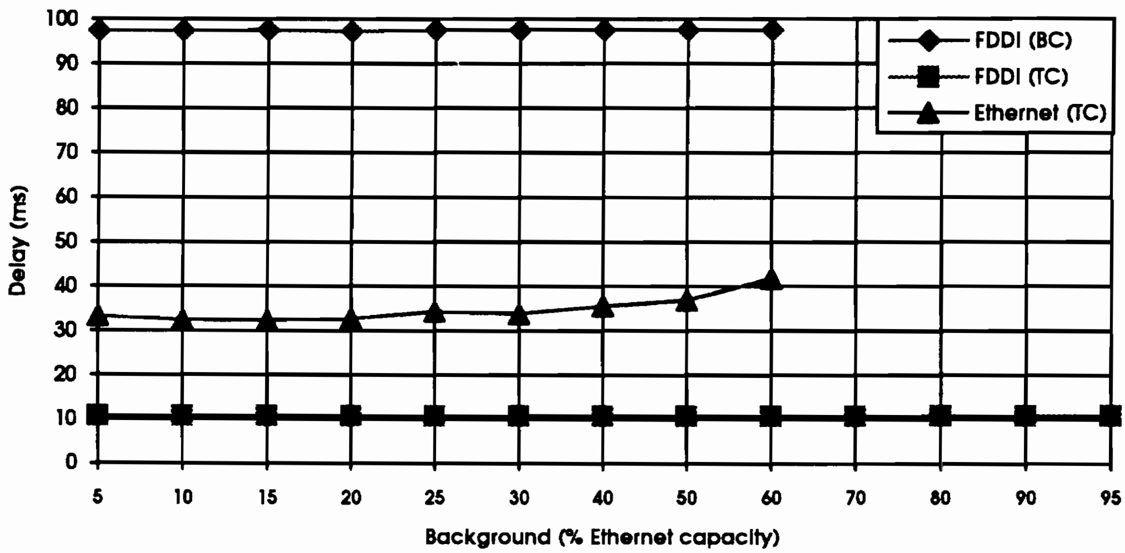


(b) Upper bound of delay.

Figure 5.3. Background traffic delay in the single source system.



(a) Average delay.



(b) Upper bound of delay.

Figure 5.4. End-to-end delay for packet video traffic in the single source system.

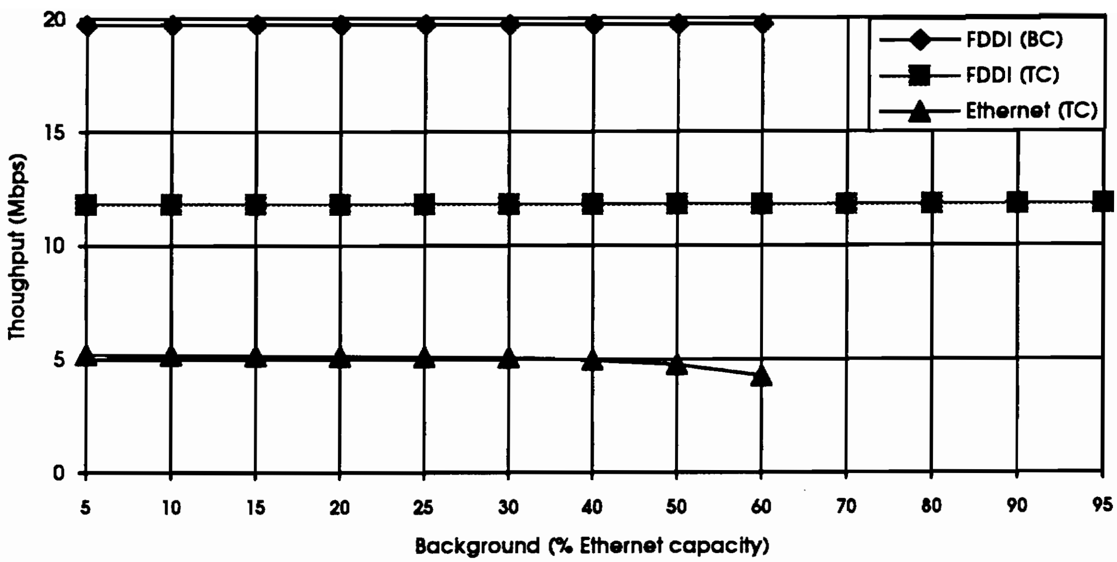


Figure 5.5. Average system throughput in the single source system.

5.2.2. Discussion

As indicated in Chapter 4, the video source model is a critical factor in system performance. The type of traffic determines the amount of information going through the underlying network. How the network reacts to different source models is reflected by the results in Figures 5.1 to 5.5. As described in Chapter 4, the length of broadcast video data units is random over a wide range with large mean and standard deviation. Therefore, broadcast video traffic consumes much more bandwidth of the underlying LAN than does teleconference video traffic. Ethernet cannot support broadcast video applications due to its insufficient bandwidth. Due to bandwidth limits, the network becomes saturated for high background loads for teleconference video traffic in Ethernet and for broadcast video traffic in FDDI.

From Figure 5.1, it is clear that the average and the upper bound of the link-level delay for video traffic are much less in FDDI than in Ethernet. Moreover, the performance degradation due to increasing background traffic in Ethernet is much more serious than in FDDI. The curves in Figure 5.3 have similar properties. In fact, the delay for background traffic is more sensitive to levels of background traffic than the delay for video. This is due to the fact that the level of background traffic is increased by increasing its transmission rate, while its frame length is fixed. When the utilization of the LAN reaches a certain level, frames carrying background traffic must wait more often than ones carrying video traffic. However, this is not the case in FDDI since FDDI is still underutilized even if background traffic equivalent to 95 percent of Ethernet capacity is applied. Therefore, FDDI is more efficient and also less sensitive to changes in background traffic level than Ethernet in the single source system.

FDDI has several advantages over Ethernet for packet video applications. First, FDDI has ten times the bandwidth of Ethernet. The LAN bandwidth is clearly the key factor affecting the speed of packet delivery. Another advantage of FDDI is that it uses a larger block size to transfer data than Ethernet. A large block size reduces the time needed for packetizing and depacketizing video packets at the NIU as fewer packets are needed. The priority scheme in FDDI may also be an advantage. Frames containing video packets are assigned high priority in FDDI. Ethernet does not have a priority scheme.

The maximum allowable background traffic can be determined based on the packet delay distribution of both video traffic and background traffic shown in Figures 5.1 and 5.3. The maximum allowable background traffic could conservatively be set to 5 to 30 percent for teleconference video traffic in Ethernet and broadcast video traffic in FDDI. By observing the curves in Figures 5.1 and 5.3, it is clear that teleconference video traffic in FDDI does not require any restriction on background traffic. This is due to the fact that system performance is not seriously affected even if background traffic equivalent to 95 percent of Ethernet capacity is applied in the system.

5.3. Multiple Source System

In contrast to a single source system, the interaction among video packets themselves is the most important issue in a multiple source system. To ensure fair competition, the same transmission priority is assigned to all video sources. Besides the two video source models, teleconference video source and broadcast video source, and the two LANs, Ethernet and FDDI, two schemes, single server (SS) and multiple server (MS), are also simulated and compared. Simulation results are obtained from the summary reports which are partially given in Appendix C.

5.3.1. Simulation Results

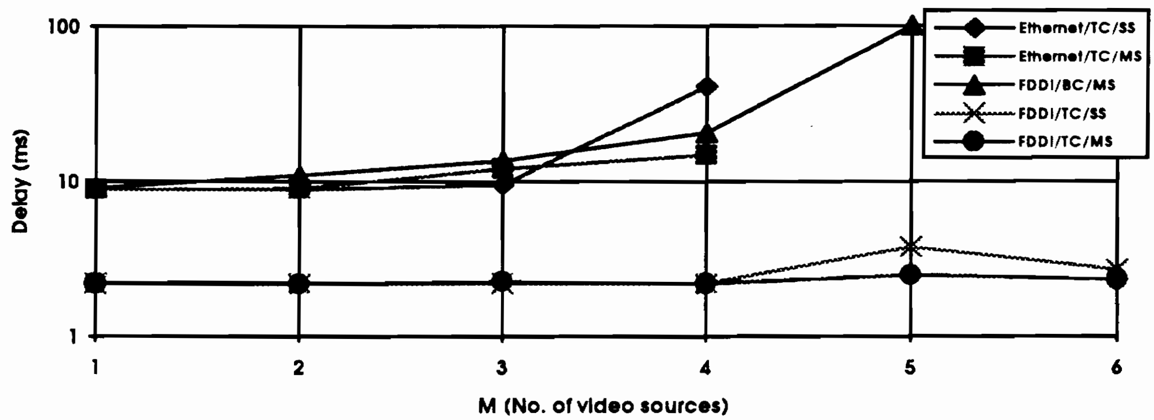
In a multiple source system, the amount of information sent by different sources is slightly different. This is due to the fact that the length of video data units is a random variable, although the transmission rate is a constant and the length of data units has the same statistical distribution for all souces. Therefore, the time consumed by the corresponding modules from different sources is slightly different. Also, since much more traffic goes through the network in a multiple source system than in a single source system, hosts or the LAN may be busy when they receive a processing request from a module. Thus, modules have to wait even if they are already active. In this case, packets that need to be delivered must wait in a queue until the host or the LAN becomes available. To reduce the probability of packet loss, a buffer at the SNIU or the UNIU is needed. The size of the buffer is determined by the queuing time of a module in the SNIU or the UNIU.

The initial or beginning execution time of the *Send Data* module has an effect on the buffering time at the NIU input queue. This is called the source periodic effect and will be discussed in Section 5.3.2.5. The initial time of the send data module is uniformly distributed between 0 ms and 40 ms for all sources. Also, low background traffic, 5 percent of Ethernet capacity, is simulated to ensure that most of the LAN bandwidth is available to video traffic. The effect of background traffic will be discussed in Section 5.3.2.3.

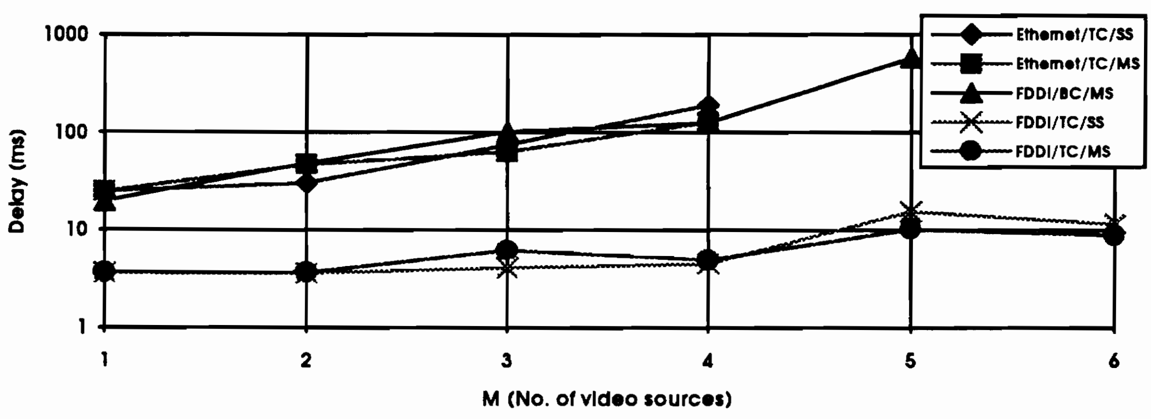
From simulation results, Ethernet can support a maximum of four teleconference video sources under both the SS scheme and the MS scheme. Although not shown in detail, FDDI can support more than sixty teleconference video sources for the MS scheme, fourteen teleconference video sources for the SS scheme, five broadcast video sources for the MS scheme, and two broadcast video sources for the SS scheme. Broadcast video traffic in FDDI for the SS scheme is not discussed in detail here since a two-source system is too limited to be considered as a multiple source system.

5.3.1.1. Packet Delay

Packet delay, including link-level and end-to-end delay, is considered only for video traffic, not for the background traffic in the multiple source system. Figure 5.6 illustrates the link-level delay, including the overall average and the upper bound of delay. They are measured in the same manner as in the single source system. It should be noted that the packet delay for different video connections in the same system may be different since the length of video data units is random. Only the overall average and the observed upper bound are considered in these results.



(a) Average delay.



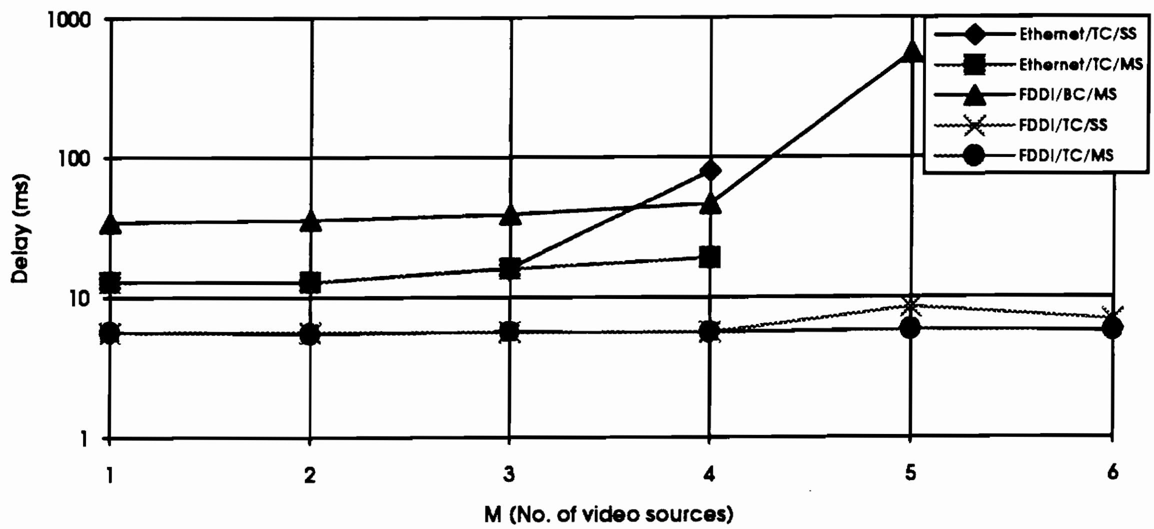
(b) Upper bound of delay.

Figure 5.6. Link-level delay in the multiple source system.

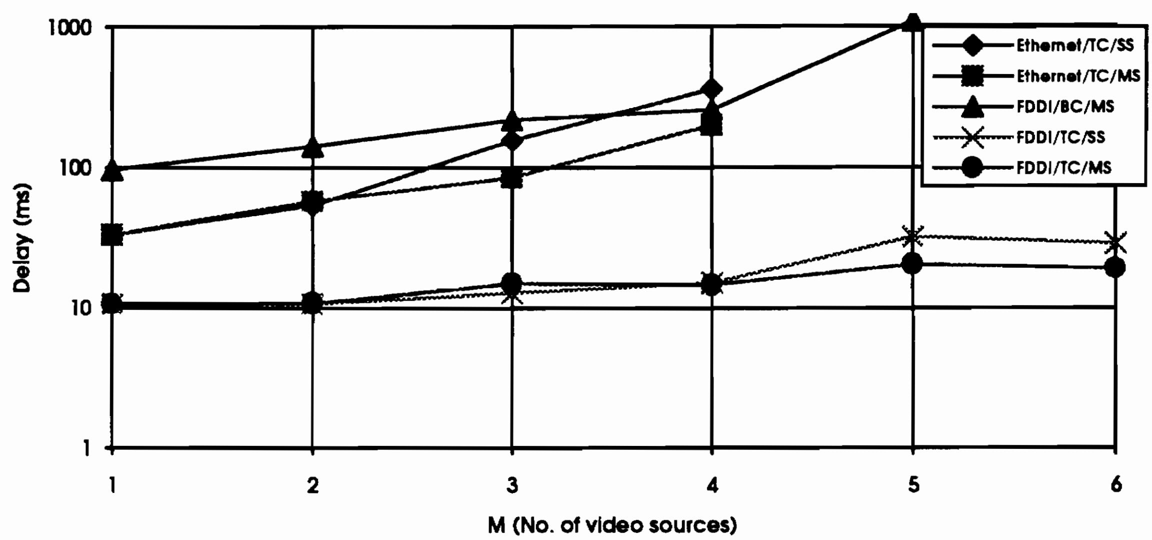
In general, the delay for teleconference video traffic in Ethernet is much higher than in FDDI. Also, the delay for teleconference video traffic in FDDI increases slowly with increasing M, the number of sources, for both the SS scheme and the MS scheme. It is hard to tell the difference between the two schemes in FDDI. The average delay for teleconference video traffic in Ethernet under the SS scheme is close to that under the MS scheme when M does not exceed three and much higher when M is four. As for broadcast video traffic in FDDI, the delay increases gradually with increasing M when M is less than four. However, the delay is high for broadcast video traffic when M is five. Figure 5.7 illustrates the overall average and the upper bound of end-to-end delay for video traffic. It is clear that the curves in Figure 5.7 have similar properties to those in Figure 5.6.

5.3.1.2. System Throughput

Figure 5.8 shows the overall average system throughput for the two video source models over Ethernet and FDDI. This is measured in the same manner as in the single source system. Like packet delay, the throughput for different video connections in the same system may be different. Only the overall average is considered. In Figure 5.8, the throughput for teleconference video traffic in Ethernet with the SS scheme is close to that with the MS scheme when M is less than four, and is much less than with the MS scheme when M equals four. Except when M is five and six, the throughput for teleconference video traffic in FDDI with the SS scheme and the MS scheme is close. Also, the throughput for broadcast video in FDDI decreases gradually with increasing M when M is less than five, but falls rapidly to near 0 when M is five.



(a) Average delay.



(b) Upper bound of delay.

Figure 5.7. End-to-end packet delay in the multiple source system.

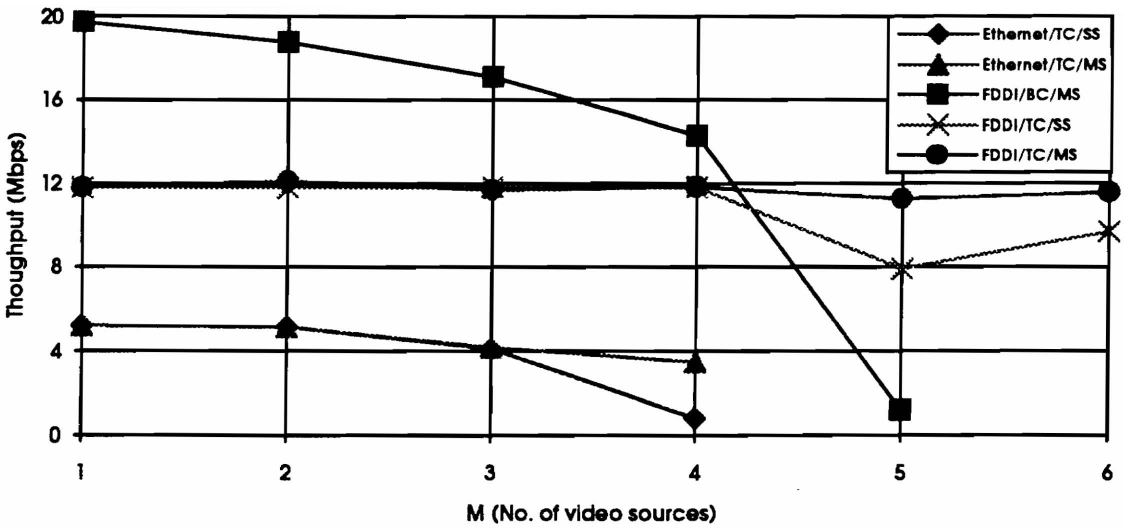


Figure 5.8. Average system throughput in the multiple source system.

5.3.1.3. Utilization of the LAN

LAN bandwidth is one of the main factors that limits the number of video sources. Figure 5.9 shows the LAN utilization for different video source models. The LAN utilization is the same for the SS scheme and the MS scheme. It is clear that the LAN utilization for broadcast video traffic in FDDI remains close to that for teleconference video traffic in Ethernet as M increases. However, the LAN utilization for teleconference video traffic in FDDI is low and increases slowly when M is less than seven. From Figure 5.9, the maximum number of video sources can be obtained for a specific system.

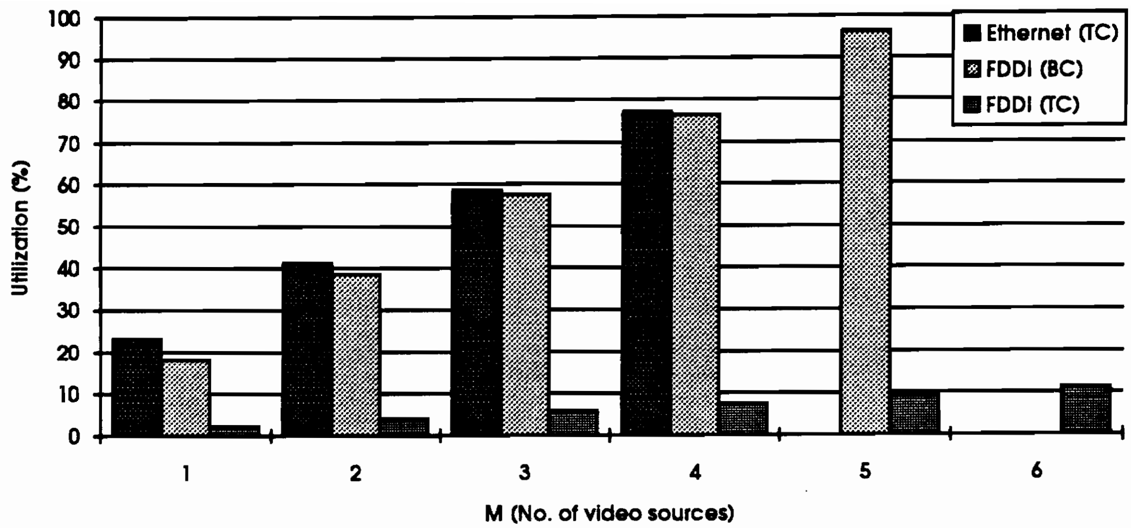
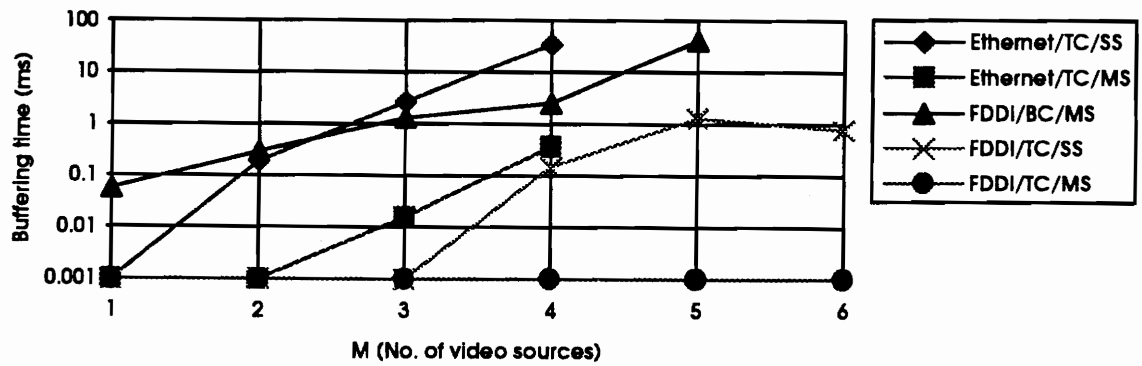


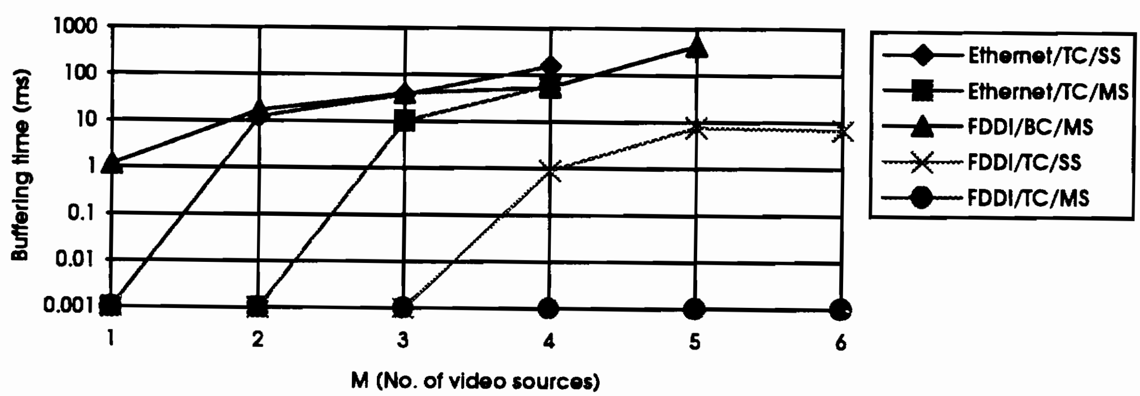
Figure 5.9. LAN utilization in the multiple source system.

5.3.1.4. Packet Buffering Time

Since there are 1500 packets going through the LAN from each source to its destination in one minute, queuing and collisions will occur more often in a multiple source system than a single source system. Thus, modules *Packetize Data* and *Depacketize and Relay Data* as well as the *Transfer Data* module may need to wait to be processed if the host is busy. Figure 5.10 illustrates the input buffering time at the SNIU for different multiple source systems. It should be noted that a buffering time of 0.001 ms represents 0 ms since the logarithmic chart cannot plot zero or negative values. It is clear that the input buffering time for FDDI systems is much less than for Ethernet systems. Also, when M is greater than two, the average input buffering time with the SS scheme is much higher than with the MS scheme for teleconference video traffic in Ethernet. However, this is not the case for the upper bound of input buffering time until M exceeds three. As for broadcast video traffic in FDDI, the input buffering time goes up dramatically from the four-source system to the five-source system, while it increases gradually with increasing of M when M is less than five.



(a) Average buffering time.



(b) Upper bound of buffering time.

Figure 5.10. Input buffering time at the SNIU in the multiple source system.

For broadcast video traffic in FDDI, video packets may also need to wait at the UNIU. Table 5.1 shows the average and the upper bound input buffering time at the UNIU for broadcast video traffic under the MS scheme. The average input buffering time slightly decreases with increasing *M*, while its upper bound remains the same until *M* exceeds three. This is due to the fact that queuing or collisions occur more often on the LAN for frames containing video packets when *M* increases. Thus, on average, more time is consumed for frames to be transferred over the LAN with increasing *M*. The inter-arrival rate of video frames at the UNIU then decreases relative to the rate when M is smaller. According to the Pollaczek-Khinchin formula [28], waiting time in the input buffer of the UNIU should decrease. However, the upper bound of the input buffering time is supposed to be a constant. When *M* is four and five, it is actually increasing. This may be due to the fact that the maximum length of video data generated at one of the SAs has been increased by the random number generator in the system.

Table 5.1. Input Buffering Time at the UNIU for Broadcast Video Traffic in FDDI

M	1	2	3	4	5
Average (ms)	2.862	2.486	2.278	1.885	1.783
Upper-bound (ms)	26.582	26.582	26.582	26.715	32.767

5.3.1.5. Packet Loss

The input buffering time consumed by video packets at the NIU reflects the size of the input buffer needed at the NIU to avoid or at least reduce the transport layer packet loss. Figure 5.11 shows the packet loss probability for three sample input buffer sizes at the SNIU in the four-source teleconference video system. It is measured as the percentage of lost packets out of the 6,000 packets that are sent by the four sources during one minute. As shown in Figure 5.10, packets do not encounter any input buffering at the SNIU in the MS scheme for teleconference video traffic in FDDI. Thus, only the SS scheme results in packet buffering at the SNIU. The packet loss probability for teleconference video traffic in Ethernet with the SS scheme is not shown in Figure 5.11. In fact, following the same procedure, it needs much larger buffers at the SNIU to ensure that the packet loss probability is sufficiently small. For broadcast video traffic in FDDI, a buffer is also needed at the UNIU, in addition to the SNIU, since packets also encounter waiting there.

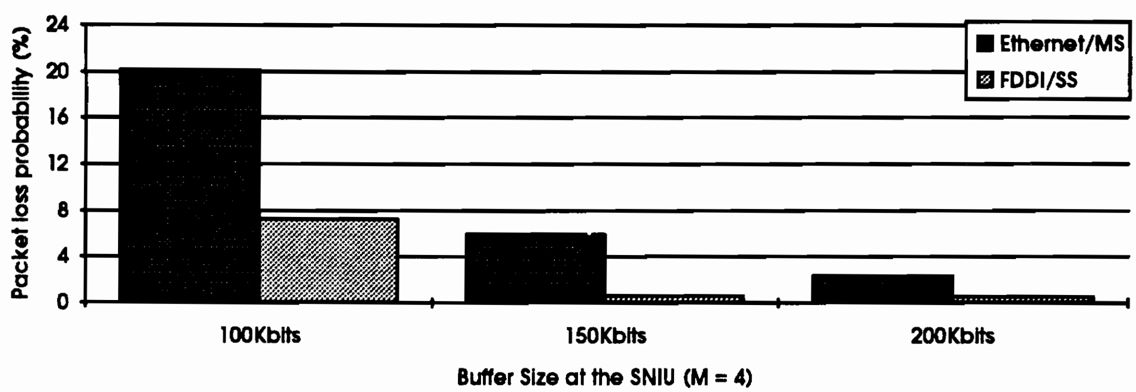


Figure 5.11. Packet loss probability for teleconference video traffic (M = 4).

5.3.2. Discussion

As mentioned previously, a multiple source system is much more complex than a single source system. More factors affect the system performance. A brief discussion of these factors is given below.

5.3.2.1. Effect of LAN

Like the single source system, the LAN is an important factor that affects system performance. It is clear that FDDI is more suitable for multiple source systems. In fact, Ethernet cannot support broadcast video. Besides a higher bandwidth and a larger block size in FDDI than in Ethernet, the priority scheme in FDDI can also reduce the impact of background traffic on frames caring video packets. Nevertheless, competition still exists among video data themselves, even in FDDI. Therefore, FDDI only support at most five sources for broadcast video traffic.

5.3.2.2. Effect of Sources

There are two parameters of the video sources that will affect system performance in the multiple source system: source models and the number of sources. A comparison of source models was discussed in the single source system. That discussion still holds for the multiple source system.

The effect of the number of sources for teleconference video traffic in Ethernet is different from that in FDDI. The information from each individual source has a significant effect on the LAN bandwidth in Ethernet, while it occupies a small portion of the bandwidth in FDDI. Hence, Ethernet becomes saturated when the number of sources is increased to five, while FDDI is still under utilized in this case. Broadcast video traffic in FDDI is similar to teleconference video traffic in Ethernet in terms of the underlying relationship between the source model and the bandwidth of the LAN.

5.3.2.3. Effect of Background Traffic

Although the allowed background traffic in the multiple source system is low, it still has an obvious effect on system performance. Table 5.2 shows a comparison of background traffic effect. When background traffic is increased from 5 percent to 10 percent, the input buffering time at the SNIU and the link-level delay for video traffic increases. This means that background traffic affects both packet delay and packet loss. It is clear that background traffic has a significant impact on teleconference video traffic in Ethernet, but a small impact on the same source model in FDDI. This is due to the fact that FDDI has higher bandwidth than Ethernet. Background traffic resulting in a five percent increase in Ethernet utilization causes only 0.5 percent increase in FDDI utilization. Although broadcast video traffic in FDDI and teleconference video traffic in Ethernet have a similar bandwidth problem, the effect of background traffic on broadcast video traffic in FDDI is less than on teleconference video traffic in Ethernet. This is due to the fact that FDDI has a priority scheme while Ethernet does not. In FDDI, background traffic can be assigned low priority and video traffic can be assigned high priority. Thus, background traffic has less influence on video traffic in FDDI than in Ethernet.

Table 5.2 (a). Input Buffering Time at the SNIU for Different Background Traffic (M=4)

Input buffering time at the SNIU (ms)		Ethernet	FDDI	
		TC/MS	TC/MS	BC/MS
Average	5% background	0.372	0.000	2.598
	10% background	0.670	0.000	2.672
Upper bound	5% background	60.336	0.000	55.662
	10% background	72.266	0.000	55.822

Table 5.2 (b). Link-level Delay for Different Background Traffic (M=4)

Link level delay (ms)		Ethernet	FDDI	
		TC/MS	TC/MS	BC/MS
Average	5% background	14.857	2.212	20.617
	10% background	16.487	2.212	20.622
Upper bound	5% background	128.922	4.990	126.693
	10% background	187.520	4.990	133.862

5.3.2.4. Effect of System Implementation

From the packet delay and system throughput shown in Figures 5.6 through 5.8 and the input buffering time at the SNIU given in Figure 5.10, it is clear that the MS scheme results in better system performance than the SS scheme, but the difference is small when the number of video sources is less than four. However, a larger buffer is needed at the NIU to reduce the packet loss with the SS scheme than with the MS scheme. Since the same amount of video information moves over the LAN in both schemes, their effect on LAN utilization is the same.

As mentioned in Chapter 4, the hardware structure for the SS scheme is less complicated than the MS structure. For the SS scheme, one SNIU packetizes packets coming from all sources and acts as a multiplexer for transferring these packets. The SNIU for the MS scheme, however, only handles an individual video source. Therefore, the SNIU in a SS system saturates more quickly than in a MS system as M increases. In fact, the SNIU's utilization in the SS system limits the maximum number of video sources that the system can support, while the LAN utilization in the MS system determines the maximum number of video sources.

5.3.2.5. Effect of Source Periodicity

As mentioned in the discussion of video source coding schemes in Chapter 3, the output iteration period from the encoder is fixed at 40 ms, while the length of data units is random. Due to this fact, a phenomenon called the source periodicity effect [5] occurs if the initial transmit time for two sources is relatively close. Three initial time conditions are considered in this research: uniformly distributed, equally spaced, and synchronized. They are discussed below.

Previously, the initial execution time for the *Send Data* module is uniformly distributed between 0 ms and 40 ms. Suppose SA 1 invokes the *Send Data* module immediately after SA 2. For the SS scheme, the packets from SA 1 have to wait in an input queue at the SNIU for packetization until the last packet from SA 2 has finished packetization, and then wait again for transmission until the last packet from SA 2 has entered the LAN. Thus, the input buffering time and the link-level delay for the packets from SA 1 are increased due to the waiting. For the MS scheme, packets from SA 1 do not have to wait for packetization since they do not share the same SNIU with the packets from SA 2. However, they must wait at the LAN since SNIU 2 begins transmission before SNIU 1.

One way to reduce the effect of source periodicity is to space the initial time within the range from 0 ms to 40 ms. This scheme avoids, or at least reduces, the waiting time at the SNIU for the packets from the SA which invokes the *Send Data* module later than another SA in the multiple source system.

To compare the effect of different initial time conditions, a synchronized initial time scheme is also simulated. In this scheme, all sources begin transmission at the same time. It is clear that synchronized initial times are the worst case due to the most severe competition among the sources. Table 5.3 (a) shows the input buffering time for a four-source Ethernet system and a five-source FDDI system for synchronized initial time and equally spaced initial time. Table 5.3 (b) illustrates the link-level delay under the two conditions. It is obvious that the equally spaced initial time scheme significantly improves system performance.

Table 5.3 (a). Input Buffering Time at the SNIU for Teleconference Video under Two Initial Time Conditions

Input buffering time (ms)		Ethernet (M = 4)		FDDI (M = 5)	
		SS	MS	SS	MS
Average	Synchronized	32.775	0.828	1.000	0.000
	Equally Spaced	20.644	0.377	0.073	0.000
Upper bound	Synchronized	181.063	83.671	5.486	0.000
	Equally spaced	139.883	93.450	0.551	0.000

Table 5.3 (b). Link-level Delay for Teleconference Video Traffic under Two Initial Time Conditions

Link level delay (ms)		Ethernet (M = 4)		FDDI (M = 5)	
		SS	MS	SS	MS
Average	Synchronized	38.110	19.756	7.872	2.838
	Equally spaced	31.729	14.253	2.208	2.208
Upper bound	Synchronized	200.503	201.995	15.834	11.691
	Equally spaced	166.792	202.458	4.768	5.026

It should be noted that the results for uniformly distributed initial times are supposed to be between the results for synchronized initial times and equally spaced initial times. However, the simulation results happen to differ. This is due to the fact that the amount of information sent from the SA for the uniformly distributed initial time condition is different from that for the other two conditions due to random number generation. To avoid confusion, the values for the uniformly distributed initial time condition that have been plotted in previous figures are not listed in Table 5.3.

5.3.2.6. Buffer Size Determination

In a multiple source system, the input buffer size is an important issue. Packets from different sources encounter different input buffering times at the NIU. The buffering time is not constant due to several factors such as the source model, initial time condition, and background traffic in the LAN. Hence, it is difficult to determine a proper buffer size. A buffer size that is too small will result in severe packet loss, while buffer size that is too large will waste hardware.

For teleconference video applications, packets generally need to be buffered only at the SNIU; no extra buffer is needed at the UNIU. However, for broadcast video applications, an input buffer is also needed at the UNIU to reduce packet loss. The statistical property of the input buffering time is used to determine a specific buffer size at the NIU since the input buffering time itself is random.

5.4. Summary

This chapter presented simulation results and analyses for both the single source system and the multiple source system. The system performance was studied and compared for two LANs, Ethernet and FDDI, for two source models, teleconference video and broadcast video, and for two system implementations, single server and multiple server. It is clear that FDDI provides better system performance than Ethernet for packet video applications. Moreover, teleconference video allows better performance than broadcast video since less LAN bandwidth is consumed. For teleconference video applications in the multiple source system, the multiple server scheme provides relatively better performance than the single server scheme in Ethernet, but provides almost the same performance in FDDI. The hardware for the single server scheme is simpler than for the multiple server scheme, while a larger input buffer is needed at the NIU for the single server scheme than for the multiple server scheme.

Other factors that affect the system performance, such as background traffic, number of sources and initial time conditions, were also discussed. In the single source system, a maximum level of background traffic can be determined based on the performance of video traffic and background traffic. In the multiple source system, the maximum number of sources depends on the specific requirements for system performance. Also, an equally spaced initial time condition can significantly improve performance, while a synchronous initial time condition seriously degrades performance.

Chapter 6. Summary and Conclusions

This research investigated the implementation and performance of packet video transfer over local area networks. System performance was evaluated using simulation for different video sources, LANs, and implementation methods. Conclusions are presented in Section 6.1. Suggestions for future work are given in Section 6.2.

6.1. Conclusions

Conclusions are drawn from the simulation results and analyses presented in Chapter 5. The video source models adopted in this research are from measured data reported in [5] and [14]. Two existing LANs, Ethernet and FDDI, are simulated. In the multiple source system, two hardware implementation methods, single server and multiple server, and three initial time conditions, synchronized, uniformly distributed and equally spaced, are considered.

The length of video data units from the encoder is larger for broadcast video than for teleconference video if their encoding rate is the same. The statistical distribution of the length of data units for the teleconference video is a combination of gamma and exponential functions with 99.4 percent gamma distribution and 0.6 percent exponential distribution. The length of data units for broadcast video follows a normal distribution with much larger mean and standard deviation than for teleconference video. Hence, broadcast video traffic requires much higher LAN bandwidth than teleconference video traffic to successfully transfer video. According to the simulation results presented in Chapter 5, Ethernet cannot support broadcast video traffic due to insufficient bandwidth, while FDDI can support at most five simultaneous broadcast video sources. The packet delay for broadcast video traffic is also much larger than the packet delay for teleconference video traffic. However, the throughput of the broadcast video system is higher than the throughput of the teleconference video system.

FDDI provides better system performance than Ethernet in terms of packet delay, system throughput, and LAN utilization. This is due to three reasons: (1) FDDI has ten times the bandwidth of Ethernet, (2) FDDI uses a larger frame size to transfer packets, and (3) the priority scheme in FDDI accelerates the delivery of video frames, while Ethernet does not support priorities.

In the multiple source system, the multiple server scheme provides better system performance than the single server scheme for Ethernet, but the difference is small when the number of sources is less than four. The single server scheme produces almost the same performance as the multiple server scheme in FDDI for teleconference video. However, a larger buffer is needed at the NIU for the single server scheme than with the multiple server scheme. As for the initial time condition, an equally spaced initial time condition is the best among the three schemes in terms of system performance, while a synchronized initial time condition is the worst.

6.2. *Future Work*

This research focused on how different LANs support packet video applications. Simulations were done under the assumptions that a connection is established for each video session and video images are digitized and properly stored in the database of one or more servers. Some suggestions are given below for further study.

Further research can explore higher layers of the OSI reference model for packet video. Most of the work load is performed by higher layers, while lower layers merely provide real-time services. Although complex, end-to-end simulations could be conducted to obtain an overall understanding of packet video over networks.

From the simulations done in this research, it is clear that bandwidth is the key reason why FDDI provides better system performance than Ethernet. Further research can study the effect of different network structures, e.g. multi-bus and ring-bus, by setting the same bandwidth for Ethernet and FDDI.

In this research, the same video source model is applied to Ethernet and FDDI for the purpose of comparison. However, as mentioned in Chapter 3, better coordination among entities in the system can further improve performance. Thus, further work is needed to examine the FDDI's priority scheme by applying subband coding and two-layer coding schemes in FDDI.

In recent years, asynchronous transfer mode (ATM) switching has become widely recognized as a viable solution for broadband networks [4]. ATM can also be used in local area networks. Therefore, further research can be done to optimize the system performance by transferring packet video over an ATM network with subband and two-layer coding schemes. With high bandwidth, dynamic priority scheme and genetic flow control, ATM should provide good performance for packet video applications.

There may be other alternative schemes that can be studied for video coding, implementation methods, and underlying networks. Hence, further improvement of system performance can be obtained by adjusting the coordination among entities in the system.

REFERENCES

1. C. J. Turner and L. L. Peterson, "Image Transfer: An End-to-End Design," *Proceedings ACM SIGCOMM*, pp. 258-268, 1992.

2. G. Karlsson and M. Vetterli, "Packet Video and Its Integration into the Network Architecture," *IEEE J. Select. Areas Commun.*, vol. 7, no. 5, pp. 739-751, June 1989.

3. R. J. Moorhead, J. S. Ma, and C. A. Gonzales, "Realtime Video Transmission over a Fast Packet-Switched Network," *Proceedings SPIE Conference: Digital Image Processing Applications*, vol. 1075, pp. 118-123, 1989.

4. W. Verbiest, L. Pinnoo, and B. Voeten, "The Impact of the ATM Concept on Video Coding," *IEEE J. Select. Area Commun.*, vol. 6, no. 9, pp. 1623-1632, December 1988.

5. D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE Transactions on Circuits and Systems for Video Tech.*, vol. 2, no. 1, pp. 49-58, March 1992.

6. M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks," *IEEE J. Select. Areas Commun.*, vol. 7, no. 5, pp. 771-781, June 1989.

7. E. W. Biersack, "A Simulation Study of Forward Error Correction in ATM Networks," *Proceedings ACM SIGCOMM*, pp. 36-47, 1992.

8. K. Joseph, D. Raychaudhuri, and J. Zdepski, "Shared Access Packet Transmission Systems for Compressed Digital Video," *IEEE J. Select. Areas Commun.*, vol. 7, no. 5, pp. 815-825, June 1989.

9. J. Zdepski, K. Joseph, and D. Raychaudhuri, "Packet Transport of Interframe DCT Compressed Digital Video on a CSMA/CD LAN," *Proceedings IEEE GLOBECOM*, pp. 886-892, 1989.

10. K. Joseph, D. Raychaudhuri, and J. Zdepski, "Packet Video Transmission over a Broadband Implicit Token Passing LAN," *Proceedings IEEE GLOBCOM*, pp. 633-639, 1988.

11. CACI Products Company, *Network II.5 User's Manual*, version 5.0, August 1989.

12. S. D. Thomas, "Vector Processor Services for Local Area Networks," M.S. Thesis, Virginia Polytechnic Institute and State University, January 1991.

13. N. Modiri, "The ISO Reference Model Entities," *IEEE Network Magazine*, vol. 5, no. 4, pp. 24-33, July 1991.

14. W. Verbiest, L. Pinnoo, and B. Vosten, "The Impact of the ATM Concept on Video Coding," *IEEE J. Selected Area Commun.*, vol. 6, no. 9, pp. 1623-1632, December 1988.

15. D. G. Morrison, "Variable Bit Rate Video Coding for Asynchronous Transfer Mode Networks," *British Telecom. Tech. J.*, vol. 8, no. 3, pp. 70-80, July 1990.

16. P. Sen, N. Maglaris, and D. Anastassiou, "Models for Packet Switching of Variable Bit-Rate Video Sources," *IEEE J. Selected Areas Commun.*, vol. 7, no. 5, pp.865-869, June 1989.

17. D. L. Mclaren and D. T. Nguyen, "Variable Bit-Rate Source Modelling of ATM-Based Video Services," *Signal Processing: Image Communications*, vol. 4, no. 3, pp. 233-244, June 1992.

18. F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM Networks," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 5, pp. 801-806, June 1989.

19. D. L. Mclaren and D. T. Nguyen, "Removal of Subjective Redundency from DCT-Coded Images," *IEE Proceedings-Part I*, vol. 138, no. 5, pp. 345-350, October 1991.

20. R. Plompen, Y. Hatori, W. Geuen, J. Guichard, M. Guglielmo, and H. Brusewitz, "Motion Video Coding in CCITT SG XV — The Video Source Coding," *Proceedings IEEE GLOBECOM*, pp. 997-1004, 1988.

21. J. Speidel and P. Vogel, "Improved Hybrid Coders with 2D-Signal Processing for Moving Pictures," *Proceedings SPIE Conference: Advances in Image Processing,* vol. 804, pp. 385-394, 1987.

22. T. Kitami and I. Tokizawa, "Cell Loss Compensation Schemes Employing Error Correction Coding for Asynchronous Broadband ISDN," *Proceedings IEEE INFOCOM*, pp. 116-123, 1990.

23. A. J. McAuley, "Reliable Broadband Communications Using a Burst Erasure Correction Code," *Proceedings ACM SIGCOMM*, pp. 287-306, 1990.

24. R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Area Computer Networks," *Communications of the ACM*, vol. 19, no. 7, pp. 395-404, July 1976.

25. S. F. Midkiff, "EE 5516: Computer Networks," Unpublished class notes, April 1993.

26. VMEbus International Trade Association, *The VMEbus Specification*, Printex, Scottsdale AZ, October, 1985.

27. E. H. Frank, "The SBus: Sun's High Performance System Bus for RISC Workstations," *Digest of Papers Compcon Spring*, pp. 189-194, 1990.

28. D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 1992.

29. M. Schwartz, *Telecommunication Networks*, Addison-Wesley, Reading, MA, 1987.

30. P. Vaidyanathan and S.F. Midkiff, "Performance Evaluation of Communication Protocols for Distributed Processing," *Computer Communications*, vol. 13, no. 5, pp. 275-281, June 1990.

31. R. Jain, "Performance Analysis of FDDI Token Ring Networks: Effect of Parameters and Guidelines for Setting TTRT," *Proceedings ACM SIGCOMM*, pp. 264-275, 1990.

32. Computer Science Facilities Group, "Introduction to the Internet Protocols," Rutgers University, 1987.

33. B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Roberts, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Trans. Commun.*, vol. 36, no. 7, pp. 834-843, July 1988.

34. J. S. Turner, "New Directions in Communications (or Which Way to the Information Age?)," *IEEE Communications Magazine*, vol. 24, no. 10, pp. 8-15, October 1986.

Appendix A. Glossary of Simulation Parameters

The models described in Chapter 4 were simulated using NETWORK II.5. Specific values of the input parameters are needed for the hardware and software components. This glossary lists these values and gives brief explanations.

Table A.1 lists the parameters for the SA, SNIU, UNIU, UA, and load, i.e. the background host. The values are obtained from [12] where a detailed explanation can be obtained. In Table A.1, "NR" means "no response," i.e. the default value provided by NETWORK II.5 is used.

Table A.1. Processing Element Parameters

Parameters	SA	SNIU	UNIU	UA	Load
Cycle time (μs/cyc)	0.0625	0.0625	0.08	0.08	0.1
Time slice (ms)	100	100	100	100	100
Input controller	No	Yes	Yes	No	Yes
I/O set-up time (ms)	NR	1.5	1.5	NR	1.5

Table A.2 and Table A.3 list parameters for the transfer devices and network protocol, respectively. Again, the rationale for these values can be found in [12]. It should be noted that the token passing time used in this research is set as NR (0). This is due to the fact that a non-zero value will cause the LAN to be 100 percent utilized over the full simulation period since the LAN is busy passing the token when no message is being delivered. It results in an unacceptably long simulation time.

Table A.2. Transfer Device Parameters

Parameters	SBus	VME Bus	Ethernet	FDDI
Cycle time (μs/cyc)	0.06	0.0625	0.1	0.01
Bits/cyc	32	32	1	1
Cycles/word	1	1	8	8
Words/block	1	1	1518	4500
Block overhead (μs)	0	0	16	2.24

Table A.3 (a). Ethernet Protocol Parameters

Parameters	Value
Collision window	9.6 µs
Contention interval	Uniform distribution: Lower bound = 0 Upper bound = 1024
Retry interval	IEEE backoff: Slot time = 51.2 µs Retry limit = 16
Jam time	3.2 µs

Table A.3 (b). FDDI Protocol Parameters

Parameter	Value
Token passing time	NR
Target token rotation time	8.0 ms

Table A.4 lists the parameters for the *Send Data* instruction. As mentioned in Chapter 4, this instruction is a mix of two instructions that send video data with different data length according to a given probability assignment. It should be noted that K, a shape parameter, is needed in the gamma distribution. The value used is from [5]. Other parameters have been discussed in Chapter 4 in regard to video source models.

Table A.4. Parameters for the *Send Data* Instruction Mix

<table>
<tr><th>Initial time (ms)</th><th>Period</th><th colspan="3">Data length (Bits)</th></tr>
<tr><td rowspan="4">Uniform:
Lower bound = 0
Upper bound = 40</td><td rowspan="4">40 ms</td><td colspan="2">Teleconference</td><td>Broadcast</td></tr>
<tr><td>Probability (%)</td><td>Statistical distribution</td><td>Normal distribution:</td></tr>
<tr><td>99.4</td><td>Gamma:
Mean = 66712
K = 3.066
Lower bound = 12800
Upper bound = 204800</td><td rowspan="2">Mean = 67200
Standard deviation
= 973030
Lower bound = 12800
Upper bound = 1790000</td></tr>
<tr><td>0.6</td><td>Exponential:
Mean = 66712
Lower bound = 204800
Upper bound = 320000</td></tr>
</table>

Besides the statistical distributions addressed previously, an important statistical distribution, Message Linear, is employed as the processing instruction cycle time for the *Packetize Data* and *Depacketize Data* modules. This distribution is of the form Ax + B where x is the number of bits in the received message. By using the mean of the data length in the teleconference video source model, 66,712 bits, as a reference, the time consumed by packetization or depacketization is calculated for Ethernet and FDDI.

The maximum block size is 1,518 bytes in Ethernet and 4,500 bytes in FDDI. The overhead is 46 bytes for an Ethernet frame and 48 for a FDDI frame, both including IP headers. Therefore, the maximum data size is 1,472 bytes in Ethernet and 4,452 bytes in FDDI. By using the same method that is described in [12] for the total packet processing time for UDP, the values of the parameters needed for this statistical distribution are given in Table A.5.

Table A.5. Parameters for the Message Linear Statistical Distribution

Host modules	Statistical distribution: Message Linear	
	Ethernet	FDDI
Packetize Data *Depacketize Data*	A = 0.207 B = 30.410 Lower bound = NR Upper bound = NR	A = 0.164 B = 54.275 Lower bound = NR Upper bound = NR

Appendix B. A Sample Input Data File

The program below is the NETWORK II.5 input data file for a single source system in Ethernet with 5 percent background traffic.

```
* UNTITLED
***** NETGIN RELEASE 6.03
**
***** GLOBAL VARIABLES
GLOBAL FLAGS =
 TEXT SCALE FACTOR = 3  24
 DIAGRAM BOUNDARIES = 0.       365.833     109.750      0.
 ANTITHETIC VARIATE = NO
 RANDOMIZER = 0
 CLOCK = YES
**
***** STATISTICAL DISTRIBUTION FUNCTIONS
STATISTICAL DISTRIBUTIONS =
*Message length of the background traffic
 NAME = NORMAL1
  TYPE = NORMAL
   MEAN = 4900.000
   STANDARD.DEVIATION = 100.000
   LOWER.BOUND = 4400.000
   UPPER.BOUND = 5300.000
*The iteration period of the background traffic
 NAME = NORMAL2
  TYPE = NORMAL
   MEAN = 50000.000
```

```
     STANDARD.DEVIATION = 100.000
     LOWER.BOUND = 40000.000
     UPPER.BOUND = 60000.000
*Start time point for sending data
 NAME = UNIFORM
   TYPE = UNIFORM
     LOWER.BOUND = 0.
     UPPER.BOUND = 40000.000
 NAME = UNFORM1
   TYPE = UNIFORM
     LOWER.BOUND = 0.
     UPPER.BOUND = 1024.000
 NAME = EXPONENTIAL
   TYPE = EXPONENTIAL
     MEAN = 66712.000
     LOWER.BOUND = 204800.000
     UPPER.BOUND = 320000.000
*Time consumed by instructions Packetize Data and Depacketize Data
 NAME = LINEAR1
   TYPE = MESSAGE.LINEAR
     A = .207
     B = 30.410
 NAME = IEEE BACKOFF
   TYPE = IEEE.BACKOFF
     SLOT.TIME = 51.200
     RETRY.LIMIT = 16.000
 NAME = GAMMA
   TYPE = GAMMA
     MEAN = 66712.000
     K = 3.066
     LOWER.BOUND = 12800.000
     UPPER.BOUND = 204800.000
**
***** PROCESSING ELEMENTS
HARDWARE TYPE = PROCESSING
*There are 1500 data frames in total coming from the SA to the SNIU
*99.4 percent of the data follows a gamma distribution, while 0.6 percent of
*the data follows a exponential distribution
 NAME = S.A.
   LOCATION = 208.288      21.260
   STYLE/COLORS = 1   3    15
   BASIC CYCLE TIME = .062 MICROSEC
   INPUT CONTROLLER = NO
```

```
INSTRUCTION REPERTOIRE =
 INSTRUCTION TYPE = MESSAGE
  NAME; GAMMA DATA
   LENGTH; GAMMA
   MESSAGE; SEND DATA
   DESTINATION PROCESSOR; SNIU
   QUEUE FLAG; NO
   RESUME FLAG; NO
   ALLOWABLE BUSSES;
    VME BUS
  NAME; EXPONENTIAL DATA
   LENGTH; EXPONENTIAL
   MESSAGE; SEND DATA
   DESTINATION PROCESSOR; SNIU
   QUEUE FLAG; NO
   RESUME FLAG; NO
   ALLOWABLE BUSSES;
    VME BUS
 INSTRUCTION TYPE = SEMAPHORE
  NAME; SET COUNT
   SEMAPHORE; SET COUNT
   EQUAL TO;   1501
  NAME; CHANGE COUNT
   SEMAPHORE; SET COUNT
   DECREMENT BY;   1
*The packetising time is obtained by the mathod given in thomas' thesis
*Protocol is UDP.
 NAME = SNIU
  LOCATION = 191.497      37.929
  STYLE/COLORS = 1  15    3
  BASIC CYCLE TIME = .062 MICROSEC
  INPUT CONTROLLER = YES
  TIME SLICE = 100000. MICROSEC
  I/O SETUP TIME = 1500. MICROSEC
  INSTRUCTION REPERTOIRE =
   INSTRUCTION TYPE = PROCESSING
    NAME; PACKETISE DATA
     TIME; LINEAR1
   INSTRUCTION TYPE = MESSAGE
    NAME; TRANSFER DATA
     MESSAGE; TRANSFER DATA
     DESTINATION PROCESSOR; UNIU
     QUEUE FLAG; YES
```

```
      RESUME FLAG; YES
      ALLOWABLE BUSSES;
        LAN
*The time for Depacketize Data is calculated in the same way as that for Packetize
*Data
 NAME = UNIU
   LOCATION = 101.822      37.722
   STYLE/COLORS = 1  13    5
   BASIC CYCLE TIME = .080 MICROSEC
   INPUT CONTROLLER = YES
   TIME SLICE = 100000. MICROSEC
   I/O SETUP TIME = 1500. MICROSEC
   INSTRUCTION REPERTOIRE =
     INSTRUCTION TYPE = PROCESSING
       NAME; DEPACKETISE DATA
         TIME; LINEAR1
     INSTRUCTION TYPE = MESSAGE
       NAME; RELAY DATA
         MESSAGE; RELAY DATA
         DESTINATION PROCESSOR; U.A.
         QUEUE FLAG; YES
         RESUME FLAG; YES
         ALLOWABLE BUSSES;
           SBUS
*The number of processing cycles for Read Data is optional, since this module is
*not counted in the time delay
 NAME = U.A.
   LOCATION = 81.815      22.138
   STYLE/COLORS = 1   5    13
   BASIC CYCLE TIME = .080 MICROSEC
   INPUT CONTROLLER = YES
   TIME SLICE = 100000. MICROSEC
   INSTRUCTION REPERTOIRE =
     INSTRUCTION TYPE = PROCESSING
       NAME; READ DATA
         TIME;  1000. CYCLES
 NAME = LOAD1
   LOCATION = 126.831      63.651
   STYLE/COLORS = 1  11    7
   BASIC CYCLE TIME = .100 MICROSEC
   INPUT CONTROLLER = YES
   TIME SLICE = 100000. MICROSEC
   I/O SETUP TIME = 1500. MICROSEC
```

```
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME; SEND MESSAGE
    LENGTH; NORMAL1
    MESSAGE; SEND MESSAGE
    DESTINATION PROCESSOR; LOAD2
    QUEUE FLAG; YES
    RESUME FLAG; YES
    ALLOWABLE BUSSES;
     LAN
NAME = LOAD2
 LOCATION = 158.039      62.919
 STYLE/COLORS = 1   7   11
 BASIC CYCLE TIME = .100 MICROSEC
 INPUT CONTROLLER = YES
 TIME SLICE = 100000. MICROSEC
 I/O SETUP TIME = 1500. MICROSEC
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = PROCESSING
   NAME; RECEIVE
    TIME; 1000. CYCLES
NAME = LOAD3
 BASIC CYCLE TIME = .100 MICROSEC
 INPUT CONTROLLER = YES
 TIME SLICE = 100000. MICROSEC
 I/O SETUP TIME = 1500. MICROSEC
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
   NAME; SEND MESSAGE
    LENGTH; NORMAL1
    MESSAGE; SEND MESSAGE
    DESTINATION PROCESSOR; LOAD2
    QUEUE FLAG ; YES
    RESUME FLAG; YES
    ALLOWABLE BUSSES;
     LAN
NAME = LOAD4
 BASIC CYCLE TIME = .100 MICROSEC
 INPUT CONTROLLER = YES
 TIME SLICE = 100000. MICROSEC
 I/O SETUP TIME = 1500. MICROSEC
 INSTRUCTION REPERTOIRE =
  INSTRUCTION TYPE = MESSAGE
```

```
        NAME; SEND MESSAGE
          LENGTH; NORMAL1
          MESSAGE; SEND MESSAGE
          DESTINATION PROCESSOR; LOAD2
          QUEUE FLAG; YES
          RESUME FLAG; YES
          ALLOWABLE BUSSES;
            LAN
  NAME = LOAD5
    BASIC CYCLE TIME = .100 MICROSEC
    INPUT CONTROLLER = YES
    TIME SLICE = 100000. MICROSEC
    I/O SETUP TIME = 1500. MICROSEC
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME; SEND MESSAGE
          LENGTH; NORMAL1
          MESSAGE; SEND MESSAGE
          DESTINATION PROCESSOR; LOAD2
          QUEUE FLAG; YES
          RESUME FLAG; YES
          ALLOWABLE BUSSES;
            LAN
  NAME = LOAD6
    BASIC CYCLE TIME = .100 MICROSEC
    INPUT CONTROLLER = YES
    TIME SLICE = 100000. MICROSEC
    I/O SETUP TIME = 1500. MICROSEC
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME; SEND MESSAGE
          LENGTH; NORMAL1
          MESSAGE; SEND MESSAGE
          DESTINATION PROCESSOR; LOAD2
          QUEUE FLAG; YES
          RESUME FLAG; YES
          ALLOWABLE BUSSES;
            LAN
**
***** TRANSFER DEVICES
HARDWARE TYPE = DATA TRANSFER
  NAME = LAN
    NAME/MSG LOCATION = 177.039     54.919     177.039     60.919
```

```
  SEGMENTS = 2
    110.039      58.919
    174.039      58.919
  CYCLE TIME = .100
  BITS PER CYCLE = 1
  CYCLES PER WORD = 8
  WORDS PER BLOCK = 1518
  WORD OVERHEAD TIME = 0.   MICROSEC
  BLOCK OVERHEAD TIME = 16. MICROSEC
  PROTOCOL = COLLISION
  RETRY INTERVAL = IEEE BACKOFF
  COLLISION WINDOW = 9.600 MICROSEC
  CONTENTION INTERVAL = UNFORM1
  JAM TIME = 3.200 MICROSEC
  BUS CONNECTIONS =
    SNIU
      KEY = 10.000
      SEGMENTS = 169.703     58.919     198.999     43.075
      STYLE/WIDTH = 1   60
    UNIU
      KEY = 10.000
      SEGMENTS = 118.614     58.919     111.111     43.075
      STYLE/WIDTH = 1   60
    LOAD1
      KEY = 10.000
      SEGMENTS = 141.122     58.919     135.405     64.683
      STYLE/WIDTH = 1   60
    LOAD2
      KEY  10.000
      SEGMENTS = 166.039     58.919     166.039     62.919
      STYLE/WIDTH = 1   60
    LOAD3
      KEY = 10.000
    LOAD4
      KEY = 10.000
    LOAD5
      KEY = 10.000
    LOAD6
      KEY = 10.000
NAME = SBUS
  NAME/MSG LOCATION = 116.815     27.138     116.815     33.138
  SEGMENTS = 2
    81.815      31.138
```

```
    113.815      31.138
   CYCLE TIME = .060
   BITS PER CYCLE = 32
   CYCLES PER WORD = 1
   WORDS PER BLOCK = 1
   WORD OVERHEAD TIME = 0.    MICROSEC
   BLOCK OVERHEAD TIME = 0.    MICROSEC
   PROTOCOL = FIRST COME FIRST SERVED
   BUS CONNECTIONS =
    U.A.
     SEGMENTS = 89.815      31.138      89.815      27.138
     STYLE/WIDTH = 1   60
    UNIU
     SEGMENTS = 107.181      31.138      107.181      37.722
     STYLE/WIDTH = 1   60
  NAME = VME BUS
   NAME/MSG LOCATION = 227.926      26.521      227.926      32.521
   SEGMENTS = 2
    192.926      30.521
    224.926      30.521
   CYCLE TIME = .062
   BITS PER CYCLE = 32
   CYCLES PER WORD = 1
   WORDS PER BLOCK = 1
   WORD OVERHEAD TIME = 0.    MICROSEC
   BLOCK OVERHEAD TIME = 0.    MICROSEC
   PROTOCOL = FIRST COME FIRST SERVED
   BUS CONNECTIONS =
    S.A.
     SEGMENTS = 217.935      30.521      217.935      25.992
     STYLE/WIDTH = 1   60
    SNIU
     SEGMENTS = 197.928      30.521      197.928      37.929
     STYLE/WIDTH = 1   60
**
***** MODULES
SOFTWARE TYPE = MODULE
*set 0 as the time start point
 NAME = SET COUNT
  PRIORITY = 0
  INTERRUPTABILITY FLAG = NO
  CONCURRENT EXECUTION = NO
  START TIME = 0. MICROSEC
```

```
  ALLOWED PROCESSORS =
   S.A.
  INSTRUCTION LIST =
   EXECUTE A TOTAL OF; 1 SET COUNT
*There is a data frame sent out each 40 ms
*99.4% probability to send gamma data, 0.6% probability to send exponential data
**
 NAME = SEND DATA
  PRIORITY = 0
  INTERRUPTABILITY FLAG = NO
  CONCURRENT EXECUTION = YES
  ITERATION PERIOD = 40000. MICROSEC
  START TIME = UNIFORM
  ALLOWED PROCESSORS =
   S.A.
  REQUIRED HARDWARE STATUS =
   VME BUS
    TO BE; IDLE
  REQUIRED SEMAPHORE STATUS =
   RUN UNTIL; SET COUNT
    IS; 0
  INSTRUCTION LIST =
   EXECUTE A TOTAL OF; 1 CHANGE COUNT
   EXECUTE A TOTAL OF;  1 SEND DATA
  COMPLETED IF RUN CANCELLED = NO
 NAME = PACKETISE DATA
  PRIORITY = 0
  INTERRUPTABILITY FLAG = YES
  CONCURRENT EXECUTION = YES
  ALLOWED PROCESSORS =
   SNIU
  REQUIRED MESSAGES =
   SEND DATA
  INSTRUCTION LIST =
   EXECUTE A TOTAL OF; 1 PACKETISE DATA
  ANDED SUCCESSORS =
   CHAIN TO; TRANSFER DATA
   WITH ITERATIONS THEN SKIP COUNT OF 0
 NAME = TRANSFER DATA
  PRIORITY = 0
  INTERRUPTABILITY FLAG = YES
  CONCURRENT EXECUTION = YES
  REQUIRED HARDWARE STATUS =
```

```
      LAN
        TO BE; IDLE
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF; 1 TRANSFER DATA
    ANDED PREDECESSOR LIST =
      PACKETISE DATA
  NAME = RECEIVE DATA
    PRIORITY = 0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = YES
    ALLOWED PROCESSORS =
      UNIU
    REQUIRED HARDWARE STATUS =
      SBUS
        TO BE; IDLE
    REQUIRED MESSAGES =
      TRANSFER DATA
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF; 1 DEPACKETISE DATA
      EXECUTE A TOTAL OF; 1 RELAY DATA
  NAME = READ DATA
    PRIORITY = 0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = YES
    ALLOWED PROCESSORS =
      U.A.
    REQUIRED MESSAGES =
      RELAY DATA
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF; 1 READ DATA
  NAME = SEND MESSAGE
    PRIORITY =          0
    INTERRUPTABILITY FLAG = YES
    CONCURRENT EXECUTION = YES
    ITERATION PERIOD = NORMAL2
    START TIME = UNIFORM
    ALLOWED PROCESSORS =
      LOAD1
    REQUIRED HARDWARE STATUS =
      LAN
        TO BE; IDLE
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF; 1 SEND MESSAGE
```

```
NAME = SEND MESSAGE2
 PRIORITY = 0
 INTERRUPTABILITY FLAG = YES
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = NORMAL2
 START TIME = UNIFORM
 ALLOWED PROCESSORS =
  LOAD3
 REQUIRED HARDWARE STATUS =
  LAN
   TO BE; IDLE
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF; 1 SEND MESSAGE
NAME = SEND MESSAGE3
 PRIORITY = 0
 INTERRUPTABILITY FLAG = YES
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = NORMAL2
 START TIME = UNIFORM
 ALLOWED PROCESSORS =
  LOAD4
 REQUIRED HARDWARE STATUS =
  LAN
   TO BE; IDLE
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF; 1 SEND MESSAGE
NAME = SEND MESSAGE4
 PRIORITY = 0
 INTERRUPTABILITY FLAG = YES
 CONCURRENT EXECUTION = YES
 ITERATION PERIOD = NORMAL2
 START TIME = UNIFORM
 ALLOWED PROCESSORS =
  LOAD5
 REQUIRED HARDWARE STATUS =
  LAN
   TO BE; IDLE
 INSTRUCTION LIST =
  EXECUTE A TOTAL OF; 1 SEND MESSAGE
NAME = SEND MESSAGE5
 PRIORITY = 0
 INTERRUPTABILITY FLAG = YES
 CONCURRENT EXECUTION = YES
```

```
   ITERATION PERIOD = NORMAL2
   START TIME = UNIFORM
   ALLOWED PROCESSORS =
    LOAD6
   REQUIRED HARDWARE STATUS =
    LAN
     TO BE; IDLE
   INSTRUCTION LIST =
    EXECUTE A TOTAL OF; 1 SEND MESSAGE
  NAME = RECEIVE
   PRIORITY = 0
   INTERRUPTABILITY FLAG = YES
   CONCURRENT EXECUTION = YES
   ALLOWED PROCESSORS =
    LOAD2
   REQUIRED MESSAGES =
    SEND MESSAGE
   INSTRUCTION LIST =
    EXECUTE A TOTAL OF; 1 RECEIVE

***** INSTRUCTION MIXES
SOFTWARE TYPE = INSTRUCTION MIX
 NAME = SEND DATA
   INSTRUCTIONS ARE; 99.400 %  GAMMA DATA
   INSTRUCTIONS ARE; .600 %  EXPONENTIAL DATA
```

Appendix C. Time Statistics

I. Single Source System

For Ethernet, only the teleconference video traffic model is considered. For FDDI, both the teleconference and broadcast video traffic models are used.

Table C.1.(a). Time Statistics for Teleconference Video in the Single Source System (μs)

Statistics	*Send Data*		*Packetize Data*		*Depacketize and Relay Data*	
	Ethernet	FDDI	Ethernet	FDDI	Ethernet	FDDI
AVG	140.3	137.5	930.8	725.0	2836.8	2568.6
MAX	405.0	396.8	2684.3	2085.7	5355.5	4575.3
MIN	24.8	24.8	166.2	133.5	1738.4	1696.2
STD	83.8	83.7	555.2	439.4	797.4	618.0

Table C.1.(b). Time Statistics for Broadcast Video in the Single Source System for FDDI (μs)

Statistics	*Send Data*	*Packetize Data*		*Depacketize and Relay Data*	
	Execution	Queuing	Execution	Queuing	Execution
AVG	1467.2	56.4	7703.0	2859.2	12859.1
MAX	3468.2	1237.8	18204.1	26581.7	28345.4
MIN	24.8	0.0	133.5	0.0	1696.2
STD	1267.8	233.7	6653.2	6893.2	9811.6

Table C.2. Time Statistics for the *Transfer Data* Module for Teleconference Video in the Single Source System (μs)

Background in Ethernet (%)	Ethernet				FDDI			
	Queuing		Execution		Queuing		Execution	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
5	19.6	504.5	8872.7	24336.4	.3	48.4	2215.3	3611.4
10	29.0	499.5	8918.2	23567.1	.3	42.9	2215.5	3561.4
15	40.6	508.0	8971.9	23368.6	.5	50.3	2215.6	3561.4
20	52.6	512.9	9041.9	23813.0	.7	48.5	2215.8	3566.6
25	72.4	510.2	9105.8	25413.1	.8	74.7	2215.8	3561.4
30	75.2	521.0	9190.2	24856.8	.9	72.7	2216.0	3600.1
40	102.5	521.3	9431.4	26682.3	1.0	75.5	2216.7	3581.0
50	150.5	518.1	9954.0	27936.0	1.5	96.1	2216.5	3596.1
60	192.4	514.7	11592.8	32788.6	1.4	83.3	2217.1	3561.4
70					1.8	86.3	2217.2	3594.3
80					2.3	139.8	2218.2	3610.3
90					2.9	138.5	2217.8	3590.6
95					2.8	102.6	2218.2	3650.5

Table C.3. Time Statistics for Background Traffic in the Single Source System (μs)

Background in Ethernet (%)	Ethernet		FDDI			
	Teleconference video		Teleconference video		Broadcast video	
	AVG	MAX	AVG	MAX	AVG	MAX
5	3,641.7	44,296.7	1,571.2	5,163.0	3,035.3	54,831.7
10	3,453.2	44,420.9	1,567.2	5,589.0	3,050.7	39,010.6
15	3,575.2	44,958.0	1,569.5	5,634.7	3,043.4	39,012.9
20	3,750.6	46,365.7	1,569.9	5,655.3	3,153.7	39,227.1
25	3,930.4	51,243.9	1,570.3	5,666.9	3,176.5	39,205.3
30	4,266.7	49,577.9	1,571.4	5,724.7	3,246.3	39,294.2
40	5,001.5	63,239.7	1,571.3	5,782.8	3,501.5	223,466.3
50	6,737.3	101,210.4	1,571.0	5,761.5	5,304.0	675,828.3
60	24,443.6	203,692.0	1,572.4	5,812.4	22,023.3	1,165,158.3
70			1,574.0	5,827.1		
80			1,574.2	5,900.0		
90			1,575.7	5,942.9		
95			1,576.6	6,015.5		

II. Multiple Source System

Table C.4 (a). Time Statistics of the *Packetize Data* Module for Teleconference Video with the SS Scheme (μs)

M	Ethernet				FDDI			
	Queuing		Execution		Queuing		Execution	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
2	191.0	12,499.8	916.8	3,585.2	0.0	0.0	725.5	2,109.6
3	2,689.5	38,236.3	907.7	3,771.5	0.0	0.0	724.3	2,661.8
4	35,445.3	158,997.7	915.1	4,108.7	0.1	898.4	721.7	3,041.7
5					1,353.5	7,544.4	739.2	3,257.1
6					783.8	6,797.0	726.1	3,257.1

Table C.4 (b). Time Statistics of the *Packetize Data* Module for Teleconference Video with the MS Scheme (μs)

M	Ethernet				FDDI			
	Queuing		Execution		Queuing		Execution	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
2	0.0	0.0	916.8	3,585.2	0.0	0.0	688.5	2,109.6
3	15.6	10,159.2	907.0	4,037.3	0.0	0.0	728.8	2,661.8
4	372.5	60,336.3	915.1	4,108.7	0.0	0.0	721.7	3,041.7
5					0.0	0.0	717.0	3,257.1
6					0.0	0.0	726.1	3,257.1

Table C.5 (a). Time Statistics of the *Transfer Data* Module for Teleconference Video with the SS Scheme (μs)

M	Ethernet				FDDI			
	Queuing		Execution		Queuing		Execution	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
2	18.6	507.5	8,811.3	29,788.6	0.1	47.6	2,215.8	3,584.8
3	786.7	44,504.0	8,791.8	31,271.8	0.1	49.5	2,214.7	4,132.4
4	31,904.9	155,716.0	8,926.1	33,986.4	0.1	48.9	2,212.2	4,508.3
5					1,524.4	9,700.0	2,229.4	4,719.7
6					470.1	4,720.2	2,216.4	6,864.3

Table C.5 (b). Time Statistics of the *Transfer Data* Module for Teleconference Video with the MS Scheme (μs)

M	Ethernet				FDDI			
	Queuing		Execution		Queuing		Execution	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
2	198.3	17,548.5	8,825.4	29,788.6	0.2	51.0	2,179.2	3,650.3
3	2,046.5	20,705.1	10,054.0	42,926.5	41.4	2,094.3	2,226.3	4,197.7
4	3,056.8	39,802.0	11,800.0	89,120.0	0.1	50.3	2,212.3	4,508.3
5					153.9	4,203.4	2,351.3	5,827.7
6					76.5	3,173.9	2,264.1	5,703.8

Table C.6. Time Statistics of *Packetize Data* and *Transfer Data* Modules for Broadcast Video with the MS Scheme in FDDI (μs)

M	*Packetize Data*				*Transfer Data*			
	Queuing		Execution		Queuing		Execution	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
2	295.7	16,837.3	7712.4	18,204.1	1,533.8	17,944.7	9,520.4	30,445.4
3	1,337.4	39,969.5	7709.4	18,204.1	3,124.1	45,990.3	10,512.8	55,275.5
4	2,597.8	55,661.7	7685.5	18,204.1	7,391.7	69,564.9	13,225.6	57,128.2
5	42,334.9	436,259.2	7759.6	18,204.1	76,608.7	490,080.2	23,975.2	85,326.5

Vita

Jie Lu, was born in Yixin, Jiangsu Province, China on December 23, 1966. She received her high school diploma from Dingshu High School in Yixin in July 1984 and received her Bachelors degree in Communications Engineering of Nanjing Institute of Posts and Telecommunications, China in July 1988. During August, 1988 to August 1991, she worked as a systems engineer for the computing center of Jiangsu Provincial Bureau of Statistics, China, where she developed an interest in computer networks.

She came to Virginia Polytechnic Institute and State University (VPI&SU) to pursue her Master's degree in Electrical Engineering in August 1991 and finished the requirements in October 1993.

She is a member of Gamma Beta Phi honor society. Her personal interests include ancient Chinese poetry, cards, dress designing, sewing, swimming, and excursion.