NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

VIDEO COMMUNICATION OVER INTERNET

Graduation Project
EE- 400

Student: Ziad Hunaiti (991517)

Supervisor: Prof. Dr. Fakhreddin Mamedov

Lefkoşa - 2001

TABLE OF CONTENTS

ACKNOWLEDGMENTS

First I want to thank Prof. Fakhreddin Mamedov, Dean of the Engineering faculty for being my advisor, beside his all other responsibilities, he always helped me a lot either in my study or any problem I faced. What ever I inquired for he always explained to me in details.

Special thanks to Vice-President Assoc. Prof. Dr. Şenol Bektaş and Dr. Tayseer Alshanableh. With their kind help, from the first day I came to this island until my graduation. Really they were very helpful and they always take care of overseas students.

Special thanks for Assoc. Prof. Dr. Adnan Khashman (Chairman of Computer Department), for his valuable advises and great effort, he taught me Neural Networks it was very interesting course. Also he taught me more and more for my future life. So I considered him as my proverbial.

Special thank to Assist. Prof. Dr. Zayed Huneiti (my brother), for his supporting, and encouragement during my study.

I also want to thank my friends in NEU especially my home partners Qais and Jamil, for unforgotten nice days we spent together.

Finally, I want to thank my family, especially my parents; without their endless support and love for me, I would not have achieved my aim. I wish my mother lives happily always, and my father in good health.

ABSTRACT

Sending video over the Internet one of the most important aspect in the Internet world, I have chosen the project to implement the Real-time viewgraph communication system over Internet.

The main aim of this project is to implement the video communication over Internet using the available resources from the MPEG-4 standards. However the communication system is one-way and the focus is on capturing, compressing and sending compressed video to a destination address. In order to achieve the task, my initial preparations were based on exploring and experimenting MPEG-4. MPEG-4 is an ISOIIEC standard developed by MPEG (Moving Picture Experts Group).

This graduation project report covers the Introduction, Internet concept, Audio and Video transmission, MPEG-4 theory, and Protocol literature.

INDRUDACTION

Since invention of Internet, video communication over Internet as well as voice became one of the main issues to be realized.

Because TCP/IP specifications are in the public domain, allowing venders to develop royalty-free software product, which makes it suitable for video communication.

Video signal should be manipulated (digitize, filtering, compressing) in order to achieve efficient and Real time transmission.

The main aim of this project is to implement the video communication over Internet using the available resources form the MPEG-4 standards.

This project consists of introduction, five chapters and conclusion.

Chapter 1 includes of introductory concepts a bout; Moving Picture Experts Group (MPEG), Internet video communication, project overview, video compression techniques.

Chapter 2 covers the history of Internet, understanding structure, services, applications, and future of Internet.

Chapter 3 explains Voice and Video transmission; how video signal manipulated before and after transmission. Multimedia Conferencing. Internet Telephony.

Chapter 4 consists of MPEG-4 Theory, basic concept of MPEG-4 function, MPEG-4 Encoder, MPEG-4 Decoder.

Chapter 5 involves in protocol literature; TCP/IP functions, connection parameters, TCP/IP architecture, User Datagram Protocol (UDP), Networking Concepts, Internet Protocol (IP), Transport Layer, Client / Server Paradigm, Socket Interface, Real Time Applications, Real Time Protocol (RTP), and Real Streaming Protocol (RTSP).

Conclusion presents the significant results, the main requirements for the system and future applications.

CHAPTER ONE
INTRODUCTORY CONCEPTS

1.1 MPEG

The Moving Picture Experts Group (MPEG), which produced the MPEG- 1 and MPEG-2 video and audio compression standards, is at the latest developed the MPEG-4 standard. MPEG-4 video aimed at providing standardized core technologies allowing efficient storage, transmission and manipulation of video data in multimedia environments. This was a challenging task given the broad spectrum of requirements and applications in multimedia. As well as in increased compression efficiency, MPEG-4 also offer content-based functionality, i.e. the possibility of accessing and manipulating individual objects in the picture. Furthermore, MPEG-4 offer possibilities for efficient video storage and for transmission over poor audio and video channels at bit rates between 5Kbitels and 4Mbitels. MPEG-4 became an International Standard in the first months of 1999.

1.2 Internet Video communication

Success of traditional Internet services, such as the electronic mail, the file transfer, remote machine access, has inspired a row of network applications. From the beginning of 90's real-time video has gradually become more and more important as a - communication media in Local Area Networks (LAN), and during the past few years also in the Internet.

Video Over Internet (VOl) is a concept that is gaining rapid acceptance in the Internet community today. In the past with slow modem and marginal quality in the software, the capability to deliver quality video over the Internet just did not exist. An often heard claim is that packet switched networks such the Internet are not suitable for real-time video and audio communication because of its no n-isochroous nature. With today's new generation of high speed modems and the advancement of technologies such as ISDN, ADSL and Internet Over Cable TV, bandwidth availability and

throughput are no longer the constraint. By far the most popular internetworking technology is Internet suite of protocols, commonly referred to as TCP/IP. It is named after two -core protocol (TCP and IP) in the suite.

Figure (1.1) below depicts how video transfer will he achieved over the Internet.

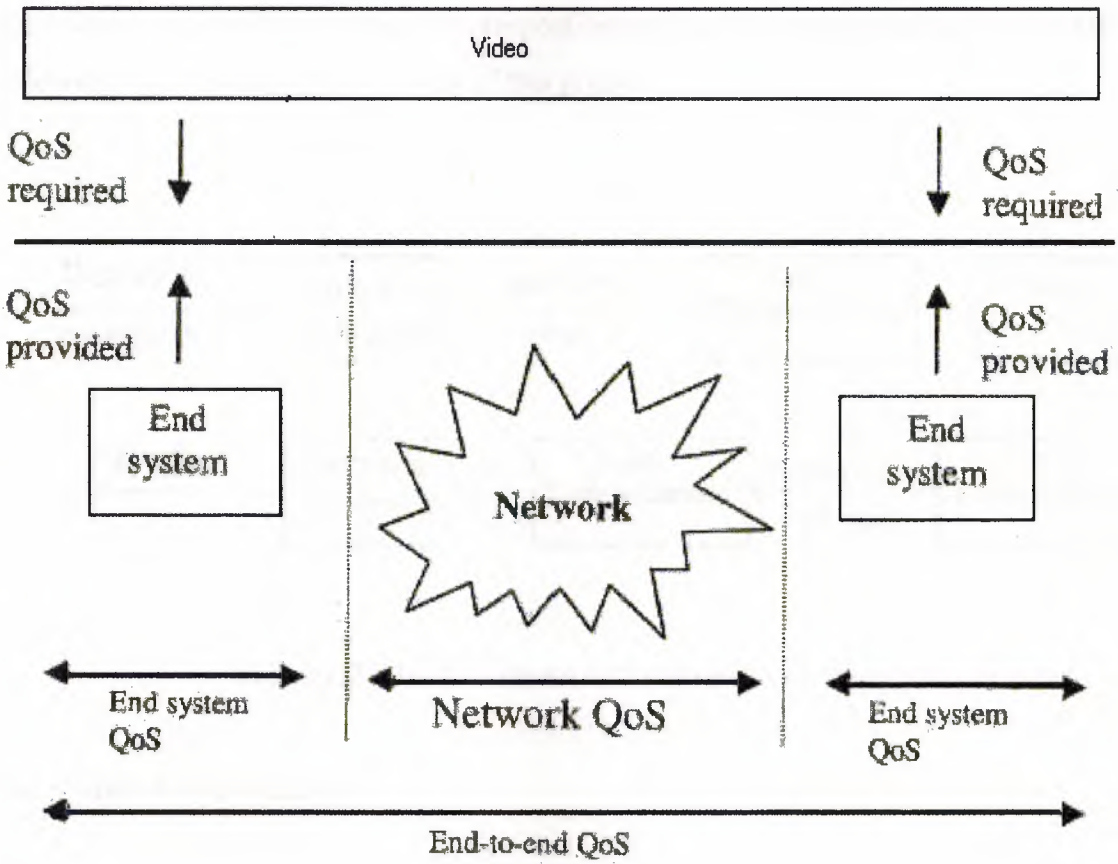


Figure1.1 Transfer of Video Over the Internet

1.3 Project Overview

The project considers the transmission of MPEG-4 video streams over the Internet. The objective of this project is to achieve the transmission in real-time. It is very challenging task to design and implement an optimally running video communication application on a packet switched networks. A good understanding of the whole video data manipulation process, among other media is required as well as good knowledge of properties of transport network and a used operating system. The following figure shows the overview of this project.

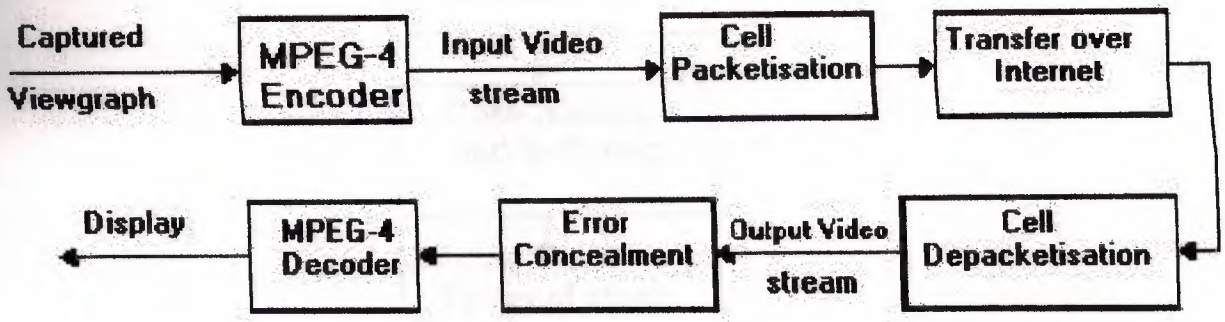


Figure1.2 Block-Diagram of the project overview

1.4 Video Transmission

To give a basic view to a video communication system, a generic functional diagram for a video transmission process in a workstation is shown in figure 1.3.

The first step in the process is to analyze the supplied analogue video signal. The analysis can include such operations as filtering, analogue to digital conversion, computation of transform coefficients, or correlation of the pixels with pre-stored vector quantisation patterns. An output accuracy of such an analysis varies typically from 8 to 12 bits. Usually no compression is done with the analysis. Data is only transformed to a format that is more compressible than the original signal format.

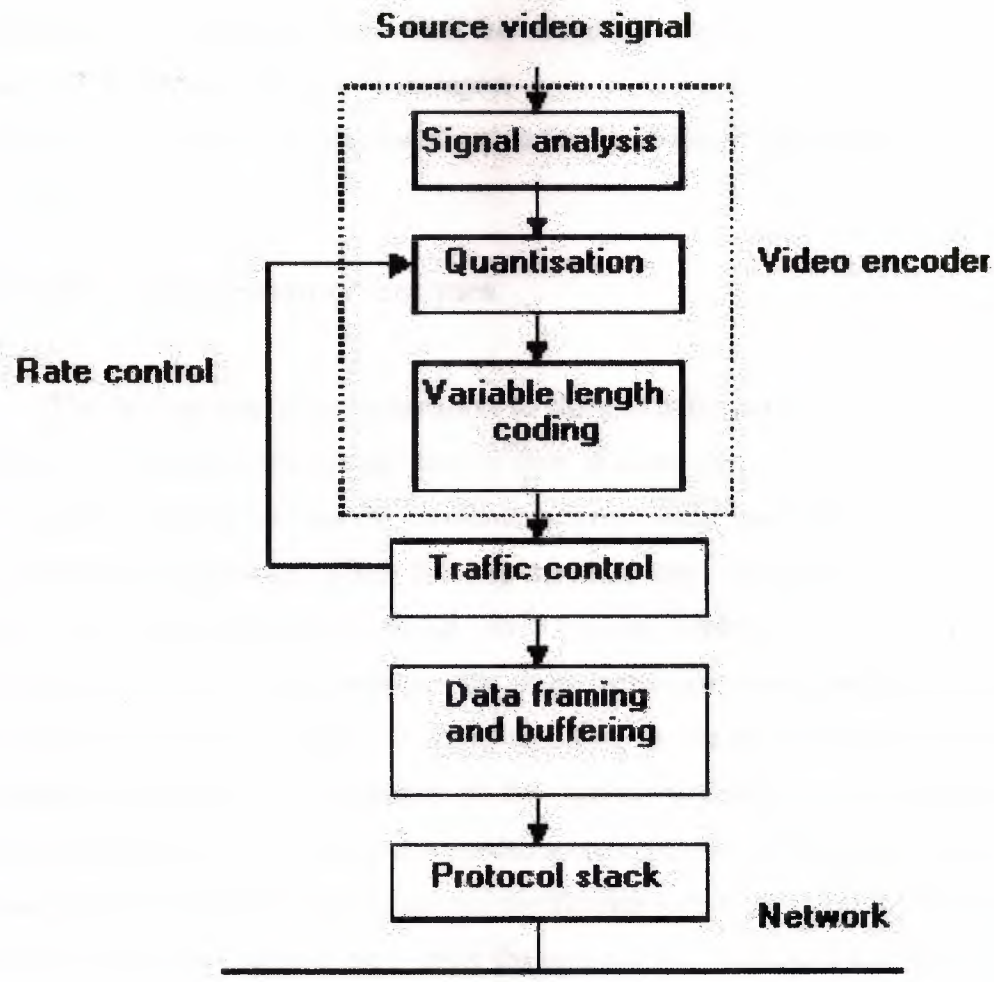


Figure1.3 A generic functional diagram for video data transmission

The second step performs quantisation of the signal, either lossless or lossy way. In a lossy system the quantiser reduces signal accuracy in a way that is as acceptable as possible to the eye. In the variable length-coding block each signal events will have a code with different number of bits. That is why it is also called entropy coding. To get compression, short codes are assigned to frequently occurring events and long codes to infrequent events.

The traffic control block follows data flow status in a communication channel, adjusts encoder parameters (Rate control) according to the data flow status in order to adapt generated video data to the communication channel. The next block forms data packets according to a used protocol. It also buffers the packets in order to serve a continuous and smooth data stream to the communication channel. Such system

parameters as buffer size and packets length are very essential for the system performance, and therefore they must be designed very carefully. If the Real Time Protocol (RTP) is used on top of a transport
protocol layer, it gives necessary information for which the parameter determination can be based on.

1.5 Video compression techniques

The human eye is more sensitive to changes in brightness than to chromaticity changes. Therefore the image data is first divided into one luminance and two chrominance components, and the chrominance components are sub-sampled relative to the luminance component. After this step the usual lossy compression method used in digital video compression is based on Discrete Cosine Transform (DCT) and quantisation. This technique reduces the high spatial frequency components from the image since the human viewer is more sensitive to the reconstruction errors of low frequency components. The purpose of the quantisation step is to represent the DCT coefficients with the precision what is needed to achieve the required image quality. The zigzag step arranges the high frequency coefficients to the end of the stream and since most of them have become zero after the quantisation, run length encoding (RLE) is used for further compression. The upper left corner coefficient represents the mean value of the block and is encoded using the difference from the previous block (DPCM). The final step in the compression process is to minimize the entropy using Huffman or arithmetic coding. The encoded frame is often called I-frame (intra frame) because the encoding process uses no information from other frames. The block diagram of the encoding process is in Figure1.4.

In addition to the previous compression technique, the temporal redundancy between frames can be utilized for further compression. The basic method is to calculate the prediction error between corresponding blocks in the current and previous frames. The error values are then send to the compression process. Compressed frames generated using prediction are usually called P-frames. When using both previous and future frames as reference, the frame is called B-frame (bi-directional frame).

Motion compensated prediction is an efficient tool to reduce temporal redundancy between frames. The concept of motion compensation contains the motion estimation between video frames (Figure 1.5). The motion is described .by a small number of motion vectors which gives the translation of a block of pixels between frames. The motion vectors and compressed prediction errors are then transmitted.

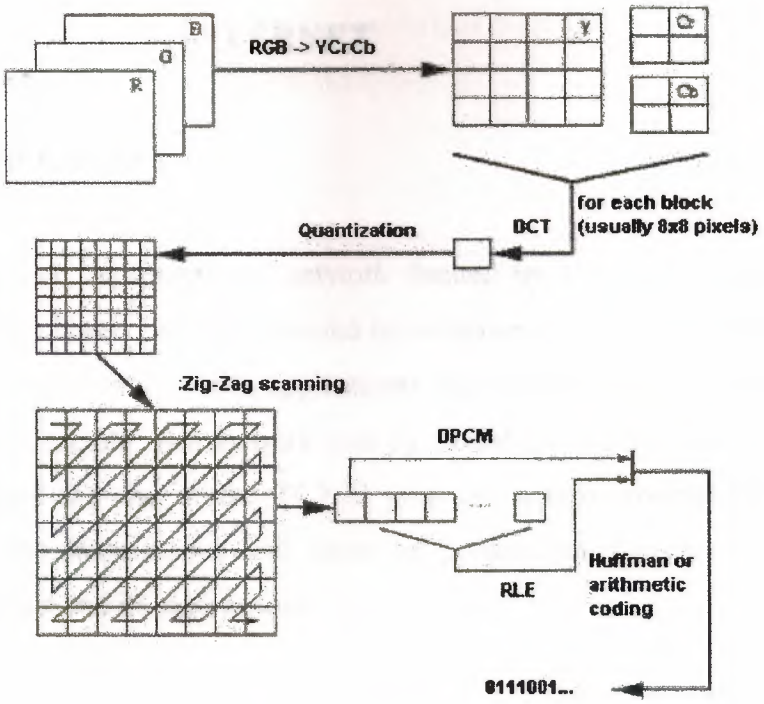


Figure 1.4 Block diagram of video compression

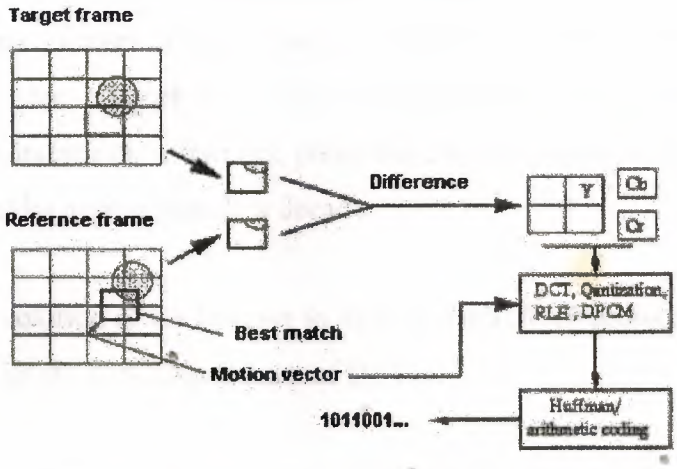


Figure 1.5 Motion compensation

CHAPTER TWO

INTERNET

2.1 About This Chapter

The Internet represents a network formed by the interconnection of sub networks. In this chapter, we will examine its evolution, some of the protocols used to trans-port data, and a few of the applications that resulted in its evolution from an academic-oriented facility to a network used by virtually everyone who has access to a computer. Because the use of the TCP/IP protocol suite is essential for transferring information on the Internet, we will focus on portions of this protocol suite as we examine the Internet and its applications.

2.2 Evolution

During the 1960s, the United States Department of Defense sponsored research that resulted in the development of a communications network to interconnect research laboratories and data centers. Funded by the Defense Advanced Research Projects Agency (DARPA), this network was known as ARPANET. It represented one of the first layered communications networks, preceding the development of the seven-layer ISO reference model by approximately a decade.

Note: The evolution of the Internet to include the TCP/IP protocol suite is based on research funded by the U.S. Department of Defense.

The work involved in establishing ARPANET resulted in the development of two specific protocols for the transmission of information: the Transmission Control Protocol (TCP) and the Internet Protocol (IP) commonly referred to collectively as TCP/IP. TCP represents a transport layer protocol that provides end-to-end reliable transmission. T6 do so, TCP includes such functions as flow control, error control, and the exchange of status information. In comparison, IP represents a connectionless-mode network layer protocol designed to route messages between networks.

To do so, IP includes the capability to segment or fragment and reassemble messages that must be routed between networks that support different packet sizes than the size supported by the source and/or destination network.

In addition to TCP, the Internet suite specifies an optional connectionless-mode layer 4-transport protocol known as the User Datagram Protocol (UDP). UDP is used for transaction-based applications, such as the transmission of network management information when transmission efficiency is more important than reliability.

2.3 Application Services

Various application services have been developed for transport by the TCP/IP protocol suite. The more well-known applications include the File Transfer Protocol (FTP); an interactive remote terminal access protocol known as TELNET, which enables a terminal to be connected to a remote host as if it were directly connected to the computer; the Simple Mail Transport Protocol (SMTP), which provides a standard method for exchanging electronic mail; the Hyper Text Transmission Protocol (HTTP), which enables the transportation of World Wide Web (Www) pages from Web servers to browsers operating on client computers; and the Simple Network Management Protocol (SNMP), which sup-ports the management of network devices.

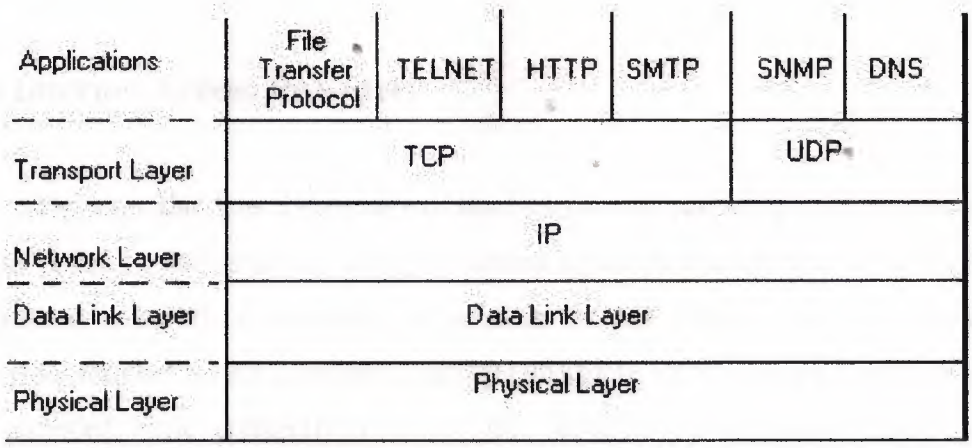


Figure 2.1 A portion of TCP/IP suite.

Figure 2.1 illustrates the layering structure of the TCP/IP protocol suite to include a few of the application services included in the suite. Note that the subdivision of the transport layer in Figure 2.1 is used as a mechanism to indicate which applications are transported by TCP and which are carried by UDP. Later in this chapter, I discuss several of the application services illustrated in Figure 2.1.

One of the key reasons *for* the dramatic growth of the Internet can be traced to the development and structure of the TCP/IP protocol suite. Because TCP/IP was developed using taxpayer funds, its specifications were placed in the public domain, and they are avail-able royalty-free for vendors to develop "protocol stacks" in software to implement TCP/ IP's operational features. The growth of the Internet can also be attributed to the structure of TCP/IP, which makes it suitable for both LAN and WAN operations.

For example, on a LAN, TCP/IP can be transported within Ethernet, Token-Ring, FDDI, or another type of local area network frame. Because a considerable amount of effort was expended in developing LAN adapter cards to support the bus structures used in Macintosh, IBM PCs and compatible computers, Sun Microsystems's workstations, and even IBM mainframes, the development of software-based protocol stacks to facilitate the transmission of TCP/IP on LANs provided the capability to interconnect tens of millions o f LAN based computers to one another. this access normally accomplished by commercial organizations, universities, and government agencies first connecting their networks to the facilities of an Internet access provider.

2.4 Internet Access Providers

Through the late 1980s when the Internet was primarily used to communicate among colleges, universities, and government agencies, most Internet access providers represented nonprofit associations of schools located within a geographical area. As such, the Internet access provider typically constructed a high-speed backbone network that provided basic connectivity within the 'geographical area served, and connected that backbone network to other backbone networks. The rapid growth in the use of the Internet by both individuals and businesses resulted in many nonprofit associations

becoming commercialized, selling Internet access to both individuals and businesses. Most individuals use a dial-up protocol to access the facilities of an Internet access provider, whereas most businesses use leased lines to connect local area networks to the Internet. Because a LAN can provide support for hundreds to thousands of workstations, there is always a high probability that one or more workstation users require access to the Internet at any particular point in time.

Thus, from an economic perspective, it is normally more economical to connect a LAN to the Internet via a leased line than to have individual users use the switched telephone network, because the latter method is commonly billed on an hourly usage basis. In addition, a leased line provides immediate access from the Internet to any application residing on the LAN, such as a Web server, an FTP server, or an electronic mail system. Thus, a leased line connection facilitates bi-directional access to and from the Internet. Figure 2.2 illustrates the two most common methods of Internet access.

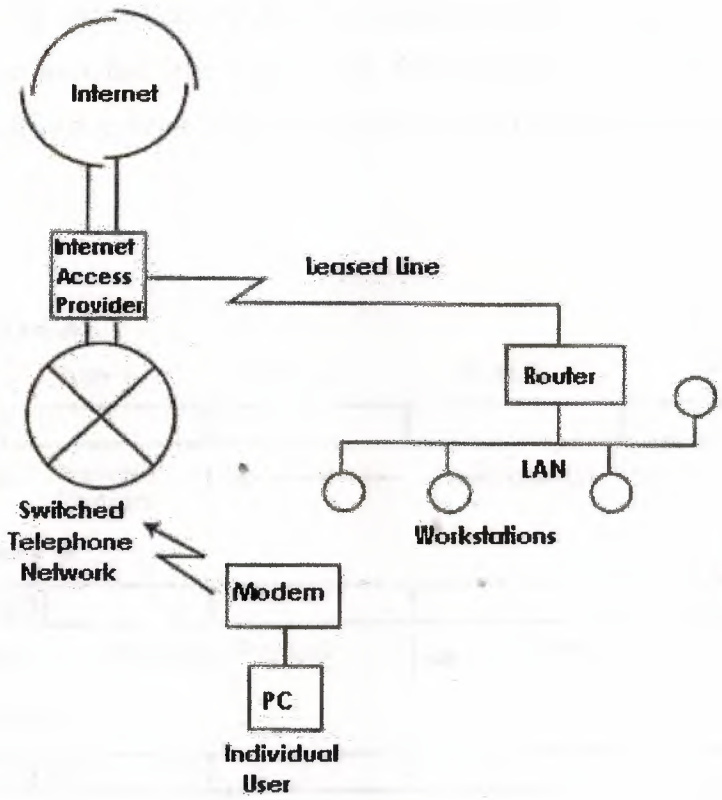


Figure 2.2 common business and individual Internet access methods.

In examining the composition of the IP address classes shown in Figure 12.3, note that the leftmost bits of the address indicate its network class. Also note that the InterNIC assignment of an IP address actually represents the assignment of the network portion of the address, resulting in the manager or administrator of an organization's network that uses an assigned address class being responsible for assigning the host portion of the address.

A Class A address provides the largest range of addresses available for assignment to host computers. Thus, the InterNIC assigns Class A addresses to large organizations and countries that have national networks. Because a Class A address uses the first bit in the first byte as the identifier for this class, a Class A network is restricted to a network number between 1 and 127 for its first decimal number.

A Class B address is evenly split, using two bytes for the network portion and two bytes for the host portion of the address. The availability of two bytes for host computer address assignments enables up to *65,636* hosts to be identified. Thus, Class B addresses are normally assigned to relatively large organizations with tens of thousands of employees.

A Class C address uses three bytes for the network portion and one byte for host identification. Thus, a Class C network address is restricted to supporting a maximum of *256* hosts. This means that a Class C network address is assigned to small organizations, or multiple Class C addresses are assigned to organizations that require more than 256 distinct host addresses but are not large enough to justify a Class B address.

2.6 Domain Name Service

Although IP addresses consist of a sequence of dotted decimal numbers, most readers are probably more familiar with names used to represent Internet addresses, such as www.whitehouse.gov , which represents the address of the World Wide Web server operated by the White House. In this address, www represents the type of service residing on a server, whitehouse represents the name of the organization, and gov indicates that the organization is a government entity.

To facilitate the use of names, each organization registers its name with the InterNIC. The registration process results in the InterNIC assigning the organization to one of six categories referred to as domain names. Thus, the registration process results in a domain name consisting of the organization's name and its domain assignment being registered. In the preceding example, Whitehouse.gov would represent the registered domain name. Table 2.1 indicates the current category of domain name suffixes assigned by the InterNIC.

Table 2.1 Domain name suffixes.

Suffix	*Type of Organization*
COM	Commercial organization
EDU	Educational organization
GOV	Government agency
MIL	Military organization
NET	Networking organization

In examining the entries in Table 2.1, note that the domain name suffix indicates the type of organization. Thus, it is entirely possible for two organizations to use the same server host name and have the same organization name yet have separate identities, be-cause each could represent a different type of organization, resulting in their registered names having different endings.

When an organization has a registered domain name, it can prefix that name to indicate specific hosts or applications residing on one host. For example, www.whitehouse.gov and ftp.whitehouse.gov , which could represent a World Wide Web (WWW) server and a File Transfer Protocol (FTP) server, could also represent two applications residing on a common computer. When you enter the IP address in an application as a name, a translation process occurs that converts the name into a 32-bit IP address. This translation process, which is transparent to a user, occurs through the use of a TCP/IP application known as Domain Name Service (DNS).

DNS commonly operates on a separate server on a network and keeps track of the host names on the network. When a user on the network enters a name rather than an IP address when using a TCP/IP application, the name is transported to the DNS server. If the name resides on the network, the DNS server returns the IP address associated with the name. If the name does not represent a computer on the network, but a recent query using that name occurred, the IP address might be in the DNS server's cache memory. If so, it is returned to the requester. If the name is not in the DNS database or in cache memory, the server sends a request to a "higher" DNS server, usually located on another network. To understand why this occurs requires a slight digression to explain the DNS hierarchy.

A top-level domain registers subdomains in its database in the form of the location of the name servers for those domains. Similarly, subdomains keep track of domains under them, in effect creating a naming hierarchy. Thus, if the name server on the current network does not have the name and associated IP address for the name in its database, it forwards the request to a higher-level DNS. This forwarding effort can be replicated several times and traverse areas around the globe until the IP address is found and returned. It is important to note that DNS names play no role in the actual routing of IP packets. Instead, they simply provide the IP address associated with a domain name. It is the router used to connect networks that is responsible for the actual touting of packets. Because we can get a better feel for the flow of data on the Internet by examining IP and TCP, let's turn our attention to those two protocols.

2.7 The Internet Protocol (IP)

As briefly discussed earlier in this chapter, IP is a network layer protocol. IP provides for the transfer of a basic unit of information referred to as a datagram. In doing so, IP operates as an unreliable, connectionless protocol. Although your first impression of those terms is negative, they are not as bad as they seem. First, although IP provides an unreliable transmission method, the term "unreliable" should be viewed in the context that delivery is not guaranteed.

This means that queuing delays or other problems can result in the loss of data; however, higher layers, such as TCP, can provide error detection and correction that results in the retransmission of IP datagrams. Second, the term "connectionless" references the fact that each datagram is treated independently from preceding and succeeding datagrams. This means IP does not require a connection to be established between source and destination before transferring the first datagram or succeeding datagrams.

The routing of datagrams through a network can occur over different paths, with some datagrams arriving out of sequence from the order transmitted. In addition, as datagrams flow between networks, they encounter physical limitations imposed on the amount of data that can be transported based on the transport mechanism used to move data on the network. For example, the Information field in an Ethernet frame is limited to 1500 bytes. Thus, as a datagram flows between networks, it might have to be fragmented into two or more datagrams to be transported through different networks to their ultimate destination.

Figure 2.4 illustrates the routing of two datagrams from workstation A on a Token Ring network to server B connected to an Ethernet LAN. Because the routing of datagrams is a connectionless service, no call setup is required, which enhances transmission efficiency. In comparison, when TCP is used, it provides a connection-oriented service regardless of the lower layer delivery system (for example, IP).

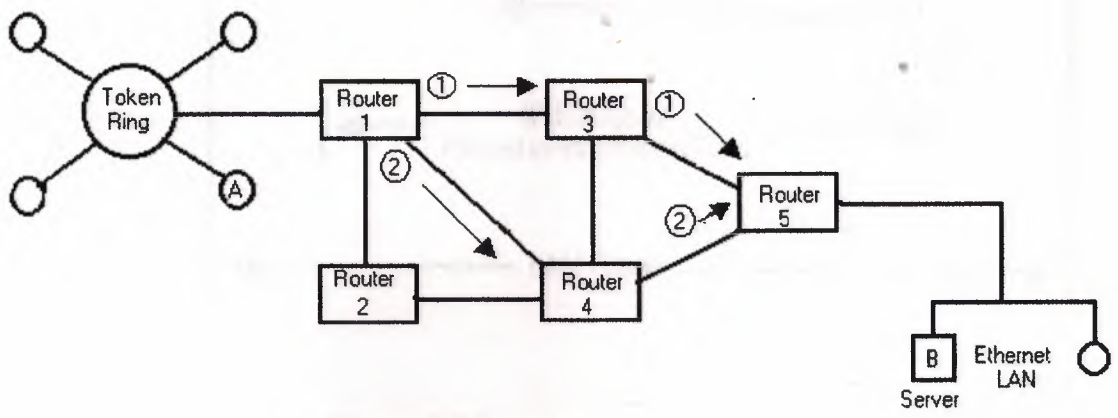


Figure 2.4 Routing of datagrams can occur over different paths.

TCP requires the establishment of a virtual circuit in which a temporary path is developed between source and destination. This path is fixed and the flow of datagrams is restricted to the path established. When UDP is used in place of TCP, the flow of data at the transport layer continues to be connectionless and results in the transport of datagrams over available paths rather than a fixed path resulting from the establishment of a virtual circuit.

Figure 2.5 illustrates the relationship of an IP datagram, a UDP datagram, and a TCP segment to a LAN frame. Note that the IP datagram flowing on a 4 Mbps Token-RingLAN can be up to 4500 bytes in length because that is the maximum length of a Token -Ring Information field.

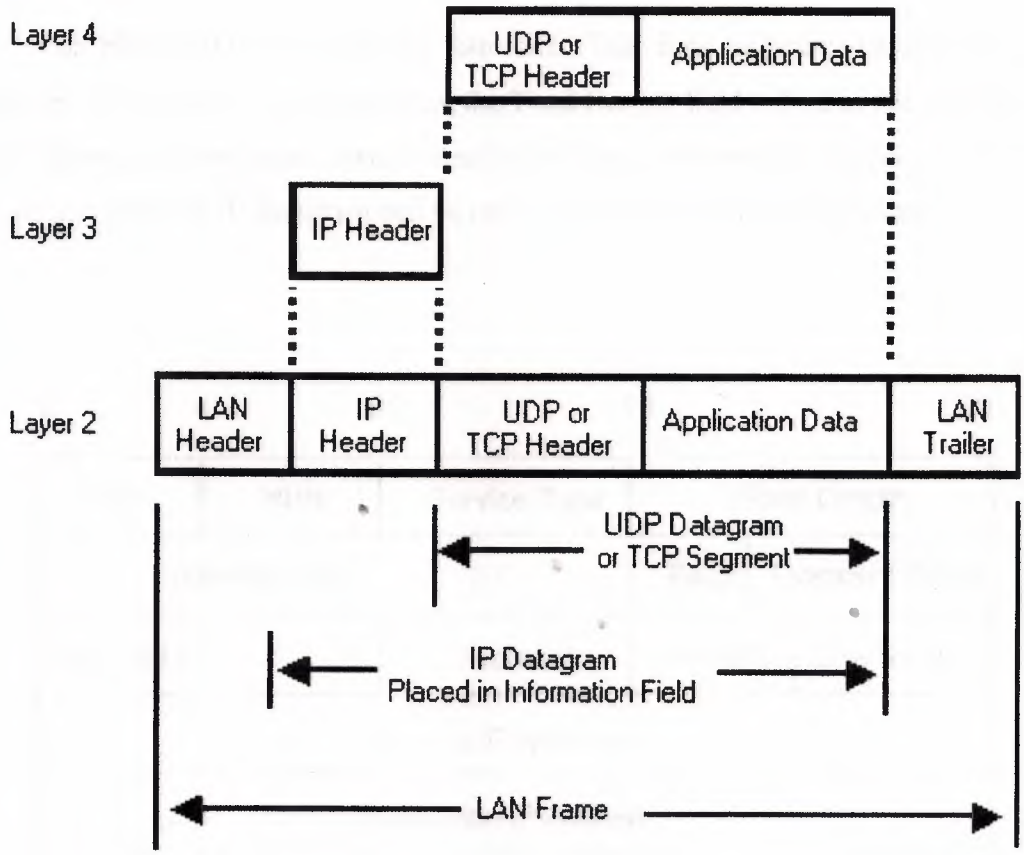


Figure 2.5 Forming a LAN frame.

However, when that datagram flows onto an Ethernet LAN whose maximum Information field length is 1500 bytes, the original datagram might have to be fragmented into thirds.

We cart gain an appreciation for how datagram fragmentation is accomplished) as well as additional information about the flow of data, by examining the composition of the fields in the IP header.

Figure 2.6 illustrates the fields contained within the IP header. Note that the header contains a minimum of 20 bytes of control information) with Figure 2.6 illustrating the width of each field with respect to a 32-bit word.

The Vers field consists of four bits that identify the version of the IP protocol used to create-the the datagram. The current version of the IP protocol is 4.

The Hlen field also contains four bits. This field indicates the length of the header in 32-bit words. In comparison, the Total Length field indicates the total length of the datagram to include its header and higher layer information. Because 16 bits are used for this field, an IP datagram can be up to 216, or 65,535 octets in length.

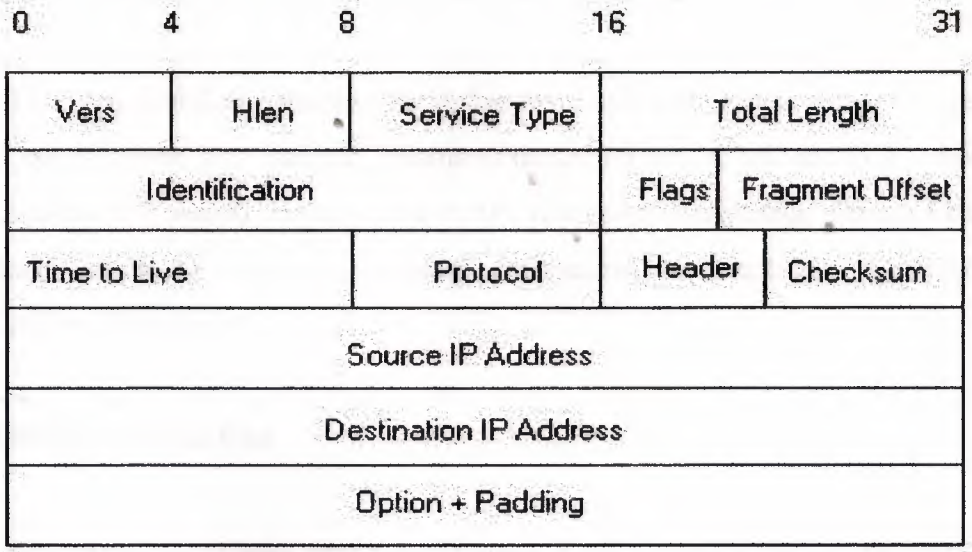


Figure 2.6 The format of the IP header.

The Service Type field defines how the datagram is handled. Three of the eight bits in this field are used to denote the precedence or level of importance assigned by the sender. Thus, this field provides a priority mechanism for routing IP datagrams.

The Identification field enables each datagram or fragmented datagram to be identified. If a datagram was fragmented, the Fragment Offset field specifies the offset in the original datagram of the data being carried. In effect, this field indicates where the fragment belongs in the complete message. The actual value in this field is an integer that corresponds to a unit of 8 octets, providing an offset in 64-bit units.

The Flags field contains two bits that indicate how fragmentation occurs, and a third bit is currently unassigned. The setting of one bit can be viewed as a direct fragment control mechanism because a value of 0 indicates that the datagram can be fragmented, whereas a value of 1 denotes that it can't be fragmented. The second bit is set to 0 to indicate that a fragment in a datagram is the last fragment, and it's set to a value of 1 to indicate that more fragments follow the current protocol.

The Time to Live (TTL) field specifies the maximum time that a datagram can live. Because an exact time is difficult to measure, many routers decrement this field by 1 as datagram flows between networks, with the datagrarn being discarded when the field value reaches zero. You can consider this field to represent a fail-safe mechanism because it pre vents misaddressed datagrams from continuously wandering the Internet.

The Protocol field specifies the higher-level protocol used to create the message carried the datagram. For example, a value of decimal 6 would indicate TCP, whereas a value; decimal 17 would indicate UDP. The source and destination address fields are both bits in length. As previously discussed, each address represents both a network and a computer on the network.

2.8 Address Resolution

The physical address associated with a LAN workstation is referred to as its hardware address. For an Ethernet network, that address is 48 bits or 6 bytes in length.

In this chapter, I noted that at the data link layer, IP uses a 32-bit logical address. One common problem associated with the routing of an IP datagram to a particular workstation on a network is the delivery of the datagram to its correct destination. To correctly deliver the datagram requires knowledge of the relationship between physical and logical addresses. This relationship is obtained by two protocols that map the physical and logical addresses to each other. One protocol, known as the Address Resolution Protocol (ARP), translates an IP address into a hardware address. The Reverse Address Resolution Protocol (RARP), as its name implies, performs a reverse mapping, converting a hardware address into an IP address.

To illustrate the use of ARP, assume that one computer user wants to send a datagrarn to another computer and both computers are located on the same Ethernet network. The first computer broadcasts an ARP packet within an Ethernet frame to all devices on the LAN. That packet contains the destination IP address, because it is known, and sets the hardware address field to zeros, because its value is unknown. Although each device on the LAN reads the ARP packet as it is transmitted as a broadcast packet, only the device that recognizes its own logical address responds. When it does, it transmits an ARP reply in which its physical address is inserted in the ARP address field previously set to zero. To reduce the necessity to constantly transmit ARP packets, the originator records received information in a table known as an ARP cache, allowing subsequent datagrams to be directed quickly to the appropriate address on the LAN. Thus, ARP and RARP provide a well-thought-out methodology for equating physical hardware addresses to IP's logical addresses.

2.9 TCP

The Transmission Control Protocol (TCP) is a Layer 4 connection-oriented protocol. This protocol is responsible for providing reliable communications between hosts and processes on different hosts. Concerning the latter, TCP is structured to enable multiple application programs on a host to communicate concurrently with processes on other hosts, as well as for a host to demultiplex and service incoming TCP traffic among different applications or processes running on the host. To accomplish this task, a TCP header includes a destination port number to identify the ultimate destination in a

computer. To gain an appreciation for the functionality and capability of TCP, let's turn our attention to its header, which is illustrated in Figure 2.7.

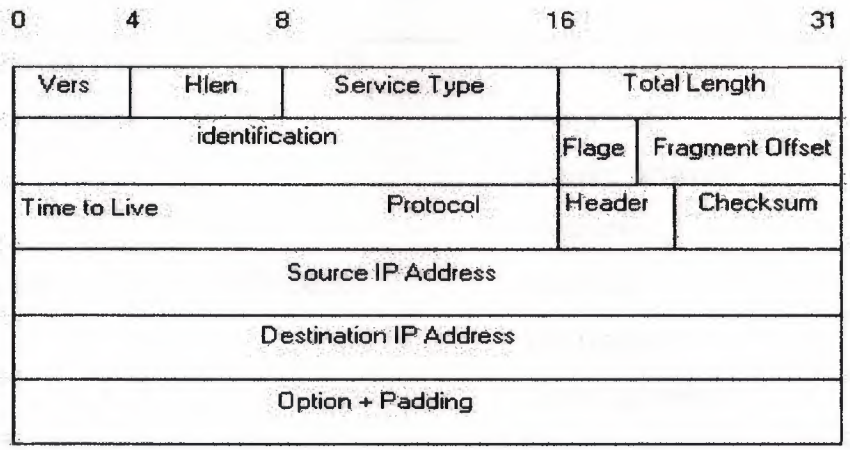


Figure 2.7 The TCP header.

The Source and Destination Port fields are each 16 bits in length and are used to identify a user process or application. The Source Port field is optional and when not used is padded with zeros. The term "well-known port, which is commonly used to denote an application layer protocol or process, actually references the port address used within a TCP header. Table 2.2 lists the well-known ports associated with eight popular TCP/IP application layer protocols. In examining the entries in Table 2.2, you will note that some protocols, such as FTP, use two port addresses or logical connections. One address (21) is for commands and replies and functions as a control path. The second port address (20) is used for the actual file transfer.

Table 2.2 Popular TCP/IP application layer protocols.

Name	*Acronym*	*Description*	*Well-Known Port*
Domain Name Protocol	DOMAIN	Defines the DNS	53
File Transfer Protocol	FTP	Provides file transfer between computers	20, 21
Finger Protocol	FINGER	Provides Information About specified User	79
Hypertext Transmission Protocol	HTTP	Conveys information between a Web Browser and a Web server	80
Post Office Protocol	POP	Enables PC users To access mail From a mail server	110
Simple Mail Transfer Protocol	SMTP	Provides electronic mail Transfer	25
Simple Network Management Protocol	SNMP	Provides the exchange of Management Information	161,162
TELNET Protocol	TELNET	Provides remote Terminal access	23

If you reexamine Figure 2.5 in conjunction with Figures 2.6 and 2.7, you will note that the use of a TCP/IP application requires three addresses at both source and destination. A port address is required to identify the process or application and is contained within the TCP header. Within the IP header, an IP address is required to identify the network and host computer where the process or application resides. Finally, the delivery of information on a LAN requires the use of a hardware address, which is used within the LAN header shown in Figure 2.5 to deliver the IP datagram.

Returning to our examination of the TCP/IP header, the Sequence Number field identifies the position in the sender's byte stream of the data transported in the TCP segment. Thus, this field provides a mechanism to maintain the sequentially of the data stream.

The Acknowledgment Number field identifies the number of the octet (Sequence Number) that the source expects to receive next. Thus, the Acknowledgment Number verifies the receipt of the prior n-1segment when the Sequence Number is *n*.

The Hlen field, which is four bits in length, denotes the length of the segment header in 32-bit multiples. The Code Bits field contains six flag bits. Some of those bits, when set, indicate that a specific field in the header is significant and the field value should be interpreted, whereas other bits are used to control the connection and data transfer operation. Table 2.3 summarizes the use of the six Code Bits.

Table 2.3 Code Bit field values.

Code Bit	*Code Bit Field Use*
URG	Urgent Pointer field significant
ACK	Acknowledgment field significant
PSH	Push function
RST	Reset the connection
SYN	Synchronize Sequence numbers
FIN	No more data from sender

The 1 6-bit Window field indicates the number of octets, beginning with the one in the Acknowledgment field that the originator of the segment can control. Because TCP rep-resents a full-duplex communications path, each end of the path can use the Window field to control the quantity of data being sent to it. This provides the recipient with the ability to, in effect, have a say over its destiny. That is, if the recipient becomes overloaded with processing or if some other situation results in its inability to receive large chunks of data, it can use the Window field to reduce the size of the chunks of data being sent to it.

The Checksum field provides reliability for the TCP header, the IP header, and data carried in the segment. Thus, this field provides the mechanism for the detection of errors in the segment.

Other than some options beyond the scope of this book, the Urgent Pointer field completes the header. This field enables the position of urgent data within a TCP segment to be identified, and a value in the field is interpreted only when the previously mentioned URG bit is set. When that bit position is set, the value in the Urgent Pointer field indicates the beginning of routine (non-urgent) data.

To illustrate the interrelated role of the Sequence, Acknowledgment, and Window fields, let's examine the transmission of data between two hosts via the use of a time chart that indicates some values for each field. This time chart is shown in Figure 12.8. At the top of Figure 2.8, we will assume that a Window size of 8 segments is in use. Although TCP supports full-duplex transmission, for simplicity of illustration we will use a half-duplex model in the time chart.

Assuming that host A is downloading a program or performing a lengthy file transfer, the first series of segments will have sequence numbers 64 through 71. Assuming that no errors occurred, host B returns an ACK value of 72 to indicate the next segment it expects to receive. Let's also assume that host B is running out of buffer space and that it reduces the window size to 4.

Thus, host A uses the Window field value in the TCP header sent to it and reduces the number of units of data it will transmit to four, using an initial SEQ field

value of 72 and increasing that value by 1 to 75 as it transmits four units of data to host B. Assuming that all data is received error-free, host B then returns an ACK value of 76 to acknowledge receipt of sequence field numbers through 75.

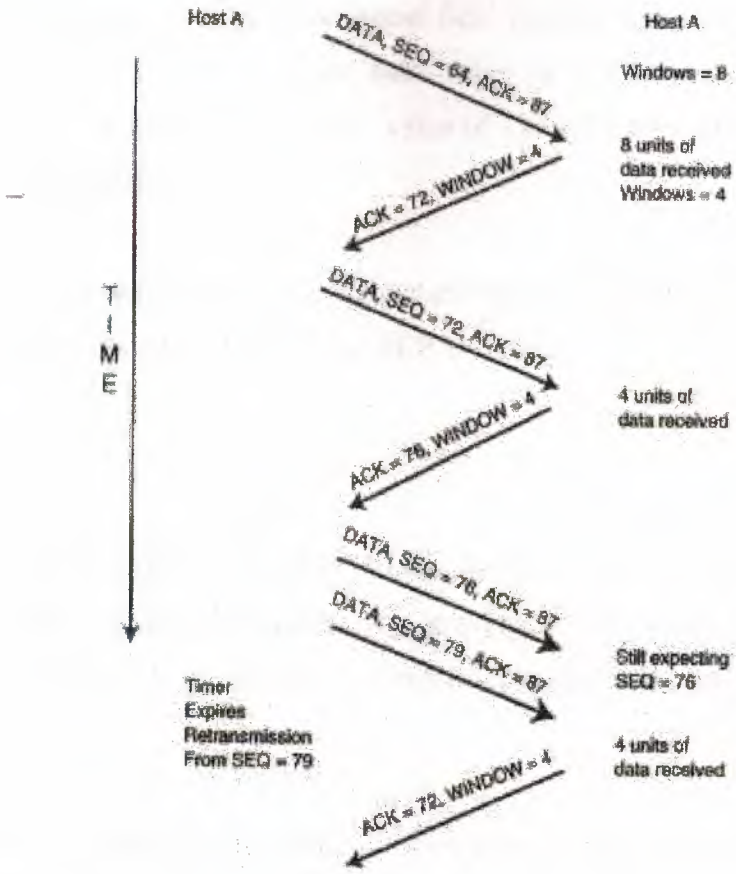


Figure 2.8 A TCP transmission sequence.

Once again host A transmits to host B, this time using sequence field values of 76 to 79. As this transmission occurs, however, let's assume that some type of transmission impairment occurs that sends the data into the proverbial bit bucket so that it is never received at host B. Because host B does not receive anything, it does not transmit anything back to host A. Although host A could wait forever, this would not be a good idea when data becomes lost. Instead, an internal timer clicks down to zero while host A waits for a response. When a response does not appear and the timer expires, host A retransmits the segment, which is then acknowledged at the bottom of Figure 2.8.

The altering of the Window field values provides a "sliding window" that can be used to control the flow of information by adjustment of the value of the Window field.

In doing so, there are two special field values, 0 and 1, that further control the flow of information. A Window field value of 0 means a host has shut down communications, whereas a Window value of 1 requires an acknowledgment for each unit of data transmitted.

Now that you have a basic understanding of IP and TCP, let's turn our attention to a few of the applications that use the TCP/IP suite.

2.9.1 FTP

The File Transfer Protocol (FTP) was developed as a mechanism to facilitate the transfer of files between computers. As previously mentioned, FTP uses two "well-known ports" port 21 for passing control information, and port 20 for the actual data transfer.

FTP supports approximately 20 commands. Those commands enable a user to change directories **(cd)**, obtain a directory list **(dir)**, initiate a file transfer **(get)**, and transfer a file **(put)**. FTP permits multiple file transfers with the **mget** and **mput** commands when used with a filename containing one or more wildcard characters. For example, the command **mget *.gif** would result in the transfer of all files in the current directory that have the extension **.gif.**

One of the key advantages associated with the use of FTP is the fact that various implementations exist that operate on a range of computers, from DOS PCs to IBM mainframes. This enables FTP to provide a mechanism to exchange files between computers as long as both computers support FTP and can be reached via a TCP/IP connection. Concerning the TCP/IP connection, FTP relies on TCP at the transport layer to provide a reliable transmission path, ensuring the error-free arrival of data at its destination.

2.9.2 TELNET

TELNET is an interactive remote access terminal protocol that was developed to enable users to access a remote computer as if they were directly connected to the computer. Similar to FTP, TELNET is based on a client/server model. Although TELNET was developed to provide terminal connectivity to hosts, the client does not have to be a physical terminal, such as one of the popular Digital Equipment Corporation's VT products. Instead, TELNET client programs have been developed that turn PCs, Macintoshes, and even various IBM, Sun, and HP workstations into an interactive terminal.

2.9.3 HTTP

Perhaps the most popular Internet TCO/IP application is hyper Text Transmission Protocol (HTTP) that is used to convey information between a Web browser and a Web server. Here the term browser represents a software product that uses HTTP to transport information and supports the display of information encoded using the Hyper Text Markup Language (HTML).

The growth in the use of the Internet can be greatly attributed to the growth of the world Wide Web, the unstructured collection of Web server containing text, graphics, and audio files whose contents can be viewed and heard through the use of browser.

CHAPTER THREE
VOICE AND VIDEO TRANSMISSION

3.1 Introduction

MBone stands for the Virtual Multicast Backbone On the interNEt. MBone is a technology that enables distribution and access to real time interactive multimedia on the Internet. Multimedia is a combination of one or more media that might include text, graphics voice and video. Where voice and video media can be digital or analogue. In normal circumstances, the human voice is analogue, as are television/VCR video signals. Analogue audio and video signal are continuous in nature-they convey meaning only when presented continuously in time.

MBone technology forms the backbone of the Internet for distributing real-time multimedia information to millions of computer desktops worldwide. The Internet Protocol (IP) is the primary networking protocol used to connect computer network word wide. It is the protocol used by Internet, which connect data networks in more than 50 countries across the seven continents. A computer network is a communication system that connects a set of computing subsystem together. The computing subsystems, also called hosts, could be a single user computer such as a DOS or Windows machine, a multi-user Unix workstation, or a network printer. Computer networks enable communication between such hosts.

3.2 The MBONE

The MBONE is a network; also it is a multicast backbone over some of the high-speed parts of the Internet that provides multi-way communication in a way that scales to many participates. Multicasting is the ability of a sender to send a single date packet to a number of receivers in such a way that the packet only transverses each network once MBone is layered on top of portions of the physical Internet to support routing of IP multicast packets since that function has not yet been integrated into many production routers. The network is composed of islands that can directly support IP multicast, such as multicast LANs like Ethane, linked by virtual point-to-point links

called "tunnels". The tunnel end points are typically workstation-class machines having operating system support for IP multicast and running the "mrouted" multicast routing daemon. Figure 3.1 shows the main system components for video Internet transmission.

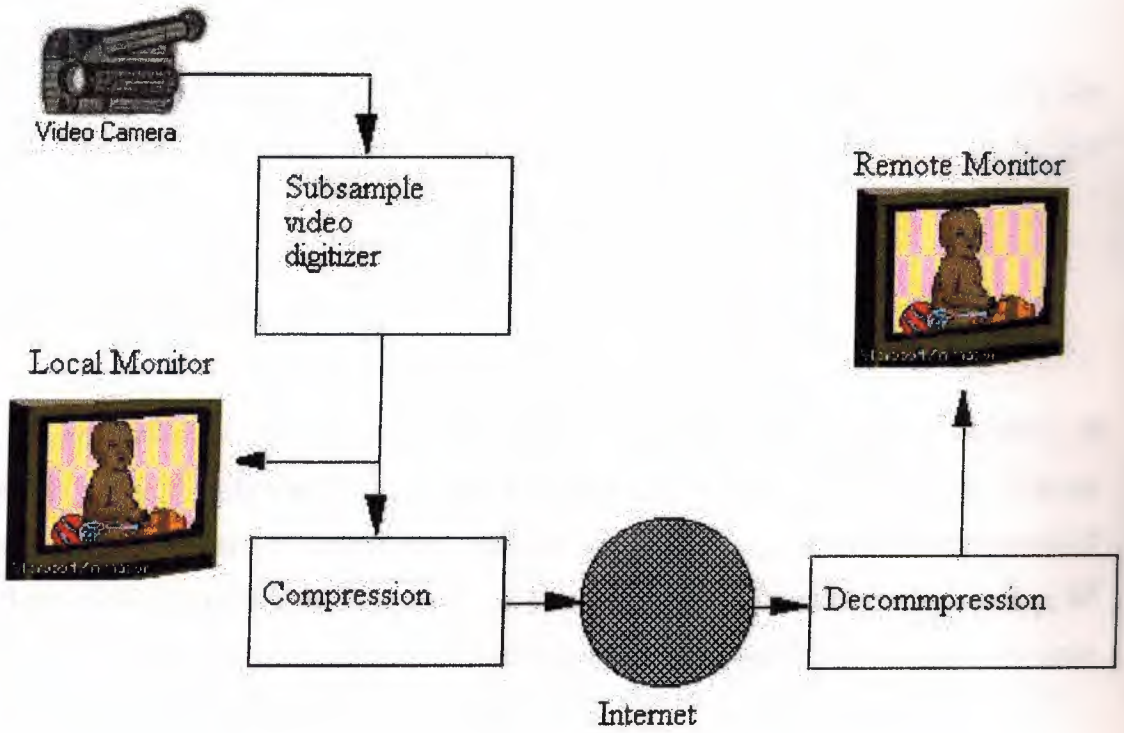


Figure 3.1 System components for Internet video transmission

Voice and video signals are usually processed separately. This is because different coding schemes provide different benefits for speech and non speech sources. In both cases the drive is to achieve acceptable fidelity and real time transmission, while reducing the bit rate, minimizing the delay and improving the robustness. The audio and video applications are briefly discussed below.

3.3 MBone Video

Most MBone application that use video communication use a video tool called Vic (Video Conference), from Lawrence Berkeley labs (LBL) in the USA. Vic offers a variety of compression algorithms, at many different image sizes. The frame rate and resolution can be varied; although a common bit rate and frame is 2 frames/s at

128kbit/s. There is currently no robustness to packet loss in any MBone video tool. To make use of the conferencing capabilities, the system must support IP multicast and, ideally, the network should be connected to the MBone.

3.4 Visual Audio Tool

The most popular application on MBone is LBL Visual Audio Tool (VAT). Vat enables two or more Internet hosts to participate in voice based conferencing, much like a normal phone.

3.5 The Role of Mrouters

IP multicast routers, referred to as Mrouters, take the responsibility of distributing and replicating the multicast data stream to their destination. The MBone topology of mrouters is designed in such a manner that it facilitates distribution of packets without congesting any node or network link inappropriately. In this way, IP multicast based routing facilitates distributed hosts to achieve time critical, real time communication over wide area IP network.

3.6 Operation of IP Multicast Tunnels

IP multicasting, as the name suggests, enables distribution of IP packets to one or more Internet sites, it does not imply a general broadcast to all the Internet hosts. IP multicast packets are encapsulated for transmission through tunnels, so that they look like normal unicast datagrams to intervening routers and subnets. A multicast router that wants to send a multicast packet across a tunnel will prepped another IP header, set the destination address in the new header to be the unicast address of the multicast router at the other end of the tunnel, and set the IP protocol field in the new header to be 4 (which means the next protocol is IP). The multicast router at the other end of the tunnel receives the packet, strips off the encapsulating IP header, and forwards the packet as appropriate.

Early versions of the IP multicast software (before March 1993) used a different method of encapsulation based on an IP Loose Source and Record Route option. This method remains an option in the new software for backward compatibility with nodes that have not been upgraded. In this mode, the multicast router modifies the packet by appending an IP LSRR option to the packet's IP header. The multicast destination address is moved into the source route, and the unicast address of the router at the far end of the tunnel is placed in the IP Destination Address field. The presence of IP options, including LSRR, may cause modern router hardware to divert the tunnel packets through a slower software-processing path, causing poor performance. Therefore, use of the new software and the IP encapsulation method is strongly encouraged.

3.7 The MBone Topology

The MBone topology on the Internet is a tree and a mesh. The connections between mrouters of major Internet service provider nodes constitute the mesh topology. It is anticipated that within a continent, the MBONE topology will be a combination of mesh and star: the backbone and regional (or mid-level) networks will be linked by a mesh of tunnels among mrouted machines located primarily at interconnection points of the backbones and regional. Some redundant tunnels may be configured with higher metrics for robustness. Then each regional network will have a star hierarchy hanging off its node of the mesh to fan out and connect to all the customer networks that want to participate.

Between continents there will probably be only one or two tunnels preferably terminating at the closest point on the MBONE mesh. In the US, this may be on the Ethernets at the two FIXes (Federal Internet eXchanges) in California and Maryland. But since the FIXes are fairly busy, it will be important to minimize the number of tunnels that cross them. This may be accomplished using IP multicast directly (rather than tunnels) to connect several multicast routers on the FIX Ethernet.

3.8 Hardware Requirements to Support the Audio and Video Applications

Most machines now support the audio and video applications, because they have sound and video cards built in. No additional hardware is required to receive audio and video on those systems that have audio built in because the rest is done in software. To send audio requires a microphone; to send video requires a camera and video capture device, which are only built-in on a few of the systems. For the camera, any camcorder with a video output will do. The wide-angle range is most important for monitor-top mounting. There is also a small (about 2x2x5 inches) monochrome CCD camera suitable for desktop videoconference applications. Generally all that is required on top of a standard PC is a camera and video card, and a headset audio card.

3.9 Operating System Requirements

One can run the audio and video applications point-to-point between two hosts using normal unicast addresses and routing, but to conference with multiple hosts, each host must run an operating system kernel with IP multicast support. IP multicast invokes Ethernet multicast to reach multiple hosts on the same subnet; to link multiple local subnets or to connect to the MBONE one needs a multicast router as described above.

From figure 3.2, it can be seen that to transmit a video data on the internet, the signal needs to be manipulated twice:

- Before transmission (known as local manipulation).
- After receipt (known as remote manipulation).

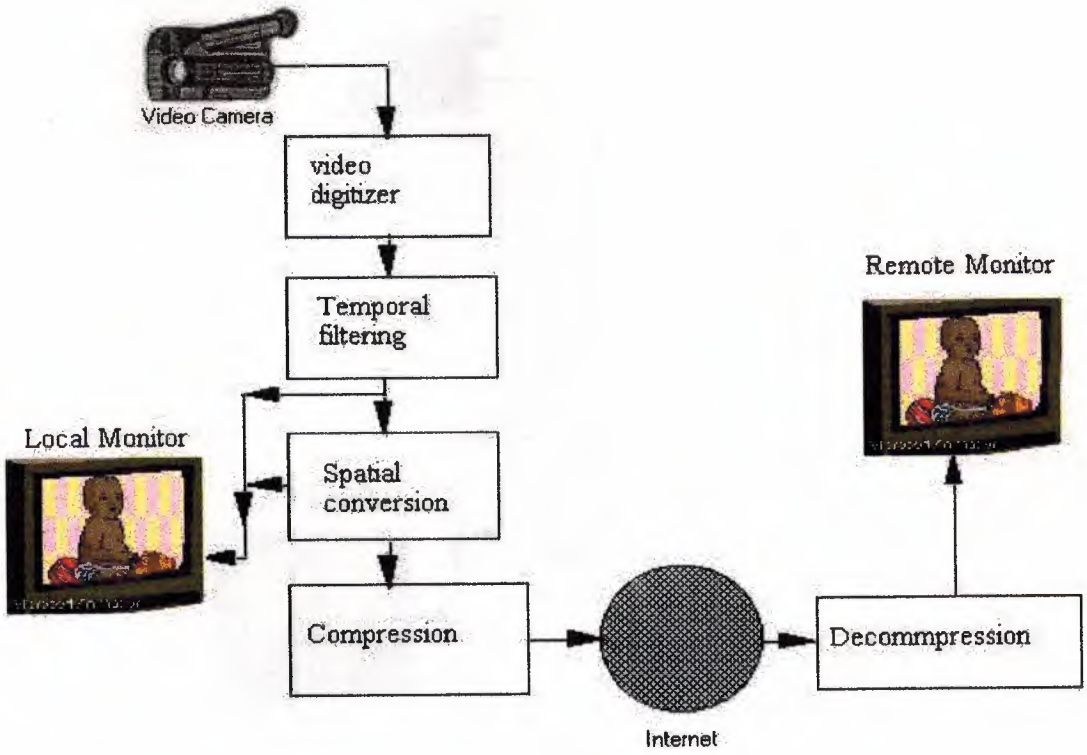


Figure 3.2 Manipulation of video signal to be transmitted on the Internet.

3.9.1 Local Transformation of Video Images

Local manipulation and transformation of video images, prior to sending them through the Internet, can be carried out as portrayed in figure. The video sequence can be either grabbed or loaded from disk. The video camera captures the images, which are in the form of an analogue signal, and then the captured images are fed into a video digitizer. The video digitizer digitizes the analogue video signal and produces digital video images as the output signal. The main principle is shown below in figure 3.3.

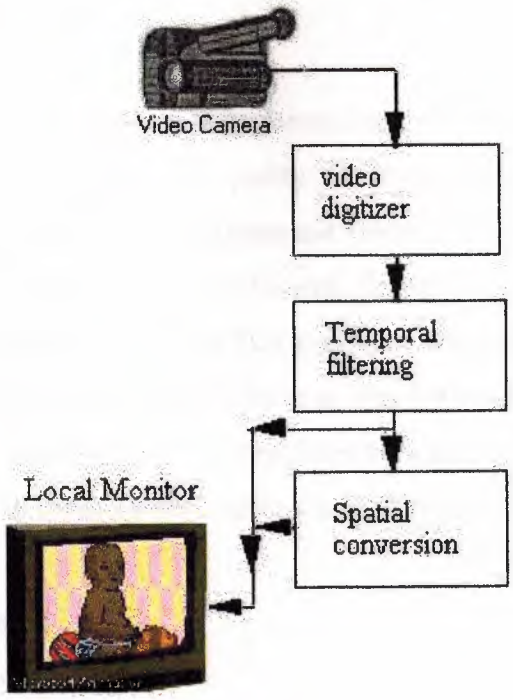


Figure 3.3 Manipulation of video signal before transmission on the Internet.

3.9.2 Converting Analogue to Digital Signals

An understanding of how analogue signals work is essential to understanding digital signals. A digital version of real-world audio and video signal is obtained by feeding the analogue signal into an Analogue-to-Digital Converter (ADC), as shown in figure3.4. The output of such a digitizer is a digital audio/video signal. Not all digital audio or video is equal to, or necessarily better than, its analogue counterpart. The ultimate digital sound or video quality depends on sample rates and resolution the faster the sample rates and the more bits used to determine the value of the samples, the better the quality.

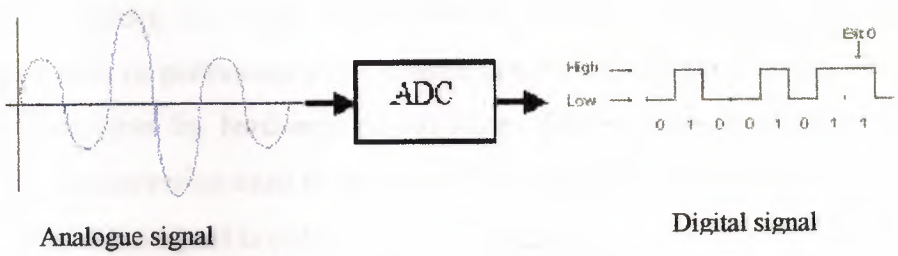


Figure 3.4 An ADC showing analogue to digital conversion

3.9.3 Signal Filtration

Filtration of real time picture is performed before transmission of video data on the Internet, in order to improve the quality of images, also to adapt the required frequency for presentation. The performance requirements for temporal filtering demand conversion of raster zed video images frame rate to a frame rate that makes possible to combine the video and the PCs graphics raster images with a minimum of temporal aliasing. Temporal filtering works as a filter between input (video) frame rate and output (graphics) frame rate. It interpolates between different input frames. This type of filtering is used to reduce jerky motions and permits better display quality.

3.9.4 Spatial Conversion

Spatial conversion is used to provide the following functions:

-Arbitrary placement of resulting video window on the screen.

- Scaling of the video images by arbitrary factor.

- Mapping of the video image to an arbitrary surface.

Basically spatial conversion is used to provide a scalable display size. The performance of the spatial conversion unit is defined by the requirement of the real time capability.

3.9.5 Compression of Video Signal

The digital transmission of uncompressed multimedia data requires networks with extremely high transfer rates. While this is not globally available, and because the digital data is generated in very large quantities, it's thus needs to be compressed for more efficient transport over the Internet. Therefore, sophisticated compression techniques to reduce the large amount of data required for digital representation of video signal must be performed prior to sending the data over the Internet. Compression is carried out either by hardware or software. Since a property of video data is time dependency, compression must be performed in real time. Once the above functions are carried out the video signal is transmitted on the Internet.

3.9.6 Manipulation of Video Images after Receipt

Figure shows the configuration for remote video manipulation. This is a reverse process of the local manipulation of the signal. However, in this case the temporal filtering is performed between decompression and display of the receiving PC. It works as a frame rate up converter to generate the frame rate necessary for display for the frame rate transmitted via the Internet. Therefore, it can be said that at the receiving side, temporal and spatial conversion are performed and the resulting image is displayed as depicted in figure3.5.

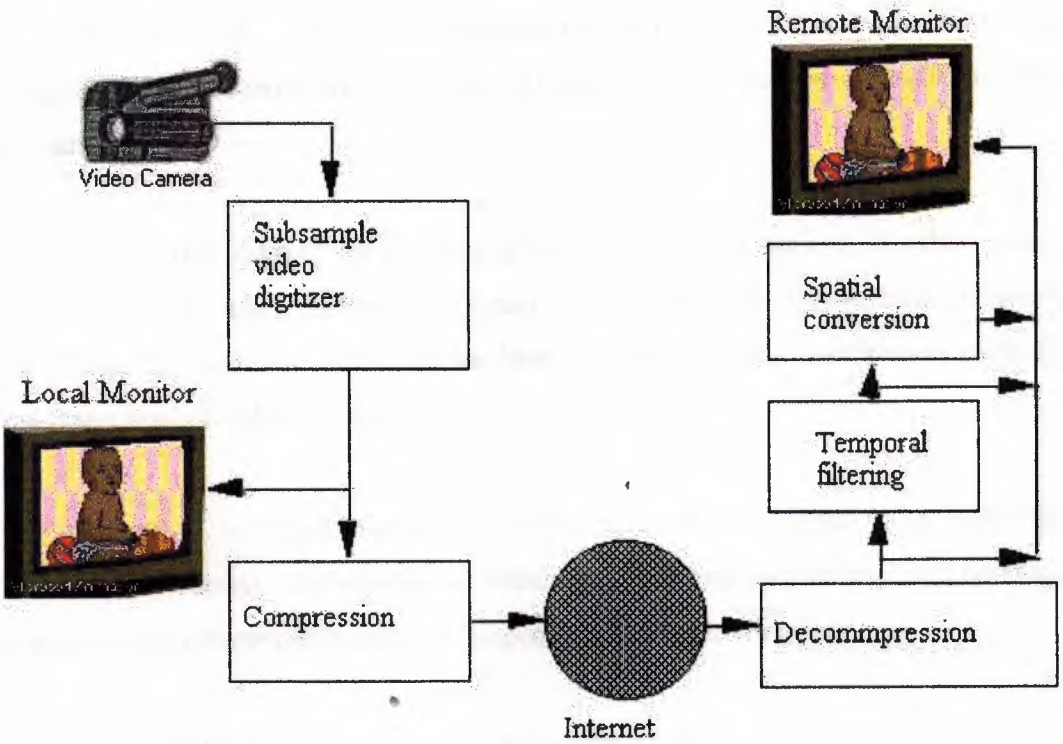


Figure 3.5 Manipulation of video signal after transmission on the Internet.

3.10 Multimedia Conferencing

The transmission and reception of multimedia data (voice and video) streaming simultaneously between various hosts is called conferencing. Nowadays most computer machines support audio and video hardware and software. As a result more and more users are beginning to share multimedia information, using the model shown in figure 6. Figure 2 illustrates how continues media like a video signal enters the system at Host A and gets delivered to Host B over computer net work in almost real time. The same model can then be extended to voice signal as well. A video Camera/ VCR provides the continuous video feed. This analogue video information is fed directly into a Frame Grabber and Encoder hardware such as a Video Blaster card on PCs or a Sun Video card on Sun machine. These video hardware boards digitize the analogue video signal into digital data, followed by encoding. Encoding is how fast the analogue incoming data is sampled.

Because this digital data is generated in very large quantities, it's thus needs to be compressed for more efficient transport to Host B over the network or internet. Compression is carried out either by hardware or software. The compressed video data is now transmitted by Host A to Host B.

At Host B, a reverse process helps to recover the video signal in its digital form on the computer screen. The speed with which a video signal can be rendered on Host B after it has been transmitted by Host A, depends on the following factors:

1) Speed (CPU and operating systems) of each host involved.
2) Speed of video encoding and decoding hardware.
3) Speed of compression and decompression.
4) Speed of the network to which each host is connected.

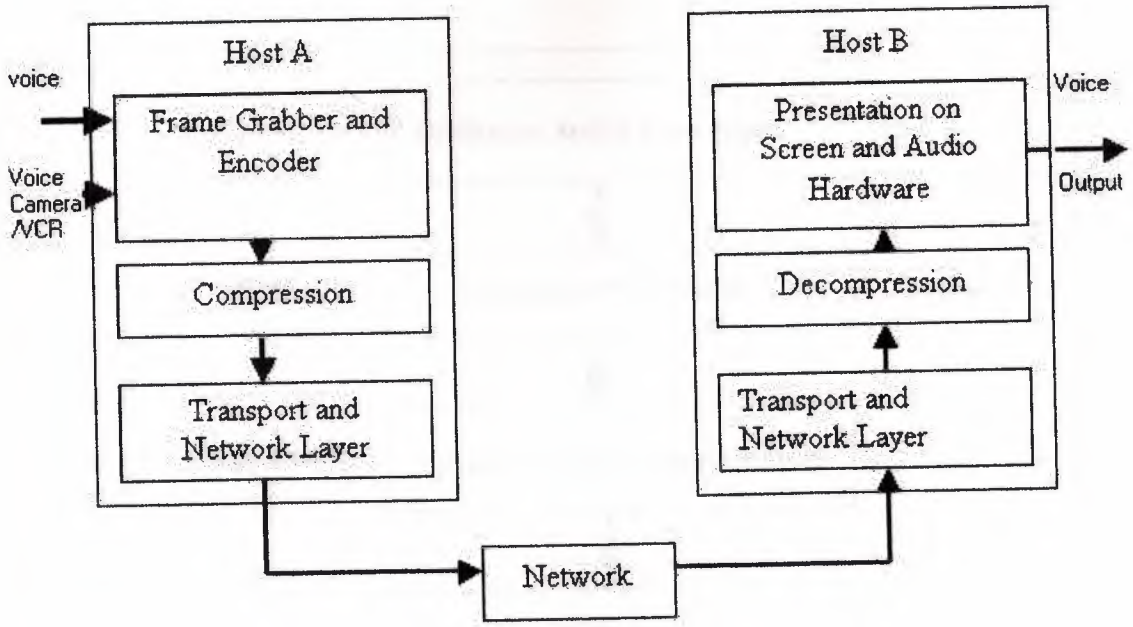


Figure3.6 The system components for audio and video network conference.

3.11 Internet Telephony

Internet telephony, or Voice over Internet Protocol (VoIP), is the provision of phone service over the Internet. Voice traffic is carried as data packets over a packet switched data network. The Internet does not establish any path, it delivers the data piecemeal, in packets, each of which traverses the net independent of all the others, finding its way much the way a letter finds its way through a postal system- on the basis of addresses and other control information contained in the header. In fact, every packet in an Internet data stream may take a different path, in order to make the best use of the networks resources, and the packets may even arrive at the receiving end in quite another order from that in which they were transmitted. At the receiving end, the packets are stored, sorted into the proper order, and passed on to the recipient. Users can make telephone calls, on the Internet, using a multimedia PC with its microphone and speaker system. A modem connection to the Internet Service Provider (ISP) is used in this procedure. The method requires both the originating PC and the intended receiver to be logged onto the internet before a voice call can be made., because the PC software needs to map the telephone numbers to active IP addresses.

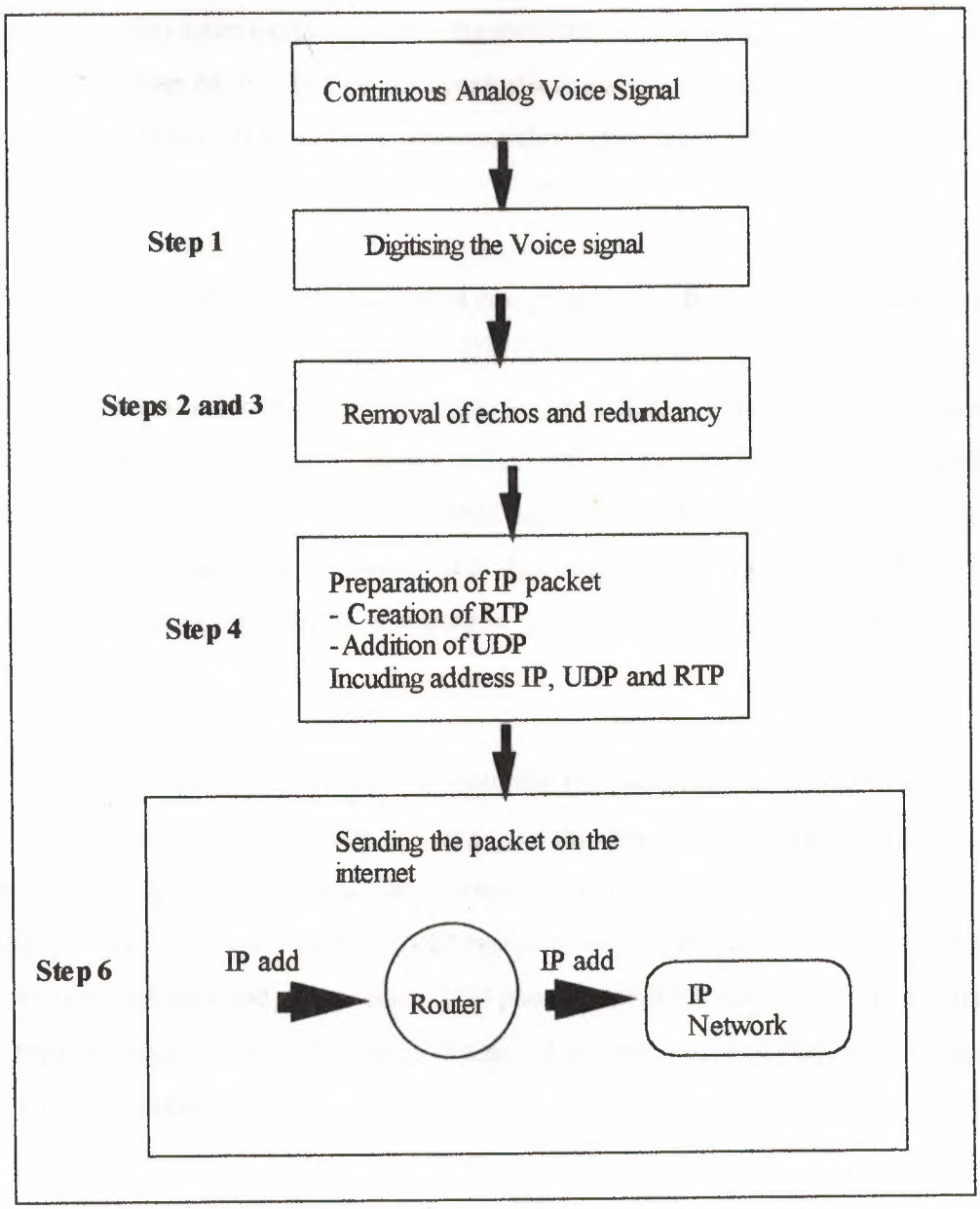


Figure3.7 Sending voice on the Internet; Internet telephony

Delivering voice from one point to another in a voice over internet protocol (VoIP) system can be thought of as a six step process, as shown in figure 3.7. In step 1, analogue voice signals generated in a telephone handset are sampled 8000 times a second and coded into a 64-kb/s-bit stream.

Step 2 involves processing the stream with a digital filtering algorithm to remove any echoes. At this step, the bit stream is also analyzed for silent periods using a Voice Activity Detection (VAD) algorithm. When silence is detected, suppressing it and

not blindly coding it into a lengthy string of zeroes save bandwidth. The fact that a silent interval has been suppressed may be explicitly communicated to the receiving end or time sampling of the data packets may allow a silence to be inferred. In either case, the system will then fill the interval with comfort noise so the listener will not think that the line has gone dead.

Step 3 is where the bit stream is compressed and framed on the basis of several International Telecommunication Union (ITU) Standards.

In step 4, the voice frame is converted into an IP packet, a process that itself takes three steps. The first of these is to create a Real time Transport Protocol (RTP) packet by adding a 12 byte header to the compressed voice frame. Then an 8 byte User Datagram Protocol (UDP) header with source and destination socket numbers is added. Finally a20 byte IP header containing the source and destination gateways IP addresses is added.

At step 5, the IP packets is sent onto the Internet, just like any other data packet. Routers and switches along the way examine the destination IP address, route it in the correct direction, and deliver it to the destination IP address.

Finally, in step 6, the destination VoIP system reverses the process for voice playback. The system extracts the IP packet, UPD packet, and RTP packet, and then extract the compressed voice frame. The voice frame is decomposed and converted to analogue form for voice playback.

CHAPTER FOUR

MPEG-4

4.1 MPEG-4 Theory

The MPEG video compression algorithm has been developed with respect to the H.261 standard and so it retains a large degree commonality with it. However, many macroblocks need information not in the previous frame when P-frames are predicted. The MPEG solution to this is a new frame type, B-frame (bi-directional frame) which is predicted both the previous and the next P- or I-frame as reference. A typical encoded sequence is:

IBBPBBPBBIBBPBB.... Actual pattern is up to an encoder.

4.1.1 Overview of the MPEG-4

The focus of the MPEG-4 video group is the development of the Video Verification Models. A Verification Model (VM) is a common platform with a precise definition of encoding and decoding algorithms, which can be presented as tools addressing specific functionality. It evolves through time by means of core experiments. New algorithms or tools are added to the VM and old algorithms or tools are replaced in the VM by successful core experiments.

It is so happened, that the traditionally distinct areas of film or television, communications, and computers start to overlap, developing new needs for standardization. During the analysis of future applications in these areas, MPEG-4 identified five requirements as the driving force of MPEG-4:

- **Interactive:** The user should be able to influence the presentation of audio/visual content.
- **Content-based:** An object-based data representation should allow contents based access to multimedia data.
- **Network independent:** Access to MPEG-4 data and communications should be possible using any communications network, e.g. mobile networks, telephone, internet and broadband network.

- **Flexible:** MPEG-4 data streams should be scalable such that receivers with different levels of computational power can be able to process them.
- **Extensible:** The transmitter should be able to configure the receiver in order to download new applications and algorithms.

MPEG-4 standardizes a rich set of audio-visual data types. These include natural and synthetic voice, audio and as well as video. These data types will allow new applications especially in the area of interactive audio-visual systems. The object-based coding will allow the manipulation and presentation of objects and not necessarily rectangular images. The standard provides tools for an efficient coding of natural and synthetic 2D and 3D audio-visual objects and syntax for describing complete animated scenes. It is foreseen that MPEG-4 terminals will be built according to standardized profiles and levels defining the required subset of MPEG-4 capabilities that a terminal has to provide. Several profiles for applications like real-time communications, content-based storage and retrieval, surveillance, and broadcast are defined.

The basic concept of these functionalities are illustrated in Figure 4.1 The scene consists of a background and several foreground object, which are encoded in such a way that the receiver can decode the individual objects separated and manipulate them in present time. For example, they can be mixed with objects from a second scene.

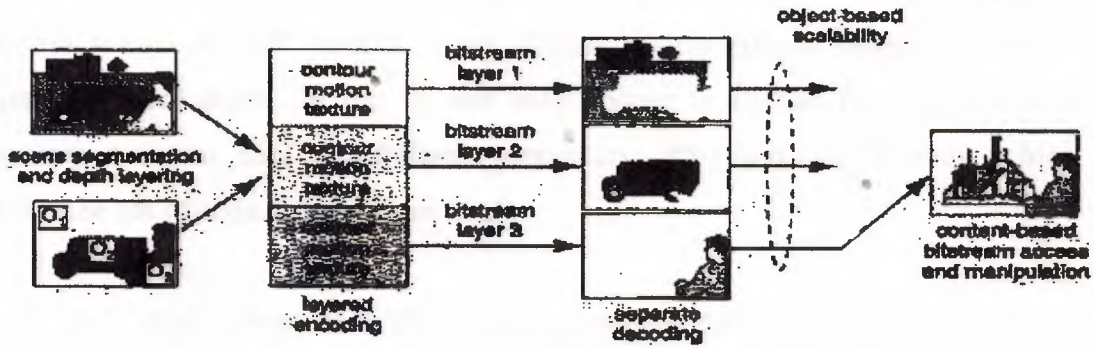


Figure 4.1 Basic concept of MPEG4 function

4.1.2 VOP Definition

The Video Object Planes (VOP) corresponds to entities in the bit-stream that the user can access and manipulate. The encoder sends together with the VOP, composition information to indicate where and when each VOP is to be displayed. At the decoder side the user may be allowed to change the composition of the scene displayed by interacting on the composition information

4.2 MPEG4 Encoder

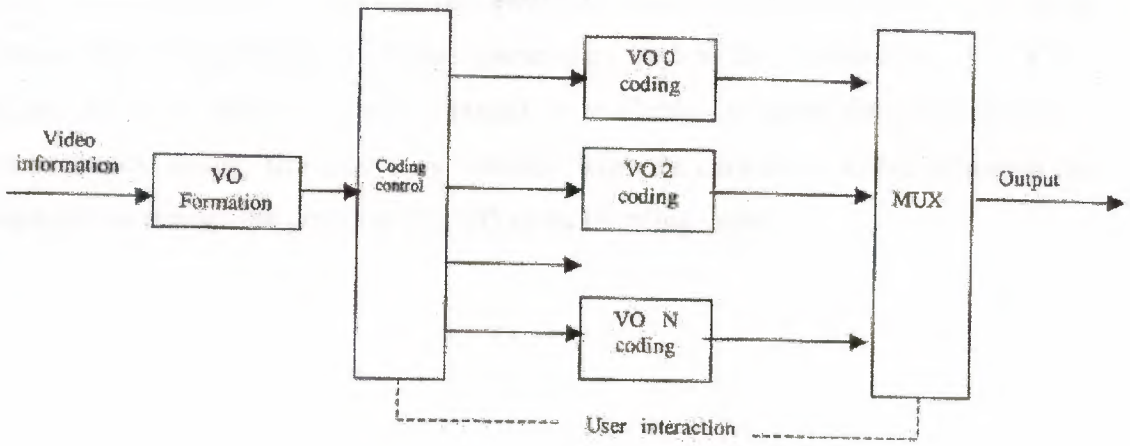


Figure 4.2 Encoder in the MPEG-4 video

Figure 4.2 shows the block diagram of the MPEG-4 video encoder. The most important feature of this encoder is the intrinsic representation based on VO when defining a visual scene. In fact, a user may choose to encode the different VOs composing a source data with different parameters, different coding methods, or May even choose not to code some of them at all.

In most applications, each VO represents a semantically meaningful object in the scene. In order to maintain a certain compatibility with available video materials, each uncompressed VO is represented as a set of Y, U, and V components plus information about its shape, stored frame after frame in predefined temporal intervals. It is important to note that although in MPEG-4 video test sequences the VOs were either known by

construction of the sequences or were defined by semi-automatic segmentation, this is done only due to practical considerations.

Another important feature of the Video VM is that no temporal frame rate is explicitly defined by this approach. This means that the encoder and decoder can therefore function in different frame rate, which do not even need to be constant throughout the video sequence.

Interactivity between the user and the encoder or the decoder can be conducted in different ways. The user may decide to interact at the encoding level, either in coding control to distribute, for instance, the available bit rate between different VOs, or to influence the multiplexing to change parameters such as the composition script at the encoder. In cases where no back channel is available, or when the compressed bit-stream already exists, the user may interact with the decoder by either acting on the compositor to change the position of a VO or its blending order.

4.3 MPEG Decoder

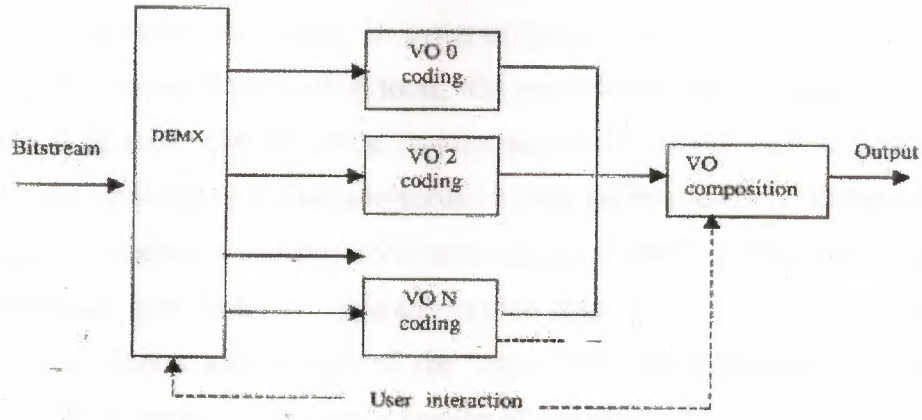


Figure 4.3 Decoder in MPEG- 4 video

Figure 4.3 shows the block diagram of the decoder in the Video VM. The structure of the decoder is basically similar to that of encoder in reverse, except for the composition block at the end. The exact method of composition (blending) of different VOs depends on the application and the Method of multiplexing used at the systems level.

4.4 Video Object Plane Based Encoder Structure in the MPEG-4 Video

The encoder is composed of two main parts, they are:

1) Shape coding
2) Motion & Texture coding

Shape coding and the traditional motion and texture coding are applied to the same VOP. Texture coding as well as motion estimation and compensation parts of the encoder are similar, in principle, to those used in other state-of-the-art standards, where care has been taken in order to extend their respective tools to objects of arbitrary shape.

Although the coding algorithm of the VM is designed to handle arbitrarily shaped object, all current tools are based on the concept of macroblock. This allows a certain compatibility with other standards and a more straightforward insertion of tools developed for such environments. In order to take advantage of both macroblock and other arbitrarily shaped object coding tools, -the multiplexing of the bitstream generated from the coding tools can be made in separate or combined motion-shape-texture modes. In the combined motion-shape-texture mode, the bitstreams of all these features are combined together on a macroblock basis and put in the final bitstream macroblock after macroblock, per VOP. In separate motion-shape-texture mode, the bitstreams generated for motion and texture of the entire VOP are multiplexed in the final bitstream, each occupying a contiguous portion of the latter.

Moreover, in applications where the shape information is not required (for example a classical rectangular frame based video coding), shape coding can be disabled, so in this project we may not have to use the shape coding.

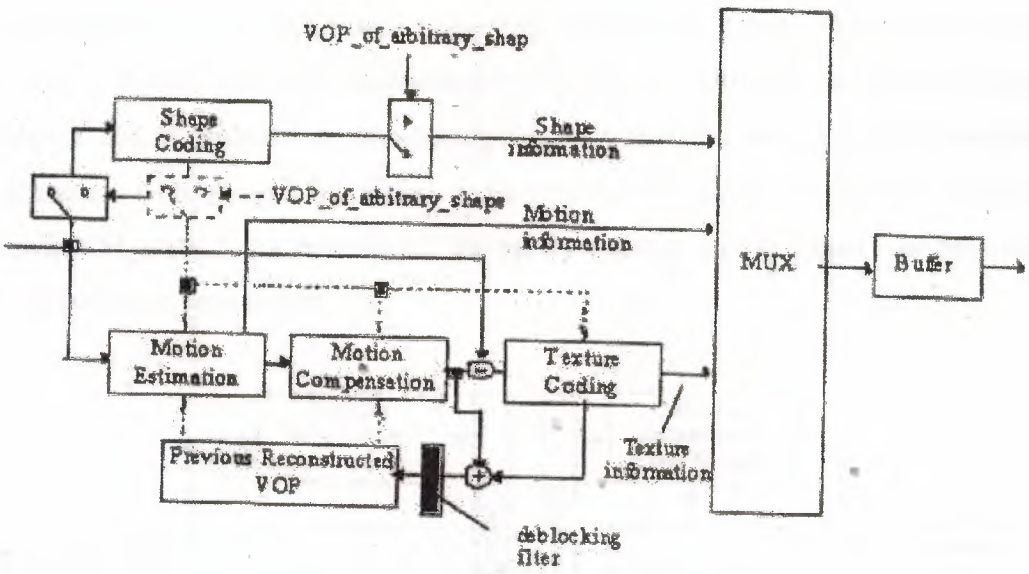


Figure 4.4 Video Object Plane based encoder structure in the MPEG-4 Video

4.5 Coding and Decoding Structure for a Two-Layer Scalability

VOPs are input to scalability pre-processor. If spatial scalability is to be performed with the base-layer at lower spatial resolution and the enhancement-layer at higher spatial resolution, the pre-processor performs spatial down-sampling of the input VOPs to generate a first base-layer which forms the input to the MPEG-4 VOP encoder in Figure 4.4. The reconstructed VOPs from the base-layer are then fed to the mid-processor which in this case performs a spatial up sampling. The other output of the pre processor corresponds to the higher spatial resolution VOPs and forms the input to the MPEG-4 enhancement layer encoder, which is similar in structure but different in strategy to that of the base-layer encoder. This project concentrates on base-layer encoder.

When the generalized codec is used to introduce temporal scalability, the scalability pre-processor performs temporal de-multiplexing of a Video object (VO) into two substreams of VOPS, one of which is input to the MPEG-4 base-layer encoder and the other to the MPEG-4 enhancement-layer encoder. In this case, mid-processor does not perform any spatial resolution conversion and simply allows the temporal prediction while encoding of enhancement-layer. The operations of the MPEG-4 system multiplexer and demultiplexer are exactly the same as in the case of spatial scalability. The decoding of base-layer bitstreams occurs in the corresponding base-layer decoders as shown in Figure 4.5. At the decoder, the post-processor simply outputs the base-layer VOPs without any conversion.

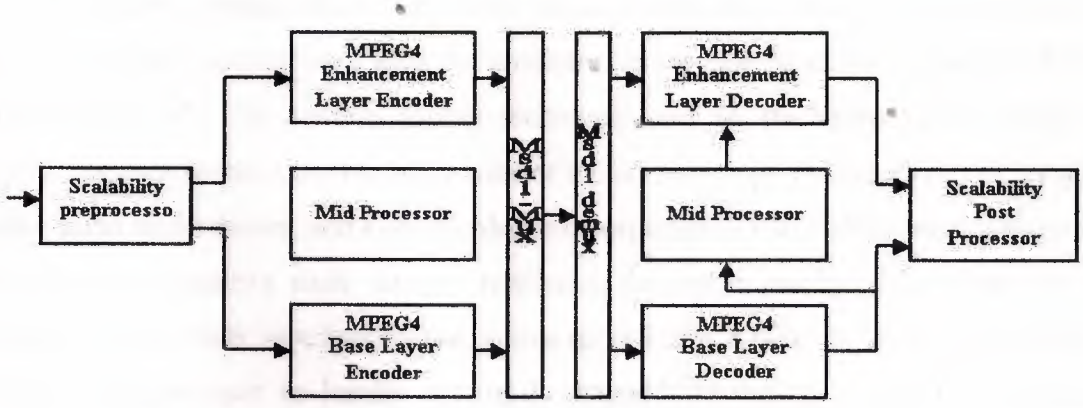


Figure 4.5 Decoding for a two-layer scalability

The VOP can be a semantic object in the scene; it is made of Y, U, V components plus shape information. In MPEG-4 video test sequences, the VOP were either known by construction of the sequences or were defined by semi-automatic segmentation. In the first case, the shape information is represented by an 8-bit component, used for composition.

4.6 Motion Estimation and Compensation

The MPEG-4 VM employs block-based motion estimation and compensation techniques to efficiently explore temporal redundancies of the video content in the separate VOP layers. In general, the motion estimation and compensation techniques used can be seen as an extension of the standard MPEG block matching techniques towards image sequences of arbitrary shape. To perform block based motion estimation and compensation between VOP's of varying location, size and shape, the shape-adaptive MacroBlock (MB) grid approach for each VOP image is employed. A block-matching procedure is used for standard Macroblocks. The prediction error is coded together with the Macroblock motion vectors used for prediction. An advanced motion compensation mode is defined which supports block-overlapping motion compensation. An image padding technique is used for the reference VOP frame N-i, which is available to both encoder and decoder, to perform motion estimation and compensation.

4.7 Texture Coding

Texture corresponds to the pixel values in the case of an intra-coded VOP (1-VOP) or to the residual error after the prediction in the case of an inter-coded VOP (P-VOP or B-VOP). The texture coding technique used in the Video VM is similar in several aspects to those used in other state of the art standards. Due to its simplicity and rather good performance, and also in order to incorporate in a straightforward manner in the VM functionality's such as error resilience, the texture coding of the Video VM is chosen to be a block based technique, where special care is taken in order to extend the block based approach to handle arbitrarily shaped VOP coding. Again, the bounding box of an intra-coded VOP or its corresponding motion compensated residual error is split into a number of macroblocks of size 16x16.

The intra VOP macroblocks and the residual macroblocks after motion compensation are coded one after the other using a Discrete Cosine Transform (DCT) scheme. DCT is performed separately on each of the luminance and chrominance planes in &given macroblock totalling 6 blocks of size 8x8.Similarly to motion estimation and compensation, thr& types of macroblocks may be encountered in a VOP bounding box: those that lie completely inside the VOP shape; those completely outside of the VOP but inside the bounding box; and those that partially cover the VOP The macroblocks that are completely outside of the VOP are not coded at all. Those macroblocks that lie completely inside the VOP are coded using a conventional DCT scheme. The 8x8 blocks of macroblocks lying partially on the VOP are first padded using repetitive padding as for the motion estimation, with the difference that for residual blocks, the region outside of the VOP within the blocks are padded with zero values. If all the pixels in an 8x8 block are transparent, their values are replaced by zero. These blocks are then coded in a manner identical to the blocks inside the VOP. Figure 4.6 illustrates typical cases of texture coding in an arbitrary shape context.

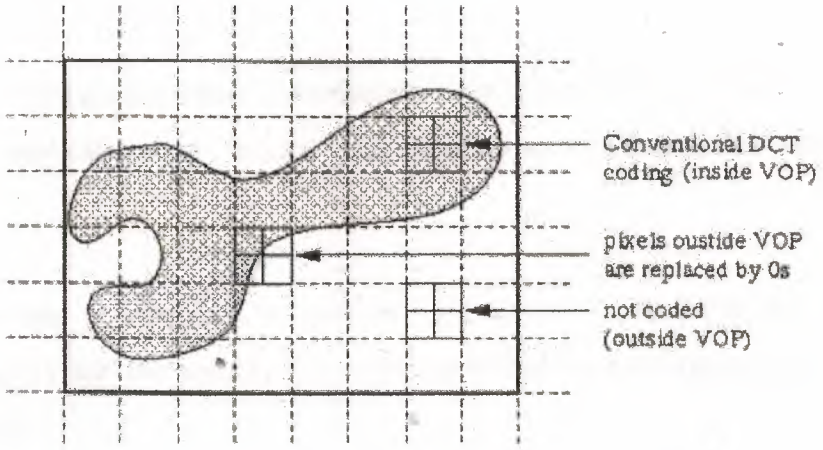


Figure 4.6 Texture coding

CHAPTER FIVE
PROTOCOL LITERATURE

5.1 Overview of the TCP/IP Protocol Suite

The transmission Control Protocol/Internet Protocol (TCP/IP) is "A suite of protocol for process-process Communication across an Internet". TCP/IP consists of a protocol stack and a number of networking-oriented applications, which provides transparent peer-to-peer process-to-process communication between host across multiple heterogeneous physical networks. TCP is a connection-oriented protocol offering a reliable, ordered, error free and timely delivery of data.

The operation of the TCP protocol is divided into three functions

1) Connection establishment.
2) Data transfer.
3) Connection termination.

Before any data can be sent, a connection must be established between the client and the server applications. A request is triggered by an active open from client Host application.

If connection is accepted, it leads to a three-way handshake to exchange the required parameters for the session. Each end sends and acknowledges the data in the TCP header fields.

Connection Parameters

Source port

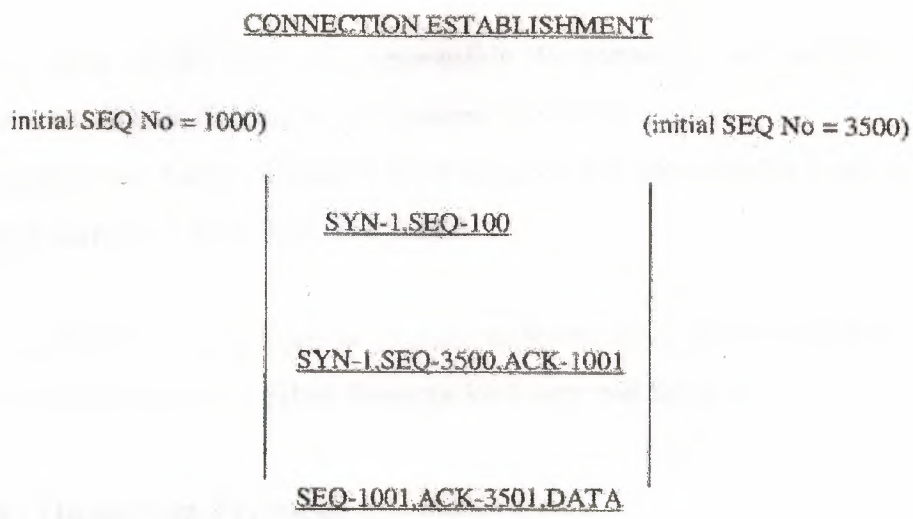
Destination port

Destination Address

SYN -bit in code field (=1)

SEQ# (chosen independently by each host)

ACK# (acknowledgement no is the sequence number of the next byte expected).



Data transfer proceeds using the sliding window protocol. In the last stage the client triggers the connection termination with the close primitive.

TCP/IP application protocol describes peer-to-peer communications, generally operating through client/server interactions:

- The client initiates a request for service.
- A server receives client request for a service at a well - known port and provides that service.
- Most TCP/IP platforms provide a Graphical User Interface (GUI) for the standard application protocol

5.2 TCP/IP Architecture

The TCP/IP architecture model consists of 4 layers.

- Network Interface - is the physical transport used to move data between hosts and servers on the same physical network. It makes use of the existing network protocols.
- Internet (IP) Layer - is responsible for routing 'datagrams' across the Internet.
- Transport (TCP) layer - is responsible for connection management. TCP ensures reliable data exchange between processes.
- Application Layer - Contains some support tools and contains a core suit of applications: Telnet, FTP, and SMTP.

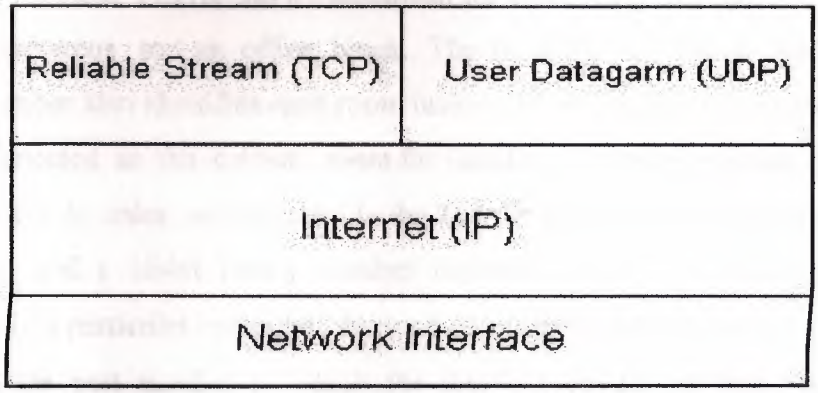
The TCP/IP architecture is slightly different from the 051-reference model, which has additional layers, called Presentation Layer and Session Layer.

5.3 User Datagram Protocol

The user datagram protocol (UDP) is an alternative transport, which provides a connectionless transport of datagrams. UDP is a lightweight protocol, adding very little to the functions provided by IP, which it uses. UDP is generally used by application services, which need fast data transmission and don't rely on reliability.

UDP does not use acknowledgements to make sure messages arrive, it does not order incoming messages, and does not provide feedback to control the rate at which information flows between the machines. Thus UDP messages can be lost, duplicated, or arrive out of order. Furthermore packets can arrive faster than the recipient can process them.

Conceptual Layering



TCP provides a reliable stream service, while UDP provides unreliable datagram service

5.4 Networking Concepts

The primary function of the TCP/IP is to provide a point-to-point Communication mechanism. One process on one machine communicates with another process on another machine or within the same machine. This communication appears as two streams of data; one stream carries data from one process to the other, while the other carries data in the other direction. Each process can read the data that have been written by the other, and in normal conditions, the data received are the same, and in the same order, as when they are sent.

In order to tell one machine from another machine and to make sure that you are connected with the machine you want there must be some way of uniquely identifying machines on a network. Early networks were satisfied to provide unique names for machines within the local network. However, Java works within the Internet, which requires a way to uniquely identify a machine from all the others in the world. This is accomplished with the IP (Internet Protocol) address, a 32-bit number.
IP Address can be given in two forms:

1. The DNS (Domain Name Service) form. Example: kenny Surrey.ac.uk.
2. Alternatively, we can use dotted quad form, which is four numbers separated by dots, such as 199.2.24.246

In addition to the machine addresses provided by the Internet Protocol part of the network system, TCP/IP has a mechanism for identifying individual processes on a machine, analogous to an office block. The building has phone number, but an extension number also identifies each room inside. When a call arrives at the building, it must be connected to the correct room for handling. Payment requests go to account payable, orders to sales, and so forth. In the TCP/IP system, the extension numbers are called ports., and a 16-bit binary number represents them. To communicate with the correct part of a particular computer, the sending machine must know both the machine address and the port number to which the message should be sent. Many common services have a dedicated port. Because some ports are reserved for common services, the programmer cannot use all the port. Ports numbered under 1024 are often referred to as reserved ports, many of which are reserved for a specific program. It is important that only ports over number 1024 are used.

Types Of Network Programming

Two general types are:

1. Connection-oriented programming (TCP).
2. Connectionless Programming (UDP).

5.5 Internet Protocol (IP)

The Internet Protocol is the glue that holds the Internet together. Communication in the Internet works as follows. The transport layer. Takes data streams and breaks them up into datagrams. In theory, datagrams can be up to 64k bytes each, but in practice they are usually around 1500 bytes. Each datagram is transmitted through the Internet, possibly being fragmented into smaller units as it goes. When all the pieces finally get to the destination, they are reassembled by the network layer into the original datagram. This datagram is then handed to the transport layer, which inserts it into the receiving process's input stream.

The Internet Protocol (IP) provides an unreliable connectionless packet delivery:

- ***Connectionless*** because all packets are transmitted independently of any other packets.
- ***Unreliable*** because packet delivery is not guaranteed. A packet could be discarded on its way to its destination if it is errored, exceeds its lifetime, encounters network congestion, or any number of other possibilities.

This unreliability can be overcome by TCP/IP applications keeping track of the status of delivery, expecting to receive replies from the destination node.

IP is implemented in all TCP/IP hosts and routers and provide a service to upper layer protocols by delivering IP datagrams from the source hosts to the destination via routers if required. In turn, IP uses the transport services of the underlying physical networks to carry the IP datagrams from host-to-host, host-to-router and router-to-host. Some of the reasons why IP has gained the status that it has today are given below.

- IP is an internetworking protocol, which is designed to operate over almost all underlying networks, and there is no requirement therefore for a homogeneous end to end infrastructure - end stations may use a variety of cheap interface cards (e.g. Ethernet) but the core network may be comprised of vastly different technologies
- IP is connectionless, which is well suited to datagram transfer, but can also support data stream transfer (via the TCP protocol).
- IP is a service independent, in that a wide range of application types are supported by a single, modular protocol suite and this means that new services are extremely easy to deploy, without requiring extensive changes to end station or network stacks.

For the above reasons, IP is becoming the preferred end to end network protocol not only for the public Internet and intranet, but also to an increasing degree, for applications such as real time applications such as video or audio data conferencing, application sharing, text-based chat, networked games.

5.6 Transport Layer

The transport layer protocol provides peer-to-peer end-to-end transport of information between the source host and the destination host.

- Interfaces with the application layer protocols (such as FTP, SMTP etc).
- Makes use of services provided by the IP layer to physically transport the data from the source IP address to the destination IP address.

  Two protocols are implemented in the transport layer of TCP/IP
- **Transmission Control Protocol (TCP) - a** reliable connection-oriented protocol
- **User Datagram Protocol (UDP)** - an unreliable connection-less datagram protocol

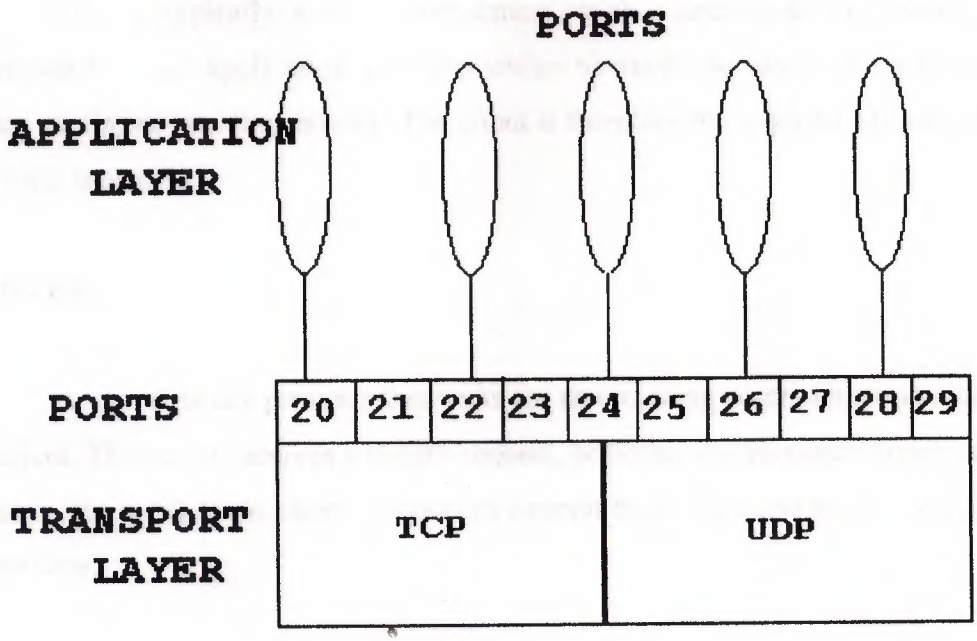


Figure 5.1 The Application layer Interface

5.7 Client/Server Paradigm

Modern computer systems make extensive use of the client / server paradigm, enabling applications to be divided into tasks and distributed across multiple platforms. This enables the designer to use the most suitable platform for each task, whilst providing implementation transparency to the user. Although the Internet provides peer-to-peer capabilities, most applications implement the client/server paradigm.

A client/server application uses two components.

CLIENT

This is typically a user workstation which is running the information delivery component of an application and who wishes to use the services, which are offered by other machines on the network. The client is therefore the originator of the connection with the server.

SERVER

A server is any program that waits for an incoming communication request from a client. The server receives a client's request, performs the necessary computation, and returns the result to the client. A host can support both client and server functions at the same time.

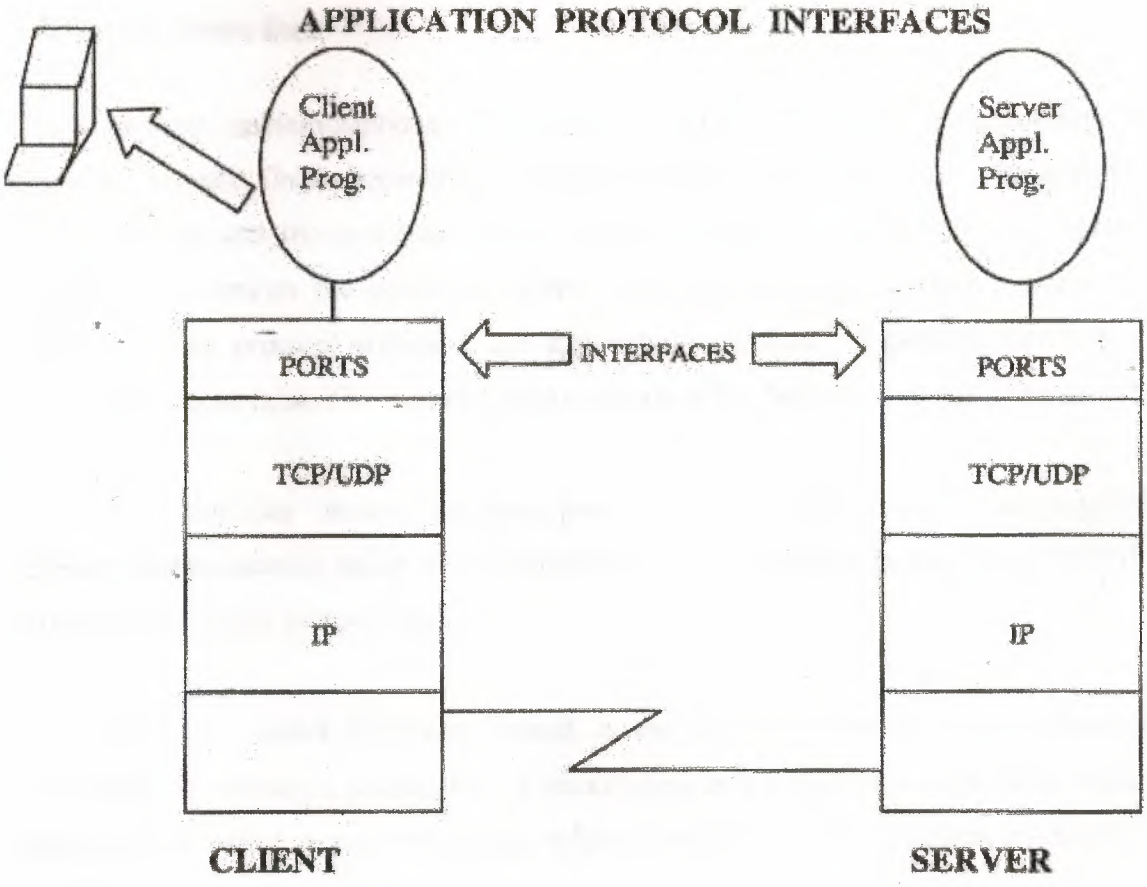


Figure 5.2 shows application protocol interface using Client-server model

5.8 Socket Interface

In most implementations, TCP/IP protocol software resides in the computer's operating system. Thus whenever an application program uses TCP/IP to communicate, it must interact with the operating system to request service. From a programmer's point of view, the routines the operating system supplies define the interface between the application and protocol software, the application interface. In general only a few TCP/IP interfaces exist. The Berkeley socket interface has been used in this program.

The Berkeley socket interface provides generalized functions that support network communication using many possible protocols. Socket calls refer to all TCP/IP protocols as a single protocol family.

Once a socket has been created, it can be used to wait for an incoming connection to initiate a connection. A socket used by a server to wait for an incoming connection is called a passive socket, while a socket used by a client to initiate a connection is called active socket.

The socket interface provides three TCP/IP services. It can be used for TCP stream communication, UDP datagram communication, and for raw datagram submission to the IP layer.

5.8.1 Socket Connections in Java

In Java, a socket is created to make the connection to the other machine. It is then possible to get an Input Stream and Output Stream from the socket in order to be able to treat the connection as an lOStream object. There are two stream based socket classes in the java.net package. They are **java.neLServerSocket** that a server uses to listen for incoming connections and a **java.net.Socket** that a **client** uses in order to initiate a connection. These classes are used in TCP connection oriented programs.

Once a client makes a Socket connection, the ServerSocket returns a corresponding server side socket through which direct communications will take place.

When a ServerSocket is created, it requires only a port number. There is no need to give it an IP address because it's already on the machine it represents. When a ClientSocket is created, however, it must be givens both the IP address and the port number to which connection needs to be established.

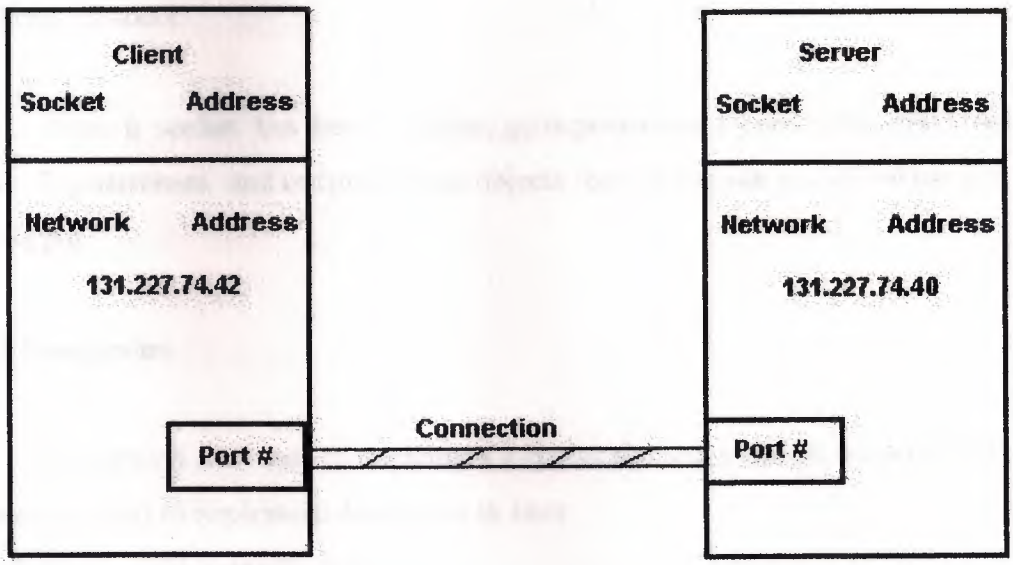


Figure 5.3 Socket Addresses

5.8.2 Server Socket

The ServerSocket represents a listening TCP connection. Once an incoming connection is requested, the ServerSocket object will. Return a Socket object representing the connection.

When a ServerSocket is created it, it listens for connections on the specified port for an optionally specified amount of time.

The most important method is accept (). It returns a Socket that is connected to the client. The close () method tells the operating system to stop listening for requests on the socket.

A method to retrieve the host name, the socket is listening on and the port number being listened to are also provided.

Java.net socket

This class implements a socket for inter-process communication over the network. The constructor methods create the socket and connect it to the specified host and port. We can also specify whether communication through the socket should be based on an underlying reliable connection-based stream protocol, or on an unreliable datagram protocol.

Once a socket has been created, **getInputstream ( )** and **getoutputStream ( )** return **Inputstream** and **outputstream** objects that we can use just as we use file input and output

5.8.3 Datagrams

Datagrams are used to implement a connectionless protocol, such as UDP. Two classes are used to implement datagrams in Java:

1. java.net.DatagramPacket
2. java.net.DatagramSocket

java. neLDatagramPacket

This class implements a "packet" of data that may be sent or received over the network through a Datagrainsocket. **DatagramPacket** is the actual packet of information, an array of bytes that is transmitted over the network.

Java. net.DatagramSocket

DatagramSocket is a socket that sends and receives DatagramPackets across the network. We can think of the DatagramPacket as a letter and a DatagramSocket as the mailbox that the mailcarrier uses to pick up and drop off our. Letters.

A datagram is a very low-level networking interface: it is simply an array of bytes sent over the network. A datagram does not implement any kind of stream-based communication protocol, and there is no semi-permanent "connection" established between the sender and the receiver. Datagram packets are called unreliable because the

protocol does not make any attempt to ensure that they have arrived or to re-send them if they did not.

5.9 Real Time Applications

It is useful to divide real time applications into conversational applications and streamed applications. Conversational applications are primarily interactive and typical applications include audio and video conferencing, application sharing etc. Streamed applications are essentially one-way flows of information. Typical examples would include information services such as stock prices and broadcast or on-demand video and audio services.

5.9.1 End-to-End Quality of Service

A particular issue facing real time applications is the difficulty in consistently achieving the required quality of service. Real time applications have quality of service (QoS) requirements that must be considered on an end-to-end basis. For real time applications designers are primarily concerned with temporal properties such as delay, jitters, bandwidth and synchronization and reliability properties such as error free delivery, ordered delivery.

Audio quality is highly sensitive to jitters, and the watchability of video is sensitive to available bandwidth.

The Internet caries all type of traffic, each type has different characteristics and requirements. But one of the major problems that real-time traffic suffers from congestion.

The solution for multimedia over IP is to classify all traffic, allocate priority for different applications and make reservations. The Integrated Services working group in the IETF (Internet Engineering Task Force) developed an enhanced Internet service model called Integrated Services that includes best effort service and real time service. The real-time service will enable IP to provide quality of service to multimedia

applications. RTP (Real Time Protocol) is one such protocol providing a working foundation for real time applications

5.10 RTP - Real time Transport Protocol

RTP is an IP- based protocol proving support for the transport of real-time data such as video and audio streams. The service provided by RTP includes time reconstruction, loss detection, and security and content identification. RTP is primarily designed for multicast of real time data, but it can also be used in unicast. It can also be used for one-way transport such as video-on-demand as well as interactive services such as Internet telephony.

RTP is a shared datagram network. Packets sent on the Internet have unpredictable delay and jitter. But multimedia applications require appropriate timing in data transmission and playing back. RTP provides time stamping, sequence numbering and other mechanisms to take care of the timing issues. Through these mechanisms, RTP provides end-to-end transport for real-time data over datagram network.

RTP is typically run on top of UDP to make use of its multiplexing and checksum functions. UDP was chosen as target transport protocol for RTP because for real-time data reliability is not as important as timely delivery. Even more, reliable transmission provided by retransmission as in TCP is not desirable. For example, in network congestion, some packets might get lost and the application would result in lower but acceptable quality

5.10.1 RTP Features

RTP provides end-to-end delivery services for data with real time characteristics, such as interactive video and audio. But RTP itself does not provide any mechanism to ensure timely delivery. RTP does not assume anything about underlying network, except that it provides framing.

RTP is typically run on top of UDP to make use of its multiplexing and checksum services, but efforts have been made to make RTP compatible with other transport protocols, such as ATM AAL5 and Ipv6.

5.10.2 RTP Implementation Resources

RTP is an open protocol that does not provide pre-implemented system calls. Implementation is tightly coupled to the application itself. Application developers have to add the complete functionality in the application layer by themselves. However, it is always more efficient to share and reuse code rather than starting from scratch. The RFC 1889 specification itself contains numerous code segments that can be borrowed directly to the application.

5.11 RTSP - Real Time Streaming Protocol

Instead of storing large multimedia files and playing back, multimedia data is usually sent across the network in streams. Streaming breaks data into packets with size suitable for transmission between the servers and clients. The real-time data flows through the transmission, decompressing and playing back pipeline just like a water stream. A client can play the first packet; decompress the second, while receiving the third. Thus the user can start enjoying multimedia without waiting to the end of transmission.

RTSP, the Real Time Streaming Protocol, is a client-server multimedia presentation protocol to enable controlled delivery of streamed multimedia data over IP network. It provides "VCR-style" remote control functionality for audio and video streams like pause, fast-forward, reverse, and absolute positioning.

RTSP is an application-level protocol designed to work with lower-level protocols like RTP to provide a complete streaming service over Internet.

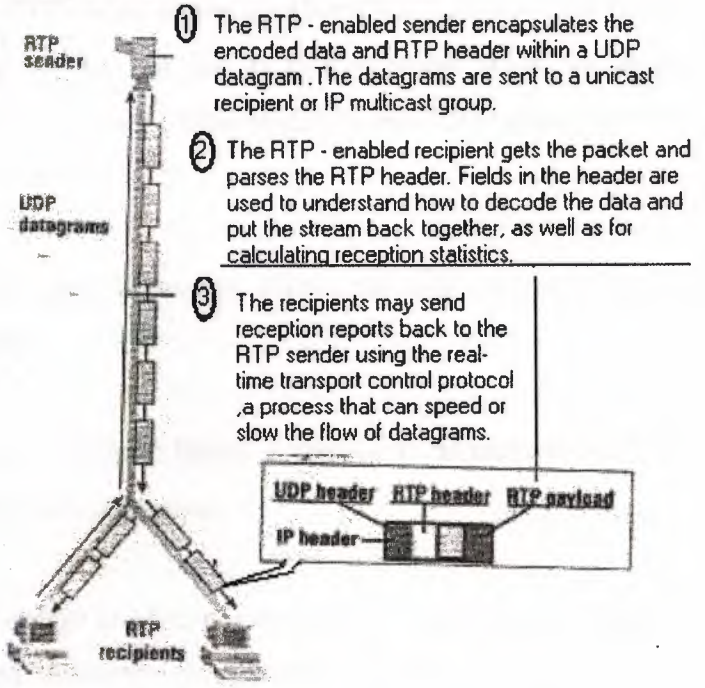
5.12 Sending Data with Real -time Transport Protocol

Real-time Transport Protocol (RTP) is used to send data in one direction with no acknowledgement. The header of each RTP datagram contains a time stamp so the recipient can reconstruct the timing of the original data, as well as a sequence number, which lets the recipient deal with missing, duplicate or out-of-order datagrams.

You'll find RTP ideal for sending streaming audio and video, whether to one (unicast) or to multiple recipients (multicast).

HOW IT WORKS

Sending data with realtime Transport Protocol
RTP is used for sending multi-media data ,such as streaming audio and video, to one recipient (unicast)or to multiple recipients (multi- cast) Date is usually sent one-way without acknowledgment.



RTP has a sister protocol Real-time Transport Control Protocol (RTCP), which lets a recipient give feedback to the RTP sender (and vice versa). For example, a receiving application might tell the sending application to slow down the video stream. At a slower rate, the video can still be shown, but it might appear jerky or in lower resolution. Guidelines in the RTCP specification help programmers avoid consuming too much network bandwidth with control flows.

RTP handles the real-time characteristics of multimedia applications well. Streaming applications differ from traditional data applications in the requirements they place on the sender, recipient and network. When streaming audio or video, it's OK to lose some data, but you don't want large gaps. In your payroll application, however, losing data is unacceptable.

The Internet Engineering Task Force (IETF) describes RTP in RFC 1889. RTP is the transport of choice for telephone calls, as well as streaming audio or video. The

International Telecommunication Union employs it in the multimedia communications standard H.323, and it's used by the real-time streaming protocol (RTSP).

RTP rides inside the User Datagram Protocol (UDP) and is thus connectionless. RTP is not part of the TCP/IP protocol stack, so applications are coded to add and recognize a new 12-byte header in each UDP datagram. The sender fills in each header, which contains:

Payload type: Describes the type of data - such as voice, audio or video - and how is it encoded.

Sequence number: Helps a recipient reassemble the data and detect lost, out-of-order and duplicate datagrams.

Time stamp: Used to reconstruct the timing of the original audio or video. Also, helps a recipient determine consistency or the variation of arrival times (sometimes known as jitter).

Source ID: Helps a recipient distinguish multiple, simultaneous streams, using a unique sender-generated value.

The RTP header can constitute a lot of overhead, depending on the size of the data payload. For example, a typical voice-over-IP data payload is 40 bytes. With RTP, the total header overhead consists of RTP (12 bytes) + UDP (8 bytes) + IP (20 bytes) =40

Bytes. Therefore, around *50%* of the datagram is the header. With video applications, the typical payload is usually larger, such as 1460 bytes, so the header overhead is a smaller percentage of the total size.

Some routers can reserve queues and priorities RTP traffic because they can recognize the standard headers. Some routers can also perform RTP header compression, often reducing the total header size from 40 bytes to 2 to 5 bytes.

Although compression reduces header overhead, it increases latency, so performing header compression on links faster than 500K bit/sec doesn't save much.

Most protocol analyzers can decode RTP headers and provide information about the data being sent. For instance, an analyzer can identify the payload type in the header, letting it show the actual payload data.

An advantage of RTP is its consistency among applications. Before RTP, application programmers using UDP would create their own datagram headers. When each application had a different header size and format, it was difficult for routers or other network devices to perform compression. In addition, sending and receiving applications had to be tightly coupled. With RTP, a receiving application from one vendor can receive RTP data from a sending application of another vendor.

New applications are bringing together rich mixes of voice, music, video and data on the same networks. RTP is positioned to be a core technology for transporting multimedia for some time to come.

CONCLUSIONS

The twentieth century has brought many technological revolutions to this world. Certain period during that century saw modern technologies that were useful to everyday life of mankind. The last decade is said to be the time of 'networks'. It opened up the way to revolutionize communication techniques for the future. To this date, the development on multimedia networks has deepened towards finding newer technological innovation for the future 'information age'. Nevertheless the demand for such a technology did not decline; especially there is a need for transmission protocol of live video over the Internet. Applications are such as video conferencing, live video chat, live video broadcast, etc. Its not only the video data need to be transmitted on real time, but the audio information also should be an integral part of the whole data.

The aim of this project is show how to transmit video image data (MPEG4) on real time over the Internet. Transport Control Protocol/internet Protocol (TCP/IP) was used to achieve the video data transfer.

An integrated system can be implemented using data transmission techniques and software.

REFERENCES

[1] Gilbert Held, Understanding data communication 5th, Sams, USA, 1996.

[2] Andrew S. Tanenbaum, Computer and Data Networks, Prentice Hall, New Jersey, 1998.

[3] MPEG4 - Home PAGE.
http://drogo.cselt.it/mpeg/

[4] IETF RFC 1889. "Real Time Protocol (RTP)".
http://www.cs.columbia.edu/~hgs/rtpl/, January 1996.

[5] IETF draft. "Real Time Streaming Protocol (RTSP)".
http://www.cs.columbia.edu/~hgs/rtsp/, March 27, 1997.

[6] Protocol Directory.
http://www.protocols.com/pbook/

[7] MPEG-2 Delivery system based on RTP AT&T Labs Research.
http://www.research.att.com/~mrc/pv99/CONTENTS/PAPERS/BASSO/pv99livenetlast.htm

[8] Java Tutorial.
http://java.sun.com/docslbooks/tutorial/