DEVELOPING A PACKET VOICE/DATA WORKSTATION:

AN APPROXIMATE ANALYSIS OF A SINGLE USER

INTEGRATED PACKET VOICE/DATA LINK

Victor S. Frost
Thomas J. Baumgartner

Telecommunications and Information Sciences Laboratory
University of Kansas
Lawrence, Kansas 66045

Technical Report TISL-6731-1
May 1985

Supported by:

National Science Foundation
Presidential Young Investigator Award
and
AT&T Information Systems

Principal Investigator: Victor S. Frost

ABSTRACT

Recently, the synergetic effort to integrate voice and data into a coherent framework has received considerable attention. Because voice and data exhibit bursty characteristics, packet voice and packet data have been considered as approaches to integration. Within the Private Branch Exchange (PBX) environment, advanced voice and data services are becoming more popular. A voice/data workstation applies these services within the PBX subscriber loop. The voice/data workstation we are considering packetizes both voice and data and statistically multiplexes the packets onto a high capacity transmission link. By using a non-preemptive protocol to multiplex the packets, each source gives up some capacity in the form of a delay when the non-preemptive situation occurs. Thus, this structure decreases the data packet delay at the expense of an acceptable amount of voice packet delay and voice packet loss.

A theoretical development is presented that gives approximate expressions for data packet delay, voice packet delay and the percentage of lost voice packets for the statistical multiplexer. This approximation degrades under high link utilization. To validate our theoretical predictions, a network simulation written in SLAM was developed. Extensive experiments were compiled that showed an agreement between simulated and predicted values. Consistent results also show that high data packet rates and small data packet sizes reduce data packet delay, voice packet delay and

voice packet loss. Voice packet delay is also decreased when using fixed length data packets.

Apart from the theoretical description, an initial approach to a hardware implementation for the workstation is given. This approach reduces most of the logical functions to a form that is conducive to a cost-effective implementation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

1.0 Introduction

The purpose of this report is to describe an initial approach in the design of an integrated voice/data workstation. A voice/data workstation is a communications device that brings voice and data services into an individual's office. These services may include access to separate networks within a business exchange or in the public network. The business exchange is defined as a Private Branch Exchange (PBX) and it controls all the communications switching functions for the business. A voice/data workstation interfaces to the PBX to give users easy access to voice/data information.

The present day PBX is designed strictly for switching voice traffic. The addition of data traffic through these PBXs severely decreases the capacity of the switch. Thus, in order to handle data services, Local Area Networks (LANs) have been added in parallel with a PBX. This may be technically feasible, but is surely not cost-effective. Today PBX manufacturers are recognizing the market that the LANs are servicing and redesigning a new generation PBX to satisfy voice and data needs. To accomplish this integration effectively, several different architectures must be investigated.

The conventional approach to voice/data integration has been to circuit switch voice traffic and packet switch data traffic. Because circuit switching for voice is a mature technology, it will be available for many years. Packet switching is not fully mature; it has, however, found a place in various networks carrying data traffic. This is because data traffic is "bursty" in nature and

packet switching is well suited for bursty traffic. Unfortunately, in the public network data traffic can only be circuit switched, because packet switched data is not an option. By using circuit switched data as a means of transport, the utilization of a transmission line is substantially reduced. This is because data traffic only uses the transmission line for short periods of time. Thus, circuit switched data represents an unfavorable approach to data communications in the public network. Because of this, packet switching of data in the public network is becoming a popular alternative.

Within a PBX environment, packet switching is receiving considerable attention. One reason for this attention is that high utilization is an important issue inside a PBX. The better utilized the PBX, the less equipment needed to build the switch. This is important within a competitive market.

Recently, packet switched voice has been considered to take advantage of the bursty characteristics of speech. One basic problem of packet voice is that voice packets cannot tolerate large delays when considering interactive communications. An advantage of packet voice is that considerable error can be tolerated in the packets up to an acceptable level. Conversely, data packets can tolerate large delays because interactive communication is not important. A disadvantage of switching data packets, however, is the need for error checking because errors cannot be tolerated. Thus, the issues related to packet voice and packet data are

considerably different. In Chapter 3.0 of this report, we will elaborate on these issues.

As previously mentioned, the purpose of this report is to describe a design of a voice/data workstation attached to the subscriber loop of a PBX. The approach taken in this design is to packetize both voice and data sources within the workstation. The packetized voice/data information is then statistically multiplexed to the PBX over a high capacity link. The statistical multiplexer follows a non-preemptive protocol that doesn't allow for packets to be preempted during transmission. Once a packet is on the link another packet cannot stop its transmission. The newly arrived packets are queued until the link is free. Also in this protocol, voice packets are given absolute priority. Voice packets waiting in the queue are immediately serviced once the link is free. When the packets arrive at the PBX they are separated with address and control information and routed to their appropriate destination. This model will be explained in great detail in Chapter 4.0.

The initial motivation for a packet voice/data approach is to attempt to reduce the hardware associated with the workstation. A comparison between a packetized voice/data design and circuit switched voice with packet switched data design will then be possible. A more concrete motivation to packet voice/data integration would be strictly economical. If we could handle all voice and data in a single network with a unified switching and transmission approach, the cost of such a network may be reduced.

It follows from this statement that a nonunified switching and transmission approach may require several different network architectures to accomplish the same functions. Thus, a packet voice/data integration approach should be investigated.

The thrust of this report is to define the necessary hardware functions to control the packets within the workstation. A more immediate problem, however, is to derive analytical expressions that predict the delays associated with the voice and data packets. The data packet delay will be used to predict the size of the data buffer (using Little's formula [11]). The voice packet delay will be important in determining the percentage of lost voice packets. This performance measure is defined when the voice packet delay exceeds one voice packet interarrival time. When this situation occurs, the voice packet waiting in the queue is discarded and considered lost. Thus, four important design parameters for the link controller are defined:

1. Data Packet Delay
2. Data Buffer Size
3. Voice Packet Delay
4. Percentage of Lost Voice Packets

In order to enhance the workstation design, we define these parameters using analytical methods. Using these analytical methods, we can make performance evaluations and comparisons with different architectures. These issues will be described in Chapter 4.0.

Chapter 2.0 of this report introduces the topic of packet communication. A historical perspective is given that outlines some early packet networks. Some issues are then given that pertain to packet switching and the different schemes used to deliver packets. Finally, a short introduction is presented that describes some popular networks that use packet switching techniques.

Chapter 3.0 introduces several ideas related to the motivation for voice/data integration. The Integrated Digital Services Network (ISDN) is defined as a network that links together a cluster of services into an integrated form. Some different approaches to the integration of voice and data that have been developed will then be outlined. Finally, a detailed description of the voice/data workstation is presented outlining the importance of the workstation in a PBX environment.

Chapter 4.0 comprises the bulk of this report. In this chapter a theoretical description will be given that predicts data and voice packet delays, plus percentage of lost voice packets as described earlier. This chapter begins by describing the function of a packet voice/data workstation. An explanation of the design goals and performance measures of the workstation model follows. A detailed description of the link controller (workstation) model is presented, including an example that explains the operation of the statistical multiplexer. Section 4.4 presents a theoretical analysis of the link controller considering variable length data packets. This section leads to the derivation of approximate expressions for the

design parameters described previously. Section 4.5 modifies this analysis by considering fixed length data packets. Again, this leads to approximate expressions for the design parameters listed earlier. In order to compare these results with a different architecture, Section 4.6 derives expressions for a Hybrid/TDM voice/data link. In Section 4.7 the expressions derived in Sections 4.4, 4.5 and 4.6 are put to the test by modeling the link controller with an event driven simulation. This validation leads to some interesting comparisons.

Chapter 5.0 introduces an initial solution in the design of the packet voice/data workstation. A flow diagram is given that describes the full duplex operation of the link controller. A functional diagram is then presented that defines the transmission functions of the workstation. Finally, the last section summarizes the hardware design and presents some important issues relating to the design. These may be helpful as the design evolves.

Chapter 6.0 presents an overview of the report and gives several recommendations for future work.

2.0 **Packet Communication**

2.1 Introduction

Packet Communication remains at the forefront of the communication industry today. Ever since its introduction in the early 1960s, [24] communication by the packet has been evolving towards a wide variety of applications. A strict definition of a "packet" would be a group of binary digits, including data and control signals, which are switched as a composite whole [33]. In a different sense, packets are small chunks of information that are exchanged independently through a communication media. Embedded within the packet is other information that handles the control functions of the packet such as routing and error control. Routing functions are concerned with delivering the packet to its appropriate destination. Error control is used to preserve the information in the packet because errors are inevitable as packets transverse through the network.

A descriptive analogy of this concept would be the mail system. A packet could be a letter that is being sent through this network. The address that is written on the envelope is used to route the letter to its destination. Similarly, a packet carries with it an address to route it to its destination. Smaller packets could be constructed by enclosing each individual word or character into an envelope and sending it through the network. Upon reception the reader would have to reconstruct the letter in an intelligible fashion. If some of the envelopes were lost in the network the

letter may be unacceptable. This is analagous to errors or lost packets in packet communication. Figure 2.1.1 shows how information flows in a packet voice network. This figure shows voice signals being transformed into a digital bit stream to be packetized and sent through the network. Since data is composed of binary digits, this figure could equally well represent the flow of packet data through a network. The path that the packets take in Figure 2.1.1 can be fixed or variable depending on the application. At the receiver the packets are reassembled for playback. Thus, a packet network is formed.

To better understand the advantages of packet switching some basic techniques used in switching are introduced. These are presented so that comparisons can be made. Communication by circuit switching is one of the earliest forms of switching. Circuit switching implies there is a dedicated communication path from the transmitter to the receiver for the entire length of the connection. Some of the advantages of this structure include a continuous transmission of information and the ability to sustain interactive communications. One disadvantage of circuit switching is the fixed use of the transmission bandwidth for the entire length of the call. The telephone network is an example of circuit switching.

Another approach to switching is to switch information in the form of messages that occupy the network only for the duration of the message. These messages are organized into logical units of data and contain address information that is used to route the

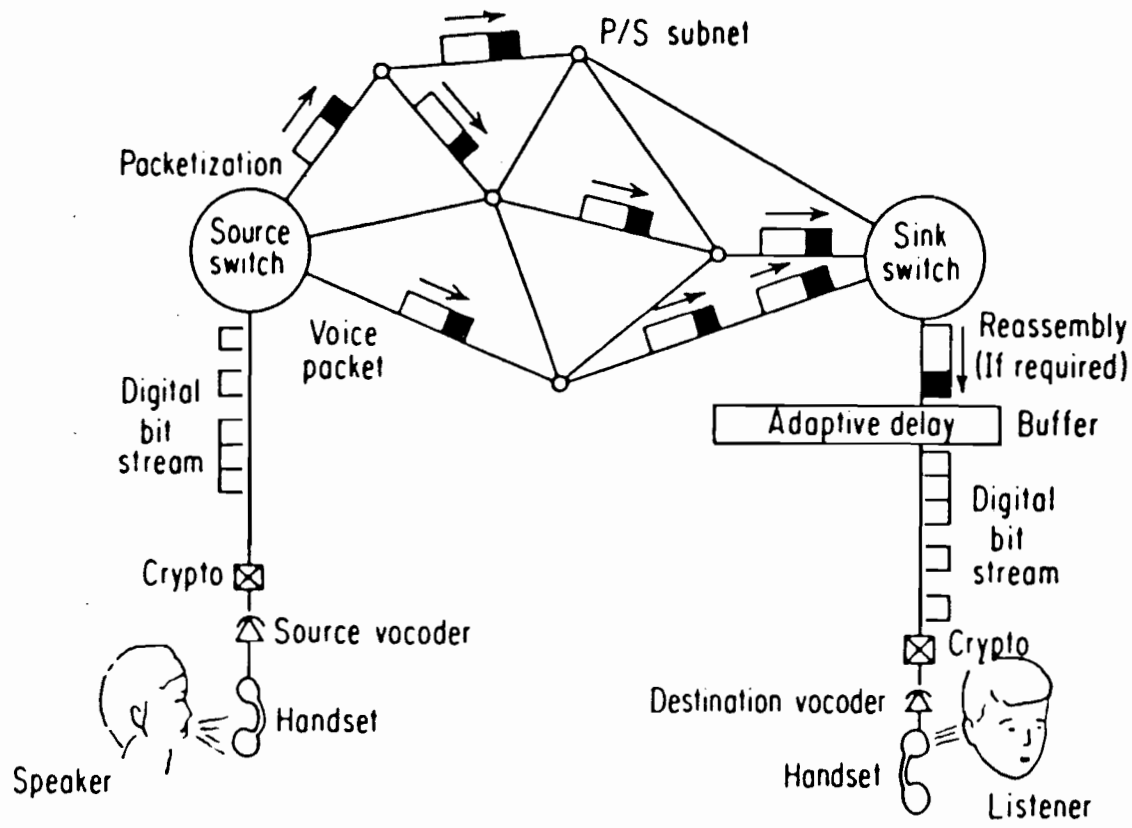


FIGURE 2.1.1 - Information Flow is a Packet Voice Network. Since data is also represented by a digital bit stream the network above is synonymous. Reprinted from D. Tugal/O. Tugal pp. 129 [33].

message through the network. The message switch at a network node must be capable of storing a message but need not wait for the whole message to be received before starting the onward transmission [26]. Examples of messages would be telegrams, electronic mail, computer files or other facsimile. This approach is similar to message switching because it uses the network for the duration of the call. Message switching, however, uses the network more efficiently. This approach is also known as store-and-forward switching since the messages may be buffered at nodes within the network. One advantage of message switching would be that there is not a dedicated path in the network, thus the transmission media is better utilized. A disadvantage of message switching would be that it is too slow for interactive communications. A more complete description of message switching is found in the reference material [11,23,26].

Packet switching attempts to combine the advantages of circuit switching and message switching while minimizing the disadvantages of both [23,34]. Packet switching is very much like message switching with the exception that packet switching restricts the number of bits that form a packet. A typical maximum length is 1000 to a few 1000 bits. Message switching uses a much greater number of bits. By restricting the size of the packets, interactive communication is possible which is similar to circuit switching. The advantage obtained from message switching is that there is no dedicated path to set up the circuit. The path is shared by other users, thus, it

is better utilized. A disadvantage of packet switching is that each packet may be queued up at nodes in the network causing packet delay. Thus, packet delay is an important design issue. An excellent comparison between circuit, message and packet switching is found in Stallings [23].

The comparison previously made shows packet switching as being advantageous from a technical viewpoint. Another question to ask is whether packet switching is feasible from an economic standpoint. The answer to this is yes, because computing costs today are low enough that packet switching becomes cost effective [28]. Furthermore, since the concept of packet switching is heavily dependent on computers, additional features can be added to the packet network to enhance its reliability. These features include checking for errors in packets, using computers to implement alternate routing schemes, acknowledgment schemes and others [11,33].

Throughout the rest of this chapter we will investigate some interesting topics that involve packet switching. First, however, a historical perspective is given that outlines the early beginnings of packet switching. Next, some important issues pertaining to packet communication will be discussed. Finally, a brief overview of some networks that use packet technology are presented.

2.2 Historical Perspective

In any engineering system it is known that the approach to the problem sometimes severely limits the system performance. The two basic approaches in designing a communications system are to either

preallocate or dynamically allocate transmission bandwidth [26]. Circuit switching, as defined in the previous section, is a pre-allocation technique that reserves bandwidth for the duration of the call. Packet switching, however, allocates bandwidth dynamically whenever it is needed. This approach remains more flexible in some aspects. Early engineers recognized the advantages of packet switching and put them to use.

Packet switching was not really an invention, but simply an application of dynamic allocation techniques that had been used by the mail and telegraph systems for many years [24]. The first published description of what we now call packet switching was an 11 volume analysis prepared by Paul Baran of the Rand Corporation in August 1964 [35]. This study was conducted for the Air Force as a proposal for a fully distributed packet switching network using voice and data. The goal in this system was to create a totally survivable communications network containing no critical components. Unfortunately, the Air Force never developed this project and this report sat idle for many years until other applications discovered it.

Also during the 1962-1964 time frame, the Advanced Research Projects Agency (ARPA) was working on the development of time-sharing computer systems under the direction of J.C.R. Licklider. One of Licklider's big interests was to link time-sharing systems together into large networks. The ideas Licklider initiated generated interest from others in the field at the time. Others

interested were Donald Davies and Lawrence G. Roberts authors of references [26] and [24,28], respectively. In 1965 Donald Davies of the National Physical Laboratory in the United Kingdom developed the details behind a store-and-forward packet switching system. In June 1966 the description of his proposal first coined the term 'packet' to describe the 128-byte blocks that were being transferred around inside a communications network [24]. It was only after Davies' report was complete that he discovered Paul Baran's work described earlier. Nonetheless, Davies was able to develop a local communications network using a single packet switch at the National Physical Laboratory. By 1973, this local network was operating effectively and providing a reliable distribution service within the laboratory.

In January 1967 Lawrence G. Roberts joined ARPA in a management position of computer research. At this time the environment within ARPA's research arena was ideal for the development of a packet network. Thus, during 1967 the ARPANET was conceived as a network to link together several major universities that were interested in networking. This network was used as a test bed for the verification of several different network issues. The ARPANET utilized minicomputers, in a fully distributed fashion, to handle switching procedures at each node. The links of this network were leased lines operating at a 50 Kbps rate. The ARPANET also contained a dynamic routing scheme that operated on the packets in order to minimize delay and enhance reliability. The success of the ARPANET

demonstrated the effectiveness of packet switching to the world by showing how well packet switching provided an efficient and highly reliable interactive data communications service. Since the ARPANET linked together several major universities in a common research arena, its performance was widely published [29,34]. The work of Kleinrock also became popular in defining network performance by using applications of queueing theory [27].

Other data networks began to emerge during this era (late 1960s to early 1970s). TYMNET was originally developed to provide a cost-effective connection to central time-sharing computers (it became operational in 1969) [23]. Systems Network Architecture (SNA) was introduced by International Business Machines (IBM) in 1974. SNA allowed users to construct their own private networks within a coherent framework [11]. Prior to SNA, IBM had several hundred communication products maintaining a variety of different protocols. SNA tied together a collection of these products to form a unique distributed communication network. In 1975, Digital Equipment Corporation (DEC) announced its Digital Network Architecture (DNA). Because DEC was a minicomputer manufacturer, DNA was mainly concerned with resource sharing in a distributed environment. Other network architectures are described in differing amounts of literature [11,23].

A final historical note to mention in this summary is the standardization that evolved around computer networks. With several public networks under development during 1974-1975, there was a

strong incentive to have all groups agree upon a standard user interface. Consequently, a committee was formed and a protocol was defined. The result of this effort was the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation X.25, adopted in March 1976. This recommendation was the first attempt by the developers of computer networks to adopt some form of standardization. A complete description of X.25 can be found in references [11,23].

2.3 Packet Switching Issues

In this section some of the basic principles behind packet communication will be addressed. Packet communication can be broken into two different forms depending on how the packets flow through the network. Two different approaches are defined as virtual circuits or datagrams. A virtual circuit is a packet switching service in which a path is established between two stations at the start of transmission. A virtual circuit needs call setup procedures and acknowledgments when it is active. All packets follow the same route and arrive in sequence. With a virtual circuit the user performs a call request to set up the circuit (or path) and uses sequence numbers to handle flow control and error control [11,23]. Another important feature of virtual circuits is that the destination address is only needed during call setup. This is important for protocol efficiency.

With datagrams, the network only agrees to handle the packets independently. Several differences are noted between virtual

circuits and datagrams. Datagram packet switching requires no call setup. The difference is that each node along the route may begin transmission of each packet as soon as the packet arrives. The datagram packet is almost always faster than message switching. Datagrams also need a destination address in each packet since datagram packets may follow different paths. One disadvantage of using datagrams is that more intelligence is needed at the node to handle packet processing. Virtual circuits are assured of a path through the network, thus less processing is needed. The trade-off is that datagrams are good for short messages and for flexibility while virtual circuits are best for long exchanges and to relieve nodes of packet processing [11,21].

Another issue that is important in packet switching networks is the function of routing packets from node to node. This section will attempt to summarize routing issues that are widely published [5,11,23,26]. In order to get a grasp on how routing affects the network some performance criteria are defined. To begin the discussion we must be given a topology or structure of the network that outlines the communication paths available. With a topology specified, we can then define performance measures such as the shortest route or minimum delay path that a packet has to transverse before its destination is reached. Other performance measures could be the maximum throughput or even the physical cost of each link. These issues all restrict the performance of the network from different points of view.

Other issues of concern with routing schemes are the routing strategy or the rules that the packets must follow as they progress through the network. Depending on the application, these rules can follow a fixed structure or an adaptive structure depending on the utilization. Still other issues are concerned with the form of the packets (virtual circuits or datagrams) and the intelligence within the nodes (distributed or centralized architecture). These issues are all addressed in references [5,11,23,26].

Aside from routing problems, traffic control and error control are also important topics in the design of a packet network. Traffic control is important because it regulates the number of packets entering and using the network. The goal of flow control is to prevent the network from developing a bottleneck. Another important issue is error control. Because errors occur in packets as they transverse through the system, steps must be taken to correct these occurrences. These topics and others are discussed in reference [29] and others mentioned above.

2.4 Networking with Packets

Ever since the introduction of the ARPANET, and countless other computer networks, the information age has taken off. Not only are Wide Area Networks (WANs) being developed, but Local Area Networks (LANs) are also receiving considerable attention. This section will briefly introduce some popular local area networks and other applications of packet communication.

Probably the earliest local area network to receive a lot of attention is Ethernet [11]. This network uses a contention based protocol so that users can broadcast packets to other users on a single shared cable. It was in 1976 that Xerox introduced this architecture [32]. Another popular architecture is the Token-Ring network. In this network a ring is formed where users access the ring at different points. A token is circulated around the ring and the station receiving the token may transmit data when it has the token. When the station is finished transmitting, the token is passed to the next station [30]. A host of different protocols are summarized in Tobagi's paper [31]. Packet communication is also being applied to radio and satellite networks [11]. Because of the characteristics of voice signals, packet voice is also receiving a large amount of attention [22]. Packet voice will be discussed in the remaining chapters. Other information on networks are found in several references [11,26,30,33,37].

3.0 **Motivation for Voice/Data Integration**

3.1 Introduction

In the previous chapter packet communications was introduced as a primary carrier for data information. It was briefly mentioned that packet voice may also be effective in a transmission media. In this chapter we will address issues concerning packet voice, packet data and the mixture of both.

It has been known for many years that data information has characteristics that are "bursty" in nature giving packet switching an advantage over any other means of transport [22,41]. It has not been until recent years, however, that packet voice has been considered [22,33]. As described in Chapter 2.0, the goal behind packet communication is to increase the utilization of transmission lines by removing the silent parts of information. An early technique employed by the Bell System for voice was Time Assignment Speech Interpolation (TASI). TASI improves the utilization by combining the talkspurts of several sources of speech on to a single transmission line. To do this, TASI removes the silent intervals of speech and transmits only the talkspurts. A "TASI advantage" is defined as the ratio of the number of callers supported to the number of circuits required to maintain a certain probability of blocking. Because this is a concentration technique, there will be occasions when some sources will be blocked from transmission. This may cause an undesirable "clipping" of an individual's talkspurt [39]. An interesting paper [51] describes a packetized speech

multiplexer and the way it can improve the TASI advantage at modest costs in packet delay. Other forms of speech interpolation using digital methods have also been investigated [40].

A more recent technique that combines voice and data into one integrated network while attempting to increase transmission utilization is the Slotted Envelope Network (SENET). The SENET concept divides the transmission line into fixed capacity frames that define the maximum capacity of the link. Within each frame there are slots that carry Time Division Multiplexed (TDM) circuit switched voice traffic and packet switched data traffic. A Class I region defines the circuit switched voice traffic and a Class II region defines the packet switched data traffic. The voice traffic maintains a fixed capacity on the link throughout the holding time of the call. As calls terminate their service on the link, the capacity of the data packets is increased to take advantage of the excess voice capacity. Thus, a movable boundary is defined that increases the data capacity as TDM voice calls leave the link. A complete description of this technique can be found in reference [43].

The efficient operation of TASI shows that the characteristics of voice have been exploited for many years. More recently, the development of SENET has shown the interest in voice/data integration. Since voice is also bursty in nature it seems feasible to develop packet voice networks. If packet voice is feasible, then it seems likely that packet voice and packet data should be unified

into a packet-switched network. Because of different characteristics, the integration of voice and data may not be as easy as it seems. For example, in a packet voice network, voice packets must be delivered under minimum delay constraints. Also, depending on the system parameters, voice packets can endure large error probabilities and even lost packets without severe loss of intelligibility [9]. This is because the human ear acts as an integrator and smooths out the error. The opposite characteristics are found in data packets. Data packets can tolerate substantial delay as long as the user can wait. High error probabilities for data packets cannot be tolerated because machines do not have the ability to smooth information as humans do unless complexity is added. In summary, data packet networks need added complexity to control errors while voice packet networks need complexity to deliver packets in a minimum amount of time. With these concepts in mind, the marriage of voice and data into a unified packet network can be described.

Although the technical feasibility of an integrated packet-switched voice/data network can be demonstrated, there may exist economic reasons for not developing such a network. One reason would be because of the emergence of Fiber-Optic transmission systems with their enormous potential for capacity. If bandwidth is unlimited, then why should engineers worry about sharing transmission lines. This issue is receiving considerable attention

throughout the telecommunications industry. Nevertheless, the work in packet voice/data integration goes on.

The next section of this chapter will introduce the ideology behind the Integrated Services Digital Network (ISDN) and how it affects packet voice/data networks. After that a description of some different approaches to voice/data integration will be given. Finally, the last section will introduce the Private Branch Exchange (PBX) and an approach to an integrated voice/data workstation.

3.2 Integrated Services Digital Network

The Integrated Services Digital Network (ISDN) is a planned worldwide telecommunication service that will use digital transmission and switching technology to support voice and digital data communication [23]. One goal of this service is to satisfy the user requirements with a minimal amount of investment. The adoption of digital techniques in a public network make it possible and cost effective to integrate voice, data and even video into an ISDN. The advantage of the ISDN is that the same transmission and switching facilities can be shared by services that presently exist in separate networks [42]. This section of the chapter will attempt to outline the issues of concern in an ISDN.

Because the ISDN concept is not completely defined to date, a detailed description of this network cannot be given. However, an introduction into some important design issues can be presented. Figure 3.2.1 shows a proposed concept for the ISDN connection (reprinted from Stallings [23]). The user in this system has access

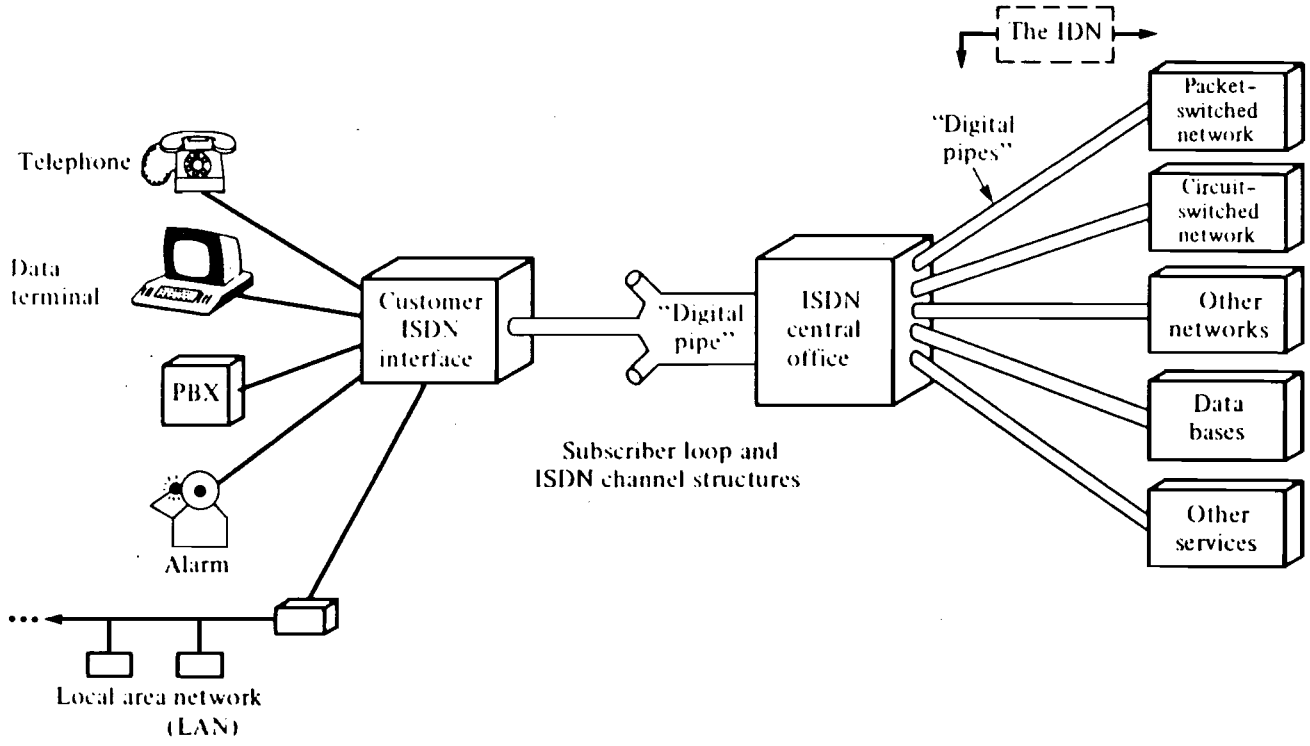


FIGURE 3.2.1 - Conceptual View of ISDN Connection Features.
Reprinted from Stallings pp. 540 [23].

to the ISDN at the customer ISDN interface. The services offered to the user are numerous. These include voice, data, alarms, video and even access to a Private Branch Exchange (PBX) or a Local Area Network (LAN). The customer interface is connected to the central office by means of a digital pipe branching to separate users. The capacity of the pipe given to each user is fixed, but determined by the customer's needs. At any given point in time, the pipe to the user has a fixed capacity but the traffic may be variable depending on its utilization. Charges for the use of the digital pipe can be a function of its utilization.

The function of the ISDN central office in Figure 3.2.1 is to allow the users access to several different networks. This side of the central office is defined as the Integrated Digital Network (IDN). These networks include packet switched, circuit switched and other unique forms. Access is also provided to different data bases. Other services are also defined for the future expansion of the system.

Currently, the development of an ISDN is being described by several different groups in the communications field. The International Telegraph and Telephone Consultative Committee (CCITT) is working extensively to define an ISDN. The most essential objective in any network design is to define standards that permit universal access, as well as cost-effective products that interface with the ISDN. Another objective is that the network should be transparent from the user's viewpoint. This should allow users the

ability to develop their own protocols that interface to the ISDN. Another important objective is that the conversion to the ISDN should evolve gradually and must have the ability to coexist with existing services. These objectives and others are described in references [1,2,23,42].

A final note to mention about the ISDN is the transmission structure being considered. The digital pipe from the central office and the ISDN interface in Figure 3.2.1 carries several channels. The transmission structure that is defined by the CCITT consists of the following channels:

- B channel 64 Kbps
- D channel 16 Kbps

There are other channels defined, however, the "B" and "D" channels are the most popular. The "B" channel is the basic user channel and it can carry different forms of traffic. This traffic can be Pulse Code Modulation (PCM) digital voice [36] for voice service or digital data using circuit switched or packet switched applications. Lower bit rate voice traffic can also be used on the "B" channel.

The "D" channel in this structure will be used to support signaling and control functions within the network. Thus, the signaling will be moved out-of-band as opposed to the in-band signaling of the Bell System's T1 carrier [43]. Another feature of the "D" channel is that it will also be able to support low-speed digital data when the channel is not being occupied with control

signals. This adds valuable flexibility that also improves the "D" channel utilization.

The basic service for the user is defined as two "B" channels and one "D" channel. This gives a 144 Kbps service to the user. With framing and synchronization the total service requires 192 Kbps of capacity. A complete description of these structures are found in Stallings [23].

3.3 Different Approaches to Integration

Aside from the ISDN approach just described, there have been several different suggestions for defining an integrated network. For a basic service the ISDN strictly defines 64 Kbps channels for voice and data. This approach fixes the capacity for both voice and data while giving continuous service to both sources whether it is needed or not.

A second approach presented in the previous section uses the SENET. This network multiplexes several sources of voice and data and dynamically allocates bandwidth depending on the load of the network. As the voice source load decreases (fewer voice users), a movable boundary is defined that increases the capacity allowed for data. This approach is similar to the ISDN approach in that the total capacity given to all sources is fixed, however, it is dissimilar because SENET allows a variable capacity depending on the load [43].

A third approach to this integration would be to dynamically allocate bandwidth to the data source during the voice silent

intervals. In a different sense this can be described as injecting data between voice pauses. This approach has received considerable attention in research [46,47,48,49,50]. All of these approaches take advantage of voice silent intervals by transmitting data at this time. Some of these approaches give voice absolute priority by preempting data packets when voice is ready to be serviced [46,47] while others do not [48]. Reference [49] increases its capacity by using silence detection [20] during heavy loads. Still other approaches combine the movable boundary (SENET) structure while injecting data packets in voice silent intervals [50].

The approaches presented in this section are all similar in the sense that the voice information is being circuit switched and the data information is being packet switched. One reason for maintaining a circuit switched channel for voice is because packet voice has not been widely accepted [22]. Another reason is that circuit switching technology is a mature technology that is well defined. Considering packet voice as a replacement would take many years of effort. Thus, before this effort is initiated, considerable research must be completed.

The next section of this chapter will introduce a unique approach to integration by considering packet voice as a viable alternative. This unique approach will be used in a description of a voice/data workstation that interfaces to a Private Branch Exchange (PBX). The remainder of this report will be dedicated to the description of a voice/data workstation architecture.

3.4 A Voice/Data Workstation

At the present time (1985), a large interest in voice/data integration is occurring in the Private Branch Exchange (PBX) market [45]. Because of the increased demand for communications facilities, business today is faced with two alternatives. The first alternative is to keep an existing PBX for voice communication and add a second parallel network to handle data communication. The data network would take the form of a Local Area Network (LAN) and may prove to be costly and inefficient.

The second alternative involves the latest breed of voice/data communication systems. These are described in reference [45] as the fourth generation of PBX systems. These new systems combine advanced voice features, high speed data transmission and integrated LAN services. Such advanced systems are giving LANs an enormous amount of competition since they are meeting all office communication needs within one integrated network. Thus, the second alternative seems more attractive than the addition of a new data network to the office environment.

An important peripheral device that defines the user interface to this new generation of PBXs is the voice/data (V/D) workstation. This piece of equipment defines voice services and data services for an integrated network. Dudley presents an excellent description of a workstation's features [44].

The approach taken in defining a V/D workstation is quite different than the schemes presented in section 3.3. The approach

we are taking is to packetize both voice and data and statistically multiplex these packets at a high rate to a PBX. The advantage to this approach may be in the reduction of some hardware within the workstation or PBX resulting in a cost-effective solution. This observation, unfortunately, requires research to verify. Thus, the thrust of this report is to investigate the implications of a packet voice, packet data workstation. A complete description of this problem will be presented in the following chapters.

4.0 **Developing an Integrated Voice/Data Multiplexer**

4.1 Introduction

As described in the previous chapter, voice/data integration is an important issue in the telecommunications industry today. Customer service requirements predict the need for circuit switched and packet switched facilities in an integrated network [1]. The ISDN defines a circuit switched channel for voice and a packet switched channel for data. In this chapter a packet voice and packet data statistical multiplexer is investigated. This multiplexer combines voice and data by injecting data packets on to a link when a voice packet is not being serviced. The end result of this investigation is to define a functional description (Chapter 5) of this piece of equipment. An intermediate result, which is described in this chapter, introduces a theoretical model of the statistical multiplexer. This model is used to predict voice and data packet delays, the size of the data buffer and the percentage of lost voice packets using this scheme. This chapter will first introduce the design problem described earlier, then discuss some important issues in this design and finally present a theoretical model used to predict the design parameters. The last section of this chapter will validate the theory by comparing it with an event driven simulation [4].

A physical description of a piece of equipment that would use this multiplexing scheme would be a voice/data workstation presented in Figure 4.1.1. The link access controller in this drawing has the

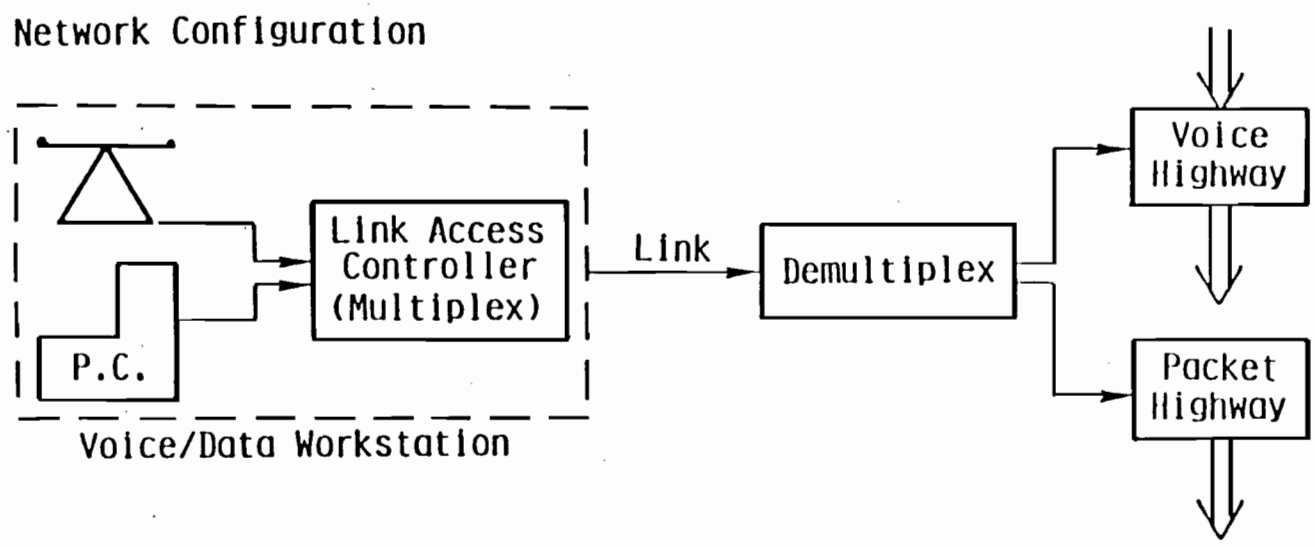


FIGURE 4.1.1 - Network Configuration for Voice/Data Workstation. Link Controller is described as a statistical multiplexer.

job of multiplexing voice and data to a single link. The demultiplexer at the end of the link will then separate voice and data to their appropriate destinations. The underlying problem to be resolved is to develop an efficient link between a voice/data workstation and some local voice/data highways. These highways could be present inside a private branch exchange (PBX) or any form of local area network (LAN).

4.2 Important Issues to Address

In order to define a functional description of the link access controller described in Figure 4.1.1, we first address issues that are important to our design. The first set of issues are described as goals in design and are presented below.

- Design Goals
    1. Simplicity in hardware construction
    2. Cost effective in implementation
    3. Have the ability to access the voice highway with minimum (or acceptable) delay and lost packets
    4. Have the ability to access the data highway with minimum (or acceptable) delay

Apart from the design goals is the method the link access controller should multiplex the packets. One alternative would be to reserve time slots for voice and use the rest of the capacity for data. For our purposes this is labeled as a Hybrid/TDM form of multiplexing [6]. This form is similar to the ISDN proposal which defines one "B" channel for voice (64 Kbps) and a "B+D" channel for data plus signaling (80 Kpbs) [2].

A second alternative, used in this development, is to packetize both voice and data signals and use a statistical multiplexer to deliver the packets. Hopefully, by using a statistical multiplexer, the performance of the voice/data link controller will be improved. If we can't do as well as or better than the Hybrid/TDM form in matching our design goals, then this alternative will be less desirable. A further comparison between these two multiplexing schemes will be presented periodically in Chapters 5 and 6.

As described above, a packet voice/data statistical multiplexer is the model used to describe the link controller in Figure 3.1.1. In order to evaluate the performance of this model, some design variables need to be defined. These variables are listed in Table 4.0 below.

Table 4.0 Design Issues

1. Data Packet Delay
2. Data Queue (Buffer) Size
3. Voice Packet Delay
4. Percentage of Lost Voice Packets

Using analytical methods, these variables will be approximated as a function of the design parameters (e.g. voice/data packet sizes, voice/data rates, etc.) and will be used for comparison purposes in this chapter and Chapters 5 and 6. A complete definition of their meaning will be given shortly.

4.3 Description of the Model

The next important topic that must be introduced is a detailed description of the link controller or statistical multiplexer given in Figure 4.1.1. This is best explained by introducing Figure 4.3.1 which depicts a queuing model of the statistical multiplexer. This model shows a single voice source and a single data source entering separate queues. The voice packets used in this model are assumed to arrive periodically to the voice queue. This is obviously not true, as previous work in speech patterns has shown [3]. The reason we chose periodic voice packets was to analyze the worst case and see how well our model would hold up under these conditions. Revising our model to handle talk spurts and pauses leading to the implementation of silence detection is something that will be addressed later. It should also be noted that the voice source queue only allows for the buffering of one voice packet. The rationale is that a voice packet is considered lost if it is not delivered before the next voice packet arrival. This is an important issue and is directly related to the percent probability of lost voice packets shown in Table 4.0. The data packets in this model follow a Poisson arrival process exhibiting exponential interarrivals. The data queue in this model is considered finite and is also an important design variable.

At this point a description of the switch operation in Figure 4.3.1 is needed. Since voice packets must be delivered as soon as possible under minimum delay constraints, voice packets are given

QUEUEING MODEL

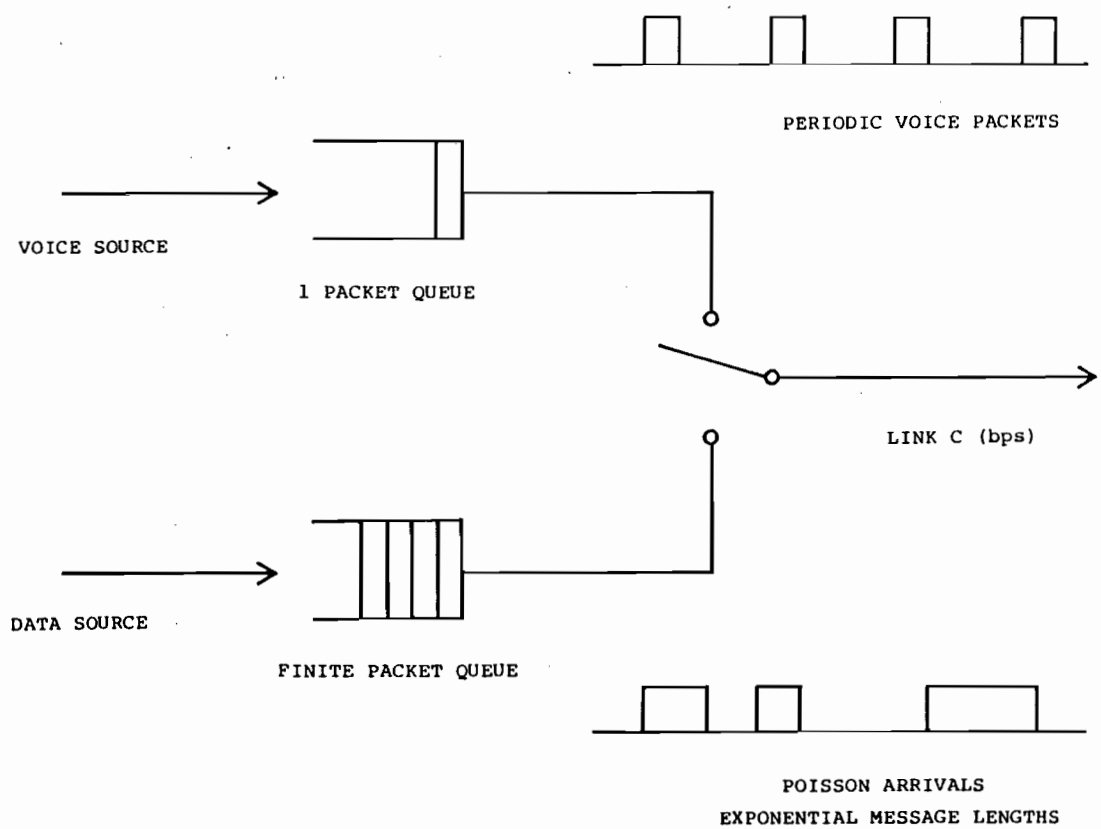


FIGURE 4.3.1 - Queueing Model for Voice/Data Workstation. Model assumes periodic voice packet arrivals and poisson data packet arrivals with exponential messages lengths. Switch is controlled in a non-preemptive fashion.

the highest priority. Data packets, however, can suffer considerable delay as long as the data buffer has a large enough capacity. With these observations a protocol giving non-preemptive priority for voice packets was devised. This scheme gives voice packets absolute priority over data packets but does not allow the link to be preempted by either source. This protocol is best explained by looking at Figure 4.3.2. In Figure 4.3.2a a voice packet is on the link (being served) and a data packet arrives in the middle of transmission. The protocol described above would not preempt the voice packet on the link. The voice packet would finish transmitting and a data packet in the queue (not necessarily the one that arrived during transmission) would follow. The opposite case is also true shown in Figure 4.3.2b. In this case a data packet is on the link and a voice packet arrives. As before, this results in the data packet finishing service and the voice packet following. With these examples, the delays that occur from this overlap can be accounted for in the next section of this chapter. It should also be noted at this point that the description above does not define the case where one data packet interacts with two or more voice packets. This case only becomes important for long data packets or high link utilization. The opposite case is not true since more than one voice packet arriving when servicing a data packet would constitute a voice balk (a voice packet that arrives when the previous voice packet is waiting for service passes over or balks

LINK SERVICE CONFIGURATIONS

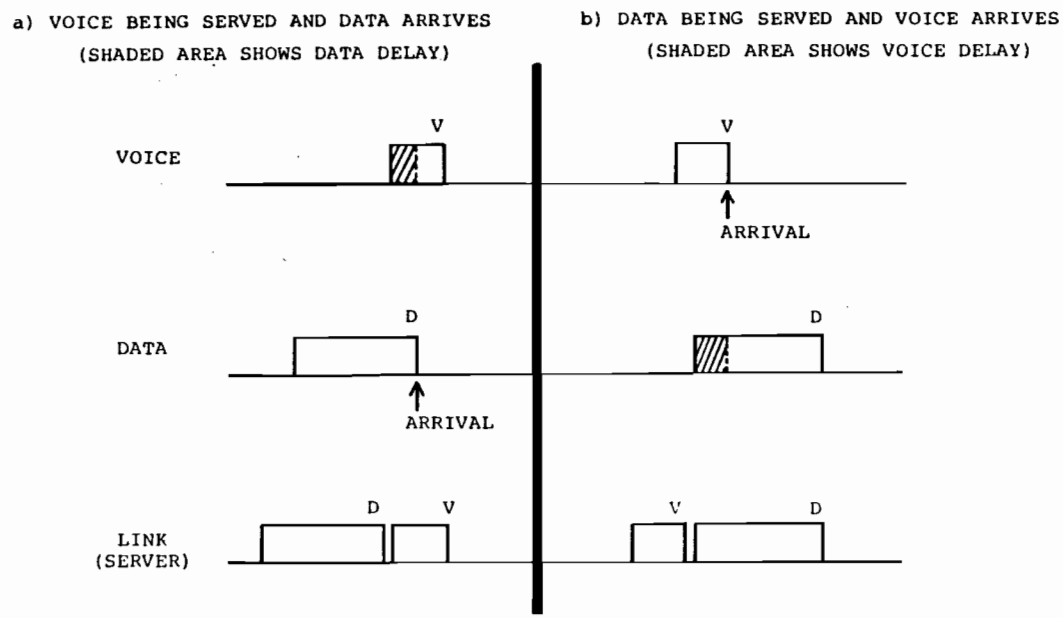


FIGURE 4.3.2 - Link Service Configurations of Queueing Model. a) Voice packet is being served and data packet arrives. b) Data packet is being served and voice packet arrives.

the waiting queue and is lost; this adds to lost voice packets). A further explanation of these phenomena will be given in the theoretical development.

In the next two sections of this chapter, a theoretical analysis of the model presented earlier will be developed. The first of these sections will develop the analysis using variable length data packets (exponential message lengths) and the second section will use fixed length data packets. The outcome of this analysis will allow us to predict the design parameters of Table 4.0. These predictions will then allow us to refine a functional description of the workstation by giving us an insight into the way these parameters interact. Verification of these predictions will then follow by simulating the queueing model of Figure 4.3.1 using SLAM* [4].

4.4 Theoretical Analysis using Variable Data Packets

In this section of the chapter, a theoretical analysis of the queueing model described earlier (Figure 4.3.1) will be presented using exponential length data packets. This analysis will lead to the prediction of the design parameters listed in Table 4.0. Poisson arrivals for the data packets and periodic arrivals for the voice packets are assumed as described in the previous section. The queueing model of Figure 4.3.1 is helpful in understanding this development.

---

*SLAM - Simulation Language for Alternative Modeling, Pritsker 1981.

The first parameter that we will derive from Table 4.0 is the delay of the data packets. In developing this parameter we must first find a model that completely defines the variables interacting in the data queue. The first observation that is made by looking at Figure 4.3.1, is that the Poisson arrivals into the data queue are not affected in any manner by the operation of the switch. The service time of the data queue, however, is affected and must be investigated further for a complete definition. Looking back to Figure 4.3.2 we can see how the non-preemptive priority of the voice packet added to the delay of a data packet. This is exactly the motivation that led us to Figure 4.4.1 which shows the data queue by itself with the service time broken up into two different variables. These variables, which are assumed statistically independent, add up to form a service time model yielding an M/G/1 description of the data queue. The variable "τ_D" in this figure is simply the service time of a queue given no interaction of voice packets or simply the service time for an M/M/1 queue. The second term "τ_V", however, is the term that originates from the added delay of the non-preemptive protocol. This term (τ_V) is a pure delay term that we are appending to the service time to form the general distribution of the M/G/1 model. To avoid confusion this term will be labeled as a service time along with its distribution. But it must be kept in mind that this is done for modeling purposes only.

If superposition applies, assuming statistical independence, the total service time "τ" will be used to describe the probability

M/G/1 QUEUE MODEL

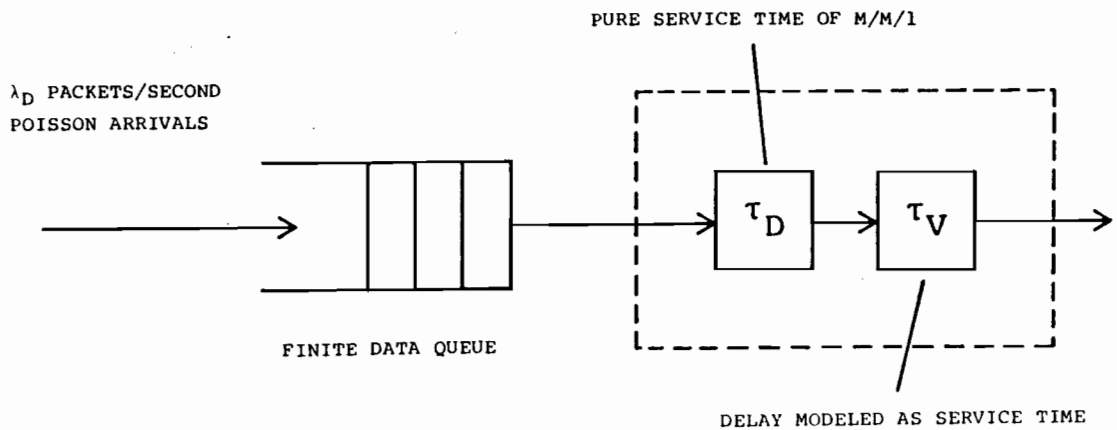


FIGURE 4.4.1 - Service Time Model for Data Queue (M/G/1 Queue).

$\tau_V \triangleq$ Service time due to non-preemptive priority of voice.

$\tau_D \triangleq$ Service time due to M/M/1 model.

$\tau = \tau_V + \tau_D \triangleq$ Total service time for M/G/1 Queue.

density function (p.d.f.) of the server so that the data delay can be found (using a canned formula from Schwartz [5] for M/G/1 queues). Before we start with the analysis, a description of the variables used in this development is given. Table 4.1 below defines the variables associated with the data queue.

Table 4.1 Data Source Parameters

$L_D \triangleq$ Average message length (bits/packet)

$\lambda_D \triangleq$ Average data packet rate (packets/second)

$\rho_D \triangleq \frac{\lambda_D L_D}{C} \triangleq$ Link utilization due to data packets

$T_D = \frac{L_D}{C} \triangleq$ Average service time of data packets (no voice)

$R_D = \lambda_D L_D \triangleq$ Data rate (bps)

$C \triangleq$ Link capacity (bps)

The first step in this development is to define the p.d.f. for the service time "τ" of Figure 4.4.1. We do this first by defining the service time "τ_D" which has an exponential distribution due to the M/M/1 assumption. This p.d.f. is described below.

$$f_{\tau_D}(\tau_D) = \frac{1}{T_D} e^{-\tau_D/T_D} \qquad \tau_D \geq \emptyset$$

$$= \emptyset \qquad \tau_D < \emptyset \tag{4.4.1a}$$

From this we find the mean and variance of equation 4.4.1a.

Mean and Variance for "τ_D" service model:

$$E\{\tau_D\} = T_D \text{ (seconds)}$$

$$\text{VAR}\{\tau_D\} = T_D^2 \qquad (4.4.1b)$$

The next step in the development is to define a p.d.f. for the service time "τ_V". Before we start with the analysis a description of the voice source parameters used in this development are given in Table 4.2 below.

Table 4.2 Voice Source Parameters

$L_V \triangleq$ Voice packet size (bits/packet)

$\lambda_V \triangleq$ Voice packet arrival rate (packets/second)

$R_V = \frac{\lambda_V L_V}{C} \triangleq$ Voice coding rate (bps)

$\rho_V = \frac{R_V}{C} \triangleq$ Link utilization due to voice

$T_V = \frac{L_V}{C} \triangleq$ Service time of voice packets (no data)

$\rho_T = \rho_V + \rho_D \triangleq$ Total link utilization

To define a density function for "τ_V" we first define the set of events that control "τ_V". These are the event that a voice packet is in service and the event that a voice packet is not in service. This set forms a mutually exclusive and exhaustive set of events that completely describe "τ_V" as a service time. Using basic probability we can find the marginal p.d.f. from the conditional p.d.f.'s. This is described in the equation below:

service) is shown in Figure 4.4.2. Since the length of the voice packet is fixed, a data arrival within this region is assumed to have an equally distributed delay time (or service time) until the data packet is actually serviced. This leads to the assumption that the delay is uniformly distributed in time within this interval (when a voice packet is on the link). Thus, the service time "τ_V" is uniformly distributed from $\emptyset$ to T_V (length of voice packet). This argument is not without limitations since it assumes that one data packet will not interact with more than one voice packet. This assumption was introduced in the previous section and its ramification will be addressed later. For now, however, when a voice packet is in service we will assume the conditional p.d.f. is:

$$f_{\tau_V}(\tau_V|\text{voice packet in service}) = \frac{1}{T_V} \text{RECT} \left\{ \frac{\tau_V - T_{V/2}}{T_V} \right\} \qquad (4.4.4)$$

The final conditional p.d.f. that must be defined is $f_{\tau_V}(\tau_V|\text{voice packet not in service})$. This is defined by noting that a voice packet not in service would indicate there is no interaction of the switch function with the data queue. This observation would allow us to say that $\tau_V = \emptyset$ and the queue now looks like an M/M/1 model. With these statements we can define the density function as a common dirac-delta (impulse) function shown below:

$$f_{\tau_V}(\tau_V|\text{voice packet not in service}) = \delta\ (\tau_V) \qquad (4.4.5)$$

The reason for this formulation is because $\tau_V = \emptyset$ for all conditions forcing the total probability of this event occurring to be concentrated at zero.

By combining equations 4.4.3, 4.4.4 and 4.4.5 we arrive at a p.d.f. for "τ_V" shown below.

$$f_{\tau_V}(\tau_V) = \rho_V \cdot \frac{1}{T_V} \text{RECT} \left\{ \frac{\tau_V - T_{V/2}}{T_V} \right\} + (1-\rho_V) \cdot \delta(\tau_V) \qquad (4.4.6a)$$

A drawing of this mixed density function is shown in Figure 4.4.3. The mean and variance of this random variable is given below.

Mean and variance for "τ_V" service model:

$$E\{\tau_V\} = \frac{\rho_V \, T_V}{2}$$

$$\text{VAR}\{\tau_V\} = \frac{\rho_V \, T_V^{\,2}}{12} (4 - 3\rho_V) \qquad (4.4.6b)$$

Thus the density function of "τ_V" is now completely defined for our purposes.

In summary, a density function for the random variable "τ_D" was found (equation 4.4.1a) and a density function for the random variable "τ_V" was found (equation 4.4.6a). Again recalling Figure 4.4.1 we now form a model for the total service time and calculate its moments. Since we have assumed statistical independent service times we add the means and variances of each of the service time random variables and obtain the result below.

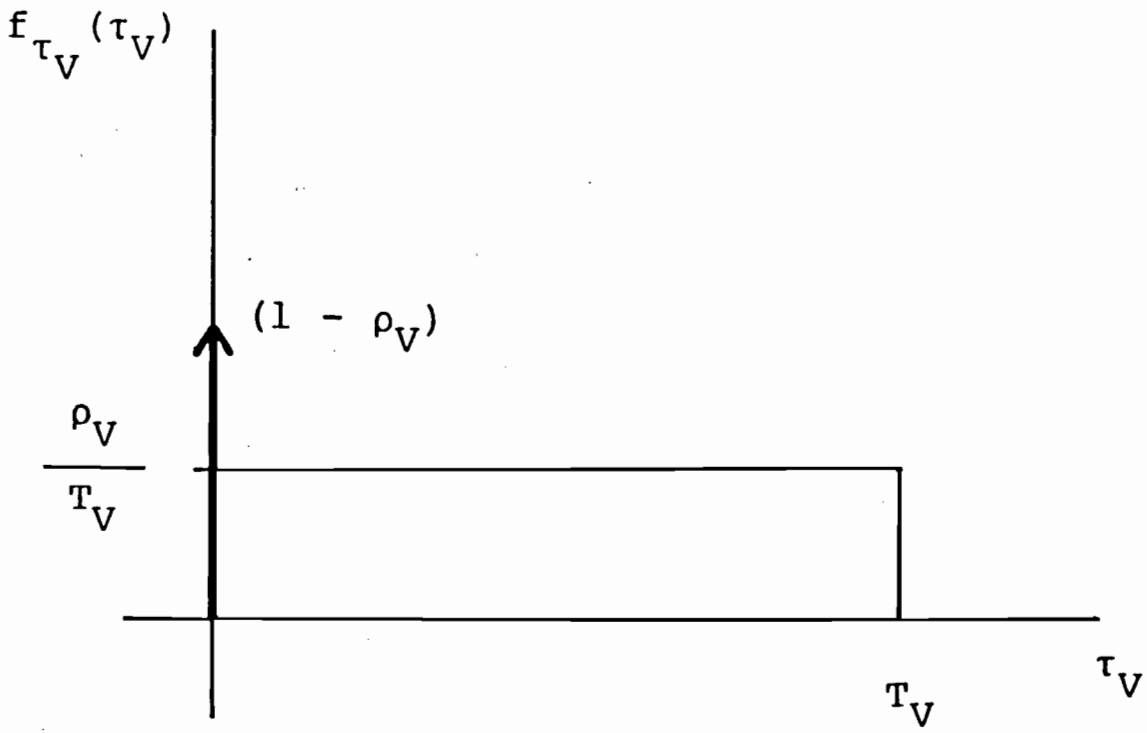


FIGURE 4.4.3 - Service Time p.d.f. for Data Delay Model

$$f_{\tau_V}(\tau_V) = \rho_V \cdot \frac{1}{T_V} \text{ RECT } \{\frac{\tau_V - T_V/2}{T_V}\} + (1 - \rho_V) \cdot \delta(\tau_V)$$

$$E\{\tau_V\} = \rho_V T_V/2$$

$$VAR\{\tau_V\} = \frac{\rho_V T_V^2}{12} (4 - 3 \rho_V)$$

Mean and variance for total service time "τ" :

$$E\{\tau\} = T_D + \frac{\rho_V T_V}{2}$$

$$VAR\{\tau\} = T_D^2 + \frac{T_V^2 \rho_V}{12}(4 - 3\rho_V) \qquad (4.4.7)$$

Now that the density function for the service time in Figure 4.4.1 has been described and its moments determined, the data delay of the M/G/1 queue can be found. From Schwartz [5], the average data packet delay for an M/G/1 queue is described in terms of the moments of the service time. A modified form of this equation from Schwartz [5] (pp. 125) is given below.

$$E\{T_{DP}\} = E\{\tau\} + \frac{\lambda_D}{2}\left[\frac{VAR\{\tau\} + (E\{\tau\})^2}{(1-\rho_{eff})}\right] \qquad (4.4.8)$$

$E\{T_{DP}\} \triangleq$ Average M/G/1 data packet delay (waiting plus service)

An interesting observation to note about equation 4.4.8 is that this formula includes both service and waiting time of an M/G/1 queue. The first term ($E\{\tau\}$) is the service time, as expected, and the rest of the equation defines the waiting time in the queue.

It should also be noted that "ρ_{eff}" is the M/G/1 utilization, not simply the data utilization of the data queue. This is mathematically defined below.

$$\rho_{eff} = \lambda_D \, E\{\tau\} = \lambda_D T_D + \frac{\lambda_D \rho_V T_V}{2}$$

$$= \rho_D + \frac{\lambda_D \rho_V T_V}{2} \qquad (4.4.9)$$

The above equation shows that the effective utilization of the M/G/1 queue contains a term due to the data packets only (ρ_D) and an extra term that contains information about the voice packet interaction in the model $\left(\frac{\lambda_D \rho_V T_V}{2}\right)$. If the voice utilization were zero ($\rho_V = \emptyset$), indicating no voice packets are served, equation 4.4.9 reduces to simply the utilization due to data packets ($\rho_{eff} = \rho_D$).

Using equation 4.4.8 and plugging in the moments calculated in equation 4.4.7 the average data packet delay for the M/G/1 queue is found.

$$E\{T_{DP}\} = T_D + \frac{\rho_V T_V}{2} + \frac{\lambda_D}{2(1-\rho_{eff})}\left[2T_D^{\,2} + \rho_V T_V T_D + \frac{\rho_V T_V^{\,2}}{3}\right] \qquad (4.4.10)$$

The above equation approximates the average data packet delay for an M/G/1 queue model derived for a voice/data link using a non-preemptive priority for voice packets. With this result, an average data queue (buffer) size is found using Little's formula [5].

$$E\{N_{DQ}\} = \lambda_D \, E\{T_{DP}\} \qquad (4.4.11)$$

$E\{N_{DQ}\} \triangleq$ Average data queue size (packets)

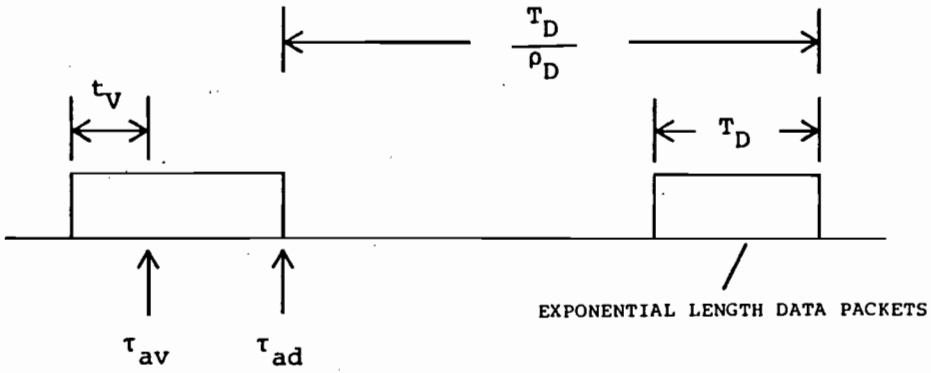
If the intention of this effort was to pick the buffer size for a hardware implementation the value found from equation 4.4.11 could be multiplied by a factor of ten as a rule of thumb. These results completely define the design issues for the data queue. The next subset of this section will look at the voice source and its design issues listed in Table 4.0.

Here the voice source design parameters (voice delay and percentage lost voice packets) are derived using variable length data packets. This derivation uses the same approach as the previous derivation in solving for these parameters. The first parameter that is derived is the voice packet delay due to the non-preemptive model of the switch in Figure 4.3.1. Differing from the previous development, the voice delay that results from the workstation protocol is modeled strictly as a delay and a distribution must be found to describe this delay. In this case "t_v" is the random variable used to distinguish between the data source analysis. The density function for the voice delay time can be written as:

$$f_{t_v}(t_v)=P\text{ (data packet in service)} \cdot f_{t_v}(t_v|\text{data packet in service}) \tag{4.4.12}$$

$$+ P\text{ (data packet not in service)} \cdot f_{t_v}(t_v|\text{data packet not in service})$$

As in the case of equation 4.4.2 the conditional densities need to be described. The first one described is $f_{t_v}(t_v|\text{data packet in service})$ and is best described by looking at Figure 4.4.4 (the



$$t_V = \tau_{av} - (\tau_{ad} - T_D) \longrightarrow \text{EXPONENTIALLY DISTRIBUTED RANDOM VARIABLE}$$

FIGURE 4.4.4 - Data Packet Waveform: Voice Delay Model

$1/\lambda_v = \tau_{av} \triangleq$ Voice packet arrival time (periodic)

$1/\lambda_D = \tau_{ad} \triangleq$ Data packet arrival time (exponential)

$t_v \triangleq$ Delay of voice packet when data is being served

analogous case is described in Figure 4.4.2). In this case a data packet is being served and has exponential message lengths and interarrival times (τ_{ad}). The voice packet arriving within a data packet interval (τ_{av}, shown with an upward arrow) exhibits periodic interarrivals. The delay of a voice packet in this case is described by the variable "t_v". Since the data packet length in time (seconds) is exponentially distributed, an arrival of a voice packet within this interval will cause the delay time "t_v" to have an exponential distributon. Thus, the density is described as:

$$f_{t_v}(t_v|\text{data packet in service}) = \frac{1}{T_D} e^{-t_v/T_D} \qquad t_v \geq \emptyset$$

$$= \emptyset \qquad t_v < \emptyset \tag{4.4.13}$$

The rest of the parameters needed to complete equation 4.4.12 are analogous to the previous equations. These are given below.

- Probability data packet in service = ρ_D
- Probability data packet not in service = $1 - \rho_D$ (4.4.14)
- $f_{t_v}(t_v|\text{data packet not in service}) = \delta(t_v)$

The last density described above is derived from the data packet not in service and all the probability for this occurrance is concentrated at zero.

By combining the above equations a voice delay density can be formed as:

$$f_{t_v}(t_v) = \rho_D \cdot \frac{1}{T_D} e^{-t_v/T_D} + (1-\rho_D) \cdot \delta(t_v) \qquad (4.4.15)$$

This density function completely describes an approximation of the voice packet delay using the non-preemptive protocol. The mean for this density is found by integration and is given below. Note the voice packet delay is a function of data parameters.

$E\{T_{VP}\} \triangleq$ Average voice packet delay

$$E\{T_{VP}\} = \rho_D \cdot T_D \qquad (4.4.16)$$

where:

$\rho_D \triangleq$ Data packet utilization

$T_D \triangleq$ Average service time of data packets (no voice)

The density function for equation 4.4.15 is shown in Figure 4.4.5. This function depicts the average delay of voice packets as they are presented to the link. By definition a voice packet is lost if it is delayed by an amount greater than one voice inter-arrival time (or generation period). Since Figure 4.4.5 represents

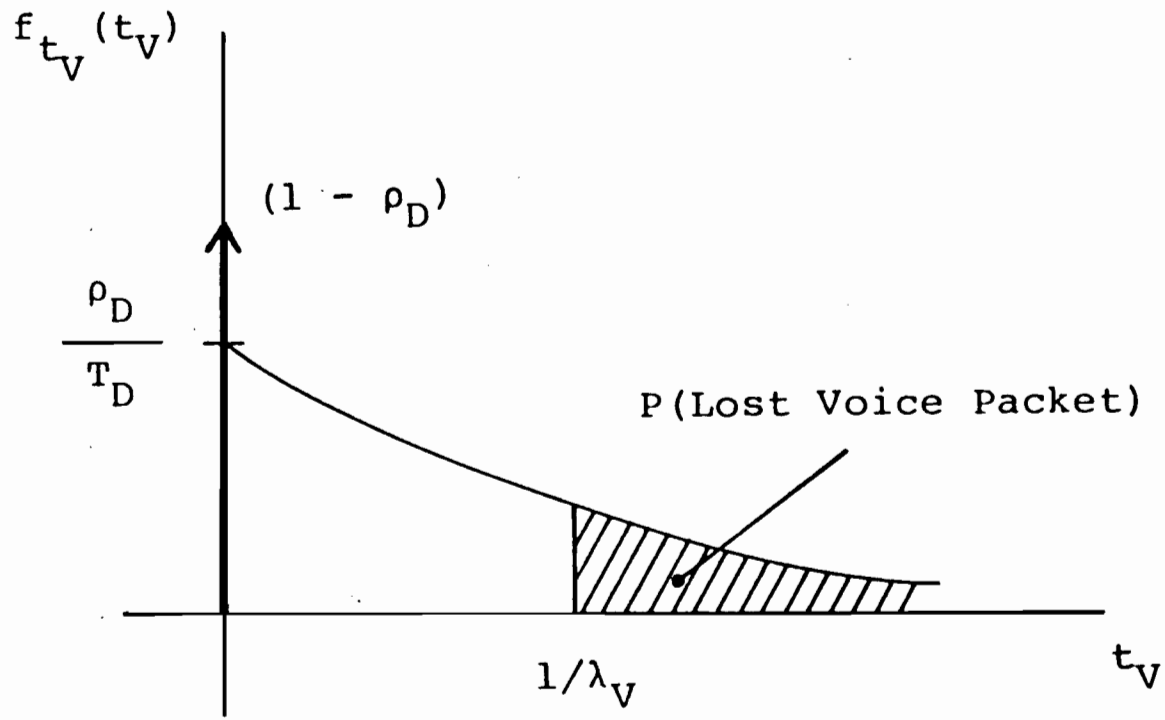


FIGURE 4.4.5 - Voice Delay p.d.f.

$$f_{t_v}(t_v) = \rho_D \cdot \frac{1}{T_D} e^{-t_{v/T_D}} + (1 - \rho_D)\ \delta(t_v)$$

$$E\{T_{VP}\} = \rho_D \cdot T_D \triangleq \text{ Average voice packet delay}$$

the voice packet delay for all time greater than zero, it would make sense that the probability of a lost voice packet would be the area under this density for times greater than one interarrival time (or $1/\lambda_V$). Thus, the percent of lost voice packets or probability of lost voice packets is given by:

$$P\ (\text{Lost Voice Packets}) = \int_{1/\lambda_V}^{\infty} f_{t_v}(t_v)\, dt_v \tag{4.4.17}$$

$$= \rho_D \cdot e^{-1/(\lambda_V \cdot T_D)} \qquad 1/\lambda_V > 0$$

The equation shown above allows us to predict the number of lost voice packets as a function of only three variables. These variables are the same as defined earlier throughout this development.

In summary, we have developed expressions for all four design parameters listed in Table 4.0 using exponential length data packets. It was also mentioned that these predictions are only approximations since they become invalid for high offered loads. The approximation in the model becomes apparent when it is compared using a simulation. A SLAM [4] simulation was created for the queueing model and parameters were collected to compare with the theory. This simulation will be described at a later time. The next section of the report will look at a modification of the above theory for fixed length data packets.

4.5 Theoretical Analysis Using Fixed Data Packets

In this section of the chapter the theoretical analysis given in Section 4.4 will be extended to model fixed length data packets. The reason for this extension is to make the implementation of this type of model easier from a physical standpoint. Hopefully, this approach will perform as well as the previous development and possibly exhibit some desirable characteristics. A comparison between fixed and variable data packets will be given in the last section along with figures generated for several different parameters.

The set of parameters generated in this section are the same as the previous section and those shown in Table 4.0. The first of these parameters is the data delay for the M/G/1 queue. The analysis for fixed length data packets is similar with minor modifications. Recalling the M/G/1 model in Figure 4.4.1, along with the service time distributions for the random variables "τ_V" and "τ_D", the derivation for fixed data packets is explained rather easily. The only important change in the probability density functions for "τ_V" and "τ_D" occurs in the p.d.f. $f_{\tau_D}(\tau_D)$. Since the data packet size is fixed, the service time due to the data packet is constant resulting in a constant mean of "T_D" (average service time of data packets) and a variance of zero. This result leads to a minor change in the previous development. This result is repeated below for convenience.

Mean and variance for "τ_D" service model:

(using fixed length data packets)

$$E\{\tau_D\} = T_D$$

$$\text{VAR}\{\tau_D\} = \emptyset \tag{4.5.1}$$

Furthermore, the p.d.f. for the random variable "τ_V" is exactly the same as equation 4.4.6a since its development is not affected by the constant length of the data packets. Thus, the development for this p.d.f. is not repeated in this section. However, the mean and variance is given below (the same as equation 4.4.6b).

Mean and variance for "τ_V" service model:

(using fixed length data packets)

$$E\{\tau_V\} = \frac{\rho_V T_V}{2}$$

$$\text{VAR}\{\tau_V\} = \frac{\rho_V T_V^2}{12}(4 - 3\rho_V) \tag{4.5.2}$$

Using the same approach as in Section 4.4 the service models above (random variables "τ_D" and "τ_V") are combined using the transformation "$\tau = \tau_D + \tau_V$". The random variable "τ" defines the total service time of the M/G/1 queue. Again assuming statistically independent random variables, "τ_D" and "τ_V", the means and variances of equations 4.5.1 and 4.5.2 are added to form the moments of "τ" below.

Mean and variance for total service time "τ":

(using fixed length data packets)

$$E\{\tau\} = T_D + \rho_V T_{V/2}$$

$$VAR\{\tau\} = \frac{\rho_V \, T_V^{\,2}}{12} (4 - 3\,\rho_V) \qquad (4.5.3)$$

It should be noted at this point that the only difference in equation 4.5.3 and the previous analysis (equation 4.4.7) shows up in the variance of the two different developments. The variance for fixed data packets is reduced by the term "$T_D^{\,2}$" as shown in equation 4.4.7. This reduction in the M/G/1 service time variance (VAR $\{\tau\}$) will exhibit some very interesting properties in the validation section.

The next step in the development is to use the moments of equation 4.5.3 and find the data packet delay in the M/G/1 queue. Equation 4.4.8 from Schwartz [5] (pp. 125) is again used for this purpose. It is repeated below for convenience.

$$E\{T_{DP}\} \triangleq E\{\tau\} + \frac{\lambda_D}{2} \left[\frac{VAR\{\tau\} + (E\{\tau\})^2}{(1 - \rho_{eff})} \right] \qquad (4.5.4)$$

The variables used in equation 4.5.4 are the same as defined in the previous development. A final result is obtained by plugging the moments from equation 4.5.3 into equation 4.5.4 resulting in the equation given below.

$$E\{T_{DP}\} = T_D + \frac{\rho_V T_V}{2} + \frac{\lambda_D}{2(1-\rho_{eff})}\left[T_D^{\,2} + \rho_V T_V T_D + \frac{\rho_V T_V^{\,2}}{3} \right] \qquad (4.5.5)$$

where:

$$\rho_{eff} = \lambda_D\, E\{\tau\} = \rho_D + \frac{\lambda_D \rho_V T_V}{2} \qquad \text{(same as equation 4.4.9)}$$

Equation 4.5.5 above approximates the data packet delay of the M/G/1 queue for fixed data packets. The difference between equation 4.4.10 using variable length packets and equation 4.5.5 using fixed length packets is noticed in the "$T_D^{\,2}$" term enclosed in the last large brackets. The result of using fixed length packets yields a slight reduction in data packet delay. This is displayed in the last section where these models are analyzed.

Again, as before, Little's formula [5] is used to find the average data queue (buffer) size. This is shown below.

$E\{N_{DQ}\} \triangleq$ Average data queue size

$$E\{N_{DQ}\} = \lambda_D\, E\{T_{DP}\} \qquad (4.5.6)$$

The next part in this development looks at the voice source parameters that were described in the previous section. The first parameter that is developed is the voice packet delay that results from using fixed length data packets. Again the random variable "t_v" is used to describe the voice packet delay. As before a voice

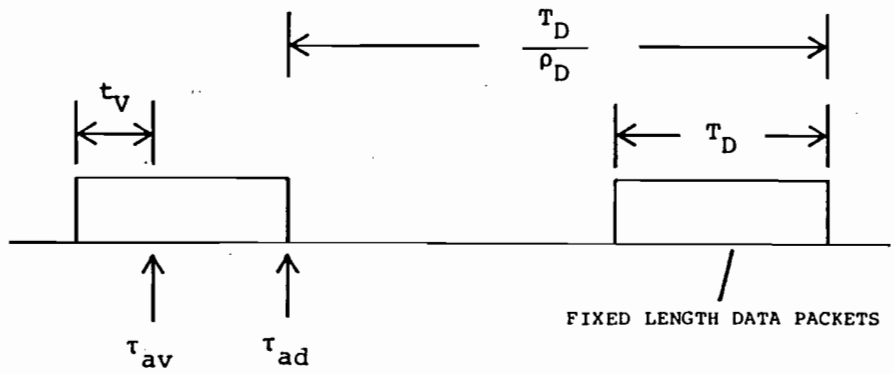
delay density function is needed to describe this process. Equation 4.4.12 is the same in this analysis and is repeated below for convenience.

$$f_{t_v}(t_v)=P(\text{data packet in service}) \cdot f_{t_v}(t_v|\text{data packet in service}) \tag{4.5.7}$$

$$+P(\text{data packet not in service}) \cdot f_{t_v}(t_v|\text{data packet not in service})$$

Proceeding in the same fashion as the variable length data packets, the separate parameters in the above equation need to be defined. The only parameter that is different for the fixed data packet analysis is the conditional p.d.f. $f_{t_v}(t_v|\text{data packet in service})$. This density is described by looking at Figure 4.5.1. This case is synonymous to Figure 4.4.2 with the exception that the data packets are now fixed in length. In this case, from Figure 4.5.1, a data packet is being served and has exponential interarrivals (τ_{ad}). A voice packet arriving within a data packet interval (τ_{av}, shown with an upward arrow) exhibits periodic interarrivals. Since the data packet length in this case is fixed, an arrival of a voice packet within this interval will cause the voice packet to have a delay that is uniformly distributed in time. Thus, the p.d.f. is described as:

$$f_{t_v}(t_v|\text{data packet in service}) = \frac{1}{T_D} \text{RECT} \left\{ \frac{t_V - T_{D/2}}{T_D} \right\} \tag{4.5.8}$$



$$t_V = \tau_{av} - (\tau_{ad} - T_D) \longrightarrow \text{UNIFORMLY DISTRIBUTED RANDOM VARIABLE}$$

FIGURE 4.5.1 - Data Packet Waveform: Voice Delay Model (fixed length data packets)

$\frac{1}{\lambda_V} = \tau_{av} \triangleq$ Voice packet arrival time (periodic)

$\frac{1}{\lambda_D} = \tau_{ad} \triangleq$ Data packet arrival time (exponential)

$t_V \triangleq$ Delay of voice packet when data is being served

Looking back to equation 4.5.7 we need to describe the rest of the parameters in this equation. In this case these parameters are unaffected by the fixed length restriction and are the same as those used in Section 4.4. These are listed below for convenience.

- probability data packet in service = ρ_D
- probability data packet not in service = $1 - \rho_D$ (4.5.9)
- $f_{t_v}(t_v | \text{data packet not in service}) = \delta(t_v)$

Combining equation 4.5.8 and equations 4.5.9 into the equation 4.5.7, a density function for the voice packet delay using fixed length data packets is found. The result is described below.

$$f_{t_v}(t_v) = \rho_D \cdot \frac{1}{T_D} \text{RECT} \left\{ \frac{t_v - T_{D/2}}{T_D} \right\} + (1-\rho_D) \cdot \delta(t_v) \qquad (4.5.10)$$

Using the equation above and solving for its mean, the average voice packet delay is found for fixed data packets.

$E\{T_{VP}\} \triangleq$ Average voice packet delay

$$E\{T_{VP}\} = \frac{\rho_D T_D}{2} \qquad (4.5.11)$$

A noticable difference between equation 4.4.16 (for variable data packets) and equation 4.5.11 (for fixed data packets) is that the mean of the voice delay is smaller by a factor of two for the fixed

data packet case. This will be noted in the model verification section following.

The final parameter that must be defined for the fixed data packet case is the percentage of lost voice packets (or probability). The p.d.f. for the voice packet delay (equation 4.5.10 shown in Figure 4.5.2) will be used in this development. As before Figure 4.5.2 represents the voice packet delay for all time greater than zero and the probability of a lost voice packet is the area under this p.d.f. for all times greater than one interarrival time (or $1/\lambda_V$). Thus, the percent of lost voice packets or probability of lost voice packets is given by:

$$P(\text{Lost Voice Packets}) = \int_{1/\lambda_V}^{\infty} f_{t_v}(t_v)\, dt_v \qquad (4.5.12)$$

$$= \begin{cases} \rho_D \left[1 - 1/(\lambda_V \cdot T_D)\right] & \emptyset < \frac{1}{\lambda_V} < T_D \\ \emptyset & \frac{1}{\lambda_V} > T_D \end{cases}$$

Equation 4.5.12 shows the percentage of lost voice packets using fixed length data packets. One important note is that no voice packets are lost when "$T_D < \frac{1}{\lambda_V}$". This issue is addressed in Section 4.7.

To summarize, we have developed four design parameters (defined earlier) using fixed length data packets. These parameters are also approximate for the same reason as in the previous section (model

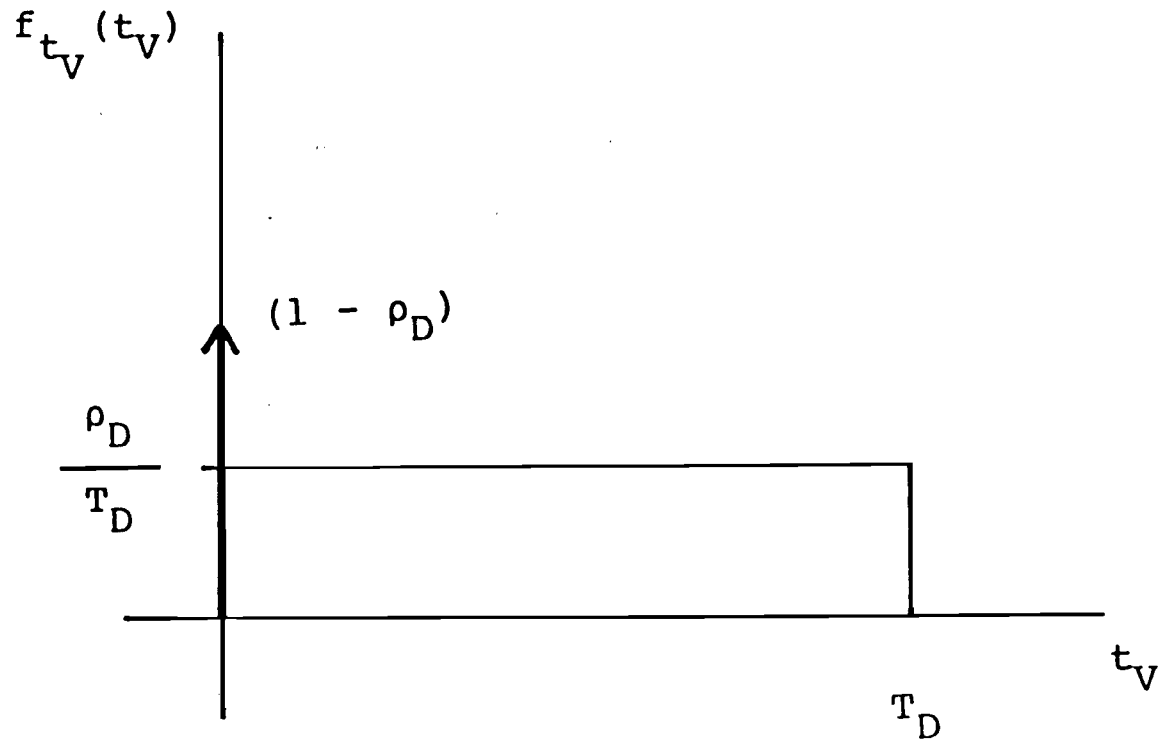


FIGURE 4.5.2 - Voice Delay p.d.f. using Fixed Data Packets

$$f_{t_V}(t_V) = \rho_D \cdot \frac{1}{T_D} \text{ RECT } \left\{\frac{t_V - T_D/2}{T_D}\right\} + (1 - \rho_D) \ \delta(t_V)$$

$$E\{T_{VP}\} = \frac{\rho_D \cdot T_D}{2} \triangleq \text{ average voice packet delay}$$

degrades for high utilization). In the next and final section of this chapter, the parameters that were developed in the previous two sections are put to the test. A SLAM simulation [4] is created for the queueing model and a comparison is made between predicted and simulated values. Some interesting results are found.

4.6 Comparative Measure for the Link Controller

In this development a first approximation of the voice/data link controller is given. This approach allocates capacity to the link using a Hybrid/TDM structure where a fixed capacity is defined for both voice and data. An example of this would be to limit the capacity of the link to 144 Kbps and allow 64 Kbps for voice and 80 Kbps for data ([2,6] and described in Section 4.2). A comparative measure will then be made between the Hybrid/TDM form and the statistical multiplexer developed earlier. With this it will be shown that the Hybrid/TDM form encounters a higher data packet delay at higher utilization because of its fixed link capacity.

The statistical multiplexer, however, relaxes this situation by allowing some of the capacity of either source to be used when needed. The capacity that is stolen is in the form of a delay due to the non-preemptive nature of the link as described in Section 4.4. The trade-off is to allow some voice packet delay (from the non-preemptive link) so that a smaller data packet delay will occur. In a different sense the trade-off is how much voice delay can be tolerated in order to reduce the data delay. This is the basic principle behind the statistical multiplexer and is discussed

in more detail in the next section. Table 4.3 summarizes the two models used in this comparison.

Table 4.3. Summary of Models Used in Comparison

Model	Description
Hybrid/TDM (M/M/1)	1) Assumes fixed link capacity "C" bps 2) Allows 64 Kbps for voice and 80 Kbps for data 3) Model developed for exponential message lengths only 4) Data queue is described as an M/M/1 queue 5) Model degrades at high utilization because capacity is fixed
Statistical Multiplexer	1) Assumes fixed link capacity "C" bps 2) Allows variable capacity for voice and data when needed 3) Model developed for exponential and fixed message lengths 4) Data queue is described as M/G/1 queue 5) Model degrades at high utilization but less than Hybrid/TDM because source capacity is variable 6) Trade-off between how much voice packet delay can be tolerated in order to reduce the data packet delay

Using the descriptions defined in Table 4.3 we compare the two different approaches (Hybrid/TDM or statistical multiplexer). With this comparison we should be able to show that the statistical multiplexer allows a better data throughput at higher utilization (this occurs with an increase in voice packet delay). In the opposite sense, the Hybrid/TDM structure should degrade at high utilization because its capacity is fixed. It should be noted that this comparison is made using exponential message lengths only. A comparison between the two models could also be made using fixed

length messages. To do this the Hybrid/TDM model would need to be defined for the fixed message case (using the M/G/1 equation from Schwartz [5], equation 4.4.8). For our purposes though, the variable data packet case is only considered. This will allow us to model the Hybrid/TDM structure using ordinary M/M/1 equations. This development is presented below.

In the Hybrid/TDM structure the data queue is modeled as an M/M/1 queue with Poisson arrivals and exponential service times. To describe this model we use Little's equation below,

$$E\{T_{DP}\} = E\{N_{DQ}\} \cdot (1/\lambda_D) \tag{4.6.1}$$

where all the variables are as previously defined. The next step is to solve for the size of the data queue "$E\{N_{DQ}\}$" using the M/M/1 equation below,

$$E\{N_{DQ}\} = \frac{\rho_M}{1 - \rho_M} \tag{4.6.2}$$

where "ρ_M" is the utilization of the M/M/1 queue. The utilization is found by first defining the effective capacity of the data queue as:

$$C_{eff} = C - R_V \tag{4.6.3}$$

That is, C_{eff} is the link capacity less the voice packet coding rate. The utilization of the M/M/1 queue is then given by:

$$\rho_M = \frac{\lambda_D \cdot L_D}{C_{eff}} \tag{4.6.4}$$

Where "λ_D" and "L_D" are the data rate and data packet size respectively. With these variables defined the data delay of equation 4.6.1 is determined for the M/M/1 queue and is shown below.

$$E\{T_{DP}\} = \frac{L_D}{(C_{eff} - (\lambda_D \cdot L_D))} \tag{4.6.5}$$

Equation 4.6.5 describes the data delay for the Hybrid/TDM model where the data queue is described as a simple M/M/1 queue with variable message lengths. This equation is used as a comparative measure in the next section to investigate the data delay using the two different approaches introduced in Table 4.3. The validation section following will summarize this comparison in more detail.

4.7 Model Validation

In this section of the chapter the analytical models developed in the previous sections will be verified by using a computer simulation. A complete description of the simulation model (written in SLAM [4]) is given in Appendix C. The output generated from this simulation is given in the following figures along with the predicted performance for each model. Each of the figures below describe one experiment. Within each experiment the simulated and predicted results are compared and the trends in these results are observed. These observations will be discussed in more detail shortly. In this section the results are placed in tables for easy reference.

Before presenting a complete description of the experiments, a summary of the basic assumptions used in these models are described. These assumptions are listed in Table 4.4.

Table 4.4 Basic Model Assumptions

1) Predictions are approximate since the model degrades at high utilization (see Section 4.4).

2) Voice packets arrive periodically and data packets are injected between voice packets being serviced.

3) Data packets arrive exponentially with exponential or fixed message lengths.

4) We are considering only voice and data packets in our models. There is no interest in the contents of these packets.

5) The capacity of the link remains fixed throughout all experiments.

With these assumptions defined, the next step is to describe the experiments. To begin this discussion we will describe the parameters that are used throughout these experiments. The parameters on the dependent axis (Y-axis) are the data packet delay (which includes service plus waiting time in the queue by definition), the voice packet delay (which is simply the waiting time by definition) and the percentage of lost voice packets (which are the packets that are delayed longer than one voice packet interarrival time). The parameters on the independent axis (X-axis) are the data packet size (not including overhead), the total link utilization

Table 4.6. Description of Experiments

Note: All Voice Parameters are the same for all experiments.

Voice Source Inputs:
Coding Rate = 64Kbps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec

Experiment		Other Input Parameters	Description
4.7.1	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D = 30 pak/sec L_D = variable C = 144 Kbps	Determine Data Delay while varying Data Packet Size Using: Prediction - Equations 4.4.10 and 4.5.5 Simulation - SLAM (Appendix C) Hybrid/TDM (M/M/1) - Equation 4.6.5
4.7.2	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D = 15 pak/sec L_D = variable C = 144 Kbps	Determine Data Delay while varying Data Packet Size Using: Prediction - Equations 4.4.10 and 4.5.5 Simulation - SLAM (Appendix C) Hybrid/TDM (M/M/1) - Equation 4.6.5
4.7.3	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D = variable L_D=1024 bits/pak C = 1.544 Mbps	Determine Data Delay while varying Data Rate Using: Prediction - Equations 4.4.10 and 4.5.5 Simulation - SLAM (Appendix C) Hybrid/TDM (M/M/1) - Equation 4.6.5
4.7.4	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D = 30 pak/sec L_D = variable C = 144 Kbps	Determine Voice Delay while varying Data Packet size using: Prediction - Equations 4.4.16 and 4.5.11 Simulation - SLAM (Appendix C)

Table 4.6. Description of Experiments (continued)

Experiment		Other Input Parameters	Description
4.7.5	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D = 15 pak/sec L_D = variable C = 144 Kbps	Determine Voice Delay while varying Data Packet size using: Prediction - Equations 4.4.16 and 4.5.11 Simulation - SLAM (Appendix C)
4.7.6	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D = variable L_D=1024 bits/pak C = 1.544 Mbps	Determine Voice Delay while varying Data Rate using: Prediction - Equations 4.4.16 and 4.5.11 Simulation - SLAM (Appendix C)
4.7.7	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D=15&30 pak/sec ρ_T&T_D = variable C = 144 Kbps	Determine Percentage of Lost Voice Packets while varying Total Utilization and Data Service Time Using: Prediction - Equations 4.4.17 and 4.5.12 Simulation - SLAM (Appendix C)
4.7.8	a) Variable Length Data Packets ---------- b) Fixed Length Data Packets	λ_D=15&30pak/sec ρ_T&T_D = variable C = 1.544 Mbps	Determine Percentage of Lost Voice Packets using: Prediction - Equations 4.4.17 and 4.5.12 Simulation - SLAM (Appendix C)

Figure 4.7.1 shows data delay vs. data packet size and total link utilization for variable (exponential) and fixed length data packets. The link rate is 144 Kbps. The data packet rate in this experiment is 30 packets/sec and as expected the data delay increases as the utilization (or data packet length) increases. Both Figures 4.7.1a and 4.7.1b, show the simulated values vs. predicted values for the data delay equations derived earlier. In particular, Figure 4.7.1a shows the Hybrid/TDM or M/M/1 queue comparison developed in Section 4.6. A complete description of this comparison is listed in Table 4.3. This figure shows that the M/M/1 queue's data delay degrades faster at high utilization because its capacity is fixed. The statistical multiplexer model, however, does not degrade as fast because some of the capacity is shared due to the non-preemptive operation of the switch. A final observation made from this figure is the similarity between fixed and variable data packets. The predicted values for fixed data packets in Figure 4.7.1 are smaller than the variable data packets (comparing equation 4.4.10 to equation 4.5.5). The predicted values in this figure show this trend but the simulated values do not. This may possibly be the result of our approximation or some other source of concern.

Figure 4.7.2 shows the same system configuration as above with the data rate decreased to 15 packets/sec. An important observation to note is the scale change of both the data delay and the data packet size (differing from Figure 4.7.1). The data delay is scaled up due to the decrease in the data rate and the data packet size is

inflated to observe the same variables at the same utilization. The predicted value of data delay for the same data packet size (comparing Figures 4.7.1 and 4.7.2) is about the same, however, the link utilization is quite different from this viewpoint. This is best explained by looking at Table 4.5 entries 5-7. By fixing the data packet size (L_D), the utilization becomes a function of the data packet rate (λ_D). Thus, by decreasing the data packet rate a corresponding decrease will occur in the data utilization and the total link utilization (since the voice utilization is fixed). In the opposite sense, for the same link utilization, for example 70 percent, the predicted data delay in Figure 4.7.1a is lower (compare a predicted data delay of about 15ms for 4.7.1a to a predicted data delay of about 25ms for 4.7.2a). It is also noted that experiment 4.7.1a results in the use of smaller data packet sizes which might be desirable. Another observation made by looking at Figure 4.7.2a is the comparison between the M/M/1 queue and the other values on this figure. This observation follows the same description as in Figure 4.7.1a and also in Table 4.3, Section 4.6. A final observation is that fixed data packets yield the same trends as the variable data packets do in this case.

Figure 4.7.3 shows data delay vs. data rate and total link utilization for a link capacity of 1.544 Mbps. The size of the data packet is fixed at 1024 bits/packet. Again, the simulated vs. predicted vs. M/M/1 values are shown. An important result to note from this figure is that the data delay is virtually unaffected by an

increase in data rate at this high link capacity. Only until the link reaches a very high utilization do the values deviate causing a larger delay. Thus, a high link capacity results in less interaction between data packet size and data rate for the statistical multiplexer. The M/M/1 comparison is also similar because the link capacity is so large that a noticeable change in data delay does not occur until the utilization or data rate is very high. Again, the fixed data packet results in Figure 4.7.3b show the same trend.

Figure 4.7.4 shows voice delay vs. data packet size and total link utilization for a link capacity of 144 Kbps. The data rate is 30 packets/sec. As expected the voice packet delay increases as data packet size increases. An important point to note is that the voice delay is decreased by a factor of two in the fixed packet case using the same input parameters. This is consistent with equations 4.4.16 and 4.5.11. These two equations are repeated below for convenience.

Voice delay for variable data packets (Equation 4.4.16):

$$E\{T_{VP}\} = \rho_D \cdot T_D = \rho_D\left(\frac{L_D}{C}\right) = \lambda_D \cdot T_D^{\,2} = \lambda_D \left(\frac{L_D}{C}\right)^2 \qquad (4.7.1)$$

Voice delay for fixed data packets (Equation 4.5.11):

$$E\{T_{VP}\} = \frac{\rho_D \cdot T_D}{2} \qquad (4.7.2)$$

These equations exhibit some interesting properties that are displayed in Figure 4.7.4. Because the voice delay is reduced by a factor of two in the fixed data packet case, this approach may be more desirable to implement.

Figure 4.7.5 shows the same system configuration as above except the data rate is decreased to 15 packets/sec. Again, the scales are adjusted to yield the same link utilization as in Figure 4.7.4. In the same context as the data delay discussion earlier, a similar discussion is developed for the voice delay (comparing Figures 4.7.4a and 4.7.5a). For a fixed utilization (implying a fixed data utilization "ρ_D"), for example 70 percent, the predicted voice delay is smaller in Figure 4.7.4a (compare a predicted data delay of about 2 ms in Figure 4.7.4a to a predicted data delay of about 4 ms in Figure 4.7.5a). This is accomplished at a higher data rate (30 packets/sec). By looking at equation 4.7.1 and fixing the data utilization "ρ_D", the voice delay becomes a function of the data packet size. Thus, a smaller data packet size for a fixed utilization yields a smaller voice delay (as in Figure 4.7.4). In summary, the explanation described above for Figure 4.7.5 shows that small data packet sizes and large data packet rates are desirable in reducing voice packet delay. This seems consistent with both voice and data delay descriptions (refer to Figure 4.7.1 and 4.7.2 explanations). Furthermore, in Figure 4.7.5 the voice delay is reduced by choosing fixed length data packets. The simulated vs. predicted values also follow this trend.

Figure 4.7.6 shows voice delay vs. data rate and total link utilization for a link capacity of 1.544 Mbps. The data packet size is fixed at 1024 bits/packet. At this high rate the simulated and predicted values are extremely close. The most important observation is that the voice delay is reduced by a factor of two in the fixed packet case. The best description of this interaction is by looking back at equations 4.7.1 and 4.7.2. These equations show that by fixing the data packet size, the voice delay becomes a linear function of the data rate. Again, this data rate and data packet size interaction exhibits interesting features that are useful in our design.

Figure 4.7.7 shows the percentage of lost voice packets vs. link utilization for a link capacity of 144 Kbps. The data rates for Figure 4.7.7a are varied the same as in the previous experiments. In order to place the two different curves on the same figure (vs. utilization) the data packet sizes are varied in parallel, as in the earlier examples. The observation that is made from Figure 4.7.7a is similar to the earlier results. For a fixed utilization, the lowest percentage of lost voice packets occur at a data rate of 30 packets/sec. This seems to be the general consensus of all the experiments. The same results occur in experiment 4.7.7b. The percentage of lost voice packets for the 30 packet/sec case is zero in this figure for the total span of the utilization given (this is not shown in Figure 4.7.7b). One interesting note about Figure 4.7.7b relates directly to equation 4.5.12. When the

upper limit of this equation is reached (i.e., $T_D > 1/\lambda_V$ where $1/\lambda_V$ = 16ms) lost packets begin to occur as depicted in Figure 4.7.7b. This limit occurs when the data service time approaches one interarrival time of voice packets. At that point voice packets are gradually lost. These observations can be used to set limits on some important design variables. The large differences between simulated and predicted values are explained by the large variance in the simulated values [7].

A final experiment was investigated (experiment 4.7.8 in Table 4.6) using the same parameters as experiment 4.7.7 but with a link capacity of 1.544 M bps. In both cases, for variable and fixed length data packets, the percentage of lost voice packets vs. utilization was zero. This important observation is noted for completeness.

The final task to complete in this section is to summarize the above experiments in a convenient manner for future reference. This is done in two different ways. First, the results are summarized in detail in the last few paragraphs and then they are listed in Table 4.7.

In summary, Figures 4.7.1 and 4.7.2 show that a higher data packet rate (30 packets/sec) and a lower number of data bits per packet yield smaller data delays for a fixed utilization. Figure 4.7.3 shows that a higher link capacity results in less interaction between data packet size and data rate. Figures 4.7.4 and 4.7.5 display the following two observations. First, the voice packet

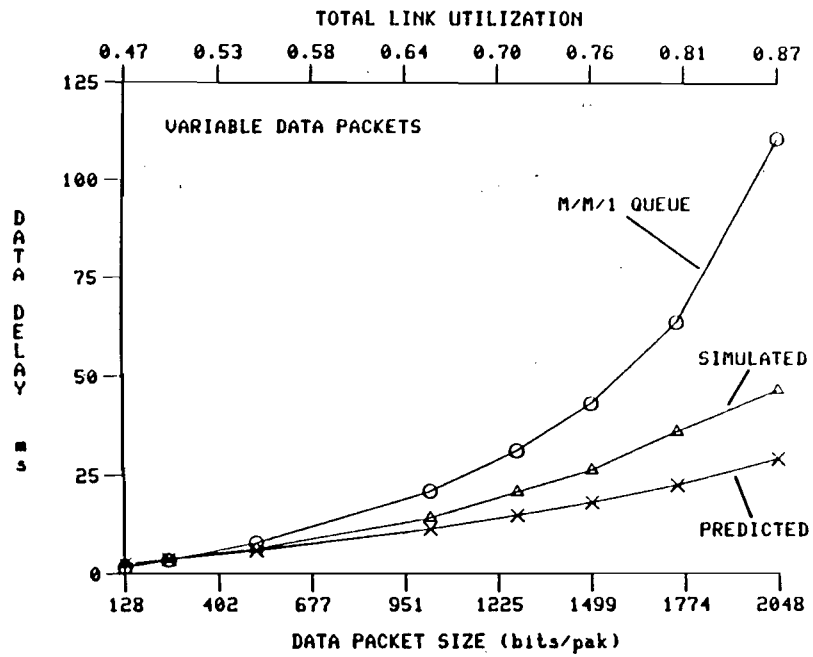
delay is reduced by a factor of two when using fixed length data packets. Secondly, a higher data packet rate (30 packets/sec) and a lower number of data bits per packet yield smaller voice delays for a fixed utilization as in the data description above. Figure 4.7.6 shows the reduction in voice delay between fixed and variable data packets in more detail. And finally, Figure 4.7.7 shows again that a higher data rate (30 packets/sec) and a lower number of data bits per packet is more attractive in yielding a small percentage of lost voice packets. Fixed data packets in this case show an even better improvement in this observation. Experiment 4.7.8 showed for both the 15 and 30 packet/sec case there were no lost voice packets for a variable link utilization. This is a result of the high link capacity and the sensitivity of our measurements. Table 4.7 summarizes these observations in tabular form.

Table 4.7. Experimental Observations

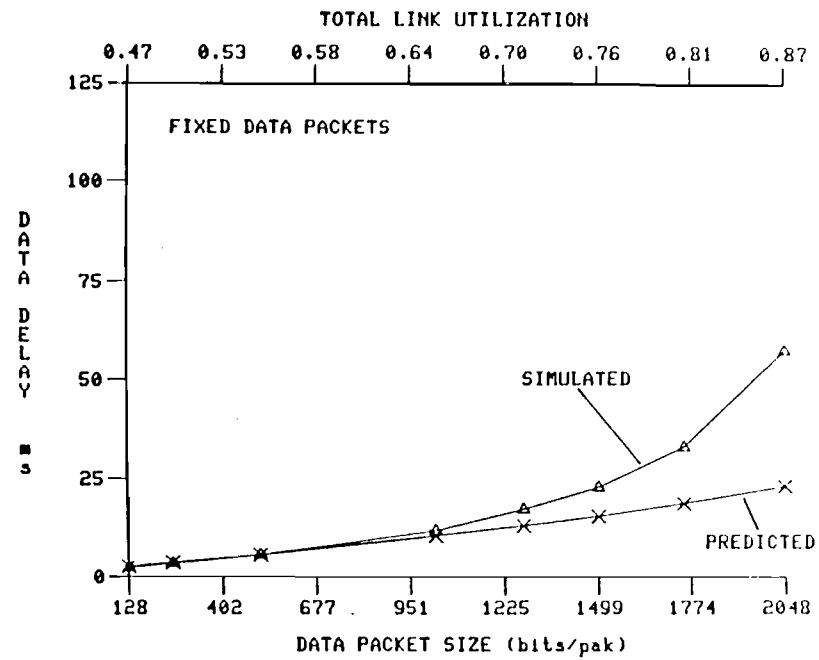
	Experiment	Results
4.7.1	a) Variable Length Data Packets b) Fixed Length Data Pakcets	1) Displayed in Figure 4.7.1 2) Data delay for a fixed utilization is reduced using a higher data rate (30 packets/sec) and a smaller number of bits/packet 3) M/M/1 comparison degrades faster at higher utilization
4.7.2	a) Variable Length Data Packets b) Fixed Length Data Pakcets	1) Displayed in Figure 4.7.2 2) Data delay for a fixed utilization is reduced using a higher data rate (30 packets/sec) and a smaller number of bits/packet 3) M/M/1 comparison degrades faster at higher utilization
4.7.3	a) Variable Length Data Pakcets b) Fixed Length Data Packets	1) Displayed in Figure 4.7.3 2) Higher link capacity results in less interaction between data packet size and data rate 3) M/M/1 comparison is the same as statistical multiplexer because there is more than enough capacity in both cases
4.7.4	a) Variable Length Data Packets b) Fixed Length Data Packets	1) Displayed in Figure 4.7.4 2) Voice delay for a fixed utilization is reduced using a higher data rate (30 packets/sec) and a smaller number of bits/packet 3) Voice Delay is reduced by a factor of two when using fixed data packets
4.7.5	a) Variable Length Data Packets b) Fixed Length Data Packets	1) Displayed in Figure 4.7.5 2) Voice delay for a fixed utilization is reduced using a higher data rate (30 packets/sec) and a smaller number of bits/packet 3) Voice Delay is reduced by a factor of two when using fixed data packets

Table 4.7. Experimental Observations (continued)

	Experiment	Results
4.7.6	a) Variable Length Data Packets b) Fixed Length Data Packets	1) Displayed in Figure 4.7.6 2) Voice delay is reduced by a factor of two when using fixed data packets
4.7.7	a) Variable Length Data Packets b) Fixed Length Data Packets	1) Displayed in Figure 4.7.7 2) The percentage of lost voice packets for a fixed utilization is reduced using a higher data rate and a smaller number of bits/packet 3) In Figure 4.7.7b the 30 pak/sec case is zero for all points
4.7.8	a) Variable Length Data Packets b) Fixed Length Data Packets	1) Not Displayed in any Figure 2) For both 15 and 30 packets/sec the results were no lost voice packets for all points because of the high link capacity (1.544 M bps)



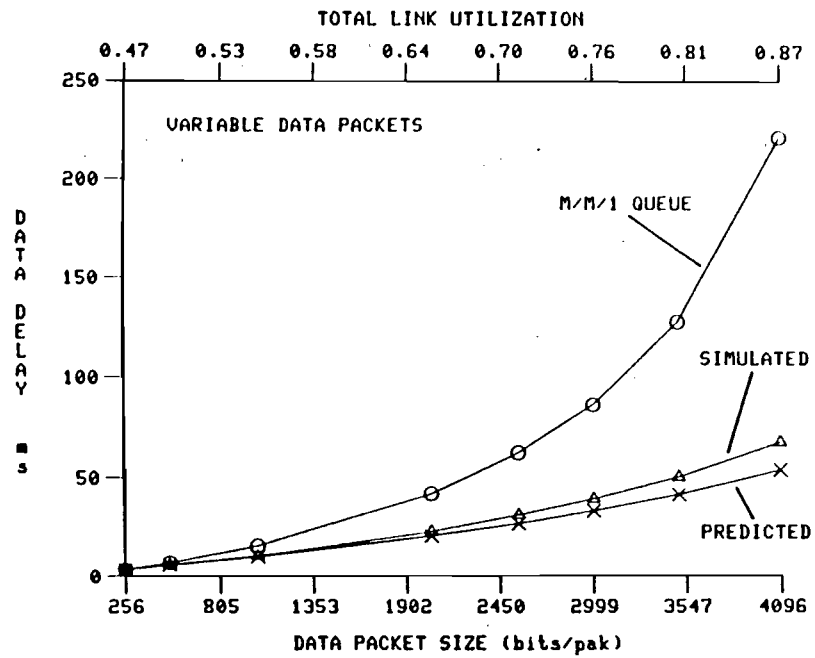
(a)



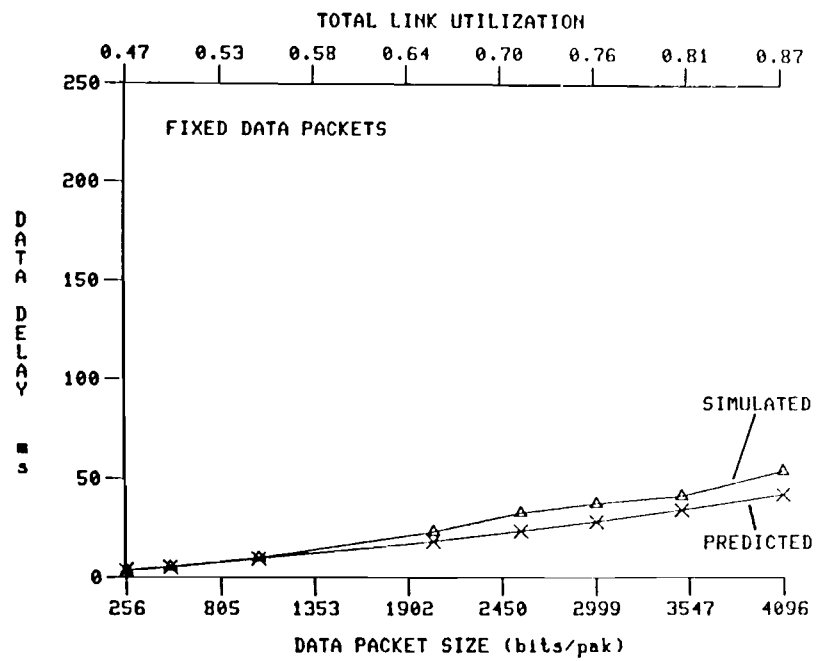
(b)

FIGURE 4.7.1 - Data Delay (Service + Waiting) vs. Data Packet Size and Total Link Utilization (Voice and Data)
Link Capacity = 144K bps, Data Rate = 30 pak/sec
(a) Variable Data Packets (b) Fixed Data Packets

Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec



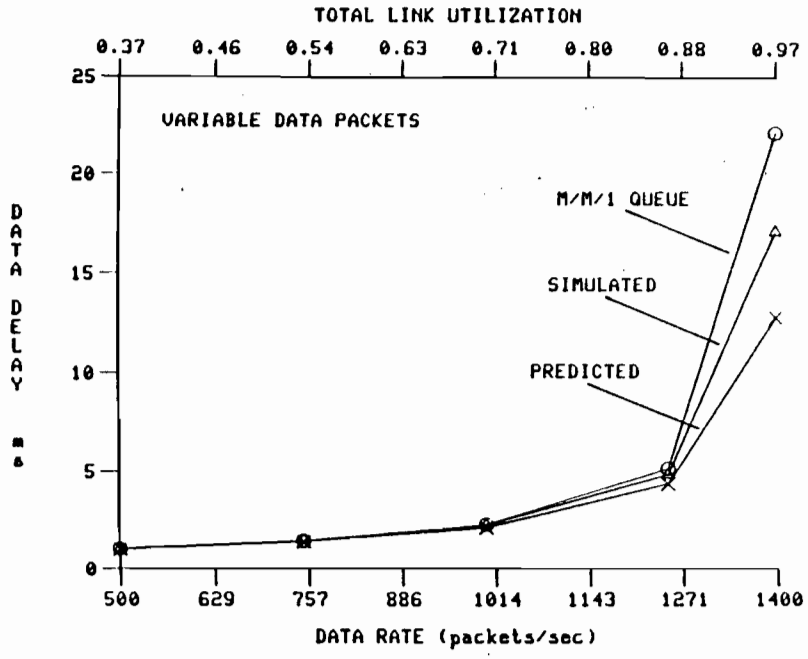
(a)



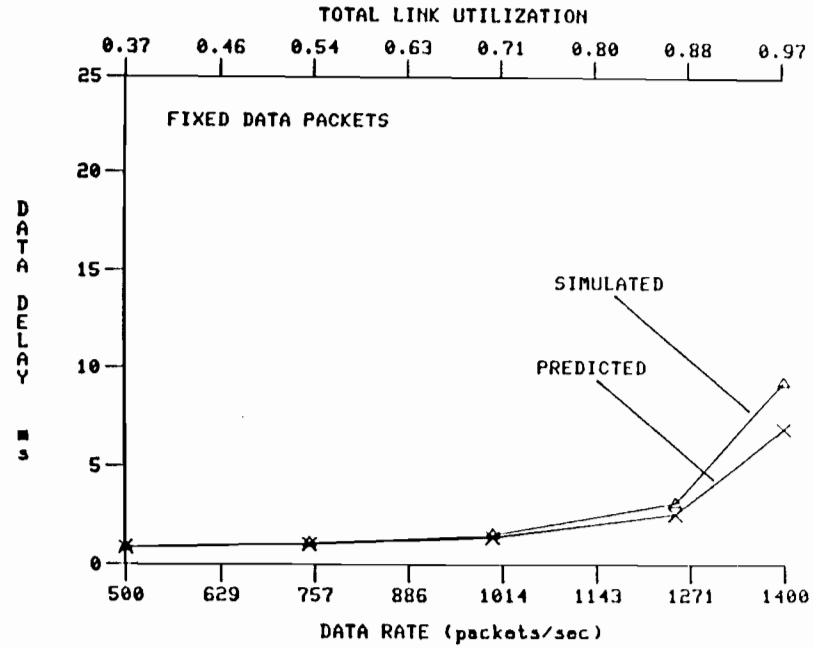
(b)

FIGURE 4.7.2 - Data Delay (Service + Waiting) vs. Data Packet Size and Total Link Utilization (Voice and Data)
Link Capacity = 144K bps, Data Rate = 15 pak/sec
(a) Variable Data Packets (b) Fixed Data Packets

Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec



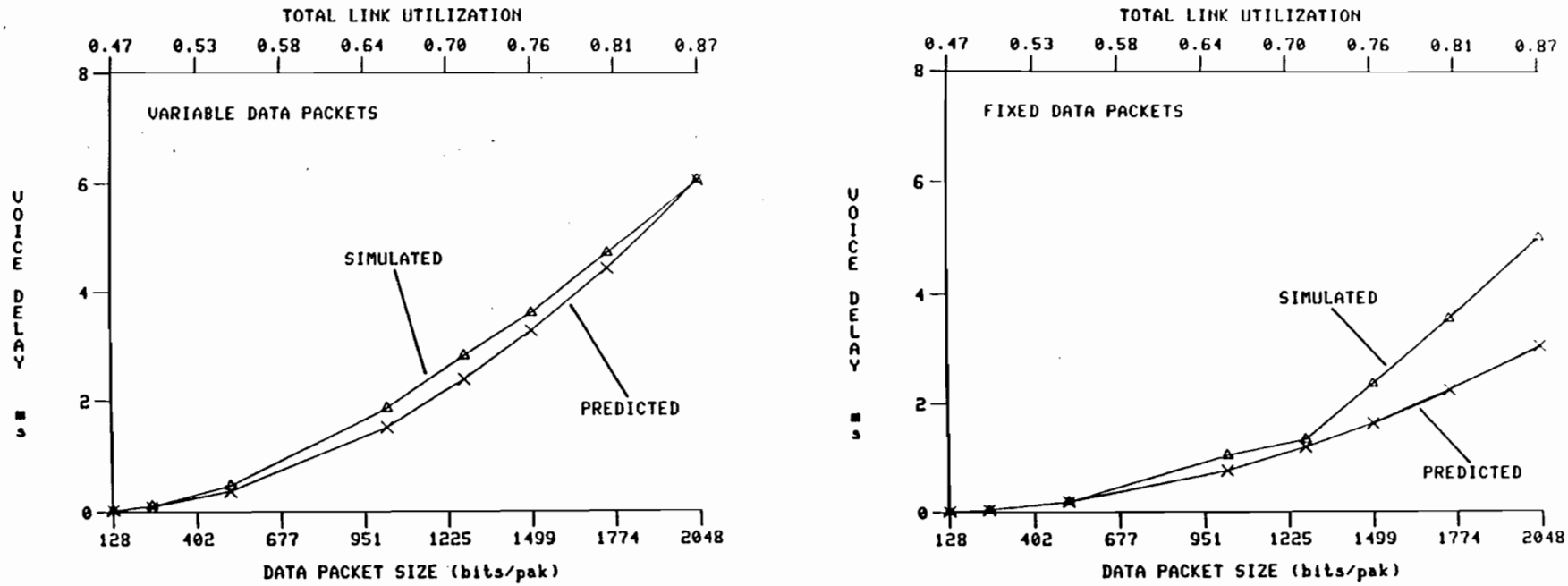
(a)



(b)

FIGURE 4.7.3 - Data Delay (Service + Waiting) vs. Data Rate and Total Link Utilization (Voice and Data)
Link Capacity = 1.544M bps, Data Packet Size = 1024 bits/pak
(a) Variable Data Packets (b) Fixed Data Packets

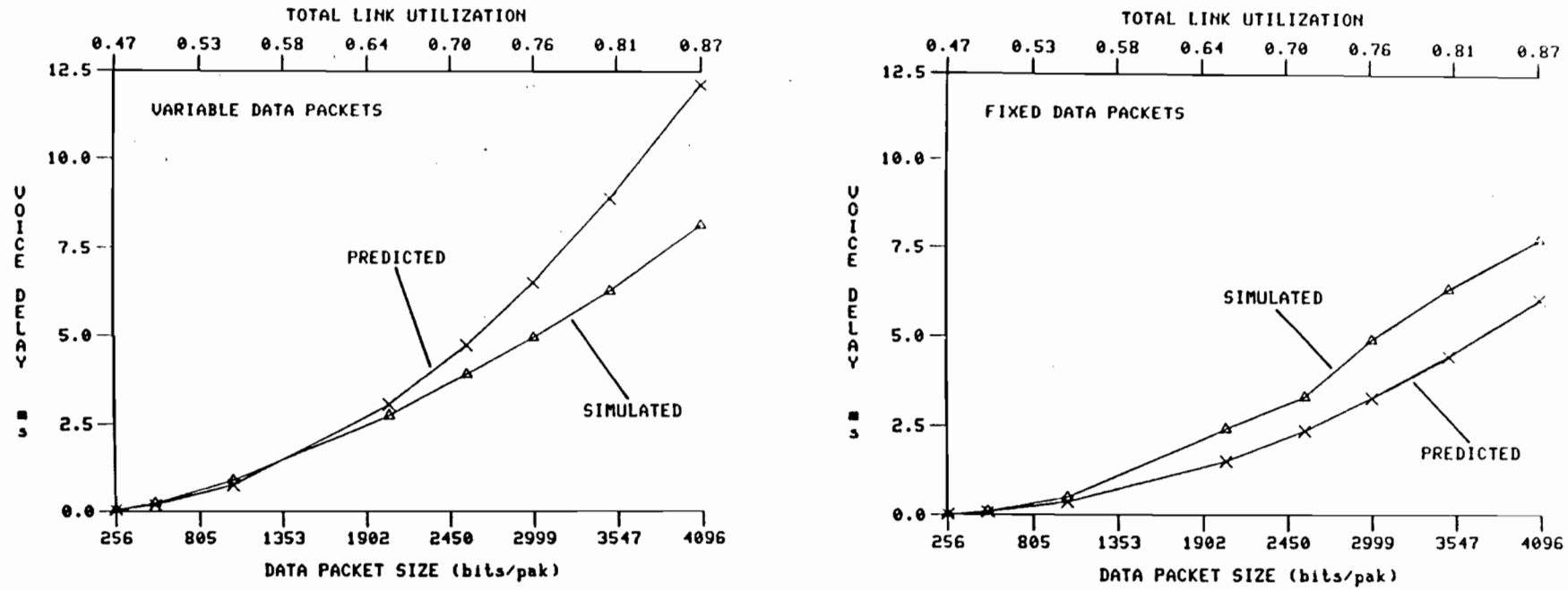
Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec



(a) (b)

FIGURE 4.7.4 - Voice Delay vs. Data Packet Size and
Total Link Utilization (Voice and Data)
Link Capacity = 144K bps, Data Rate = 30 pak/sec
(a) Variable Data Packets (b) Fixed Data Packets

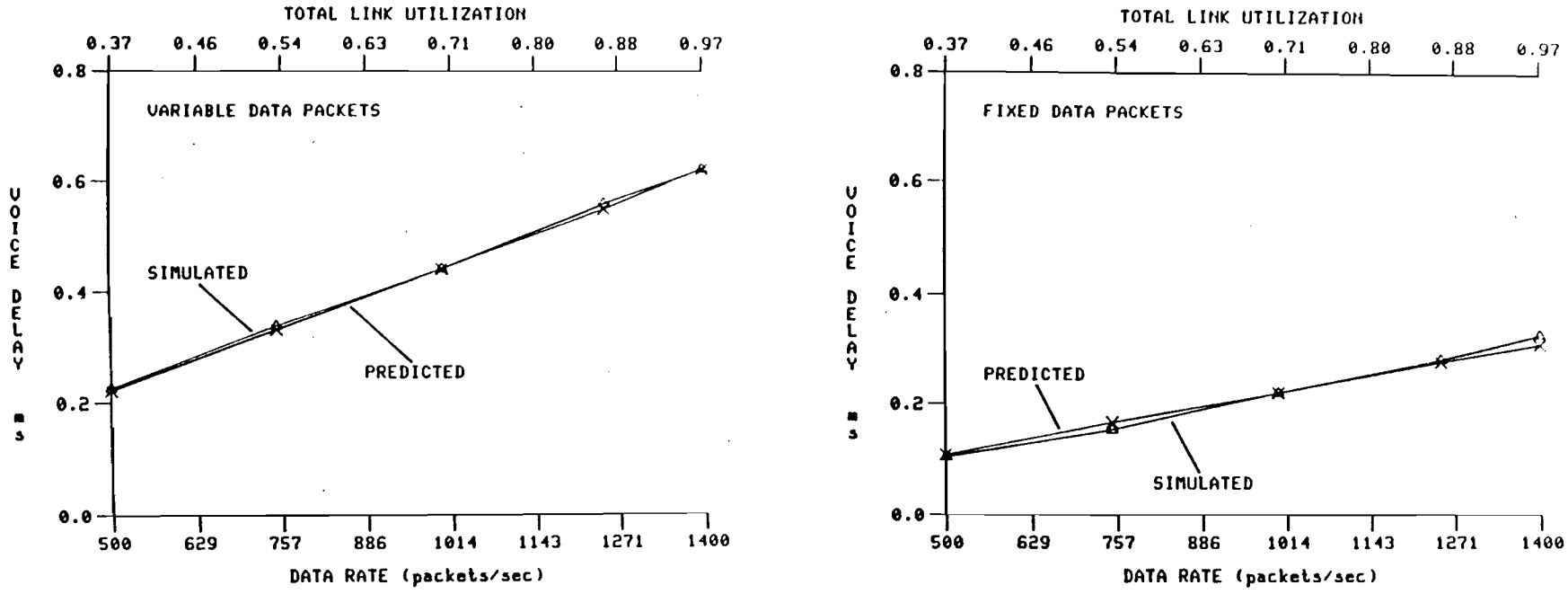
Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec



(a) (b)

FIGURE 4.7.5 - Voice Delay vs. Data Packet Size and
Total Link Utilization (Voice and Data)
Link Capacity = 144K bps, Data Rate = 15 pak/sec
(a) Variable Data Packets (b) Fixed Data Packets

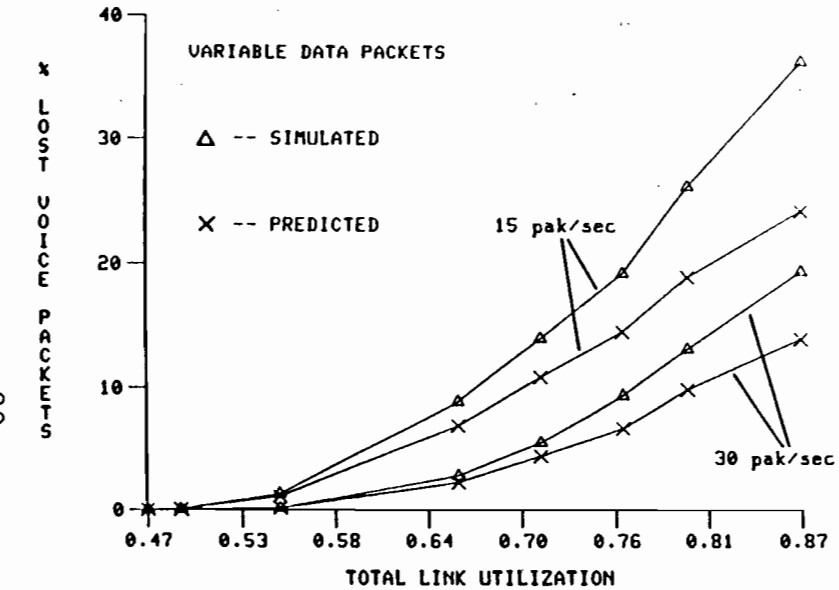
Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec



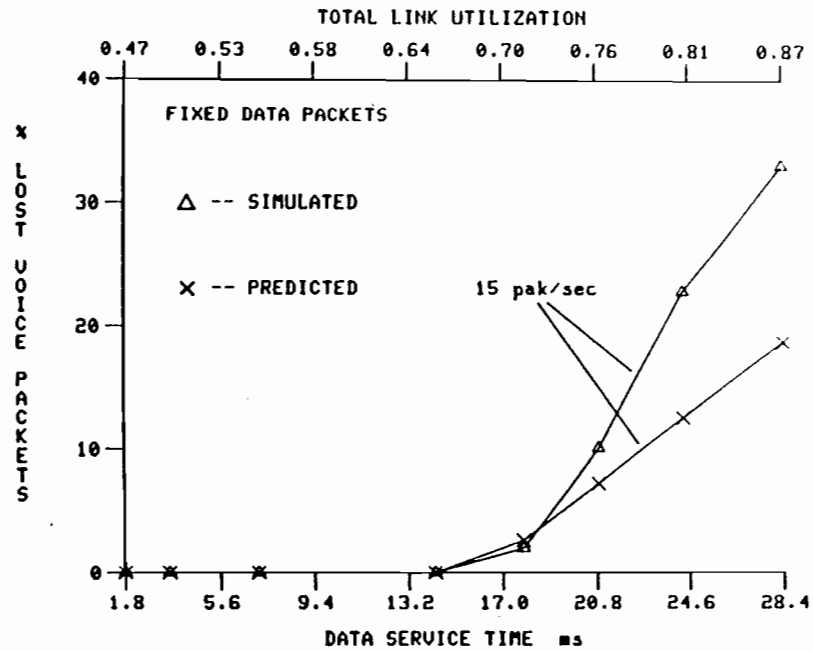
(a) (b)

FIGURE 4.7.6 - Voice Delay vs. Data Rate and
Total Link Utilization (Voice and Data)
Link Capacity = 1.544M bps, Data Packet Size = 1024 bits/pak
(a) Variable Data Packets (b) Fixed Data Packets

Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec



(a)



(b)

FIGURE 4.7.7 - Percentage of Lost Voice Packets vs. Total Link Utilization (Voice and Data) Link Capacity = 144K bps, Data Rates = 15 and 30 pak/sec (a) Variable Data Packets (b) Fixed Data Packets (Includes Data Service Time)

Voice Source:
Coding Rate = 64K bps, Packet Size = 1024 bits/pak, Packet Rate = 62.5 pak/sec

5.0 A Hardware Description

5.1 Introduction

In this chapter a physical description of the packetized voice/data model of Figure 4.3.1 will be introduced. This description will define the voice/data workstation from a hardware point of view and take a look at the system as a whole. In the previous chapter, a theoretical model was developed to gain insight into some important design parameters. This chapter will describe the voice/data workstation as a system and use the theory developed in Chapter 4.0 to enhance this description. The description introduced in this chapter is not intended to be comprehensive. It simply defines an initial approach to the problem from a system engineers' viewpoint.

The organization of this chapter first introduces a flow diagram that describes the workstation and the flow of packets through this system. Some of the assumptions made in this description will also be discussed. Next, a description of the voice/data workstation from a functional viewpoint is presented. This description is intended to address the issues involved in a hardware design. Finally, a summary is given that addresses specific design issues and options relating to the overall design.

5.2 The Flow Diagram

In this section, a system flow diagram will be introduced. This diagram will address some important functions that are needed to describe the full-duplex operation of the integrated voice/data

statistical multiplexer. Figure 5.2.1 depicts the flow of packets into and out of the voice/data workstation. The physical layer and the data link layer in Figure 5.2.1 are defined by the International Standards Organization (ISO). These layers describe Layer 1 and Layer 2 respectively, of the Open Systems Interconnection (OSI) model [11,21]. The intent of the OSI model is that protocols be developed to perform the functions of each layer. The following description attempts to follow this format.

In this description, the voice source is assumed to be digital so that packets can be constructed. The voice bit stream crossing the physical layer in Figure 5.2.1 can be some form of pulse code modulation (PCM) [34]. The definition at this level is left to the designer. The bit streams from the voice and data sources cross the physical layer and enter separate buffers. The voice source enters a one packet buffer and the data source enters a finite packet buffer. A one packet buffer is used for the voice source because a voice packet is considered lost if delayed by more than one voice packet interarrival time. The content of the one packet buffer is simply overwritten when a new packet is delivered. At the receiving end the receiver will have to recognize this event and reconstruct the speech gap by some known procedure [8,10]. It should also be mentioned that this description does not define an acknowledgment (ACK) protocol for voice packets. This is done mainly because sending ACKs would require large voice packet delays which are not tolerated in speech. The main benefit of ACKs, however, is not

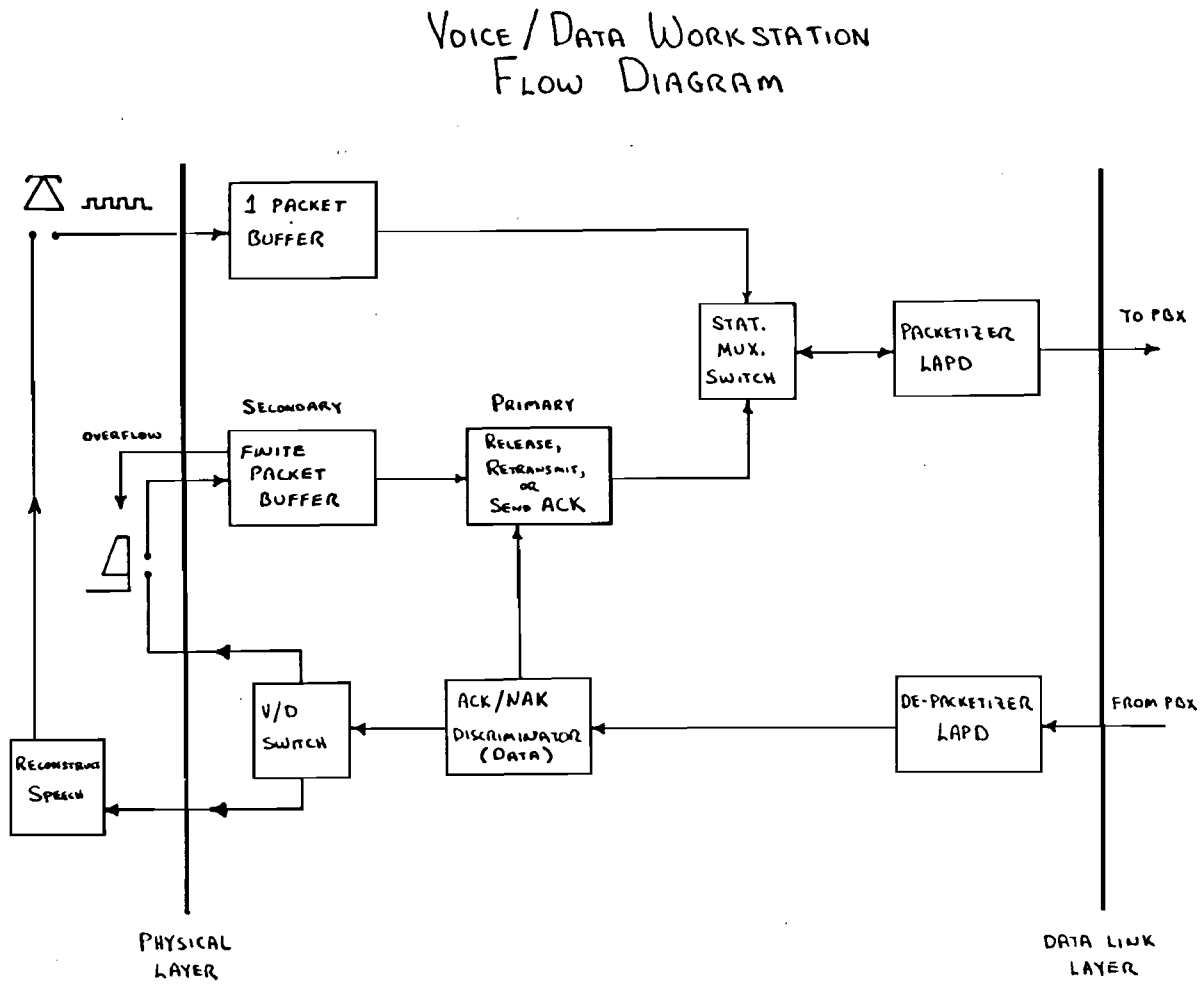


FIGURE 5.2.1 - Voice/Data Workstation Flow Diagram

needed here since voice signals can tolerate an acceptable amount of lost packets and bit errors [9,21].

The data bit stream crosses the physical layer and enters a finite packet buffer. The size of this buffer can be determined using the equations developed in Chapter 4.0. An overflow flag is also used to indicate the status of the data buffer. When overflow occurs the data workstation must have enough intelligence to save the data on a disc or lose it. This concept may also need some special attention in the design. The data buffer is partitioned into two areas in order to handle acknowledgments and negative acknowledgments for data (ACKs and NAKs, respectively). The first area is the primary buffer that saves data packets until an ACK is received. The organization of this function can define ACKs after every received packet or use a sliding window protocol that allows several outstanding packets to build up until an ACK is delivered [11,21]. In either case, this buffer saves the packets in case a retransmission is required. Another function that is contained within this primary buffer is the ability to transmit ACKs to indicate to the receiver that a data packet has arrived. This can be done by placing the ACK to the front of the data queue or allowing this indicator to be added to the outgoing data packet. Whatever the construction, it should be noted that this handshaking is important for data packets to insure a reliable link as well as small error probabilities [11]. The secondary buffer in this diagram is separated only because the contents of this buffer is not

fully committed to being serviced. Also, as described earlier, this secondary buffer must be able to post an overflow flag when required. However the implementation is defined, the above description outlines the issues involved in buffering data packets when acknowledgments are required.

Both voice and data buffers are then presented to the statistical multiplexer switch that combines the packets using the non-preemptive protocol introduced earlier. The switch operates by buffering both voice and data when the packetizer is busy and servicing one packet when the packetizer is idle. This arbitration gives voice packets absolute priority while data packets are serviced between voice packets. The switch implements the basis of the non-preemptive protocol introduced in Chapter 4.0.

The packetizer in Figure 5.2.1 encapsulates the data or voice buffer contents and ships the complete packet out to the network. The switch must inform the packetizer of the origin of the packet so the identity of the packet is not lost. The functions and header information embedded in the packetization process is an important topic to address [13,14]. Both voice and data packets may not use the same protocol format. For example, voice packets need a minimal amount of error correction compared to data packets which need a large amount of error detection/correction. This is because voice can tolerate higher probabilities of error where data can not. Error checking for voice packets may only be needed to preserve the

voice packet identifier [12]. However, simplicity in protocol design for voice and data may override these observations [15].

The family of data link procedures used in the packetization process of Figure 5.2.1 are commonly labeled as Bit-Oriented protocols [11,16]. They are all similar to the familiar HDLC (High-Level Data Link Control). An interesting version of this family is LAP-D (Link Access Protocol-D). This protocol is currently being reviewed by the CCITT in Issue 6 of Recommendation Q.920 [18]. The reason for the interest in LAP-D is because this protocol may become a completely defined standard for the ISDN interface. Thus, these issues must be kept in mind when the packetizer block in Figure 5.2.1 is completely defined.

The return path of the packets from the network in Figure 5.2.1 will now be discussed. The packets are received from the network where the information is removed and errors checked in the de-packetizer block. If the packet contained an ACK or NAK, a signal is relayed to the primary buffer so that old packets can be released or retransmitted respectively (described earlier). Finally, the voice/data switch will route the packets to their appropriate destinations using the header information contained within the packets.

The last important concept to address is the issue of receiving packets. Data packets are simply placed into a buffer and released to the terminal when needed (in an asynchronous fashion). Voice packets, however, must be delivered to the handset in a synchronous

fashion. In this case a synchronous buffer is needed for playback. If a lost packet occurs there must be some form of reconstruction procedure. Both of these issues are addressed in references [9] and [10]. Gruber also addresses some important delay issues in voice/data networks [38].

5.3 A Functional Description of the Transmitter

In this section of the chapter a functional description of the voice/data workstation is given. This is done only for the transmitter block of the workstation described in the previous section. Earlier work in the implementation of packet communication systems utilize microprocessors to implement the functions of the system [19,20]. One goal in our design is to make the system cost-effective. Thus, the approach taken here is to reduce several functions of the workstation on to a single chip. With this approach, we would hope the system will remain reliable as well as cost-effective. As before, it should also be kept in mind that this is an initial description.

Before a complete description of the workstation is given, a listing of some of the basic functions of this design are presented. A listing of these basic functions are given in Table 5.0. Figure 5.3.1 also shows these functions in a diagram form. These functions will be described shortly.

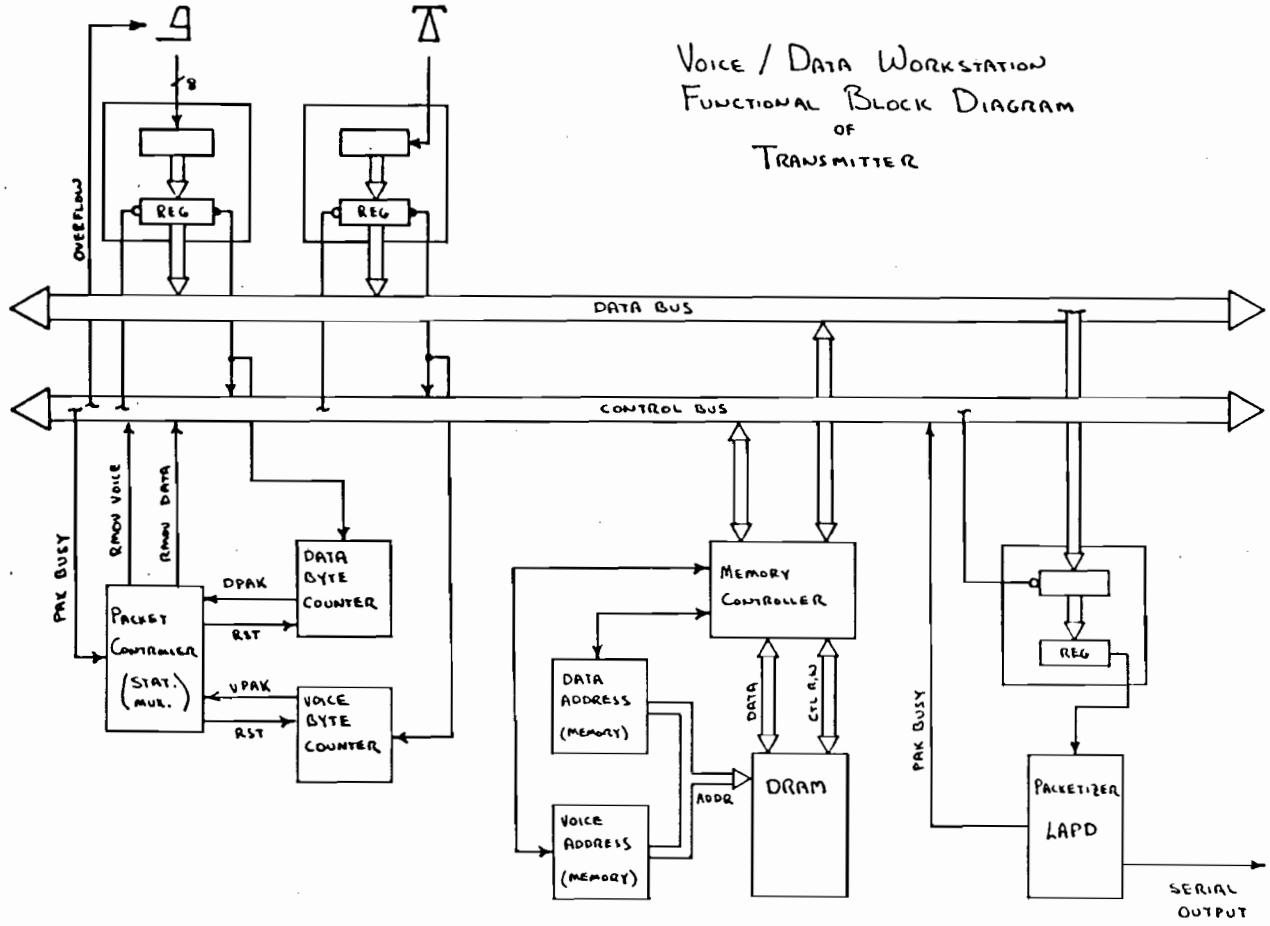


FIGURE 5.3.1 - Voice/Data Workstation Functional Block Diagram of Transmitter

Table 5.0 V/D Workstation Transmitter Functions

1) Voice and Data Inputs (Top Left): This section inputs 8-bits (1 byte) of voice or data information at a time to the data bus.

2) Packet Control (Bottom Left): This section counts bytes that form a packet and initiates removal of a packet from memory.

3) Memory Access (Bottom Middle): This section handles all I/O operations from memory and keeps track of address locations.

4) Packetization (Bottom Right): This section receives packets from memory to be packetized and outputted.

While reviewing the functions in Table 5.0 it must be kept in mind that these functions are being defined to be placed on a silicon chip. The memory function (Dynamic Random Access Memory, DRAM) and the packetization function may or may not be placed on the chip. This is contingent upon the cost of chip real estate. These ideas may change as the design evolves.

The input section in Figure 5.3.1 (Top Left) defines the interface to the data bus. These inputs are presented to the data bus one byte (8-bits) at a time. The voice source shifts this information into the register serially. When the serial shift register is full, a voice byte is sent to a secondary register to be delivered to the data bus. At the same time, the control bus is informed that a voice byte is ready to be serviced. The data source input operates in a similar fashion, with the exception that the information is presented in parallel at the input. When a data byte is ready it follows the same action described previously by inform-

ing the control bus that a data packet is ready to be serviced. It is easily noted that timing constraints might become important as the design becomes more refined. The last signal to define in this section is the overflow signal from the control bus to the data source. This is used to preempt the data source when buffers are full or data bus access cannot be made.

The next section of the block diagram in Figure 5.3.1 is the Packet Control (Bottom Left). The packet controller in this section is the statistical multiplexer that initiates the removal of voice or data packets from memory. The data and voice byte counters receive inputs from the input section discussed earlier and count the bytes as they are placed in memory. When a voice or data packet is complete, the counters tell the packet controller, which initiates a removal of a packet depending on the state of the system. If the packetizer is busy, the packet controller remains idle until service is available. When the packetizer becomes available, the packet controller reviews its state and initiates a packet removal according to the non-preemptive priority discussed earlier. If a voice packet is waiting in memory, it must be removed first; otherwise a data packet is removed. Furthermore, while the packets are being serviced by the packetizer, they cannot be preempted. Thus, the packet controller interfaces with all of the other transmitter functions.

The next function of the voice/data workstation in Figure 5.3.1 is the memory access or buffer store (Bottom Middle). In this

section the buffer is implemented with a Random Access Memory (RAM). This was selected to take advantage of the speed of memory technology and also to remove the memory function from the chip to conserve chip space. The memory controller in this section maintains the I/O functions of the memory buffer. It also utilizes separate memory stores that keep track of the active addresses of the packets stored in memory. The reason for this is to separate the memory address storage from the pure buffer (voice and data) storage. It may be reasonable to place the address storage on the chip while the buffer storage is reserved for dynamic ram (DRAM).

At this point in time the most straightforward way to organize the memory buffer would be a stack structure. As more than one packet enters memory (considering data because only one voice packet is buffered), it is simply placed in sequential order on a stack. The data address block keeps track of the oldest packet and removes that one first. If an overflow occurs, the memory controller will initiate the appropriate signal to the input device. The voice address block should only allow for the maximum storage of two packets. One area is for a packet that is being removed and one for a packet that is being inputted. If more than two voice packets are being operated on, the new incoming packet simply writes over the youngest of the two being serviced and this packet is considered lost. This operation is contained within the packet controller function described earlier.

The last function of Figure 5.3.1 is the packetization function

(Bottom Right). This function receives packets from memory a byte at a time, converts the parallel byte to a serial byte and sends it to the packetizer. The packetizer encapsulates the information and ships the data out serially. A packet busy signal is also posted to notify the packet controller. The packetization function was kept separate from the rest of the description since it is possible the packetizer may be available on a chip using an appropriate protocol for this application. Details of this topic may require additional research.

5.4 Summary of Hardware Issues

In this section of the chapter a summary is given describing some of the issues pertaining to this design. The previous sections introduced a flow diagram (Figure 5.2.1) and a functional diagram (Figure 5.3.1). This section will bring together the ideas developed in the previous sections for easy reference. They are listed in Table 5.1.

Table 5.1 Summary of Hardware Issues

1) Cost Effective
   i) Is a microprocessor a better approach?

2) Chip Implementation
   i) Should the packetizer be on or off the chip?
   ii) Should a Random Access Memory be used as a buffer store?

3) Organization within Packetizer
   i) Is LAP-D the best protocol?

4) Input source parameters
   i) What is best packet size using the development in Chapter 4.0?
   ii) Is there an optimum voice and data packet size?

5) Acknowledgments
   i) Should a sliding window protocol be used?
   ii) How should the data buffer be organized?
   iii) Are short data packets better when considering acknowledgments?

6) Memory Organization
   i) Timing and speed.
   ii) Memory overflow to disc.

7) Digital Telephone
   i) Should an interface be added to the chip?

8) Speech Intelligibility
   i) Is voice acceptable with lost packets?

These issues are summarized as follows: 1) Is the implementation in the previous sections cost-effective or is a microprocessor a better approach yielding more flexibility? 2) If one were to implement this design on a chip, should the packetizer function be included? This depends on the availability of a standard packetizer chip. Also, should a random access memory be used to buffer packets or is this better implemented on the chip? 3) What is the best way to packetize a stream of data bits? This is an important topic that

may include several different protocols. 4) What is the best way to select input parameters for the workstation? Is the theory acceptable in Chapter 4.0 and can it be modified to include an optimization? 5) How should acknowledgments be handled? The sliding window protocol may be important in this design [11]. Also, how should the data buffer be organized to handle ACKs and NAKs? Are short data packets better when considering ACKs? 6) Is the memory organized efficiently and is timing and speed important in this design? Memory overflow must also be handled effectively. 7) Should a digital telephone be used or should an analog interface be added to the chip? 8) How important is speech intelligibility when lost packets are allowed to occur?

Different concepts that are introduced define some important ideas for future expansion. These ideas are summarized in Table 5.2.

Table 5.2 Hardware Enhancements

1) Add Silence Detection
2) Look ahead for periodic voice packets

The addition of silence detection is defined as the removing of voice packets that contain no information, such as pauses in speech. This idea, in addition to other topics in speech are given in reference [21]. Another enhancement would be to look ahead for periodic voice packets and defer data transmission when a voice packet is expected to occur. The goal behind this enhancement is to

reduce the voice packet delay by not allowing the data source to be on the link when a voice packet is expected to arrive. Combinations of silence detection and the look ahead feature can also be defined. References [21] and [22] may be helpful for further investigation.

6.0 **Conclusions**

The theme throughout this report focuses on packet communication technology. The concept of exchanging information by the packet originated in the early 1960s. Since then, the importance of packet communication is recognized by all who work in the telecommunications industry. Some important applications of packet communication have been demonstrated with data, voice and even integrated voice/data networks. Furthermore, the interest in distributed communication networks is also dependent on packet technology. The concepts introduced in Chapters 2.0 and 3.0 provided the necessary background material for this technology.

The major thrust of this report was to define a packet voice/data workstation architecture that interfaces to a Private Branch Exchange (PBX). Our approach defined the workstation by packetizing both voice and data and statistically multiplexing the packets to a PBX over a high capacity link. This multiplexer operated on the packets in a non-preemptive fashion. Furthermore, voice packets were given absolute priority to be delivered as soon as possible.

The approach taken in this description was to analyze the multiplexer using theoretical methods. This analysis led to the prediction of several design parameters. These parameters were data packet delay, voice packet delay and the percentage of lost voice packets. These predictions are only approximate, however, since the model degrades at high link utilization. The predicted values were

then validated by comparing them with a discrete event simulation (SLAM [4]). The simulated values were compared with the predicted values and the similarities were noted. A complete sumary of this comparison is given in Chapter 4.0, Section 4.7.

The important results of the analytical description show that the statistical multiplexer approach offers advantages over an approach that uses a fixed capacity for voice and a fixed capacity for data. The rationale is that our approach allows capacity to be taken when needed, due to the non-preemptive protocol. With this approach, we are trading off voice packet delay for increased link capacity. When the packet delay exceeds a certain amount for voice packets, however, we consider those packets lost. Thus, a subjective measure must be defined to maintain an acceptable level of voice quality.

Other results of the analyses show that small data packets and high data packet rates reduce the voice delay, data delay and the percentage of lost voice packets. By using fixed length data packets the voice packet delay is also reduced.

Finally, the result of introducing a hardware implementation in Chapter 5.0 raised several issues. The basic issue of interest in this report is whether or not a chip implementation is feasible. The results developed in this report show the technical aspects of the problem. More work, however, must be completed in order to show that this implementation can be cost-effective. To do this, continued research is necessary.

One recommendation from this study would be to apply the analytical results from Chapter 4.0 to the hardware design. Other theoretical predictions may be considered to determine whether optimal packet sizes exist for this model. Furthermore, continued research may lead to the investigation of silence detection or other issues raised in Chapter 5.0. With these issues in mind, we can gain further insight in the design and implementation of a packet voice/data workstation.

REFERENCES

1. G.S. Bhusri, "Considerations for ISDN Planning and Implementation," IEEE Commun. Mag., Vol. 22, No. 1, January, 1984.

2. R.M. Wienski, "Evolution to ISDN within the Bell Operating Companies," IEEE Commun. Mag., Vol. 22, No. 1, January, 1984.

3. Paul T. Brady, "A Technique for Investigating On-Off Patterns of Speech," The Bell System Technical Journal, Vol. XLIV, January, 1965.

4. A. Alan B. Pritsker, Introduction to Simulation and SLAM II, Second Edition, Halsted Press Book, John Wiley and Sons, 1984.

5. Mischa Schwartz, Computer-Communication Network Design and Analysis, Prentice-Hall, Inc., N.J., 1977, pp. 117-125.

6. I. Gitman, et al., "Analysis and Design of Hybrid Switching Networks," IEEE Trans. on Commun., Vol. COM-29, No. 9, September, 1981.

7. K. S. Shanmugan and P. Balaban, "A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," IEEE Trans. on Commun., Vol. COM-28, No. 11, November, 1980.

8. P. C. Ravasio, R. Marcogliese, R. Novarese, "Voice Transmission over an Ethernet Backbone," Local Computer Networks, North-Holland Publishing Company, 1982.

9. Nuggehally S. Jayant, "Effects of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample-Interpolation Procedure," IEEE Trans. on Commun., Vol. COM-29, No. 2, February, 1981.

10. Giulio Barberis, Daniele Pazzaglia, "Analysis and Optimal Design of a Packet-Voice Receiver," IEEE Trans. on Commun., Vol. COM-28, No. 2, February, 1980.

11. Andrew S. Tanenbaum, Computer Networks, Prentice-Hall Inc., N.J., 1981.

12. Giulio Barberis et al., "Coded Speech in Packet-Switched Networks: Models and Experiments," IEEE JSAC, Vol. SAC-1, No. 6, December, 1983.

13. H. Ohnishi, et al., "Protocols for NTT High-Speed and Multiplexed Switched Service by the CS-2 Satellite," IEEE Commun. Mag., Vol. 23, No. 3, March, 1985.

14. W. Hoberecht, "A Layered Network Protocol for Packet Voice and Data Integration," IEEE JSAC, Vol. SAC-1, No. 6, December, 1983.

15. Weinstein and Forgie, " Experience with Speech Communication in Packet Networks," IEEE JSAC, Vol. SAC-1, No. 6, December, 1983.

16. D. Carlson, "Bit-Oriented Data Link Control Procedures," IEEE Trans. on Commun., Vol. COM-28, No. 4, April, 1980.

17. M. Decina, "Progress Towards User Access Arrangements in Integrated Services Digital Networks," IEEE Trans. on Commun., Vol. COM-30, No. 9, September, 1982.

18. "CCITT Working Pary XI/6 Draft Recommendation Q.920 (I.441), Specification of the ISDN User Network Interface Data Link Layer Protocol," Issue 6, November, 1983.

19. H. Ueda, I. Tokizawa, T. Aoyama, "Evaluation of an Experimental Packetized Speech and Data Transmission System," IEEE JSAC, Vol. SAC-1, No. 6, December, 1983.

20. J. Flanagan, J. Johnston, J. Upton, "Digital Voice Storage in a Microprocessor," IEEE Trans. on Commun., Vol. COM-30, No. 2, February, 1982.

21. M. R. Aaron, N. S. Jayant, Guest editors, "Special Issue on Bit Rate Reduction and Speech Interpolation," IEEE Trans. on Commun., Vol. COM-30, No. 4, April, 1982.

22. M. Decina and D. Vlack, Guest editors, "Special Issue on Packet Switched Voice and Data Communication," IEEE JSAC, Vol. SAC-1, No. 6, December, 1983.

23. W. Stallings, Data and Computer Communications, Macmillan Publishing Company, New York, 1985.

24. L. G. Roberts, "The Evolution of Packet Switching," Proc. of the IEEE, Vol. 66, No. 11, November, 1978.

25. R. E. Kahn, Guest editor, "Special Issue on Packet Communication Networks," Proc. of the IEEE, Vol. 66, No. 11, November, 1978.

26. D. W. Davies et al., Computer Networks and their Protocols, John Wiley and Sons, New York, 1979.

27. L. Kleinrock, Queueing Systems-Volume I: Theory, John Wiley and Sons, New York, 1975.

28. L. G. Roberts, "Data by the Packet," IEEE Spectrum, February, 1974.

29. L. Kleinrock, "Principles and Lessons in Packet Communication," Proc. of the IEEE, Vol. 66, No. 11, November, 1978.

30. W. Stallings, Local Networks, Macmillan Publishing Company, New York, 1984.

31. F. A. Tobagi, "Multiaccess Protocols in Packet Communication Systems," IEEE Trans. on Commun., Vol. COM-28, No. 4, April, 1980.

32. Metcalfe and Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Commun. of the ACM, Vol. 19, No. 7, July, 1976.

33. D. A. Tugal and O. Tugal, Data Transmission: Analysis Design and Applications, McGraw-Hill, New York, 1982.

34. H. M. Heggestad, "An Overview of Packet-Switching Communications," IEEE Comm. Mag., Vol. 22, No. 4, April, 1984.

35. P. Baran et al., "On Distributed Communications, Vols. I-XI," RAND Corporation Research Documents, August, 1964.

36. K. S. Shanmugan, Digital and Analog Communication Systems, John Wiley and Sons, New York, 1979.

37. K. Kuemmerle, B. W. Stuck, and F. A. Tobagi Guest editors, "Special Issue on Local Area Networks," IEEE JSAC, Vol. SAC-1, No. 5, November, 1983.

38. J. G. Gruber, "Delay Related Issues in Integrated Voice and Data Networks," IEEE Trans. on Commun., Vol. COM-29, No. 6, June, 1981.

39. K. Bullington and J. Fraser, "Engineering Aspects of TASI," Bell Syst. Tech. J., Vol. 38, pp. 353-364, March, 1959.

40. D. Seitzer, H. Gerhäuser and G. G. Langenbucher," High Quality Digital Speech Interpolation Methods," Proc. of the ICC, Toronto, Ont., Canada, 1978, Vol. III, pp. 50.2.1-50.2.5.

41. E. Fuchs and P.E. Jackson, "Estimates of Distributions of Random Variables for Certain Computer Communications Traffic Models," Commun. of the ACM, Vol. 13, No. 12, December, 1970.

42. M. Gerla, R. A. Pazos-Rangel, "Bandwidth Allocation and Routing in ISDN's," IEEE Commun. Mag., Vol. 22, No. 2, February, 1984.

43. J. C. McDonald et al., Fundamentals of Digital Switching, Plenum Press, New York, 1983.

44. R. Dudley, "Evaluating Integrated Voice/Data Terminals," Data Communications, Vol. 13, No. 11, October, 1984.

45. "Feature Section: Integration of Voice and Data Communications," Telecommunications Mag., Vol. 19, No. 2, February, 1985.

46. D. N. Sherman, "Storage and Delay Estimates for Asynchronous Multiplexing of Data in Speech," IEEE Trans. on Commun., Vol. COM-19, August, 1971.

47. M. J. Fischer, "Data Performance in a System Where Data Packets are Transmitted During Voice Silent Periods--Single Channel Case," IEEE Trans. on Commun., Vol. COM-27, No. 9, September, 1979.

48. H. B. Kekre et. al., "Buffer Behavior for Mixed Arrivals and Single Server with Random Interruptions," IEEE Trans. on Commun., Vol. COM-28, No. 1, January, 1980.

49. T. Aoyama et al., "Packetized Service Integration Network for Dedicated Voice/Data Subscribers, IEEE Trans. on Commun., Vol. COM-29, No. 11, November, 1981.

50. K. Sriram et al., "Discrete-Time Analysis of Integrated Voice/Data Multiplexers with and without Speech Activity Detectors," IEEE JSAC, Vol. SAC-1, No. 6, December, 1983.

51. C. J. Weinstein and E. M. Hofstetter, "The Tradeoff Between Delay and TASI Advantage in a Packetized Speech Multiplexer," IEEE Trans. on Commun., Vol. COM-27, No. 11, November, 1979.

APPENDIX A

Glossary of Notation

General Terms

L_D	Average Data Message Length (bits/packet)
R_D	Data Rate (bps)
λ_D	Data Packet Arrival Rate (packets/second)
$\tau_{ad} = 1/\lambda_D$	Data Packet Interarrival Time (seconds/packet)
$T_D = L_D/C$	Average Service Time of Data Packets (seconds/packet)
$\rho_D = \lambda_D T_D$	Link Utilization due to Data Only (no voice) Also probability that a data packet is being served
L_V	Average Voice Message Length (bits/packet)
R_V	Voice Coding Rate (bps)
λ_V	Voice Packet Arrival Rate (packets/second)
$\tau_{av} = 1/\lambda_V$	Voice Packet Interarrival Time (seconds/packet)
$T_V = L_V/C$	Average Service Time of Voice Packets (seconds/packet)
$\rho_V = \lambda_V T_V$	Link Utilization due to Voice Only (no data) Also probability that a voice packet is being served
C	Link (service) Capacity (bps)
$\rho_T = \rho_D + \rho_V$	Total Link Utilization (voice plus data)

Data Source Analysis

τ_V	Data Delay Modeled as a Service Time Random Variable due to the Non-preemptive Priority of Voice Packets
τ_D	Service Time Random Variable for Data Packets
$\tau = \tau_V + \tau_D$	Total Service Time Random Variable for M/G/1 Model
$f_{\tau_V}(\tau_V)$	Service Time p.d.f. for τ_V

Glossary of Notation (continued)

$f_{\tau_D}(\tau_D)$	Service Time p.d.f. for τ_D
ρ_{eff}	Effective Utilization of M/G/1 Model that Includes Non-Preemptive Voice Interaction
$E\{T_{DP}\}$	Average Data Packet Delay (waiting plus service) for M/G/1 Data Queue
$E\{N_{DQ}\} = \lambda_D E\{T_{DP}\}$	Little's Formula to predict size of data queue (buffer)

Voice Source Analysis

t_v	Voice Delay Random Variable due to the Non-preemptive Data Packet Protocol (a delay only)
$f_{t_v}(t_v)$	Delay Time p.d.f. for t_v
$E\{T_{VP}\}$	Average Voice Packet Delay
P(LVP)	Probability of a Lost Voice Packet

M/M/1 Queue Analysis

ρ_M	Utilization of M/M/1 Queue
$C_{eff} = C - R_V$	Effective link capacity for the data source
$E\{N_{DQ}\} = \frac{\rho_M}{1-\rho_M}$	Size of the data queue (packets)

APPENDIX B

Summary of Important Results

I. Analysis using Variable Data Packets (Section 4.4):

1) Average Data Packet Delay for M/G/1 Queue using non-preemptive priority (Equation 4.4.10)

$$E\{T_{DP}\} = T_D + \frac{\rho_V T_V}{2} + \frac{\lambda_D}{2(1-\rho_{eff})}\left[2T_D^2 + \rho_V T_V T_D + \frac{\rho_V T_V^2}{3}\right]$$

where:

$$\rho_{eff} = \lambda_D\, E\{\tau\} = \rho_D + \frac{\lambda_D \rho_V T_V}{2} \qquad \text{(Equation 4.4.9)}$$

2) Average Data Buffer Size (Equation 4.4.11)

$$E\{N_{DQ}\} = \lambda_D \cdot E\{T_{DP}\}$$

3) Average Voice Packet Delay using non-preemptive priority (Equation 4.4.16)

$$E\{T_{VP}\} = \rho_D \cdot T_D$$

4) Probability of Lost Voice Packets (Equation 4.4.17)

$$P(LVP) = \rho_D \cdot e^{-1/(\lambda_V \cdot T_D)} \qquad 1/\lambda_V > 0$$

II. Analysis using Fixed Data Packets (Section 4.5):

1) Average Data Packet Delay for M/G/1 Queue using non-preemptive priority (Equation 4.5.5)

$$E\{T_{DP}\} = T_D + \frac{\rho_V T_V}{2} + \frac{\lambda_D}{2(1-\rho_{eff})}\left[T_D^2 + \rho_V T_V T_D + \frac{\rho_V T_V^2}{3}\right]$$

where:

$$\rho_{eff} = \lambda_D\, E\{\tau\} = \rho_D + \frac{\lambda_D \rho_V T_V}{2} \qquad \text{(same as Equation 4.4.9)}$$

2) Average Data Buffer Size (Equation 4.5.6)

$$E\{N_{DQ}\} = \lambda_D \cdot E\{T_{DP}\}$$

3) Average Voice Packet Delay using non-preemptive priority (Equation 4.5.11)

$$E\{T_{VP}\} = \frac{\rho_D \cdot T_D}{2}$$

4) Probability of Lost Voice Packets (Equation 4.5.12)

$$P(LVP) = \begin{cases} \rho_D \left[1 - 1/(\lambda_V \cdot T_D)\right] & 0 < \frac{1}{\lambda_V} < T_D \\ \emptyset & \frac{1}{\lambda_V} > T_D \end{cases}$$

III. Analysis of Hybrid/TDM Model (Section 4.6):

Note: Developed for variable length data packets only!

1) Average Data Packet Delay using an ordinary M/M/1 Queue (Equation 4.6.5)

$$E\{T_{DP}\} = \frac{L_D}{(C_{eff} - (\lambda_D \cdot L_D))}$$

APPENDIX C

Description of SLAM Simulation

In this appendix we will describe a SLAM simulation that was used to simulate the queueing model given in Figure 4.3.1. SLAM (Simulation Language for Alternative Modeling [4]) is an event-driven simulation used to simulate networks of queues and service activities. The simulation presented here was used to validate the analytical expressions developed in Chapter 4.0. These analytical or predicted results are summarized in Appendix B for easy reference. A listing of the SLAM simulation is given in Figures C.1 through C.6. In all of these figures line numbers were added for easy reference.

Figure C.1 shows the input parameters of the queueing model simulation. The global variables XX(1) through XX(5) define the complete set of input parameters. The definition of these variables are found on lines 23 through 27 of Figure C.1. For this particular simulation lines 4 through 8 show the initialization of these global variables. An example of this initialization would be line 5 which sets the global value XX(2) to 144 Kbps. This is the value of the line rate in the queueing model of Figure 4.3.1.

Figure C.2 shows the actual simulation of the queueing model. This is presented in lines 37 through 86 of code. In this figure lines 43 through 51 simulate the voice source delivering packets to a one packet queue. Line 46 creates voice packets periodically to be placed in the voice queue. Lines 47 and 48 assign characteris-

tics to each packet as it flows into the queue. Line 51 models a one packet queue that accounts for arrivals of voice packets when a voice packet is waiting in the queue. When more than one packet arrives the new packet "balks" to node "LBL1" (Figure C.4 line 107) where a counter keeps track of this occurrence. This is the scheme used to simulate and count lost voice packets during the simulation. Voice packets leaving the voice queue branch to line 72 which models the switch.

Lines 55 through 65 of Figure C.2 models the data source for the queueing model. Line 58 creates data packets with exponential interarrival times. Again characteristics are assigned to the packets. At line 65 the packets enter a data queue which again branches to line 72 to model the switch of the queueing model in Figure 4.3.1.

Before we discuss the operation of the arbitration switch we will first look at Figure C.3. In Figure C.3 the queueing model was revised to handle fixed length data packets. The only difference between Figure C.2 and Figure C.3 occurs in line 61. Figure C.3, line 61 shows how the fixed length packets are created.

The select node on line 72 selects between QUE1 and QUE2 using a non-preemptive priority for the queueing model. The control that is used for this selection is imbedded in the NQS(1) routine shown on line 72 and in Figure C.5. Figure C.5 shows a Fortran subroutine that controls the switch. This is done using a conditional IF, THEN, ELSE structure. Line 11 of Figure C.5 looks into each queue

(file 1 or 2) using a SLAM function NNQ(i). If nothing is in either queue the select node does nothing. At line 18 the condition is whether there is no voice packets and some data packets. When this occurs the data queue is selected and processing continues. The rest of the conditions follow.

Turning back to Figure C.2 we see that as soon as a queue is selected it enters a service activity (Line 75). This activity lasts for a duration that depends on the link rate and the number of bits in the packet. When this service is complete a new packet is selected and at the same time the old packet is routed to Figure C.4 (by way of line 80 or 82) for statistics collection. The structure of lines 120 through 128 of Figure C.4 should be noted. In these lines the simulation is executed from 0 to 50 seconds (line 120). Line 123 clears all the statistical arrays to compensate for queueing transients. Line 124 then runs the simulation over for 100 seconds and collects statistics on the model.

Figure C.6 shows an output summary report for the queueing model. Table C.0 highlights some important features of this report. The labels on Figure C.6 correspond to the labels in Table C.0.

Table C.0 Highlights of SLAM Summary Report

A) Run Number 2 of 2 - statistics collected on second run after arrays have been cleared.

B) Current Time = 100.0 - this simulation lasted 100 seconds.

C) Statistical arrays cleared at time 0.0.

D) Voice waiting plus service time - used to calculate voice delay in queue.

E) Data waiting plus service time - used to calculate data delay in system.

F) Number of voice balks - used to determine how many voice packets are lost.

G) Number of data balks - used to determine how many data packets are lost.

H) Queue 1 (Voice queue) - statistics.

I) Queue 2 (Data queue) - statistics.

J) Service Activity (Link) - statistics: utilization, packet count (shown as entity count)

K) Percentage of Lost Voice Packets - (calculation)

331	= # voice packet balks
5919	= total # voice packets in simulation
331/5919	= .0559 >> 5.59% lost voice packets

```
 1   ;-----------------------------------------------------------------------------
 2   GEN,TISL,STAT MULTIPLEXER,10/22/84,2,NO,NO;
 3   LIMITS,2,3,1500;
 4   INTLC,XX(1)  = 64000.0,
 5         XX(2)  = 144000.0,
 6         XX(3)  = 0.016,
 7         XX(4)  = 0.0333333,
 8         XX(5)  = 1280.0,
 9         XX(20) = 0.0,
10         XX(21) = 1.0,
11         XX(30) = 100.0;
12   ;
13   NETWORK;
14   ;
15   ;      VARIABLE DEFINITIONS
16   ;
17   ;        ATRIB(1) = MARK TIME
18   ;        ATRIB(2) = NODE IDENTIFER
19   ;        ATRIB(3) = # BITS/PACKET (VOICE OR DATA) PRESENTED TO THE CHANNEL
20   ;
21   ;        INPUT PARAMETERS:
22   ;
23   ;        XX(1)     = SAMPLE RATE OF VOICE  ( BPS )
24   ;        XX(2)     = LINE RATE ( BPS )
25   ;        XX(3)     = VOICE GENERATION PERIOD AND ARRIVAL (SECONDS/PACKET)
26   ;        XX(4)     = AVERAGE DATA ARRIVAL (SECONDS/PACKET)
27   ;        XX(5)     = AVERAGE NUMBER OF DATA BITS/PACKET
28   ;
29   ;        UTILITY VARIABLES:
30   ;
31   ;        XX(20)    = PACKET TYPE ON THE CHANNEL (V/D INDICATOR)
32   ;        XX(21)    = DUMMY VARIABLE FOR BALKS
33   ;        XX(30)    = MAXIMUM TIME FOR SIMULATION (SECONDS)
34   ;
35   ;-----------------------------------------------------------------------------
```

FIGURE C.1 - SLAM [4] Network Model: Input Variable Definition (Lines 1 - 35) Note Input Parameters (XX(1) - XX(5))

```
37  ;-----------------------------------------------------------------------------
38  ;     MODEL FOR VOICE / DATA STATISTICAL MULTIPLEXER  ( VARIABLE DATA PACKETS )
39  ;     ==========================================================================
40  ;-----------------------------------------------------------------------------
41  ;
42  ;
43  ;**** MODEL FOR VOICE SOURCE
44  ;      ----------------------
45  ;
46        CREATE,XX(3),0,1                          ; CREATE PERIODIC VOICE PACKETS
47        ASSIGN,ATRIB(2) = 1.0,
48               ATRIB(3) = XX(1)*XX(3),1           ; SET NODE ID AND PACKET SIZE
49  ;                                                A(3) -- BITS/PACKET
50  ;
51  QUE1  QUEUE(1),0,1,BALK(LBL1),SELS              ; QUEUE UP ONE VOICE PACKET
52  ;                                                AND BALK IF OVERFLOW OCCURS
53  ;
54  ;
55  ;**** MODEL FOR DATA SOURCE
56  ;      --------------------
57  ;
58        CREATE,EXPON( XX(4),1.0 ),0,1             ; CREATE DATA PACKETS WITH
59  ;                                                EXPONENTIAL INTERARRIVALS
60        ASSIGN,ATRIB(2) = 2.0,
61               ATRIB(3) = EXPON(XX(5),1.0),1      ; SET NODE ID & PACKET SIZE AS
62  ;                                                BEING A VARIABLE EXPONENTIAL
63  ;                                                A(3) -- BITS/PACKET
64  ;
65  QUE2  QUEUE(2),0,10000,BALK(LBL2),SELS          ; QUEUE UP DATA PACKETS
66  ;                                                AND BALK FOR FINITE QUEUE
67  ;
68  ;
69  ;**** MODEL THE ARBITRATION SWITCH
70  ;      ---------------------------
71  ;
72  SELS  SELECT,NQS(1),,,QUE1,QUE2                 ; SELECT PROPER PACKET USING
73  ;                                                FORTRAN SUBROUTINE NQS(1)
74  ;
75        ACT/1,ATRIB(3)/XX(2)                      ; SIMULATE CHANNEL SERVICE
76  ;                                                (BITS/PACKET)/(LINE RATE)
77  ;
78        GOON,1                                    ; CONTINUE WITH NEXT ACTIVITY
79  ;
80        ACT,,ATRIB(2).EQ.1,VOIC                   ; BRANCH FOR VOICE STATISTICS
81  ;
82        ACT,,ATRIB(2).EQ.2,DATA                   ; BRANCH FOR DATA STATISTICS
83  ;
84        TERM                                      ; TERMINATE
85  ;
86  ;-----------------------------------------------------------------------------
```

FIGURE C.2 - SLAM [4] Network Model: Model for Voice/ Data Mux (Lines 37 - 86) for Variable Data Packets (Line 61)

```
37   ;-----------------------------------------------------------------------------
38   ;     MODEL FOR VOICE / DATA STATISTICAL MULTIPLEXER  ( FIXED DATA PACKETS )
39   ;     ==============================================================
40   ;-----------------------------------------------------------------------------
41   ;
42   ;
43   ;**** MODEL FOR VOICE SOURCE
44   ;     ---------------------
45   ;
46         CREATE,XX(3),0,1                          ; CREATE PERIODIC VOICE PACKETS
47         ASSIGN,ATRIB(2) = 1.0,
48                ATRIB(3) = XX(1)*XX(3),1           ; SET NODE ID AND PACKET SIZE
49   ;                                                A(3) -- BITS/PACKET
50   ;
51   QUE1  QUEUE(1),0,1,BALK(LBL1),SELS              ; QUEUE UP ONE VOICE PACKET
52   ;                                                AND BALK IF OVERFLOW OCCURS
53   ;
54   ;
55   ;**** MODEL FOR DATA SOURCE
56   ;     ---------------------
57   ;
58         CREATE,EXPON( XX(4),1.0 ),0,1             ; CREATE DATA PACKETS WITH
59   ;                                                EXPONENTIAL INTERARRIVALS
60         ASSIGN,ATRIB(2) = 2.0,
61                ATRIB(3) = XX(5),1                 ; SET NODE ID & PACKET SIZE
62   ;                                                TO A CONSTANT
63   ;                                                A(3) -- BITS/PACKET
64   ;
65   QUE2  QUEUE(2),0,10000,BALK(LBL2),SELS          ; QUEUE UP DATA PACKETS
66   ;                                                AND BALK FOR FINITE QUEUE
67   ;
68   ;
69   ;**** MODEL THE ARBITRATION SWITCH
70   ;     ---------------------------
71   ;
72   SELS  SELECT,NQS(1),,,QUE1,QUE2                 ; SELECT PROPER PACKET USING
73   ;                                                FORTRAN SUBROUTINE NQS(1)
74   ;
75         ACT/1,ATRIB(3)/XX(2)                      ; SIMULATE CHANNEL SERVICE
76   ;                                                (BITS/PACKET)/(LINE RATE)
77   ;
78         GOON,1                                    ; CONTINUE WITH NEXT ACTIVITY
79   ;
80         ACT,,ATRIB(2).EQ.1,VOIC                   ; BRANCH FOR VOICE STATISTICS
81   ;
82         ACT,,ATRIB(2).EQ.2,DATA                   ; BRANCH FOR DATA STATISTICS
83   ;
84         TERM                                      ; TERMINATE
85   ;
86   ;-----------------------------------------------------------------------------
```

FIGURE C.3 - SLAM [4] Network Model: Model for Voice/ Data Mux (Lines 37 - 86) for Fixed Data Packets (Line 61)

```
88   ;-----------------------------------------------------------------------------
89   ;
90   ;**** COLLECT STATISTICS ON THE VOICE PACKETS
91   ;     ---------------------------------------
92   ;
93   VOIC  COLCT,INT(1),VCE WAIT p SERV              ; VOICE WAIT + SERVICE TIME
94   ;
95         TERM                                      ; TERMINATE VOICE STATISTICS
96   ;
97   ;**** COLLECT STATISTICS ON THE DATA PACKETS
98   ;     ---------------------------------------
99   ;
100  DATA  COLCT,INT(1),DAT WAIT p SERV              ; DATA WAIT + SERVICE TIME
101  ;
102        TERM                                      ; TERMINATE DATA STATISTICS
103  ;
104  ;**** COLLECT STATISTICS ON THE NUMBER OF VOICE PACKET BALKS
105  ;     -----------------------------------------------------
106  ;
107  LBL1  COLCT,XX(21),# OF VOICE BALKS             ; COUNT NUMBER OF BALKS
108  ;
109        TERM                                      ; TERMINATE
110  ;
111  ;**** COLLECT STATISTICS ON THE NUMBER OF DATA PACKET BALKS
112  ;     -----------------------------------------------------
113  ;
114  LBL2  COLCT,XX(21),# OF DATA BALKS              ; COUNT NUMBER OF BALKS
115  ;
116        TERM                                      ; TERMINATE
117  ;
118        END                                       ; END NETWORK
119  ;
120  INIT,0,50;
121  ;
122  SIMULATE;
123  MONTR,CLEAR,0;
124  INIT,0,100;
125  ;
126  FIN;
127  ;
128  ;-----------------------------------------------------------------------------
```

FIGURE C.4 - SLAM [4] Network Model: Output Statistics Collection (Lines 88 -128) Note Run Clear Run (Lines 120 -124)

```
 1          FUNCTION NQS(N)
 2          COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
 3         1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
 4   C
 5   C      XX(20)    = PACKET TYPE ON THE CHANNEL (V/D INDICATOR)
 6   C
 7          GO TO (1),N
 8   C
 9   C      SELECT PACKET FROM V-D QUEUE'S GIVING VOICE QUEUE PRIORITY
10   C
11       1 IF ( (NNQ(1).EQ. 0) .AND. (NNQ(2).EQ. 0) ) THEN
12   C
13   C          THERE ARE NO ENTRIES IN EITHER QUEUE
14   C
15              NQS = 0
16              XX(20) = 0.0
17   C
18          ELSE IF ( (NNQ(1).EQ. 0) .AND. (NNQ(2).GE. 1) ) THEN
19   C
20   C          NO VOICE PACKETS BUT SOME DATA PACKETS
21   C
22              NQS = 2
23              XX(20) = 2.0
24   C
25          ELSE IF ( (NNQ(1).EQ. 1) .AND. (NNQ(2).EQ. 0) ) THEN
26   C
27   C          NO DATA PACKETS BUT SOME VOICE PACKETS
28   C
29              NQS = 1
30              XX(20) = 1.0
31   C
32          ELSE IF ( (NNQ(1).EQ. 1) .AND. (NNQ(2).GE. 1) ) THEN
33   C
34   C          BOTH QUEUE'S HAVE PACKETS AND VOICE HAS PRIORITY
35   C
36              NQS = 1
37              XX(20) = 1.0
38   C
39          ELSE
40   C
41              WRITE(NPRNT,50)
42      50      FORMAT('  **** ERROR ...  SELECT SUBROUTINE OVERFLOW ****')
43   C
44          END IF
45   C
46          RETURN
47          END
```

FIGURE C.5 - SLAM [4] Network Subroutine: Used in Select Node to Select Voice or Data Queue for Servicing (Line 72, Figures A.2 and A.3)

S L A M S U M M A R Y R E P O R T

SIMULATION PROJECT STAT MULTIPLEXER BY TISL (A)

DATE 10/22/1984 RUN NUMBER 2 OF 2

(B)

CURRENT TIME 0.1000E+03
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

(C)

STATISTICS FOR VARIABLES BASED ON OBSERVATION

		MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
(D)	VCE WAIT p SERV	0.9904E-02	0.6249E-02	0.6310E+00	0.7111E-02	0.8767E-01	5919
(E)	DAT WAIT p SERV	0.2064E-01	0.2148E-01	0.1041E+01	0.1526E-04	0.1533E+00	3033
(F)	# OF VOICE BALKS	0.1000E+01	0.0000E+00	0.0000E+00	0.1000E+01	0.1000E+01	331
(G)	# OF DATA BALKS		NO VALUES RECORDED				

331/5919 = 0.0559

(K)

FILE STATISTICS

	FILE NUMBER	ASSOCIATED NODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
(H)	1	QUEUE	0.1653	0.3715	1	0	0.0028
	2	QUEUE	0.3568	0.8469	8	1	0.0118
(I)	3	CALENDAR	2.6904	0.4620	4	3	0.0085

SERVICE ACTIVITY STATISTICS

	ACTIVITY INDEX	START NODE LABEL/TYPE	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
(J)	1	SELS SELECT	1	0.6901	0.4624	1	0.0000	0.0089	0.5153	8952

FIGURE C.6 - SLAM [4] Network Model: SLAM Output Summary Report