3-1999

A Trade-Off Analysis for Quality of Service in Real-Time Voice Over IP

Tuncel Altunbasak

AFIT/GCE/ENG/99M-01

A TRADE-OFF ANALYSIS

FOR QUALITY OF SERVICE IN REAL-TIME

VOICE OVER IP

Tuncel Altunbasak, 1st Lt.
AFIT/GCE/ENG/99M-01

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1998	Master's Thesis

4. TITLE AND SUBTITLE

A TRADE-OFF ANALYSIS FOR QUALITY OF SERVICE IN REAL-TIME VOICE OVER IP

5. FUNDING NUMBERS

6. AUTHOR(S)

Tuncel Altunbasak, First Lieutenant, TuAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
2950 P Street
WPAFB, OH, 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCE/ENG/99M-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

During the past several years, there has been a significant increase in interest in the use of packetized audio over packet-switched networks due to economic and technical feasibility. However, distribution of real-time voice traffic is basically different from traditional reliable data transfer since continuous voice is sensitive to end-to-end delays and variations of delays. The distribution of continuous voice across a packet-switched network requires consideration of encoding schemas, end-to-end network delays, delay variations, and packet loss, all of which significantly affect the playback quality at the receiving side. There is a trade-off among these factors that affects the quality of service. This trade-off is analyzed in this effort. The packet voice system is modeled and analyzed in order to determine this trade-off. Experimental network measurements are accomplished in order to provide realistic inputs to these simulations. The quality of service (QoS) is measured by end-to-end and delay and probability of gap due to late or lost packets in the analysis. Several mathematical expressions of QoS factors and metrics are developed based on simulation results. These mathematical expressions can be used to optimize the system or to estimate the quality of service for a given operating condition.

14. SUBJECT TERMS

VoIP, Voice over IP, Packet Voice Communication

15. NUMBER OF PAGES

146

16. PRICE CODE

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

The views expressed in thesis are those of the author and do not reflect the official policy or position of the Turkish Air Force, the Department of Defense or the United States Government.

AFIT/GCE/ENG/99M-01

A TRADE-OFF ANALYSIS
FOR QUALITY OF SERVICE IN REAL-TIME
VOICE OVER IP

THESIS

Presented to the faculty of the Graduate School of Engineering

Of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Engineering

Tuncel Altunbasak, B.S.E.E.

1st Lt., TuAF

March, 1999

AFIT/GCE/ENG/99M-01

A TRADE-OFF ANALYSIS

FOR QUALITY OF SERVICE IN REAL-TIME

VOICE OVER IP

Tuncel Altunbasak, B. S.

First Lieutenant, TuAF

Approved:

Gregg H Gunsch

Lt.Col. Gregg H. Gunsch Ph. D.
Committee Chairman

11 Mar 99
Date

Richard A. Raines

Maj. Richard A. Raines Ph. D.
Committee Member

11 Mar 99
Date

Michael Talbert

Maj. Michael L. Talbert Ph. D.
Committee Member

11 Mar 99
Date

Acknowledgment

I would like to express my sincere appreciation to all AFIT faculty members, especially to my thesis advisors Lt.Col. Gregg H. Gunsch, for his valuable input and guidance. I would also like to thank my reader, Maj. Richard A. Raines and Maj. Michael L. Talbert, for proofreading this thesis and offering suggestions.

I would also like to express my gratitude to my wife, Mrs. Ferhan Altunbasak, and my brother, Dr. Yucel Altunbasak, for their enduring love, words of encouragement, support and inspiration.

Finally, I would like to dedicate this thesis to my son Egemen Altunbasak who had to spend his 4 months away from me to keep me on schedule and provide a better research environment.

Above all, I promise to serve my country and people with my best to be able to express my thanks for sending me to this challenge for two years at AFIT.

Table of Contents

List of Figures

List of Tables

Abstract

During the past several years, there has been a significant increase in interest in the use of packetized audio over packet-switched networks due to economic and technical feasibility. However, distribution of real-time voice traffic is basically different from traditional reliable data transfer since continuous voice is sensitive to end-to-end delays and variations of delays. The distribution of continuous voice across a packet-switched network requires consideration of encoding schemas, end-to-end network delays, delay variations, and packet loss, all of which significantly affect the playback quality at the receiving side. There is a trade-off among these factors that affects the quality of service. This trade-off is analyzed in this effort. The packet voice system is modeled and analyzed in order to determine this trade-off. Experimental network measurements are accomplished in order to provide realistic inputs to these simulations. The quality of service (QoS) is measured by end-to-end and delay and probability of gap due to late or lost packets in the analysis. Several mathematical expressions of QoS factors and metrics are developed based on simulation results. These mathematical expressions can be used to optimize the system or to estimate the quality of service for a given operating condition.

1 Introduction

1.1 Motivation

Today, both the telephone and the personal computer are fundamental helpers in most offices and homes. These helpers are connected to different networks. The telephones are connected to a telecommunications (telephony) network that is circuit-switched and designed for point-to-point communication of real-time audio. The computers, on the other hand, are connected to data networks that employ store-and-forward packet technologies created primarily for data transport over local and wide areas.

Traditionally, continuous voice service was carried over circuit-switched telecommunication networks by using analog methods and considered separate from digital computer communications. But new application areas are introducing radical changes to the field of computer networking. High-speed fiber optic networks and increasingly powerful desktop computers are driving the trend toward a much higher degree of connectivity and the incorporation of new standards. Applications such as digital multimedia and distributed computing are creating a new network traffic mix and introducing network flows with requirements quite different from traditional applications. The new mix of network traffic has a rapidly growing emphasis on digital continuous media. Digital representation of audio signals is fundamentally attractive since it offers more flexibility in manipulating and processing this data type than the analog representation. Integration of digital continuous media in general-purpose computing systems promises to significantly enhance the quality and bandwidth of human-computer interaction. As computing and communication merge, new digital communication services will be created.

In the recent years, real-time distribution of continuous voice traffic over packet-switched networks has become increasingly attractive due to economical and technical feasibility. Packet switching can be an effective technology to integrate voice and data in a single network since it can exploit the bursty nature of data and voice to reduce the transmission bandwidth needed to carry a particular mix of traffic over a circuit network. Packet switching also offers some other advantages, including a more flexible allocation of bandwidth to individual calls, and interfaces. The potential gain in efficiency is significant. Studies show 20-30% of the time in a voice conversation consists of silence [Bra65].

Distribution of real-time voice traffic is basically different from traditional reliable data transfer, such as file transfer, remote login and electronic mail, since real-time voice data is sensitive to end-to-end delays and variations in the delays. Traditional data traffic requires a transmission channel with a low error rate and some minimum bandwidth. The performance metrics of these applications are typically average packet delay and throughput. On the other hand, voice traffic requires some upper bound on the delay of the voice packets. This delay bound is an end-to-end constraint of the application level. If the packet arrives at the packet voice receiver after its playout time, it may be useless or have significantly lost its value. Early arrival may even be undesirable since it requires buffering at the receiver. One advantage, however, is that, packet voice systems are able to tolerate some loss of the packets up to a limit.

1.2 Background

The nature of human speech consists of an alternating series of speech activity periods or talkspurt, and silence periods. The packet voice source takes continuous analog voice signals generated by user and generates voice packets, as shown in Figure 1.1.

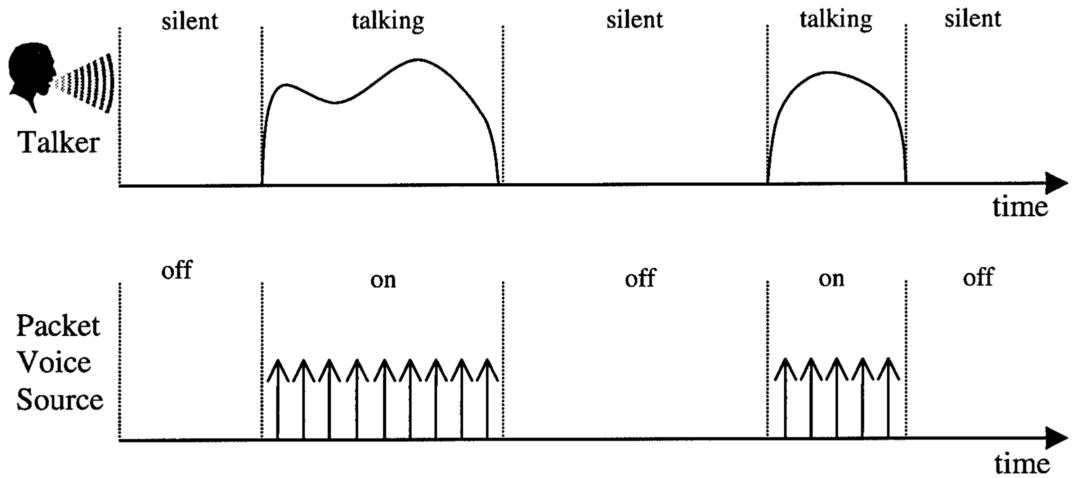


Figure 1.1 Typical Human Speech and Voice Source Behavior.

A voice source generates packets, in the ON period, when the talker is actually speaking. During this period of the packet voice source, a coder digitizes a continuous analog voice signal generated by user. In the process of digitizing, the analog voice signal is sampled at the Nyquist rate (twice the signal rate) in order to recover the original signal correctly. Human speech or voice is typically in the 0-4 kHz frequency range. Therefore, the voice-sampling rate is 8 kHz. This means one sample every 125 μs. For instance, a typical Pulse Code Modulation (PCM) encoder produces an 8-bit word every 125 μs. The generated samples are then accumulated in a packetizer for a fixed period of time, known as the *packetization interval.* When the packetization time reaches the predetermined packetization interval, a header is attached and a voice packet is generated. These packets are then placed into a network. Figure 1.2 shows the packet voice system block diagram.

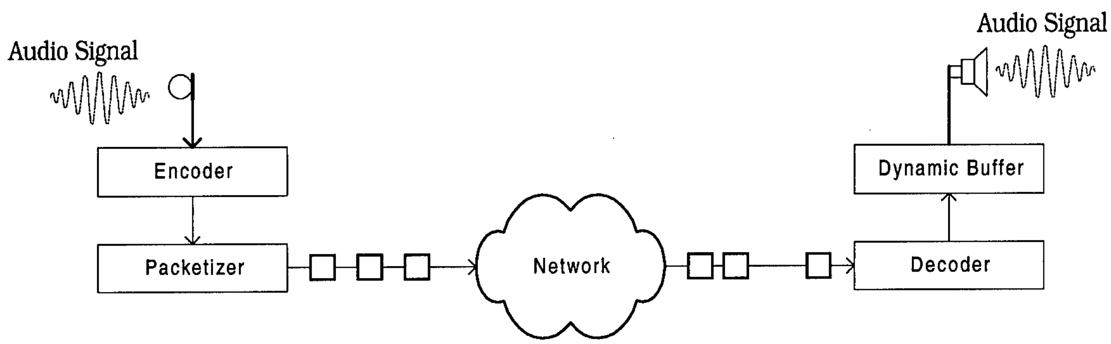


Figure 1.2 Packet Voice System Block Diagram

The packet voice source generates no packets during periods in which the speaker is silent. If the network delays of these packets are free from variations, a receiving site can simply play out an audio packet as soon as it is received. However, statistical multiplexing in packet switched networks introduces variations in the network delay experienced by individual packets. These variations of transit delays are called *jitter.* Since the network delay is not free from jitter, packets may arrive out of order and before or after their playback time. In order to compensate for these variable delays, a buffer is used at the receiver. The first packet in a talkspurt is artificially delayed for a period of time known as the *control time* (also known as *playout delay*) before it plays. The control time builds up a buffer of arriving packets in the presence of delay jitter.

The packets arriving with shorter delay may have to wait in the receiver's buffer in order for the packets with longer delay to arrive before their playback time.

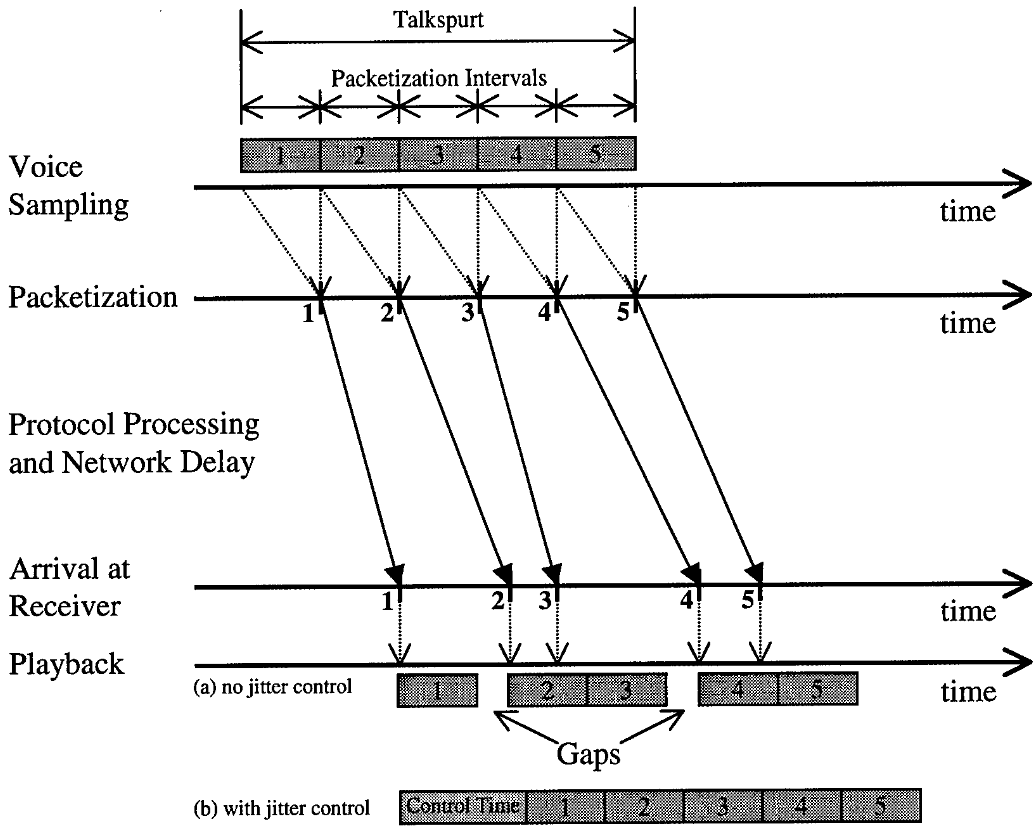


Figure 1.3 Transmission and Playback of a Talkspurt

If the packet does not arrive at or before the playback time, discontinuity of the voice playback, which is known as *gap*, occurs at the receiver, as shown in Figure 1.3. Such gaps are undesirable since they destroy the continuous playback of voice as well as human conversation.

It is obvious that to eliminate gaps completely, the control time must be set equal to the maximum variation of the network delay. However, the control time can not be arbitrarily large due to quality of service constraints on the end-to-end delay. For instance, high-quality voice applications require less then 200 ms round-trip delay [BoG98] [RaR92].

1.3 Quality of Service in IP Network Distribution of Real-Time Voice

The distribution of continuous voice across a packet-switched network requires consideration of all factors that significantly affect the quality of the playback at the receiving side. The issues that are important for high quality transmission are encoding schemes, end-to-end delays, delay variations, and packet losses.

1.3.1 Encoding

In recent years, considerable progress has been made in the design of efficient techniques for digital encoding of analog audiovisual data [Jay93]. When distributed across the network, the quality of a reconstructed signal at the receiving side depends on the encoding scheme and the distortions introduced by network imperfections. In most applications, this quality is ultimately judged by a human receiver, and hence subjective metrics linked to human perception factors are used.

Given a fixed packetization interval, the encoding scheme determines the actual number of bits per packet. The PCM encoding scheme samples every 125 μs with 8 bits per sample to yield a 64 Kbit/s channel. Bandwidth reduction can be achieved through the use of fewer bits per sample, less frequent sampling, suppression of transmission during silence periods, and compression of the digitized data. Adaptive Differential Pulse Code Modulation (ADPCM), for example, encodes only the difference between consecutive samples, reducing the number of bits per sample to 2-5 bits. Coding techniques with even lower bit rates, e.g., Linear Predictive Coding (LPC), exist, though speech fidelity is frequently poor [Jay93].

Performance criteria for encoding schemes include efficiency, sampling rate, complexity, and processing time [Jay93]. These performance criterions and other related issues about encoding schemas are explained in Section 2.4.1.

1.3.2 Delay

In an interactive continuous voice session, human perception factors produce a requirement for bounded round-trip delays. If round-trip delays are too long, the interactive nature of the session is degraded. Quantifying this quality factor is difficult since individual human users may have different tolerances for delay, and these tolerances will vary with the application. This phenomenon has been studied extensively for participants in interactive phone conversations [Kle67], and these results provide information on the delays acceptable in other interactive applications. Table 1.1 gives a summary of the effects of one-way delay on speech quality [RaR92]. In general, high-quality voice applications require less than 200 ms round-trip delays, but delays of up to 600 ms have been shown to be acceptable [Kle67]. Recent guidelines from CCITT suggest that even round-trip delays of up to 800 ms have a limited impact on quality [G.114].

Table 1.1 Summary of the effects of delay on speech quality

One-way Delay	**Effect on Speech Quality**
>600 ms	Conversation becomes incoherent and unintelligible.
600 ms	Speech is barely coherent.
250 ms	Annoying. Conversation style has to be changed.
100 ms	Imperceptible if listener hears from network only and not off the air
50 ms	Imperceptible even if the listener is in the same room and can hear naturally off the air and from the network

1.3.3 Delay Jitter

Statistical multiplexing of packets at internal network nodes introduces variations in the network delay experienced by individual packets. As discussed, these variations are referred to as delay jitter. Delay jitter can cause discontinuities in the playback of the voice stream at the receiver. These discontinuities are referred as gaps in this effort (Figure 1.3).

Current packet-switched networks generally do not provide jitter control in the network. Current protocols typically deal with delay jitter through buffering at the receiving site. When the first packet in a continuous voice stream arrives at the receiver, instead of beginning playback immediately, playback is delayed for the control time. Packets arriving from the network wait in a buffer, and this buffer provides protection from network delay variations.

1.3.4 Packet Losses

IP networks do not guarantee delivery of packets. Due to the strict delay requirements of real-time interactive voice applications, reliable transport protocols such as TCP cannot be used. Packet loss occurs due to bit errors and resource contention. The network transmission media is susceptible to random bit errors. In most networks, when a packet is corrupted in transmission, it is subsequently discarded by the data link layer protocol at the next receiving side. In fiber networks, random bit errors are rare, but hardware buffers and switches can lose packets during periods of high load and temporary periods of overload in the network. The impact of individual packet losses on the quality of a continuous voice stream is variable since, in general, all bits in the encoded stream are not equally important. Signal processing techniques can significantly improve loss tolerances, but even the loss of a single packet may noticeably degrade playback quality at the receiver. In any case, the tolerance to packet loss is low, and packet losses greater than 10% are generally not tolerable [Rya98].

1.4 Problem

A number of technical problems exist in sending voice packets in a network. One of the most significant technical problems is the reconstruction of a continuous stream of voice from a set of packets that arrive with varying transit delay. Quality of service (QoS) of such a packet voice system can be measured by the probability of no gap at the receiver and the end-to-end packet delay. Encoding, packet loss, delay and variation in delay are the factors affecting quality of service. There are trade-offs among these factors that affects the quality of service. For instance, the selection of control time represents a trade-off between compensation of delay jitter in the network and the end-to-end delay constraint of voice conversation.

In this effort, trade-offs among factors that affect QoS is researched. The impact of these parameters on packet voice communication quality is determined, and optimum combinations of these factors are explored for various network conditions.

1.5 Scope

In order to determine the trade-off results among quality of service parameters and explore the optimum combination(s) of these parameters, packet voice communication is simulated on BONeS DESIGNER network simulation software. Experimental measurements are used to simulate different IP network condition characteristics, such as network delay, loss and jitter. A PC-to-PC implementation configuration (Section 2.5.1) is used for experimental measurements and simulation modeling. Experimental network measurements are taken from the Air Force Institute of Technology (AFIT) to other U.S. Air Force bases over the current Internet connection in order to provide real-word sampling of network characteristics for short and long-haul networks. Simulation models are executed for the network characteristics obtained by sample measurements and for various control time values. Simulation results are analyzed to explore the trade-offs among these factors.

1.6 Document Organization

The remainder of the thesis is organized as follows. Chapter 2 gives background information in detail and surveys the literature on the transmission of real-time voice over IP networks. Chapter 3 presents experimental network measurements and the end-to-end packet voice transmission model used for simulations. Chapter 4 reports the results of the simulations and discusses the implications of factors on quality of packet voice communication. Chapter 5, finally, summarizes results and explores the trade-offs among these factors for packet-based transport of continuous voice.

2. Literature Review

2.1 Introduction

This chapter reviews reference materials for Voice over Internet Protocol (VoIP). Section 2.2 describes two different approaches to communication networks. Section 2.3 describes the Internet Protocol (IP) and related protocols, such as the Transmission Control Protocol (TCP), and User Datagram Protocol (UDP), used in VoIP since this effort concerns transmission of voice over IP networks. Important aspects of these protocols with respect to transmission of voice over IP networks are also discussed in this section. The Internet Control Message Protocol (ICMP) and its functions, which are used for experimental measurements are also described in this section. Section 2.4 gives in detail background information about encoding and delay factors. Configuration types for implementing voice over IP networks are given in Section 2.5. Finally, Section 2.6 discusses significant problems, related works and techniques of this area.

2.2 Communication Networks

Whether communication networks provide connections between one computer and another computer or between terminals and computers, they can be divided into two fundamental types: circuit-switched (sometimes called connection-oriented) and packet-switched (sometimes called connectionless). Circuit-switched networks work by forming a dedicated connection (circuit) between two points. The traditional telephone system is an example of using circuit-switching technology. A telephone call establishes a circuit from the originating phone to the destination phone through the switching offices and trunk lines. The advantage of circuit switching is its guaranteed capacity: once a circuit is established, other network activities do not decrease the capacity of the circuit. The major disadvantage of circuit switching is cost: circuit costs are fixed, independent of traffic. For example, someone pays a fixed rate for a phone call, even when the two parties do not talk.

Packet-switched networks are typically used to connect computers. These networks take a completely different approach compared to circuit-switching networks. They are also known as connectionless communication networks. Connectionless means that no connection between the source and the destination is established prior to data transmission. In a packet-switched network, data to be transferred across a network is divided into small pieces called packets that are multiplexed onto high capacity inter-machine connections. A packet carries information that enables the network hardware to know how to send it to the final destination. The major advantage of packet switching is that multiple communications among computers can proceed concurrently. All pairs of machines that are communicating share inter-machine connections. The disadvantage is that as network activity increases, a given pair of communicating computers receives less of the network capacity. If the workload of the packet-switched network increases, computers using the network must wait before they can send additional packets.

Although packet-switched communication networks are not able to guarantee network capacity, they have become extremely popular. Because multiple machines can share the network hardware, fewer connections are required and cost is kept low.

2.3 The TCP/IP Family

The Transmission Control Protocol/Internet Protocol (TCP/IP) represents a family of protocols that evolved over a period of time to perform predefined tasks. The TCP/IP family includes the Internet Protocol (IP) as a network protocol. There are two transport protocols in the TCP/IP protocol family: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). Some applications, such as the File Transfer Protocol (FTP), the Telnet, the Simple Mail Transport Protocol (SMTP), and the HyperText Transport Protocol (HTTP), were developed to use TCP, while other applications, such as the Domain Name Service (DNS), and the Real-Time Transport Protocol (RTP), were developed to use UDP. A portion of the TCP/IP protocol family is given in Figure 2.1.

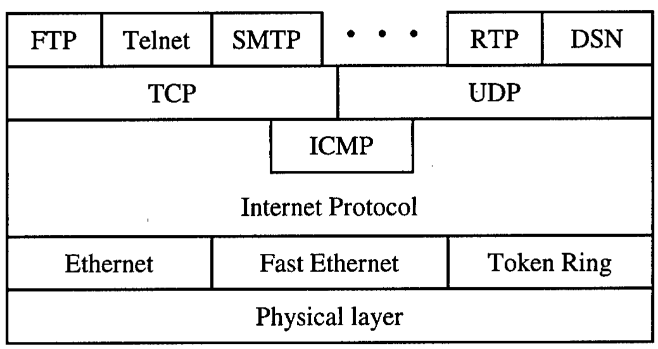


Figure 2.1 Partition of the TCP/IP Protocol Family

The roots of TCP/IP can be traced to the U.S. Defense Department Advanced Research Projects Agency (DARPA), which developed a series of communications protocols for transporting data between geographically separated networks via a common network infrastructure known as the Advanced Research Projects Agency Network (ARPANET)[Gil98].

2.3.1. Internet Protocol (IP) (v4)

The Internet Protocol (IP) was developed to provide the functions necessary to deliver a package of bits (an internet datagram) from a source to a destination over a interconnected system of networks. IP is primarily concerned with delivery of the datagram. However, IP is not concerned with some other issues such as the end-to-end reliable delivery of data or the sequential delivery of data. IP leaves those issues for the host-to-host layer and the implementation of the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) that reside there.

A datagram is a package of data transmitted over a connectionless network. Datagram transmission is similar to mailing a letter. With both a letter and a datagram, you write a source and destination address on the envelope, place the information inside, and drop it into a mailbox for pickup. At that point you've turned it over to the post office for delivery, and you trust that it will be delivered to the address on the front if at all possible. The Internet works the same way, reading the address on the outside of the packet (there's no need for it to read the data inside), and forwarding it along the way until it finally reaches the appropriate network node.

2.3.2. Transmission Control Protocol (TCP)

TCP was developed to provide a reliable flow of data between two hosts and is responsible for verifying the correct delivery of data from sender to the receiver. TCP also allows a process on one end system to reliably send a stream of data to a process on another end system. It is connection-oriented; before transmitting data, participants must establish a bi-directional connection. Data can be lost in the intermediate networks, but

TCP adds support to detect lost data and to trigger retransmission until the data is correctly and completely received.

The request to retransmit and the retransmission process delay the data flow of the packet through a TCP/IP network. Thus, although TCP is used by the FTP, Telnet, the SMTP, and other applications where the integrity of data is of primary concern, it can result in unacceptable delays when transporting digitized voice. Therefore, it is rarely, if used at all, for packet voice transmission.

2.3.3. The User Datagram Protocol (UDP)

In the TCP/IP Protocol family, the User Protocol Datagram Protocol (UDP) provides an unreliable, connectionless transport service. The UDP also provides the mechanism that application programs use to send datagrams to other application programs. UDP provides protocol ports used to distinguish among multiple programs being executed on a single machine. That is, in addition to the data sent, each UDP message contains both a destination port number and a source port number. This makes it possible to deliver the message to the correct recipient for the UDP software at the destination recipient and for the recipient to send a reply. UDP uses the underlying Internet Protocol (IP) to transport a message from one machine to another, and provides the same unreliable, connectionless datagram delivery service as IP. It does not use acknowledgements to make sure the messages arrive, it does not order incoming messages, and it does not provide feedback to control the rate at which information flows between the machines. Thus, UDP messages can be lost, duplicated, or arrive out of order. As a summary:

The User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines. It uses IP to carry messages, but adds the ability to distinguish among multiple destinations within a given host computer [Com95].

An application program that uses UDP accepts full responsibility for handling the problem of reliability, including duplication, delay, out-of-order delivery, message loss, and loss of connectivity. Currently, most implementations of voice over IP networks use TCP to establish communications between network devices, while UDP is used for the flow of digitized voice.

2.3.4. Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP), documented in RFC 792, is a required protocol tightly integrated with IP. ICMP messages, delivered in IP packets, are used for out-of-band messages related to network operation or misoperation. Since ICMP uses IP, ICMP packet delivery is unreliable, so hosts can't count on receiving ICMP packets for any network problem.

One of the ICMP's services used in this effort is an *echo* function. ICMP supports an echo function, which just sends a packet on a round-trip between two hosts. *Ping*, a common network management tool, is based on this feature. Ping transmits a series of packets, measures average round-trip times, and computes loss percentages.

2.3.4.1. Ping

Ping is one of the most useful network utility tools available. It takes its name from a submarine sonar search : you send a short sound burst and listen for an echo - a *ping* - coming back [Ste90].

In an IP network, ping sends a single packet and listens for a single packet in reply. This tests the most basic function of an IP network (delivery of a single packet). Ping is implemented using the required ICMP echo function. Normally, all hosts should implement ICMP echo function (RFC 792). Of course, administrators can disable ping messages because of security or some other considerations. However, ping service is usually used to measure round-trip delay. The information that can or can't be obtained by using ping is as follows.

What Ping can tell you:

- Ping places a unique sequence number on each packet it transmits and reports which sequence numbers it receives back. Thus, you can determine if packets have been dropped, duplicated, or reordered.
- Ping checksums each packet it exchanges. You can then detect some forms of damaged packets.
- Ping places a timestamp in each packet, which is echoed back and can easily be used to compute how long each packet exchange took.
- Ping reports other ICMP messages that might otherwise get buried in the system software. It reports, for example, if a router is declaring the target host unreachable.

What Ping can't tell you:

- Some routers may silently discard undeliverable packets. Others may believe a packet has been transmitted successfully when it has not been (This is especially common over Ethernet, which does not provide link-layer acknowledgments). Therefore, ping may not always provide reasons why packets go unanswered.
- Ping can't tell you why a packet was damaged, delayed, or duplicated. It can't tell you where this happened either.
- Ping can't give you a blow-by-blow description of every host that handled the packet and everything that happened at every step of the way. It is an unfortunate fact that no software can reliably provide this information for a TCP/IP network [Ste90].

2.4 Background

This section provides detailed information about encoding algorithms, which are used in packet voice transmission, and end-to-end (speaker-to-speaker) delay of the system.

2.4.1 Encoding

Human speech, and in fact everything we hear, is naturally in analog form, and early telephone systems were likewise. Analog signals are often depicted as smooth "sine waves," but voice and other signals contain many frequencies and have more complex structures. While humans are well equipped for analog communications, analog transmission is not particularly efficient. When analog signals become weak because of transmission loss, it's hard to separate the complex analog structure from the structure of random transmission noise. Amplifying analog signals also amplifies noise, and eventually connections became too noisy to use. Digital signals, having only "one-bit" and "zero-bit" states, are more easily separated from noise and can be amplified without corruption. Over time, it became obvious that digital coding was more immune to noise corruption on long-distance connections, and the world's communications systems converted to a digital transmission format.

In traditional telephony applications, a digital transmission format, PCM or ADPCM, is used on synchronous digital channels, which means that there is a constant stream of bits generated at the specified rate, whether there is conversation or not. There are, in fact, hundreds of brief silent periods in the average call, and each of them waste bandwidth and money. On standard telephone connections, there is no alternative to this waste.

Encoding is an alternative if packet voice transport is used. In packet voice applications, speech is transported as data packets, and these packets are generated only when there is actual speech to transport.

Performance criteria for encoding schemes include efficiency, sampling rate, complexity, and processing time [Jay93]. Efficiency is measured as the number of bits per sample to represent the signal. Common encodings for an 8 kHz voice, for instance, include the PCM encoding at 8 bits/sample and ADPCM encoding at 2-5 bits/sample.

The sampling rate and efficiency of an encoding scheme determine the bandwidth required for network distribution. Since low-bit-rate encoding schemes result in a less precise reconstruction of the original analog signal, the selection of an encoding scheme represents a trade-off between consumption of bandwidth on the network and playback quality at the receiving side. For voice, a common technique for bandwidth reduction without a loss in quality is the suppression of transmission during silence periods between speech activity periods.

Other dimensions of an encoding algorithm are the complexity of the algorithm and the processing time that it requires. Complexity can be measured in terms of computing speed (millions of instructions per second-MIPS), Random Access Memory (RAM), and Read-Only Memory (ROM). Complexity determines cost; as it increases, cost goes higher. The International Telecommunications Union (ITU) has defined a series of recommendations for speech/audio coding, including the PCM and ADPCM already briefly discussed. A summary of ITU speech/audio codec recommendations is shown in Table 2.1 [MM98][Cis98] [BoG98].

Table 2.1 Summary of ITU Speech/Audio Codecs

ITU Recommendation	**Bit Rate (kbps)**	**Bandwidth (kHz)**	**Quality**	**Complexity**	**Delay (ms)**
G.711	48,56,64	3	Good	Lowest	0.75
G.726	32	3	Good	Low	1
G.723.1	5.3, 6.3	3	Good	Highest	67-97
G.728	16	3	Good	Low	3-5
G.729 (A)	8	3	Good	High	25-35

2.4.2 Delay

Delay, as another factor, may have a great impact on packet voice communication. Delivering voice packets over packet-switched networks requires some upper bound on the delay of the voice packets because of human perception factors in an interactive continuous voice session. Delay causes two basic problems: echo and talker overlap. Echo is caused by the signal reflections of the speaker's voice from the far end telephone equipment back into the speaker's ear. Echo becomes a significant problem when the round-trip delay becomes greater than 50 ms [Rya98]. Since echo is perceived as a significant quality problem, voice over packet systems must address the need for echo control and implement some means of echo cancellation. Talker overlap (or the problem of one talker stepping up the other talker's speech) becomes significant if the one-way delay exceeds end-to-end delay constraints.

Each piece in the voice packet flow line, from encoding at the source to playback at the receiver (Figure 1.2), adds delay to the overall transmission. Codec delay, network delay, and buffering delay are the sources of end-to-end delay in voice over a packet call. Some of these end-to-end system delay components are relatively fixed like codec delay, while others depend on network conditions.

2.4.2.1 Encoder Delay

Encoder delay consists of two parts: accumulation delay and processing delay.

Accumulation delay

This delay is caused by the need to collect a frame of voice samples to be processed by the voice coder. It is related to the type of voice coder used and varies from a single sample time (0.125 μs) to many milliseconds.

Processing delay

The actual process of encoding and collecting the encoded samples into a packet for transmission over the packet network causes this delay. The encoding delay is a function of both the processor execution time and the type of algorithm used. Often, multiple voice coder frames will be collected in a single packet to reduce the packet network overhead.

The effective one-way latency of the encoder is the sum of the accumulation and processing delay. Table 2.1 also shows one-way latency of some of ITU codec recommendations.

2.4.2.2 Network Delay

This delay is caused by the physical medium and protocols used to transmit the voice data. Network delay is a function of the capacity of the links in the network and the processing that occurs as the packet transits the network.

The network delay includes access delay, transmission delay and transit delay as shown in Figure 2.2.

1. Access delay, the time necessary at the source in waiting for the medium to be available or for the network to be ready to accept the block of information.

2. Transmission delay, the time necessary to transmit the sequence of bits of block once the medium or network readies.

3. Transit delay, the time it takes between emission of the first bit of a data block by the transmitting end-system and its reception by the receiving end-system. This, delay for a block of information to transit a network is also called the network latency. The network can't avoid transit delays because of the propagation time of the electrical and optical signals.

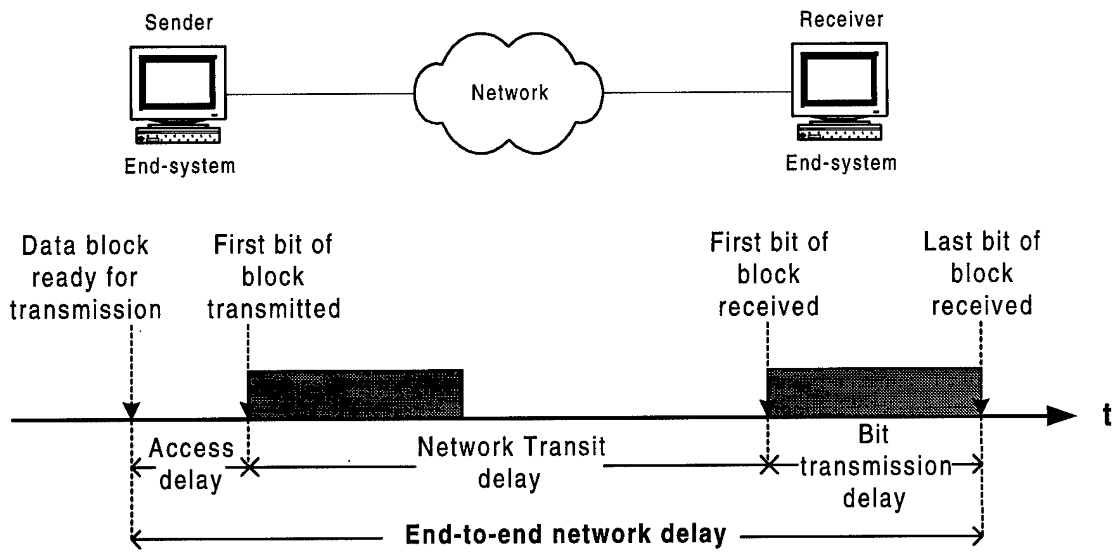


Figure 2.2 End-to-End Network Delay

Another delay definition type is the round trip delay shown in Figure 2.3. Round trip delay is the time between emission of the first bit of a data block and its reception by the same end-system after the block has been echoed by the destination end-system. It is not an intrinsic characteristic of the communications subnetwork, but its definition permits a simple way of estimating the one-way delay since the source and the destination is the same machine.

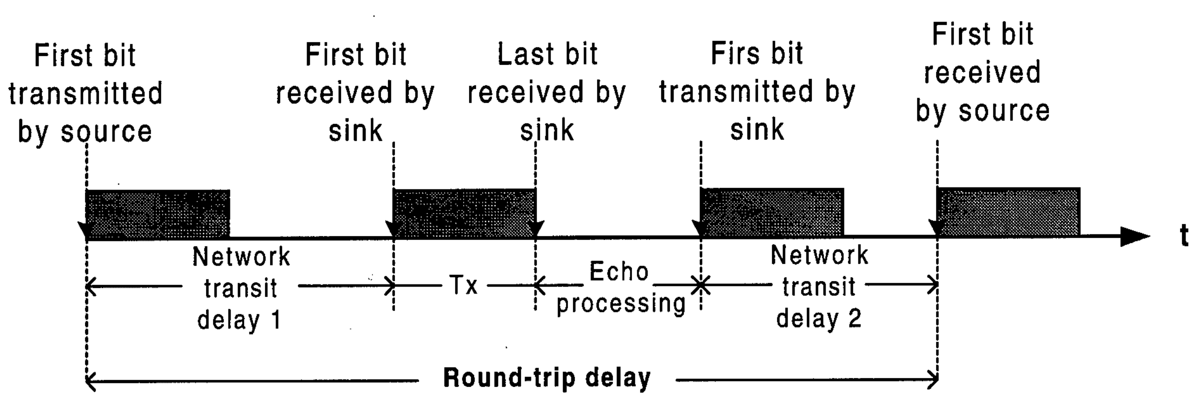


Figure 2. 3 Round-trip Delay

2.4.2.3 Buffering Delay

This delay is caused by the buffers used to remove packet jitter on the receiving side. Due to delay jitter, a packet may arrive after or before its playback time. Therefore, the packet is placed in a queue at the packet voice receiver until it is due for playback. This delay can be a significant part of the overall delay if packet delay variations are high in the IP network.

2.5 Configurations

There are several configurations for implementing voice over IP networks according to type of the sender and the receiver equipment and their connections. These configurations are explained below.

2.5.1 PC-to-PC

In a PC-to-PC configuration, shown in Figure 2.4, end point computers can be on a local area network (LAN), as in the case of many corporate computers, or connected to the network (i.e., Internet) via telephone lines. Analog voice signals are digitized, compressed, and packetized by using soundcards and software at the sender's PC.

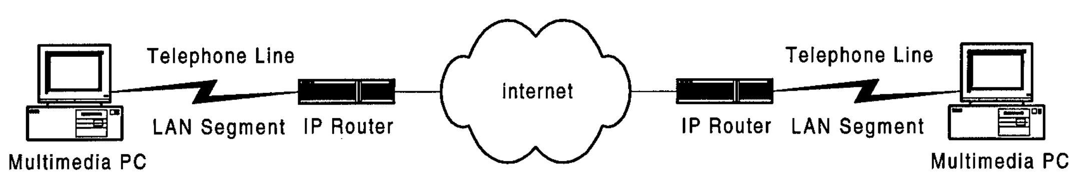


Figure 2.4 PC-to-PC Configuration

The digital voice information then passes through the sender's modem or network segment and is sent to the IP router where the voice packets are routed to the destination. Playbacks of received packets occur through a soundcard on the receiver PC. A codec algorithm also could be implemented in the hardware, perhaps as part of a modem, network interface card, or soundboard.

2.5.2 Phone-to-Phone

In this configuration, Telephony Gateway (TG) will allow users at both ends of the conversations to make calls using a regular phone as shown in Figure 2.5. At the sending end, when the sender connects to his local TG via the normal circuit switching telephony line, the TG digitizes and compresses the analog voice signals into packets for travelling over the interconnected networks to the destination's gateway. At the destination end, for packets coming in from the network, TG decompresses and then converts the digitized voices back to analog signals which then go through the standard Public Switched Telephony Network (PSTN) to the receiver's phone. The TG works as a connector between a telephone system PBX switch and a data network server.

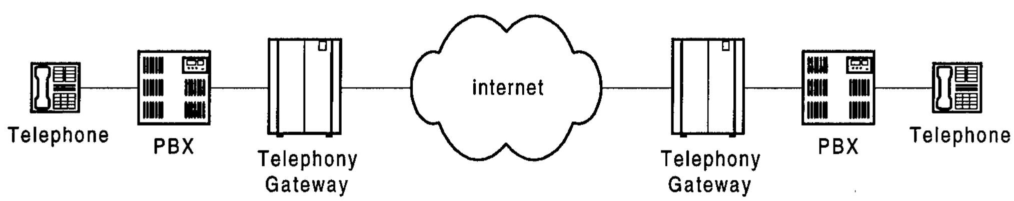


Figure 2.5 Phone-to-Phone Configuration

2.5.3 PC-to-Phone or Phone-to-PC

This configuration uses one TG on either end of the calling parts to place calls using a regular telephone. At the phone user's end, the TG is used to contact the destination PC through the interconnected networks. At the PC user's end, a modem is used to convert the digital packet signals into analog audio signals or vice versa. In this scheme, a telephone user can dial directly to the other part's computer when the other part is logged on to the network. Meanwhile, the PC user can also initiate a call to the telephone user. The functionality of the TG is completely transparent to the users. PC-to-Phone or Phone-to-PC configuration is shown in Figure 2.6.

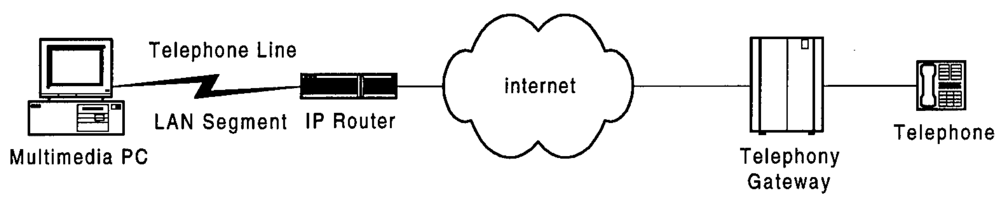


Figure 2.6 PC-to-Phone/Phone-to-PC configuration

2.6. Problems and Related Works

Besides the technical and economic motivations, a number of technical problems exist in sending voice over packet-switched networks. One of the significant technical problems is voice synchronization. This significant issue involves how to design an appropriate voice construction strategy that reproduces acceptable quality speech from packets that arrive with varying network delay and may arrive out of order.

Packets are produced at the packet voice sender and sent through the network. As the packets pass through the packet network, each can encounter a varying amount of delay. The variation in delay depends on the nature of packet network, the traffic on network, and the speed of network facilities. For a local area network, variable delay is typically small. For a long-haul network, variable delay can be significantly larger. The packet voice synchronization problem is, thus, generally more significant in a long-haul network than a local area network [Mon83].

As the packets arrive at the packet voice receiver, they are reconstructed into a continuous stream of voice samples and delivered to the destination customer. Typically, this reconstruction is done by choosing a target playback time for each incoming packet as a fixed interval after the packet is introduced. This fixed interval includes the estimated network delay plus control time. This is required to ensure that the entire system introduces a fixed delay into the speech path, and that continuous speech can be reconstructed without varying delay. Each packet that arrives before its playback time is placed in the proper sequence in a queue of packets from which the speech is reconstructed. If a packet does not arrive before its playout time, the packet will be effectively lost.

Once a control time is chosen for a packet voice call, a mechanism is needed to determine the playback time for each incoming packet. To do so, the packet voice receiver must determine the production time for each incoming packet, or equivalently, the actual delay experienced by each packet. In general, the delay estimation or packet production time determination need not to be performed on every incoming packet. Once an estimate of delay has been made for one packet, relative production time of subsequent packets can be encoded in information sent in each packet, such as the sequence number.

Montgomery [Mon83] discussed several aspects of the packet voice synchronization problem and techniques that can be used to address it. These techniques estimate in some way the delay encountered by each packet and use the delay to estimate how speech is constructed. The delay estimates produced by these techniques can be used in managing the flow of information in the packet network to improve overall performance. Many methods can be used to estimate either the packet production time or transport delay of an incoming packet. Montgomery discussed four general methods described below.

- *Blind Delay*

The simplest strategy for estimating the production time of an arriving packet is to make a worst case assumption. Once the arrival time has been estimated, the packet voice receiver uses sequence information in subsequent packets to determine the proper play out time for each. This method is called blind delay, because the packet voice receiver makes its estimate blindly, with no information on the actual packet production time or transit delay. If the network delay variation is small, blind delay may be an appropriate strategy.

- *Round-trip Measurement*

Blind delay is a simple technique, but it may not be adequate in a long-haul network. A second delay estimation technique commonly used in maintaining synchronized clocks in a distributed networks is to actually measure the round-trip delay in the communication path between the packet voice sender and receiver, and assume that delay is equally distributed in both directions. Measurements can be made by sending a packet containing a local clock value from the sender. When the packet arrives at the receiver, it is immediately sent back to the sender. When it arrives back at the sender, the sender calculates the round-trip delay by subtracting the clock value in the packet from the current time as in the ICMP echo function discussed before. The round-trip delay is then sent to the receiver, and subsequent packets sent by the sender to the receiver contain timing information relative to the first packet sent by the packet voice sender.

While this technique gives an accurate measurement of round-trip delay, estimation of one-way delay may not be accurate, because the delay in both directions may not be equal. While round-trip delay estimation can't provide a completely accurate delay measurement, it does reduce error substantially over the blind delay method.

- *Absolute Timing*

The third method that can be used is to maintain clocks in the sender and receiver synchronized to the absolute time reference. In this case, each packet carries an indication of its production time and the receiver uses that to compute the target playback time. However, maintaining synchronized clocks in geographically distributed systems is, in general, a complex problem. It requires the distribution of the standard reference frequency to each local clock via a reliable channel.

- *Added Variable Delay*

The fourth method for estimating the delay experienced by a packet in a packet switched network is to actually measure the variable delay where it occurs. Variable delay results from queueing and processing delay in packet switches and packet terminals. The variable delay measurement can be made by carrying a "delay stamp" indicating the accumulated delay in each packet. Each network element adds its delay to the delay stamp as the packet passes through. The packet voice receiver uses the delay stamp in an arriving packet to determine playback time. Like absolute timing, the added variable delay method provides a good measurement of packet network delay, but it requires the support of network elements.

Several packet voice-receiving schemes are discussed by Barberis [BP80] with respect to the timing information, contained in the header part of the packet, and to the network delay estimation. The following three schemes are developed [BP80].

- *Null Timing Information (NTI) Device*

Null Timing Information means that the packet voice receiver delays every first packet by a given amount of time (T). NTI device behavior is illustrated in Figure 2.7.

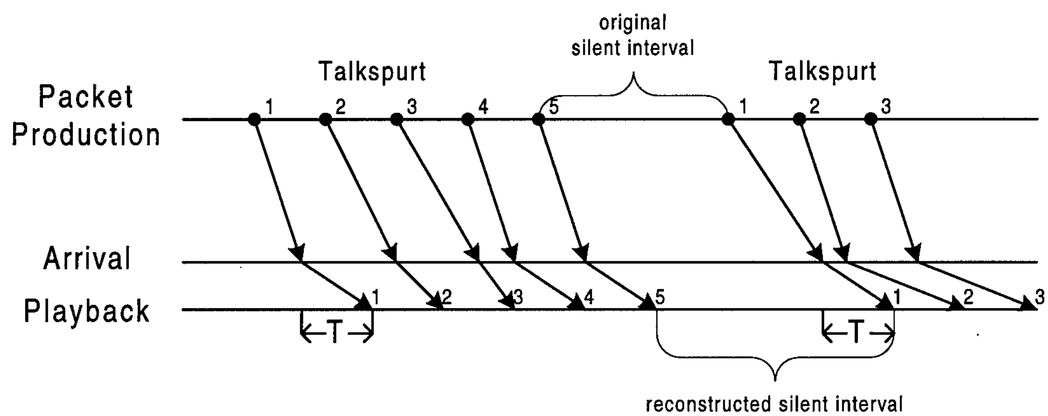


Figure 2.7 NTI Device Behavior

By this mechanism, a good cancellation of gaps introduced by the stochastic network between packets of the same talkspurt is achieved. This algorithm, however, has no means of reproducing exactly original speech pauses against their stochastic network perturbation. This happens because, without knowing exactly the talkspurt departure times and without having a good estimate of the transit time of the first packets of talkspurts, the receiver can't recover the speech pauses.

- *Incomplete Timing Information (ITI) device*

In this case, if the estimated network transit time is below a threshold control parameter, then that packet is additionally delayed by an amount equal to the threshold minus the estimated transit time. If the header of the first packet of talkspurt contains the time stamp of its departure and if the actual transit time is estimated by some algorithm, a reasonable operation to be performed on the first packet is to delay playback until its estimated delay plus control time. The ITI device behavior is illustrated in Figure 2.8.

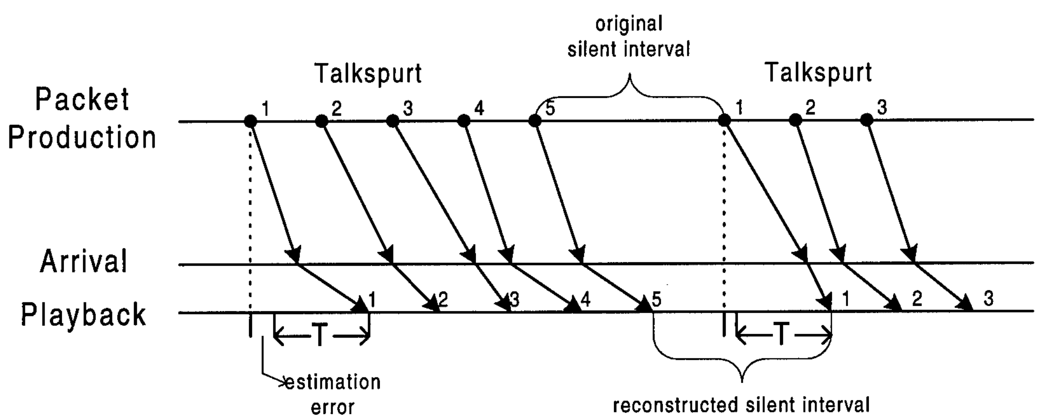


Figure 2.8 ITI Device Behavior

- *Complete Timing information (CTI) device*

This device works in the same manner as the preceding one, but in this case, the exact transit time is assumed to be known. This means that the estimation error (the offset between sending and receiving clocks) vanishes and a perfect voice reconstruction can be achieved in the CTI case only.

2.7 Summary

In the first section, different approaches to the communication networks are discussed. Next, a discussion on VoIP and measurement related protocols is provided. Important aspects of these protocols are presented in Section 2.3. Then, background information about encoding and end-to-end delay is provided in detail in Section 2.4. VoIP configurations are presented in Section 2.5. Finally, significant problems of these areas and techniques associated with them are presented in Section 2.6.

3. Methodology

3.1 Introduction

This chapter presents a methodology that is used to analyze the effects of the quality of service parameters on the packet voice transmission over IP networks. This chapter consists of two basic sections: measurement and simulation. While Section 3.2 discusses measurements in detail, issues related to the modeling and simulation are discussed in Section 3.3.

3.2 Network Measurements

Experimental network measurements were accomplished in order to provide real-world sampling of the network characteristics for short and long-haul networks. Therefore, a series of measurements were conducted on the Internet and the results of these experiments were used in the simulations. The following subsections cover the issues related to these measurements.

The difficulties in measuring network characteristics, especially delay, are discussed in Section 3.2.1. Section 3.2.2 introduces a network utility tool used to measure network characteristics. The answer to the question of how the measurement is made is given in Section 3.2.3. Results and analysis of the measurements are given in Section 3.2.4. Finally, Section 3.2.5 and Section 3.2.6 discuss verifications and validations of the experiments respectively.

3.2.1 Measurement Barriers

In order to evaluate the effect of network characteristics on real-time voice transmission over IP networks, knowledge of unidirectional delay, delay jitter, and loss rates is necessary. These metrics must be available in order to analyze the effect of QoS factors. However, accurately determining one-way packet delay from a sender host to a receiver host in the Internet is difficult due to the need for synchronization of the sender and receiver clocks. In fact, there is little knowledge about unidirectional latencies in large internetworks like the Internet or corporate intranetworks [Fas97]. In theory, unidirectional delays can be measured accurately by equipping each end point with a global positioning system (GPS) satellite transceiver, but this is an expensive solution. Therefore, one-way network delay of a packet could not be measured directly.

Measuring the jitter of one-way Internet delay has been done [BCP93] by comparing the relative difference between sender-receiver and receiver-sender delay. It was shown that the second moments of sender-receiver and receiver-sender delays are usually not symmetrical. This fact is likely due to the asymmetric end-to-end routes common in the Internet [Pax97]. Thus, it is known that measuring round-trip delays and simply dividing the results by two does not necessarily give us realistic one-way delays. Given that there is no reasonably accurate technique for measuring unidirectional delays, round-trip delays of packets are measured for a network latency metric. While these measurements cannot be used directly to infer one-way delays, they can be used to extrapolate the one-way network delay.

3.2.2 Software Tool

The Internet Protocol *ping* service is commonly used to measure round-trip delay. ICMP echo packets with timestamps are used for network latency measurements. To obtain a network traffic profile for packet voice, the *Fing* latency measurement tool [Fas97] is used in this effort. Fing has its roots in the ping facility. The latter utilizes the Internet Control Message (ICMP) to gather statistics about host reachability and about the round-trip times to a specified Internet destination.

In the standard ping service, the sending process generates one message per second and inserts its current system time into the data field before delivering it to the destination. The echo process at the receiver instantly returns the unmodified packet by generating an ICMP echo reply message. The sender then uses the sequence number and the time instant of acknowledgement reception to calculate the round-trip delay.

Numbers of enhancements were made over standard ping to support investigations on network latencies for application specific load profiles [Fas97]. Fing has a more sophisticated load generator supporting burst traffic models and Poisson departures, both with pre-definable departure rates that can be as high as a few thousands packets per second. Additionally, it comprises extensive statistical evaluation functionality to determine delay jitter, packet loss rates, number of duplicates and out-of-sequence packets. Fing optionally calculates relative frequencies, distributions and correlation functions, means with confidence intervals, and variances of the samples at the end of each trial. Also the ICMP timestamp option is integrated to lead the receiver to insert t_1 upon reception and t_2 before returning the probe packets, as depicted in Figure 3.1.

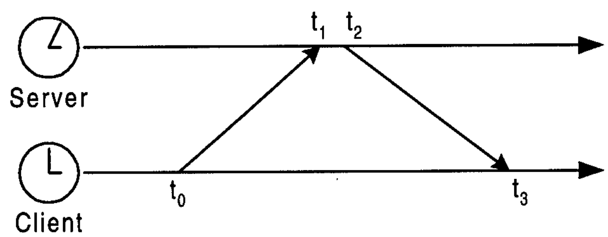


Figure 3.1 Round-trip Timestamps

3.2.3 Measurement Structure

In order to better understand the characteristics of the end-to-end behavior of paths in a network, a series of measurements were conducted on the Internet. These measurements were taken with the Fing network latency tool discussed in the previous section. Figure 3.2 shows the paths measured for network characteristics.

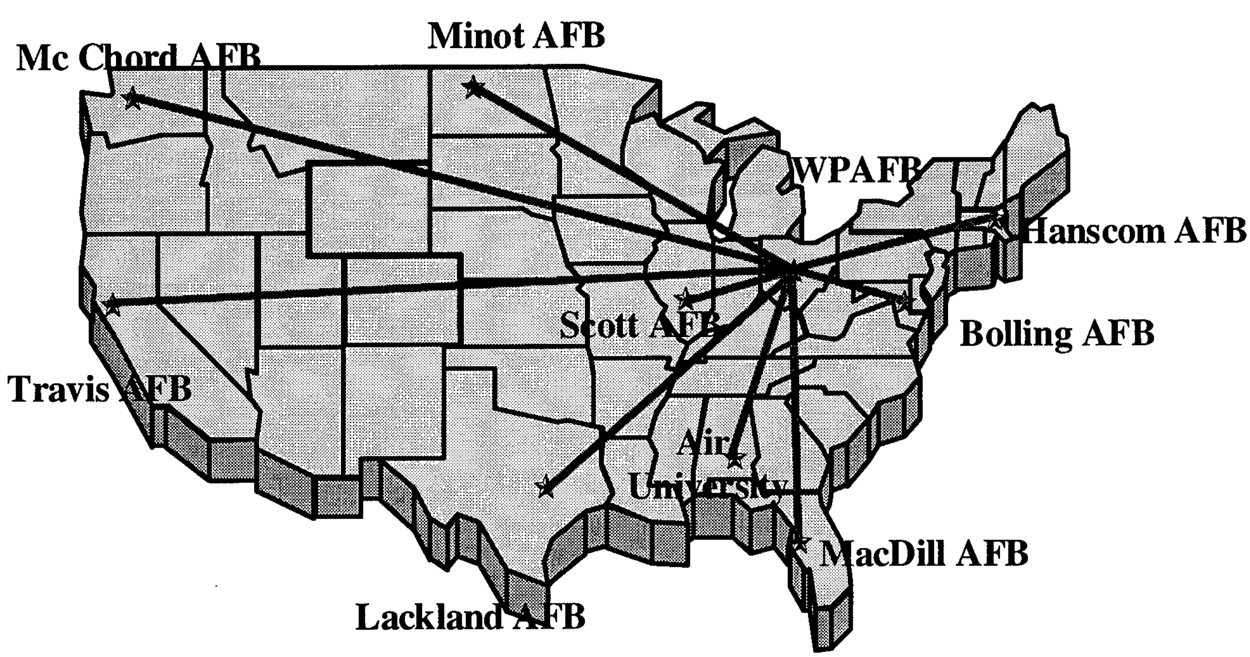


Figure 3.2 Measurement Paths

Delay and loss characteristics were measured from the Air Force Institute of Technology (AFIT) to 10 selected U.S. Air Force bases all over the country. These sites were chosen based on availability and geographical dispersion. Selected Air Force bases, destination machine IP numbers, and number of hops to arrive at the destination from AFIT are shown in Table 3.1.

Table 3. 1 Destination IP Address and Hop Counts

Path	**Base**	**IP Address**	**Hop (#)**
1	Wright Patt. AFB.	129.48.28.13	4
2	MacDill AFB.	131.24.120.30	9 (13)
3	Lackland AFB.	131.32.120.13	10 (15)
4	Minot AFB.	132.32.201.7	11
5	Travis AFB.	132.33.132.10	11
6	Hanscom AFB.	129.53.216.5	12
7	Bolling AFB.	162.24.56.211	12
8	Scott AFB.	140.175.186.49	13
9	Air University	132.60.128.15	13 (11)
10	McChord AFB.	131.30.242.35	15

Each experiment was conducted as follows. At the beginning of each hour, Fing transmitted measurement packets to the destination for three minutes. Packet size was 64 bytes, and interdeparture times were 25 ms apart. Thus, 7200 packets were transmitted during a 3-minute transmission period. After the last packets were sent, echo packets were awaited for one minute. Echo packets that had not been received at the end of this one-minute waiting period were assumed lost, and the measurement session was ended. Then this measurement session was repeated in sequence for all other paths. At the beginning of the next hour, this procedure was repeated.

The measurements produced 14 session/path per day, from 7 AM EST (beginning of working hours at the East Coast) to 8 PM EST (end of the working hours at the West

Coast). The collection period started on December 14, 1998 and continued until December 19. Thus, each path was measured for 84 sessions each having 7200 transmitted packets.

In order to observe the conditions under a typical real-time voice application run, measurements were taken during the daytime. Besides, in order to see end-to-end behavior differences during out of standard working hours (8 AM. to 5 PM) and weekends, measurements were also taken during those periods. All experiments were conducted by the fing network latency tool working on the *nimrod* workstation located on AFITNET under the 129.92.20.8 IP address. Nimrod is a SPARCstation II model Sun workstation running Unix v.4.1.3_U1.

Hop counts, number of hops to reach the destination, are measured by CyberKit® (v 2.4) software that uses winsock 2.0. The hop counts given in Table 3.1 are based on several trials taken on different days during the experiments. While most of these trials gave the same results, some others rarely produced different results. Those different results are given parenthetically in Table 3.1.

3.2.4 Results

The results obtained by experimental measurements were analyzed according to round-trip delay, delay standard deviation, and packet loss rate for all of the 10 paths. The outcomes are given in following subsections.

3.2.4.1 Round-trip Delay

The mean values of the packet round-trip delays are shown for each path in Figure 3.3. If the relationship between the round-trip delays and the hop counts is observed, in general, the delay values of the paths with larger hop counts have larger network delays. However this relation is not directly proportional and has exceptions like AFIT-Hanscom AFB path.

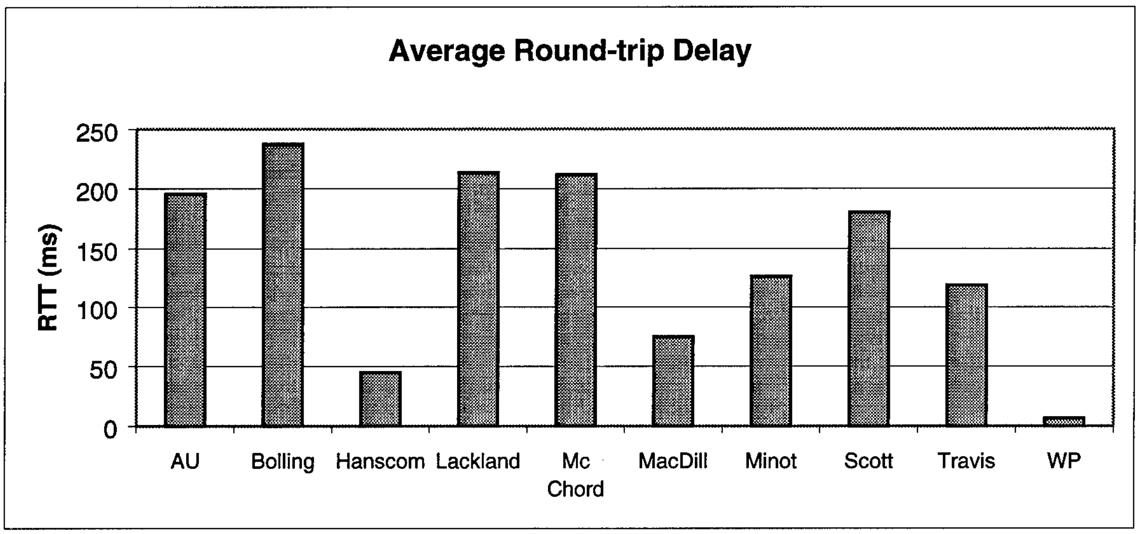


Figure 3.3 Average Round-trip Delay for Each Path

In terms of the mean round-trip delay, each path is in the range of the delay constraints for the packet voice communication. But other end-to-end system delays like buffer delay, and distribution of these delays also had to be considered. For this reason, delay distribution of each path was also examined. In order to find the distribution that best describes the characteristics of the sample measurements, Crystal Ball® software was used. The supported distribution types, given in Appendix A, were ranked according to the chi-square goodness-of-fit test results and the highest-ranking fit is chosen to represent characteristics of the sample measurements. The test results are discussed in Section 4.2. Appendix B gives the assumed probability functions for each path.

Figure 3.4 shows, as a sample, the mean round-trip delay of each session for the AFIT-Bolling AFB path during 6 days. This figure and all the other paths' behaviors, given in Appendix C, show that Internet latency follows daily business life. While delay values are smaller at the beginning of each day, they become greater at the middle of the business day and become smaller again at the end of the working hours. Delay values of the weekend day (day 6) and even Friday afternoons are also small and free from significant changes. The zero delay values given in the figures show that the measurement could not be made because of technical problems or getting no echo packets for the session.

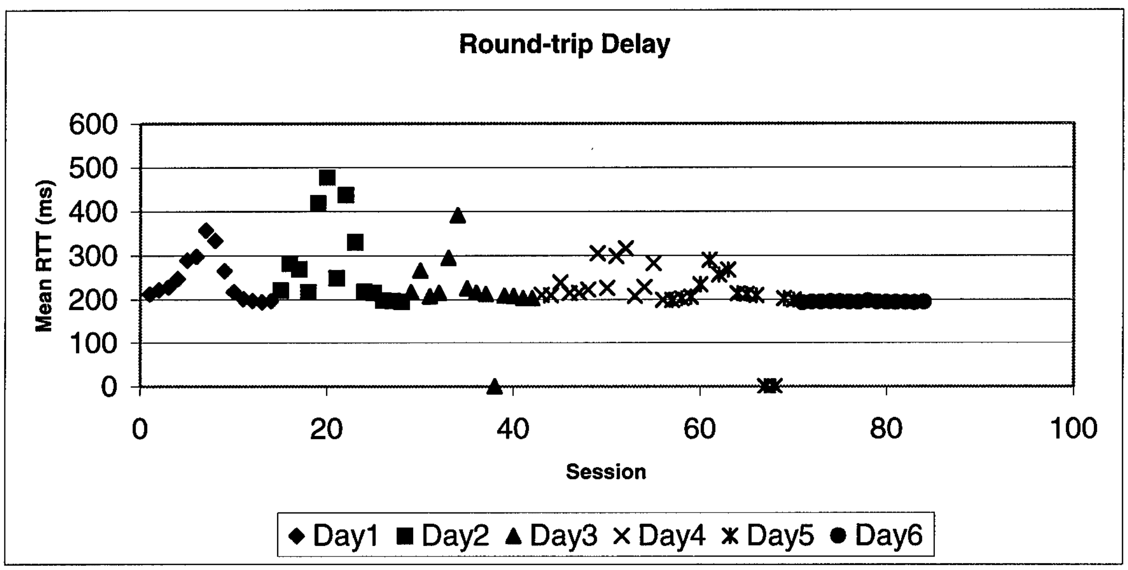


Figure 3.4 Mean Round-trip Delay of Each Session for AFIT-Bolling AFB Path

3.2.4.2 Delay Standard Deviation

The standard deviation in the round-trip delays is obtained for each measurement session. The average of these standard deviation values is given in Figure 3.5 for each path.

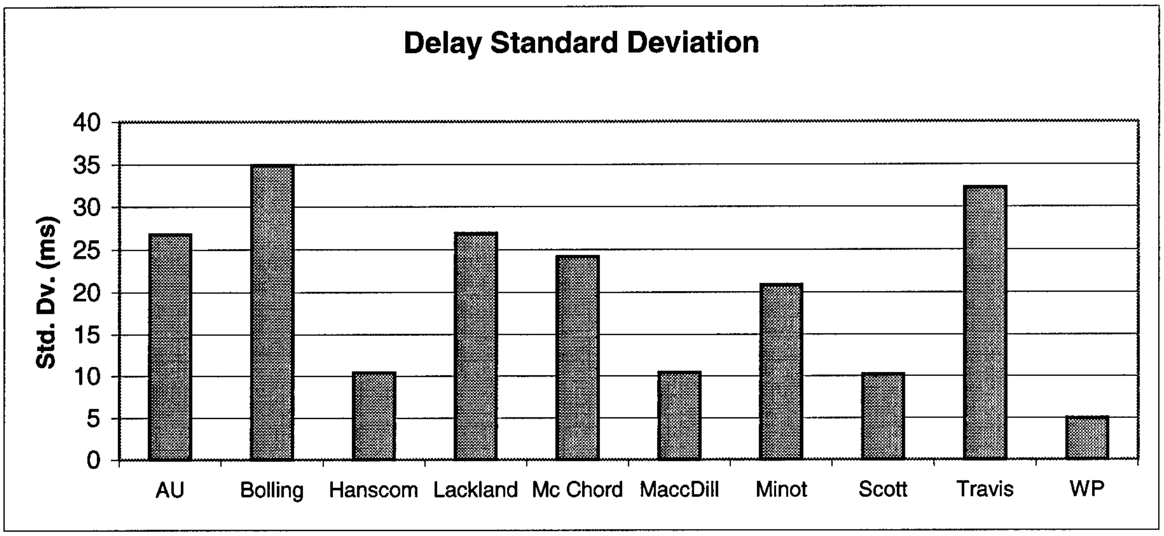


Figure 3.5 Average Delay Standard Deviation for Each Path

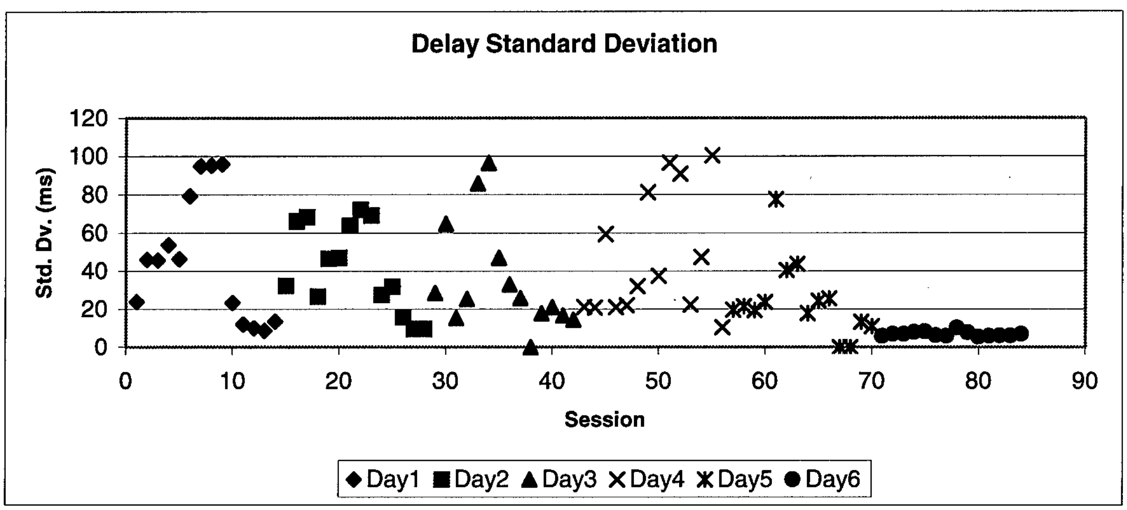


Figure 3.6 Delay Standard Deviation of Each Session for AFIT-Bolling AFB Path

A sample of these measurements is seen in Figure 3.6, which represents the mean delay standard deviation of each session for AFIT-Bolling AFB path. The delay standard deviations of each session for the other paths are given in Appendix C.

A positive correlation between mean delay and standard deviation of delay is seen. During hours of higher delay, greater standard deviations are evident.

3.2.4.3 Packet Loss Rate

The average packet loss rate of each path is given in Figure 3.7. No correlation between packet loss rate, hop count, and path delay is seen according to the average values based on sample measurement results.

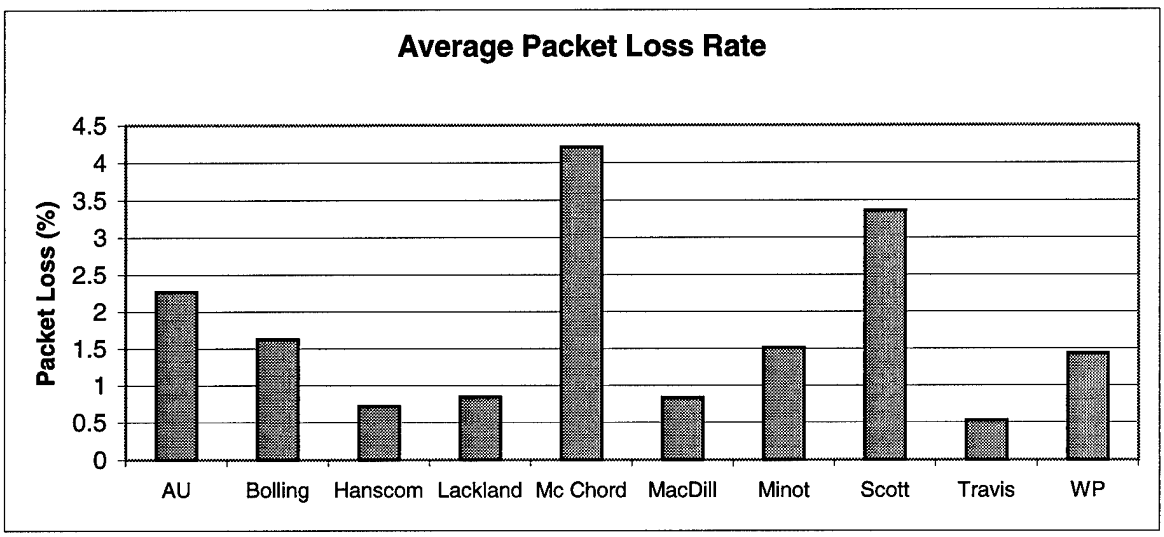


Figure 3.7 Average Packet Loss Rate for Each Path

However, a positive correlation between packet loss rate and mean delay in the sessions can be seen. As shown in Figure 3.8, which represents the packet loss rate of each session for the AFIT-Bolling AFB path, in the sessions of higher delay, greater packet loss rates can be seen. The packet loss rates of each session for all paths are given in Appendix C.

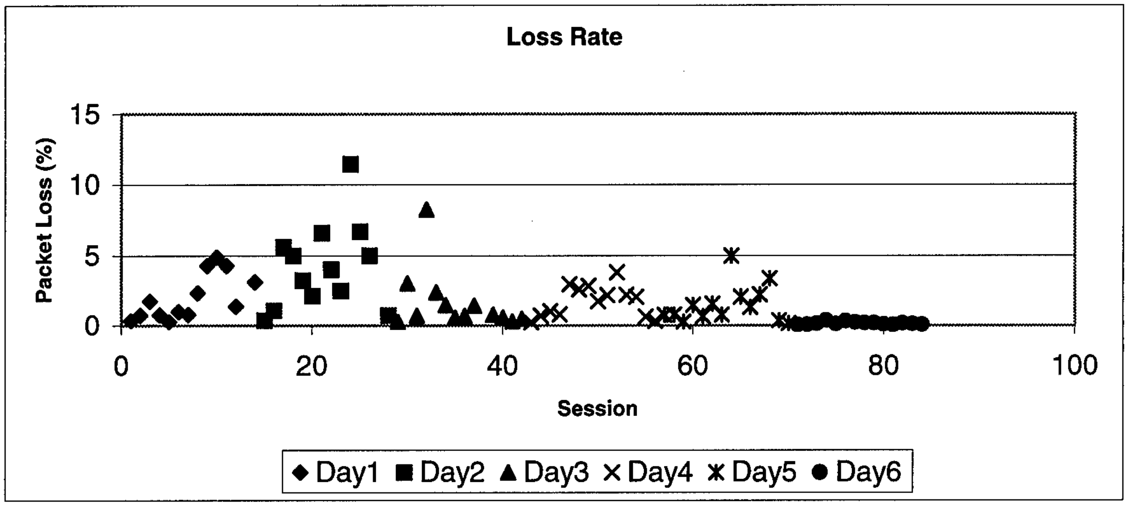


Figure 3.8 Packet Loss Rate of Each Session for AFIT-Bolling AFB Path

3.2.5 Verification

Verification is concerned with getting the measurement results right. Verification of the measurements was accomplished by tests and by closely examining the ICMP packet timestamps.

Since these experiments involve measurements of time with resolutions in the millisecond range, it is important that the clocks at the source and the destination are synchronized. With source and destination processes running on the same host, clock synchronization is not of concern for round-trip delay measurements.

Scheduling and timing mechanisms in Unix may interfere with the ability of processes to take an accurate timestamp and send the packet at an exact time. Workload of the machine plays a big role here. The Nimrod workstation is located in a room separate from the public area in the ENG laboratory of AFIT. Therefore, it is not used, in general, unless special requirements are needed like Unix v.4.1.3. During all the experiments, no others used the system. In order to verify arithmetic calculations of the fing such as mean and variance, results were also checked by other software programs like MS Excel.

3.2.6 Validation

Validation is concerned with making the right measurements. Validation can be achieved by comparing the structure and operation of real-world voice packet transmission with those of the measurements.

Since there is no reasonable method of measuring one-way packet transit delay, round-trip transit delay was measured. Although round-trip delay is not an intrinsic characteristic of the communications networks, its definition permits a simple way of estimating one-way delay. Therefore, round-trip time was used to characterize the network delay not only in this effort but also in previous research [AgG93][BoG98].

As discussed in Section 2.3.3, packet voice applications use UDP for packet transmission because of end-to-end delay constraints. UDP uses the underlying Internet Protocol and provides an unreliable, connectionless, transport service. The fing utility tool is a sophisticated version of the ping, which utilizes ICMP. Since ICMP uses IP, ICMP packet delivery is also unreliable and connectionless as in UDP packet delivery. Datagrams carrying ICMP messages are routed exactly like datagrams carrying voice data; there is no additional reliability or priority [Com95]. Besides, fing also can provide a network traffic profile as in real-world packet voice communication applications.

VoIP applications over the Internet or intranets are getting more wide spread day by day. The network measurements provide real-world sampling of the network characteristics. The procedure for analyzing input data from sample measurements consists of three steps [BaC96]:

1. Identifying the appropriate probability distribution
2. Estimating the parameters of the hypothesized distribution
3. Validating the assumed statistical model by a goodness-of-fitness test, such as the chi-square or Anderson-Darling test.

The use of goodness-of-fitness tests is an important part of validating the assumptions. These steps were performed for the delay distribution of each path by using the Crystal Ball software.

3.3 Modeling and Simulation

Modeling and simulation issues are presented in this section. First, a brief description of the BONeS Designer Network Simulator (version 3.16) and its application to computer network modeling is provided in Section 3.3.1. Section 3.3.2 defines the performance metrics. As determined from the literature review, probability of no gap and end-to-end packet delay are excellent measurements to evaluate the effect of QoS factors on the packet voice communication quality. In Section 3.3.3, there is a discussion of the relevant simulation assumptions. Then, the construction of the packet voice system and its parameters are given in Section 3.3.4 and Section 3.3.5 respectively. Section 3.3.6 discusses the simulation run time. Section 3.3.7 discusses the steps taken to verify the network model construction and the output results. Finally, Section 3.3.8 highlights the validation techniques.

3.3.1 Computer Network Modeling and BONeS Designer

The world of telecommunication and computer networks has experienced an unexpected evolution in the offered service areas, transmitted traffic types, used technologies, and universality of users. The technological changes in the computing, switching and transmission, and the integration of digital services introduced a new set of techniques and methodologies such as multimedia. These new techniques and methodologies, due to changes experienced in the world of networking, have started a parallel evolution in the analysis, design, and management of networks. The new target techniques and methodologies had in common the use of the computer as the fundamental working tool. The combined effect of increased complexity of systems and the availability of computers initiated the creation of computer-based modeling and simulation techniques in the design projects. Presently, modeling and simulation are among the most widely used techniques in the design of complex systems due to their capacity, versatility, and efficiency. The use of simulation makes feasible the study of systems and processes impossible to analyze using traditional methods. Three major types of software products are used for simulating communication networks: general-purpose simulation languages, communications-oriented simulation languages, and communications-oriented simulators [LaM94]. Years ago, general-purpose programming languages such as FORTRAN or Pascal and simulation languages were used to model discrete event systems. Presently, this task is implemented using software packages specifically designed to model telecommunication networks [OO98]. BONeS Designer is such a software package developed by Alta Group of Cadence Design Systems Inc. Designer is an integrated software package for modeling and simulating event-driven data transfer systems such as communication networks, computer architectures and distributed processing systems [AIR94].

Designer provides a Motif-based graphical environment for modeling and simulating the performance of systems using discrete-event simulation techniques [AIR94]. The system can be designed by using hierarchical, data-flow type block diagrams. The top layer in the hierarchy represents general system characteristics, and the lower layers represent increasingly more detail in the system [AIR94]. The Designer core library contains prebuilt models of traffic sources, queues, timers, delays, server resources, random number generators, arithmetic, and logical operators. The main modules of the Designer software are Data Structure Editor (DSE), Block Diagram Editor (BDE), Symbol Editor (SE), Simulation Manager (SM), Post Processor (PP), and Project Editor. The Block Diagram Editor creates documents and stores block diagrams. The Data Structure Editor creates, edits, documents, and stores data structures. The Simulation Manager generates, submits, and monitors the execution of simulation programs. The Post Processor helps to analyze the results, computes statistics and displays results. The Symbol Editor is used to create custom symbols and to modify existing symbols for block diagrams.

The prebuilt powerful model library and graphical environment make it ideally suited for network simulations. Also the hierarchical structure of the Designer makes it easy to learn. For these reasons, BONeS Designer was chosen as the tool for use in modeling the real-time packet voice communication over data networks.

3.3.2 Performance Metrics

Before proceeding, performance metrics were needed for comparing the quality of the packet voice communication. Previous researchers [Mon83][WF83][KlN82] suggest that the performance of the packet voice communication can be determined by the probability of no gap and end-to-end delay. This effort compares the probability of no gap and end-to-end packet delay of a system under various conditions and parameters. There is a discussion below on each of the performance metrics.

- ***Probability of no gap (Prob[no gap])***

This is the probability that no gaps are observed during the playback of the talkspurt due to late packet arrival. Prob[no gap] is calculated as follows.

$$P[\text{no gap}] = \frac{\text{Total number of played out packets}}{\text{Total number of packets arrived at receiver}} \tag{1}$$

which is also

$$P[\text{no gap}] = \frac{\text{Total number of played out packets}}{[\text{Total number of late packets}] + [\text{ Total number of played out packets}]} \tag{2}$$

- ***End-to-End Delay***

The end-to-end (speaker-to-speaker) delays of the voice packets are important since the human perception factors produce a requirement for bounded delays. However, quantifying the voice session quality is difficult since individual human users may have different tolerances for delay. In this effort, end-to-end delays of packets are classified according to the table on page 7 given in [RaR92] and reproduced in Table 1.1.

3.3.3 Assumptions

In order to accomplish the performance studies in the simulation model, certain assumptions must be established. Some of the assumptions, e.g., the source model [HeL86] [SrW86], were used in previous research. The assumptions and configurations for the packet voice system simulation are as follows:

1. The times that a packet source spends in the ON and OFF states are exponentially distributed.
2. The number of packets in a talkspurt is geometrically distributed on the positive integers.
3. Packetized voice is encoded using Adaptive Pulse Code Modulation (ADPCM)
4. The packetization interval is 16 ms.
5. Every ADPCM coded packet fits in a frame.
6. The system is PC-to-PC configured.
7. The round-trip delay estimation technique is used for synchronization.
8. The Incomplete Timing Information (ITI) device behavior is used as the voice-receiving scheme.
9. The packet loss is random and independent.
10. The delay probability distribution function(s) of each path is determined according to goodness-of-fitness test results.

3.3.4 End-to-End System Model

The complete end-to-end packet voice communication system is modeled by three components as in real life applications. These three basic components are sender, network or transmission media, and receiver as shown in Figure 3.9.

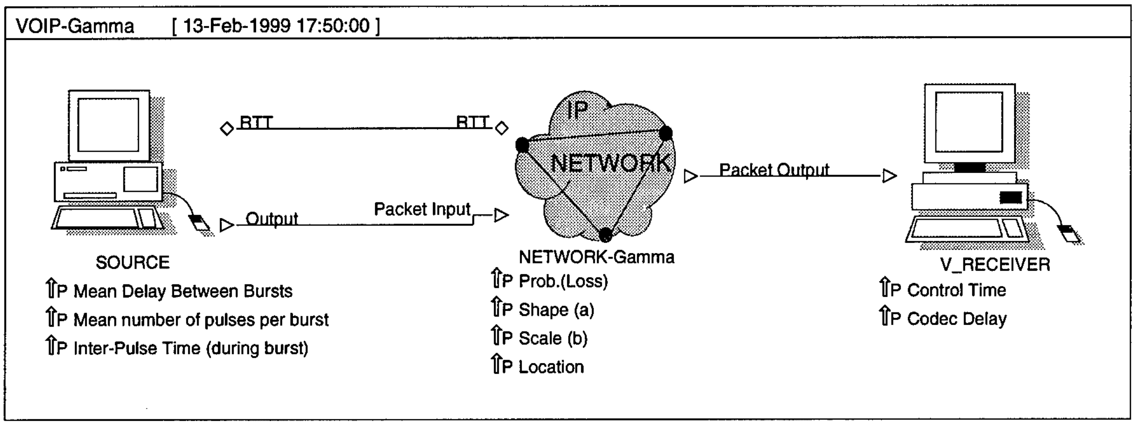


Figure 3.9 End-to-End System Model

The source component generates voice packets and fills the fields associated with itself. The network component of the model represents the delay and loss characteristics of the medium where the voice packets are transmitted. The final component, the receiver, is responsible for buffering, reordering, and scheduling of packets for proper playback.

In the following subsections, the voice source model used in the sender component, the voice packet data structure, and each of the basic components are explained in detail.

3.3.4.1 Voice Source Model

The packet stream from a single voice source is characterized by packet generations at fixed intervals of T ms during the ON state (talkspurt) and no packets during the OFF state (silence) as in Figure 3.10. The times that a source spends in the ON and OFF states are exponentially distributed with means α^{-1} and β^{-1} respectively. The number of packets in a talkspurt is approximated to be geometrically distributed on the positive integers where the mean number of packets generated is $\lceil(\alpha T)^{-1}\rceil$.

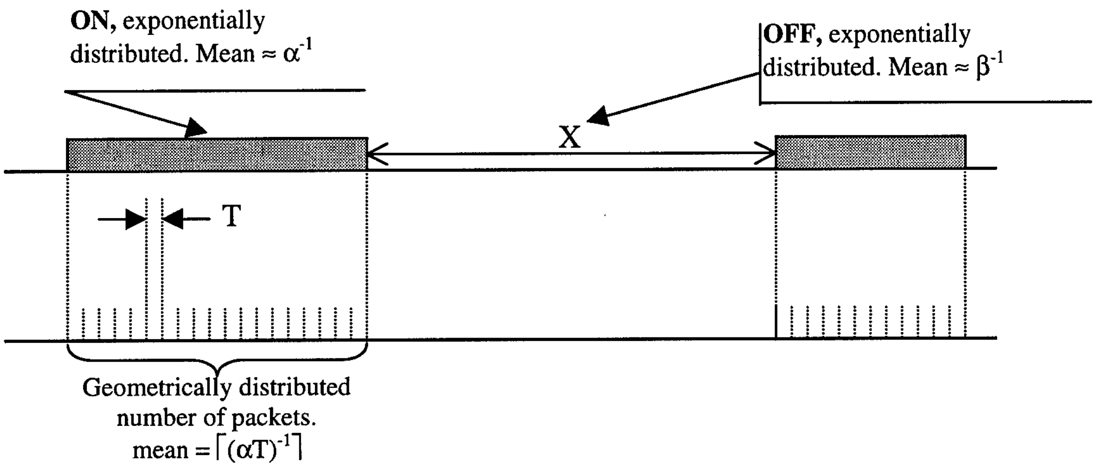


Figure 3.10 Voice Source Model

The literature review reveals that choosing α^{-1}= 352 ms and β^{-1} = 650 ms gives a model used by other researchers for a packetized voice encoded using ADPCM [HeL86][SrW86]. This model is used here also as a voice source model. To specify the voice model completely, the packetization period is chosen as T=16 ms, the silence periods are exponentially distributed with mean β^{-1} = 650 ms, and talkspurt periods are exponentially distributed with mean α^{-1}= 352 ms. Thus, the mean number of packets per talkspurt is 22. The voice packet size depends on the coding scheme. For instance, for 32 Kbps ADPCM coding and T=16 ms, the packet size is 64 bytes.

3.3.4.2 Voice Packet Data Structure

This data structure represents the voice packet of the simulation and includes three attribute and four timestamp fields. This packet structure was developed to keep track of voice packet. No voice data is contained in this simulation packet structure. Figure 3.11 shows the fields of the voice packet data structure.

Name: v_packet [vp]
Date: Saturday, 2/6/99 05:28:40 pm EST

Name	Type	Subrange	Default Value
Packet ID	INTEGER	(0, +Infinity)	...
Talkspurt Number	INTEGER	(0, +Infinity)	1
Estimated Delay	REAL	(0, +Infinity)	...
Time Created	REAL	(0, +Infinity)	...
Time Arrived	REAL	(0, +Infinity)	...
Playback Time	REAL	(0, +Infinity)	...
Q_Out Time	REAL	(0, +Infinity)	...

Figure 3.11 Voice Packet Data Structure

The *Packet ID* is the positive integer number that uniquely identifies each voice packet. The Packet ID number increases by one for each generated packet. The *Talkspurt Number* is also a positive integer number incremented after each silent period. The Talkspurt Number is same for all packets in the same period of speech activity (talkspurt). The *Time Created* field, the first of four timestamps, contains, as seen from its name, the time information when the packet is created. The *Estimated Delay* field of packet carries the estimated network delay time of each talkspurt that the packet is in. *Time Arrived* is another timestamp field used to keep track of the voice packet. Whenever voice packets arrive at the destination, the time of arrival is inserted in this field. The

Playback Time field holds the time when the voice data is supposed to be played back. Finally, the *Q-Out Time* field holds the time of arrival from the receiver buffer. Although it may not be used in real packet voice implementations, it is useful for statistical purposes in this effort.

3.3.4.3. Sender

The sender, as the first function, generates voice packets as described in the previous section. Then, it fills Packet ID, Talkspurt Number, Estimated Delay and Time Created fields of the packet. Figure 3.12 shows the interior design of this component.

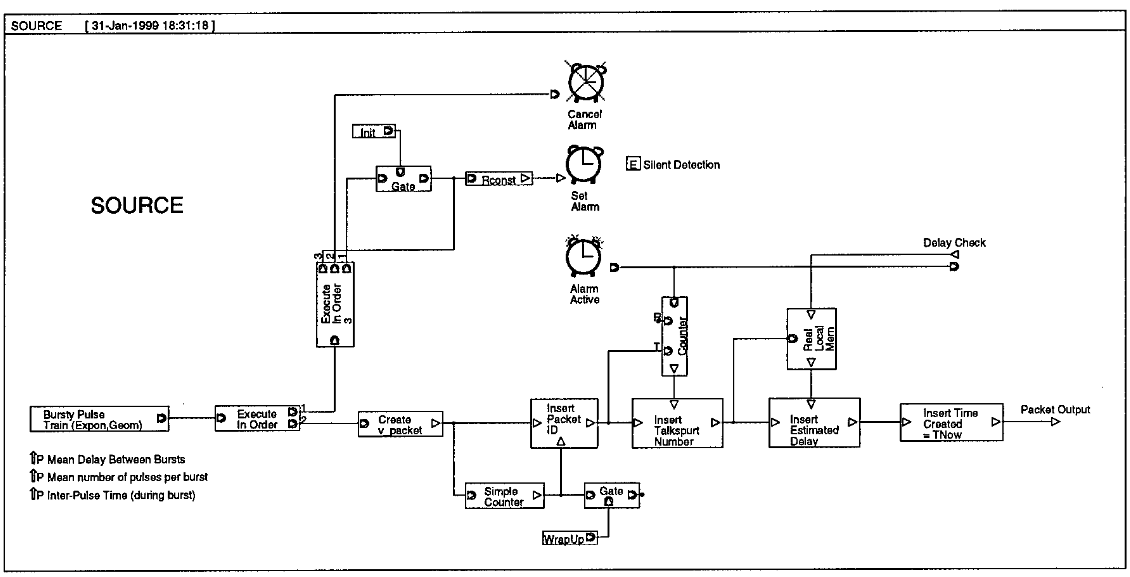


Figure 3. 12 Source Block Diagram

In order to generate voice packets, the Bursty Pulse Train Traffic Generator module of the Designer library is used. This traffic source can be used to model bursty sources. The module has random times between bursts, a random number of output pulses in each burst, and a fixed time between pulses in a burst. The time between bursts is selected from an exponential distribution, and the number of pulses in each burst is selected from a geometric distribution. Mean values for both distributions are left as parameters, as is the time between pulses during a burst.

Each pulse of the traffic generator triggers the Create v_packet module, and this module generates the specified voice packet. When the voice packet is generated, all fields in the newly created packet are set to their default values. Each pulse of the traffic generator also triggers a simple counter. The value of the counter is initially set to zero. Each time the counter is triggered, its value is incremented by one, and this value is used as the Packet ID. The Insert Packet ID module sets a Packet ID field of the voice packet to the current counter value.

In order to determine the talkspurt number, a silence detection mechanism is implemented by using alarm modules of Designer (Figure 3.12). The silence detection algorithm keeps track of the interval time between subsequent voice packets. If the interval time between packets exceeds the predetermined fixed packetization interval time, silence is detected, and this means a new talkspurt. Therefore, the detection alarm is set to the packetization interval time at the beginning of the packet generation. Each traffic generator pulse, or each new packet, first cancels the alarm and then resets it. If the alarm is not cancelled, the alarm active module provides a signal that indicates that a packetization interval time has expired. This signal is used to trigger a counter to increment the talkspurt number by one. The T, standing for test, input port of the counter is triggered for each packet to get the current talkspurt number. The Talkspurt Number field is also set to this value by the Insert Talkspurt Number module.

Delay estimation is made for each talkspurt, and the estimated delay value is used for all packets in the same talkspurt. Delay estimation is made by using a simple divide by two method since there is no reasonably accurate technique for measuring unidirectional delays. Whenever the silent detection alarm becomes active, the delay generator that is in the network component is triggered, and the output value of the delay generator is stored in the local memory module. After insertion of the talkspurt number, the estimated delay value is taken by enabling the read port of the real local memory module and placed into the estimated delay field by the Insert Estimated Delay module. In order to get the initial value of the estimated delay that is needed for the first talkspurt, the network delay generator is triggered once at the beginning of the simulation. At the end of the sender component, the Time Created field is set to TNow (the simulation time) and the packet is sent to the network component.

3.3.4.4. Network

The Network component is designed to simulate where the packet passes through to the network to arrive at the receiver. Figure 3.13 shows the internal design of this component. This component represents the delay and packet loss characteristics of the transmission media.

After packets are sent from the sender component, they arrive at the random switch module of the network component. The random switch module was used to represent packet loss rate with loss probability P. This module randomly switches the input packet to one of the two outputs. The input packet is placed on the output Lost with probability P and is placed on the output Network with probability 1-P. The packets placed on the output Network continue their way while others end their journey in the sink module.

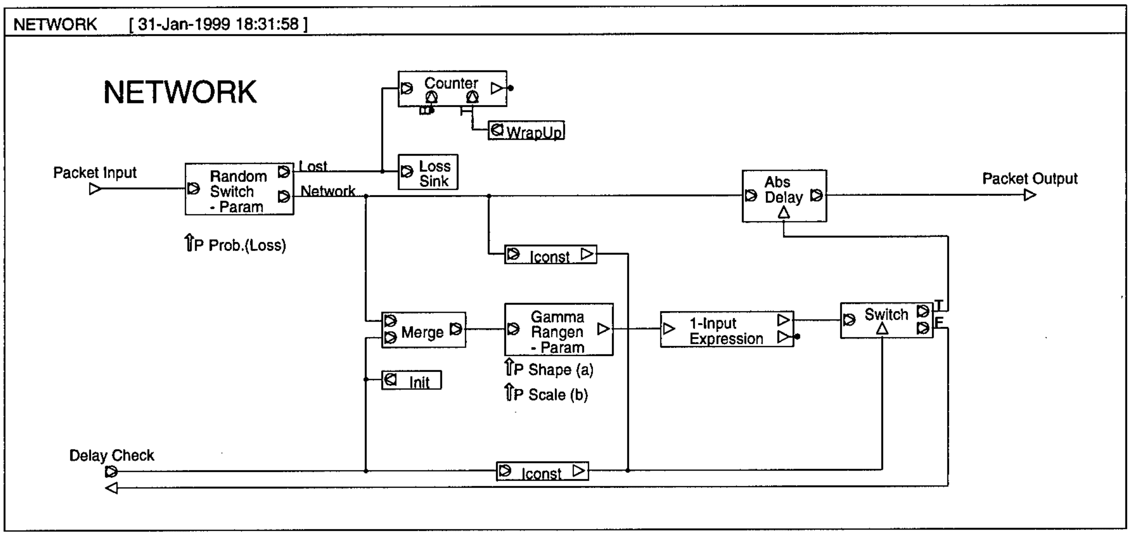


Figure 3.13 Network Block Diagram

This configuration utilizes the Random Number Generator module to generate the time spent by each packet in the transmission medium. The type of random number generator, Gamma, Beta, etc., is determined by the results of the experimental measurements for each real world path and discussed more fully in Section 4.2. In Figure 3.13, the Gamma Random Number Generator is illustrated as an example.

The Absolute Delay module delays each packet for the delay time generated by the number generator. As a summary, some packets leave the network component with some delay and arrive at the receiver component while others become lost at the random switch module. The delay value, delay jitter, and loss probability are based on the measured data acquired in Section 3.2.

3.3.4.5. Receiver

Figure 3.14 shows the block diagram of the receiver component of the system. The receiver component first determines the arrival time of the packet at reception, then inserts this time information into the Time Arrived field of the packet. After reception, playback time of the packet is calculated. In order to calculate the playback time, the packet creation time, estimated delay time of the talkspurt which the packet is in, and the control time are taken into account as explained in the ITI receiving scheme (Section 2.6).

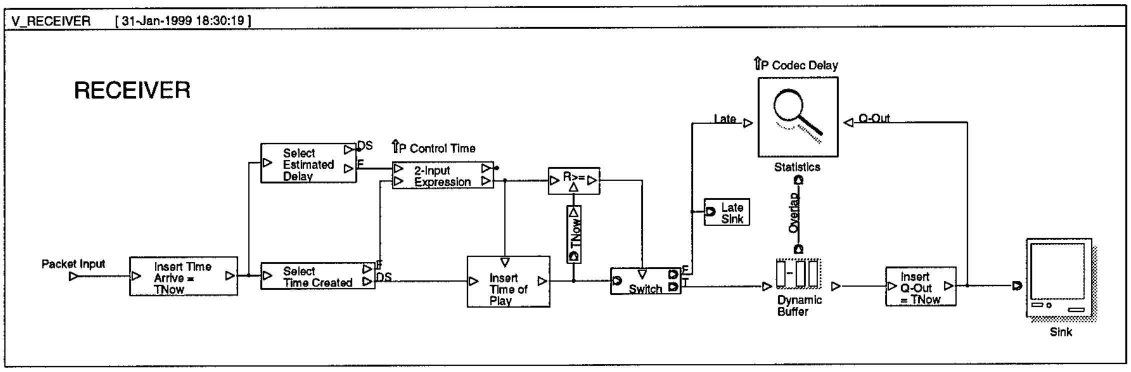


Figure 3.14 Receiver Block Diagram

Therefore;

$$T_{playback} = T_{created} + T_{estimated} + T_{control} \tag{3}$$

Equation 3 is solved in the 2-input Expression module. Control Time is taken as a system parameter in this module. After determination of the playback time, it is compared with the simulation time (Tnow). If packet playback time is in the past of the simulation time, this means that the packet is too late and causes a gap at the playback. Therefore, the late packet is sent to the sink. Otherwise, the packet enters the Dynamic Buffer module. The Dynamic buffer is the reordering and scheduling mechanism for proper playout of packets encountering varying transit delay times. Figure 3.15 shows the block diagram of the Dynamic Buffer.

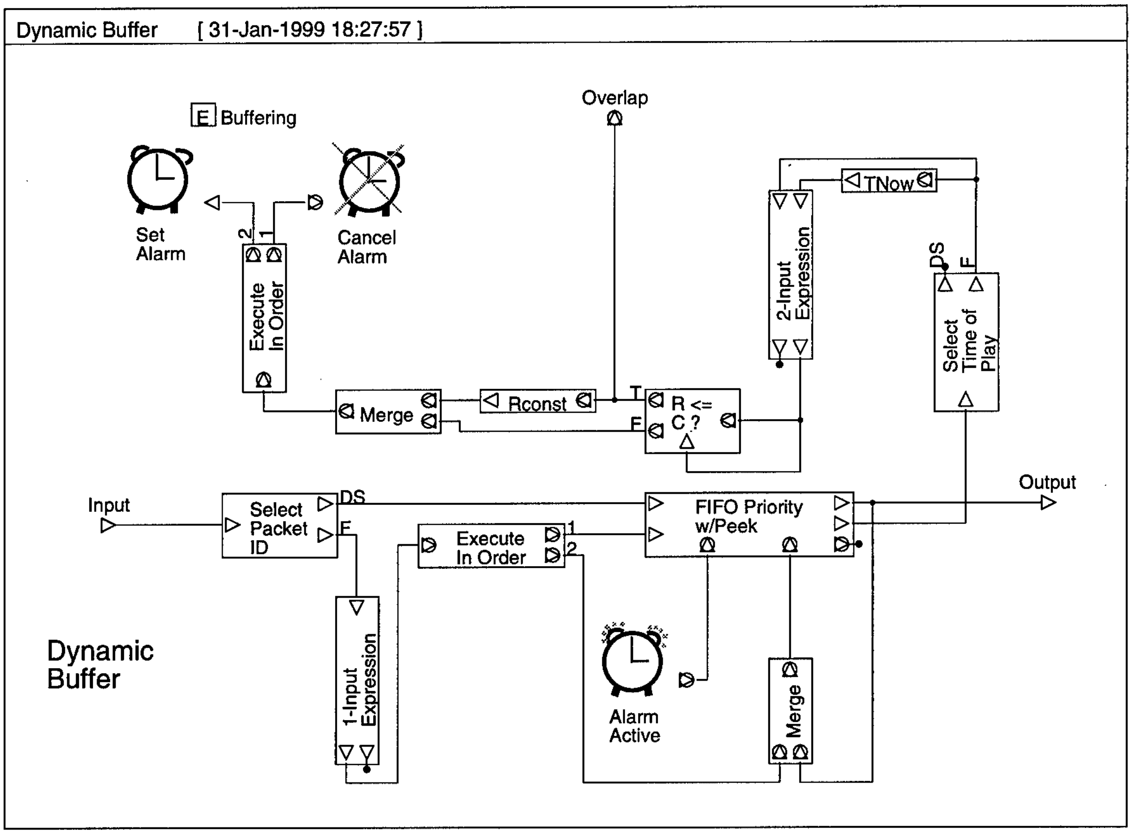


Figure 3.15 Dynamic Buffer Block Diagram

The FIFO Priority w/Peek queue module of Designer is utilized to play back each packet at the scheduled time. Every incoming packet is ordered in the queue with its priority. The priority of each packet is inversely proportional to its packet ID number, so packets with smaller Packet ID have higher priority. The FIFO queueing mechanism of this module is not utilized since every packet has a unique packet ID, interpreted as priority. Every packet that arrives at the Dynamic Buffer module refreshes the current scheduling of packets in the queue. In the refreshing procedure, playback time information of the packet at the head of the queue is compared to the simulation time, and the buffering alarm is set to the difference between these two times. Whenever the buffering alarm has been activated, the packet at the head of the queue leaves the queue and also triggers the refreshing procedure for a new condition of the queue.

In order to measure the performance of the end-to-end system, performance metric elements are collected in the Statistics module of the receiver. Figure 3.16 shows the block diagram of the Statistics module.

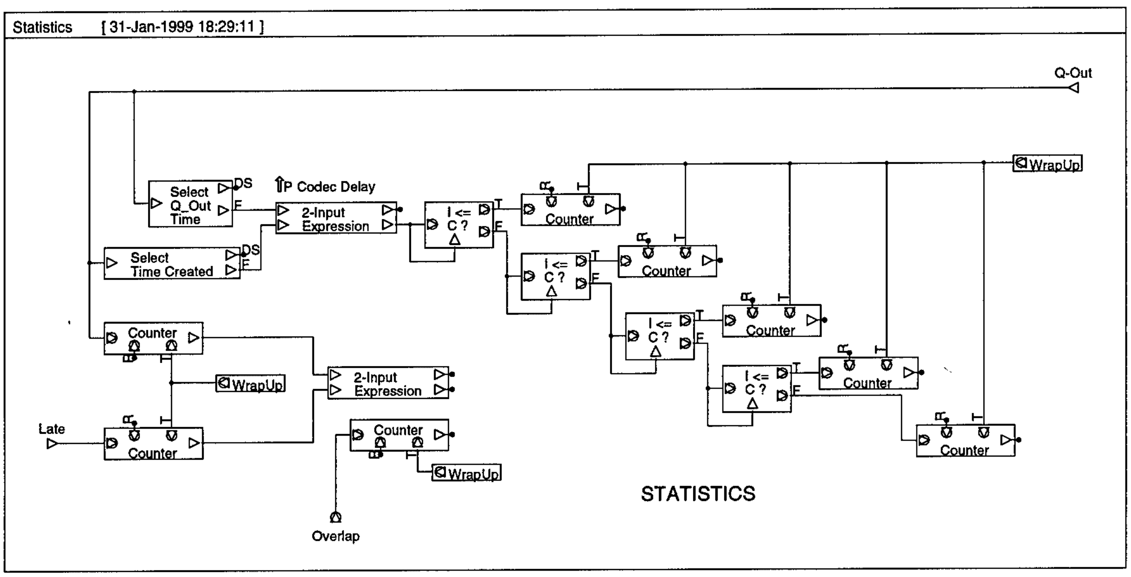


Figure 3.16 Statistics Module Block Diagram

This module collects the total number of played out and late packets. It also computes the one-way end-to-end delay of each packet and classifies them according to one-way delay intervals discussed in Section 3.3.2.

3.3.5 System Parameters

Several parameters are used to simulate different configurations and characteristics of the end-to-end system. These parameters are as follows:

1. Mean Delay Between Bursts: The mean value for the exponential random number generator from which the delay between bursts is taken.
2. Mean number of pulses per burst: Each burst has a random number of output pulses. This parameter specifies the mean of the geometric distribution from which individual burst lengths are chosen.
3. Inter-Pulse Time (during burst): The fixed time period between output pulses during a burst.
4. Loss Probability: The probability for each packet of being lost in the network.
5. Delay Distribution Parameters: The parameters of the delay distribution function obtained by experimental measurements and its parameters such as shape, scale, and location for Gamma distribution.
6. Control Time: The fixed delay time at the receiver for each packet in order to compensate delay jitter.
7. Codec Delay: The one-way delay of encoding and decoding processes.

3.3.5.1 Output Metrics

Section 3.3.2 gives the required metrics. In order to compute these required metrics, the following output metrics must be obtained from the simulation.

1. Total number of packets generated.
2. Total number of talkspurts.
3. Total number of lost packets.
4. Total number of late packets.
5. Total number of packets played.

6. One-way end-to-end delay of packets.

3.3.6 Simulation Run Time

Simulations exhibit random variability when random number generators are used to produce the values of the input variables [BaC96]. Therefore, the simulation run time has to be long enough in order to get results in the steady-state condition. In order to estimate the steady-state time of the simulation, several tests were made for different simulation run times and different random number generators. Figure 3.17 gives the result of the one of these tests. P[no gap] was used as a performance metric for these tests.

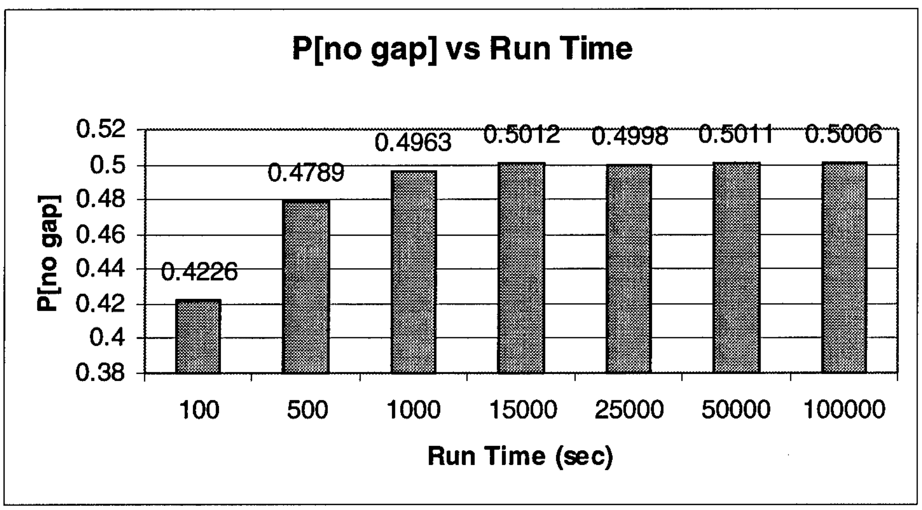


Figure 3.17 Steady-state Time Estimation Test Results

The result of this test shows that each simulation should run for at least 15000 simulation seconds. The simulation run time was chosen as 20000 seconds in this effort.

3.3.7 Verification

Verification is concerned with building the model right. The conceptual model must be accurately represented by the computerized system [BaC96]. Verification of the packet voice communication simulation models was accomplished by Designer Module Block Verification, Designer's Interactive run controller, and closely examining the model's output for accuracy under a variety of settings of the input parameters.

3.3.7.1 Designer Module Block Verification

Each block was built and tested at the lowest level before building upward. Each level of a block was verified before building further upward to the system level. This verification process ensured that all dependencies and block construction were correct. Testing of each block was accomplished by placing probes at the output and comparing the output data with the expected output. For example, placing a probe at the output of the source verified that the correct Packet ID, Talkspurt Number and timestamps were inserted in the voice packet data structure.

3.3.7.2 Designer Interactive Run Controller

The simulation model was monitored as it progressed using the Interactive Run Controller (IRC). The IRC verified that the correct path was taken within modules and throughout the system. For example, if the packet playback time was less than the simulation time, the packet was sent to the late sink. Any warnings or errors encountered were quickly resolved with the IRC's help.

3.3.7.3 Accuracy of Output Data

A design may be very good, but if the output does not make sense, the entire effort will be wasted. For this reason, all output was checked for accuracy. For example, the total number of packets generated was compared with the total number of the lost, late and played packets.

3.3.8 Validation

Validation is concerned with building the right model [BaC96]. Validation tests can sometimes be performed by comparing the model and its behavior to the real system and its behavior. Validation of the model by comparing to real systems can't be accomplished, since no data is available from a real system with same configuration. Instead, the validation of models is determined by using a three-step approach [NaF67], which has been widely followed. The three steps are building a model that has high face validity, validating the model assumptions, and comparing the model and real system input-output transformations. The first two steps can be performed on these models, but the last one can't, since a system with same configuration does not exist.

3.3.8.1 Face Validity

Building a model with high face validity involves construction a model that appears reasonable on its face to model users and others who are knowledgeable about the real system being simulated [BaC96]. These simulation models were built with high face validity. The voice source model of the system is well known and was used in previous research [SrW86] [HeL86].

Sensitivity analysis can also be used to check a model's face validity. Sensitivity analysis involves changing the input variables and observing the changes in the output. For example, increasing the Control Time of the receiver resulted in expected increase in the probability of no gap and increase in the end-to-end delay of the packet.

3.3.8.2 Validation of Model Assumptions

Model assumptions fall into two general classes: structural assumptions and data assumptions. Structural assumptions involve questions of how the system operates. The voice source model assumptions, choosing ON and OFF periods exponentially distributed is also well known and widely used in previous research [Bra68][HeL86][SrW86]. The time estimation and receiving scheme of the system being modeled were taken from previous research [Mon83] [BP80].

Data assumptions involve the correct input data. Validation of input measurements and relevant assumptions is given in the Section 3.2.6.

3.4 Summary

This chapter presented a methodology that was used to analyze the impact of the QoS factors and parameters on the packet voice communication quality over the IP networks. Sample real-world IP network characteristics were obtained by measurements for short and long-haul networks in order to use realistic simulation inputs. The packet voice communication system was designed in modular fashion using the Designer. The measurements and simulation models were verified and validated. Performance metrics and assumptions were defined.

4 Results and Analysis

4.1 Introduction

The impact of the factors that affect quality of the real-time packet voice communication over IP networks is analyzed in this chapter. First, there is a discussion about the network delay distributions obtained from sample measurements. Then, the simulation outputs are examined in Section 4.3. Analysis of the results is discussed in Section 4.4. The impact of the control time is studied in detail since it is the only parameter that can be controlled by the receiver. This study developed several mathematical expressions of the probability of no gap in terms of the control time and the standard deviation based on the simulation results. The comparison of these mathematical expressions is provided in this section. The effect of the delay and the packet loss over packet voice communication are discussed in Section 4.5 and Section 4.6 respectively. The mathematical expressions of these factors are given in terms of the other system parameters in the sections associated with them. Finally, the evaluation of analysis results is presented in Section 4.7.

4.2 Input Delay Distributions

Round-trip delay distribution of each path was determined by the result of the chi-square goodness-of-fit test. According to the test results, the distribution function with the best rank was selected to represent the delay distribution of the path. The selected delay distributions and parameters of each path are given in Appendix B. Chi-square goodness-of-fit test results gave Gamma as the first best-fit distribution function to the given data for five of the ten paths. Pareto distribution functions were found for four other paths, and the Extreme Value distribution function was found for the final path. However, Gamma was also the second best-fit distribution function for all of these five paths that gives the Pareto and the Extreme Value distribution as a first best-fit function. The goodness-of-fit ranks of the first best-fit functions (Pareto or Extreme Value) and Gamma function, as a second best-fit function, were very close to each other. Gamma function was even found first best-fit instead of second best-fit function for some of these paths according to the other, Anderson-Darling, goodness-of-fit test results. Therefore, Gamma function was also selected to represent the delay distribution of these other five paths. This assumption is not unrealistic since the chi-square goodness-of-test ranks are very close, and Gamma is already first best-fit distribution function for some of these paths according to the Anderson-Darling test results. The functions that cause conflicts (the Pareto and the Extreme Value) are also given in Appendix B in addition to the Gamma so that the reader can compare and see the similarities.

4.3 Simulation Results

Simulations were executed for the delay distributions discussed above and different control time values for each path. For the first group of simulations, the control time system parameter was started from zero and incremented by ten milliseconds for each iteration of the simulation. After the first group of simulations, it was shown that the probability of no gap sometimes has big jumps between two sequential control time inputs. These significant changes of probability of no gap were occurred at different values of the control time according to the value of the delay variance. Additionally, a second group of simulations were performed for control times that are in the interval of these significant changes in order to see the probability of no gap response of the system more accurately.

After the second group of simulations were finished, the results were examined in order to explore the trade-offs among the control time, delay standard deviation, and probability of no gap. During this investigation:

- The probability of no gap for different standard deviation values in the case of no jitter control (Control Time = 0)
- The relation between control time, standard deviation and probability of no gap in the case of jitter control (Control Time > 0)
- The correlation between the control time and the standard deviation during significant changes of the probability of no gap
- The jitter compensation rate of the control time, P[no gap], in the case where the control time is equivalent to the standard deviation
- The probability of no gap results according to the increases of the control time
- And other correlation or relations that are believed to deserve examination

are examined carefully.

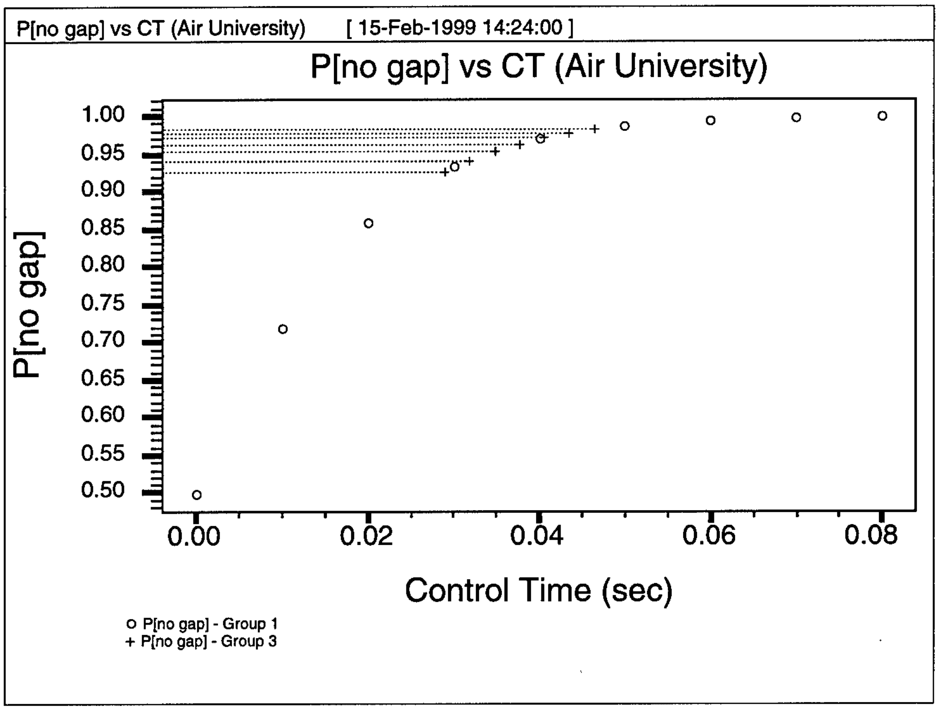
The relationship between P[no gap] and the control time was found to be remarkable for the values of the control time that are equal to or greater than the standard deviation. Therefore, the third group of simulations was run for the control times that are exactly equal to the standard deviation and, 10% increments of standard deviation for each iteration.

The outputs of these three group simulations were given for each path in the following figures. Figures 4.1-4.10 show:

- The delay distribution function with parameters, and standard deviation values,
- The probability of no gap results of all simulations.

Larger scaled representation and numerical values of the figures can be found in Appendix D.

4.3.1 AFIT-Air University Path



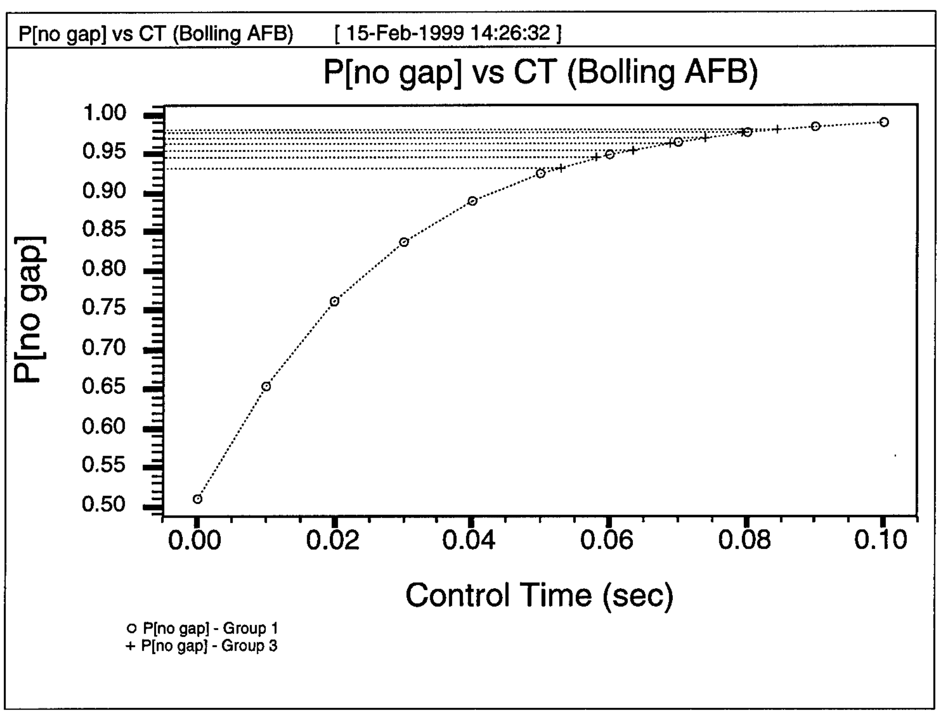
Shape : 2.10

Scale : 19.99

Std. Dev: 28.97 ms.

Figure 4.1 P[no gap] for AFIT-Air University Path

4.3.2 AFIT-Bolling AFB Path



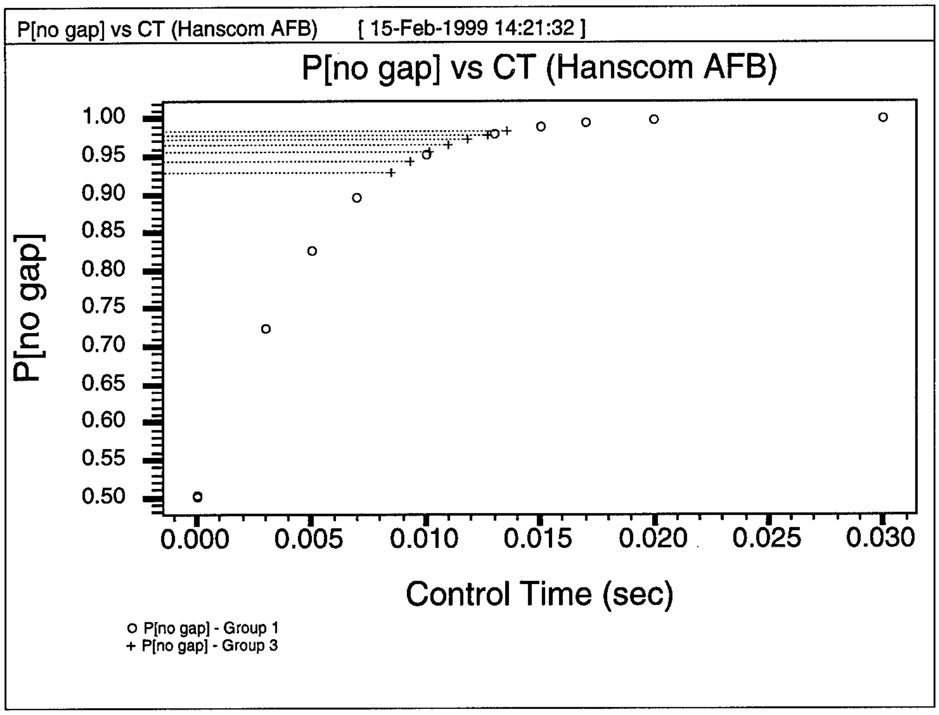
Shape : 1.25

Scale : 47.25

Std. Dev: 52.83 ms.

Figure 4.2 P[no gap] for AFIT-Bolling AFB Path

4.3.3 AFIT-Hanscom AFB Path



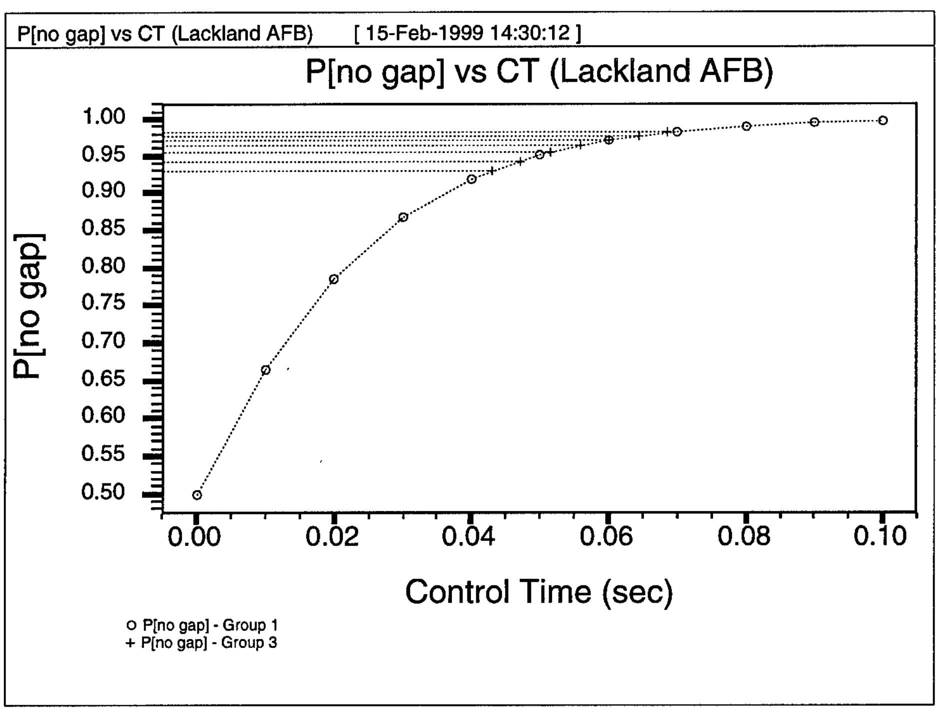
Shape : 2.18

Scale : 5.72

Std. Dev: 8.45 ms.

Figure 4.3 P[no gap] for AFIT-Hanscom AFB Path

4.3.4 AFIT-Lackland AFB Path



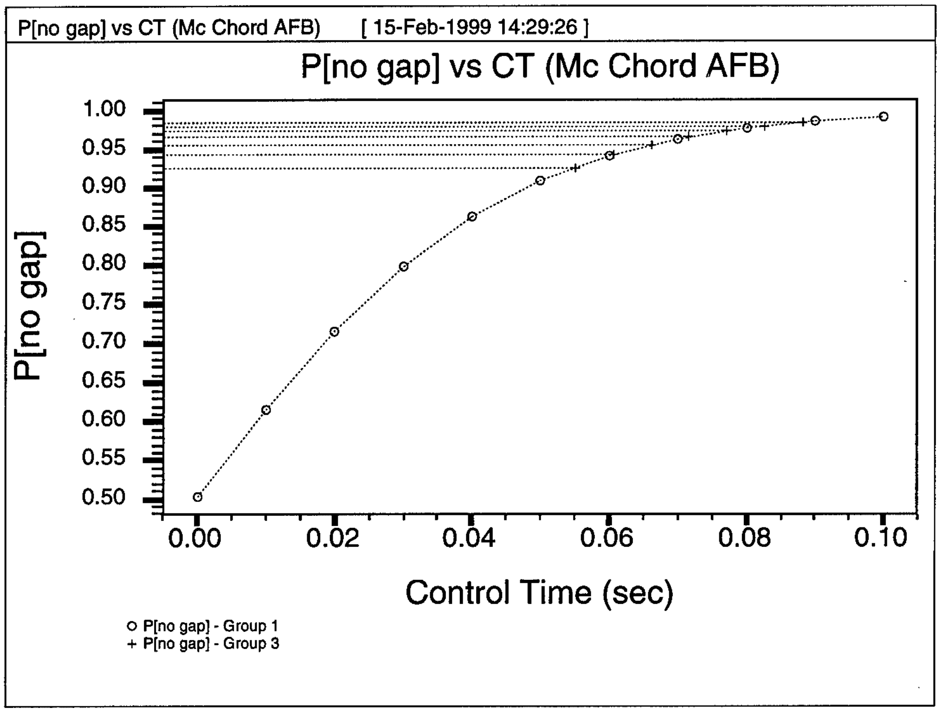
Shape : 1.55

Scale : 34.45

Std. Dev: 42.91 ms.

Figure 4.4 P[no gap] for AFIT-Lackand AFB Path

4.3.5 AFIT-Mc Chord AFB Path



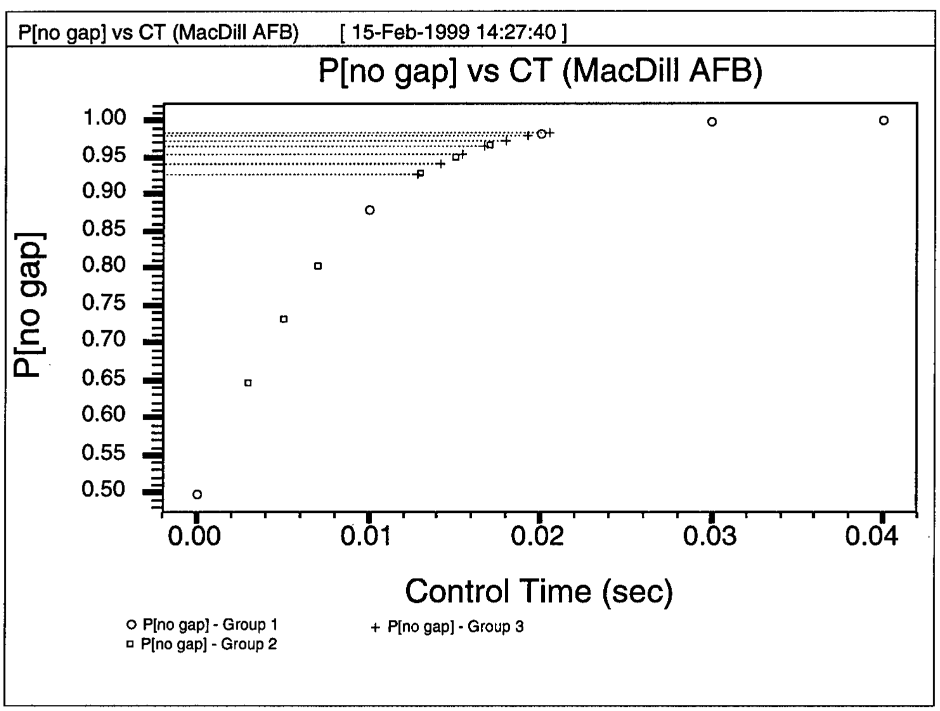
Shape : 3.79

Scale : 28.28

Std. Dev: 55.08 ms.

Figure 4.5 P[no gap] for AFIT-Mc Chord AFB Path

4.3.6 AFIT-MacDill AFB Path



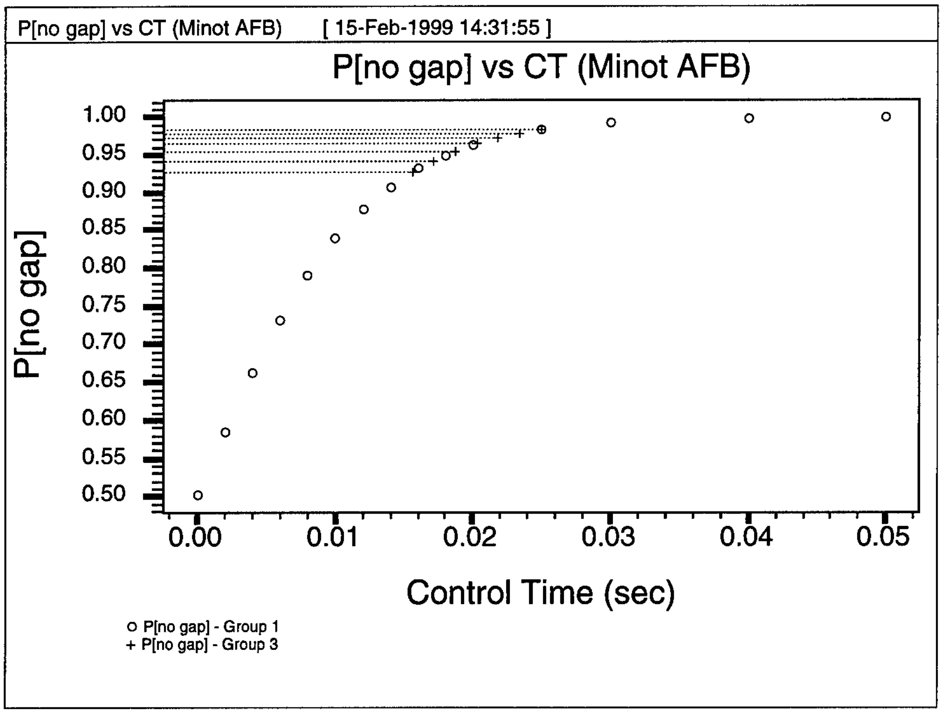
Shape : 2.91

Scale : 7.52

Std. Dev: 12.84 ms.

Figure 4.6 P[no gap] for AFIT-MacDill AFB Path

4.3.7 AFIT-Minot AFB Path



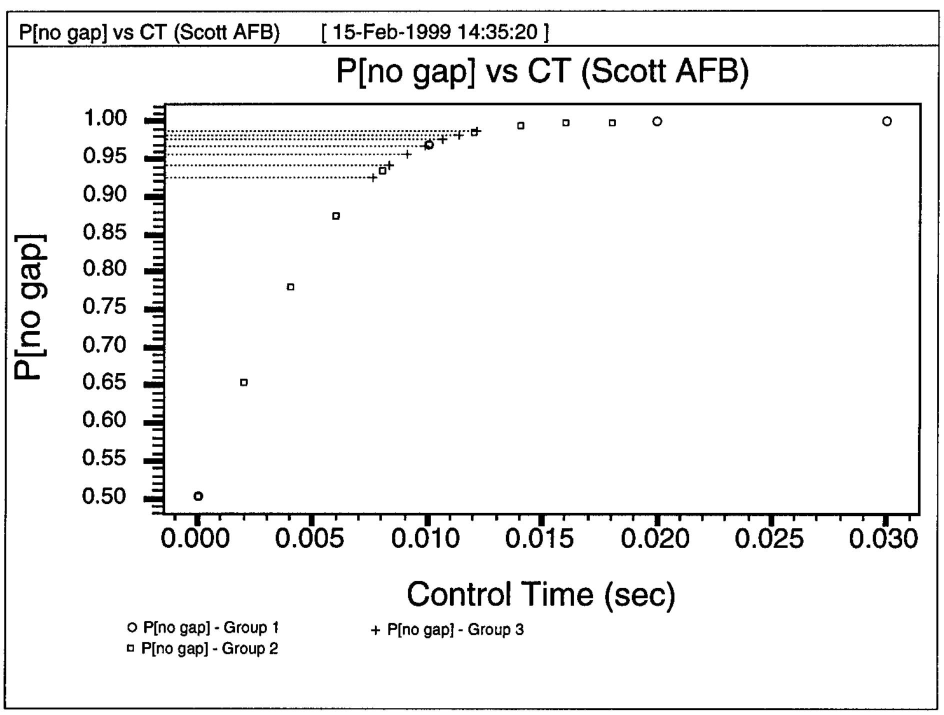
Shape : 2.95

Scale : 9.07

Std. Dev: 15.58 ms.

Figure 4.7 P[no gap] for AFIT-Minotl AFB Path

4.3.8 AFIT-Scott AFB Path



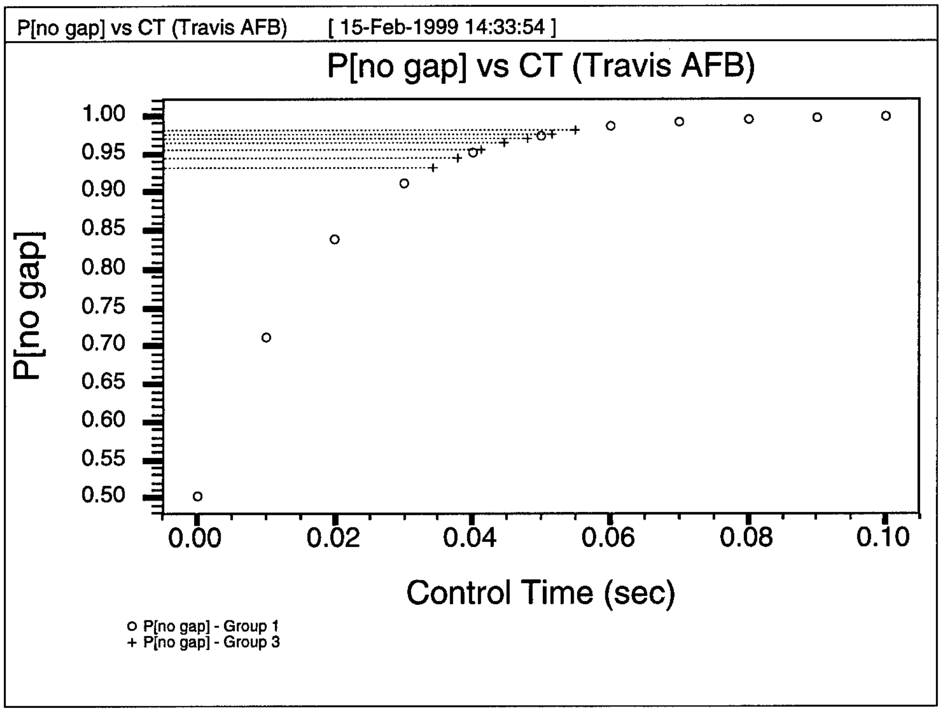
Shape : 10.28

Scale : 2.36

Std. Dev: 7.56 ms.

Figure 4. 8 P[no gap] for AFIT-Scott AFB Path

4.3.9 AFIT-Travis AFB Path



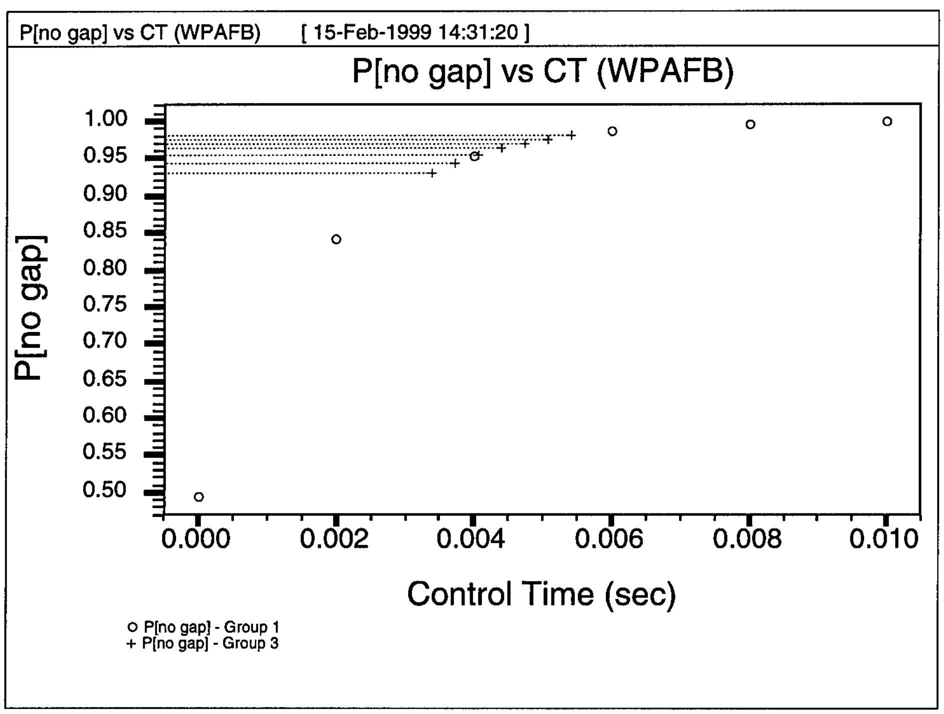
Shape : 1.22

Scale : 30.92

Std. Dev: 34.27 ms.

Figure 4.9 P[no gap] for AFIT-Travis AFB Path

4.3.10 AFIT-WPAFB Path



Shape : 1.17

Scale : 3.11

Std. Dev: 3.37 ms.

Figure 4.10 P[no gap] for AFIT-WPAFB Path

4.4 Analysis of Results

The factors that affect quality of service were given as delay, delay jitter, packet loss, and codec in Section 1.3. The end-to-end delay of the voice packet has bounds for the interactive nature of human conversation, and the delay jitter causes gaps at the receiver. Control time is a part of the end-to-end delay of a voice packet and also used to compensate for delay jitter. The value of the control time affects both delay and the probability of no gap. In addition to these functions, the control time also has a very important feature, that is, controllability by the user. We can control the value of it, but not the network delay and the packet loss quality of service factors. Therefore, control time plays a key role in case of optimization of the system to achieve the best performance. Because of this fact, the analysis of simulation outputs was started from the control time in order to take advantage of its controllability.

After results were analyzed, following conclusions can be made according to the results of the simulations:

1. The probability of no gap is approximately 0.5 for all paths, standard deviation values, in case of no jitter control mechanism (control time = 0)
2. The probability of no gap is approximately 0.93 for all paths if we set the control time equal to the standard deviation.
3. If we look at the trend of the probability of no gap for the control time values larger than the standard deviation, the rise of the P[no gap] slows down significantly and causes a curve in the figures (Figure 4.1-4.10). This curved area is subject to investigation in the trade-off between P[no gap] and the control time, because the increment of the control time does not change the value of P[no gap] effectively after some point in this area.

Curve fitting and regression methods such as the methods of least squares were used to express the relationship between the P[no gap] and the control time in mathematical form. One of the main purposes of curve fitting is to estimate one of the variables (the dependent variable) from the other (the independent variable). The process of estimation is often referred to as *regression*. Generally, more than one curve of a given type will appear to fit a set of data. To avoid individual judgment in constructing lines, parabolas, or other approximating curves, it is necessary to agree on a definition of best-fitting. The R-square value of goodness-of-fit measures the proportion of the total variation accounted for by the model. R-square is 1 if the model fits perfectly. A R-square of 0 means that the fit is no better than the mean of input data.

The relationship between P[no gap] and the control time was studied by using regression and curve fitting methods. The JMP IN® statistic software tool was used to implement different regression methods.

The k is defined as the ratio of the control time and the standard deviation as in Equation 10.

$$k = \frac{Control\ Time}{Standard\ Deviation} \tag{10}$$

The k variable is used in the following sections to represent the control time in terms of the standard deviation.

4.4.1 P[no gap] Estimation for $1 \leq k \leq 1.6$

Exact values of the probability of no gap for control times that are increased gradually by 10% of standard deviation are given in Table 4.1 and shown in Figure 4.11.

Table 4.1 P[no gap] vs Control Time for Paths ($1 \le k \le 1.6$)

P[no gap]		CONTROL TIME						
		1*SD	1.1*SD	1.2*SD	1.3*SD	1.4*SD	1.5*SD	1.6*SD
PATH	Air University	0.926	0.941	0.953	0.963	0.971	0.977	0.982
	Bolling AFB	0.933	0.946	0.956	0.965	0.971	0.977	0.981
	Hanscom AFB	0.929	0.943	0.955	0.965	0.972	0.978	0.983
	Lackland AFB	0.930	0.943	0.955	0.963	0.971	0.977	0.982
	Mc Chord AFB	0.926	0.943	0.956	0.966	0.973	0.979	0.984
	MacDill AFB	0.927	0.942	0.955	0.965	0.973	0.979	0.984
	Minot AFB	0.927	0.942	0.955	0.965	0.972	0.978	0.983
	Scott AFB	0.925	0.942	0.956	0.967	0.976	0.982	0.987
	Travis AFB	0.932	0.945	0.955	0.964	0.970	0.976	0.981
	WPAFB	0.931	0.944	0.955	0.963	0.970	0.976	0.980

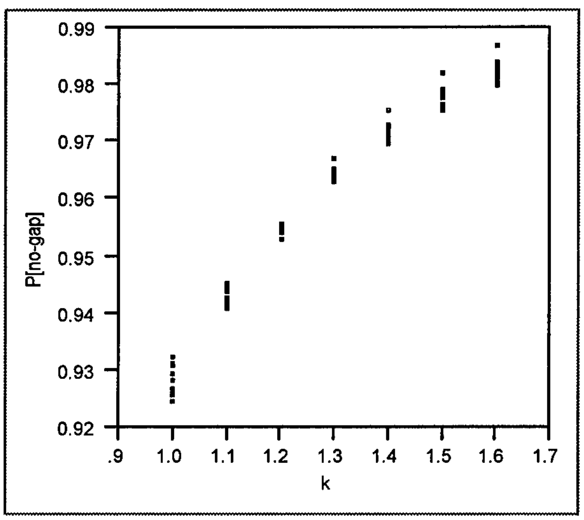


Figure 4.11 P[no gap] for k values

Several expressions of P[no gap] in mathematical form were obtained from different methods and transformations. Two of these formulas that have at least 0.98 of R-square value are given in the following sections.

4.4.1.1 Estimation Formula 1 (Polynomial Fit degree=2)

Probability of no gap was expressed by a polynomial equation with degree two in this estimation. Equation 11 was developed to estimate the probability of no gap in terms of k.

$$P[no\text{-}gap] = 0.68236 + 0.34514\,k - 0.09855\,k^2 \qquad (11)$$

If we substitute k from Equation 10, this equation can be derived in terms of the control time (CT) and the standard deviation (SD) as in Equation 12.

$$P[no\text{-}gap] = 0.68236 + 0.34514\,\frac{CT}{SD} - 0.09855\,\left(\frac{CT}{SD}\right)^2 \qquad (1 \leq k \leq 1.6) \qquad (12)$$

This formula can represent the probability of no gap and the k relation with an R-square value of 0.990. The simulation and formula outputs are plotted in Figure 4.12. Details of this and the next estimation, such as Root Mean Square Error, Mean of Response, are given in Appendix E.

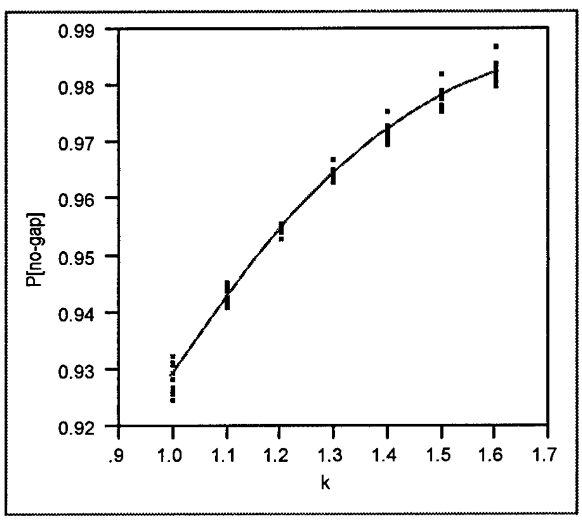


Figure 4. 12 P[no gap] from formula 1

4.4.1.2 Estimation Formula 2 (Transformed Fit to Reciprocal)

Probability of no gap was expressed by transformation of the k in reciprocal form in this estimation. Equation 13 was developed to estimate the probability of no gap.

$$P[no\text{-}gap] = 1.0749 - 0.14508\left(\frac{1}{k}\right) \qquad (1 \leq k \leq 1.6) \qquad (13)$$

This equation can be derived in terms of the control time (CT) and the standard deviation (SD) as shown in Equation 14.

$$P[no\text{-}gap] = 1.0749 - 0.14508\left(\frac{SD}{CT}\right) \qquad (1 \leq k \leq 1.6) \qquad (14)$$

The R-square value of this formula is 0.988. The simulation and formula outputs were plotted in Figure 4.13.

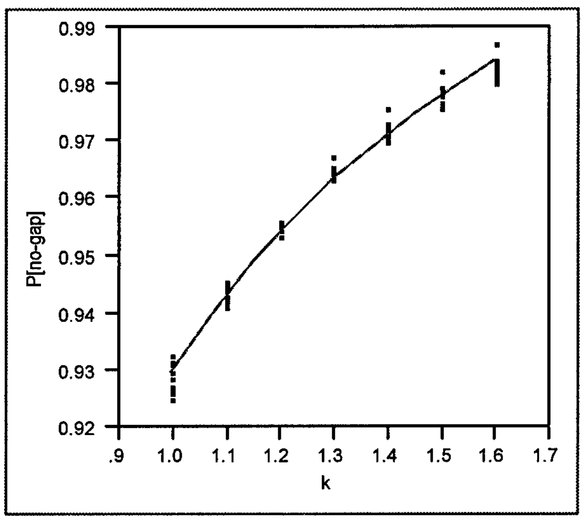


Figure 4. 13 P[no gap] from formula 2

4.4.1.3 Comparison of Formula 1 and the Formula 2

It may not be valid to compare formula 1 and formula 2 just according to the R-square values since the R-square values (0.990 and 0.988) of both fitting formulas are close to each other. In order to make a comparison for the behavior of our system, we have to consider that the positive reaction of the P[no gap] is decreasing for the increasing values of the control time in this area ($1 \leq k \leq 1.6$). The first formula represents this behavior better than the second one as shown in Figure 4.14. Thus, Formula 1 is assessed to be more accurate for our system in this area.

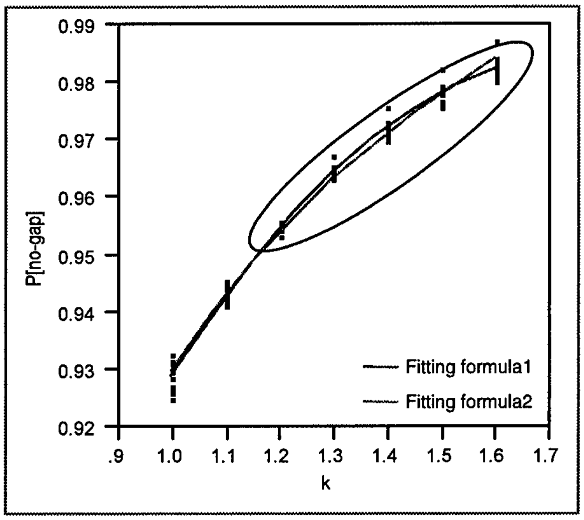


Figure 4. 14 Formula 1 vs. Formula2

4.4.2 P[no gap] Estimation for $0 \leq k \leq 4$

The fitting formulas given above were found for k values that are equal or larger than 1 and less then 1.6. Obviously, they can't be used to represent the trend of probability of no gap for all values of the control time and the standard deviation. Additional curve fit and regression methods were employed to involve all control time, standard deviation, and probability of no gap values. In this case k value starts from zero and goes up to four. Figure 4.15 shows the values of probability of no gap for all k values obtained by all simulations. Several formulas were obtained from regression methods in order to estimate probability of no gap in terms of k, or the control time and standard deviation. Three of these fitting formulas that have at least R-square value of 0.98 are given below.

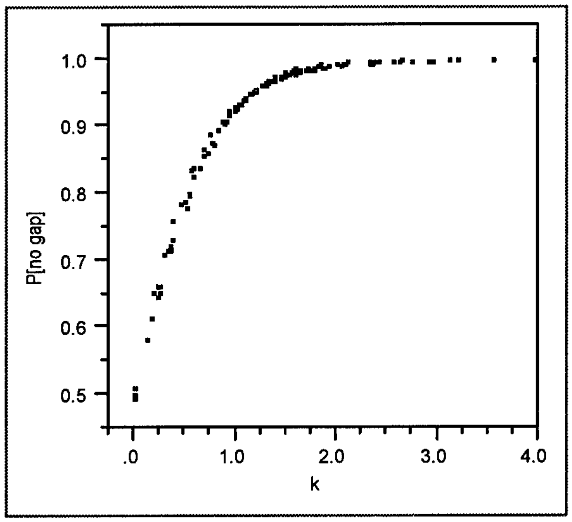


Figure 4. 15 P[no gap] for all k values

4.4.2.1 Estimation Formula 3 (Polynomial Fit degree=4)

Probability of no gap was expressed by a polynomial equation with degree four in this fitting. Equation 15 was developed to estimate the probability of no gap in terms of k.

$$P[no\ gap] = 0.50093 + 0.78095\,k - 0.46367\,k^2 + 0.12111\,k^3 - 0.01159\,k^4 \quad (15)$$

The outputs of this formula and the original simulation results for the k values are given in Figure 4.16. The R-square value of this formula is 0.998. The linear regression detail, such

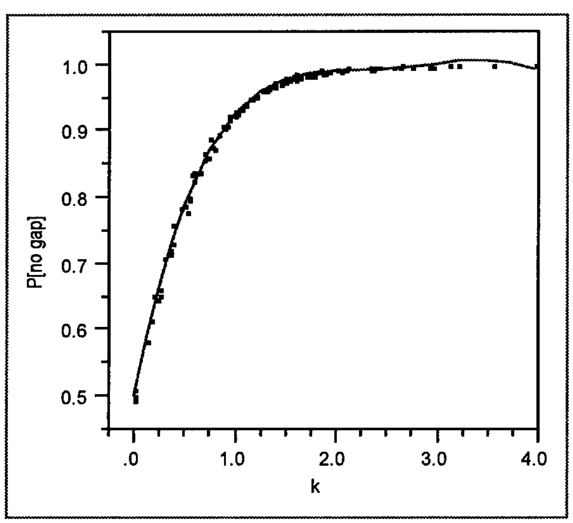


Figure 4.16 P[no gap] from formula3

as R-square, R-square Adjustment, Root Mean Square Error, and Mean of Response of this and the following formulas created for all the k values, can be found in Appendix F.

4.4.2.2 Estimation Formula 4 (Transformed Fit Reciprocal to Reciprocal)

In this mathematical form, both probability of no gap and k were transformed to their reciprocal forms in order to formulate a relationship between P[no gap] and k. Equation 16 was developed with 0.94 R-square value with these transformations.

$$\frac{1}{P[no\ gap]} = 0.9502 + 0.1276\,\frac{1}{k}$$

$$\text{or} \qquad (16)$$

$$P[no\ gap] = (0.9502 + 0.1276\,\frac{1}{k})^{-1}$$

The output values of this formula for all k values were given in Figure 4.17.

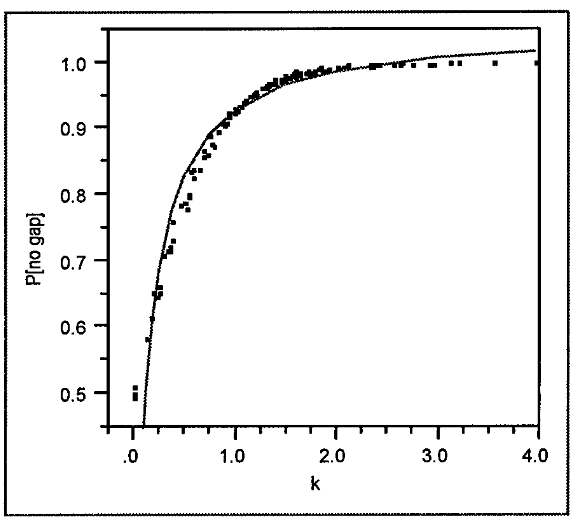


Figure 4.17 P[no gap] from formula 4

4.4.2.3 Estimation Formula 5 (Transformed Fit Log to Reciprocal)

The probability of no gap is transformed to logarithmic form, and k is transformed to reciprocal form in this method to formulate the relation between them. Equation 17 was created by using this transformation with 0.91 R-square value. The output values of this formula were given in Figure 4.18.

$$Log(P[no\ gap]) = 0.03213 - 0.10206\,(\frac{1}{k}) \tag{17}$$

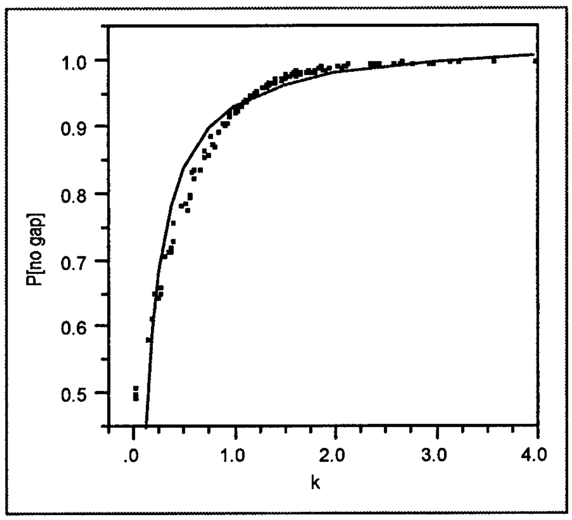


Figure 4.18 P[no gap] from formula 5

4.4.2.4 Comparison of Formula 3, Formula 4 and Formula 5

Formula 3 fits our simulation results better than Formula 4 and Formula 5 according to the R-square values associated with them. As a matter of fact, Formula 3 catches the trend of the P[no gap] very well for the k values less than three. The significant deviation of this formula begins after this point as shown in Figure 4.16.

Formula 4 and Formula 5 also have significant deviations for different values of k. In the case of comparison between Formula 4 and Formula 5, the area where the optimization most likely happens in real world applications plays a key role. As shown in Figure 4.15, the significant changes at the trade-off between the control time and the acceptable P[no gap] happens between one and two of the k values. Therefore, the optimization most likely takes place in this area in the real world applications. Formula 4 gives better performance in this area as shown in Figure 4.19 and Figure 4.20.

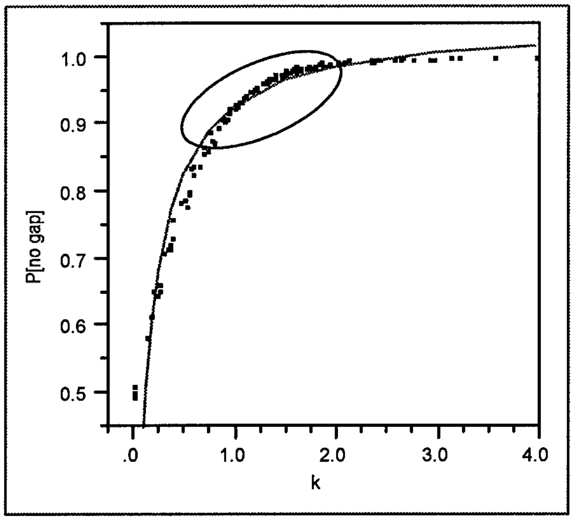


Figure 4.19 Estimation at trade-off area (Formula 4)

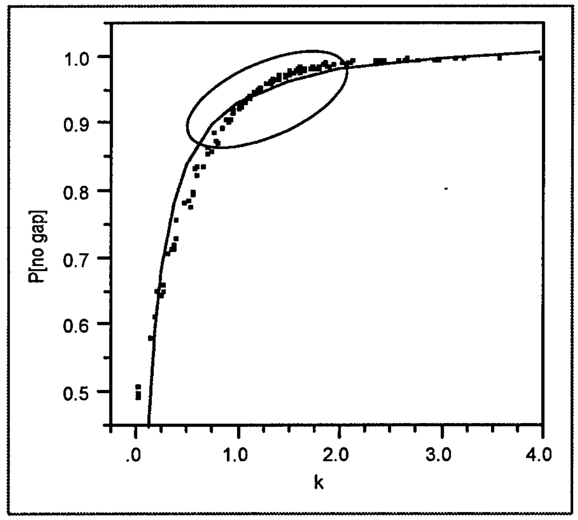


Figure 4.20 Estimation at trade-off area (Formula 5)

The difference between the P[no gap] values for k values two and three is too small (approximately 0.004) based on the simulation results. Therefore, a k value of three or more is too inefficient to use in real world applications: it increases end-to-end delay without noticeably improving P[no gap]. Thus, the deviation in the formula 3 for k values greater than three does not hurt the estimation in most cases. For this reason, Formula 3 is assessed to the best in this comparison.

4.4.3 Overall Comparison and Precision of Formulas

In the overall comparison: Formula 1 has advantage of computational simplicity over Formula 3 but it can not be used for all possible values of P[no gap]. On the other hand, Formula

3 includes all possible values of the P[no gap] but has computational complexity. Therefore, if desired P[no gap] is between 0.93 and 0.98, Formula 1 is recommended to estimate P[no gap]. If not, Formula 3 is recommended to estimate P[no gap] in this effort.

The simulation results for same k values show that the difference at the five-digit floating point values of P[no gap] begins at third decimal place. Therefore, the probability of no gap value should be computed using at least 3-digit floating-point arithmetic. In worst case, the error caused by decimal places that come after third is 0.0009 and this value is acceptable for our estimation. Thus, Formula 3 can be rewritten in a less computationally complex form for 3-digit floating-point arithmetic as given Equation 18.

$$P[no\ gap] = 0.501 + 0.781 * k - 0.464 * k^2 + 0.121 * k^3 - 0.0116 * k^4 \tag{18}$$

4.5 Delay

The end-to-end delay of a packet includes three main components that are codec delay, network delay and buffer delay, as discussed in Section 2.4.2 and given in Equation 19.

$$\text{Delay}_{\text{end-to-end}} = \text{Delay}_{\text{codec}} + \text{Delay}_{\text{network}} + \text{Delay}_{\text{buffer}} \tag{19}$$

The control time or buffer delay analyzed in Section 4.4 is one of these main components and can be controlled by user. The codec delay, as another delay component, is a relatively fixed amount of time associated with the codec used in the sender and receiver. The delays of the different ITU recommended codecs can be found in Table 2.1. The network delay, as a final component, is uncontrollable and depends on the network condition. Delay due to the transport network is nondeterministic in nature.

The control time can be found by submitting the control time in estimation Formula 1 or Formula 3 for a given P[no gap] and the standard deviation of the delay distribution. If one-way network delay of the packets is determined by using the common divide by two assumption (ITI receiving scheme uses this assumption), the end-to-end delay can be expressed as a function of the control time and the mean of the delay distribution as given in Equation 20.

$$Delay_{\text{end-to-end}} = Delay_{\text{codec}} + \frac{mean}{2} + Control\ Time \tag{20}$$

The codec delay is relatively fixed. The control time variable in this equation can be determined by submitting the control time in Equation 12 or Equation 18. For example, the control time can be written as given in Equation 21 by submitting in Equation 12 for P[no gap] between 0.93 and 0.98.

$$Control\ Time = \frac{-0.345 + \sqrt{0.119 + 0.392 * (0.682 - P[no\ gap])}}{-0.196} * SD \tag{21}$$

Thus, the end-to-end delay can be rewritten as follows for P[no gap] between 0.93 and 0.98.

$$Delay_{end\text{-}to\text{-}end} = Delay_{codec} + \frac{mean}{2} + \frac{-0.345 + \sqrt{0.119 + 0.392 * (0.682 - P[no\ gap])}}{-0.196} * SD$$

4.6 Packet Loss

The "probability of no gap" system performance metric discussed so far was the probability of no gaps caused by late packets at the receiver. If we consider the probability of no gap of the entire end-to-end system (speaker-to-speaker), we have to take packet losses of the network into account, because every packet lost in the network also causes gaps at the receiver. Therefore, total end-to-end system probability of no gap can be written as given Equation 21.

$$P_{end\text{-}to\text{-}end}[no\ gap] = (\ 1 - P_{network}[loss]\ ) * P_{late}[no\ gap] \tag{22}$$

This equation can be rewritten in terms of the control time and the standard deviation, e.g., by replacing P[no gap] from Equations 12 or Equation 18.

The upper limit of the system probability of gap is linked to the human perception. But it should be considered that as the network packet loss rate increases, the probability of no gap of the system decreases proportionally as seen in Equation 22.

4.7 Evaluation of Developed Equations

Two groups of equations are developed from the simulation results and their analysis. The first group of equations, Formula 1 through Formula 5, estimate the P[no gap] performance metric for given control time and standard deviation values. Formula 1 was recommended among them since it can be used for all values of k and has better performance. The second group of equations, Equation 20 and Equation 22 express quality of service metrics in terms of QoS factors such as network delay, delay standard deviation and packet loss rate.

A packet voice communication system, now, can be adjusted to a desired quality of service by adjusting the control time variable. If Equations 20 and 22 are solved for acceptable or desired end-to-end delay and end-to-end probability of no gap QoS metrics, the control time value required for requested quality of service can be determined. In Chapter 5 these equations will be used, in a sample system condition, to set the control time value for an acceptable and desired quality of service.

4.8 Summary

This chapter showed the analysis results. First, delay distribution functions were determined. Then, simulation outputs and their points needing to be examined were given. The simulation results were analyzed and two groups of equations were developed. Probability of no gap was represented as a function of the control time and standard deviation in the first group of equations. While Equation 11 was recommended to calculate the value of P[no gap] for k values between 1 and 1.6, Equation 15, or the less precise form of it, Equation 18, was recommended for all possible values of k. Then, QoS metrics were represented as a function of QoS factors in the second group of equations. The quality of service offered by a system can be determined for given operating conditions according to these mathematical expressions. Furthermore, quality of service can be adjusted to the desired level by using the control time variable in these equations.

5 Conclusion and Future Recommendations

5.1 Introduction

The objective of this research was to explore the trade-off between quality of service (QoS) factors and the optimum combination(s) of these factors for packet voice communications over IP networks. This chapter concludes the research effort. An overview of the research effort is first presented. This overview summaries the major topics in each of the previous chapters. Next, the conclusions are presented. The quality of service metrics is estimated for a given sample operating conditions per equations developed in Chapter 4. Then, the system control time variable is estimated for a requested quality of service. Following the conclusion, recommendations for future research are presented. Finally, an overall summary is provided.

5.2 Overview

Chapter 1 began with a short background of real-time packet voice communication over IP networks. Next, there were a definition of the problem and the scope of this effort. The scope of the research was narrowed to explore the trade-offs among QoS factors.

The review of the QoS factors was accomplished in Chapter 2. The literature review found three different delay estimation methods and receiving schemas. The blind delay estimation, the first method, was not appropriate for long-haul networks or large delay variation. The absolute timing, the third method, requires synchronized clocks of the sender and receiver. Therefore the round-trip estimation, second method, and corresponding incomplete timing information (ITI) receiving schema have been chosen to be implement in the simulation model. Three VoIP configurations were explained to implement VoIP according to the type of the sender and the receiver equipment and connections of them. The PC-to-PC configuration was selected for this effort since it does not employ Internet/Intranet telephony gateway. The communication network types and protocols used in VoIP and measurements were also reviewed in this chapter.

Chapter 3 presented the methodology used to analyze the impact of the QoS factors on the packet voice communication. First, the experimental network measurement structure and results were provided. These measurements were accomplished in order to provide real-world sampling of the network characteristics for short and long-haul networks. Next, the simulation model was presented. The performance metrics, input parameters, assumptions, and validation and verification of the simulation model and measurements were also stated in this chapter.

The simulation results were presented in Chapter 4. The probability of no gap performance metric was represented as function of control time and standard deviation of delay distribution. Mathematical expressions were developed to estimate P[no gap] and end-to-end delay performance metrics of the system for given operating conditions.

5.3 Conclusion

The P[no gap] and end-to-end delay system performance metrics can be estimated by the equations developed in Chapter 4. The system can also be adjusted for a desired quality of service, if at all possible. The control time is the key factor to set the system for acceptable or requested quality of service since it is variable and can be controlled by the user. However, there is an opposing relationship between control time and the performance metrics for any network condition as depicted in Figure 5.1

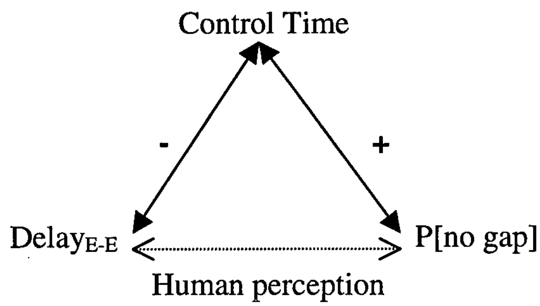


Figure 5.1 Interaction between performance metrics and control time

Increasing the control time has positive effect on the P[no gap] while producing a negative effect on the end-to-end delay. The limits of these performance metrics, P[no gap] and delay, depend on the human perception.

Therefore, the trade-off analysis for quality of service also involves human perception. Quantifying the performance factors is difficult since individual human users may have different tolerances for the delay and gap probability. These tolerances will also vary with the application. Several studies on users provide information for acceptable delay and probability of gap values. These studies indicate that an end-to-end delay up to 600 ms and a gap probability up to 0.1 can be tolerated [KuD94] [BoG98] [RaR92]. Thus, Equations 23 and 24 can be written for an acceptable quality of service by using these performance metric limits and the equations developed in Chapter 4.

$$Delay_{\text{end-to-end}} = Delay_{\text{codec}} + \frac{mean}{2} + Control\ Time \leq 600 \qquad (23)$$

and

$$P_{end\text{-}to\text{-}end}[no\ gap] = (\ 1\text{-}P_{network}[loss]\ ) * P_{late}[no\ gap] \geq 0.9 \qquad (24)$$

Where;

- $Delay_{codec}$ is a fixed amount of time associated with encoding algorithm used in the system. The value of $Delay_{codec}$ is one millisecond for ADPCM (G.722) encoding algorithm used in this effort.
- Mean is the mean value of the network delay distribution.
- $P_{network}$[loss] is the packet loss probability of the network.
- $P_{receiver}$[no gap] is the probability of no gap due to the late packets. The value of $P_{receiver}$[no gap] can be estimated by Equation 18.
- The control time is the system variable controlled by the user or receiver.

These equations are used for a path characterized by 230 ms mean round-trip delay, 29 ms delay standard deviation and 0.015 network packet loss probability in order to set an example.

Mean	= 230 ms.		
Standard deviation	= 29 ms.	$\Rightarrow$	$26.95 \leq CT \leq 484$
$P_{network}$[no gap]	= 0.015		

The value of the control time can be found by Equation 23 and Equation 24 for an acceptable quality of service.

- The solution of Equation 24 gives the minimum value of P_{late}[no gap]

$$(1-0.015) * P_{late}[no\ gap] \geq 0.9 \qquad \Rightarrow \qquad P_{late}[no\ gap] \geq 0.913$$

- The solution of Equation 15 for the minimum value of P_{late}[no gap] gives that the value of k is equal to 0.929. Therefore, the minimum value of control time is equal to 26.95 ms.

$$Control\ Time = k * Standard\ Deviation = 0.929 * 29 = 26.95$$

- The solution of Equation 23 gives upper limit of the control time

$$1 + \frac{230}{2} + \mathit{Control\ Time} \leq 600 \Rightarrow \text{Control Time} \leq 484$$

- Therefore, $26.95 \leq \text{Control Time} \leq 484$

These equations also can be used to determine the control time value for a desired quality of service by simply changing end-to-end delay and end-to-end probability of gap limits. For example, in order to get a service that offers 0.96 end-to-end probability of no gap with end-to-end delay values less then 400 ms, the value of the control time can be found as follows.

- The solution of Equation 24 gives the minimum value of P_{late}[no gap]

$$(\ 1\text{-}0.015\ ) * P_{late}[no\ gap] \geq 0.96 \qquad \Rightarrow \qquad P_{late}[no\ gap] \geq 0.974$$

- The minimum value of the control time can be found by Equation 21

$$\mathit{Control\ Time} = \frac{-0.345 + \sqrt{0.119 + 0.392 * (0.682 - 0.974)}}{-0.196} * 29 = 41$$

- The solution of Equation 23 gives upper limit of the control time

$$1 + \frac{230}{2} + \mathit{Control\ Time} \leq 400 \Rightarrow \text{Control Time} \leq 284$$

- Therefore, $41 \leq \text{Control Time} \leq 284$

However, the trade-off analysis discussed in Section 4.4 showed that setting the control time value larger than twice of the standard deviation is not efficient. The minor improvements at the P[no gap] cost larger end-to-end delays. Therefore, it is recommended that the value of control time should be between minimum required value and twice of the standard deviation for an efficient and acceptable operation. The control time should be between 41 and 58 ms for previous example according to the recommendation.

5.4 Future Recommendation

This effort analyzed the trade-offs for quality of service in the real-time packet voice communication system that implements a round-trip time estimation method and incomplete timing information receiving schema. Some possible areas of research are as follows:

1. The control time can be changed adaptively as the call progress instead of using fixed control time for all talkspurts. Thus, better response to the variable network delays can be achieved. A new algorithm can be presented to change the control time adaptively. The equations developed in this effort can be used in this algorithm to determine the control time value for changing network conditions. The performance can be compared with the simulation results provided in this research.

2. Round-trip delay estimation was used in this effort. Round-trip delay estimation can be adapted by making use of additional information as the call progresses. The adaptive change in the delay estimation can be based either on the previous delay estimates or on the basis of repeated round-trip measurements. For example, the estimated delay value can be changed to the mean of the last predetermined number of estimated delays. This kind of adaptation mechanism can be studied to obtain improvement of the

performance over the single estimation method. Additionally, new adaptive algorithms can be presented.

3. The Gamma distribution was used in this effort to represent the packet delay distribution since it has best or near to the best goodness of fit test results according the data obtained sample measurements. Different delay distributions can be found from various measurements. Therefore, the trade-off among QoS factors also can be studied for different delay distributions by using the same simulation model with a change of the network component.

5.5 Summary

This chapter concludes the research effort. First, an overview of the previous chapters was provided. Next, new equations were provided to determine the control time for an acceptable or desired quality of service. These equations were used to obtain the control time value for a sample operating condition. Then, some possible areas of research were given.

Appendix A

Probability Distribution Types Provided by Crystal Ball

1. Binomial
2. Beta
3. Custom
4. Exponential
5. Extreme Value
6. Gamma
7. Geometric
8. Hypergeometric
9. Logistic
10. Lognormal
11. Negative Binomial
12. Normal
13. Pareto
14. Poisson
15. Triangular
16. Uniform
17. Weibull

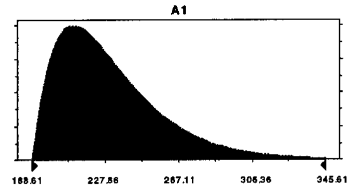
Appendix B

1- AFIT-Air University Path

Gamma distribution with parameters:

Location	188.61
Scale	19.99
Shape	2.10167

Selected range is from 188.61 to +Infinity

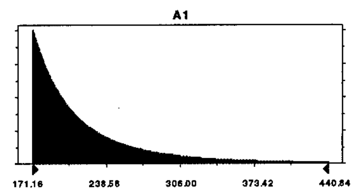


2- AFIT-Bolling AFB. Path

a-) Pareto distribution with parameters:

Location'	171.16
Shape	3.867795

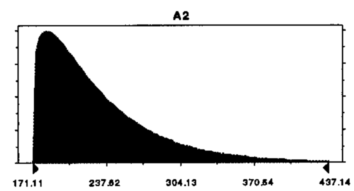
Selected range is from 171.16 to +Infinity



b-) Gamma distribution with parameters:

Location	171.11
Scale	47.25
Shape	1.250574

Selected range is from 171.11 to +Infinity

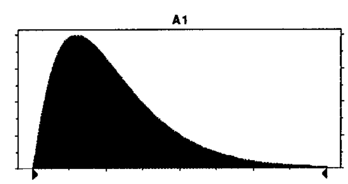


3- AFIT-Hanscom AFB. Path

Gamma distribution with parameters:

Location	30.56
Scale	5.72
Shape	2.18254

Selected range is from 30.56 to +Infinity

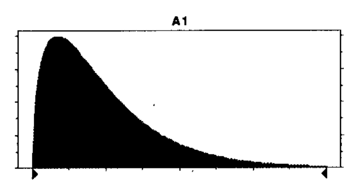


4- AFIT-Lackland AFB. Path

Gamma distribution with parameters:

Location	156.69
Scale	34.45
Shape	1.55167

Selected range is from 156.69 to +Infinity

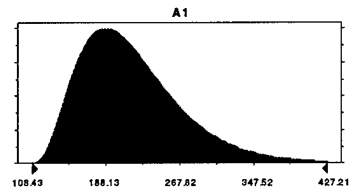


5- AFIT-Mc Chord AFB. Path

Gamma distribution with parameters:

Location	108.43
Scale	28.28
Shape	3.793618

Selected range is from 108.43 to +Infinity

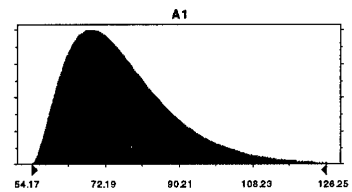


6- AFIT-MacDill AFB. Path

Gamma distribution with parameters:

Location	54.17
Scale	7.52
Shape	2.913058

Selected range is from 54.17 to +Infinity

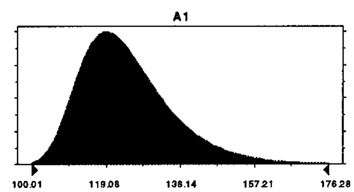


7- AFIT-Minot AFB. Path

a-) Extreme Value distribution with parameters:

Mode	119.08
Scale	9.53

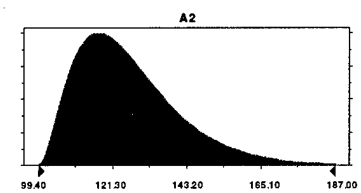
Selected range is from -Infinity to +Infinity



b-) Gamma distribution with parameters:

Location	99.40
Scale	9.07
Shape	2.950844

Selected range is from 99.40 to +Infinity

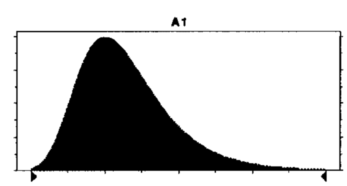


8- AFIT-Scott AFB. Path

a-) Extreme Value distribution with parameters:

Mode	173.64
Scale	4.60

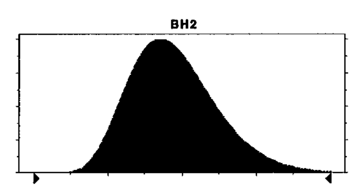
Selected range is from -Infinity to +Infinity



b-) Gamma distribution with parameters:

Location	152.52
Scale	2.36
Shape	10.2834

Selected range is from 152.52 to +Infinity

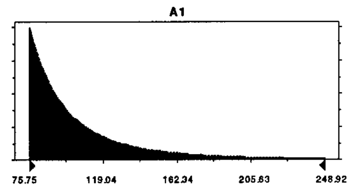


9- AFIT-Travis AFB. Path

a-) Pareto distribution with parameters:

Location'	75.75
Shape	2.870938

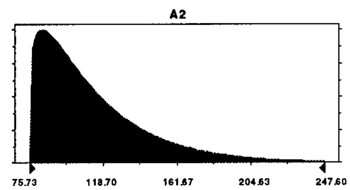
Selected range is from 75.75 to +Infinity



b-) Gamma distribution with parameters:

Location	75.73
Scale	30.92
Shape	1.228782

Selected range is from 75.73 to +Infinity

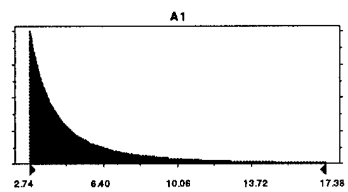


10 AFIT-WPAFB Path

a-) Pareto distribution with parameters:

Location'	2.74
Shape	1.4952

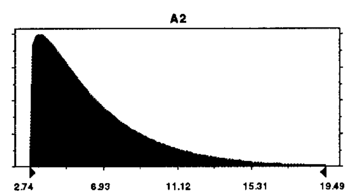
Selected range is from 2.74 to +Infinity



b-) Gamma distribution with parameters:

Location	2.74
Scale	3.11
Shape	1.175396

Selected range is from 2.74 to +Infinity



Appendix C

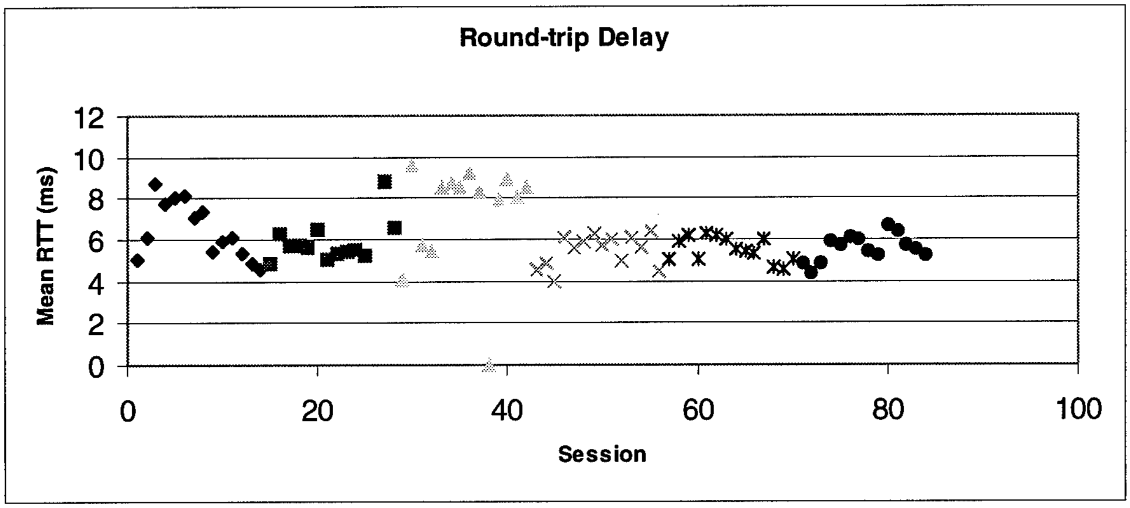


Figure C- 1 Mean Delays for AFIT-WPAFB Path

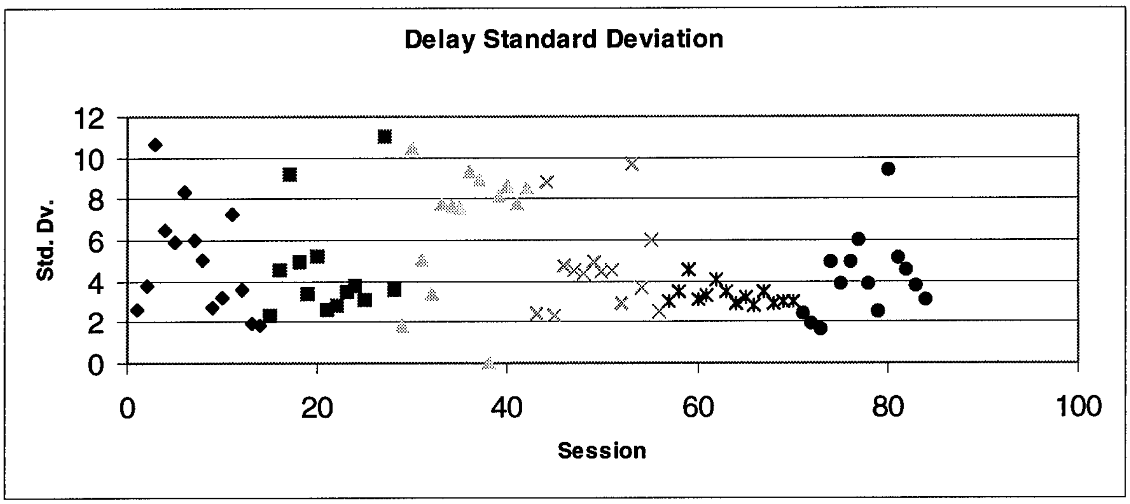


Figure C- 2 Mean Standard Deviations of Delay for AFIT-WPAFB Path

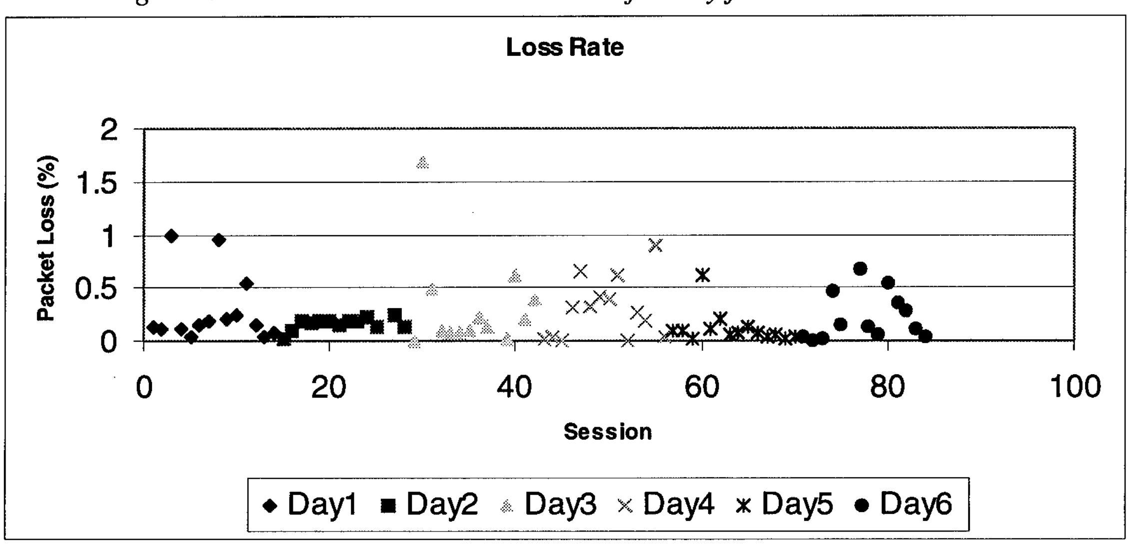


Figure C- 3 Packet Loss Rates for AFIT-WPAFB Path

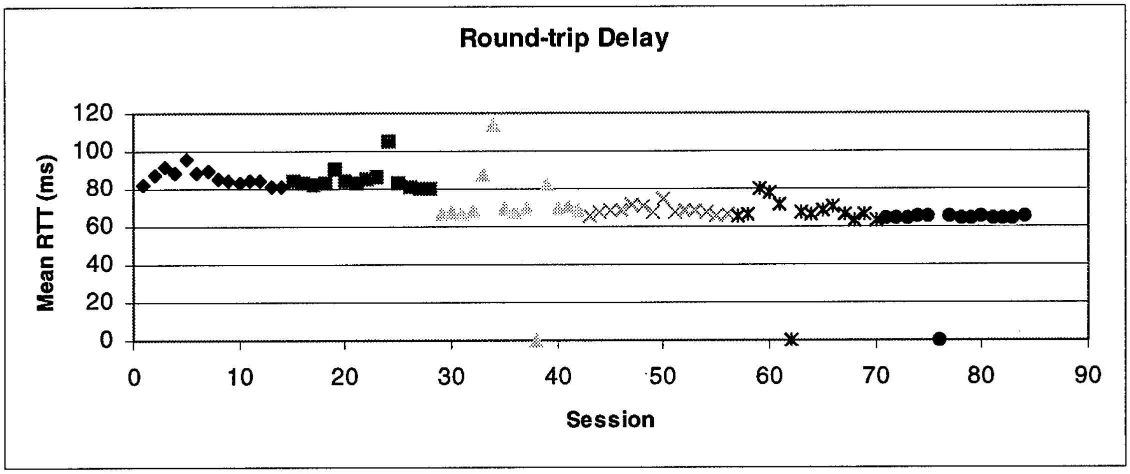


Figure C- 4 Mean Delays for AFIT-MacDill AFB. Path

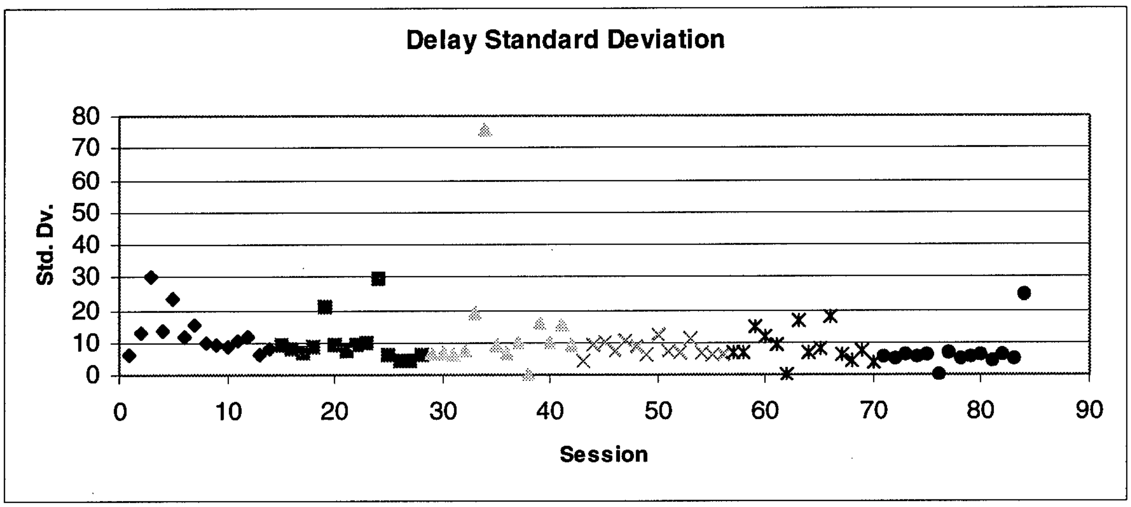


Figure C- 5 Mean Standard Deviations of Delay for AFIT- MacDill AFB. Path

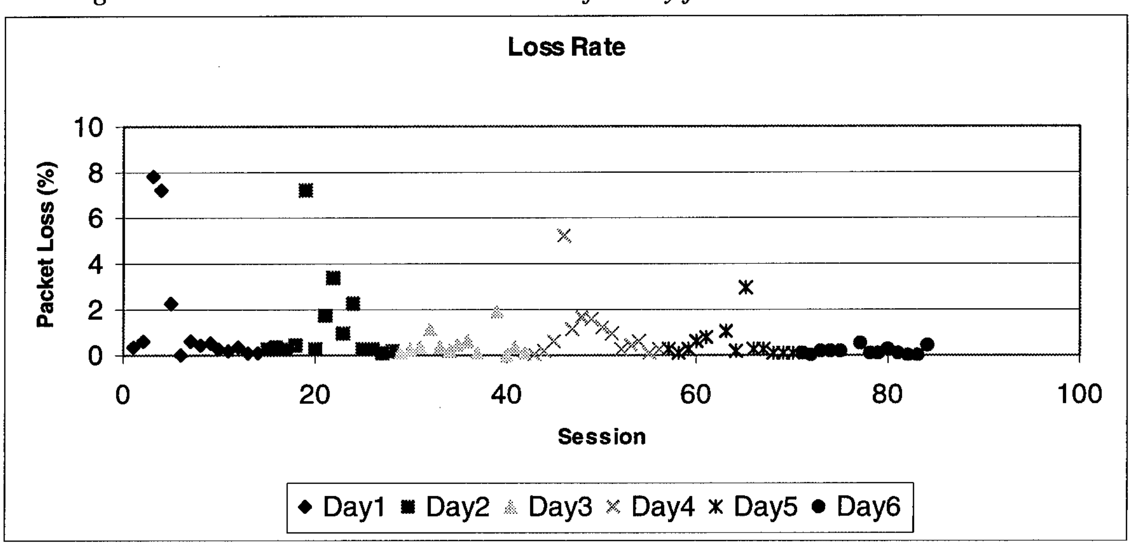


Figure C- 6 Packet Loss Rates for AFIT- MacDill AFB. Path

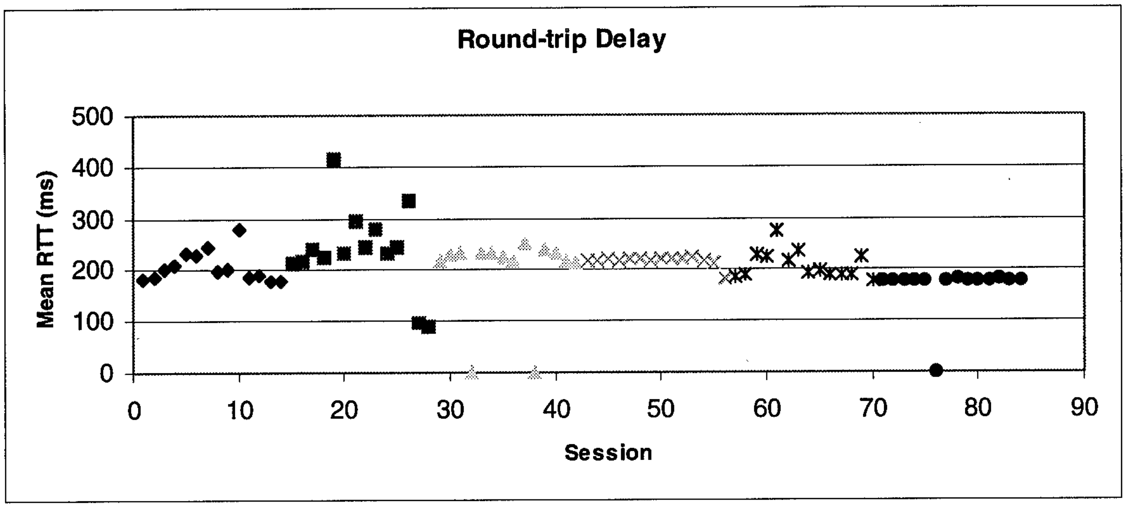


Figure C- 7 Mean Delays for AFIT-Lackland AFB. Path

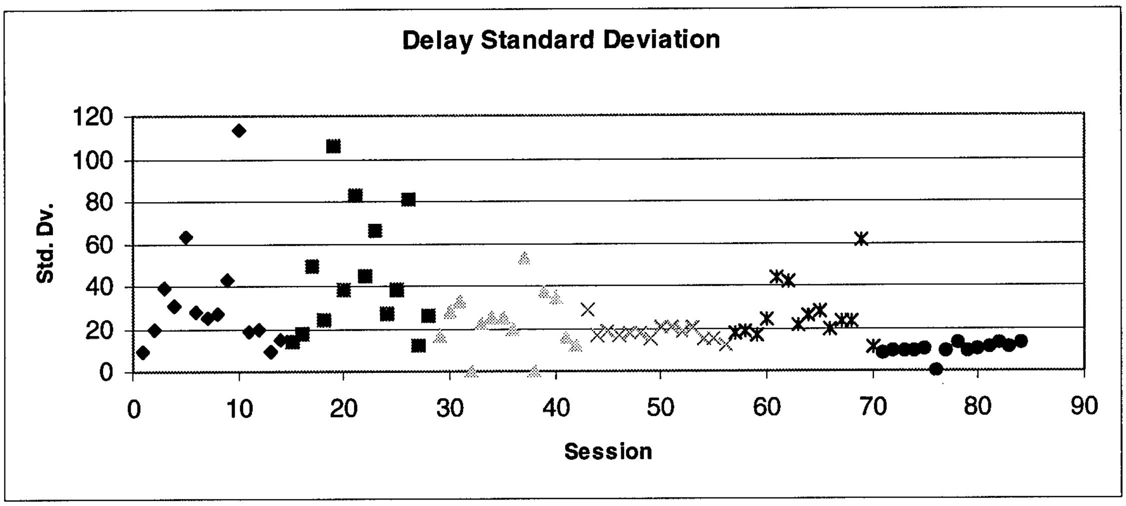


Figure C- 8 Mean Standard Deviations of Delay for AFIT- Lackland AFB. Path

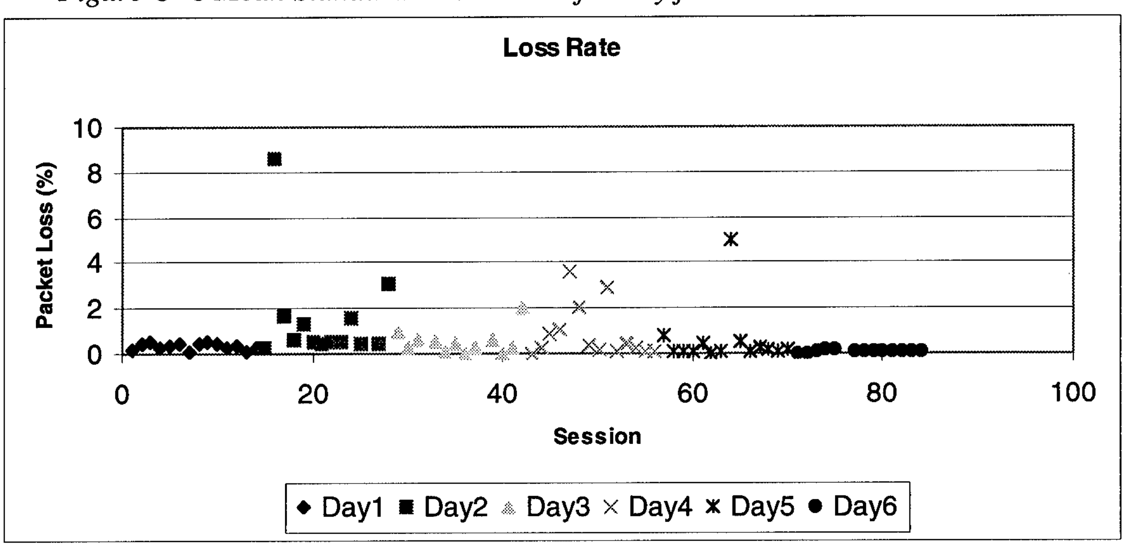


Figure C- 9 Packet Loss Rates for AFIT- Lackland AFB. Path

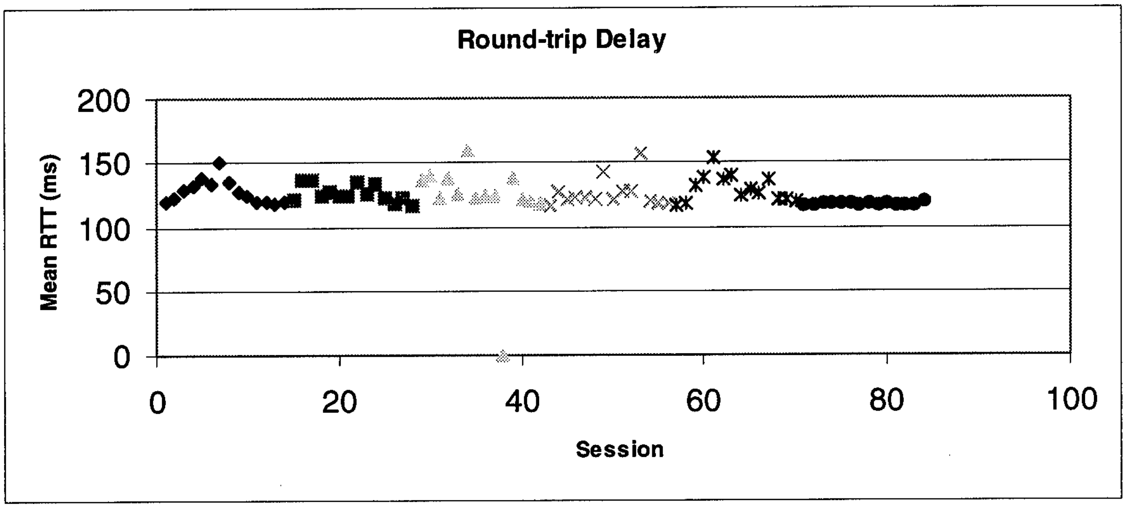


Figure C- 10 Mean Delays for AFIT-Minot AFB. Path

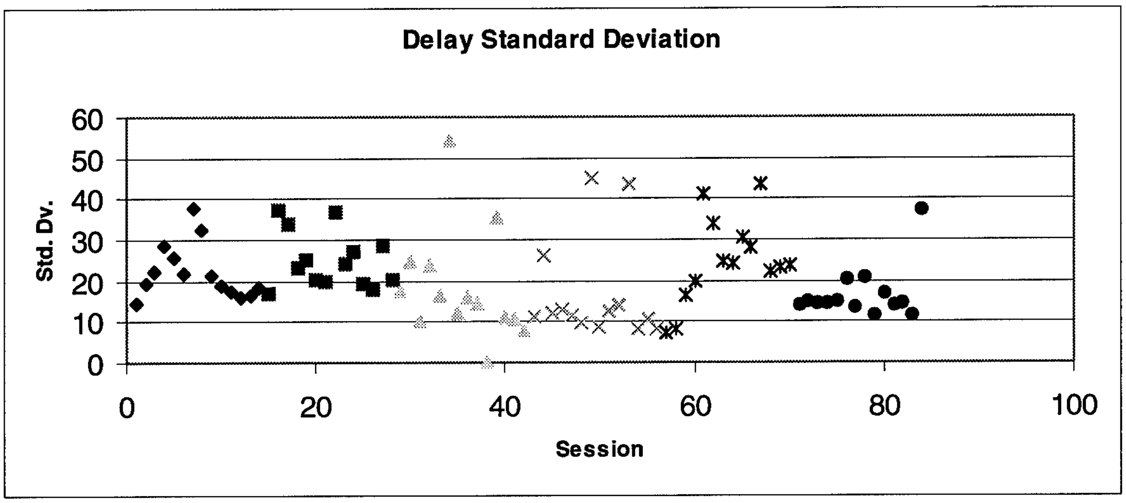


Figure C- 11 Mean Standard Deviations of Delay for AFIT- Minot AFB. Path

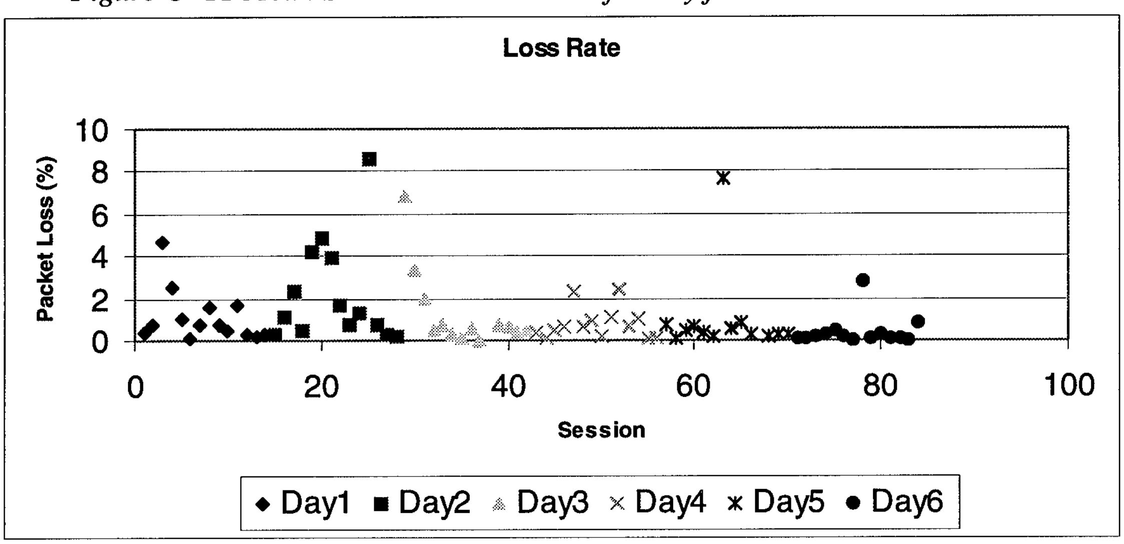


Figure C- 12 Packet Loss Rates for AFIT- Minot AFB. Path

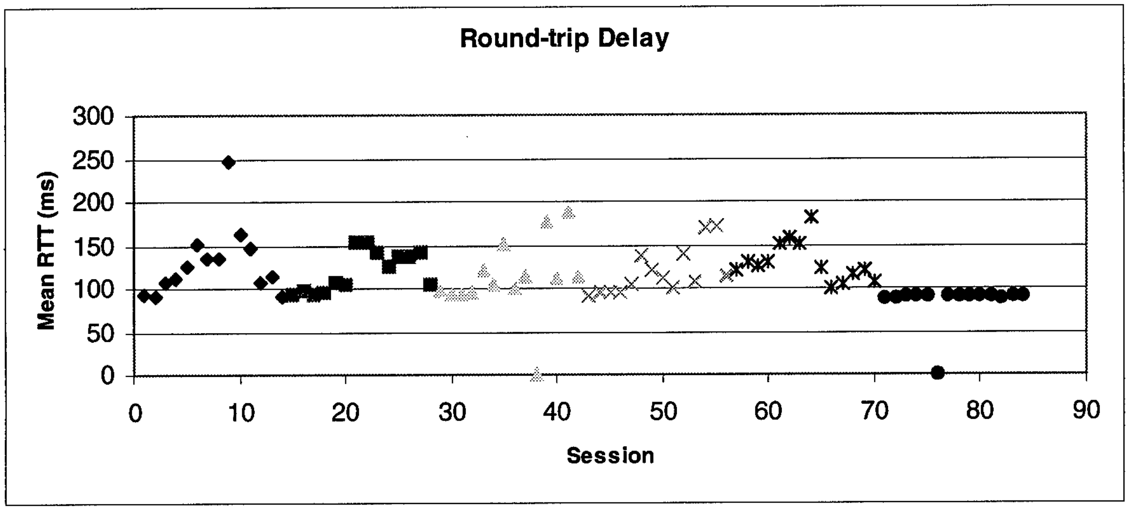


Figure C- 13 Mean Delays for AFIT-Travis AFB. Path

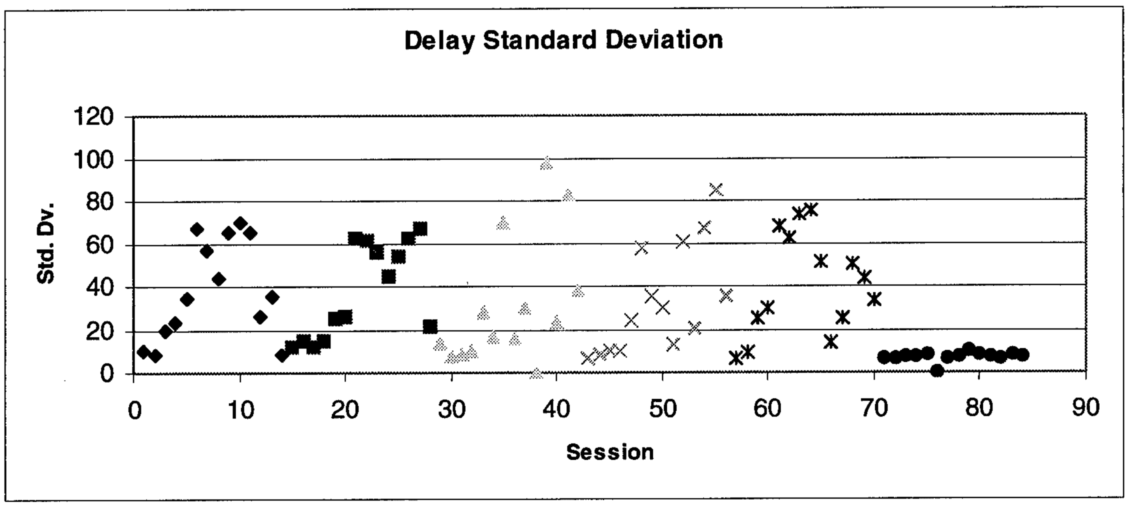


Figure C- 14 Mean Standard Deviations of Delay for AFIT- Travis AFB. Path

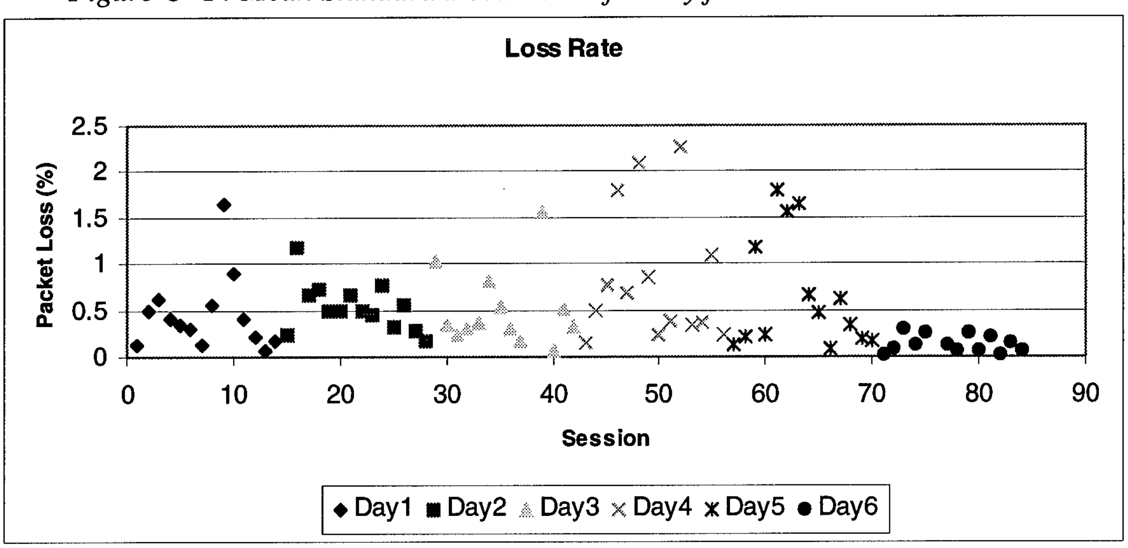


Figure C- 15 Packet Loss Rates for AFIT- Travis AFB. Path

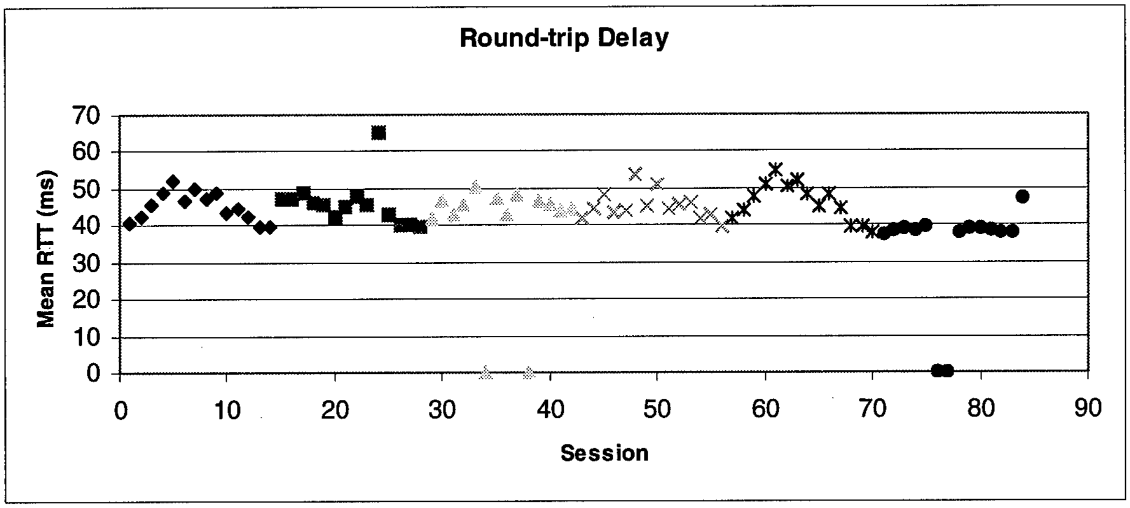


Figure C- 16 Mean Delays for AFIT-Hanscom AFB. Path

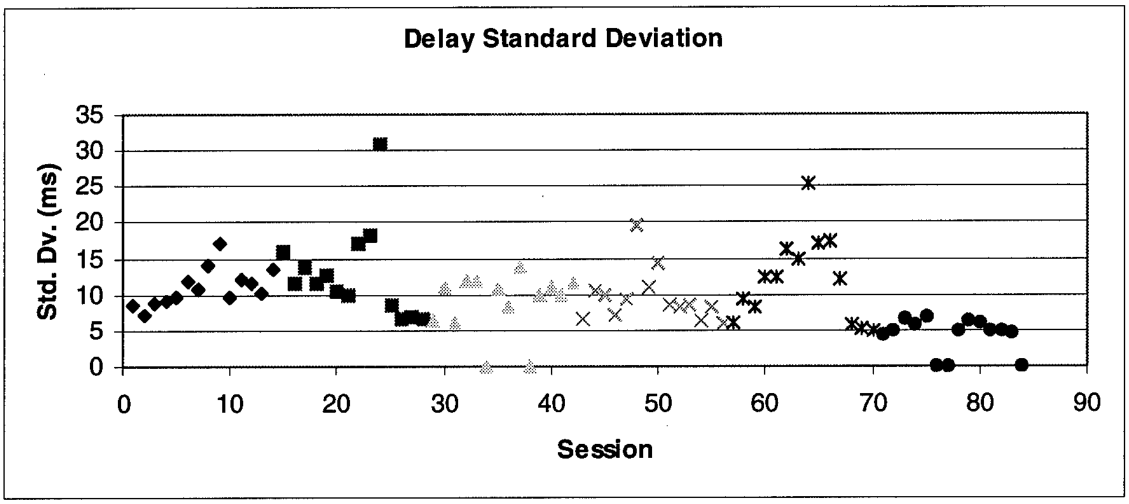


Figure C- 17 Mean Standard Deviations of Delay for AFIT- Hanscom AFB. Path

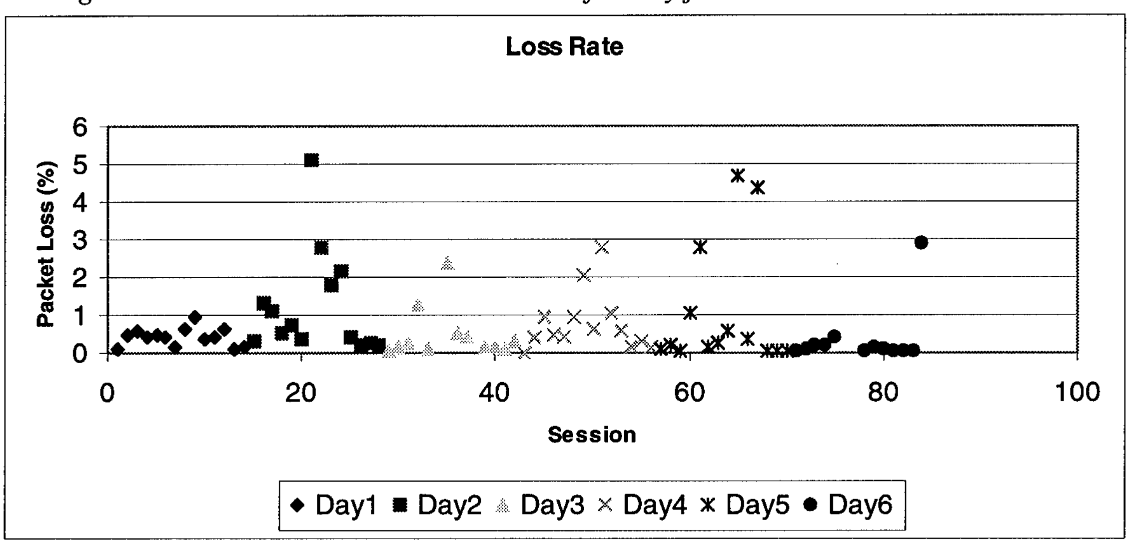


Figure C- 18 Packet Loss Rates for AFIT- Hanscom AFB. Path

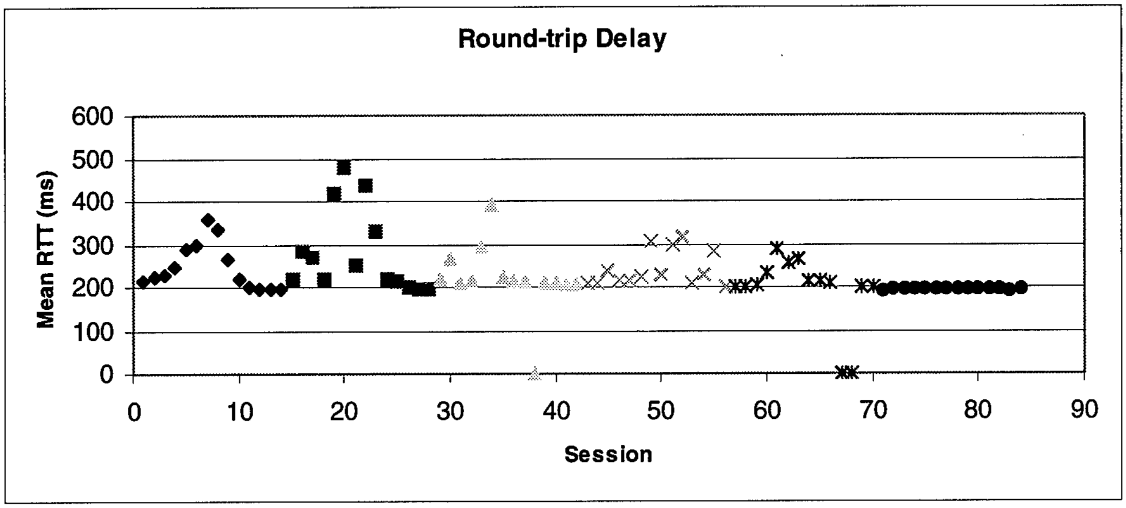


*Figure C- 19 Mean Delays for AFIT-*Bolling *AFB. Path*

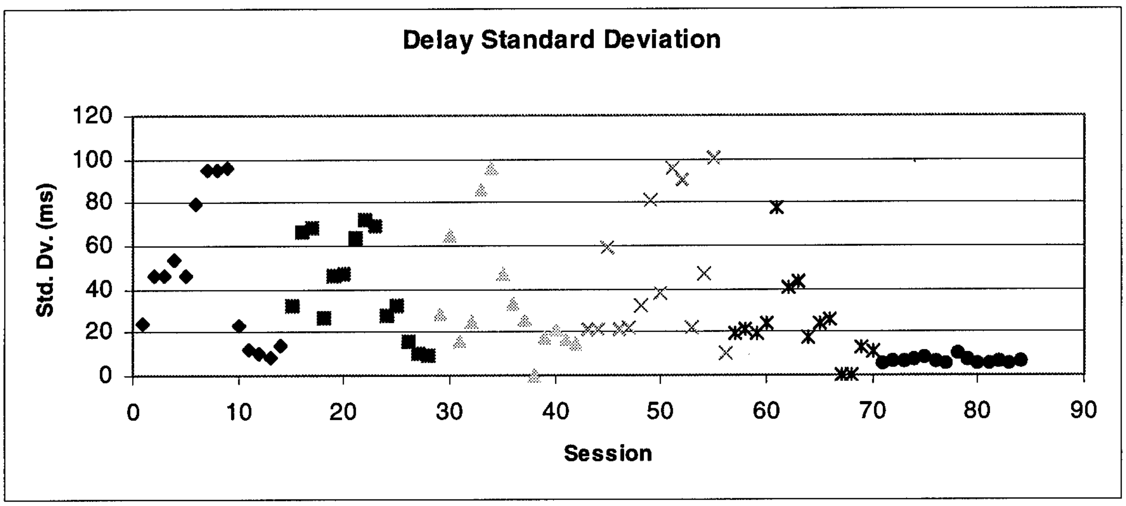


Figure C- 20 Mean Standard Deviations of Delay for AFIT- Bolling AFB. Path

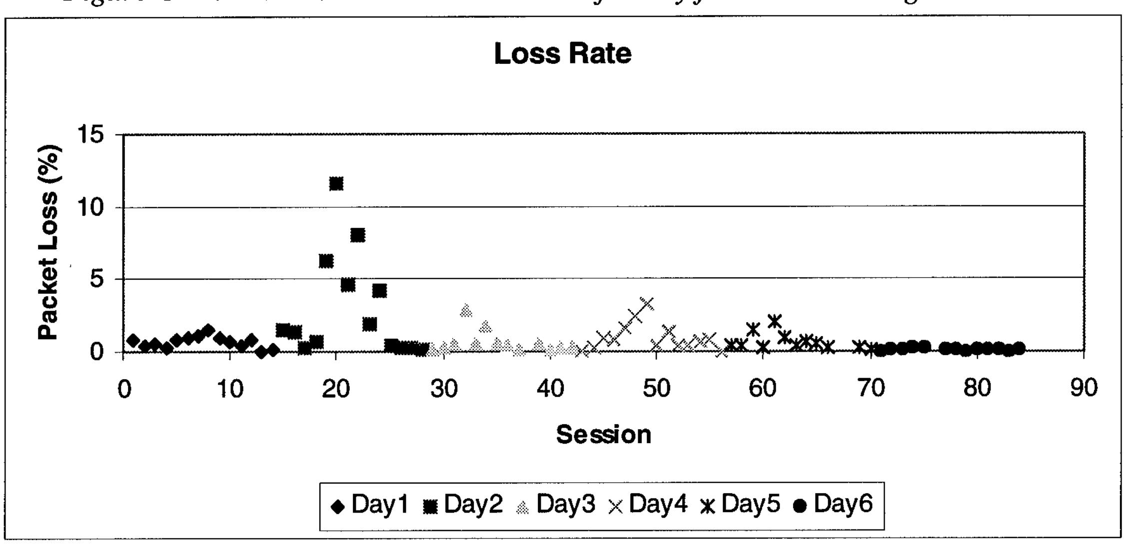


Figure C- 21 Packet Loss Rates for AFIT- Bolling AFB. Path

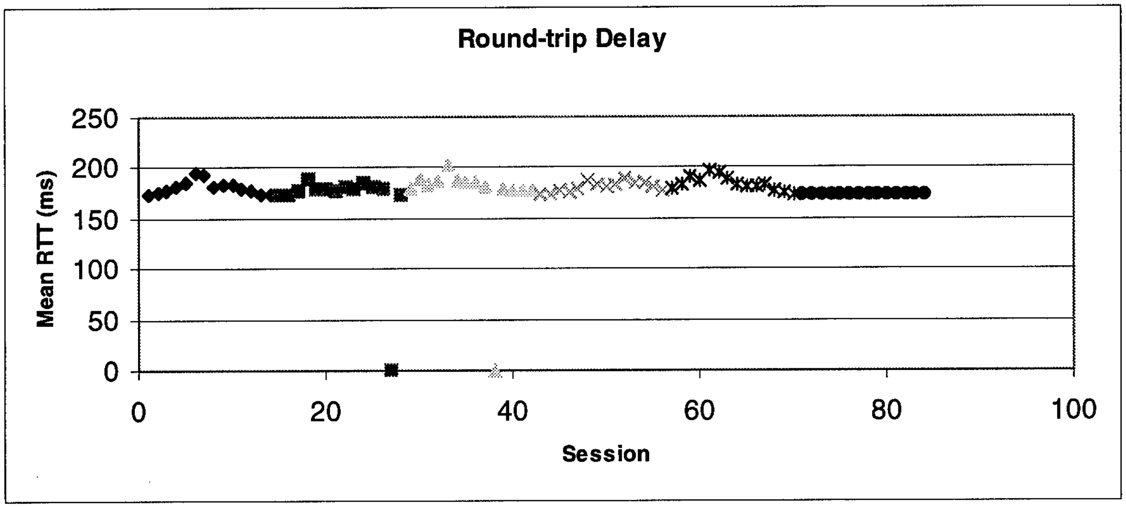


Figure C- 22 Mean Delays for AFIT-Scott AFB. Path

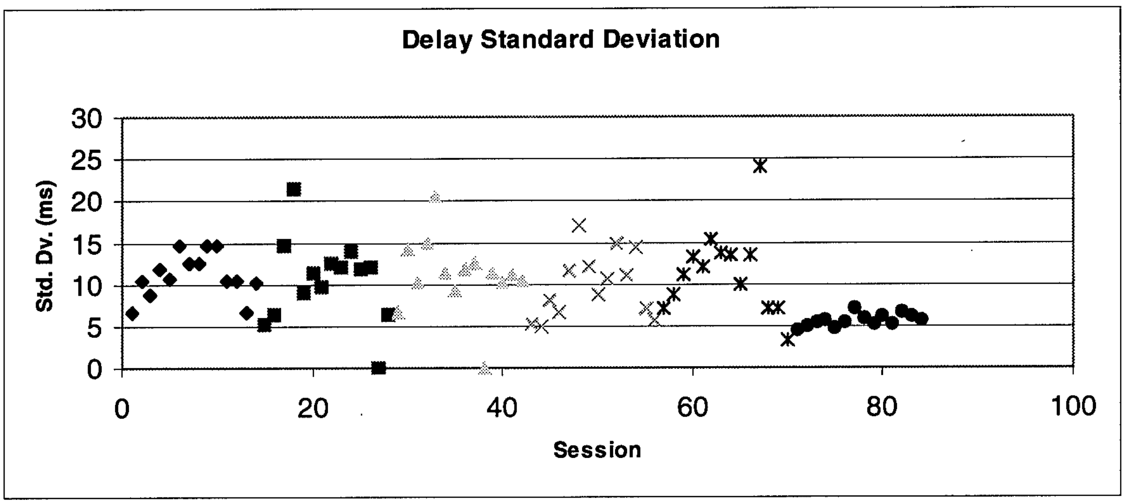


Figure C- 23 Mean Standard Deviations of Delay for AFIT- Scott AFB. Path

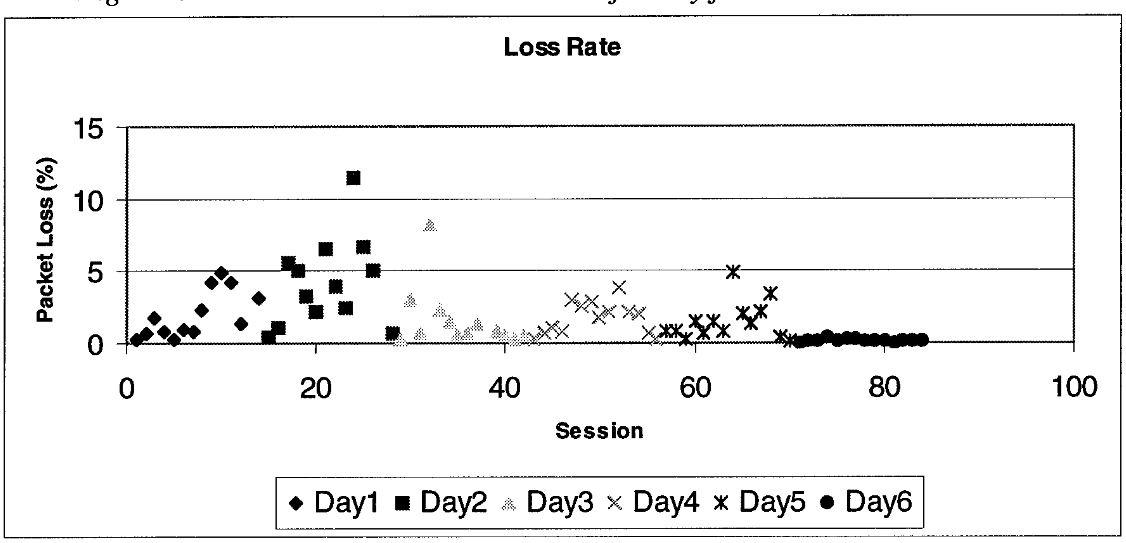


Figure C- 24 Packet Loss Rates for AFIT- Scott AFB. Path

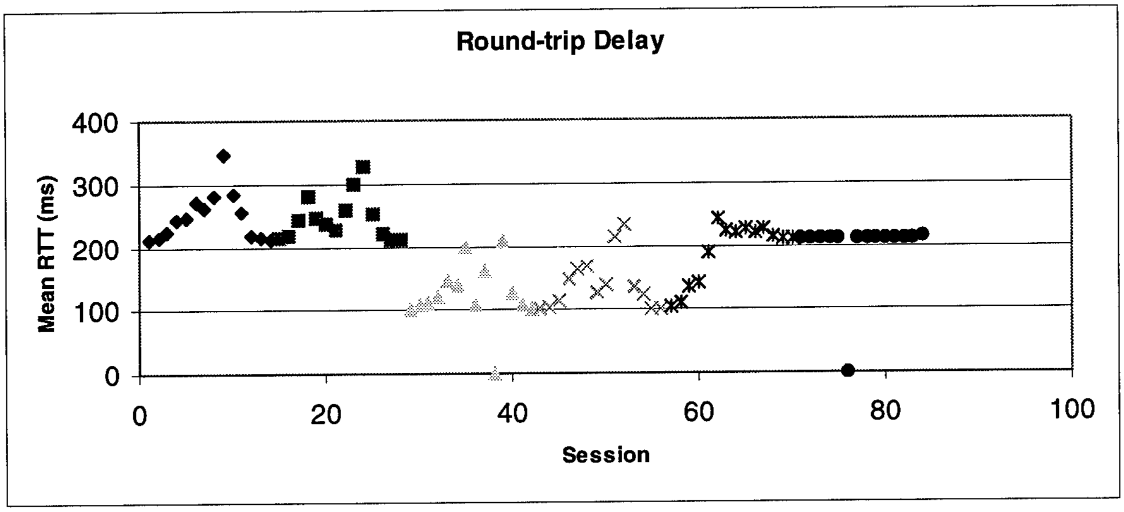


Figure C- 25 Mean Delays for AFIT-Air University Path

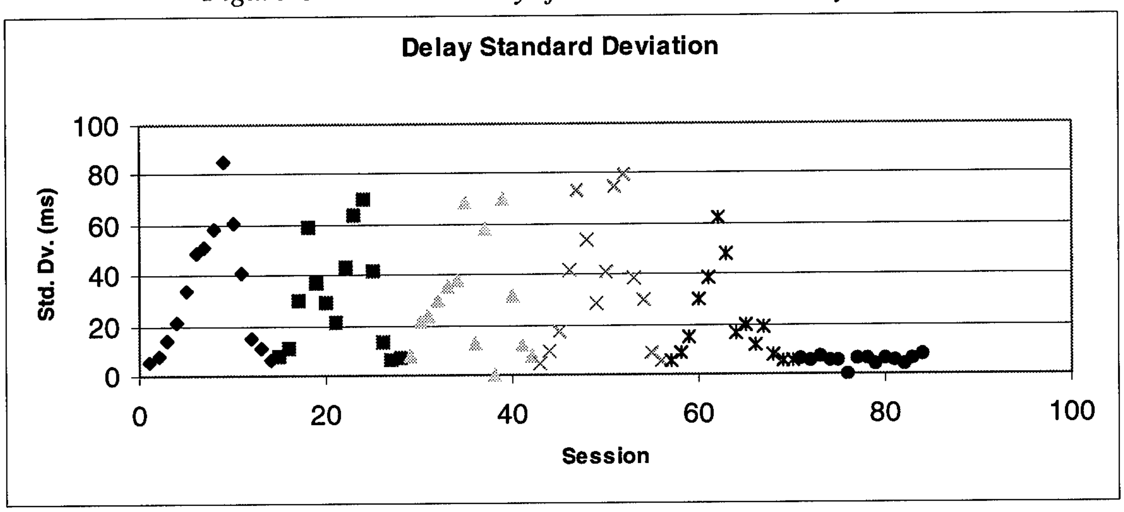


Figure C- 26 Mean Standard Deviations of Delay for AFIT- Air University Path

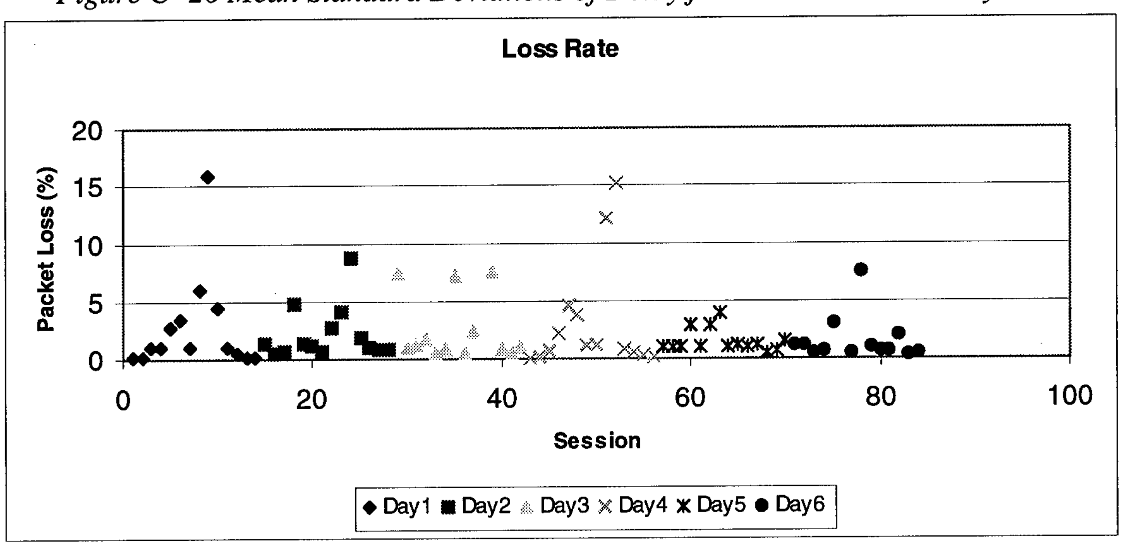


Figure C- 27 Packet Loss Rates for AFIT- Air University Path

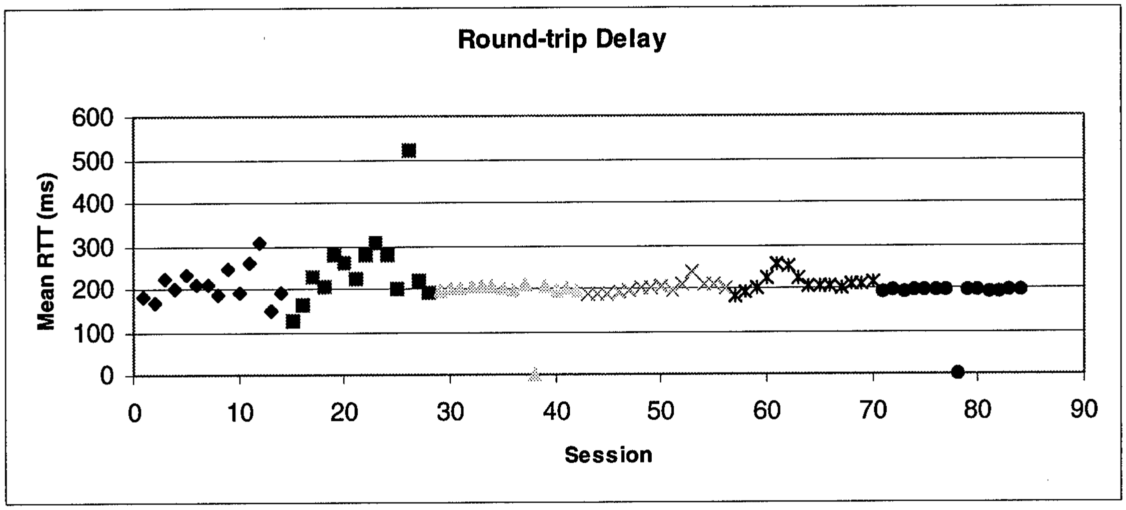


Figure C- 28 Mean Delays for AFIT-Mc Chord AFB. Path

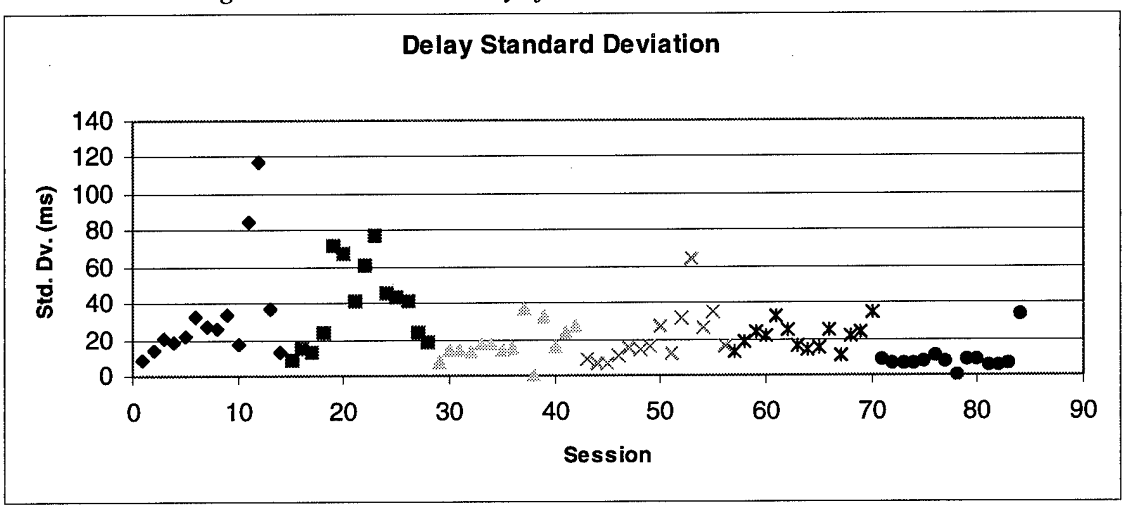


Figure C- 29 Mean Standard Deviations of Delay for AFIT- Mc Chord AFB. Path

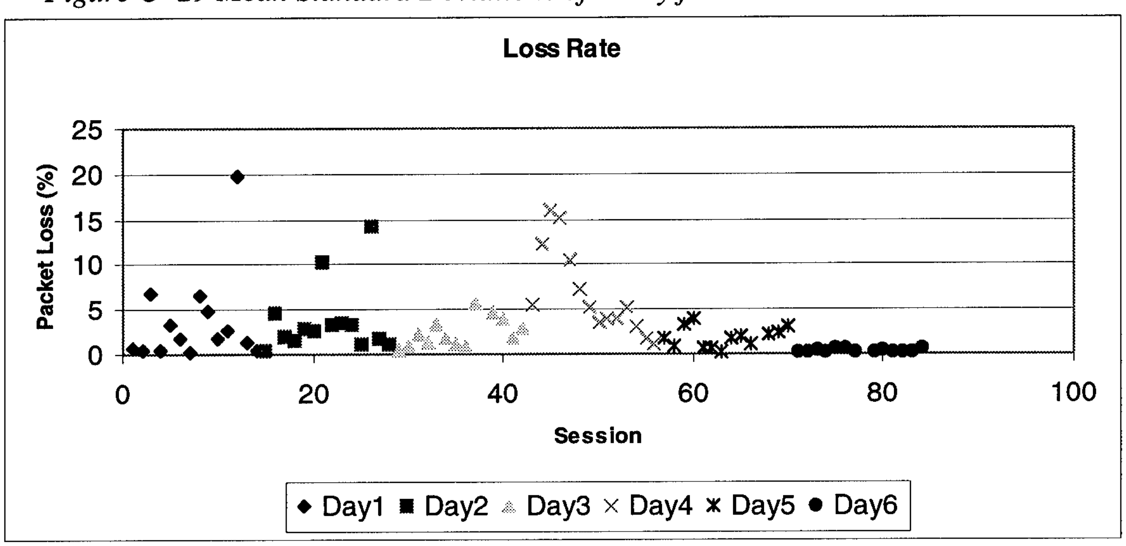


Figure C- 30 Packet Loss Rates for AFIT- Mc Chord AFB. Path

Appendix D

1. AFIT-Air University Path

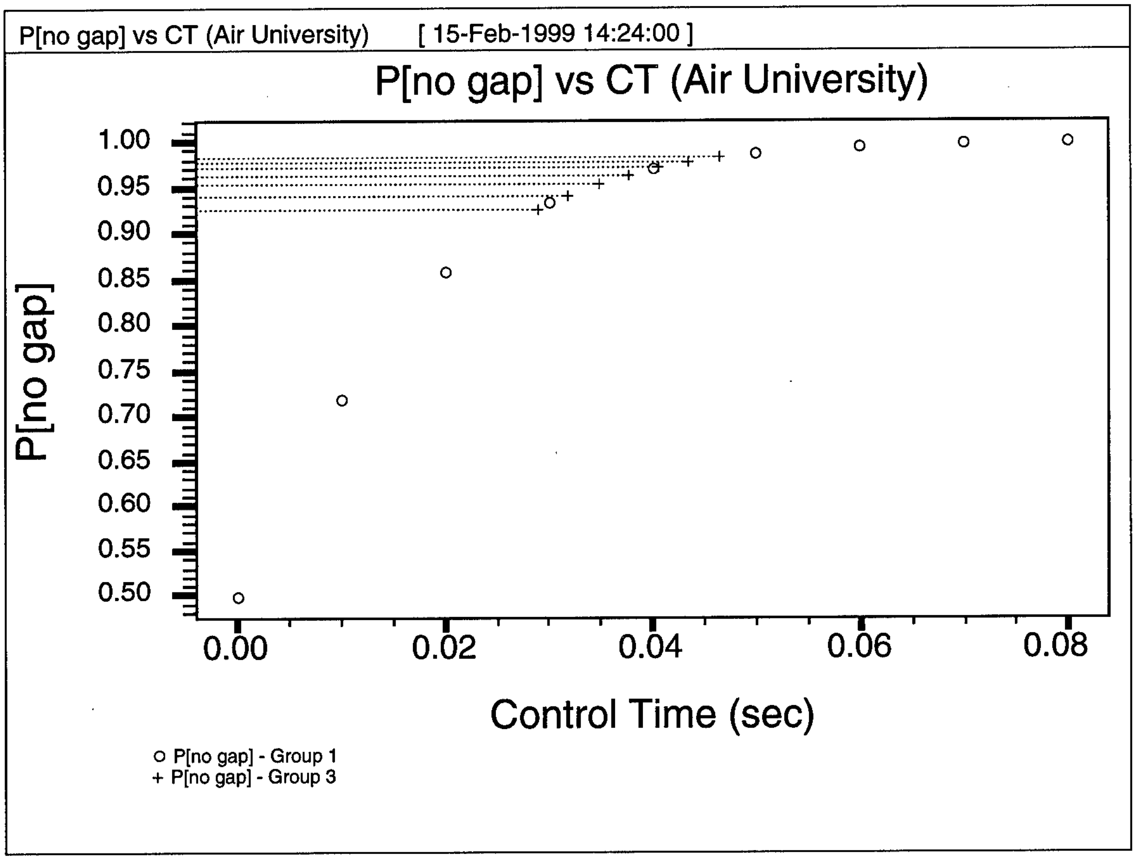


Figure D- 1 P[no gap] for AFIT-Air University Path

Table D- 1 P[no gap] for AFIT-Air University Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.4985	0.04	0.97012
0.01	0.71793	0.040569	0.9711
0.02	0.85854	0.043466	0.97688
0.028977	0.92639	0.046364	0.9819
0.03	0.93352	0.05	0.98694
0.031875	0.9413	0.06	0.99452
0.034773	0.95345	0.07	0.99761
0.037671	0.96332	0.08	0.99899

2. AFIT-Bolling AFB Path

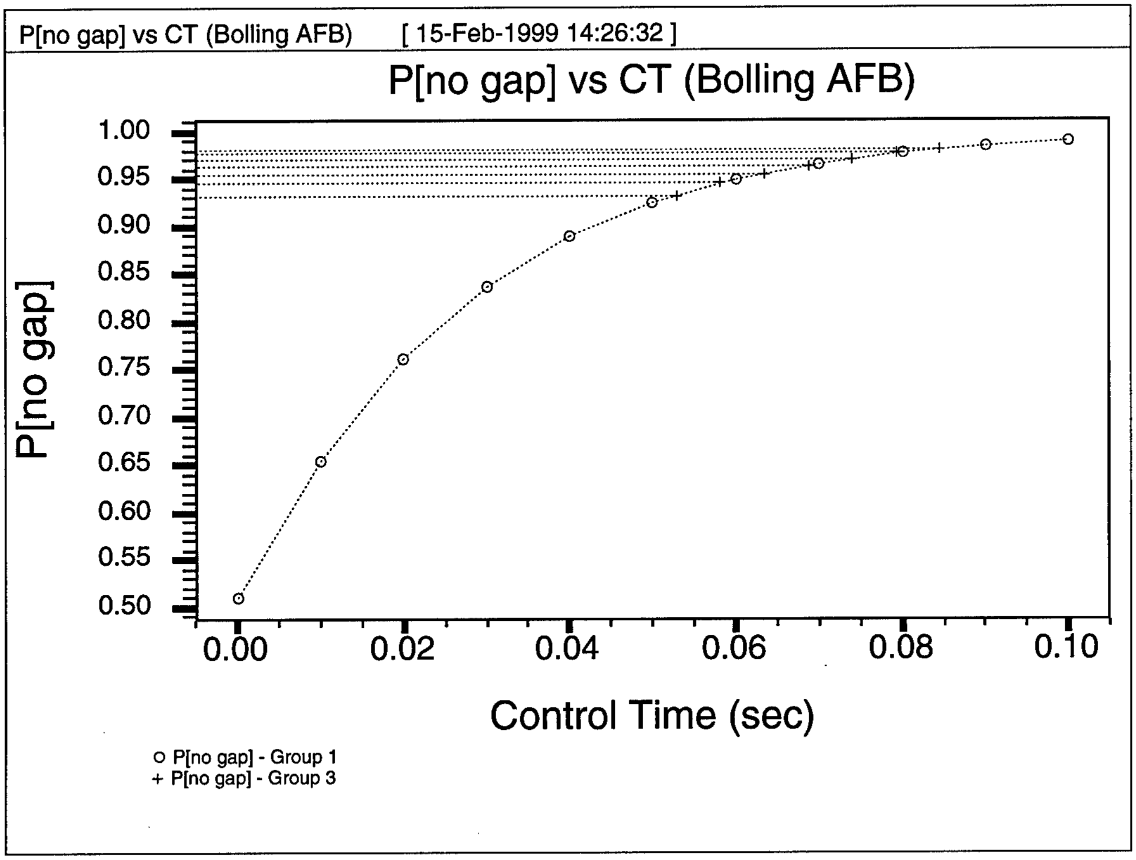


Figure D- 2 P[no gap] for AFIT-Bolling AFB Path

Table D- 2 P[no gap] for AFIT-Bolling AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.51061	0.063405	0.95587
0.01	0.65478	0.068689	0.96455
0.02	0.76222	0.07	0.96643
0.03	0.83764	0.073937	0.97132
0.04	0.88946	0.079256	0.97708
0.05	0.92523	0.08	0.97772
0.052837	0.9328	0.08454	0.98133
0.058121	0.94556	0.09	0.98507
0.06	0.94961	0.1	0.98994

3. AFIT-Hanscom AFB Path

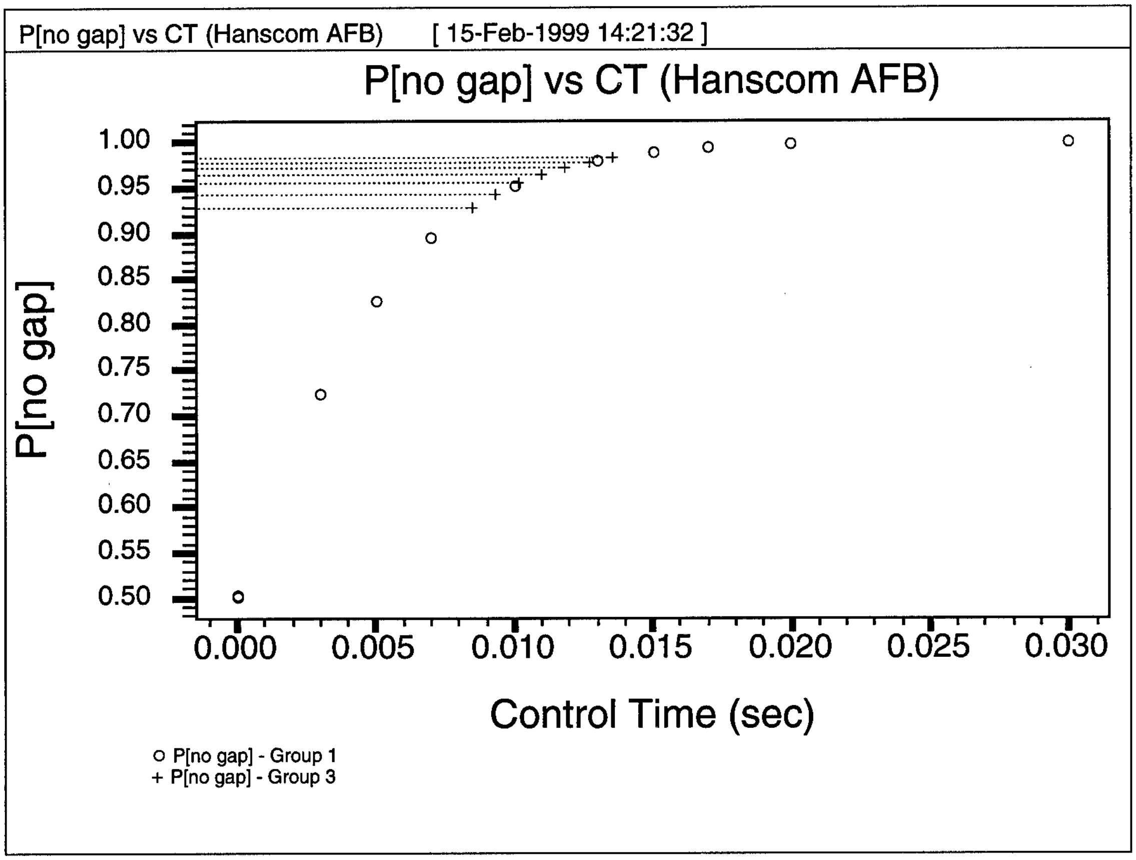


Figure D- 3 P[no gap] for AFIT-Hanscom AFB Path

Table D- 3 P[no gap] for AFIT-Hanscom AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.50288	0.01184	0.97204
0.003	0.72408	0.012686	0.97798
0.005	0.827	0.013	0.97994
0.007	0.89534	0.013532	0.9828
0.008457	0.92884	0.015	0.98884
0.009303	0.94321	0.017	0.994
0.01	0.95327	0.02	0.99747
0.010149	0.95543	0.03	0.99985
0.010994	0.96461		

4. AFIT-Lackland AFB Path

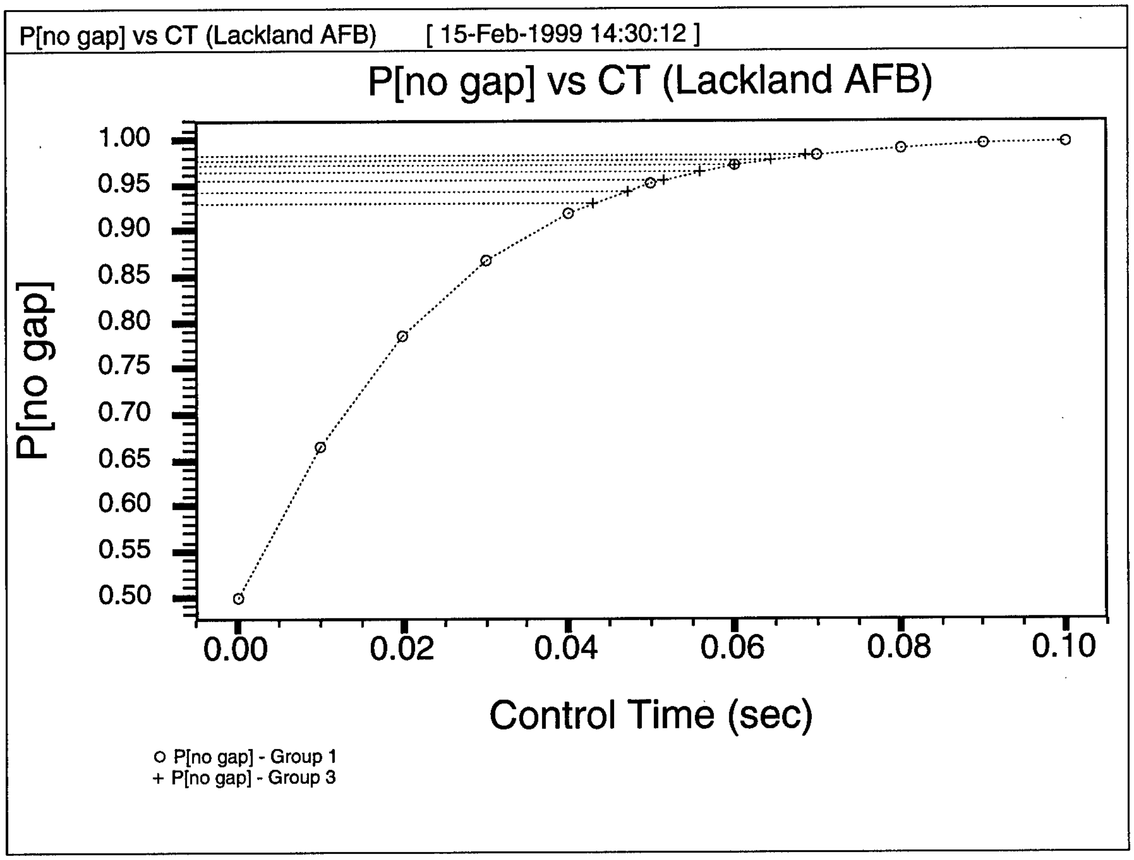


Figure D- 4 P[no gap] for AFIT-Lackland AFB Path

Table D- 4 P[no gap] for AFIT-Lackland AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.49878	0.055786	0.96339
0.01	0.66436	0.06	0.97081
0.02	0.78584	0.060077	0.9705
0.03	0.86724	0.064368	0.97661
0.04	0.91865	0.068659	0.98166
0.042912	0.92984	0.07	0.98262
0.047203	0.94287	0.08	0.98985
0.05	0.95118	0.09	0.99398
0.051494	0.9546	0.1	0.99646

5. AFIT-Mc Chord AFB Path

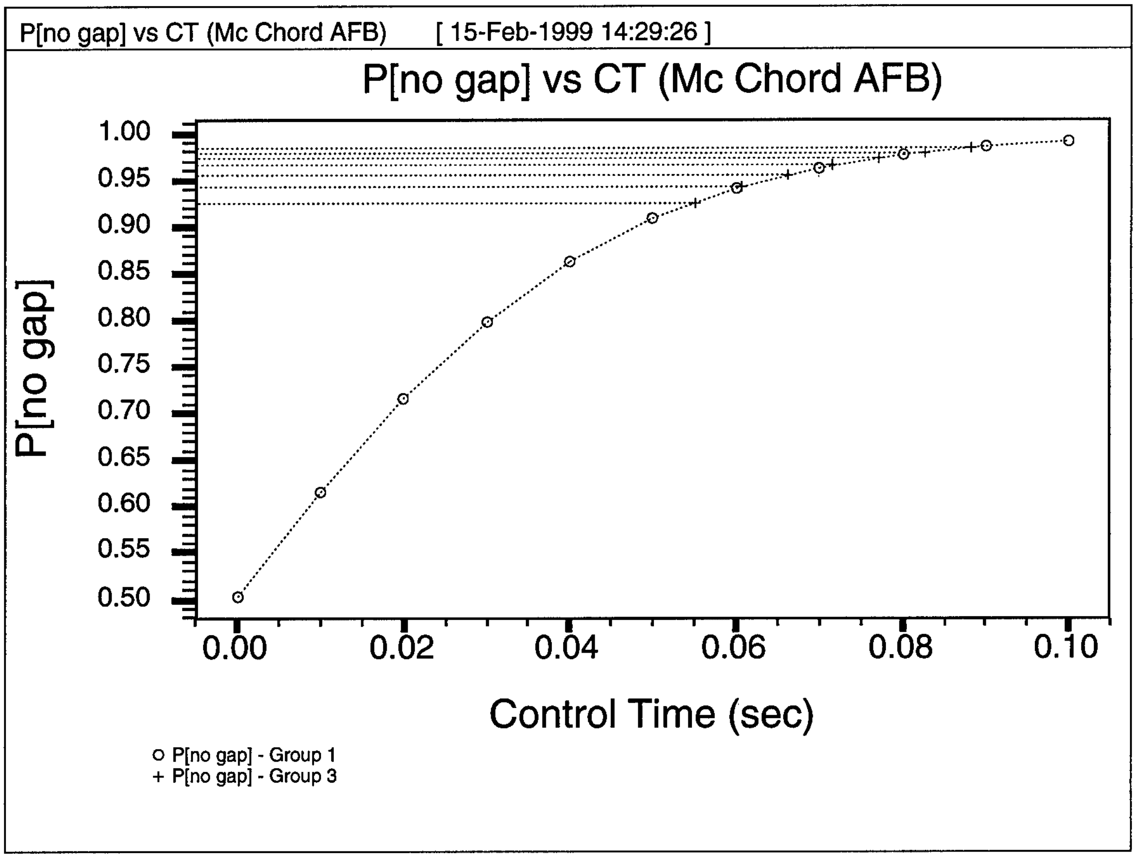


Figure D- 5 P[no gap] for AFIT-Mc Chord AFB Path

Table D- 5 P[no gap] for AFIT-Mc Chord AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.50318	0.066102	0.95566
0.01	0.61564	0.07	0.96311
0.02	0.71598	0.07161	0.9656
0.03	0.79829	0.077119	0.97334
0.04	0.86208	0.08	0.97695
0.05	0.90887	0.082627	0.9794
0.055085	0.92612	0.088136	0.98437
0.06	0.94163	0.09	0.98568
0.060593	0.94265	0.1	0.9913

6. AFIT-MacDill AFB Path

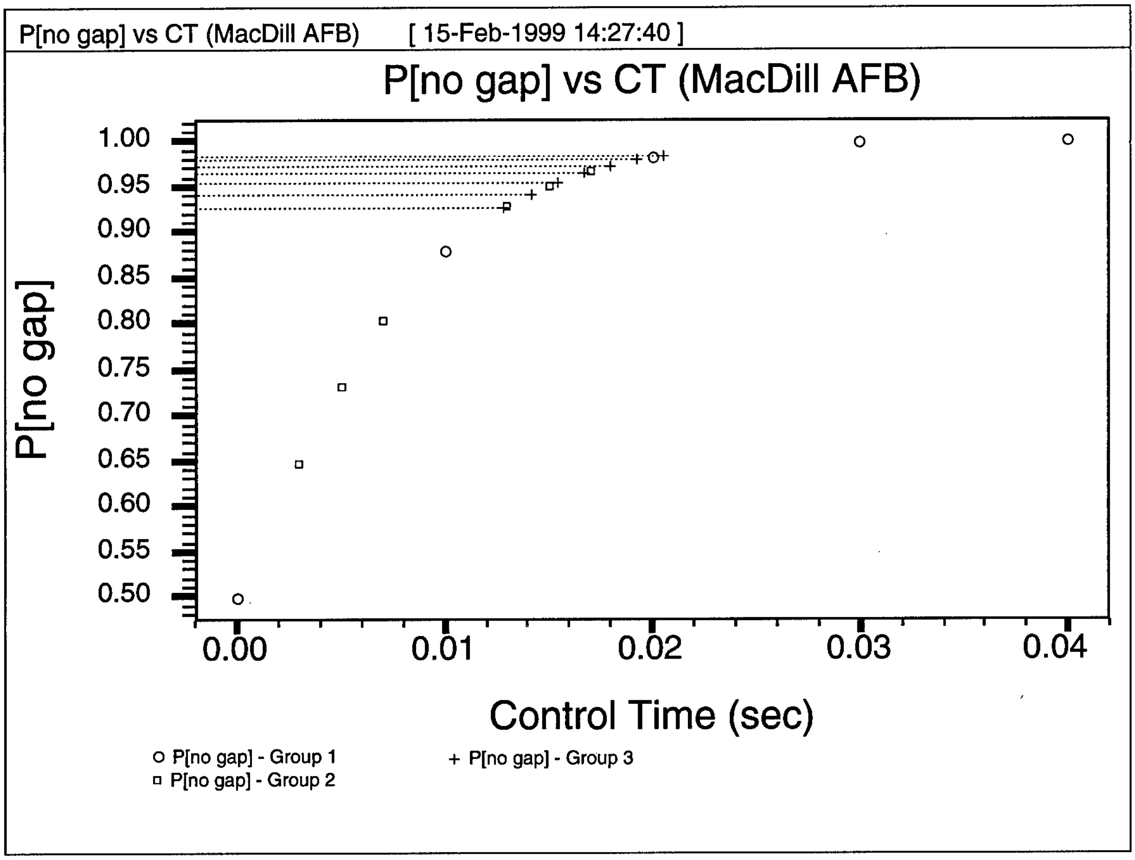


Figure D- 6 P[no gap] for AFIT-MacDill AFB Path

Table D- 6 P[no gap] for AFIT-MacDill AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.49826	0.015412	0.95481
0.003	0.64674	0.016696	0.96488
0.005	0.73163	0.017	0.9671
0.007	0.80261	0.01798	0.97311
0.01	0.87949	0.019265	0.97892
0.012843	0.92657	0.02	0.98165
0.013	0.92884	0.020549	0.98362
0.014127	0.942	0.03	0.99763
0.015	0.95106	0.04	0.99971

7. AFIT-Minot AFB Path

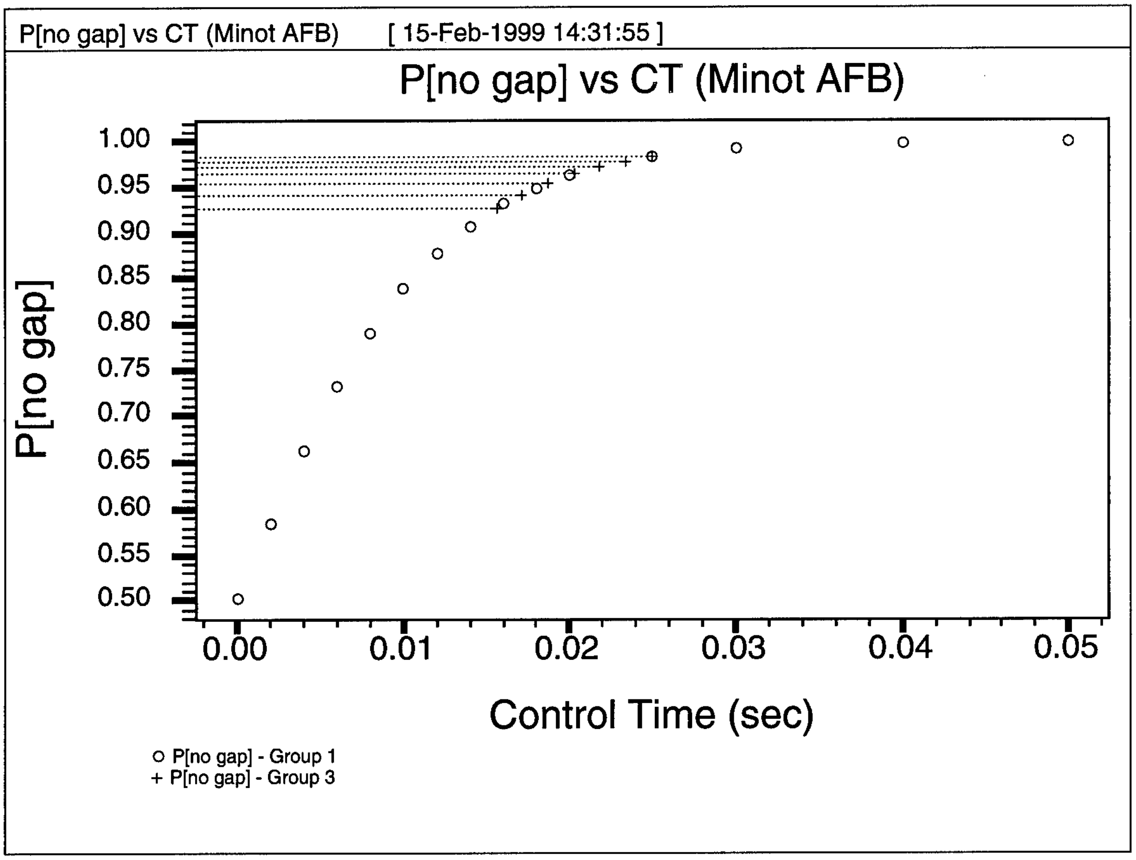


Figure D- 7 P[no gap] for AFIT-Minot AFB Path

Table D- 7 P[no gap] for AFIT-Minot AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.50253	0.018	0.94945
0.002	0.58492	0.018705	0.95465
0.004	0.66256	0.02	0.96331
0.006	0.73129	0.020264	0.96471
0.008	0.79003	0.021823	0.97238
0.01	0.83876	0.023381	0.97828
0.012	0.87778	0.02494	0.98328
0.014	0.90787	0.025	0.98343
0.015587	0.92731	0.03	0.99313
0.016	0.9317	0.04	0.99905
0.017146	0.94237	0.05	0.99979

8. AFIT-Scott AFB Path

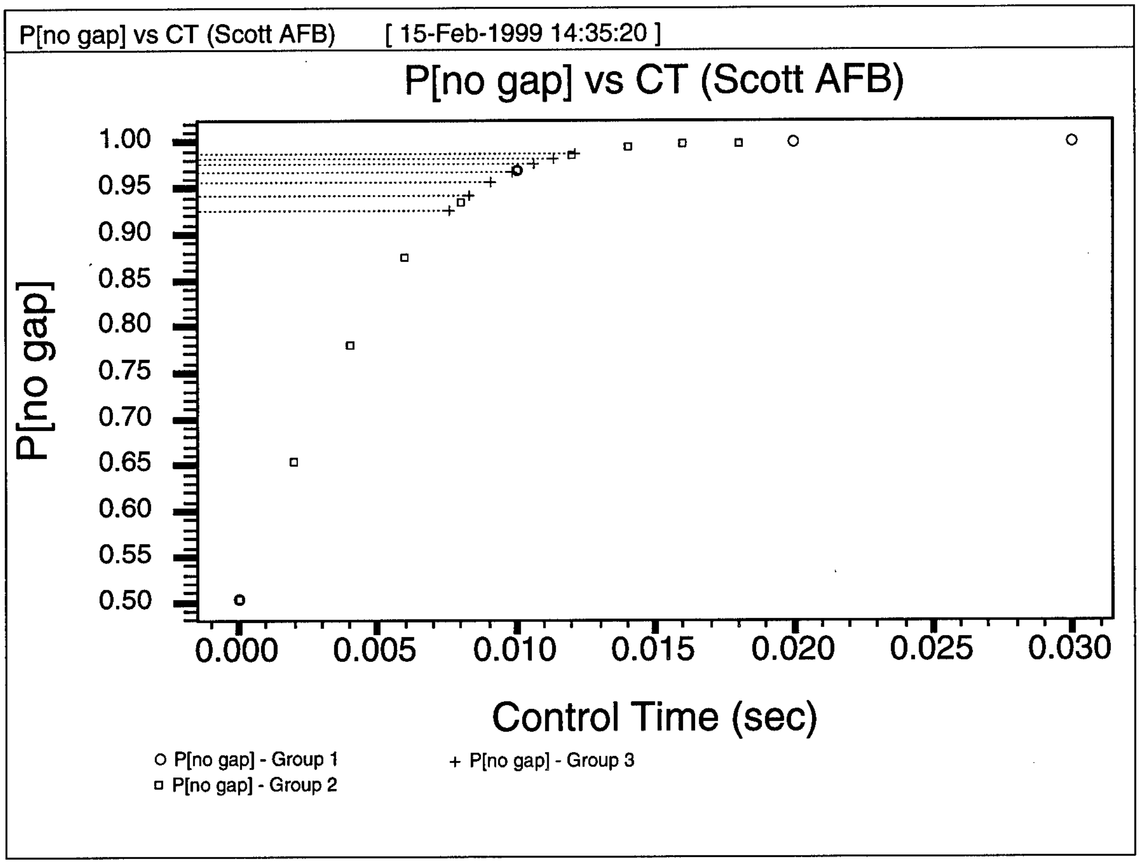


Figure D- 8 P[no gap] for AFIT-Scott AFB Path

Table D- 8 P[no gap] for AFIT-Scott AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.50334	0.010597	0.97596
0.002	0.65431	0.011354	0.9823
0.004	0.78015	0.012	0.98649
0.006	0.87502	0.012111	0.9871
0.007569	0.92495	0.014	0.99458
0.008	0.93516	0.016	0.99804
0.008326	0.94228	0.018	0.99925
0.009083	0.95611	0.02	0.99978
0.00984	0.96724	0.03	1
0.01	0.96929		

9. AFIT-Travis AFB Path

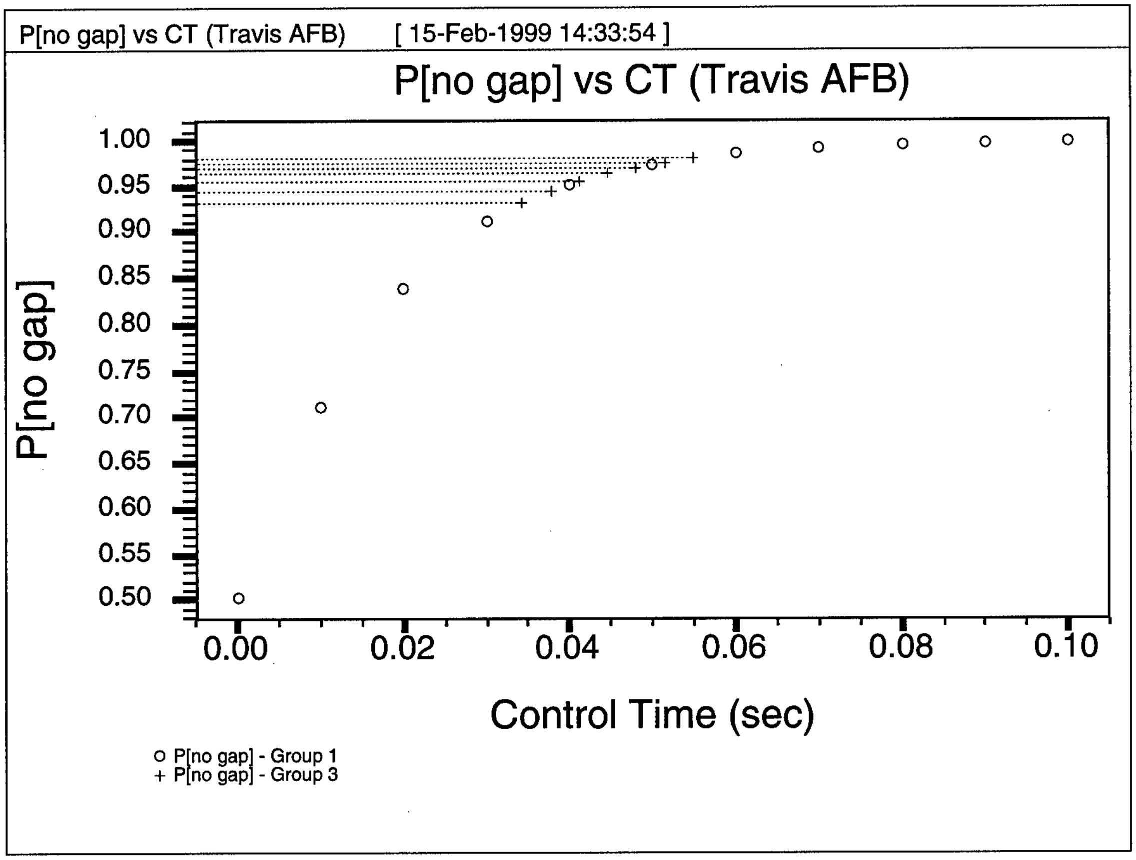


Figure D- 9 P[no gap] for AFIT-Travis AFB Path

Table D- 9 P[no gap] for AFIT-Travis AFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.50297	0.047981	0.97036
0.01	0.71196	0.05	0.97379
0.02	0.83923	0.051408	0.97593
0.03	0.911	0.054835	0.98056
0.034272	0.93173	0.06	0.98593
0.037699	0.94472	0.07	0.99221
0.04	0.9516	0.08	0.99598
0.041126	0.95486	0.09	0.99788
0.044554	0.96383	0.1	0.99882

10. AFIT-WPAFB Path

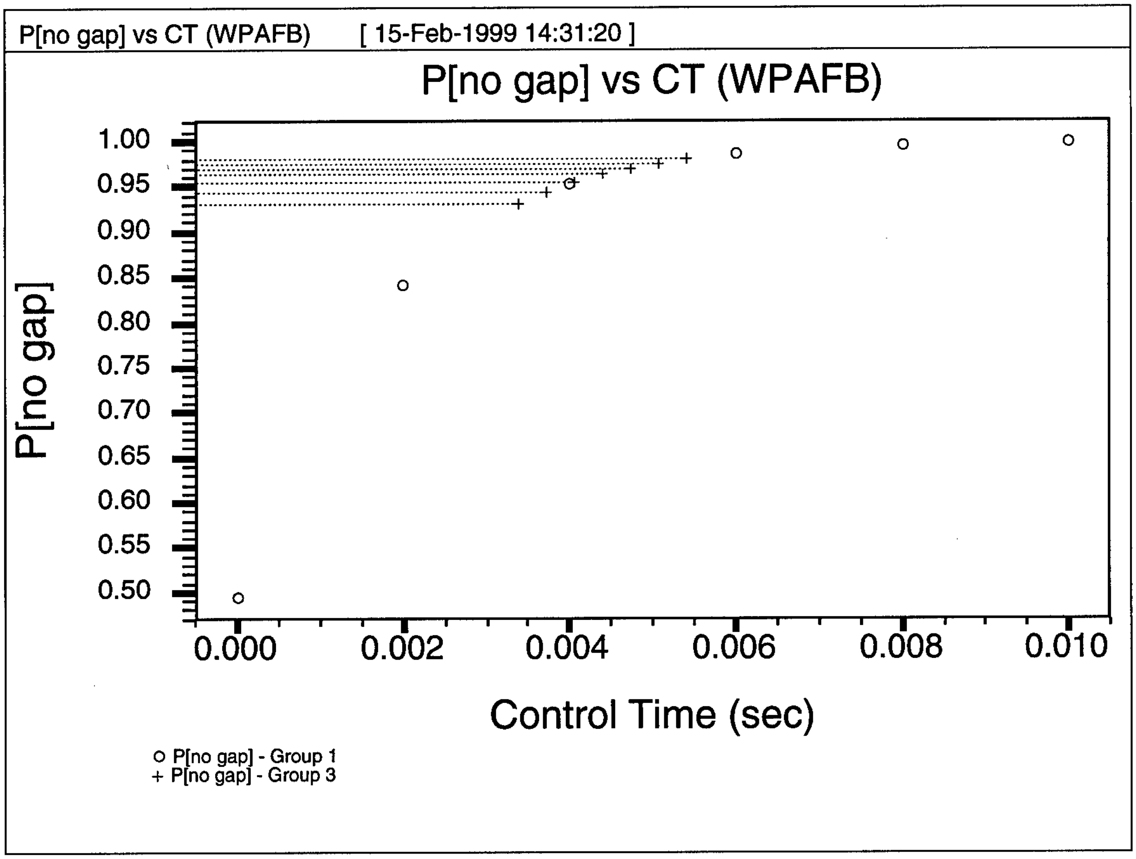


Figure D- 10 P[no gap] for AFIT-WPAFB Path

Table D- 10 P[no gap] for AFIT-WPAFB Path

Control Time	P[no gap]	Control Time	P[no gap]
0	0.49478	0.004727	0.96995
0.002	0.84148	0.005065	0.97574
0.003376	0.93129	0.005403	0.98029
0.003714	0.94402	0.006	0.98662
0.004	0.95311	0.008	0.99627
0.004052	0.95469	0.01	0.99879
0.00439	0.96312		

Appendix E

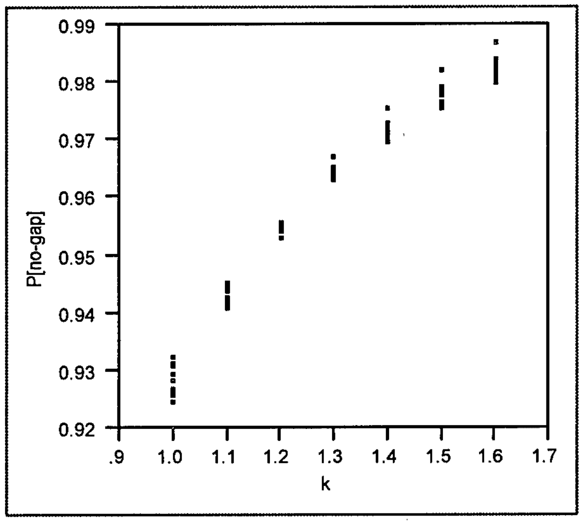


Figure E- 1 P[no gap] by k from simulations

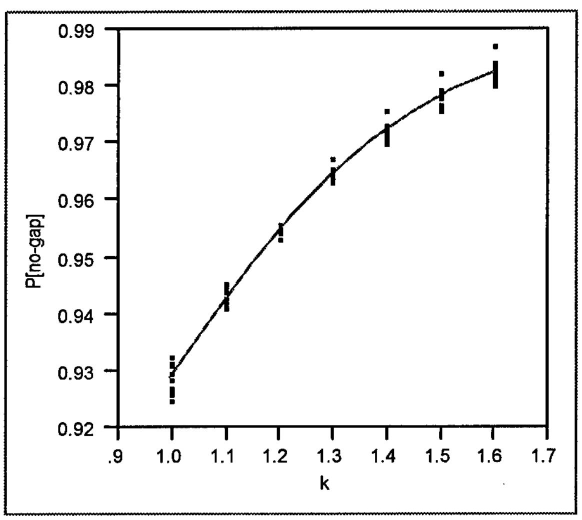


Figure E- 3 P[no gap] by k from formula 1

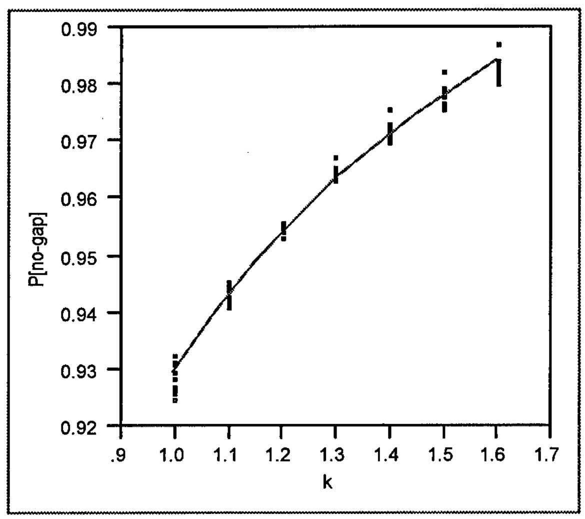


Figure E- 2 P[no gap] by k from formula 2

Polynomial Fit degree=2

Transformed Fit to Recip

FITTING FORMULA -1

(Polynomial Fit degree=2)

$$P[\text{no-gap}] = 0.68236 + 0.34514\,k - 0.09855\,k^2$$

Summary of Fit

RSquare	0.990873
RSquare Adj	0.9906
Root Mean Square Error	0.001776
Mean of Response	0.960547
Observations (or Sum Wgts)	70

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	2	0.02294872	0.011474	3636.872
Error	67	0.00021139	0.000003	Prob>F
C Total	69	0.02316010		<.0001

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.6823593	0.010208	66.84	<.0001
k	0.3451379	0.01597	21.61	<.0001
k^2	-0.09855	0.006129	-16.08	<.0001

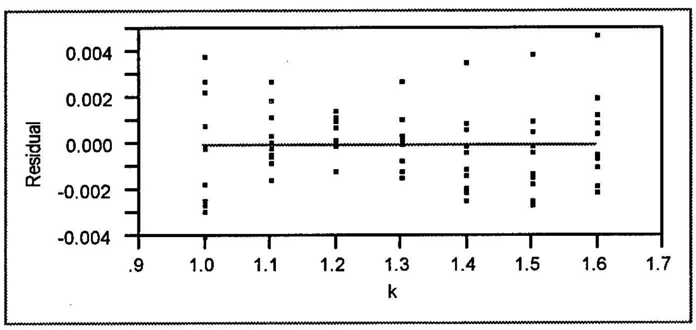


Figure E- 4 Residual by k for polynomial fit degree=2

FITTING FORMULA -2

(Transformed Fit to Reciprocal)

P[no-gap] = 1.0749 – 0.14508 Recip(k)

Summary of Fit

RSquare	0.988222
RSquare Adj	0.988049
Root Mean Square Error	0.002003
Mean of Response	0.960547
Observations (or Sum Wgts)	70

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	0.02288732	0.022887	5705.506
Error	68	0.00027278	0.000004	Prob>F
C Total	69	0.02316010		<.0001

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	1.0749028	0.001533	701.29	<.0001
Recip(k)	-0.145079	0.001921	-75.53	<.0001

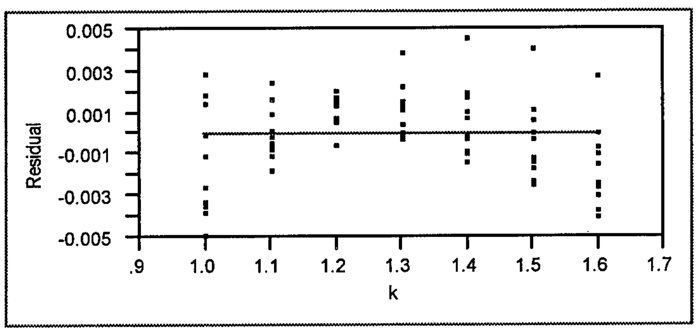


Figure E- 5 Residual by k for Transformed Fit to Recip

Appendix F

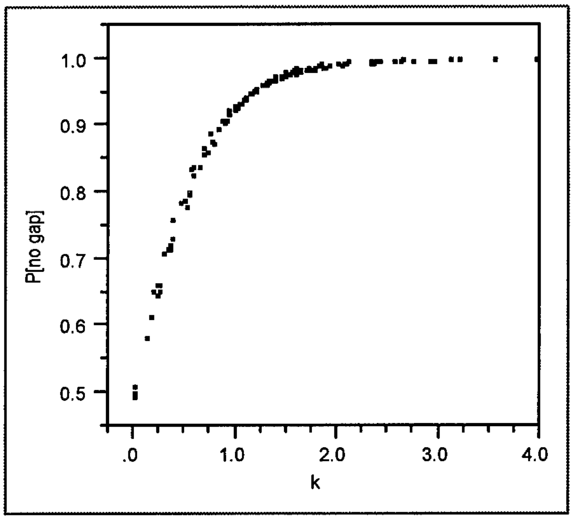


Figure F- 1 P[no gap] by k from simulations

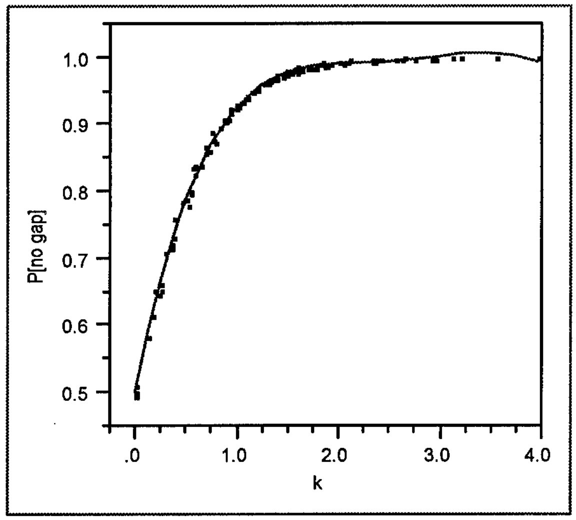


Figure F- 2 P[no gap] by k from formula 3

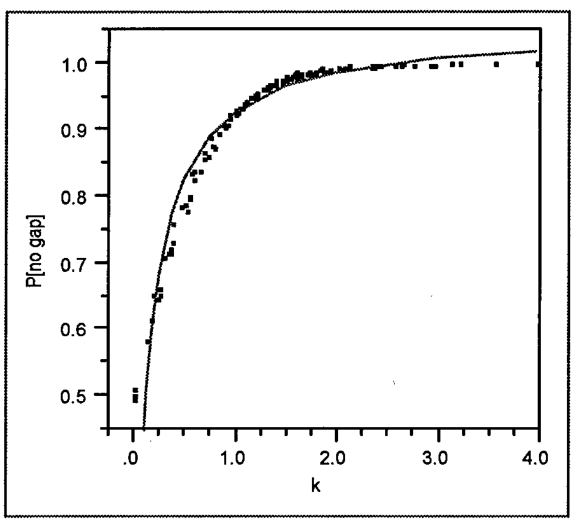


Figure F- 4 P[no gap] by k from formula 4

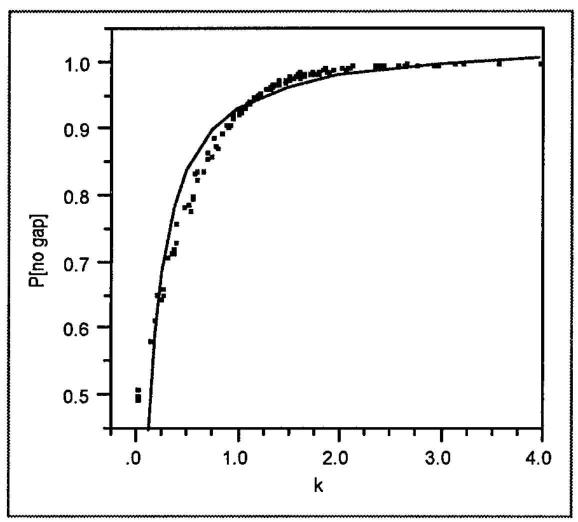


Figure F- 3 P[no gap] by k from formula 5

Polynomial Fit degree=4

Transformed Fit Recip to Recip

Transformed Fit Log to Recip

FITTING FORMULA -3

(Polynomial Fit degree=4)

$$P[\text{no gap}] = 0.50093 + 0.78095\,k - 0.46367\,k^2 + 0.12111\,k^3 - 0.01159\,k^4$$

Summary of Fit

RSquare	0.998012
RSquare Adj	0.997966
Root Mean Square Error	0.005925
Mean of Response	0.905746
Observations (or Sum Wgts)	177

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	4	3.0320043	0.758001	21590.89
Error	172	0.0060385	0.000035	Prob>F
C Total	176	3.0380428		<.0001

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.5009335	0.001655	302.69	<.0001
k	0.7809544	0.006645	117.53	<.0001
k^2	-0.463667	0.007975	-58.14	<.0001
k^3	0.121114	0.003398	35.64	<.0001
k^4	-0.011588	0.000465	-24.92	<.0001

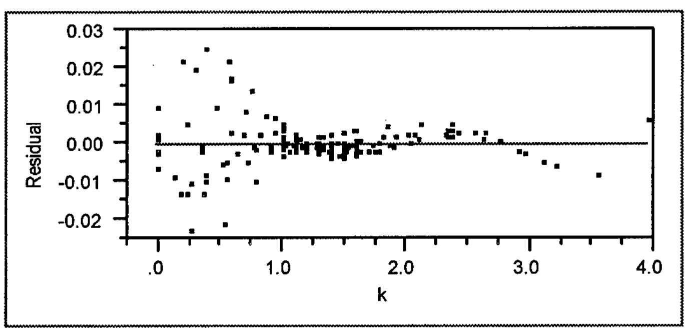


Figure F- 5 Residual by k for polynomial fit degree=4

FITTING FORMULA -4

(Transformed Fit Reciprocal to Reciprocal)

Recip(P[no gap]) = 0.9502 + 0.1276 Recip(k)

Summary of Fit

RSquare	0.943722
RSquare Adj	0.943381
Root Mean Square Error	0.031535
Mean of Response	1.087789
Observations (or Sum Wgts)	167

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	2.7515922	2.75159	2766.885
Error	165	0.1640880	0.00099	Prob>F
C Total	166	2.9156803		<.0001

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.950202	0.003577	265.62	<.0001
Recip(k)	0.1275983	0.002426	52.60	<.0001

Fit Measured on Original Scale

Sum of Squared Error	0.0483152
Root Mean Square Error	0.017112
R-square	0.9630219
Sum of Residuals	0.2055717

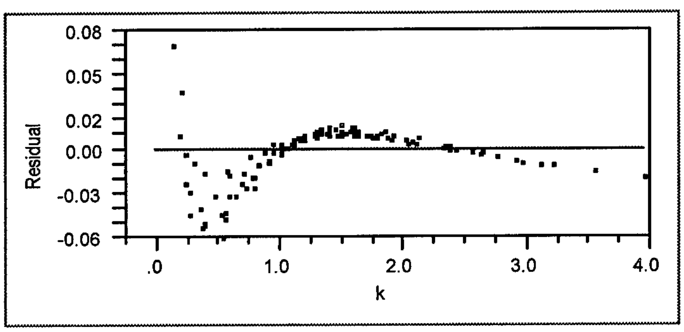


Figure F- 6 Residual by k for transformed fit recip. to recip.

FITTING FORMULA -5

(Transformed Fit Log to Reciprocal)

Log(P[no gap]) = 0.03213 – 0.10206 Recip(k)

Summary of Fit

RSquare	0.919229
RSquare Adj	0.918739
Root Mean Square Error	0.030618
Mean of Response	-0.07792
Observations (or Sum Wgts)	167

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	1.7604312	1.76043	1877.813
Error	165	0.1546859	0.00094	Prob>F
C Total	166	1.9151170		<.0001

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>ltl
Intercept	0.0321326	0.003473	9.25	<.0001
Recip(k)	-0.102062	0.002355	-43.33	<.0001

Fit Measured on Original Scale

Sum of Squared Error	0.0806144
Root Mean Square Error	0.0221037
R-square	0.9383017
Sum of Residuals	0.0928185

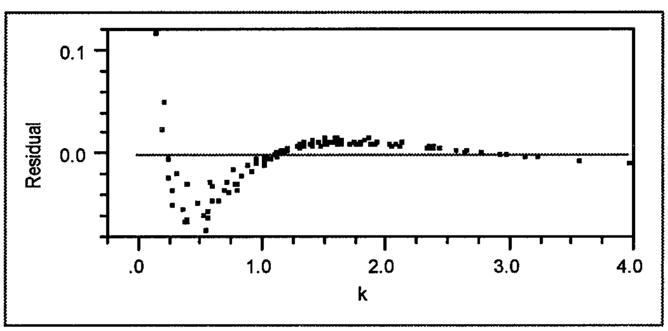


Figure F- 7 Residual by k for transformed fit log to recip.

Table F- 1 P[no gap] by k for all simulation results

k	P[no gap]	PATH	K	P[no gap]	PATH	k	P[no gap]	PATH	k	P[no gap]	PATH
0	0.49478	10	0.908	0.90887	5	1.2	0.95611	8	1.6	0.98029	10
0	0.49826	6	0.932	0.91865	4	1.271	0.96311	5	1.6	0.98056	9
0	0.4985	1	0.946	0.92523	2	1.283	0.96331	7	1.6	0.98133	2
0	0.49878	4	1	0.92495	8	1.3	0.96312	10	1.6	0.98166	4
0	0.50253	7	1	0.92612	5	1.3	0.96332	1	1.6	0.9819	1
0	0.50288	3	1	0.92639	1	1.3	0.96339	4	1.6	0.9828	3
0	0.50297	9	1	0.92657	6	1.3	0.96383	9	1.6	0.98328	7
0	0.50318	5	1	0.92731	7	1.3	0.96455	2	1.6	0.98362	6
0	0.50334	8	1	0.92884	3	1.3	0.96461	3	1.6	0.98437	5
0	0.51061	2	1	0.92984	4	1.3	0.96471	7	1.6	0.9871	8
0.128	0.58492	7	1	0.93129	10	1.3	0.96488	6	1.604	0.98343	7
0.182	0.61564	5	1	0.93173	9	1.3	0.9656	5	1.631	0.98262	4
0.189	0.65478	2	1	0.9328	2	1.3	0.96724	8	1.634	0.98568	5
0.233	0.66436	4	1.012	0.92884	6	1.321	0.96929	8	1.703	0.98507	2
0.234	0.64674	6	1.026	0.9317	7	1.324	0.9671	6	1.725	0.98694	1
0.257	0.66256	7	1.035	0.93352	1	1.325	0.96643	2	1.751	0.98593	9
0.264	0.65431	8	1.057	0.93516	8	1.38	0.97012	1	1.774	0.98884	3
0.292	0.71196	9	1.089	0.94163	5	1.398	0.97081	4	1.777	0.98662	10
0.345	0.71793	1	1.1	0.9413	1	1.399	0.97132	2	1.815	0.9913	5
0.355	0.72408	3	1.1	0.942	6	1.4	0.96995	10	1.849	0.99458	8
0.363	0.71598	5	1.1	0.94228	8	1.4	0.97036	9	1.864	0.98985	4
0.379	0.76222	2	1.1	0.94237	7	1.4	0.9705	4	1.893	0.98994	2
0.385	0.73129	7	1.1	0.94265	5	1.4	0.9711	1	1.925	0.99313	7
0.389	0.73163	6	1.1	0.94287	4	1.4	0.97204	3	2.01	0.994	3
0.466	0.78584	4	1.1	0.94321	3	1.4	0.97238	7	2.042	0.99221	9
0.513	0.79003	7	1.1	0.94402	10	1.4	0.97311	6	2.071	0.99452	1
0.528	0.78015	8	1.1	0.94472	9	1.4	0.97334	5	2.097	0.99398	4
0.545	0.79829	5	1.1	0.94556	2	1.4	0.97596	8	2.114	0.99804	8
0.545	0.80261	6	1.136	0.94961	2	1.452	0.97695	5	2.33	0.99646	4
0.568	0.83764	2	1.155	0.94945	7	1.459	0.97379	9	2.334	0.99598	9
0.584	0.83923	9	1.165	0.95118	4	1.5	0.97574	10	2.336	0.99763	6
0.591	0.827	3	1.167	0.9516	9	1.5	0.97593	9	2.365	0.99747	3
0.592	0.84148	10	1.168	0.95106	6	1.5	0.97661	4	2.369	0.99627	10
0.642	0.83876	7	1.182	0.95327	3	1.5	0.97688	1	2.378	0.99925	8
0.69	0.85854	1	1.185	0.95311	10	1.5	0.97708	2	2.416	0.99761	1
0.699	0.86724	4	1.2	0.95345	1	1.5	0.97798	3	2.566	0.99905	7
0.726	0.86208	5	1.2	0.9546	4	1.5	0.97828	7	2.626	0.99788	9
0.757	0.88946	2	1.2	0.95465	7	1.5	0.97892	6	2.642	0.99978	8
0.77	0.87778	7	1.2	0.95469	10	1.5	0.9794	5	2.761	0.99899	1
0.779	0.87949	6	1.2	0.95481	6	1.5	0.9823	8	2.918	0.99882	9
0.793	0.87502	8	1.2	0.95486	9	1.514	0.97772	2	2.961	0.99879	10
0.828	0.89534	3	1.2	0.95543	3	1.537	0.97994	3	3.114	0.99971	6
0.875	0.911	9	1.2	0.95566	5	1.557	0.98165	6	3.208	0.99979	7
0.898	0.90787	7	1.2	0.95587	2	1.585	0.98649	8	3.547	0.99985	3
									3.963	1	8

Bibliography

[AgG93] D. Sanghi, A. K. Agrawala, O. Gudmundsson, B. N. Jain. "Experimental Assessment of End-to-end Behavior on Internet", IEEE INFOCOM 93, March 1993,

[AlR94] Alta Group. "BONeS DESIGNER Application Library Reference", Product Number: DES 8018. City: Cadence Design Systems, Inc., December 1994

[BaC96] Banks, Jerry., Carson, John S., and Nelson, Barry L. Discrete-Event System Simulation. New Jersey: Prentice Hall, 1996.

[BCP93] K. Claffy, G. Polyzos, and H. W. Braun. "Measurement Considerations for Assessing Unidirectional Latencies", Internetworking: Research and Experience, Vol. 4, No. 3, September 1993

[BoG98] Thomas J. Kostas, Michael S. Botella, Ikhlaq Sidhu, Guido M. Schuster, Jacek Grabiec, Jerry Mahler. "Real-Time Voice Over Packet-Switched Networks", IEEE Network, January/February,1998, pp 18-27.

[BP80] Giulio Barberis and Daniele Pazzagle. "Analysis and Optimal Design of a Packet-Voice Receiver", IEEE Transactions on Communications, Vol. COM-28, No. 2, February 1980

[Bra65] P.T. Brady. "A Technique for investigating ON-OFF Patterns of Speech" Bell System Technical Journal, 44(1):1-22, January 1965

[Cis98] Cisco Systems, Inc. Pocket Voice Primer, WWWeb, http://www.cisco.com/warp/public/779/largeent/multinet/packv_in.htm, April 1998

[Com95] Douglas E. Comer, Internetworking with TCP/IP volume I. Prentice Hall, Inc. 1995

[Fas97] Andreas Fasbender. "On Assessing Unidirectional Latencies in Packet-Switched Networks", IEEE Network, 44(1):1-22, January 1997

[G.114] CCITT. Recommendation G.114

[Gil98] Gilbert Held. Voice over Data Networks. McGraw-Hill, 1998

[HeL86] Heffes, H., and Lucantoni, D. M. " A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer

Performance", IEEE Journal on Selected Areas in Communications, SAC-4, 6 (September 1986), 852-886

[Jay93] N. Jayant. "High Quality Networking of Audio-Visual Information", IEEE Communications, 31(9):84-95, September 1993

[Kle67] E. Klemmer. "Subjective Evaluation of Transmission Delay in Telephone Conversations", Bell System Technical Journal, 46:1141-1147, July 1967

[KlN82] William E. Naylor and Leonard Kleinrock. "Stream Traffic Communication in Packet Switched Networks: Destination Buffering Considerations", IEEE Transactions on Communications, Vol. COM-30, No. 12, December 1982

[KuD94] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks", Proceedings of the Conference on Computer Communications (IEEE Infocom), pp. 680-688, June 1994

[MM98] Daniel Minoli, Emma Minoli, Delivering Voice over IP Networks. John Wiley and Sons, Inc. 1998

[Mon83] Warren A. Montgomery " Techniques for Packet Voice Synchronization", IEEE Journal on Selected Areas in Communication, Vol.Sac-1, No. 6, December 1983

[NaF67] Naylor, T. H. Finger J.M. "Verification of Computer Simulation Models", Management Science, Vol. 2 pp. B92-B101

[OO98] Miguel Angel Olabe and Juan Carlos Olabe. "Telecommunications Network Design Using Modeling and Simulation", IEEE Transactions on Education, Vol. 41, No. 1, February 1998

[Pax97] V. Paxson, "End-to-end Routing Behavior in the Internet", IEEE/ACM Networking, vol. 5, no. 5, October 1997.

[RaR92] A. Shah, D. Staddon, I. Rubin, and A. Ratkovic. "Multimedia over FDDI", 17th IEEE Conference on Local Computer Networks, pages 110-124, September 1992.

[Rya98] Jerry Ryan, "Voice Over IP Guide," WWWeb, http://www.techguide.com/index1.html, 1998

[SrW86] Sriram, K., and Whitt, W. "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", IEEE Journal on Selected Areas in Communications, SAC-4, 6 (September 1986), 833-846

[Ste90] W. Richard Stevens. UNIX Network Programming, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[WF83] Clifford Weinstein and James W. Forgie. "Experience with Speech Communication in Packet Networks", IEEE Journal on Selected Areas in Communications, 6(1), 1983