Analysing the Characteristics of VoIP Traffic

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Qinxia (Alice) He

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

In this study, the characteristics of VoIP traffic in a deployed Cisco VoIP phone system and a SIP based soft phone system are analysed. Traffic was captured in a soft phone system, through which elementary understanding about a VoIP system was obtained and experimental setup was validated. An advanced experiment was performed in a deployed Cisco VoIP system in the department of Computer Science at the University of Saskatchewan. Three months of traffic trace was collected beginning October 2006, recording address and protocol information for every packet sent and received on the Cisco VoIP network. The trace was analysed to find out the features of Cisco VoIP system and the findings were presented.

This work appears to be one of the first real deployment studies of VoIP that does not rely on artificial traffic. The experimental data provided in this study is useful for design and modeling of such systems, from which more useful predictive models can be generated. The analysis method used in this research can be used for developing synthetic workload models. A clear understanding of usage patterns in a real VoIP network is important for network deployment and potential network activities such as integration, optimizations or expansion.

The major factors affecting VoIP quality such as delay, jitter and loss were also measured and simulated in this study, which will be helpful in an advanced VoIP quality study. A traffic generator was developed to generate various simulated VoIP traffic. The data used to provide the traffic model parameters was chosen from peak traffic periods in the captured data from University of Saskatchewan deployment. By utilizing the Traffic Trace function in ns2, the simulated VoIP traffic was fed into ns2, and delay, jitter and packet loss were calculated for different scenarios. Two simulation experiments were performed. The first experiment simulated the traffic of multiple calls running on a backbone link. The second experiment simulated a real network environment with different traffic load patterns. It is significant for network expansion and integration.

Acknowledgements

I would like to express my appreciation to Dr. Derek Eager, Dr. Dwight Makaroff and Mr. Andrew Kostiuk, my instructors, for their valuable comments and suggestions to my research. I want to acknowledge my other committee members, Dr. Daniel Teng (external), Dr. Nadeem Jamali and Dr. Nathanial Osgood for their helpful comments on my thesis. I want to thank Brian Gallaway for his great help in trace collection. I would also like to thank the ITS department at the UofS for facilitating this project.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

VoIP stands for Voice over IP, or in other words, telephone service over the Internet. VoIP is simply the transmission of voice conversations over IP-based networks. Although IP was originally designed for data networking, its success has led to its adaptation to voice networking. Today, VoIP has begun to be accepted by more and more consumers and business users.

There are three major areas of VoIP study. First, VoIP quality measurement; second, methods to improve VoIP quality; third, VoIP protocol and traffic analysis which aids in the understanding and design of VoIP systems, from which more useful predictive models can be generated. This study falls into the third area. Since more and more VoIP systems are appearing, characterising the traffic of a VoIP system can help to better understand and improve these systems.

In this work, an experimental study of a deployed Cisco VoIP phone system and a SIP (Session Initiation Protocol) based soft phone system are presented. VoIP traffic was captured for over three months in a deployed Cisco VoIP system in the Department of Computer Science at the University of Saskatchewan. Traffic in a SIP based soft phone system was also captured, through which elementary understanding about a VoIP system was obtained and the experimental setup was validated. As well, performance experiments were conducted for both systems. Through the trace data, some features of VoIP phone system are revealed such as the call setup and tear down process, distribution of the traffic data and network performance. A simulation study of VoIP quality metrics delay, jitter and loss was also conducted. A traffic generator was developed to generate various simulated VoIP traffic patterns.

The aim of this work is to analyse the traffic behavior of two different VoIP systems and provide experimental data that is useful for future network integration, optimization or expansion and advanced study of VoIP quality.

1.1 VoIP Principle

VoIP is the routing of voice traffic over the Internet or any other IP-based network. Using the Internet's packet-switching capabilities, VoIP technology has been implemented to provide telephone services [29]. Figure 1.1 illustrates a typical VoIP architecture though many "possible" modifications of this architecture are implemented in existing systems.

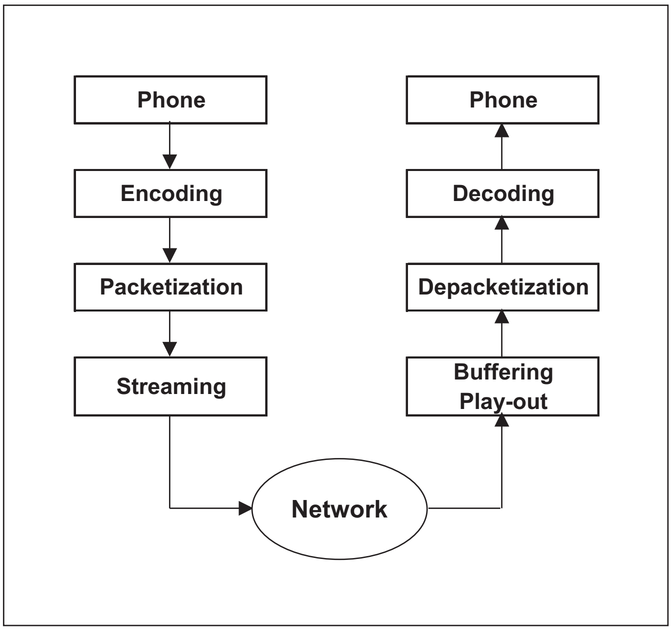


Figure 1.1: VoIP architecture

At the sending end, the original voice signal is sampled and encoded to a constant bit rate digital stream. The digital stream can then be easily compressed. This digitized and compressed data is then encapsulated into packets of equal sizes for easy transmission over the Internet. Along with the compressed voice data, these packets contain information about the packet's origin, the intended destination, and a timestamp that allows the packet stream to be reconstructed in the correct order. These packets flow over a general-purpose packet-switched network, instead of traditional dedicated, circuit-switched voice transmission lines. At the receiving end, the continuous stream of packets are depacketized and converted back into the analog signal so that it can be detected by the human ear. In general, this means voice information is sent in digital form in discrete packets rather than using the traditional circuit-committed protocols of the Public Switched Telephone Network (PSTN). In addition to IP, VoIP uses the Real-Time Transport Protocol (RTP) to help ensure that packets get delivered in a timely way. Over the last few years, VoIP has become increasingly popular and is already starting to replace existing telephone networks. It has the potential to completely substitute for the world's current phone systems.

1.2 Reasons for VoIP Deployment

There are two major reasons to use VoIP: lower cost than traditional landline telephone and increased functionality. Each of these will be described in the remainder of this section.

1.2.1 Lower Cost to Consumers

VoIP becoming popular can be mainly attribute to the cost advantages to consumers over traditional telephone networks. The traditional business model for telephone services has been that most people pay a flat monthly fee for local telephone call service and a per-minute charge for long-distance calls. The deployment of VoIP has led to the possibility of change in this because the companies and organizations have offered different business models. Though the cost for an organization to convert to VoIP is not trivial, the monthly operational costs could be lower, so the overall long-term cost of VOIP is expected to decrease.

VoIP calls can be deployed using just Internet resources from computers equipped with microphones and speakers. Additional VoIP handsets can be directly connected to the Internet or to an Intranet. Most Internet connections are charged using a flat monthly fee structure. For International calling, the savings can be significant to the consumer by switching to VoIP technology. Using the Internet connection for both data traffic and voice calls can allow consumers to eliminate one monthly payment for telephone. In addition, VoIP plans do not charge a per-minute fee for long distance.

1.2.2 Increased Functionality

VoIP makes various tasks, which are difficult or impossible with traditional phone networks, easy.

- Incoming phone calls can be automatically routed to the VoIP phone wherever the phone is plugged into the network. So incoming calls can be received anywhere in the network.
- Call center agents using VoIP phones can easily work from anywhere with a good Internet connection.
- Multi-party conferencing is also much easier and cheaper because no bridge is required for small conferences.

Another advantage of VoIP is that a stand-alone telephone or videophone can be integrated with the personal computer. One can use a computer entirely for voice and video communications (softphones), use a telephone for voice and the computer for video, or can simply use the computer in conjunction with a separate voice/video phone to provide data conferencing functions, like application sharing, electronic whiteboarding, and text chat.

VoIP technology provides more abundant and flexible foundations for establishing communication services [12]. IP networks support independent connections for signaling and media traffic. Interference between the information flows has been avoided by the decoupling of signal and bearer traffic. Signaling and media traffic don't need to be in the same band and on the same channel, and in-band signaling is not required. Thus, communication with application servers is simplified.

Though VoIP is becoming more and more popular, there are still some challenging problems such as how to improve quality and robustness of VoIP service. VoIP quality still remains sensitive to performance degradation in the network [44].

1.3 VoIP Quality Metrics

In a well planned network, the Quality of Service (QoS) features in the network equipment intelligently distinguish and route traffic based on its priority. By helping to guarantee that voice traffic gets the bandwidth it needs, the network controls the factors that compromise voice quality. These factors are:

- Latency: As a delay-sensitive application, voice cannot tolerate too much delay. Latency is the average time it takes for a packet to travel from its source to its destination. The maximum amount of latency that a voice call can tolerate one way is 150 milliseconds (100 milliseconds is preferred) [1]. If there is too much traffic on the line, or if a voice packet gets stuck behind a bunch of data packets (such as an email attachment), the voice packet will be delayed to the point that the quality of the call is compromised.

- Jitter: In order for voice to be intelligible, consecutive voice packets must arrive at regular intervals. Jitter describes the degree of variability in packet arrivals, which can be caused by bursts of data traffic or just too much traffic on the line. Jitter is the delay variance from point-to-point. Voice packets can tolerate only about 75 milliseconds (40 milliseconds is preferred) of jitter delay [47].

- Packet loss: Packet loss due to congestion is the losing of packets along the data path, which severely degrades the voice quality. Packet loss occurs frequently in data networks, but many applications are designed to provide reliable delivery using network protocols that request a retransmission of lost packets (e.g. TCP [7]). Dropped voice packets, on the other hand, are discarded, not retransmitted. Voice traffic can tolerate less than a 3 percent loss of packets [32] before callers feel perceivable gaps in conversation.

When these factors are properly controlled by QoS mechanisms, VoIP delivers better quality voice than they are accustomed to from dedicated voice networks, even over the lower speed connections. At the same time, data applications are also prioritized and assured of their share of network resources.

Though understanding the elements that contribute to VoIP call quality is the key to successful IP calling, the dynamic interaction between packet queueing and bandwidth, etc that affects the

[1] http://www.csd.uoc.gr/ hy536/VoIP.pdf

quality of an individual connection should also be taken into account. For Internet Service Provider (ISP), regular measurement of these factors can help to ensure the integrity of voice traffic and high voice quality. For end users, monitoring the VoIP connection and obtaining values of these factors can help the ISP to more quickly accept and resolve the issue without consuming the user's own time. In this research, simulation experiments on these factors were also performed.

1.4 Motivation of This Study

The increasing expectation levels for better audio and video performance has led to the need to understand the behavior of audio and video traffic as it affects end user perceived quality of the application over the Internet. Most of the VoIP research [5, 16, 35, 40] has been focused on two open protocols: H.323 and SIP.

The main objectives of this study are:

- Relatively little is known about the traffic characteristics of the Cisco VoIP system in a deployed environment. This work seems to be one of the first real deployment studies of VoIP that doesn't rely on artificial traffic.
- The experimental data provided in this study is useful for design and modeling of such systems, from which more useful predictive models can be generated.
- A clear understanding of usage patterns in a real VoIP network is important for network deployment and potential network activities such as integration, optimizations or expansion.
- The major factors affect VoIP quality such as delay, jitter and packet loss were also measured and simulated in this study, which is helpful for advanced VoIP quality study.

1.5 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 provides the background information about popular VoIP protocols, VoIP systems and the standard data format of VoIP traffic. Chapter 3 outlines the related work in the following three areas: network performance and workload studies in a VoIP network, VoIP quality studies and some other issues in VoIP systems. In chapter 4, experiments design and results are presented. A simulation study on VoIP quality metrics is described in chapter 5. Chapter 6 summarizes the thesis and outlines possible future work.

Chapter 2

Background

Just as today's data networks were built using multiple protocols and applications, the VoIP networks are also constructed using the protocols and applications that best fit the associated technology and business requirements. In this chapter, some basic VoIP protocols and some well-known VoIP systems are introduced. Although this research is focussed on Cisco VoIP systems, understanding other protocols and their implementations in popular VoIP systems can help to better understand and investigate Cisco VoIP systems.

2.1 VoIP Protocols

There are two families of protocols relevant to this VoIP study: data transport and control. Both protocols operate above the transport layer (see Figure 2.1). The Real-Time Transport Protocol (RTP), along with its associated Real-Time Control Protocol (RTCP), is used in conjunction with H.323, SIP or other call signaling protocols, making it the technical foundation of the Voice over IP industry [38]. In the following sections, these VoIP specific protocols will be introduced in detail as well as the fundamental transport protocols on which they are built.

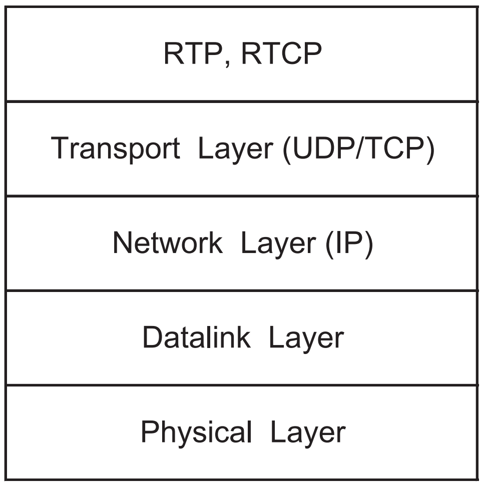


Figure 2.1: VoIP protocol structure

2.1.1 Internet Protocol – IP

IP stands for Internet Protocol [24]. It is responsible for the delivery of packets (or datagrams) between host computers. IP operates at the network layer (see Figure 2.1). It is a connectionless protocol, that is, it does not establish a virtual connection through a network before starting transmission. IP makes no guarantees concerning reliability, flow control, error detection or error correction. The result is that datagrams could arrive at the destination computer out of sequence, with errors, or not even arrive at all. Nevertheless, IP succeeds in making the network transparent to the upper layers involved in voice transmission through an IP based network. Transport layer protocols use IP services to provide various levels of service guarantees.

By definition, any Voice over IP transmission must use IP. As a real time application, voice transmission requires guaranteed connections with consistent delay characteristics. Many characteristics of IP, however, do not make it well-suited for voice transmission. Higher layer protocols address these issues. The focus of this thesis is on the higher layer protocols of VoIP; the study of lower level network protocols is not within the scope of this work.

In its most basic form, the IP header comprises 20 bytes. There are optional fields which can be appended to the basic header, but these offer additional capabilities which are not necessary for VoIP transmission.

2.1.2 Transmission Protocols – TCP & UDP

Generally, there are two protocols available at the transport layer when transmitting information through an IP network. These are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). Both protocols are associated with unique port numbers (for example, the HTTP application is usually associated with port 80).

TCP TCP stands for Transmission Control Protocol. It is a connection-oriented protocol that is responsible for reliable communication between two end processes. TCP enables two hosts to establish a connection and exchange data streams and guarantees that packets will be delivered in the same order in which they were sent. TCP was designed to dynamically adapt to properties of the internetwork and to be robust in case of failures.

The sending and receiving TCP entities exchange data in the form of segments. A segment consists of a fixed 20-byte header (plus an optional part) followed by zero or more data bytes. The source port and destination port specify the end points of the connection. The sequence number field identifies the first byte of data in this segment and the acknowledgement number contains the value of the next sequence number the sender of the segment is expecting to receive. The TCP header length field tells the TCP header length in 32-bit words. There are six 1-bit flags. The window size field tells how many bytes may be sent starting at the byte acknowledged. The checksum field checks a sum of the bytes in the header. With these information in the header, TCP provides reliable transmission between the two end points.

UDP UDP stands for User Datagram Protocol which is connectionless and unreliable. It has minimal overhead. Each packet on the network is composed of a small header and user data, and is called a UDP datagram. A datagram can be sent at any time without prior advertising, negotiation or preparation. UDP routes data to its correct destination port, but does not try to perform any sequencing, or to ensure data reliability in common with IP.

A UDP segment consists of an 8-byte header followed by the data. The two ports serve the same function as they do in TCP: to identify the end points within the source and destination machines. The UDP length field includes the 8-byte header and the data. With only these information in the header, UDP provides unreliable transmission between the two end points.

Voice is a real-time application, and mechanisms must be in place to ensure that information is received in the correct sequence, reliably and with predictable delay characteristics. Although TCP would address these requirements to a certain extent, there are some functions which are reserved for the layer above TCP. Therefore, some of TCP's functions must be reworked in some way to be more specific for VoIP as they aren't really used unmodified at the TCP layer. Also, the extra overhead of TCP and the possibility and high likelihood of increased latency make it unsuitable for real time applications. Therefore, for the transport layer, TCP is not used, and the alternative protocol, UDP, is commonly used.

2.1.3 Media Protocols – RTP & RTCP

Real time applications require mechanisms to be in place to ensure that a stream of data can be reconstructed accurately. Datagrams must be reconstructed in the correct order, and a means of detecting network delays must be in place.

Jitter is the variation in delay times experienced by the individual packets making up the data stream. In order to reduce the effects of jitter, data must be buffered at the receiving end of the link so that it can be played out at a constant rate. RTP and RTCP are protocols to support these requirements.

RTP RTP stands for Real-Time Transport Protocol. It is a protocol to carry data that has real-time properties. It is the main transport protocol used for IP Telephony media streams. RTP was defined in RFC1889 [38] and it defines a standardized packet format for delivering media over the Internet.

RTP provides end-to-end network transport functions to applications transmitting real-time data, such as interactive audio and video over multicast or unicast network services. The network services include payload type identification, sequence numbering, timestamping and delivery monitoring. Both RTP and UDP contribute parts of the transport protocol functionality. To make use of UDP's multiplexing and checksum services, RTP is generally run on top of UDP. However, RTP may be used with other suitable underlying network or transport protocols. If multicast distribution is provided by the underlying network, RTP can transfer data to multiple destinations [38].

RTP does not have a standard UDP port on which it communicates. The only standard that it obeys is that UDP communications use an even port and RTP Control Protocol (RTCP) communications are done on the next higher odd port. Although there are no assigned standards, RTP is generally configured to use ports in the range of 16384-32767. RTP only carries voice or video data. Call setup and tear-down is generally performed by the call signaling protocols such as the Skinny protocol.

The RTP header, which precedes the data payload, is depicted in Figure 2.2:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1--4	V=2		P	X	CC				M	PT							Sequence Number															
5--8	Timestamp																															
9--12	Synchronization Source Number(SSRC)																															

Figure 2.2: RTP packet header [38]

The header fields are here detailed [37]:

- Ver: Version. RTP version number. It is currently set to be 2.
- P: Padding. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit. If set, this packet contains one or more additional padding bytes at the end which are not part of the payload. The last byte of the padding contains the number of how many padding bytes should be ignored.
- X: Extension. If it is set, exactly one header extension follows the fixed header.
- CC: CSRC count. It records the number of CSRC identifiers that follow the fixed header.
- M: Marker. The interpretation of the marker is defined by a profile. Significant events such as frame boundaries are allowed to be marked in the packet stream. By changing the number of bits in the payload type field, a profile may define additional marker bits or specify that there is no marker bit.
- PT: Payload Type. This field specifies the format of the RTP payload and determines its interpretation by the application. A default static mapping of payload type codes to payload formats is specified by a profile. Additional payload type codes may be defined dynamically

through non-RTP means. A single RTP payload type is emitted by an RTP sender at any given time; this field is not intended for multiplexing separate media streams.

- Sequence Number. The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. In order to make known-plaintext attacks on encryption more difficult, the initial value of the sequence number is random (unpredictable), even if the source itself does not encrypt, because the packets may flow through a translator that does.

- Timestamp. The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations. The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient). The clock frequency is dependent on the format of data carried as payload and is specified statically in the profile or payload format specification that defines the format, or may be specified dynamically for payload formats defined through non-RTP means.

- SSRC: Synchronization source. This field identifies the synchronization source. This is an identifier field that is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC. Although the probability of multiple sources choosing the same identifier is low, all RTP implementations must be prepared to detect and resolve collisions. If a source changes its source transport address, it must also choose a new SSRC to avoid being interpreted as a looped source.

- CSRC: Contributing source. An array of 0 to 15 CSRC elements identifying the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources. For example, for audio packets, the SSRC identifiers of all sources that were mixed together to create a packet are listed, allowing correct talker indication at the receiver. (see figure 2.3).

Information in the RTP header tells the receiver how to reconstruct the data and describes how the codec bit streams are packetized.

RTCP RTCP stands for Real-time Transport Control Protocol. It provides the control services for a data stream that uses RTP. The main function of RTCP is to provide feedback on the quality of the transmission link. Other RTCP functions include carrying a persistent transport-level identifier for an RTP source, called the canonical name. The canonical name is used by receivers to synchronize audio and video and convey minimal session control information such as participant identification to be displayed in the user interface [38].

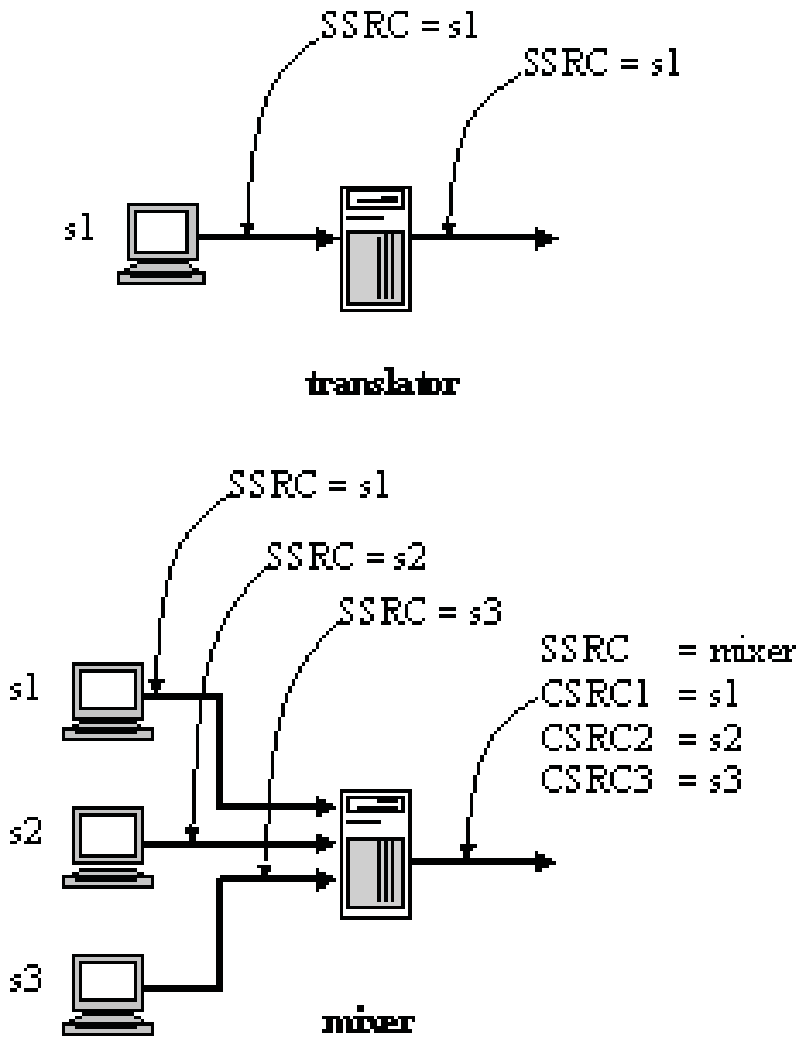


Figure 2.3: RTP header information

RTCP uses the same transport protocols as RTP to periodically transmit control packets to participants in a streaming multimedia session. As mentioned previously, every RTP channel using port number N has its own RTCP protocol channel with port number equal to N+1. It gathers statistics on a media connection and information such as bytes sent, packets sent, lost packets, jitter, feedback and round trip delay. An application may use this information to increase the quality of service perhaps by using a low compression codec instead of a high compression codec. RTCP is also used for QoS reporting.

The RTCP packet header is depicted in Figure 2.4:

The major fields of the RTCP header are:

- V,P. The same as for RTP packets.
- RC. A count of reception report blocks contained in this packet.
- PT. The packet type constant 201 designates an RTCP RR packet.
- Length. The length of this RTCP packet in 32-bit words minus one, including the header and any padding.
- SSRC. The synchronisation source identifier for the sender of this RR packet.

The data format of typical protocols that will be considered in this study are described in previous sections. As traffic data are captured from real network, the header information of trace packets should be the same as shown above.

2.2 VoIP Protocol Stacks

There are a few VoIP protocol stacks which are derived from various standards bodies and vendors, the most popular of which are H.323 and SIP.

2.2.1 H.323

H.323 [16] is an International Telecommunications Union (ITU) standard that provides a foundation for multimedia communication over networks that do not provide a guaranteed quality of service. H.323 specifies the protocols that provide real-time multimedia communication services such as audio, video and data communications over packet-switched networks [5]. As will be seen in subsequent sections, its structure is different from other VoIP systems though they all use RTP to transfer voice packets.

The H.323 standard is another fundamental technology for the transmission of real-time audio, video, and data communications over packet-based networks. It specifies the components, protocols, and procedures that provide multimedia communication over packet-based networks.

<table>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>1--4</td><td colspan="2">V=2</td><td>P</td><td colspan="5">RC</td><td colspan="8">PT=SR=200</td><td colspan="16">Length</td></tr>
<tr><td>5--8</td><td colspan="32">SSRC of Sender</td></tr>
<tr><td>9--12</td><td colspan="32">NTP Timestamp, Most Significant Word</td></tr>
<tr><td>13 - 16</td><td colspan="32">NTP Timestamp, Least Significant Word</td></tr>
<tr><td>17 - 20</td><td colspan="32">RTP Timestamp</td></tr>
<tr><td>21 - 24</td><td colspan="32">Sender's Packet Count</td></tr>
<tr><td>25 - 28</td><td colspan="32">Sender's Octet Count</td></tr>
<tr><td>29 - 32</td><td colspan="32">SSRC_1 (SSRC of First Source)</td></tr>
<tr><td>33 - 36</td><td colspan="8">Fraction Lost</td><td colspan="24">Cumulative Number of Packets Lost</td></tr>
<tr><td>37 - 40</td><td colspan="32">Extended Highest Sequence Number Received</td></tr>
<tr><td>41 - 44</td><td colspan="32">Interarrival Jitter</td></tr>
<tr><td>45 - 48</td><td colspan="32">Last SR (LSR)</td></tr>
<tr><td>49 - 52</td><td colspan="32">Delay since Last LR (DLSR)</td></tr>
<tr><td>53 - 56</td><td colspan="32">SSRC_2 (SSRC of Second Source)</td></tr>
<tr><td></td><td colspan="32">…</td></tr>
<tr><td></td><td colspan="32">Profile - Specific Extensions</td></tr>
</table>

(a) RTCP packet header – sender report(SR)

<table>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>1--4</td><td colspan="2">V=2</td><td>P</td><td colspan="5">RC</td><td colspan="8">PT=RR=201</td><td colspan="16">Length</td></tr>
<tr><td>5--8</td><td colspan="32">SSRC of Packet Sender</td></tr>
<tr><td>9--12</td><td colspan="32">SSRC_1 (SSRC of First Source)</td></tr>
<tr><td>13 - 16</td><td colspan="8">Fraction Lost</td><td colspan="24">Cumulative Number of Packets Lost</td></tr>
<tr><td>17 - 20</td><td colspan="32">Extended Highest Sequence Number Received</td></tr>
<tr><td>21 - 24</td><td colspan="32">Interarrival Jitter</td></tr>
<tr><td>25 - 28</td><td colspan="32">Last SR (LSR)</td></tr>
<tr><td>29 - 32</td><td colspan="32">Delay since Last LR (DLSR)</td></tr>
<tr><td>33 - 36</td><td colspan="32">SSRC_2 (SSRC of Second Source)</td></tr>
<tr><td></td><td colspan="32">…</td></tr>
<tr><td></td><td colspan="32">Profile - Specific Extensions</td></tr>
</table>

(b) RTCP packet header – receiver report(RR)

Figure 2.4: RTCP packet header [38]

There are four physical components in the H.323 standard: terminal, gateways (GW), gatekeepers (GK) and multipoint control units (MCU):

H.323 Terminal

H.323 terminal is an endpoint where H.323 data streams and signaling originate and terminate. It may be an IP telephone or a multimedia PC with an H.323 compliant stack that provides real-time two way communications.

Gateway (GW)

A Gateway is an optional component in an H.323-enabled network. An H.323 Gateway is an H.323 endpoint that provides translation between terminals belonging to networks with *different* protocol stacks, enabling the endpoints to communicate.

Gatekeeper (GK)

A Gatekeeper is a very useful but optional component of an H.323-enabled network. The gatekeeper provides several services such as address translation and network access control for the network resources to all endpoints in its zone. Also, it can provide other services such as bandwidth management, accounting and dial plans for scalability.

Multipoint Control Unit (MCU)

MCU is also an optional component of an H.323-enabled network and its basic function is to maintain all the audio, video data and control streams between all the participants in the conference. It is typically used for multiparty video conferences. The main components of an H.323 MCU are a mandatory Multipoint Controller (MC) and an optional Multipoint Processor (MP).

The protocols specified by H.323 are listed below [16]. H.323 is independent of the packet network. It does not specify the transport protocols over which it runs (see Figure 2.5)[1].

- audio CODECs. An audio CODEC encodes the audio signal from the microphone for transmission on the transmitting H.323 terminal and decodes the received audio code that is sent to the speaker on the receiving H.323 terminal. Since audio is the basic service provided by the H.323 standard, all H.323 terminals must have at least one audio CODEC support such as ITU-T G.711, G.723.1 or G.729.

- video CODECs. A video CODEC encodes video from the camera for transmission and decodes the received video code that is sent to the video display. Since video is an optional service provided by H.323, the support of video CODECs is optional as well.

- H.225 registration, admission, and status (RAS). Registration, admission, and status(RAS) is the protocol between endpoints (terminals and gateways) and gatekeepers. The RAS is used to perform registration, admission control, bandwidth changes, status, and disengage procedures between endpoints and gatekeepers.

[1] H.323, http://www.iec.org/online/tutorials/h323/topic04.html

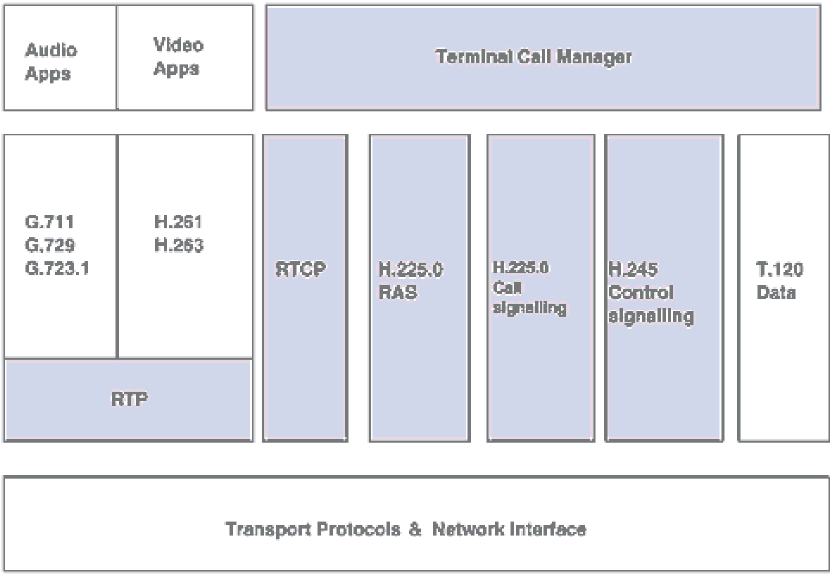


Figure 2.5: H.323 terminal-side protocol stack

- H.225 call signaling. The H.225 call signaling is used to establish a connection between two H.323 endpoints. This is achieved by exchanging H.225 protocol messages on the call-signaling channel.

- H.245 control signaling. H.245 control signaling is used to exchange end-to-end control messages governing the operation of the H.323 endpoint.

- real-time transfer protocol (RTP).

- real-time control protocol (RTCP).

This section introduced detailed information about the H.323 protocol. Though its structure is different from other VoIP systems such as Cisco VoIP system, they all use RTP protocol to transfer voice packets. So understanding the principles of these open VoIP protocols such as H.323 and SIP can help to better understand some specific VoIP systems.

2.2.2 SIP

SIP stands for Session Initiation Protocol. It is an Internet Engineering Task Force (IETF) standard protocol [35]. SIP is an application-layer control (signaling) protocol for initiating, manipulating, and tearing down sessions. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences [19]. SIP's main purpose is to help session originators deliver invitations to potential session participants. It is an alternative to H.323, but is a more lightweight and general-purpose, text-based protocol based on HTTP.

SIP makes use of proxy servers to help route requests to the user's current location, authenticates and authorizes users for services, implements provider call-routing policies, and provides features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers.

SIP is a request-response protocol, dealing with requests from clients and responses from servers. It is designed to address the functions of signaling and session management. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call. Such attributes are very important in a packet telephony network.

The peers in a SIP session are called User Agents (UAs). User Agents initiate and terminate sessions by exchanging requests and responses. A user agent can function as either a user agent client (UAC) or a user agent server (UAS).

Typically, a SIP endpoint can function as both a UAC and a UAS, but functions only as one or the other per transaction. The role in which an endpoint serves depends on the UA that initiated the request.

From an architecture standpoint, the physical components of a SIP network can be grouped into two categories: clients and servers.

SIP clients include phones, which act as either a UAS or UAC and gateways that provide call control. The functions of the gateway include translation between transmission formats and between communications procedures. SIP servers include proxy server, redirect server and registrar server. The proxy server is an intermediate entity that receives SIP requests from a client and then forwards the requests on the client's behalf. The redirect server is a server that accepts a SIP request, maps the SIP address and returns them to the client. The registrar server is a server which processes requests from UACs for the sake of registering and looking up their current location.

In current SIP or H.323 based Internet telephony client-server architectures, a registration server is employed for every domain [40]. The user agents in the domain register their IP addresses with the server so that the other users can reach them. Traditional redundancy and failover methods are used for scalability and reliability of such server-based systems. The majority of the system cost is in maintenance and configuration, so it is not easy to quickly set up the system in a small environment.

2.2.3 SCCP

SCCP stands for Skinny Client Control Protocol. It is a Cisco protocol standard for real-time calls and conferencing over Internet Protocol (IP). With the SCCP protocol, Cisco IP phones can co-exist in an H.323 environment. Skinny is a lightweight protocol that allows for efficient communication with Cisco Call Manager which may act as a proxy for signalling of call events with other common protocols such as H.323 and SIP.

A Skinny client uses TCP/IP to and from one or more Call Managers in a cluster so that the control information can be delivered reliably. RTP/UDP/IP is used to and from a similar Skinny client or H.323 terminal for the conveyed traffic (real-time audio stream).

With the SCCP architecture, the vast majority of the H.323 processing power resides in an H.323 proxy known as the Cisco Call Manager. The end stations (telephones) run what is called the Skinny Client, which consumes less processing overhead than H.323. The Client communicates with the Call Manager using connection-oriented (TCP/IP-based) communication to establish a call with another H.323-compliant end station. Once the Call Manager has established the call, the two H.323 end stations use connectionless (UDP/IP-based) communication for audio transmissions. Costs and overhead are thus reduced by confining the complexities of H.323 call setup to the Call Manager, and using the Skinny protocol for the actual audio communication into and out of the end stations. Table 2.1 shows some examples of Skinny packets format.

Table 2.1: Skinny protocol data format

Field name	**Type**	**Data**
skinny.CallStateMessage	Unsigned 32-bit integer	Call State (code), Line Instance
skinny.OpenReceiveChannel	Unsigned 32-bit integer	Receive Channel Details
skinny.KeepAliveMessage	Unsigned 32-bit integer	– (sent periodically by phone)
skinny.calledParty	String	Called Party Id
skinny.calledPartyName	String	Called Party Name
skinny.StartMediaTransmission	Unsigned 32-bit integer	Transmission Channel Details
skinny.callingPartyName	String	Calling Party Name
skinny.CloseReceiveChannel	Unsigned 32-bit integer	Conf Id, Pass Through Party Id
skinny.StopMediaTransmission	Unsigned 32-bit integer	Conf Id, Pass Through Party Id

A Skinny packet is made up of 4 fields. The first field is length of message. The second field is reserved (0x00000000). The third field denotes message type. The rest of the message is variable depending on the message type.

2.3 VoIP Codec

A codec (coder/decoder) converts from a sampled digital representation of an analog signal to a compressed digital bitstream, and another identical codec at the other end of the communication converts the digital bitstream back into an analog signal. In a VoIP system, the codec used is often referred to as the encoding method, or the payload type for the RTP packet. Codecs generally provide a compression capability to save network bandwidth. Some codecs also support silence suppression, where silence is not encoded or transmitted. Three primary factors to be optimized are the speed of the encoding/decoding operations (packetization delay), the quality and fidelity of sound and/or video signal, and the size of the resulting encoded data stream. Table 2.2 shows the basic features of representative ITU standard codecs.

Table 2.2: VoIP codec comparison

Codec	**Algorithm**	**Data Rate**	**Packetization Delay**
G.711	PCM	64.0 Kbps	1.0 msec
G.723	Multi-rate Coder	5.3 and 6.3 Kbps	67.5 msec
G.729	CS-ACELP	8.0 Kbps	25.0 msec

G.711 represents logarithmic Pulse-Code Modulation (PCM) samples for signals of voice frequencies, sampled at the rate of 8000 samples/second. Eight binary digits per sample are used. Thus, the G.711 encoder will create a 64Kbps bitstream.

The G.723 [14] speech coder recommendation was developed for use in multimedia platforms, in particular those specified by the H.32x series recommendations. It provides two compressed stream bit rates, 5.3 Kbps and 6.3 Kbps. The higher bit rate is of greater quality. It was optimized to represent high quality speech with low bandwidth requirement using a limited amount of complexity.

G.729 [15] is an 8Kbps Conjugate-Structure Algebraic-Code-Excited-Linear-Prediction (CS-ACELP) speech compression algorithm approved by ITU-T. It offers toll quality speech at a reasonably low bit rate of 8Kbps. G.729 allows moderate transmission delays, so applications such as teleconferencing or visual telephony where quality, delay and bandwidth are important, will benefit from this standard.

2.4 VoIP Quality Estimation Models

l The E-Model defined in the ITU-T Recommendation G.107 [13] is an analytic model for prediction of VoIP quality based on network impairment parameters such as packet loss and delay. The E-Model provides an objective method of assessing the transmission quality of a telephone connection. The E-Model results in an R factor ranging from a best case of 100 to a worst case of 0. The R-factor uniquely determines the Mean Opinion Score (MOS) [21]. The MOS provides a numerical indication of the perceived quality of received media after compression and/or transmission. It is expressed as a single number in a scale of 1 to 5, where 1 is the lowest perceived quality, and 5 is the highest perceived quality.

PSQM (Perceptual Speech Quality Measurement) [31] has been adopted by ITU-T to assess the speech quality for codecs. PSQM+ was proposed to improve the performance of PSQM for loud distortions and temporal clipping and it provides a more accurate measurement of perceived speech quality under frame loss situations [42].

2.5 VoIP Systems

There are three different types of VoIP service in common use today:

- ATA - A simple and most common way is through the use of a device called an ATA (analog telephone adaptor). A standard phone can be connected to a computer or an Internet connection for use with VoIP through an ATA.

- IP Phones - These specialized phones look just like normal phones. But instead of having the standard RJ-11 phone connectors, IP phones have an RJ-45 Ethernet connector which connects to a switch or router.

- Computer-to-computer - This is certainly the easiest way to use VoIP. The only need is a soft phone software running on the computers and an Internet connection.

There are several VoIP systems developed. Skype is one of the most popular systems among them.

2.5.1 Skype

Skype [2] is a peer-to-peer VoIP client system developed by KaZaa[2] that allows its users to make voice calls and send messages to other users of Skype clients. It encrypts calls end-to-end, and stores user information in a decentralized manner.

Skype is a proprietary peer-to-peer internet telephony (VoIP) network which competes against established open VoIP protocols like SIP or H.323. The system has a reputation for its broad range of features and its ability to use P2P technology to overcome common firewall and NAT problems. It does this by routing voice packets through firewalls or NAT's with no special configuration. Skype users can speak to other Skype users, use SkypeOut to call landlines and mobile phones, use SkypeIn to receive calls from any phone, and also receive voicemail messages.

The main difference between Skype and other VoIP clients is that it operates on a peer-to-peer model rather than the traditional server-client model. The Skype user directory is entirely decentralised and distributed among the nodes in the network, which means the network can scale very easily to large sizes (over forty million users) without a complex and costly centralised infrastructure.

The key components in Skype software are:

[2]Kazaa, Peer-to-peer file sharing software application, http://www.kazaa.com

- Ports. A Skype client opens a TCP and a UDP listening port at the port number which is configured in its connection dialog box.
- Host Cache. The host cache, a list of super node IP address and port pairs that Skype client builds and refreshes regularly, is the most critical part to the Skype operation.
- Codecs. Skype uses iLBC[3] or iSAC[4].
- Buddy List. Skype stores its buddy information in the Windows registry.
- Encryption. Skype uses AES to encrypt information.
- NAT and Firewall. A Skype client uses a variation of the STUN [36] and TURN [34] protocols to determine the type of NAT and firewall it is behind.

However, Skype has its own limitations [40]:

1. The protocal is Skype proprietary, unlike open standards such as SIP.
2. It provides a single service and not an architecture for new services.
3. It has centralized elements for login authentication, so if this element fails the system may not work.

The architecture of Skype is quite different from other VoIP systems such as H.323, SIP and Cisco VoIP systems because it operates on a peer-to-peer model and it is almost entirely decentralised. In part, its ability to traverse NATs and firewalls makes it a successful peer-to-peer VoIP system.

2.6 Summary

In this chapter, some general VoIP protocols and some well known VoIP systems were introduced. Understanding the basic concepts such as function, architecture, packet format, header information of general VoIP protocols and other VoIP systems can be very helpful for research on a specific VoIP system and specific protocols in the system such as the Cisco VoIP system, which will be discussed in detail in Chapter 4.

[3] iLBC codec, Free speech codec suitable for robust voice communication over IP, http://www.globalipsound.com/pdf/gips_iLBC.pdf

[4] iSAC codec, An unrelated wideband variable rate codec, http://www.globalipsound.com/pdf/gips_iSAC.pdf

Chapter 3

Related Work

There are three major areas of VoIP study. First, network performance and workload studies for a VoIP system such as VoIP protocol analysis and traffic analysis which aid to design and model VoIP system; second, VoIP quality measurement and service availability studies which provide insight as to how well current VoIP networks work; third, other issues in VoIP system such as VoIP protocol optimizations to improve VoIP quality. As well, a lot of network performance and workload studies have been conducted in the Internet and wireless network field. Similar analysis methods can be used in VoIP networks. Discussion of some of these works are presented in Section 3.1. Section 3.2 describes VoIP quality measurement and factors affecting VoIP quality. VoIP quality can be measured through subjective assessment and objective assessment. Subjective measurement is conducted by real human beings while objective measurement is conducted through tools. The major factors that affect VoIP quality are network delay, packet loss and jitter. Several studies on how these factors affect VoIP quality will be discussed later. Quality service availability also need to be considered; and it will be discussed later as well. Section 3.3 discusses some other issues in VoIP system such as methods used to improve VoIP quality.

3.1 Network Performance/Workload Studies

Understanding network performance or workload can help better manage and optimize networks. One approach to measuring network performance is to analyse network traffic. The source of measurement data can be collected from a LAN or Internet. There is a difference, however, between the LAN and Internet in terms of delay, jitter and loss. In the LAN environment, the delay, jitter and loss values configured on the traffic emulation tools were the overall values in the end-to-end path; while in the Internet tests, network paths had inherent values of delay, jitter and loss [4].

In Thompson, Miller & Wilder [45], the authors presented observations on the patterns and characteristics of wide-area Internet traffic, as recorded by MCIs OC-3 traffic monitors. They reported on measurements from two OC-3 trunks in MCIs commercial Internet backbone over two time ranges (24-hour and 7-day) in the presence of up to 240,000 flows. They revealed the characteristics of the traffic in terms of packet sizes, flow duration, volume, and percentage composition by protocol and application, as well as patterns seen over the two time scales.

In Schwab & Bunt [39], the authors presented the results of an analysis of campus-wide wireless network of the University of Saskatchewan. The analysis was based on a week-long traffic trace collected on the wireless network, which was recording every packet sent and received. The network performance of the campus wireless netwok can be determined from the trace data. There were four stages of analysis in this study. The initial analysis of the trace files was a simple validation and error-checking pass. This revealed mis-ordered or erroneous data in the trace files and the problems were resolved by sorting the mis-ordered files and removing the erroneous data. The second stage of the analysis focused solely on the aggregate traffic patterns and the protocol mix. The third stage of the analysis focused on the authentication log. Looking at authentication times, network card addresses and access points addresses revealed user behavior on the wireless network. The final stage of analysis combined the trace data and authentication log to match packets with access points.

In Balachandran *et al.* [1], the authors presented and analysed user behavior and network performance in a public-area wireless network using a workload captured at a conference. The aim of this work was to extend understanding of wireless user behavior and wireless network performance. The work also characterized wireless users in terms of a parameterized model for use with analytic and simulation studies involving wireless LAN traffic. The workload analysis results can be applied to better understand issues in wireless network deployment and potential network optimizations.

In Kotz and Essien [23], the authors presented results from the largest and most comprehensive trace of network activity at that time in a large, production wireless LAN. This work can help understand usage patterns in wireless local-area networks(WLANs) which is critical for those who develop, deploy, and manage WLAN technology, as well as those who develop systems and application software for wireless networks.

Similar methodology can be used on a VoIP network for three reasons. First, modern methods and tools, which are used in this study, were utilized in those studies; second, similar kinds of deployment are being used; third, presentation in those studies helps to figure out what kinds of results are significant. However, there are still different aspects such as wired or wireless network, and comprehensive use of collection facilities. As to the newly deployed VoIP network in the Department of Computer Science at the University of Saskatchewan, a similar study can be done to answer questions such as "how many calls are there across hours during a day", "how many calls are there across days during a week", "how is call duration distributed", "how does voice traffic vary across hours, days", "what is bandwidth utilization for VoIP traffic", etc.

3.2 VoIP Quality Studies

There are two popular approaches to determine the quality of voice calls: subjective quality assessment and objective quality assessment. The subjective quality assessment approach is to use human subjects to evaluate signal quality. The methodology presented in ITU-T is based on conversation opinion tests. The 5-point quality scale of bad-poor-fair-good-excellent (MOS) is recommended for assessing the audio and video quality [28]. However, this solution is very expensive and time consuming, especially when there are many different experiments to be conducted. Objective quality assessment approaches use a machine-executable algorithm to decide the quality of a received voice signal by comparing it with an original transmitted signal [46]. A feasible method to evaluate VoIP quality in a lab environment is to use commercially available test equipment or test software, which use objective methods to subjectively measure voice signals. Although objective signal quality measurement equipment, instead of human subjects, can be used to evaluate voice signal quality in a lab network, this solution can still be time consuming and expensive [10]. To reduce the time and costs of measuring voice quality in a lab environment, an alternate approach is to replace the test VoIP network by a network emulator. In Tao, Apostolopoulos & Guerin [43], the authors presented an approach which offers a lightweight video quality monitoring solution that is suitable for large-scale deployments.

3.2.1 Real World Studies about VoIP Quality

The major network health factors that affect the quality of VoIP sessions are network delay (latency), packet loss and packet jitter. In Calyam *et al.* [5], the authors state that their goal for the work is to obtain the performance bounds for network metrics delay, jitter and loss in terms of Good, Acceptable and Poor grades of H.323 applications based upon objective and subjective quality assessment techniques on different audio and video streams.

A previous study presented the results of 15 weeks of voice over IP measurements consisting of over 18000 recorded VoIP sessions taken from nine fully meshed sites [27]. The method they used to measure VoIP quality is to transmit pre-recorded calls between globally distributed sites on a hourly basis. The goal of this work was to obtain quality measures based solely on network variations. According to the ITU standard, the end-to-end one way delay should not exceed 150ms and the mean packet loss should not exceed 10% before glitches. Based on ITU standard and from the results presented in Marsh, Li & Karlsson [27], it can be concluded that the quality of VoIP is exceeding the requirements of many speech quality recommendations. This is attributed to more capacity and better network management. In Ding & Goubran [11], the authors investigated the impairments from packet loss and delay jitter on speech quality in VoIP using the extended E-Model which achieved good accuracy.

Another area of research on characterizing voice and video traffic behavior was based on large-scale Internet Videoconferencing systems [4]. The method they used was to collect Videoconferencing traffic traces from a world-wide voice and video traffic quality measurement testbed, which covered several sites all over the world via disparate network paths on the Internet. They collected more than 300 voice and video traffic traces by performing one-on-one testing in a LAN environment and on the Internet with over 26 sites. They analysed the end-to-end performance variations by characterizing packet size distributions, packet delay, jitter, loss and reordering for both audio and video streams. Besides the major network health factors which affect the quality of VoIP, the authors also considered multiple end-point technologies such as codecs, vendors, pc-based vs. appliance-based equipment and diverse end-user demographics to obtain subjective and objective audiovisual quality assessments. Since traffic characteristics in terms of packet-size distributions have implications for the end-to-end performance achieved by the traffic streams, a good understanding of how various packet size distributions are handled by the network could help in determing important trade-offs at the application level.

In the same work [4], the authors pointed out that packet reordering is another reason to cause performance bottlenecks such as lack of lip-synchronization. Lack of lip-synchronization refers to the case that the audio and video streams are not synchronized at the playback time causing a perceived mismatch in the audio and video content. The existence of high values of reordering could make the VoIP traffic more non-deterministic at the end-points because of increased network jitter. The non-deterministic nature impacts the buffering and playback of the audio and video streams smoothly. So it will affect the end-user perception of audiovisual quality ultimately.

The Cisco VoIP system measures network delay, jitter and packet loss for every call automatically. Thus, call quality can be assessed objectively from these data. Based on the measured data, a technique may be discovered to improve VoIP quality such as optimizing network structure.

3.2.2 Simulation Studies about VoIP Quality

As described earlier, subjective quality assessment and objective quality assessment are two popular ways to determine the quality of voice calls. Some work has been done in the area of the subjective human evaluation of packet-switched voice calls as a function of virtual circuit QoS [9, 25, 33]. Other research about the objective quality analysis of voice in packet networks as a function of network QoS has also been conducted [6, 10, 8, 17, 20, 41, 42].

In Clark [6], a tool named VQmon was developed to estimate the transmission quality of VoIP calls. It used an extended version of the ITU G.107 E-model which made it much simpler than the existing objective methods such as Perceptual Speech Quality Measurement (PSQM) [31]. Thus, it is suitable for the non-intrusive real-time passive monitoring of VoIP calls. Another methodology for non-intrusive passive monitoring was presented in Conway [8].

In Hooper & Russell [17], the authors use an automatic speech recognition (ASR) system to evaluate VoIP quality for G.723 and G.729 codecs. Results are presented for the recognition accuracy as a function of various packet loss rates and packet sizes though it is not related to a subjective score.

In Sun *et al.* [42], the effect of the location of packet loss on perceived speech quality was studied. A simulation system which used reference speech files and objective methods of speech quality analysis was implemented. In Sun [41], the authors carried out a fundamental investigation of the impact of packet loss and talkers on perceived speech quality using an objective method to provide the basis for developing an artificial neural network (ANN) model to predict speech quality for VoIP.

The same simulation approach was adopted in Conway & Zhu [10]. The simulation-based methodology in this work was made in the more general context of the development of a general purpose VoIP engineering tool which could conduct the analysis of a network QoS model along with various implementation and configuration choices such as codec type, packetization method, buffering mechanism, and playout algorithm. In the simulation tool, an artificial voice signal is encoded into a digital signal following an encoding procedure such as G.711 or G.729. The encoded digital signal is then packetized into RTP/UDP/IP packets. The packets are transmitted to the network model. The network model introduces simulated QoS impairments such as packet loss and packet delay jitter into the transmitted packet stream. The packets then arrive at a simulated destination where they are buffered and played out according to a specified play-out algorithm. The packet payloads are depacketized to form the received encoded digital signal. The encoded digital signal is then decoded to the final voice signal based on the given codec type. The subjective quality of this received signal is obtained by comparing the final received signal with the original voice signal using an objective quality measurement algorithm such as PSQM. The output of the objective quality measurement algorithm can be transformed by a mapping function to provide a subjective MOS score [30].

Simulation study on VoIP quality in a Cisco VoIP system was also conducted in this research, finding out the degree of success and remaining issues in this area can help to make better experiment plan and obtain better results.

3.2.3 Availability of a VoIP Service

Voice quality is not the only metric of interest for evaluating the feasibility of a VoIP service. The availability of the service also covers a fundamental role. Internet Service Providers (ISPs) need to provide a comparable quality both in terms of voice quality and availability of the service. If VoIP were to successfully replace the PSTN, it has to meet several stringent requirements, in particular high service availability.

In Jiang & Schulzrinne [18], the authors evaluated the service availability of VoIP achieved on the current Internet. They used call success probability, overall packet loss probability, the proportion of time the network is suitable for VoIP service, and call abortion probability induced by network outages as the main metrics. They found that packet losses are not rare events, especially on international paths and network outages cause a non-negligible portion of packet losses. There are three major causes of potential degradation of performance for VoIP services: network congestion, link failures and routing instabilities. Link failures impact not only the performance, but also the availability of VoIP services. In Boutremans, Iannaccone & Diot [3], the authors discovered that long periods of routing instability may follow link failures, during which packets can be dropped because of being forwarded along invalid paths. Such instabilities can last for tens of minutes and induce the loss of reachability of a large set of end-hosts.

Though the availability of VoIP service is not within the scope of this research, it points out the fundamental requirement for VoIP system, which is significant for VoIP study.

3.3 Other Issues in VoIP Systems

In Tao *et al.* [44], the authors evaluated the effectiveness and benefits of path switching in improving the quality of VoIP applications, and demonstrate its feasibility through the design and implementation of a prototype gateway. This study suggested that by exploiting the inherent path diversity of the Internet, application-driven path switching is a feasible way to provide quality-of-service to applications. With the deployment of increasingly disparate technologies such as wireless networks, Internet paths will become more distinct. It is an important way to improve the quality-of-service of applications to intelligently create and exploit path diversity via mechanisms such as overlays and dynamic path switching.

Some other studies also take into account the effect of the different components of a VoIP system on VoIP quality, such as the playback buffer component [26]. The playback buffer is at the receiving end, whose purpose is to absorb variations in delay and provide a smooth playout. In order to have enough packets buffered to be played out continuously, arriving packets are held until a later playout time. The playback buffer sends a continuous stream of packets to the depacketizer and eventually to the decoder which reconstructs the speech signal. The playback buffer may operate in fixed mode or adaptive mode. In fixed mode, a fixed scheme schedules the playout of a packet after a fixed delay from its sending time and this happens to all packets. In adaptive mode, an adaptive scheme dynamically adapts the playout time to follow the variations in network delays. An appropriate choice of the fixed playout buffer is the one that leads to maximum MOS (Mean Opinion Score) while a good adaptive scheme should also operate around the maximum MOS. Since an adaptive scheme learns, predicts and follows the network delays as closely as possible, it can keep both delay and loss low. When the mechanisms of an adaptive scheme are carefully tuned to match the network path, it can perform at least as well as a fixed one. So the appropriate choice of playout scheme for each path is a key factor for the end-to-end quality.

The above pointed out that exploiting dynamic path switching and adaptive mode playback buffers can improve the quality of VoIP systems. Improving the quality of VoIP systems running on the current Internet network is a pivotal and challenging topic. As the characteristics of VoIP traffic in Cisco VoIP system were analysed in this study, the experimental data provided may shed some light on how to improve the performance and quality of Cisco VoIP systems, then extend to general VoIP systems. These are beyond the scope of this research, but they provide some interesting concepts that can be used.

Chapter 4

Experiment Setup and Initial Results

In this study, experiments in two LAN environments were performed. The preliminary experiments used a soft phone system named 'Vbuzzer', through which elementary understanding about a VoIP system and validation of the experimental setup was obtained. Then an extended experiment was performed in a deployed Cisco VoIP system in the Department of Computer Science at the University of Saskatchewan. Different tools were used to capture the VoIP traffic in the two experiments. After the captured data was decoded, detailed information in the trace file was available for analysis. Table 4.1 shows the typical fields contained in a captured packet. The following sections introduce the experiment environment and tools used. Then the analysis of the distribution of packet size, distribution of interarrival time, traffic volume, protocol mix information for both experimental systems are presented. Finally some implications and insights are discussed. Experimental environment, data capture tools and analysis software in Vbuzzer system are described in Section 4.1. Details about experiments conducted in Cisco VoIP system including methods and results are discussed in section 4.2.

Table 4.1: All fields in a trace file

Captured_Fields_Name	**Meaning**
No.	packet number
Time	the timestamp for this packet
Source	the source IP address of this packet
Destination	the destination IP address of this packet
Protocol	the protocol type of this packet
Info	the information included in this packet

4.1 Vbuzzer

4.1.1 Experiment Environment

In this experiment, VoIP traffic was analysed using softphone software named Vbuzzer. Vbuzzer is an Internet Telephony software which is not designed to simply duplicate the traditional telephone system. It is a suite of instant messaging and internet telephony software. Users can send and receive voice calls to or from other Vbuzzer users on the Internet. Vbuzzer uses the SIP protocol.

A simple testbed (shown in Figure 4.1) was setup to collect the traffic on a Vbuzzer system. Vbuzzer was installed on two computers connected to the Internet. One is used to initiate calls and the other one receives calls.

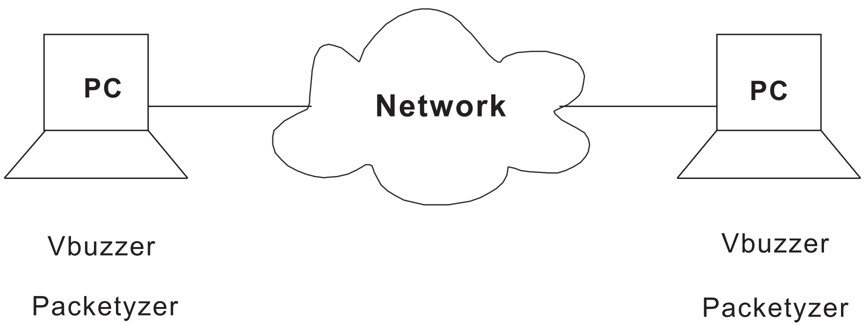


Figure 4.1: Vbuzzer testbed setup

4.1.2 Data Capture Tools

Packetyzer[1] was the primary data capture tool used in this experiment. Packetyzer was executed on both computers to collect a trace of data packets generated by Vbuzzer. The VoIP traffic was captured between phone and computer, computer and computer respectively. Packetyzer provides a Windows user interface for the Ethereal[2] packet capture and dissection library. Packetyzer can decode more than 483 protocols. Packetyzer's ability to support a number of protocols including SIP and RTP allows users to analyse packet patterns in greater detail.

[1]Network Chemistry, Packetyzer, http://www.packetyzer.com
[2]Ethereal, Ethereal Software, http://www.ethereal.com

4.1.3 Analysis Software

Ethereal

Ethereal was the primary protocol analysis software used to decode the trace file. From the decoded data, call setup processes and traffic information such as traffic volume, distribution of interarrival time, and protocol mix of VoIP traffic can be determined.

Ethereal is a popular network analyzer widely used by network professionals for troubleshooting, analysis, software and protocol development, and teaching. It reads packets from either the network or a trace file, decodes them, and presents them in an easy to understand format (see Figure 4.2). Ethereal was chosen as the tool to use because it is an actively-maintained open source program and its graphical user interface is very configurable and easy to use. Following are the primary features to be utilized from Ethereal:

- It can capture data from the network or read from a captured file.
- It supports Tcpdump format capture filters.
- It runs on over 20 OS platforms, both UNIX-based and Windows.
- It supports over 480 protocols, and because it is open source, new ones are contributed very frequently.

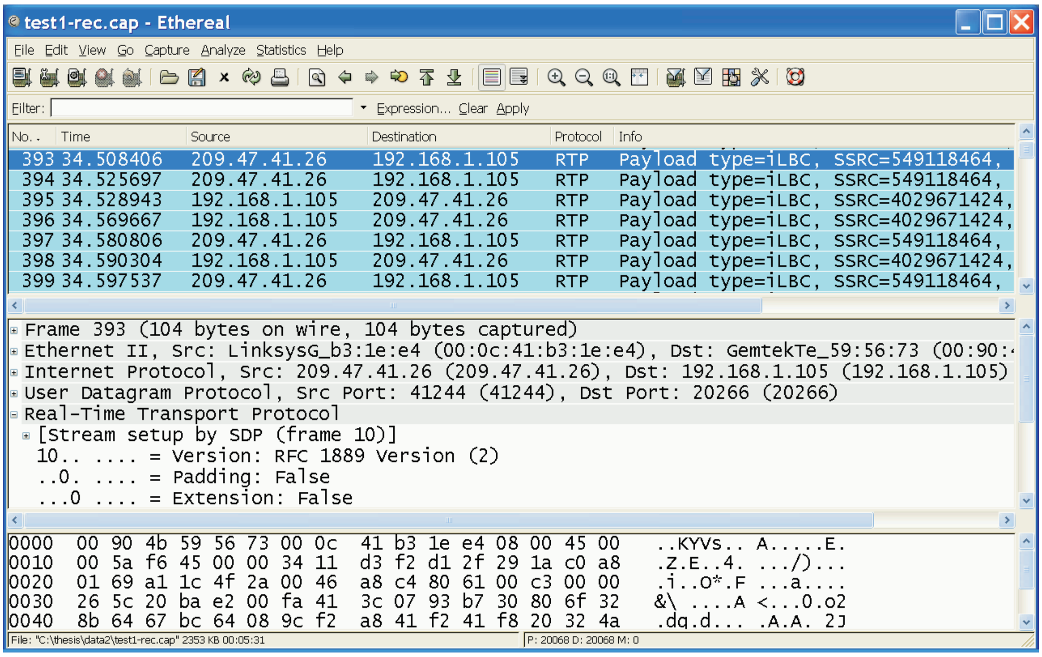


Figure 4.2: Ethereal

The name of the Ethereal product family has recently changed to Wireshark[3].

Perl

Perl was used to analyse information contained in the packet trace because of its ability to manipulate text easily. The algorithm used to analyse the packets is outlined in Figure 4.3.

```
While (Read Next Trace Line)
  Split Trace Line into Packet Fields
  Get Packet Type Field
  If (this Packet Type exists in Packet Type Array)
      Count as this Packet Type
  Else
      Push this Packet Type into Packet Type Array
      Count as this Packet Type
  Add Packet Length to total Packet Length of this Packet Type
  Add Interarrival Time to total Interarrival Time of this Packet Type
  Count total packets for each Packet Type
  Count minimal/average/maximal Packet Length for each Packet Type
  Count minimal/average/maximal Interarrival Time for each Packet Type
```

Figure 4.3: Trace analysis algorithm in Vbuzzer

4.1.4 Results

Call Setup Process

Call setup is a fundamental component of a VoIP system. It is also very complicated. Understanding call setup process is important for analysing the characteristics of a VoIP system. It is especially helpful to perform accurate simulation experiments. As seen in the captured and decoded data, different VoIP signaling protocols are used in different VoIP systems during a call setup processes, but after that, most systems use RTP to transfer voice data. The packet format for different call signaling protocols is quite different. Tables 4.2 and 4.3 display call setup process from one computer to another computer in Vbuzzer.

[3] http://www.wireshark.com

Table 4.2: Call setup process from computer to computer (callee)

No.	**Protocol**	**Info**
1	SSLv3	Encrypted Alert
2	TCP	https > 1529 [FIN, ACK] Seq=23 Ack=0 Win=32804 Len=0
3	TCP	1529 > https [ACK] Seq=0 Ack=24 Win=50019 Len=0
4	UDP	Source port: 6656 Destination port: 80
5	UDP	Source port: 6656 Destination port: 80
6	UDP	Source port: 6656 Destination port: 80
7	SIP/SDP	Request: INVITE sip:xxxxxx@70.64.8.9:6656; (receive) transport=udp, with session description
8	SIP	Status: 100 Trying (send)
9	SIP	Status: 180 Ringing (send)
10	SIP/SDP	Status: 200 OK, with session description (send)
11	SIP	Request: ACK sip:xxxxxxx@70.64.8.9:6656 (receive)
12	RTP	Payload type=iLBC, SSRC=4029671424, Seq=1, Time=2600
13	RTP	Payload type=iLBC, SSRC=4029671424, Seq=2, Time=2650
14	RTP	Payload type=iLBC, SSRC=549118464, Seq=3, Time=220
15	RTP	Payload type=iLBC, SSRC=549118464, Seq=4, Time=270

Table 4.3: Call setup process from computer to computer (caller)

No.	**Protocol**	**Info**
21	TCP	1066 > https [SYN] Seq=0 Len=0 MSS=1460
22	TCP	https > 1066 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
23	TCP	1066 > https [ACK] Seq=1 Ack=1 Win=17520 Len=0
24	SSLv3	Client Hello
25	SSLv3	Server Hello, Change Cipher Spec, Encrypted Handshake Message
26	SSLv3	Change Cipher Spec, Encrypted Handshake Message
27	SSLv3	Application Data
28	TCP	https > 1066 [ACK] Seq=147 Ack=391 Win=65535 Len=0
29	SSLv3	Application Data
30	TCP	1066 > https [ACK] Seq=391 Ack=147 Win=17374 Len=0 SLE=367 SRE=425
31	SSLv3	Application Data
32	TCP	1066 > https [ACK] Seq=391 Ack=425 Win=17096 Len=0
33	SIP/SDP	Request: INVITE sip:1416xxxxxxx@209.47.41.24:80, (send) with session description
34	SIP	Status: 100 Stop retransmission (receive)
35	SIP	Status: 401 Unauthorized (receive)
36	SIP	Request: ACK sip:1416xxxxxxx@209.47.41.24 (send)
37	SIP/SDP	Request: INVITE sip:1416xxxxxxx@209.47.41.24:80, (send) with session description
38	SIP	Status: 100 Stop retransmission (receive)
39	SIP	Status: 100 trying – your call is important to us (receive)
40	SIP	Status: 180 Ringing (receive)
50	UDP	Source port: 5188 Destination port: 80
56	SIP/SDP	Status: 200 OK, with session description (receive)
57	SIP	Request: ACK sip:xxxxxx@70.64.8.9:6656 (send)
58	RTP	Payload type=iLBC, SSRC=549118464, Seq=1, Time=120
59	RTP	Payload type=iLBC, SSRC=549118464, Seq=2, Time=170

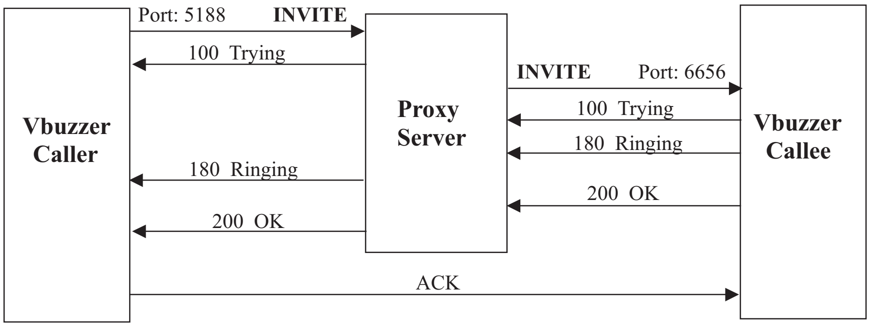


Figure 4.4: SIP call setup process in Vbuzzer

Figure 4.4 shows the SIP call setup procedure. When a client computer (caller) wants to initiate a call, it sends an INVITE message to the proxy server. When the proxy server receives the INVITE request, it sends a **100 (Trying)** response back to the caller's softphone. The **100 (Trying)** response indicates that the INVITE has been received by the proxy and that the proxy is working to route the INVITE to the destination. After the callee's softphone receives the INVITE, it sends **100 (Trying)**, then rings. The callee's softphone indicates this in a **180 (Ringing)** response, which is routed back to the caller through the proxy server in the reverse direction. If the callee answers the call, his/her softphone sends a **200 (OK)** response to indicate that the call has been answered. In this case, the **200 (OK)** is routed back through the proxy server and is received by the caller's softphone, which then stops the ringback tone and indicates that the call has been answered. Finally, the caller's softphone sends an acknowledgement message, ACK, directly to the callee's softphone to confirm the reception of the final response **(200 (OK)),** bypassing the proxy server. This completes the INVITE/**200**/ACK three-way handshake used to establish SIP sessions.

Statistics and Data

The information of the trace files in Vbuzzer systems is summarized in Table 4.4. In this experiment, a trace of ten calls has been collected with approximately five minutes of voice data for each call. The traffic was captured from five incoming calls and five outgoing calls.

Table 4.4: Overall statistics for the trace

Attribute	**Vbuzzer system**
Total Packets	208013
Total Seconds	3291
Average Traffic	63.2068 packets per second
Total Time	54 minutes, 51 seconds

Distribution by Packet Size

Understanding traffic characteristics in terms of packet-size distributions is important because it has implications for the end-to-end performance achieved by the traffic streams [4].

Figure 4.5 shows the distribution of packet size in Vbuzzer. From this figure, it can be found that the size of most packets in Vbuzzer is between 65 bytes and 128 bytes. This is because RTP packets occupy 99.6% of VoIP traffic and RTP packets have a fixed size of 104 bytes per packet.

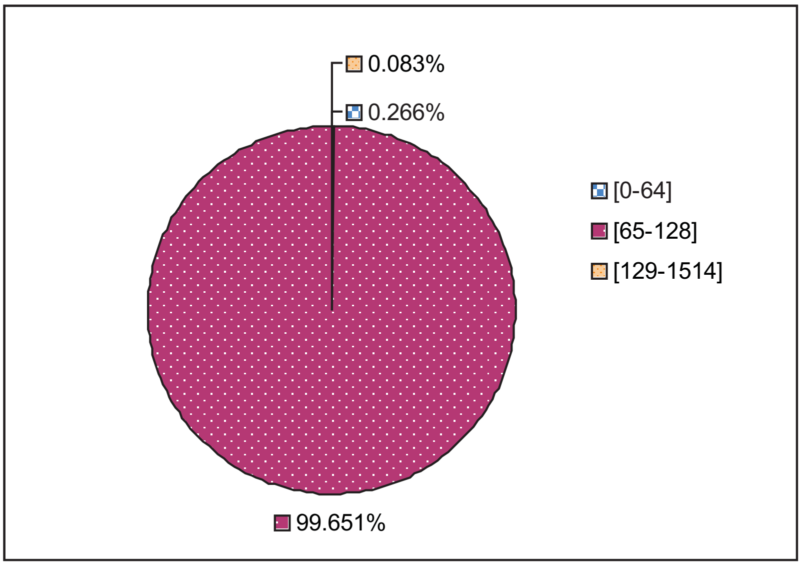


Figure 4.5: Distribution of packet size in Vbuzzer

Distribution by Packet Type

Figure 4.6 and Figure 4.7 show the protocol mix for VoIP traffic in Vbuzzer. By packet count, over 99.6% of the VoIP traffic is made up of RTP packets. The remaining 0.4% is made up of UDP packets, TCP packets and SIP packets. By byte count, the traffic is still dominated by RTP traffic (99.3%) followed by SIP packets, TCP packets and UDP packets. Figure 4.8 and Figure 4.9 display the protocol mix of signal control traffic in Vbuzzer. From both figures, it can be seen that SIP is the major voice signal control protocol.

Distribution of Interarrival Time

Interarrival time is the time difference between two consecutive packets. The Complementary Cumulative Distribution (CCDF) of interarrival time of voice packets in Vbuzzer is displayed in

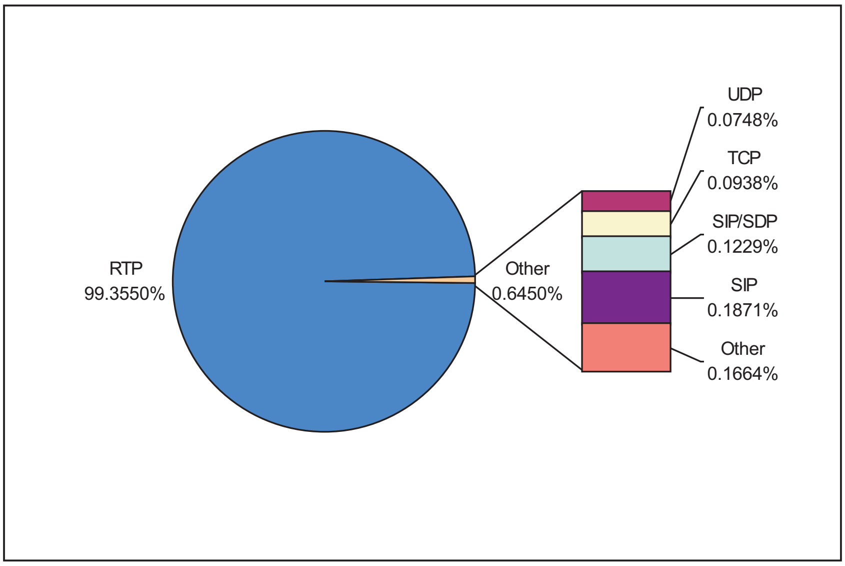


Figure 4.6: Protocol mix of VoIP traffic in Vbuzzer, by bytes

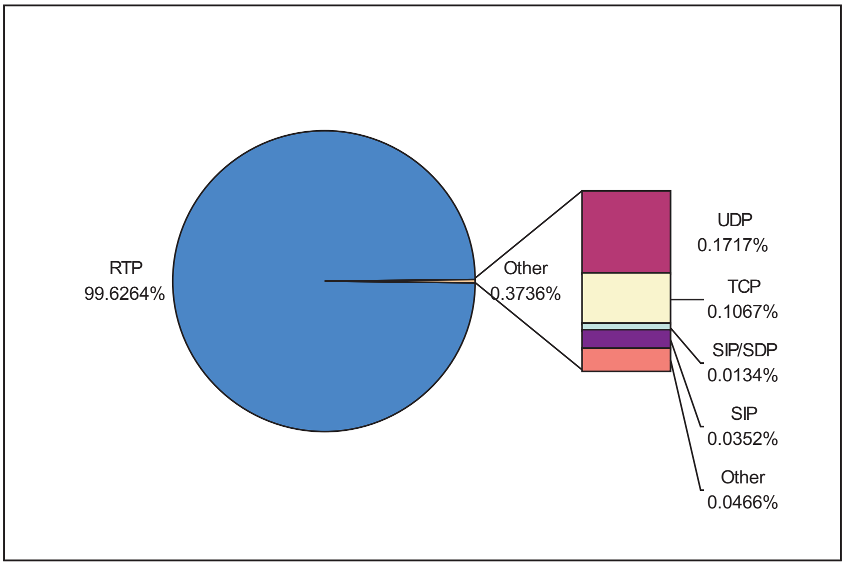


Figure 4.7: Protocol mix of VoIP traffic in Vbuzzer, by packets

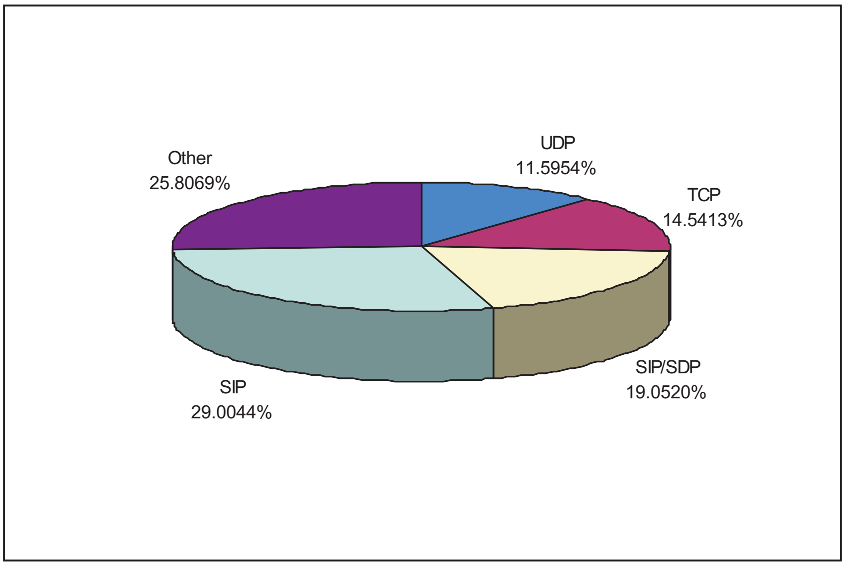


Figure 4.8: Protocol mix of signal control traffic in Vbuzzer, by bytes

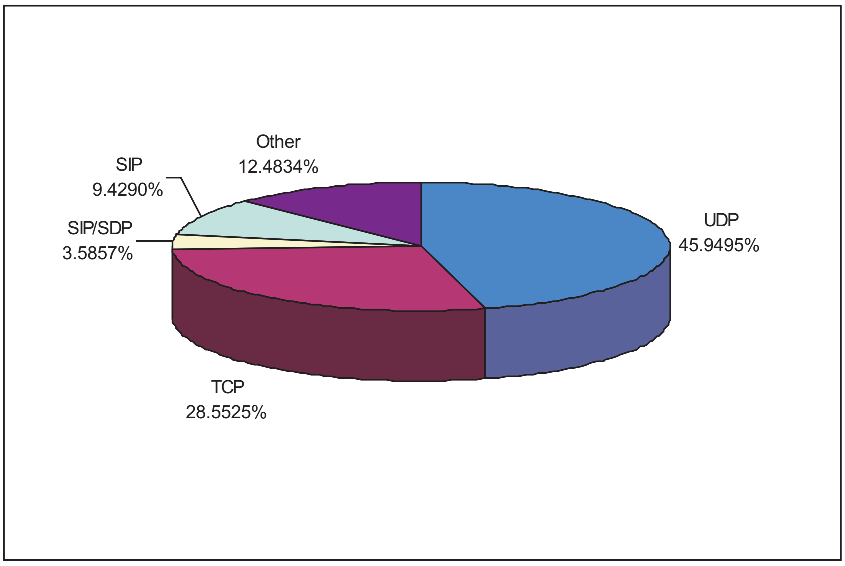


Figure 4.9: Protocol mix of signal control traffic in Vbuzzer, by packets

Figure 4.10. Figure 4.10 shows that the interarrival time of 70% of the packets is less than 20ms.

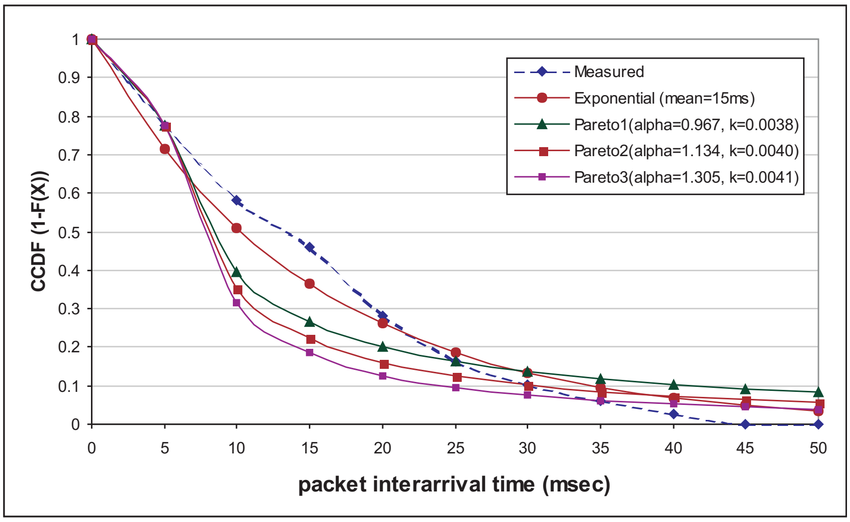


Figure 4.10: Distribution of interarrival time of voice packets in Vbuzzer

In order to determine a statistical distribution for interarrival time, the measured interarrival time samples are compared and fit to a known analytical distribution. The CCDF of the interarrival time closely follows an Exponential Distribution with a mean interarrival time of 15ms. A General Pareto Distribution has been tried to fit the measured data and found a close fit with a shape parameter of 0.967 and a scale parameter of 0.0038. However the Exponential Distribution is a better fit for the measured data.

4.2 Cisco VoIP System

4.2.1 Experimental Environment

For the major data capture activity in this thesis, VoIP traffic in a real network was captured in the Department of Computer Science at the University of Saskatchewan. The Department of Computer Science is located in the Thorvaldson Building. The administrative staff, part of faculty and MADMUC Lab are on the 1st & ground floor. About 10 faculty, 6 technical staff and graduate students of the Software Engineering Lab and HCI Lab are on the old wing of the 2nd and 3rd floor. ARIES and DISCUS Labs are in the new wing of the Thorvaldson Building. There are 57 Cisco 7960 IP phones in the department. 20 phones are on the 1st & ground floor. 30 phones are on the old wing of the 2nd and 3rd floor. 7 phones are in ARIES and DISCUS Labs. These IP phones are connected to three switch closets, each of which is located on a different floor. One IBM Thinkpad laptop was placed in each closet. The laptop was connected to an ethernet port of a switch in the closet, to which all VoIP traffic was mirrored. The data capture software *Ipsumdump*[4] was installed on every laptop. *Ipsumdump* was listening to the ethernet port to which it was connected, and captured the traffic from this port. A cron job ran for three months (October 13, 2006 to January 19, 2007) to collect data and saved the captured data to the hard disk. At the time of the study, the VoIP network in the department is assigned to a separate VLAN from campus data network. Figure 4.11 shows the detailed VoIP network architecture.

In a Cisco VoIP system, the main components are Cisco IP phone, Cisco Call Manager (CCM) and Cisco Unity. The Cisco Call Manager software is the call-processing component of the Cisco Unified Communications system. It is an enterprise IP telephony call-processing solution[5]. Cisco Unity Unified Messaging, an integral component of the Cisco IP Communications system, is a foundational element in bringing unified communication solutions to enterprise-scale organizations. It delivers unified messaging (e-mail, voice, and fax messages sent to one inbox) and intelligent voice messaging (full-featured voice mail providing advanced functions). An uninterruptible power supply (UPS) is a device which maintains a continuous supply of electric power to connected equipment by supplying power from a separate source when utility power is not available. The campus core network is a network to which the campus buildings connect.

4.2.2 Data Capture Tools

Data from voice calls was captured using a combination of tools, each of which is described in this section. The foundation is *Click*, upon which *ipsumdump*was developed. Finally, a locally

[4]Eddie Kohler, ipsumdump, http://www.cs.ucla.edu/kohler/ipsumdump/
[5]Cisco, CCM, http://www.cisco.com/en/US/products/sw/voicesw/ps556/index.html

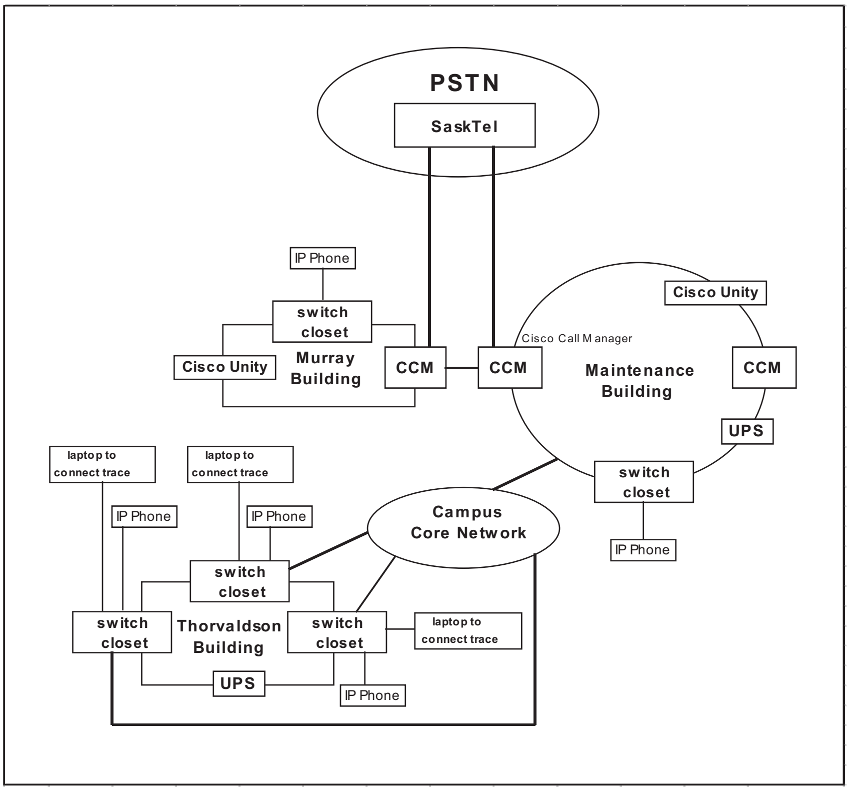


Figure 4.11: UofS VoIP network architecture

developed anonymizer was implemented to remove private information.

Click

Click is a software architecture for constructing flexible and configurable routers. A Click router is assembled from packet processing modules called elements. Individual elements implement simple router functions such as packet classification, queueing, scheduling, and interfacing with network devices [22]. Click elements can also handle all packet-related tasks, such as reading and writing tcpdump and ipsumdump files, sampling, filtering, and anonymization.

Ipsumdump

Ipsumdump is a standalone program built on Click modules. It simply constructs a Click configuration based on options provided by the user, and then runs that configuration program. Ipsumdump can read packets from network interfaces, from tcpdump files, and from existing ipsumdump files. It can sample traffic on a random basis, filter traffic based on its contents, anonymize IP addresses, and sort packets from multiple dumps by timestamp. Also, it can optionally create a tcpdump file containing actual packet data.

Anonymizer

Since the monitored packets were being captured from real calls, no private information should be revealed in the trace. The RTP payload was stripped off from the captured packets through an anonymizer[6] which was built upon Ipsumdump. The trace lasted several days. After private information such as RTP payload, IP addresses and MAC addresses was anonymized, the captured data for each day was stored on the hard disk for each day.

4.2.3 Analysis Software

Wireshark

Wireshark is the new name for Ethereal network protocol analyzer. In this experiment, Wireshark is used to decode the captured traffic data. It has been explained in Section 4.1.3.

Perl

Perl was used in this experiment to analyse the information contained in the packet trace. The algorithm used to analyse the packets is outlined in Figure 4.12.

```
While (Read Next Trace Line)
      Split Trace Line into Packet Fields
      Get Packet Arrival Time Field
      Get Packet Length
      If (this Packet is a Voice Packet)
          Count as a Voice Packet
          Add Packet Length to Voice Traffic Volume
      If (this Packet contains "OpenReceiveChannel")
          Record Packet Arrival Time as Call Start Time
      If (this Packet contains "CloseReceiveChannel")
          Record Packet Arrival Time as Call End Time
Count Daily Call Numbers for each Floor
Count Call Duration for each Call
Count Call Interarrival Time
Count Voice Traffic for each Floor on every day
```

Figure 4.12: Trace analysis algorithm in Cisco VoIP system

[6]Brian Gallaway, http://abu.usask.ca/cgi-bin/cvsweb/ipsumdump/

4.2.4 Call Analysis

Call Setup Process

Table 4.5 displays the call setup process and Table 4.6 depicts the call tear down process in the Cisco VoIP System. In order to illustrate clearly, the packet number of some packets has been renumbered.

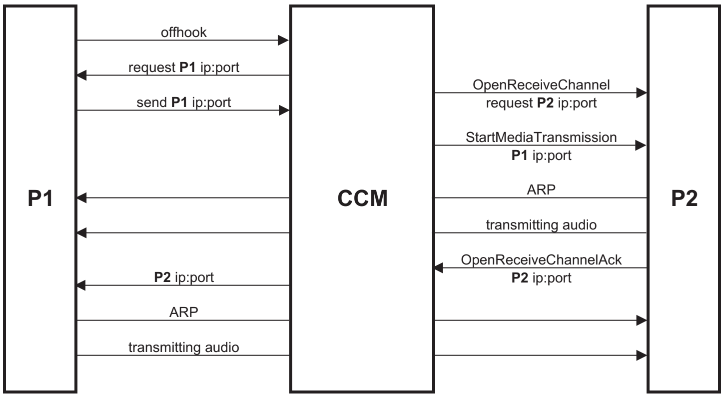


Figure 4.13: Call setup process in Cisco VoIP system

For a call from P1 (phone 1) through CCM (Cisco Call Manager) to P2 (phone 2) (P1—CCM—P2), the call setup process is as following (shown in Figure 4.13):

1. P1 goes offhook
2. CCM receives offhook msg from phone P1
3. CCM sends message to P1 asking for destination ip:port for the RTP stream (OpenLogicalChannel if H323, CRCX if MGCP)
4. CCM sends OpenReceiveChannel down to the phone P2 asking for a destination ip:port for the RTP stream
5. P1 sends port number to CCM (OpenLogicalChannelAck if H323, part of the 200 OK msg if MGCP)
6. CCM sends StartMediaTransmission to the phone P2 indicating the P1 ip:port
7. P2 ARPs for P1 IP or default gateway IP based on subnet mask
8. P2 begins transmitting audio to P1

Table 4.5: Call setup process in Cisco VoIP system (caller)

No.	**Protocol**	**Info**
5020500	SKINNY	OffHookMessage
5020501	SKINNY	KeypadButtonMessage
5020502	SKINNY	DialedNumberMessage
5020503	SKINNY	StopToneMessage
5020504	SKINNY	OpenReceiveChannel
5020505	SKINNY	CallStateMessage
5020506	SKINNY	SelectSoftKeysMessage
5020507	SKINNY	DisplayPromptStatusMessage
5020508	SKINNY	CallInfoMessage
5020509	SKINNY	StopToneMessage
5020510	TCP	51324 > sieve [ACK] Seq=10416 Ack=10412 Win=7656 Len=0
5020511	SKINNY	OpenReceiveChannelAck
5020512	SKINNY	KeepAliveMessage
5020513	SKINNY	KeepAliveAckMessage
5020514	TCP	TCP 52541 > sieve [ACK] Seq=9540 Ack=14512 Win=8192 Len=0
5020515	SKINNY	StartMediaTransmission
5020516	SKINNY	CallInfoMessage
5020517	TCP	TCP 51324 > sieve [ACK] Seq=10448 Ack=10508
5020518	RTP	Source port: 17872 Destination port: 32394
5020519	RTP	Source port: 32394 Destination port: 17872
5020520	SKINNY	KeepAliveMessage
5020521	SKINNY	KeepAliveAckMessage
5020522	TCP	TCP 51318 > sieve [ACK] Seq=9288 Ack=10172 Win=8192 Len=0
5020523	RTP	Source port: 17872 Destination port: 32394
5020524	RTP	Source port: 17872 Destination port: 32394
5020525	RTP	Source port: 17872 Destination port: 32394
5020526	RTP	Source port: 17872 Destination port: 32394

Table 4.6: Call tear down process in Cisco VoIP system

No.	Protocol	Info
5034548	RTP	Source port: 32394 Destination port: 17872
5034549	RTP	Source port: 17872 Destination port: 32394
5034550	SKINNY	KeepAliveMessage
5034551	SKINNY	KeepAliveAckMessage
5034552	TCP	sieve > 50752 [ACK] Seq=0 Ack=4236 Win=16584 Len=0
5034553	RTP	Source port: 32394 Destination port: 17872
5034554	RTP	Source port: 17872 Destination port: 32394
5034555	SKINNY	CloseReceiveChannel
5034556	SKINNY	StopMediaTransmission
5034557	SKINNY	SetLampMessage
5034558	SKINNY	ClearPromptStatusMessage
5034559	SKINNY	CallStateMessage
5034560	SKINNY	SelectSoftKeysMessage
5034561	SKINNY	DefineTimeDate
5034562	SKINNY	ConnectionStatisticsReq
5034563	SKINNY	SetSpeakerModeMessage
5034564	SKINNY	SetRingerMessage
5034565	RTP	Source port: 17872 Destination port: 32394
5034566	RTP	Source port: 32394 Destination port: 17872
5034567	TCP	51324 > sieve [ACK] Seq=10460 Ack=10812 Win=8192 Len=0
5034568	RTP	Source port: 17872 Destination port: 32394
5034569	RTP	Source port: 32394 Destination port: 17872
5034570	SKINNY	ConnectionStatisticsRes
5034571	TCP	TCP sieve > 51324 [ACK] Seq=10812 Ack=10532 Win=16352 Len=0
5034572	SKINNY	OnHookMessage
5034573	TCP	sieve > 51324 [ACK] Seq=10812 Ack=10552 Win=16332 Len=0

9. P2 sends OpenReceiveChannelAck to CCM indicating the ip:port to use for the RTP stream
10. CCM sends msg to P1 with P2 port number for phone OpenLogicalChanelAck if H323, MDCX sendrecv if MGCP
11. P1 ARPs for phone IP or default gateway IP based on subnet mask
12. P1 begins transmitting audio.

The above shows that the call setup procedure in Cisco VoIP system is quite different from the one in SIP systems. In SIP system, a caller and a callee establish a session through sending messages to the proxy server. The process occurs on the application layer. In Cisco VoIP system, a caller and a callee establish a session through sending Skinny messages to the CCM and identifying IP addresses and port numbers for each other. The process occurs at the transport and network layer.

Statistics and Data

In Table 4.7, summary information in the trace files from the Cisco VoIP system is presented. In this experiment, VoIP traffic was captured for three months between October 13, 2006 and January 19, 2007. Due to a broken hard disk of the laptop on the 1st and ground floor, there are gaps in the data for some floors and some weeks. In particular, the data from November 7, 2006 to January 19, 2007 of the 1st & ground floor is missing. A typical week of continuous data was chosen for more detailed analysis (October 23 to 29, 2006). In total, nearly 390 million individual packets were transmitted during the week.

Table 4.7: Overall statistics for the trace

Attribute	**Cisco VoIP system**
Total Packets	195,132,108
Total Voice Packets	12,120,342
Total Seconds	604,766
Average Traffic	322.657 packets per second
Average Voice Traffic	20.041 packets per second
Total Time	6 days, 23 hours, 59 minutes, 27 seconds

Distribution by Packet Type

Figure 4.14 and Figure 4.15 show the protocol mix for VoIP traffic in the Cisco VoIP system. By packet count, most of the traffic is generated by RTP (70.6%). The remaining 30% of the traffic is mainly accounted for by SKINNY (18.42%) and TCP (10.70%). In terms of byte count, the traffic is still dominated by RTP traffic (88.60%) followed by SKINNY (7.31%) and TCP(3.93%). Here only VoIP traffic is considered and other network traffic such as HSRP (Hot Standby Routing Protocol) is neglected. SKINNY is the call signaling protocol in Cisco VoIP systems and it is responsible for call setup, tear down and call status monitoring, but the overhead seems high. The voice data is only carried by RTP protocol. The longer the call lasts, the more RTP packets will be generated per call. Also with higher call activity, more RTP packets will be generated in total.

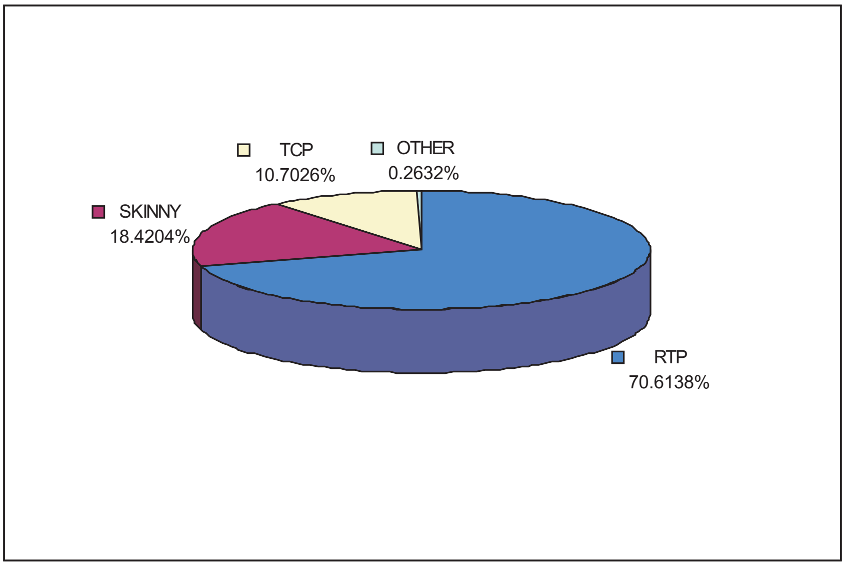


Figure 4.14: Protocol mix of VoIP traffic in Cisco VoIP system, by packets

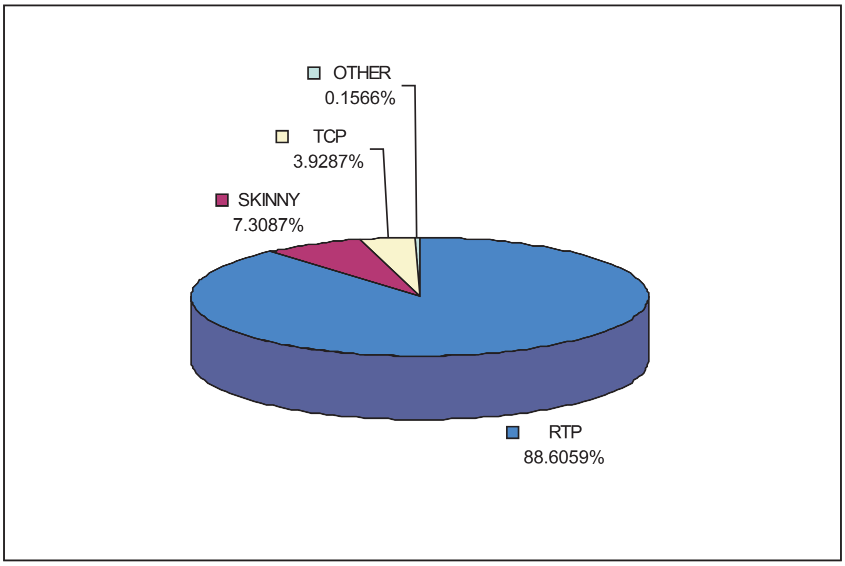


Figure 4.15: Protocol mix of VoIP traffic in Cisco VoIP system, by bytes

Distribution by Packet Size

Figure 4.16 displays the distribution of packet size in the Cisco VoIP system. It shows that 74.37% packets are between 129 bytes and 256 bytes in length because RTP packets occupy 70.8% of VoIP traffic and RTP packets have fixed size of 214 bytes per packet (see Table 4.9). The size of SKINNY packets varies from 66 bytes to 1514 bytes. Table 4.8 and 4.9 show that the majority of SKINNY packets have packet size between 65 bytes and 128 bytes. SKINNY packets account for 18% of total traffic. So there are about 16.23% of total packets with size between 65 bytes and 128 bytes. The size of TCP packets varies from 60 bytes to 1514 bytes, but very few SKINNY and TCP packets have size greater than 257 bytes. So the total number of packets with size bigger than 257 bytes is very small. The remaining 9.36% of packets, which are almost all TCP packets, have sizes smaller than 64 bytes. A better understanding of how the network handles various packet size distributions could help in determining important trade-offs at the application level.

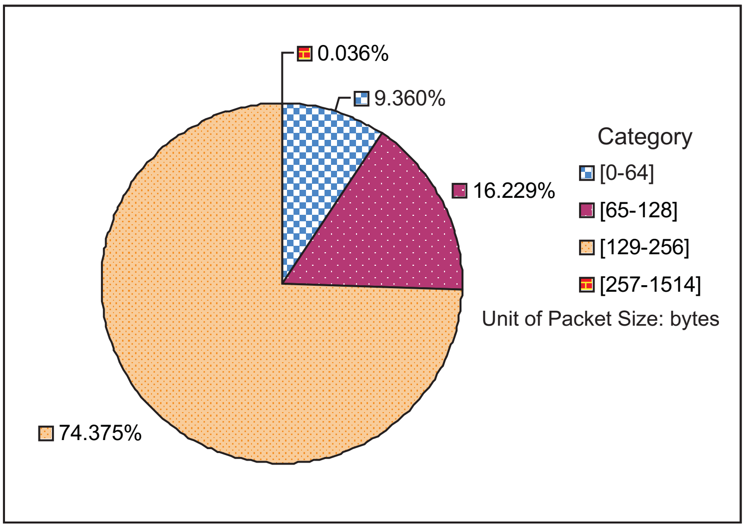


Figure 4.16: Distribution of packet size in Cisco VoIP system

Table 4.8: Number of packets for different protocols

Packet Size(bytes)	RTP	TCP	SKINNY
0-64	0	418,426	0
65-128	0	804	724,707
129-256	3,324,137	0	718
257-1514	0	38	1,576

Table 4.9: Distribution of packet size for different protocols

Packet Size(bytes)	RTP	TCP	SKINNY
0-64	0%	99.7991%	0%
65-128	0%	0.1918%	99.6844%
129-256	100%	0%	0.0988%
257-1514	0%	0.0091%	0.2168%

Distribution of Packet Interarrival Time

The Complementary Cumulative Distribution (CCDF) of interarrival time of RTP packets in Cisco VoIP system is displayed in Figure 4.17. Figure 4.17 shows that the interarrival time of most RTP packets (95.4%) in the same call is less than 20ms. The interarrival time of the remaining 4.6% packets is bigger than 20ms but still less than 25ms.

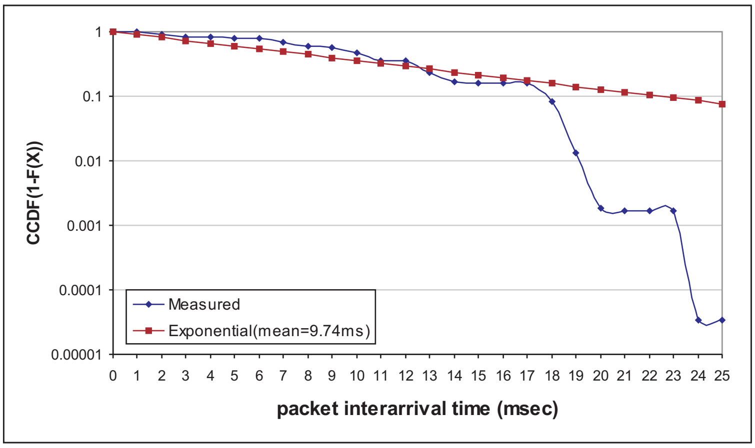


Figure 4.17: Distribution of interarrival time of RTP packets in Cisco VoIP system

Distribution of Call Interarrival Time

The Complementary Cumulative Distribution(CCDF) of call interarrival times between three different time periods are displayed in Figure 4.18, Figure 4.19 and Figure 4.20 respectively. The call interarrival time distribution can be approximated by an Exponential distribution with mean interarrival time of 6 minutes in the morning peak time and 5 minutes in the afternoon peak time. For the non-peak time such as the time period between 17:00 and 18:00, the call interarrival time distribution can still be approximated by an Exponential distribution with mean interarrival time of 10 minutes.

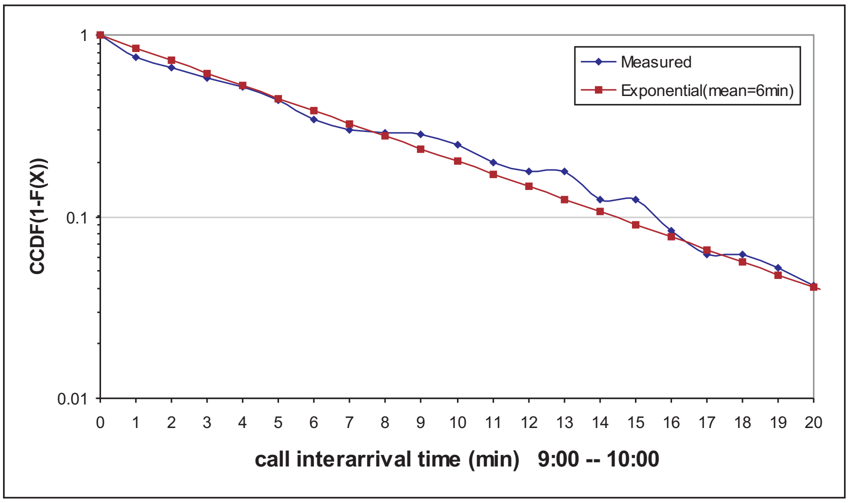


Figure 4.18: Distribution of call interarrival time between 9:00 and 10:00 (Oct, 2006 – Jan, 2007)

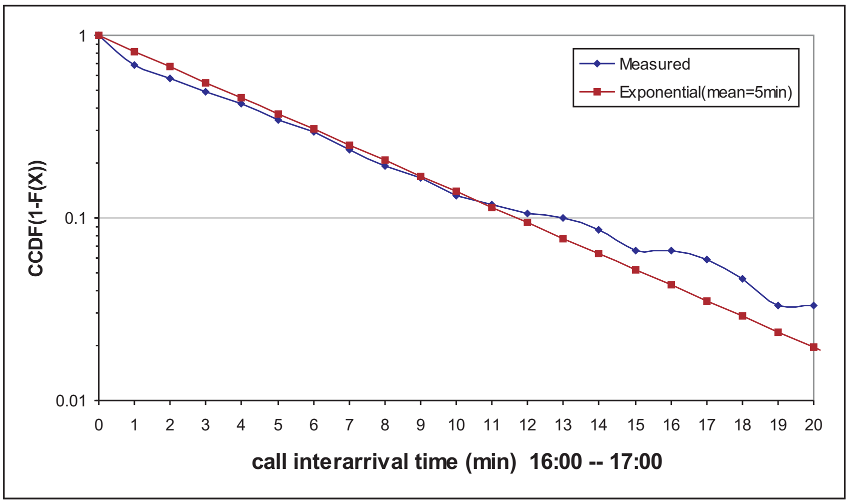


Figure 4.19: Distribution of call interarrival time between 16:00 and 17:00 (Oct, 2006 – Jan, 2007)

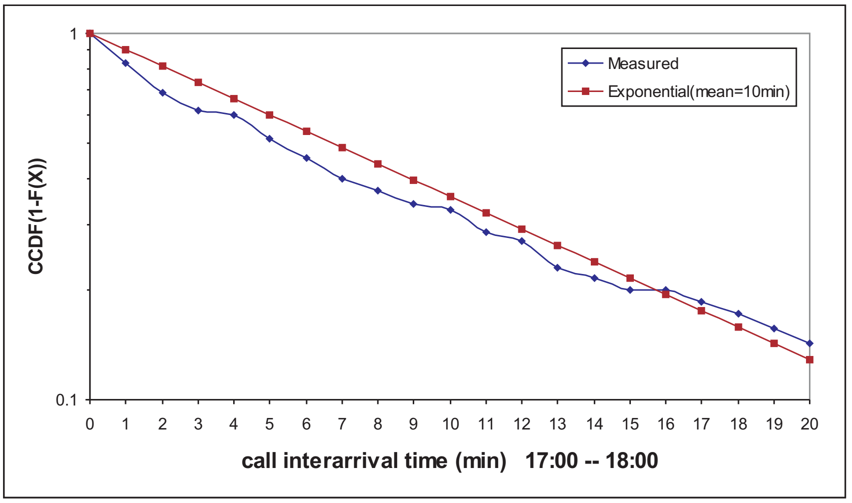


Figure 4.20: Distribution of call interarrival time between 17:00 and 18:00 (Oct, 2006 – Jan, 2007)

Figure 4.21 displays the overall call interarrival time distribution in the 1st & ground floor. It conforms to the Exponential distribution as well.

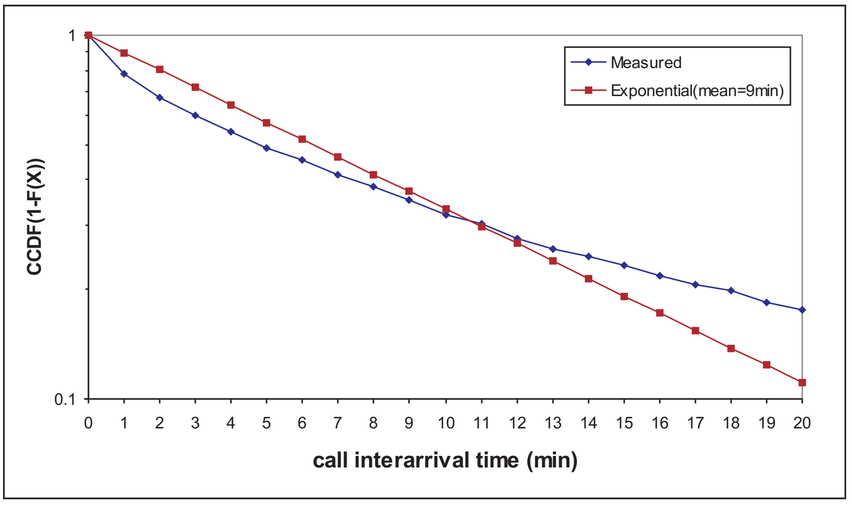
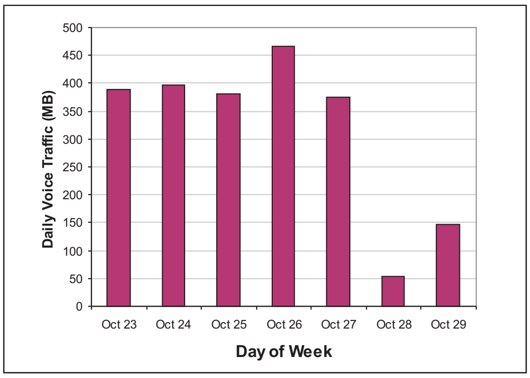


Figure 4.21: Distribution of call interarrival time in 1st & ground floor (Oct, 2006 – Jan, 2007)

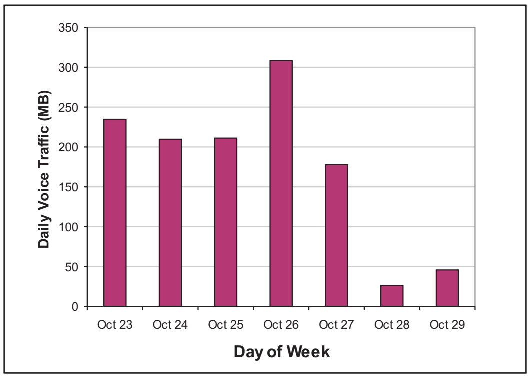
4.2.5 Workload Measurements

Traffic

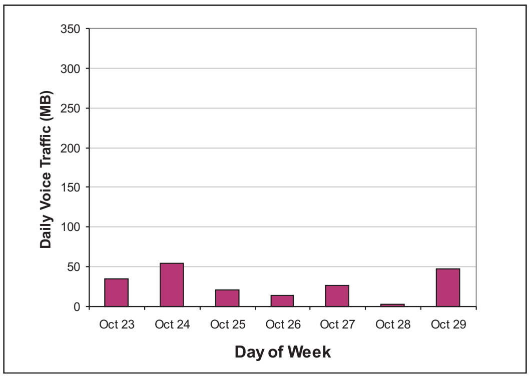
Perhaps the most fundamental questions about a new network involve when it is used and how much. Figure 4.22 shows an approximate daily pattern for voice traffic in a week. Voice traffic does not change dramatically during weekdays. There is lower traffic during the weekend because there are fewer calls during those times.



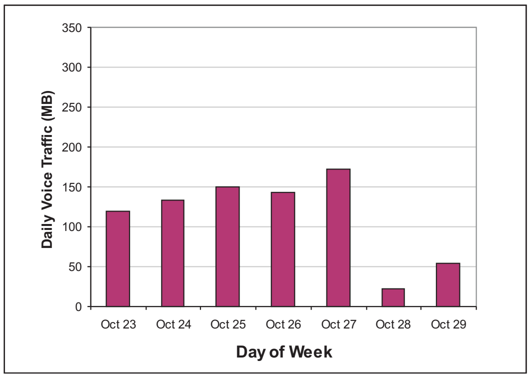
(a) Daily total voice traffic



(b) Daily voice traffic in 2nd & 3rd floor (old wing)



(c) Daily voice traffic in 2nd floor (new wing)



(d) Daily voice traffic in 1st & ground floor

Figure 4.22: Daily voice traffic

Figure 4.23 shows a weekly pattern for voice traffic in four consecutive months. One week with integrated data was chosen in every month except January. It could result from a power outage which caused that only a small part of the data on January 5, 2007 was captured. From this figure, it can be seen that from October to December, Monday, Tuesday and Thursday seem to be peak days for voice traffic though overall traffic does not change dramatically during weekdays. The data from January shows a small difference because the week is very close to holidays.

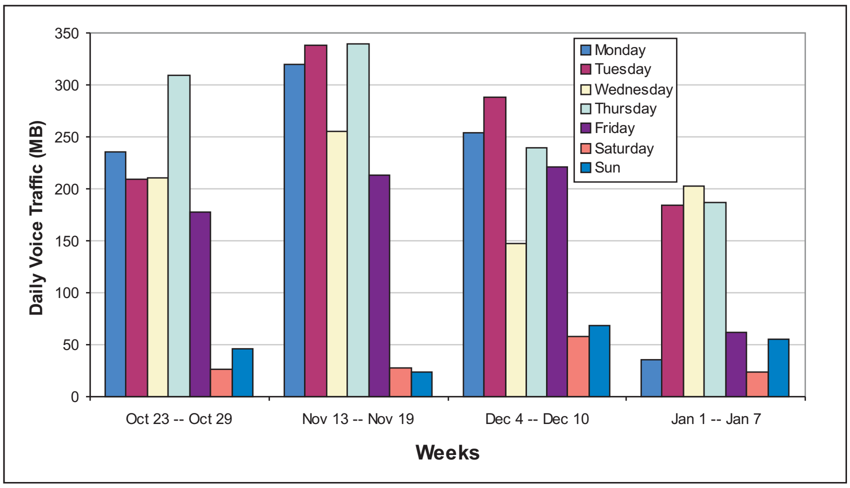


Figure 4.23: Weekly voice traffic

Figure 4.24 displays traffic transferred in terms of call duration. It is obvious that the longer the call lasts, the more voice packets (mainly RTP packets) will be generated.

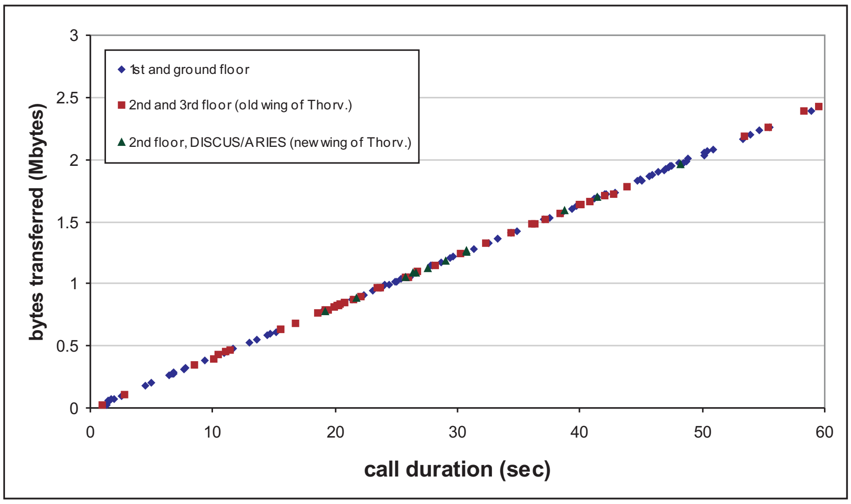


Figure 4.24: Bytes transferred in terms of call duration

Figure 4.25 shows the variation of voice traffic over the hours of the day. The peak traffic happens between 12:00 and 13:00 and between 16:00 and 17:00. The first time period between 12:00 and 13:00 is lunch time. The second time period between 16:00 and 17:00 is the time at the end of a work day. This is expected because these two time periods are the best time for students, faculty and staff to make phone calls.

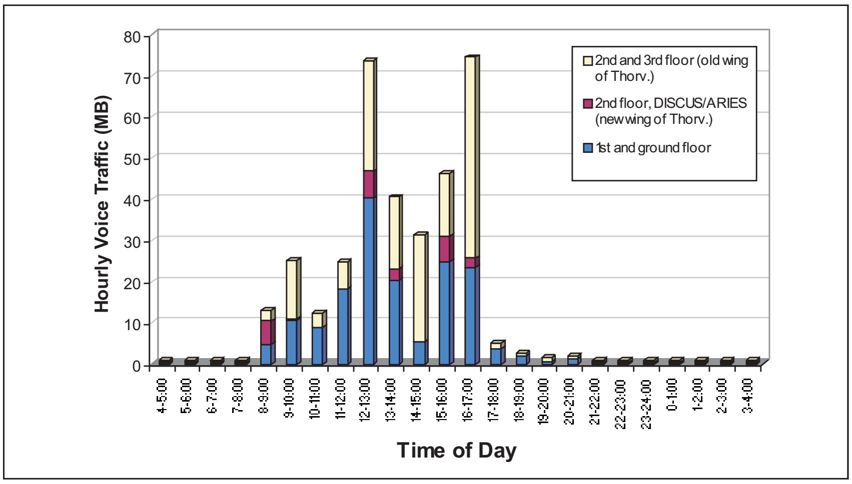


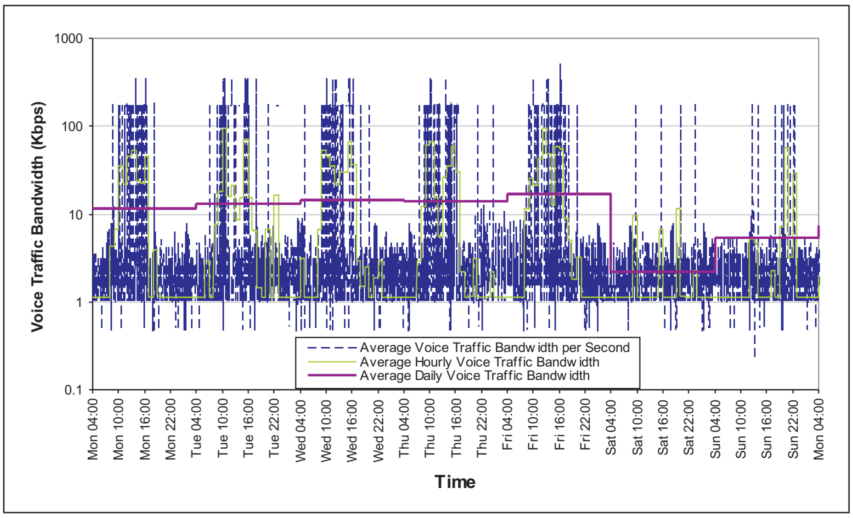
Figure 4.25: Hourly voice traffic

Figure 4.26, 4.27 and 4.28 are time-series plots of the voice traffic bandwidth usage in three different floors in the department during the trace period. The average consumed bandwidth in the 1st and ground floor is calculated using different time scales. They are displayed separately in Figure 4.26. It can be found that the voice traffic on the VoIP network remains high between 10:00 and 17:00 on each weekday. On Saturday and Sunday, the voice traffic did not increase as much as weekdays. This corresponds to the above voice traffic volume figures. The overall bursty behavior and peaks are similar at all three floors, though the absolute peak bandwidth consumption varies at different floors.

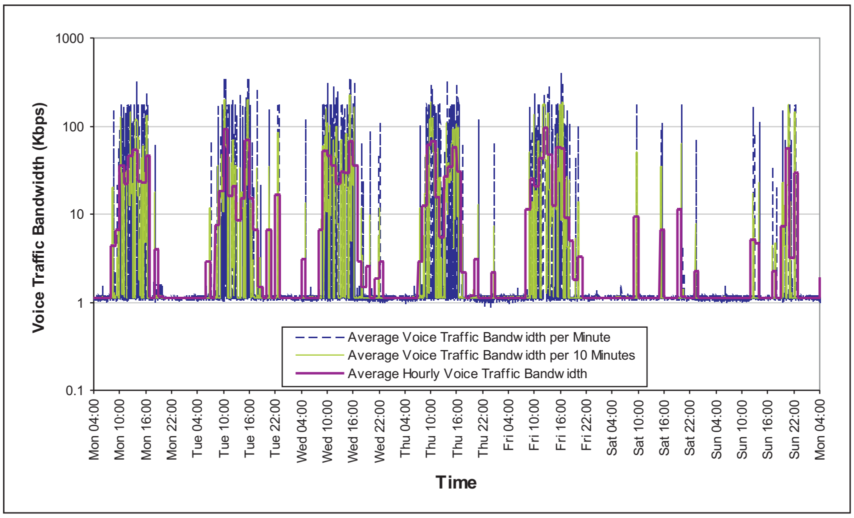
Call Statistics

The basic characteristics of user behavior in cold season in the VoIP system are analysed in this section because large part of the study was carried out during cold months. Figure 4.29 displays a daily pattern for phone calls. Obviously Friday is a busy day. There are few calls at the weekend because most people are not in the university at the weekend. Figure 4.30 shows a hourly pattern for phone calls. In order to see clearly, the hourly periods for which there are 0 calls have been removed. The busiest time is between 16:00 and 17:00, indicating that most phone calls happen in the afternoon. Figure 4.31 displays a weekly pattern for phone calls.

The next set of results concern call duration. A call session is a contiguous sequence of time points from the call setup to the call end. Figure 4.32, Figure 4.33 and Figure 4.34 show CCDF for call session times in different time periods. Most call durations are less than 10 minutes. Three different time periods (two peak time periods and one non-peak hour period) were chosen in order to find out whether the distribution of call duration time varies in different time periods within a day. From the three figures, it can be discovered that the distribution of call duration follows a



(a) Average voice traffic bandwidth usage per second



(b) Average voice traffic bandwidth usage per minute

Figure 4.26: Voice traffic bandwidth usage in 1st & ground floor

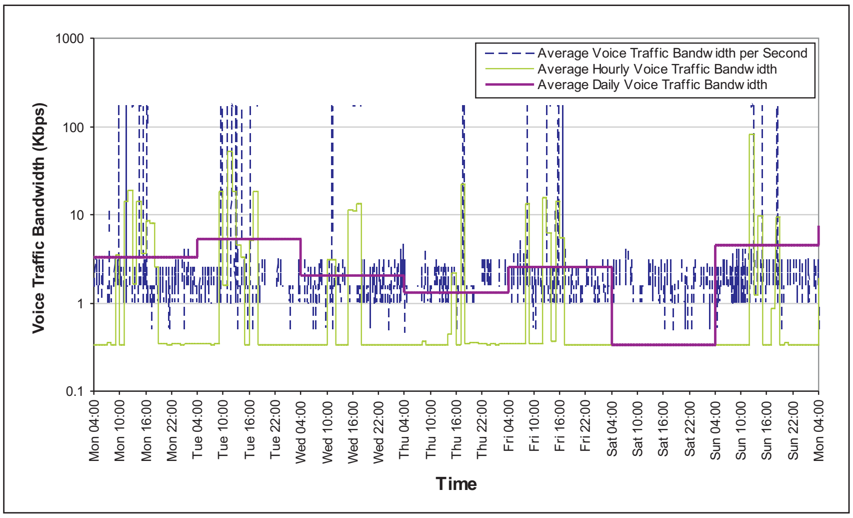


Figure 4.27: Voice traffic bandwidth usage in 2nd floor(new wing)

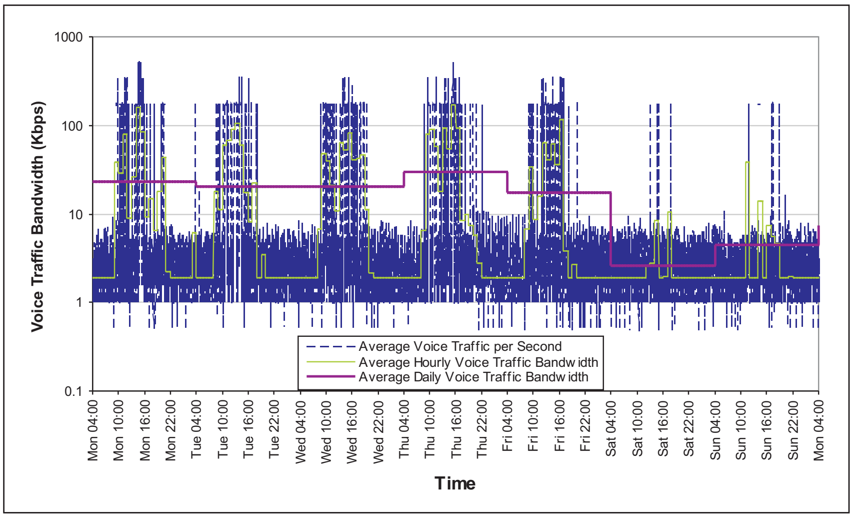


Figure 4.28: Voice traffic bandwidth usage in 2nd & 3rd floor(old wing)

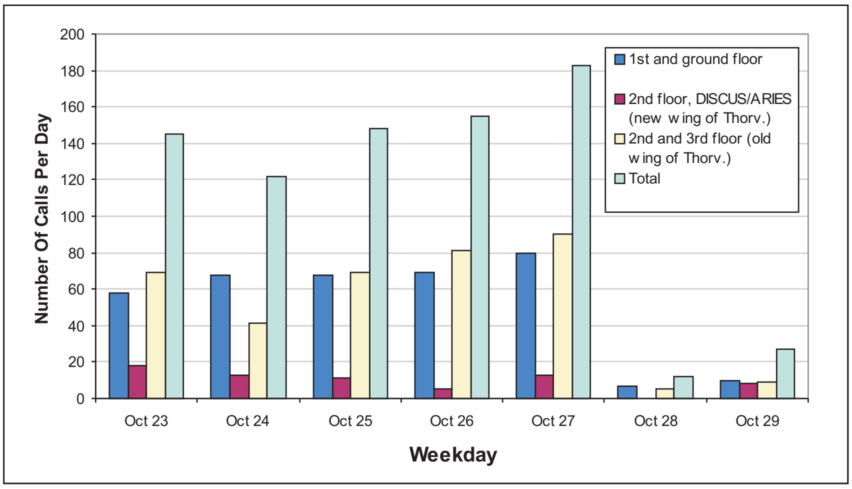


Figure 4.29: Daily calls

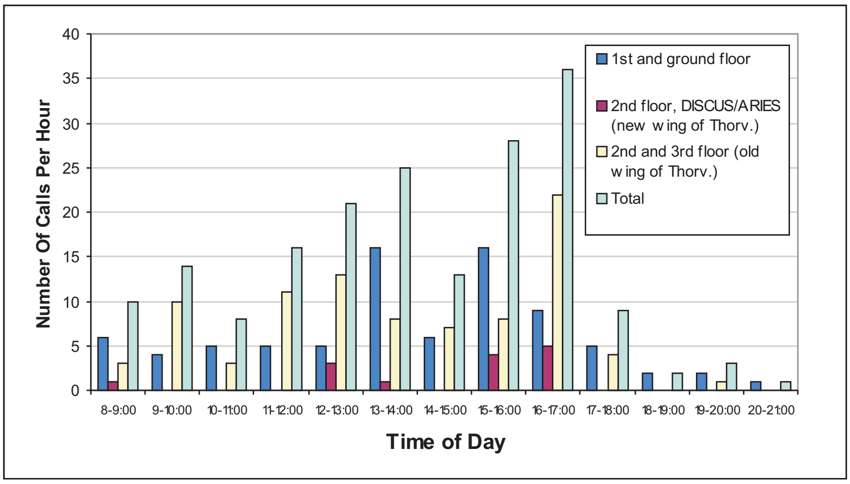


Figure 4.30: Hourly calls on peak day (Thursday, Oct 27, 2006)

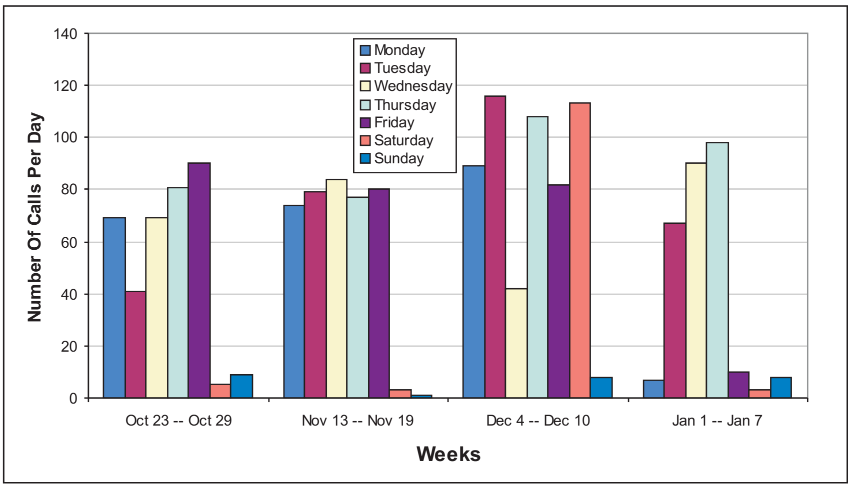


Figure 4.31: Weekly calls (Oct 2006 – Jan 2007)

General Pareto Distribution with different shape and scale parameters in different time periods.

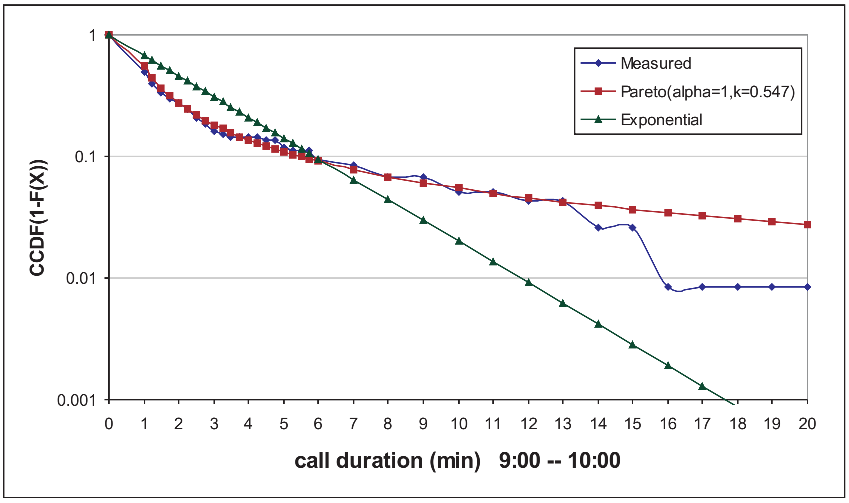


Figure 4.32: distribution of call duration between 9:00 and 10:00 (Oct,2006 – Jan,2007)

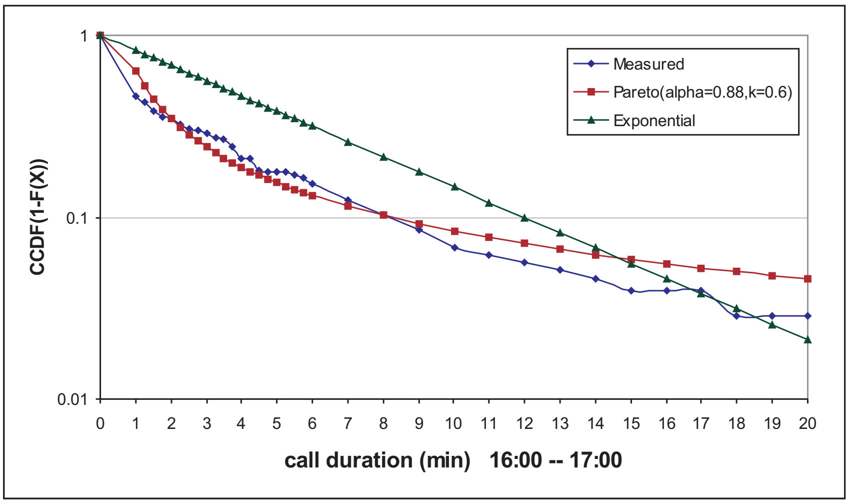


Figure 4.33: distribution of call duration between 16:00 and 17:00 (Oct,2006 – Jan,2007)

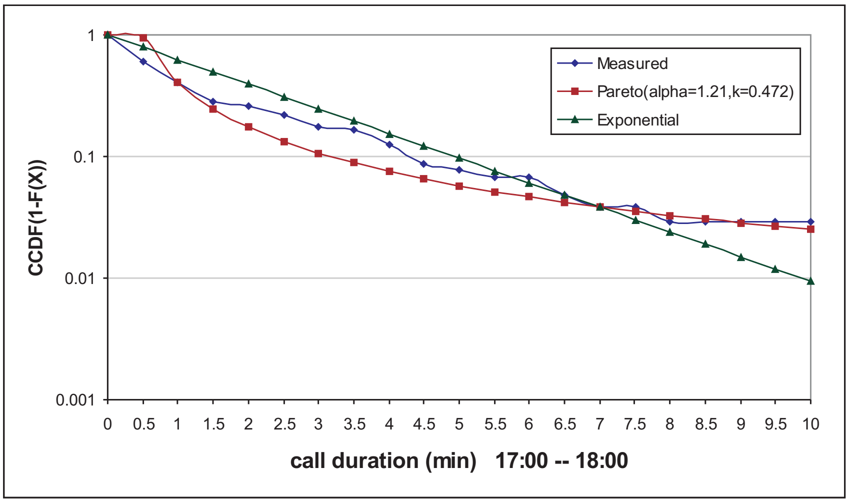


Figure 4.34: distribution of call duration between 17:00 and 18:00 (Oct,2006 – Jan,2007)

Figure 4.35 displays the overall call duration distribution in the 1st & ground floor. It conforms to the General Pareto distribution too.

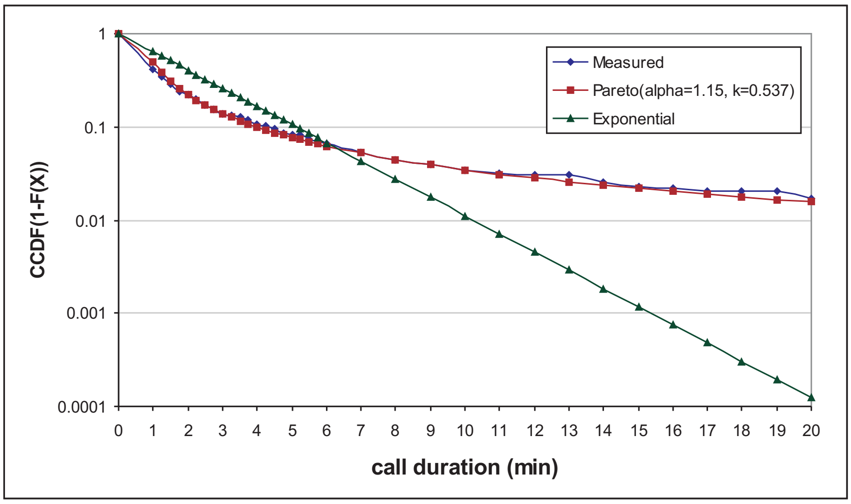


Figure 4.35: distribution of call duration in 1st & ground floor (Oct,2006 – Jan,2007)

Figure 4.36 displays average call duration within a day. The average call session time between 12:00 and 13:00 is a little bit longer than in other time periods. This is reasonable because it is break time.

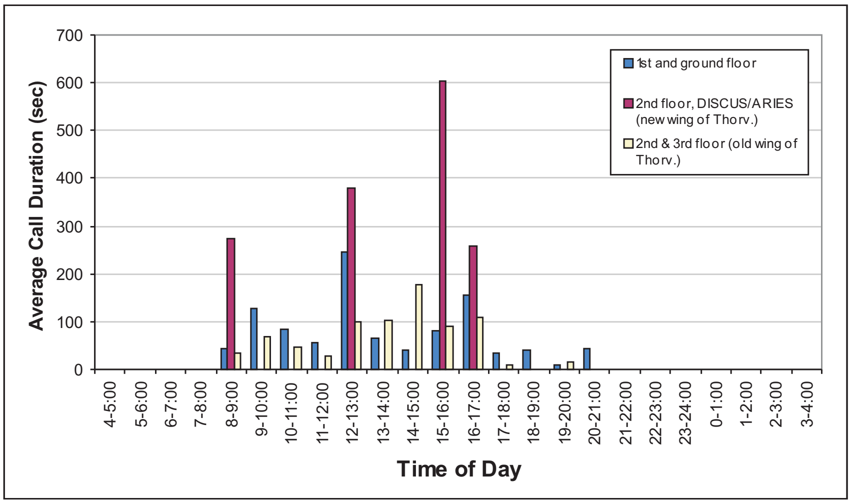


Figure 4.36: Average hourly call session time in Cisco VoIP system

4.2.6 VoIP Quality Study

In the Cisco VoIP system, jitter is an average jitter value for a call, measured in milliseconds. It is an estimate of mean deviation of the RTP data packet inter-arrival time. Latency is an estimate of the network latency, expressed in miliseconds. It represents a running average of the round-trip delay. Packet loss is the fraction of RTP data packets that have been lost since starting receiving data on this connection. It is defined as the difference between the number of expected packets and the number of packets actually received, where the number of received packets includes any that are late or duplicate.

Figure 4.37 and 4.38 shows the delay and jitter values obtained from trace data in Cisco VoIP system on a peak day (Oct 27, 2006). MOS value displayed in Figure 4.39 is calculated using an extended E-Model. The Cisco 7960 VoIP phone can support both G.711 and G.729 codecs. In this system, the Cisco VoIP phone is configured to use the G.711 codec. In these three figures, one marker represents the corresponding delay, jitter and MOS value for a received call. How these values are calculated and reported will be described in the following sections.

Jitter

Generally, the receiving phone that is being used to monitor the connection calculates and reports the value. Each RTP packet contains a header. In that header, the sender places a time stamp. Based on this time stamp and the receiver's own processor clock, the jitter variation from expected arrival time can be computed. The jitter value is computed from these jitter variations. The receiver of a call will normally report a summary jitter value in a Skinny packet at the end of the conversation.

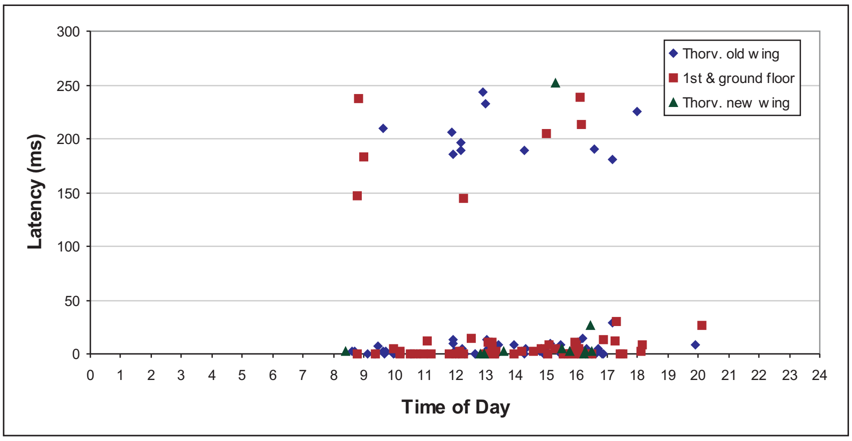


Figure 4.37: Delay variation on peak day

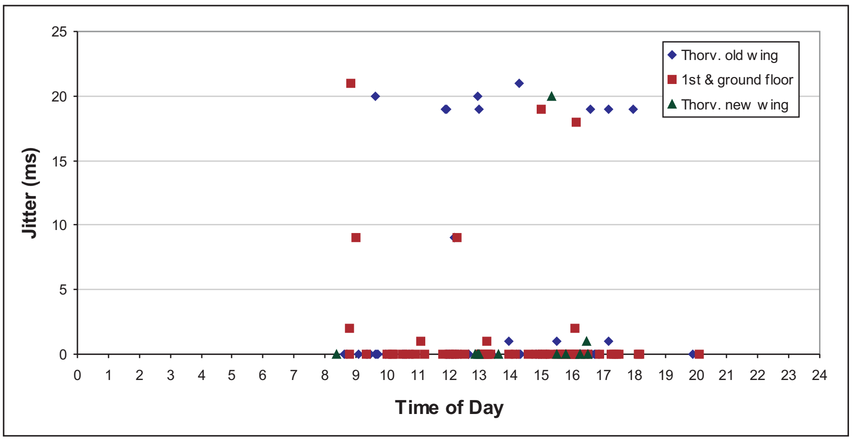


Figure 4.38: Jitter variation on peak day

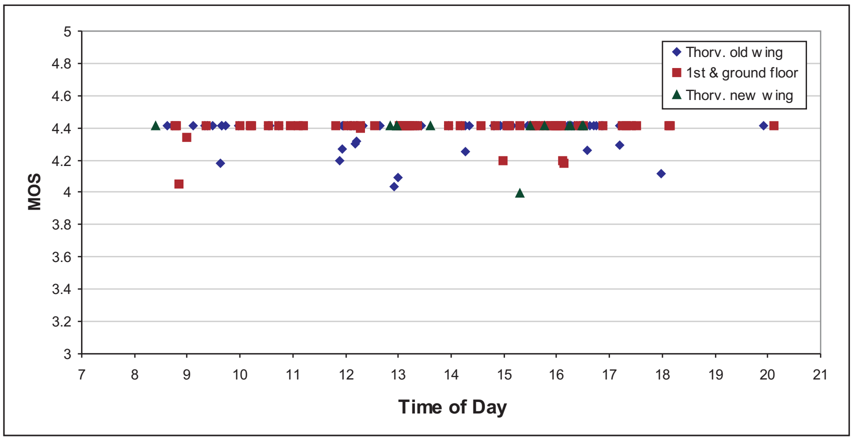


Figure 4.39: MOS variation on peak day

Delay

Delay is another metric that has a substantial impact on VoIP QoS. There are several kinds of delay to be considered such as Jitter Buffer Delay, Codec Delay, Transmission Delay and so on. Measuring transmission delay in a network can be difficult. This is because the computation of the difference between the time at which the packet is sent and when it is received must be based on a common clock and measurements taken at both ends of a conversation. If the sender and receiver are far apart, implementing measurement devices and connecting to such a clock may be difficult. Since transmission delay, in most typical situations, does not significantly impact quality scores, it is not included. However, jitter buffer delay, codec delay, and packetization delay are calculated and used in quality scores. Integrating these measurements is useful as they provide meaningful measures of call quality.

Packet Loss

Packets are marked by the sender with a sequence number. These sequence numbers are consecutive integers sometimes beginning with a random number and sometimes beginning with the number one. This sequence number is inserted in the RTP header of the packet. When the receiver receives a packet with an integer that is not consecutive with the previous packet, it knows immediately that a frame is missing.

In the deployed Cisco VoIP system in this study, there is no packet loss because all phones are connected to a 100M Intranet and this Ethernet LAN is separated from the campus data network. There is enough bandwidth available for call traffic. From above figures, it can be found that the delay and jitter for most phone calls are small because these are local calls. Therefore, it can be concluded that the quality of local calls is excellent. The calls with higher delay and jitter value might be international calls. Even then, the quality is not bad.

In this chapter, the results of a measurement study on a deployed Cisco VoIP phone system was presented. Through the experimental data, some features of a Cisco VoIP system such as call setup and tear down process, distribution of the traffic data and network performance are revealed. This is very useful for the simulation study which will be presented in the next chapter.

Chapter 5

Simulation Experiments on VoIP Quality Metrics

In this chapter, some preliminary simulation experiments on VoIP quality metrics such as delay, jitter and packet loss were performed in a simulated 10Mbps network in ns2. A traffic generator was developed to generate various simulated VoIP traffic. The data used to provide the traffic model parameters was chosen from peak traffic periods in the captured data from the deployment at the Department of Computer Science of UofS. By utilizing the Traffic Trace function in ns2, the simulated VoIP traffic was fed into *ns2*and delay, jitter and packet loss were calculated for different scenarios.

The remainder of this chapter is organized as follows. Section 5.1 presents the experiment environment and traffic generation method. Section 5.2 describes the method to analyse the experiment results. The simulation experiment results are discussed in Section 5.3.

5.1 Workload Generation

5.1.1 Experiment Environment

The experiments were performed in *ns2*[1], which stands for Network Simulator. This simulator is a discrete event simulator aimed at networking research. It provides substantial support for simulation of TCP, UDP, and all the typical IP protocols over wired networks. The simulator suite also includes a graphical visualiser called network animator (nam) to assist in getting more insights into the simulation by visualising packet trace data. The simulator together with its companion, *nam*, form a set of tools for networking research.

The simulator consists of C++ core methods and uses Tcl and Object Tcl shell as interface allowing the input file (simulation script) to describe the model to simulate. Arbitrary network topologies composed of nodes, routers, links and shared media can be defined in ns2. A rich set of protocol objects can then be attached to nodes, usually as agents.

[1] http://www.isi.edu/nsnam/ns

To use ns2, a program is created in the Tcl script language. The Tcl script will do the following:

- Initiate an event scheduler,
- Use the network objects to set up the network topology,
- Tell traffic sources when to start and stop transmitting packets through the event scheduler.

The simulation results are stored as trace files, which can be loaded for analysis by an external application.

Two simulation experiments were performed in this study. Figure 5.1 depicts the simulation topology for the experiment one. In Figure 5.1, the bandwidth between phone group 1 and switch 1, switch 1 and switch 2, switch 2 and phone group 2 is 10Mbps. The propagation delay between each component is set to be 5ms. The network connection between switch 1 and switch 2 is using DropTail queue, which implements FIFO (First In First Out) scheduling and drop-on-overflow buffer management. The maximum queue length is 100 packets. There are UDP connections between phone group 1 and phone group 2. This is a two way traffic transmission.

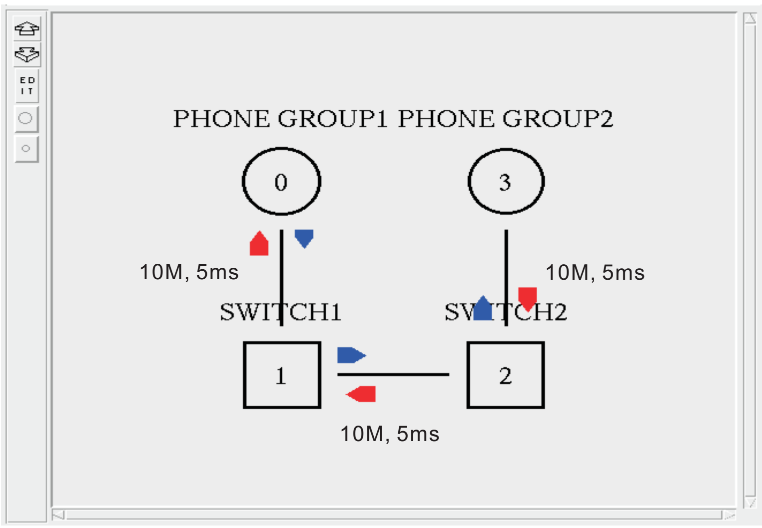


Figure 5.1: Simulation topology 1

Figure 5.2 depicts the simulation topology for the experiment two. In Figure 5.2, the bandwidth between phone group 1 and switch 1, switch 2 and phone group 2 is 100Mbps. The bandwidth between switch 1 and switch 2 is 10Mbps. The propagation delay between each component is set to be 2ms or 5ms. The network connection between switch 1 and switch 2 is using DropTail queue and the maximum queue length is 100 packets.

Besides VoIP traffic, TCP traffic is also running on the network. The TCP traffic is generated from web requests. The experiment is configured to have 100 web clients connecting to a single web server. Each web client has its own 100Mbps link to switch 1. The web server has a 100Mbps link to switch 2. Every client sends an HTTP request to the web server. The web server acknowledges and replies with the simulated content of web page. The values of the parameters are shown in the next section.

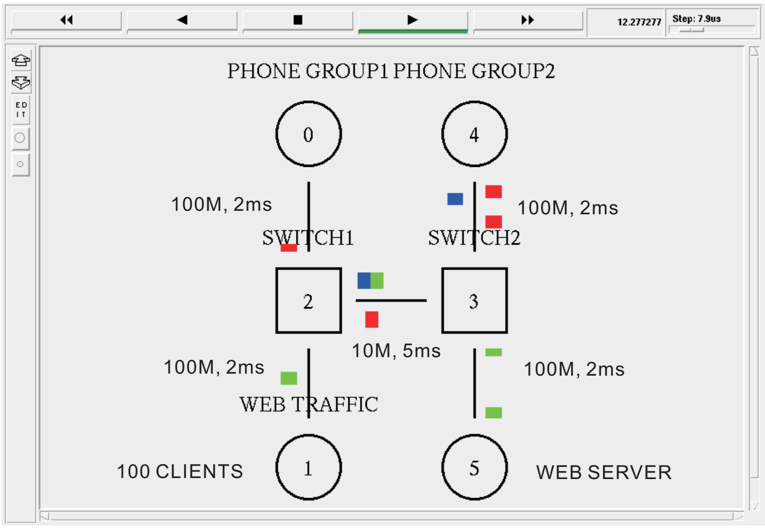


Figure 5.2: Simulation topology 2

5.1.2 Traffic Generation

A function for Traffic Trace is provided in ns2. When using Traffic Trace, it is necessary to provide a binary file, which includes two fields. The first field is inter-packet time specified in microsecond(μs). The second field is packet length. The Traffic Trace function will read information from the binary file, then generates corresponding packets.

The binary file can be transformed from an input ASCII file by a Tcl script. The ASCII file contains two fields. The first field is the packet arrival time from both directions. The second field is packet length. In order to obtain this input file, a traffic generator program was built. The program uses the packets from the real traffic and the parameters from the distributions. Two queues to hold call packets were defined in the program. One queue contains packets for a 10 minutes call while the other queue contains packets for a 40 minutes call. Two fields are defined for each packet – packet arrival time and packet length. The call interarrival time can be determined from the mean call interarrival time (λ) of an exponential distribution. The call duration can be determined from the shape (α) and the scale (k) parameters of a Pareto distribution. The parameters for Pareto distribution are taken from the previous experiments in chapter 4 of this thesis. If the duration of the coming call is less than 10 minutes, the packets from the short queue will be fetched. Otherwise, the packets from the long queue will be fetched. If the call is longer than 40 minutes, the remaining packets are generated with fixed interarrival time of 0.02 second and fixed length of 214 bytes. In this way, the call packets pattern can be changed. The fetched call packets are put in a temporary queue. These packets will be inserted into an aggregated call packets queue. The two fields for each packet in the aggregated call packets queue will be output to an ASCII file. The length of the aggregated call packets queue is determined by the simulation time. Thus the input ASCII file for the Tcl program is created.

5.1.3 Experimental Factors

Three simulation experiments were performed in a simulated 10Mbps (all links are 10Mbps) network environment in experiment one only. The mean of call interarrival time is fixed in each experiment. The mean value is 5 seconds, 20 seconds and 375 seconds respectively. 375 seconds is the value observed from the peak hour 9:00 - 10:00 in the UofS VoIP deployment. In each experiment, the parameters for call duration distribution are varying and three call duration distributions (obtained from 9:00 - 10:00, 16:00 - 17:00, 17:00 - 18:00 of real Cisco VoIP network) were tested (see Figure 5.3).

	Experiment 1	**Experiment 2**	**Experiment 3**
call interarrival time (Exponential)	mean=5sec	mean=20sec	mean=375sec
09:00 -- 10:00 call duration 1 (Pareto)	α=1, k=0.547		
16:00 -- 17:00 call duration 2 (Pareto)	α=0.88, k=0.6		
17:00 -- 18:00 call duration 3 (Pareto)	α=1.21, k=0.472		

Figure 5.3: Experiment parameters

5.2 Analysis Methodology

The result of running ns2 is an output trace file. The trace file can be filtered to only show the relevant data being sampled in each experiment. As described in the nam documentation[2], the typical format of a trace line in the output file is the following:
type -t time -e extent -s source id -d destination id -c conv -i packet id -a <attr> -x <src-na.pa> <dst-sa.na> <seq> <flags> <sname>

[2]http://www.isi.edu/nsnam/ns/doc/index.html

Perl was used to analyse the information in the output file of ns2. It reads every line of the output file. It records the start time that a packet leaves the caller and enters queue and the end time that the same packet arrives at the callee. Delay and jitter can be calculated for each packet.

- Delay. The time difference between the start time and the end time is the propagation delay (latency) for the packet.

- Jitter. Jitter is the time difference between the packet interarrival time and the packet interdeparture time (delay variance).
  $jitter = ((recvtime(j) - sendtime(j)) - (recvtime(i) - sendtime(i)))/(j - i)$,
  $(j > i)$
  "j" is the sequence number of the next new packet received after the packet with sequence number "i", which in the absence of packet loss or network reordering, would equal to i+1.

5.3 Results

5.3.1 Experiment One

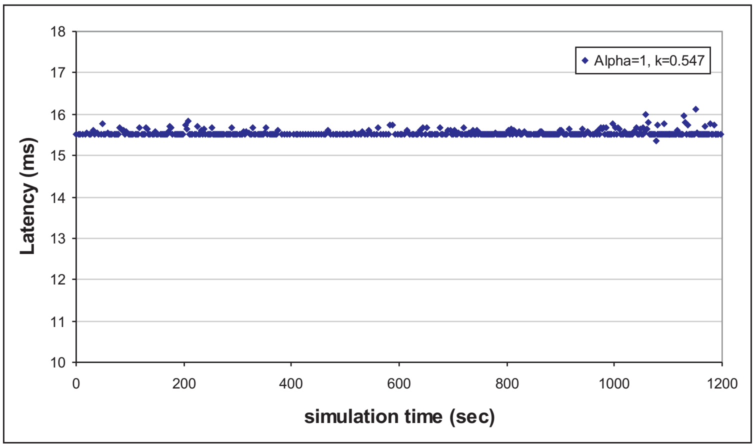
This experiment simulates the traffic of multiple calls running on a backbone link. The calculated results of delay and jitter for the simulation experiments are displayed in the following figures. By measuring the delay and jitter values in different scenarios, it can be found out how VoIP quality metrics are affected in different conditions when the phone system is busy, moderately busy or not busy at all, given the size of a normal department in a university campus. As the mean value of call interarrival time decreases, more calls happen within the simulation time (see Figure 5.6, 5.9 and 5.12). The traffic volume (load) increases accordingly. As the traffic load increases, more packets will experience relatively higher latency and jitter.

When the mean call interarrival time is 5 seconds, the phone system is busy (see Figure 5.4, 5.5 and 5.6). Though the latency for majority packets is 15.5ms, there are about 25% packets with latency between 15.5ms and 16ms (see Figure 5.4). Though the jitter for all packets are less than 1ms, about 14% of the packets have jitter bigger than 0.1ms (see Figure 5.5). The longer the conversations last (see Figure 5.4(b), 5.5(b)), the more packets have relatively higher latency and jitter. When the simulation time is close to 1000 seconds, there are about 40 concurrent calls in Figure 5.6(a), 50 concurrent calls in Figure 5.6(b) and 30 concurrent calls in Figure 5.6(c). These are the maximum concurrent calls during the simulation period. Some packets got the highest latency (above 16ms). The maximum traffic load is around 34%, 43% and 25% of network capacity respectively at this time.

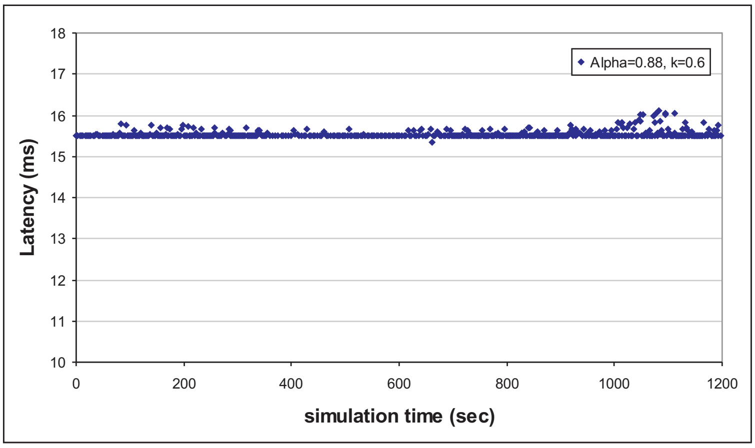
When the mean call interarrival time is 20 seconds, the phone system is moderately busy (see Figure 5.7, 5.8, 5.9). Much fewer packets have latency bigger than 15.5ms (see Figure 5.7(a), 5.7(b), 5.7(c)) and jitter bigger than 0.1ms (see Figure 5.8(a), 5.8(b) and 5.8(c)). When the simulation time is close to 1000 second, there are about 8 concurrent calls in Figure 5.9(a), 10 concurrent calls in Figure 5.9(b) and 6 concurrent calls in Figure 5.9(c). These are still the maximum concurrent calls during the simulation period. The maximum traffic load is around 6.4%, 8.6% and 5% of network capacity respectively at this time.

When the mean call interarrival time is 375 seconds, the phone system is not busy at all (see Figure 5.10, 5.11 and 5.12). No matter how long the conversations last, or in other words, no matter how the shape and scale parameters of Pareto distribution vary, the latency is very stable. It is always 15.5ms. The jitter for every packet is close to 0. The maximum concurrent call number is 4 within the simulation time (see Figure 5.12). The traffic load is always below 3%.

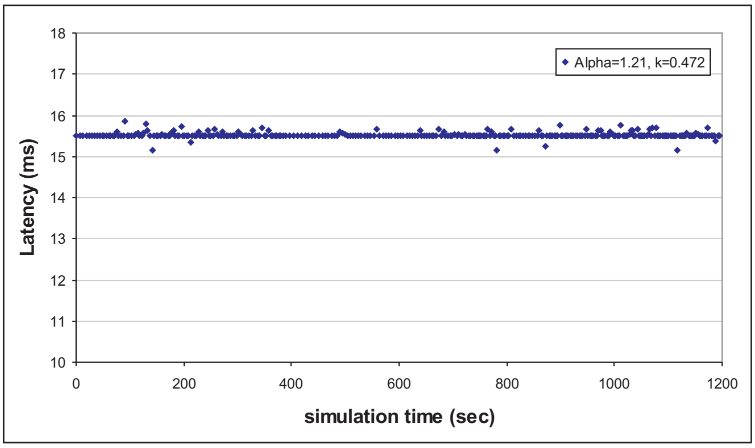
This experiment is significant for network expansion because if the simulation results indicate VoIP quality is degraded when too many concurrent calls run on a backbone link, then the backbone link capacity must be taken into account.



(a) α=1, k=0.547 (09:00 – 10:00)

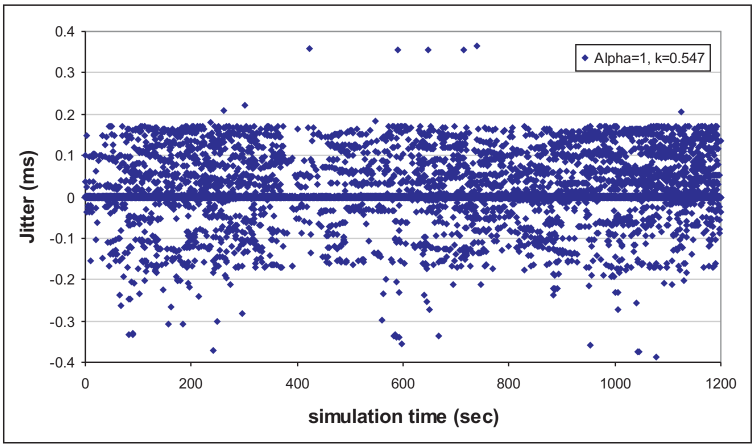


(b) α=0.88, k=0.6 (16:00 – 17:00)

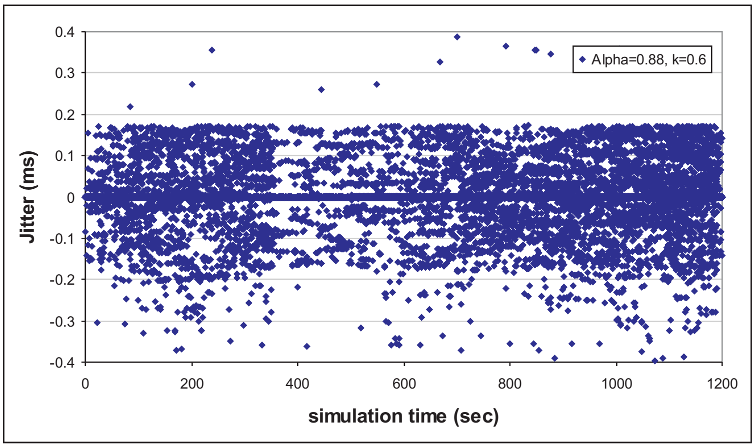


(c) α=1.21, k=0.472 (17:00 – 18:00)

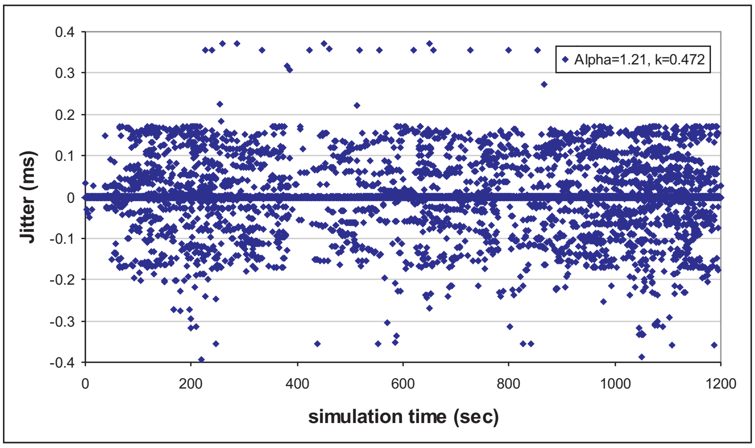
Figure 5.4: Delay, when the mean value of call interarrival time is 5 seconds



(a) α=1, k=0.547 (09:00 – 10:00)



(b) α=0.88, k=0.6 (16:00 – 17:00)



(c) α=1.21, k=0.472 (17:00 – 18:00)

Figure 5.5: Jitter, when the mean value of call interarrival time is 5 seconds

The packet loss values are displayed in Table 5.1. In order to compare with a lower speed network, the packet loss values were also obtained from a simulated 1Mbps network with 10ms delay for each link. When the mean of call interarrival time is 5 second, the packet loss is 64% in the 1Mbps network. When the mean of call interarrival time is 20 seconds, the packet loss is only 4%. From this table, it can be concluded that if there is a lot of traffic on a low speed network, the packet loss is huge. Compared to the 1Mbps network, there is no packet loss on a 10Mbps network no matter the call interarrival time is 5 seconds, 20 seconds or 375 seconds. It appears that there is bandwidth requirement for the network for VoIP services. If the network capacity is limited such as 1Mbps in this experiment, it is easier to cause packet loss which impairs VoIP quality most.

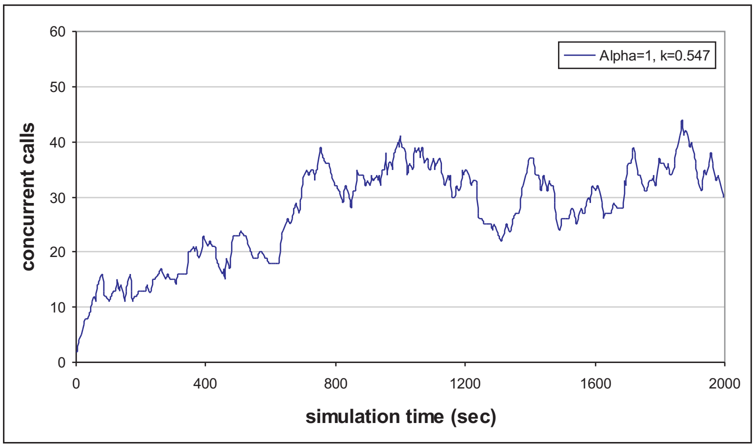
Table 5.1: Packet loss in simulated network

mean call interarrival time	**bandwidth**	
	1Mbps	**10Mbps**
λ=5sec	64.47%	0%
λ=20sec	4.1%	0%
λ=375sec	0%	0%

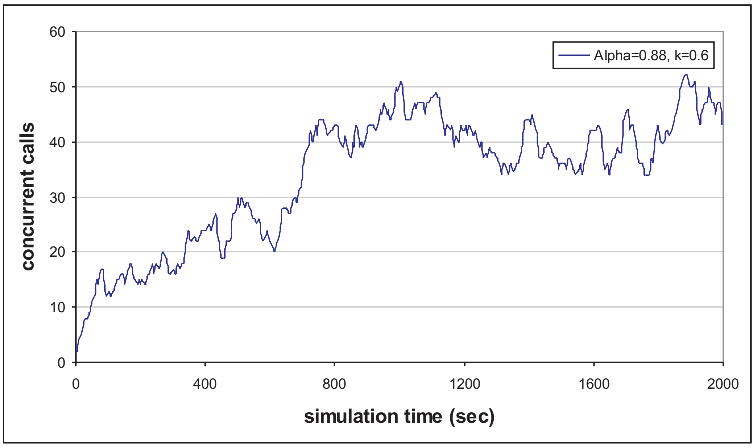
5.3.2 Experiment Two

This experiment simulates a real VoIP network environment in the Department of Computer Science at the UoS. The voice traffic pattern was obtained from the peak period from 9:00AM to 10:00AM. The aim of this experiment is to find out how VoIP quality metrics would be affected if VoIP is running on a data network. This is significant for integrating a VoIP network with a data network. If the simulation results reveal that the VoIP quality is degraded too much, then serious consideration should be made about the feasibility of the integration.

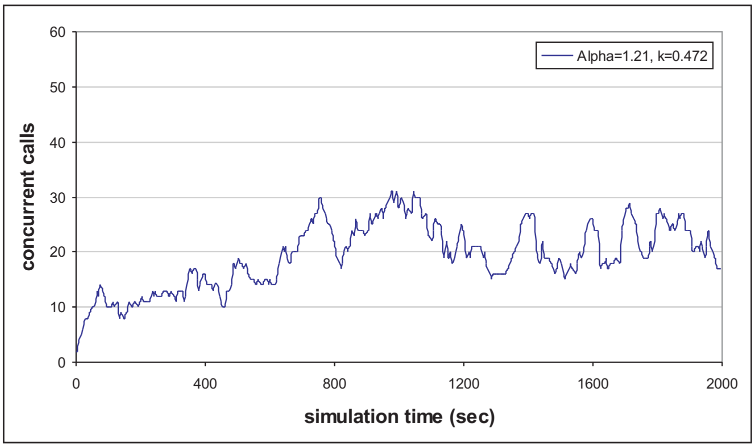
In the simulated 10Mbps network (bottleneck link is 10Mbps), 100 clients connect to a web server through the backbone link. To ensure the single link between switch 1 and switch 2 is the bottleneck link, which is required in dumbbell technology, the bandwidth between switch 1 and switch 2 is set to be 10Mbps. Each client sends HTTP request to the web server. The size of the requested web page is 48K. The distribution of HTTP request interval for every client is a Pareto distribution. When changing the average HTTP request interval time (think time), the web traffic load varies accordingly. In this experiment, when the average value of the think time is 26 seconds and shape parameter is 1.5, the average web traffic load is 20% of network capacity (see Figure 5.15(a)). Here the TCP traffic is measured in both directions including Web server to clients and the web requests as well. When the average value is 13 second and shape parameter is 1.5, the average web traffic load is 40% of network capacity (see Figure 5.15(b)). When the average value



(a) α=1, k=0.547 (09:00 – 10:00)

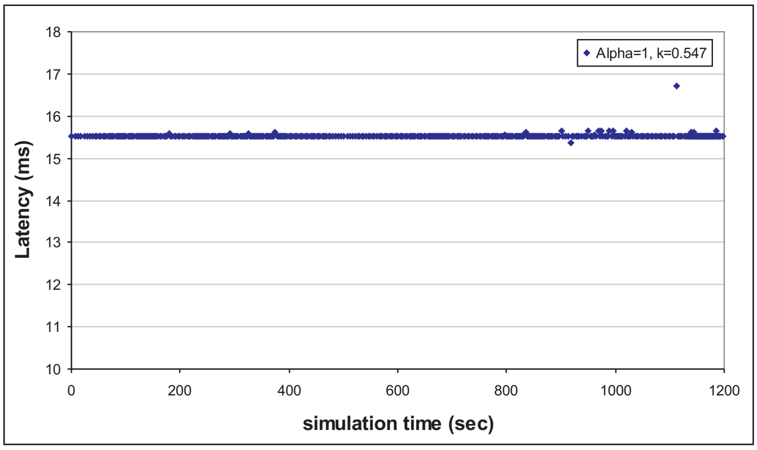


(b) α=0.88, k=0.6 (16:00 – 17:00)

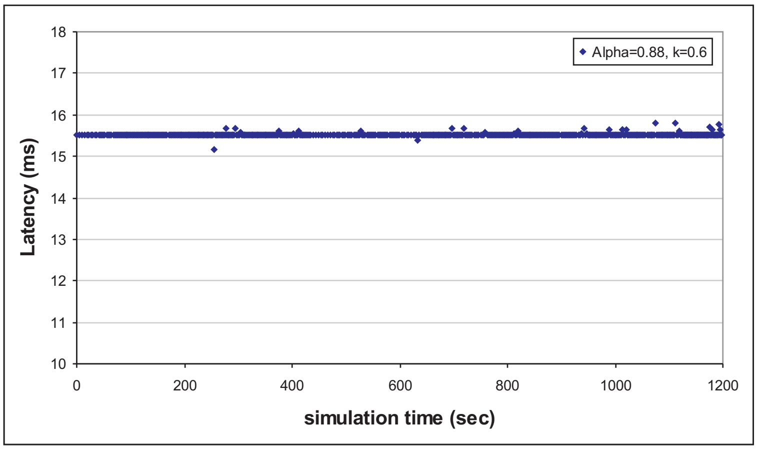


(c) α=1.21, k=0.472 (17:00 – 18:00)

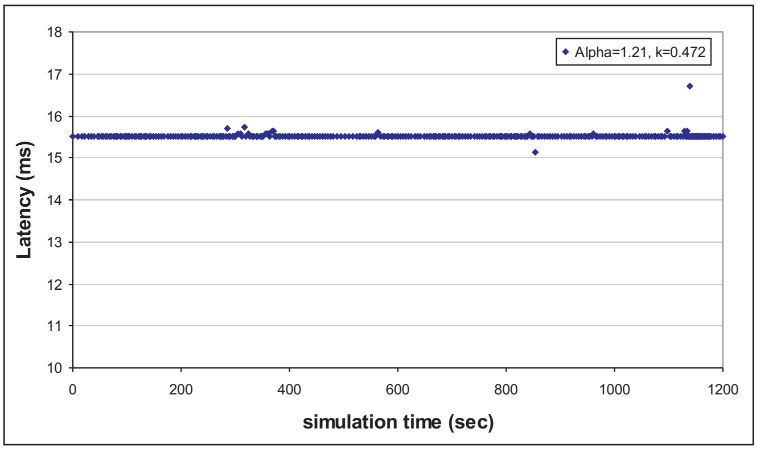
Figure 5.6: Concurrent call numbers with mean call interarrival time of 5 seconds



(a) α=1, k=0.547 (09:00 – 10:00)

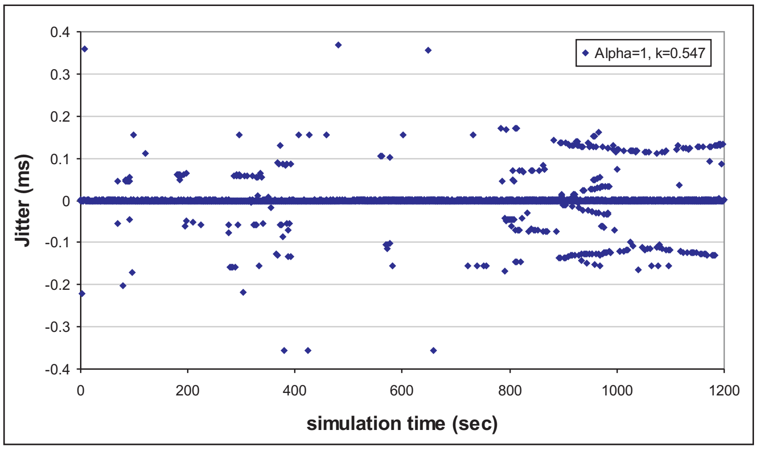


(b) α=0.88, k=0.6 (16:00 – 17:00)

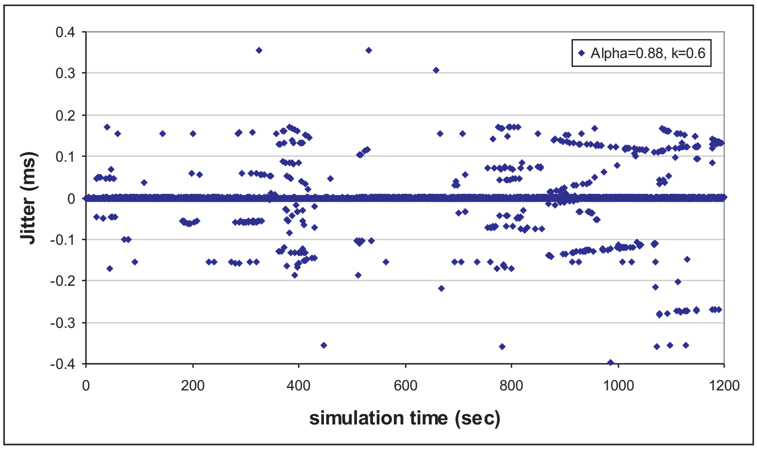


(c) α=1.21, k=0.472 (17:00 – 18:00)

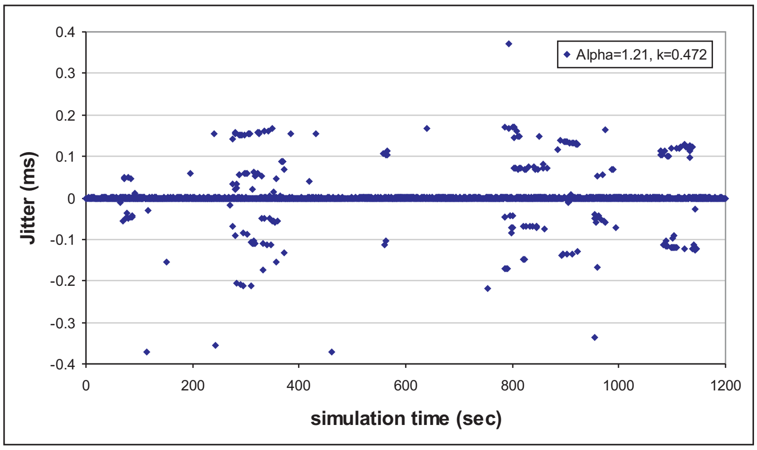
Figure 5.7: Delay, when the mean value of call interarrival time is 20 seconds



(a) α=1, k=0.547 (09:00 – 10:00)

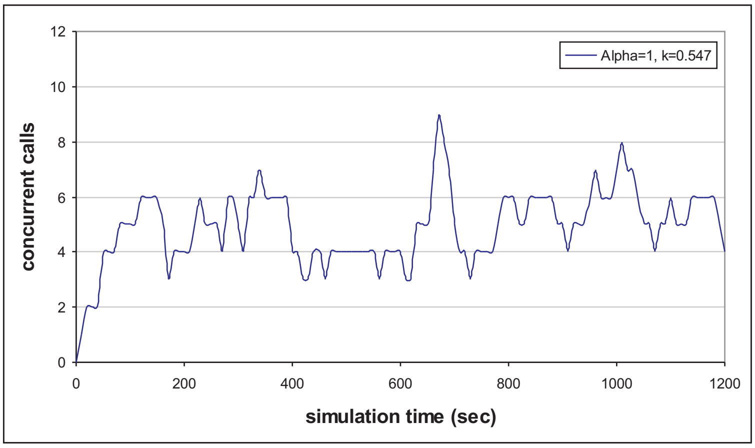


(b) α=0.88, k=0.6 (16:00 – 17:00)

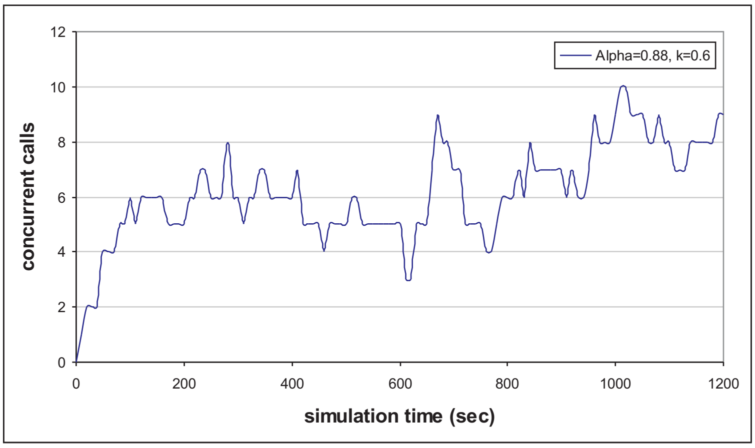


(c) α=1.21, k=0.472 (17:00 – 18:00)

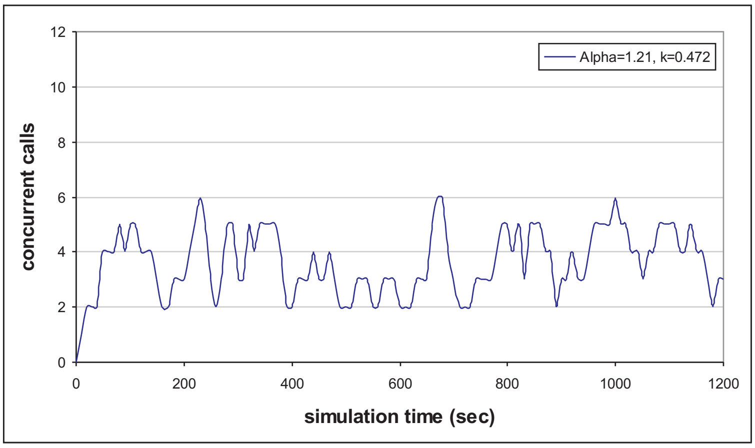
Figure 5.8: Jitter, when the mean value of call interarrival time is 20 seconds



(a) α=1, k=0.547 (09:00 – 10:00)



(b) α=0.88, k=0.6 (16:00 – 17:00)



(c) α=1.21, k=0.472 (17:00 – 18:00)

Figure 5.9: Concurrent call numbers with mean call interarrival time of 20 seconds

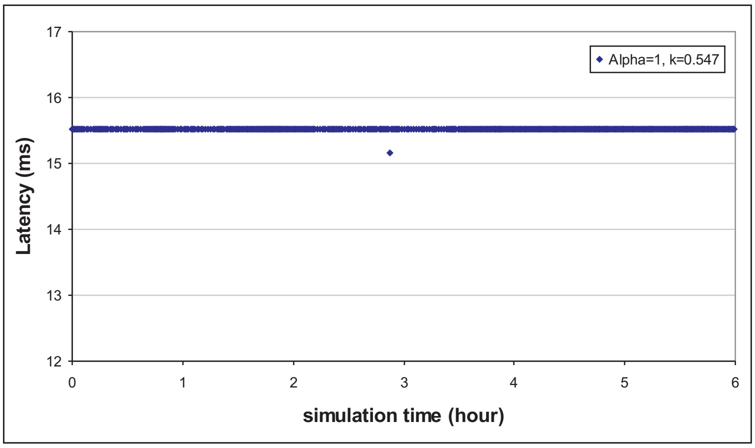


Figure 5.10: Delay, when the mean value of call interarrival time is 375 seconds, α=1, k=0.547 (09:00 – 10:00)

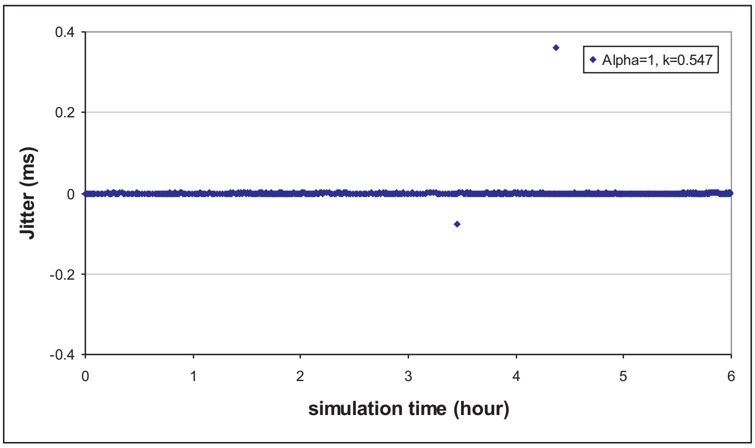
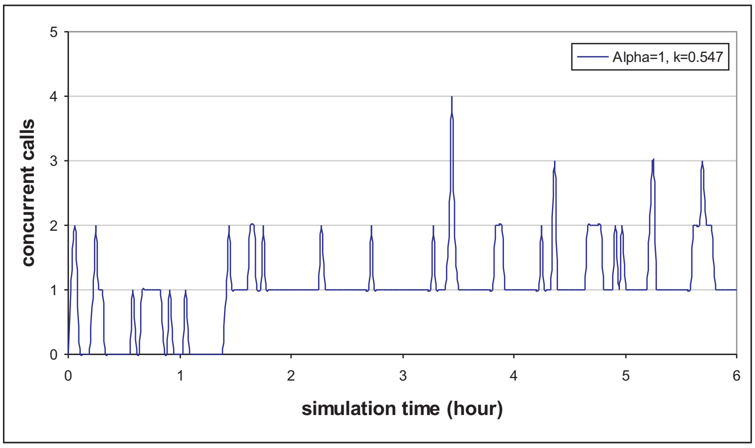
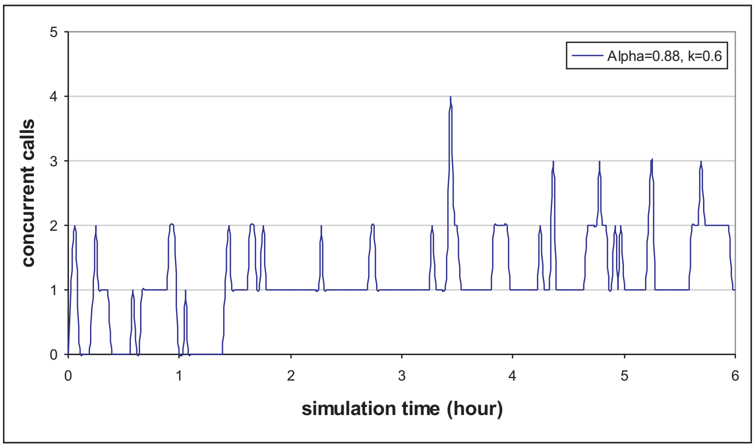


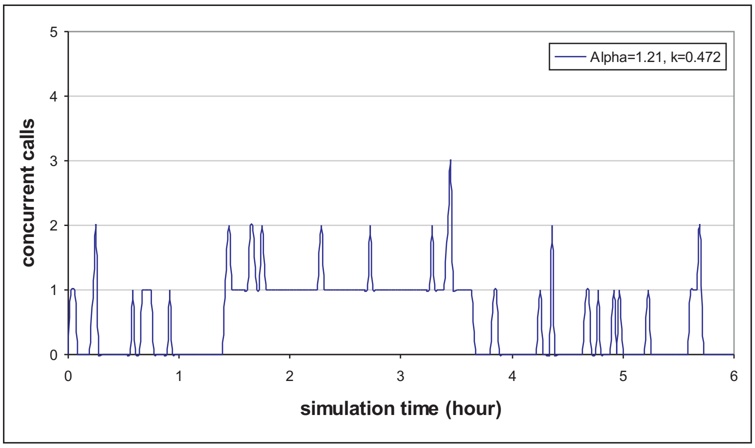
Figure 5.11: Jitter, when the mean value of call interarrival time is 375 seconds, α=1, k=0.547 (09:00 – 10:00)



(a) α=1, k=0.547 (09:00 – 10:00)



(b) α=0.88, k=0.6 (16:00 – 17:00)



(c) α=1.21, k=0.472 (17:00 – 18:00)

Figure 5.12: Concurrent call numbers with mean call interarrival time of 375 seconds

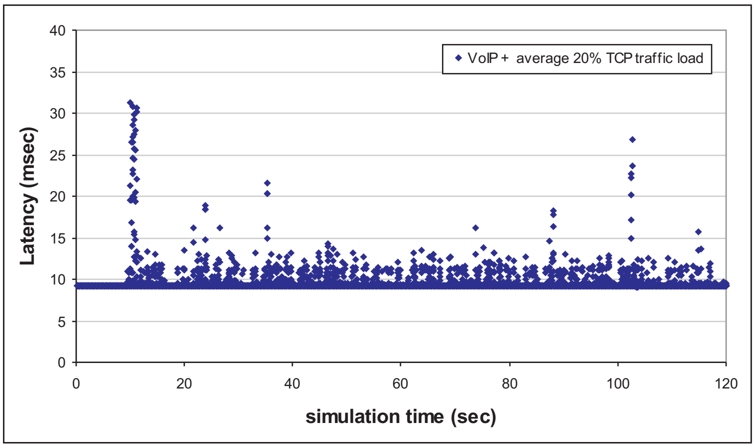
is 6 second and shape parameter is 1.5, the average web traffic load is 80% of network capacity (see Figure 5.15(c)). The delay and jitter value of VoIP packets are calculated for the different web traffic load levels.

When the web traffic load increases, more and more packets experience higher latency (see Figure 5.13(a), 5.13(b) and 5.13(c)) and higher jitter (see Figure 5.14(a), 5.14(b) and 5.14(c)). The jitter values are distributed evenly above and below zero. Though more packets experience higher jitter, the jitter value of the majority of packets are still smaller than 10ms which is acceptable even when the TCP traffic load increases to 80%. When the web traffic load is 80% of network capacity, there are 0.3% lost packets.

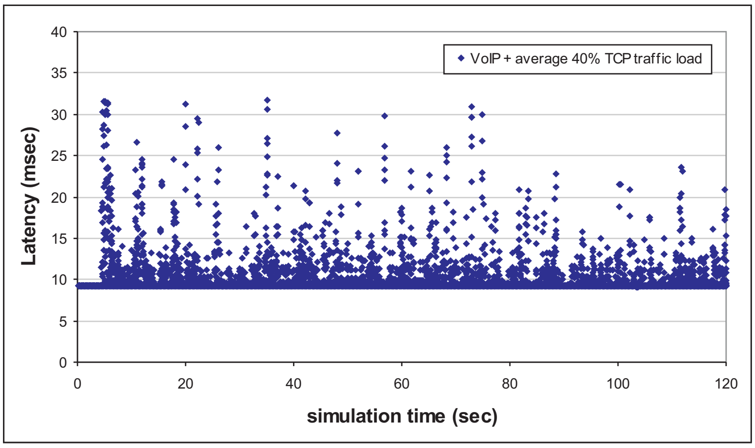
From above, it can be concluded that when the TCP traffic load is below 80% of the network capacity, the impact on VoIP quality is tolerable in this simulated environment.

5.4 Summary

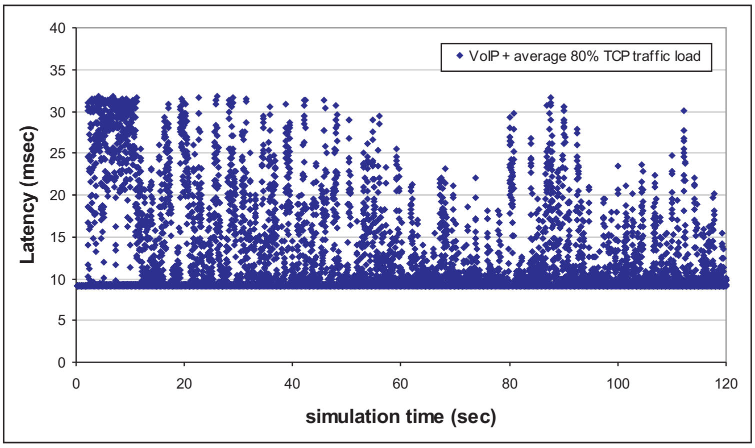
Two simulation experiments are presented in this chapter. The first experiment simulates the traffic of multiple calls running on a backbone link. The second experiment simulates a real VoIP network environment. Since the data used to provide the traffic model parameters was chosen from the peak traffic periods in the captured data from the department of Computer Science deployment of UofS, it is very helpful for the campus network designer to consider potential network integration or expansion informed by the simulation results. Several questions should be taken into account such as "what is the maximum concurrent call number on campus network" and "what is the maximum data traffic load". Occasionally measuring VoIP quality subjectively will also help to maintain VoIP services.



(a) 20% web traffic load

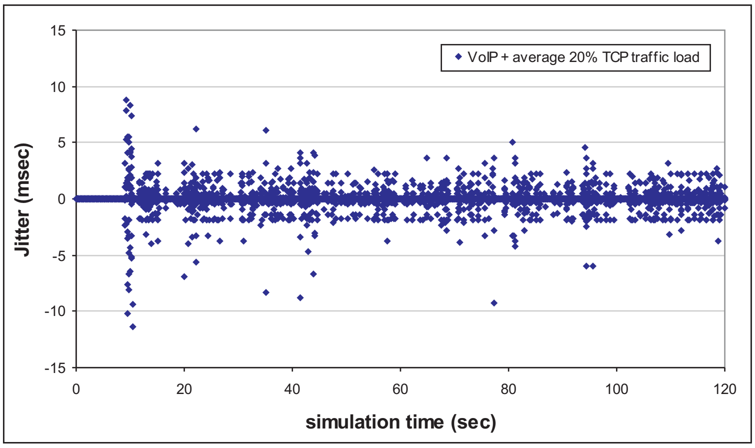


(b) 40% web traffic load

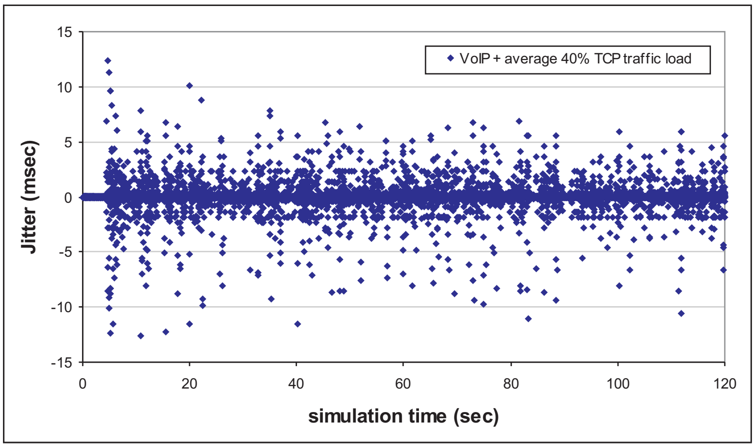


(c) 80% web traffic load

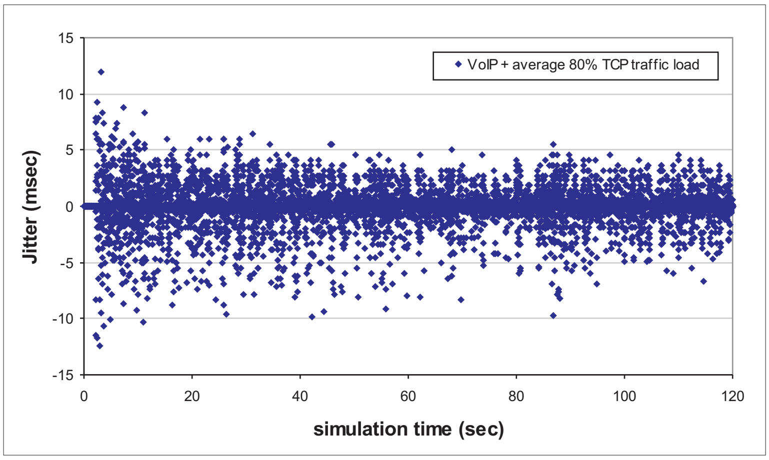
Figure 5.13: Delay of VoIP packets



(a) 20% web traffic load

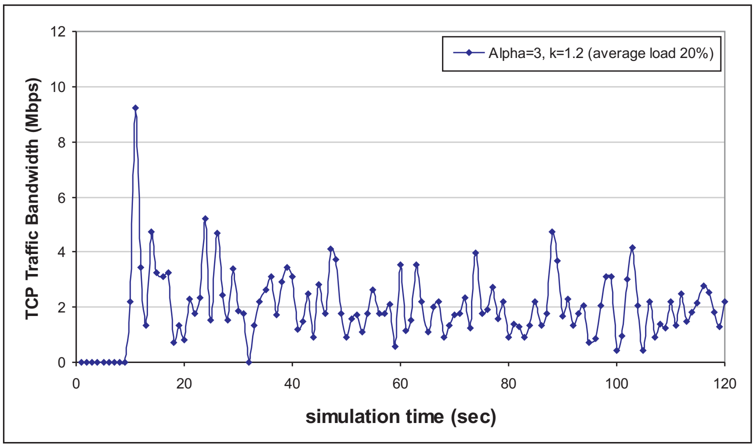


(b) 40% web traffic load

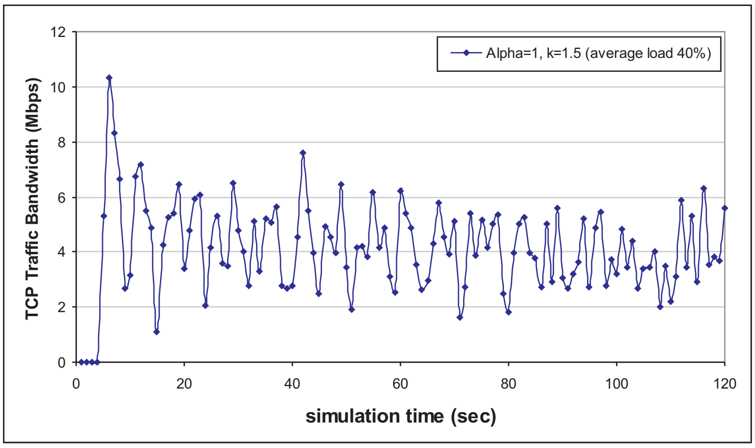


(c) 80% web traffic load

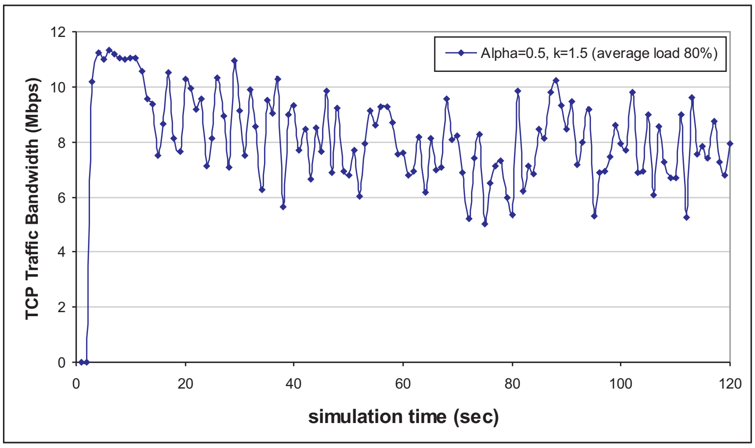
Figure 5.14: Jitter of VoIP packets



(a) 20% TCP traffic load



(b) 40% TCP traffic load



(c) 80% TCP traffic load

Figure 5.15: Web traffic load

Chapter 6

Conclusions

The telecommunications world has experienced a significant revolution over recent years. The convergence of data, voice, and video using IP-based networks is delivering advanced services at lower cost to residential users, business customers of varying sizes, and service providers. VoIP is one of the key technologies driving this convergence. With its lower cost and increased functionality than traditional landline telephone, VoIP has begun to be accepted by more and more residential users and business users today.

This thesis provided an experimental study on VoIP systems. It contains two parts: analysis experiments on the Vbuzzer soft phone system and the Cisco VoIP system, and simulation experiments. Section 6.1 summarizes each part of the thesis work. Section 6.2 states the main contribution of this research. Section 6.3 briefly outlines future research directions.

6.1 Thesis Summary

6.1.1 Analysis Experiments

In this work, an experimental study on a deployed Cisco VoIP phone system and a SIP based soft phone system are presented. Traffic was captured in Vbuzzer, a SIP based soft phone system, through which elementary understanding about a VoIP system was obtained and experimental setup was validated. VoIP traffic trace was collected over three months in a production Cisco VoIP system in the Department of Computer Science at the University of Saskatchewan. The distribution of packet type, packet size, call interarrival time, call duration time, bandwidth usage on different floors were measured from the trace data. Call patterns such as number of calls per hour, per week and per day were also measured. The delay and jitter value for each call was also obtained from the captured traffic data.

6.1.2 Simulation Experiments

A simulation study on VoIP quality metrics delay, jitter and packet loss was also conducted in this research. Two simulation experiments were performed.

The first experiment simulated the traffic of multiple calls running on a backbone link. The mean of call interarrival time was set to be 5 seconds, 20 seconds and 375 seconds respectively and the parameters for Pareto distribution was varied to obtain different call duration time. Delay and jitter values were measured in different scenarios. As the mean value of call interarrival time decreases, more calls happen within the simulation time. The traffic volume (load) increases accordingly. As the traffic load increases, more packets will experience relatively higher latency and jitter. This experiment is significant for network expansion because if the simulation results indicate VoIP quality is degraded when too many concurrent calls run on a backbone link, then the backbone link capacity must be taken into account.

The second experiment simulated a real network environment with different traffic load patterns. The aim of this experiment is to find out how VoIP quality metrics would be affected if VoIP is running on a data network. When the simulated web traffic load increases from 20%, 40% to 80%, more and more packets experience higher latency (above 15ms) and jitter (5ms). The measured latency and jitter values indicate that when the TCP traffic load is below 80% of the network capacity, the impact on VoIP quality is tolerable in the simulated environment. This experiment is significant for integrating a VoIP network with a data network. If the simulation results reveal that the VoIP quality is degraded too much, then serious consideration should be made about the feasibility of the integration.

6.2 Thesis Contribution

The main contributions of this thesis are:

- Though more and more VoIP systems are appearing, relatively little is known about the traffic characteristics of the Cisco VoIP system in a deployed environment. This work seems to be one of the first real deployment studies of VoIP that doesn't rely upon artificial traffic.

- Through the trace data, some features of Cisco VoIP phone system are understood such as the call setup and tear down process, distribution of traffic data and network performance. Some other facts such as the call interarrival time matching with an Exponential distribution and the call duration time matching with a Pareto distribution were validated. The analysis method used in this research can be used for developing synthetic workload model.

- The data collection results offer extensive insight into VoIP network usage. A clear understanding of traffic patterns in a real VoIP network is important for evaluating design principles, network deployment and potential network optimizations and expansion.

- The major factors that affect VoIP quality such as delay, jitter and packet loss are also measured and simulated in this study. Based on the measured data, the overall quality of

VoIP calls was judged to be excellent. This proves that the deployment of the Cisco VoIP network is very successful. The simulation results presented in this study is helpful for network design, potential network integration, expansion and advanced VoIP quality studies.

6.3 Future Work

Some possible future research directions include:

- Further analyses could be carried out using the measurement data captured from the University of Saskatchewan VoIP installation such as distinguishing among the individual phones on each floor. It is interesting to examine such statistics as the distribution function of the number of calls made to/from each phone from the raw traffic data.
- A synthetic workload model for VoIP could be further developed, based on the workload characteristics identified in the measurement study.
- Based on the simulation studies in Chapter 5, further simulation studies could assess VoIP performance under more realistic conditions, for example with respect to the arrival processes and duration distributions of TCP and UDP background flows, and network topology.
- By studying how network health factors affect quality of phone calls, the subjective quality of VoIP calls as a function of network health factors and other parameters can be evaluated. The basic methodology is shown in Figure 6.1.

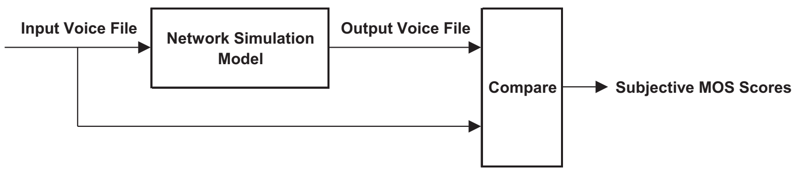


Figure 6.1: Methodology for subjective evaluation of VoIP quality

By comparing the new voice file with the original voice file, objective quality score can be obtained and mapped to subjective quality score. In this way, VoIP quality can be measured or predicted.

References

[1] A. Balachandran, G.M. Voelker, P. Bahl, and P.V. Rangan. Analysis of a campus-wide wireless network. In *Proc. of ACM SIGMETRICS'02*, pages 195–205, Los Angeles, CA, June 2002.

[2] S.A. Baset and H. Schulzrinne. An analysis of the Skype peer-to-peer internet telephony protocol. Technical Report CUCS-039-04, Department of Computer Science, Columbia University, New York, NY, December 2004.

[3] C. Boutremans, G. Iannaccone, and C. Diot. Impact of link failures on VoIP performance. In *Proc. of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video. NOSSDAV 2002*, pages 63–71, Miami, FL, May 2002.

[4] P. Calyam and C. Lee. Characterizing voice and video traffic behavior over the internet. In *Proc. of 20th Int'l Symp on Computer and Information Sciences(ISCIS)*, Istanbul, Turkey, October 2005.

[5] P. Calyam, M. Sridharan, W. Mandrawa, and P. Schopis. Performance measurement and analysis of H.323 traffic. In *Proc. of Passive and Active Measurement Workshop (PAM)*, pages 137–146, Antibes Juan-les-Pins, France, 2004.

[6] A.D. Clark. Modeling the effects of burst packet loss and recency on subjective voice quality. In *Proc. of IP-Telephony Workshop*, New York, NY, April 2001.

[7] D.E. Comer. *Internetworking with TCP/IP.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[8] A.E. Conway. A passive method for monitoring voice-over-ip call quality with itu-t objective speech quality measurement methods. In *Proc. of IEEE International Conference on Communications*, pages 2583–2586, New York, NY, April 2002.

[9] A.E. Conway, S.B. Moon, and P. Skelly. Synchronized two-way voice simulation tool for internet phone performance analysis and evaluation. In *Proc. of the 9th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools*, pages 108–122, St. Malo, France, June 1997.

[10] A.E. Conway and Y. Zhu. Analyzing Voice-over-IP subjective quality as a function of network QoS: a simulation-based methodology and tool. In *Proc. of Computer Performance Evaluation. Modelling Techniques and Tools. 12th International Conference, TOOLS 2002*, pages 289–308, London, UK, April 2002.

[11] L. Ding and R.A. Goubran. Speech quality prediction in VoIP using the extended E-model. In *Proc. of IEEE GLOBECOM*, pages 129–134, San Francisco, CA, December 2003.

[12] J.R. Ensor and S.R. Ahuja. VoIP: what is it good for? *ACM Queue*, 2(6):48–55, September 2004.

[13] ITU-T Recommendation G.107. The e-model, a computational model for use in transmission planning, December 1998.

[14] ITU-T Recommendation G.723.1. Dual rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit/s, March 1996.

[15] ITU-T Recommendation G.729. Coding of speech at 8 kbit/s using conjugate-structure algobraic-code-excited linear-prediction(cs-acelp), March 1996.

[16] ITU-T Recommendation H.323. Infrastructure of audiovisual services- systems and terminal equipment for audiovisual services, series H: Audiovisual and multimedia systems, 1999.

[17] J.B. Hooper and M.J. Russell. Objective quality analysis of a voice over internet protocol system. *IEE Electronics Letters*, 36(22):1900–1901, October 2000.

[18] W. Jiang and H. Schulzrinne. Assessment of VoIP service availability in the current Internet. In *Proc. of the 4th International Workshop on Passive and Active Network Measurement(PAM)*, La Jolla, CA, April 2003.

[19] A.B. Johnston. *SIP: Understanding the Session Initiation Protocol.* Artech House Inc., Boston, MA, 2004.

[20] M.J. Karam and F.A. Tobagi. Analysis of the delay and jitter of voice traffic over the internet. In *Proc. of IEEE INFOCOM 2001*, volume 2, pages 824–833, Anchorage, AK, April 2001.

[21] N. Kitawaki, K. Nagai, and T. Yamada. Objective quality assessment of wideband speech coding. *IEICE Transactions on Communications*, v E88-B(n3):1111–1117, 2005.

[22] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M.F. Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, August 2000.

[23] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proc. of the 8th Annual International Conference on Mobile Computing and Networking*, pages 107–118, Atlanta, GE, September 2002.

[24] C. Long. *IP network design.* Osborne/McGraw-Hill, Berkeley, CA, 2001.

[25] O. Macshima, Y. Ito, M. Ishikura, and T. Asami. A method of service quality estimation with a network measurement tool. In *Proc. of IEEE International Performance, Computing and Communications Conference - IPCCC'99*, pages 201–209, Scottsdale, AZ, February 1999.

[26] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam. Assessment of VoIP quality over Internet backbones. In *Proc. of IEEE Infocom 2002*, pages 150–159, New York, NY, June 2002.

[27] I. Marsh, F. Li, and G. Karlsson. Wide Area Measurements of Voice over IP Quality. In *Proc. of Quality of Future Internet Services(QofIS)*, pages 93–101, 2003.

[28] J. Mullin, L. Smallwood, A. Watson, and G. Wilson. New techniques for assessing audio and video quality in real-time interactive communications. *IHM-HCI Tutorial*, 2001.

[29] O. Olorunda and A. Olorunda. Bridging the Digital Divide -c The Social and Cultural Impact of VoIP in Developing Countries: Nigeria as a Case Study. In *Proc. of PTC'06*, Honolulu, Hawaii, January 2006.

[30] ITU-T Recommendation P.800. Methods for subjective determination of transmission quality, 1996.

[31] ITU-T Recommendation P.861. Objective quality measurement of telephone-band (300-3400hz) speech codecs, 1998.

[32] C. Perkins, O. Hodson, and V. Hardman. A survey of packet loss recovery techniques for streaming audio. *IEEE Network*, 12(5):40–48, September 1998.

[33] M. Perkins, K. Evans, D. Pascal, and L. Thorpe. Characterizing the subjective performance of the ITU-T 8 kb/s speech coding algorithm-ITU-T G.729. *IEEE Communications Magazine*, 35(9):74–81, September 1997.

[34] J. Rosenberg, R. Mahy, and C. Huitema. TURN: traversal using relay NAT. *Internet draft, Internet Engineering Task Force*, July 2004.

[35] J. Rosenberg, H. Schulzrinne, G. Camarillo, A.R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. *Internet Engineering Task Force, RFC 3261*, June 2002.

[36] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN: Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators(NATs). *Internet Engineering Task Force, RFC 3489*, March 2003.

[37] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *Internet Engineering Task Force, RFC 1889*, January 1996.

[38] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *Internet Engineering Task Force, RFC 3550*, July 2003.

[39] D. Schwab and R. Bunt. Characterising the use of a campus wireless network. In *Proc. of IEEE Infocom 2004*, pages 862–870, Hongkong, China, March 2004.

[40] K. Singh and H. Schulzrinne. Peer-to-peer internet telephony using SIP. In *Proc. of the 15th International Workshop on Network and Operating Systems Support for Digital Audio and Video. NOSSDAV 2005*, pages 63–68, Stevenson, WA, June 2005.

[41] L. Sun and E. Ifeachor. Perceived speech quality prediction for voice over ip-based networks. In *Proc. of IEEE International Conference on Communications*, volume 4, pages 2573–2577, New York, NY, April 2002.

[42] L. Sun, G. Wade, B. Lines, and E. Ifeachor. Impact of packet loss location on perceived speech quality. In *Proc. of IP-Telephony Workshop*, New York, NY, April 2001.

[43] S. Tao, J. Apostolopoulos, and R. Guerin. Real-time monitoring of video quality in ip networks. In *Proc. of the International Workshop on Network and Operating Systems Support for Digital Audio and Video. NOSSDAV 2005*, pages 129–134, New York, NY, 2005.

[44] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. Zhang. Improving VoIP Quality through Path Switching. In *Proc. of IEEE Infocom 2005*, pages 2268–2278, Miami, FL, March 2005.

[45] K. Thompson, G.J. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network Magazine*, 11(6):10–23, November 1997.

[46] L. Thorpe and W. Yang. Performance of current perceptual objective speech quality measures. In *Proc. of IEEE Workshop on Speech Coding*, pages 144–146, Porvoo, Finland, June 1999.

[47] F.P. Zhang, O.W.W. Yang, and B. Cheng. Performance evaluation of jitter management algorithms. In *Proc. of Canadian Conference on Electrical and Computer Engineering 2001*, volume 2, pages 1011–1016, Toronto, ON, May 2001.