Impact of Queuing Schemes and VPN on the Performance of a Land Mobile Radio VoIP System

Vijayanand S. Ballapuram

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

In

Computer Engineering

Dr. Scott F. Midkiff, Chair

Dr. Luiz A. DaSilva

Dr. Jung-Min Park

April 16, 2007

Blacksburg, Virginia

Keywords: Land Mobile Radio, Voice Over IP, Quality of Service, Virtual Private Networks

Impact of Queuing Schemes and VPN on the Performance of a Land Mobile Radio VoIP System

Vijayanand S. Ballapuram

(ABSTRACT)

Land mobile radio (LMR) systems are used for communication by public safety and other government and commercial organizations. LMR systems offer mission-critical or even life-critical service in the day-to-day activities of such organizations. Traditionally, a variety of different LMR systems have been deployed by different organizations, leading to a lack of radio interoperability. A voice application that connects LMR systems via a packet-switched network is called an LMR Voice over IP (LMRVoIP) system and is a potential solution to the interoperability problem. LMRVoIP systems are time critical, i.e., are delay and jitter sensitive. Transmission of LMRVoIP traffic in a congested packet-switched network with no quality of service (QoS) or priority mechanisms in place could lead to high delays and extreme variations in delay, i.e., high jitter, thus resulting in poor application performance. LMRVoIP systems may also have performance issues with the use of virtual private networks (VPNs). To the best of our knowledge, there has been no prior thorough investigation of the performance of an LMRVoIP system with different queuing schemes for QoS and with the use of VPN. In this thesis, we investigate the performance of an LMRVoIP system with different queuing schemes and with the use of VPN.

An experimental test bed was created to evaluate four QoS queuing schemes: first-in first-out queuing (FIFO), priority queuing (PQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ). Quantitative results were obtained for voice application throughput, delay, jitter, and signaling overhead. Results show that, compared to a baseline case with no background traffic, LMRVoIP traffic suffers when carried over links with heavy contention from other traffic sources when FIFO queuing is used. There is significant packet loss for voice and control traffic and jitter increases. FIFO queuing provides no QoS and, therefore, should not be used for critical applications where the network may be congested. The situation can be greatly improved by using one of the other queuing schemes, PQ, WFQ, or CBWFQ, which perform almost equally well with one voice flow. Although PQ has the best overall performance, it tends to starve the background traffic. CBWFQ was found to have some performance benefits over WFQ in most cases and, thus, is a good candidate for deployment.

The LMRVoIP application was also tested using a VPN, which led to a modest increase in latency and bandwidth utilization, but was found to perform well.

Acknowledgements

I would like to thank my advisor Dr. Scott Midkiff for providing invaluable guidance and support during my graduate study and the course of this thesis research. I am grateful to Dr. Midkiff for giving me an opportunity to work with him, as well as for his mentoring, genuine commitment, and constructive suggestions.

I also would like to thank my other committee members, Dr.Luiz DaSilva and Dr.Jung-Min Park, for serving in my committee and for their time and co-operation in reviewing this work. My thanks also go to Catalyst Communication Technologies, which supported me as a graduate research assistant during part of my M.S. program.

I would like to thank fellow students in the Networking Lab for their companionship. Special thanks to my parents and my sister for their unflagging support and encouragement, without whom none of this would have been possible.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

This introductory chapter begins with a discussion of the general problem area and motivation for this research. The problem statement is then discussed in Section 1.2. The last section explains the organization of the thesis.

1.1 Problem Area and Motivation

In the near future, telephone traffic will be just another application running over the Internet [Goo02]. In Voice over IP (VoIP) technology, voice traffic is transmitted in a shared packet-switched network infrastructure rather than through a circuit-switched telephony network with dedicated capacity. There is a potential cost savings with the use of packet-switched networks as there is no need for dedicated voice circuits which are typically underutilized. A packet-switched network can integrate voice and data traffic in a shared network infrastructure. This combined operation can also reduce personnel expenses because there is no need for dedicated personnel for separate voice and data networks . Personnel are among most significant expense elements in a network [DB03]. The convergence of voice and data can also reduce operating costs and increase network efficiency. An integrated infrastructure that supports all forms of communication can allow more standardization and reduce network complexity [DB03]. Thus, VoIP reduces total cost of ownership (TCO). In packet-switched networks, the necessary transmission bandwidth for voice can be reduced by silence detection and voice compression techniques which add to the statistical multiplexing gain inherent in a packet-switched network [Sl03]. Coder/decoders (codecs), like G.723.1 [ITU96a] and G.729 [ITU96b], provide toll quality speech at much lower data rates (5.3/6.3 kbit/s and 8 kbit/s, respectively, for G.723.1 and G.729) than conventional pulse code modulation (PCM) encoding (64 kbit/s) used in circuit-switched networks.

A key advantage of VoIP compared to circuit-switched voice is the potential for integration with other network services. Although telephony is the basic application for VoIP networks, the long-term benefits are expected to be derived from multimedia and multi-service applications [DPT03]. VoIP allows companies to better serve their customers by providing a host of new converged voice and data applications, such as web-enabled call centers, unified messaging, real-time multimedia and audio conferencing, distance learning, and embedding voice links into electronic documents. Other advantages of VoIP include in-house control, robust system management, greater details on call history, the ability to quickly deploy multiple voice lines with a lower TCO, and the ability of companies to better manage remote employees. In addition, VoIP can rapidly scale to meet the increasing demands of a growing organization.

There are some problems, too, in deploying VoIP. VoIP applications must operate with real-time constraints and, in particular, are sensitive to delay and delay variation, also known as jitter. The goal of VoIP is to provide speech quality at least equal to that of the public switched telephone network (PSTN), which is usually referred as “toll-quality” voice. This is a challenge as the present Internet is based on best-effort service, meaning that packets are serviced on a first-come first-served basis. Also, there are chances that during peak loads voice packets will experience high delay and jitter. There can also be loss of voice packets due to the heavy loading of links and network congestion caused by link failure or insufficient network capacity. Echo and talker-overlap can result from high end-to-end delay in a voice network. Since, echo degrades voice quality, echo cancellers must be deployed to perform echo cancellation. Talker overlap is a problem where one caller steps on or seeps into another talker's conversation and it become significant if the one-way delay becomes greater than 250 milliseconds [DPT03]. Jitter is the variation in the inter-packet arrival time and a high jitter value is more detrimental to voice quality than an equal delay value. By deploying network quality of service (QoS), a network can be configured to provide priority to real-time applications, such as voice, to provide good application performance. QoS mechanisms can significantly reduce delay and jitter and increase application throughput when compared to a network with no QoS mechanisms in place. Ideally, a QoS mechanism should minimize latency and packet-loss for delay-sensitive and loss-sensitive real-time traffic, while still allowing other traffic to achieve satisfactory performance under reasonable average network loads and sufficiently short bursts of background traffic. QoS can be achieved by properly managing queues at routers and by routing traffic around congested parts of the network. In general, QoS provides better and more predictable network service by providing the following features [Cis01]:

1) support for dedicated bandwidth;
2) improved loss characteristics;
3) avoidance and management of network congestion;
4) network traffic shaping; and
5) control of traffic priorities across the network.

1.2 Problem Statement

The focus of this thesis is Land Mobile Radio VoIP (LMRVoIP). The specific term Land Mobile Radio VoIP is coined in this thesis, but is not a new concept and has been implemented by several land mobile radio companies including Catalyst Communication Technologies, Motorola, MA-COM, Cisco Systems, EF Johnson, and JPS Communications. LMRVoIP has some differences compared to traditional VoIP. For example, LMRVoIP calls are burtsy when compared to VoIP calls where there tend to be longer-lived streams of traffic. Details about different types of VoIP, including LMRVoIP, are discussed in Section 2.1. There has been a significant amount of research in VoIP resulting in many conference papers, journal papers, and magazine articles. Even though most of the ideas and concepts of VoIP systems can be extended to LMRVoIP systems, the research in LMRVoIP is at a nascent stage and limited. There have been a few conference papers and thesis published at universities in which they model an LMR system using simulation models [Tsi02] or using an experimental test bed [MT02]. LMR systems are described in Section 2.1. Based on the developed models, researchers have perform different experiments to study the performance of LMRVoIP applications under various network scenarios [MT02, Tsi02]. There are three general areas of research that will influence the successful migration of conventional circuit-switched voice networks to integrated voice and data packet-switched networks [DB03]:

1) migration of traditional circuit-switched signaling schemes to a packet-switched signaling scheme that meets voice signaling performance requirements;

2) selection of appropriate voice coding and decoding methods so that the bandwidth utilization by the application can be reduced using silence detection and voice compression techniques; and
3) mechanisms for assembling and dissembling packet frames to transfer VoIP frames through a packet-switched network to meet performance requirements.

In this thesis, the effect of different QoS queuing schemes at routers on the performance of LMRVoIP is studied. The router queuing schemes give priority to important traffic by re-ordering the queues maintained at the output port of a router. Detailed explanations of the four different queuing schemes that were evaluated are given in Section 2.3. As discussed in Section 1.1, the metrics of interest are delay and delay variation. In addition, maximum jitter and signaling overhead are also used as performance metrics. The signaling overhead is measured in terms of the number of TCP packets used by the voice application for signaling.

The performance was evaluated using ON-OFF voice traffic, which is representative of traffic in an LMR system. In addition, the performance was also evaluated using continuous voice traffic, which presents the most demanding peak load conditions. An experimental test bed was created to perform the experiments. Although the effect of these queuing schemes on VoIP applications and other non-LMRVoIP applications has already been reported in the literature, for example in [KK01], the effect of these queuing schemes on an LMRVoIP application, where voice traffic characteristics are different, has not been studied.

Virtual private networks are used to ensure authenticated and private communication in a shared packet-switched network. When using a VPN, latency may increase due to extra packet processing and bandwidth consumption may increase due to encapsulation. The performance of an LMRVoIP application was tested with the use of VPN. This thesis is the first to report the performance of an LMRVoIP application with the use of VPN.

1.3 Organization of the Thesis

This thesis is organized in five chapters as follows.

In Chapter 2, three types of VoIP applications are discussed including LMRVoIP. The features of a particular LMRVoIP system, Catalyst IP Tone, are presented. The need for QoS mechanisms, circuit-switching, packet-switching, and the four different router queuing schemes are described. The chapter concludes with a discussion on the research and how it differs from prior research.

In Chapter 3, four specific questions answered by this thesis are presented. A high-level overview of different router queuing configurations, VPN configurations, and Catalyst IP Tone product configurations are presented. Finally, components of the experimental test bed, testing tools, and network configuration used for testing are discussed.

Chapter 4 starts with a discussion of key assumptions made in this research, followed by a discussion of traffic generation, test scripts, and test procedures. The configurations for the four different router queuing schemes and VPN are presented and the corresponding results are discussed.

Chapter 5 provides a summary of the thesis, draws conclusions that suggest possible situations where different queuing schemes should or should not be used. Finally, Chapter 5 discusses possible future research topics.

Appendix A provides the Cisco router configurations for the different queuing schemes. The EXPECT scripts to enable different queuing schemes are provided in Appendix B. Appendix C provides a TCL data analysis script to extract results from Ethereal network sniffer trace files. TCL and Ethereal are discussed in Chapter 3.

Chapter 2. Background and Related Work

This chapter discusses three types of voice over IP (VoIP) applications. Land mobile radio (LMR) voice over IP, or LMRVoIP, the main topic of this thesis, is discussed in detail. LMRVoIP products such as Catalyst's IP Tone product [Cat02b], MA-COM's NetworkFirst [Mac03c], and Motorola's ASTRO 25 [Mot04] and their features are described. The need for quality of service (QoS) mechanisms and the four different router queuing schemes considered in this research are also explained in detail. This chapter concludes with a discussion of related work and its relation to the research reported in this thesis.

2.1 Types of VoIP Applications and LMRVoIP

There are various types of VoIP applications. VoIP applications are classified based on several factors, such as the nature of transmission and the transmission media. The different types of VoIP applications are discussed in this section. LMR systems are used by public safety and other organizations. Emergency and other personnel use LMRVoIP to communicate with each other, as it can overcome the problem of the lack of radio interoperability and can provide access to remote LMR systems using a shared data network instead of dedicated leased lines. LMRVoIP systems are discussed in detail in this section.

2.1.1 VoIP Applications

There are various types of voice applications that use the Internet Protocol (IP) to carry encoded voice information. Three types of VoIP applications are internet telephony (IT), LMRVoIP, and multimedia streaming over IP (SoIP).

In internet telephony, both the called and the calling parties speak interactively, i.e. in full duplex mode, on their respective ends of a VoIP channel, as in a conventional public switched telephone network (PSTN) call. Calls are also similar in nature to PSTN calls, so they tend to relatively long lived, on the order of several minutes to even an hour or more. In LMRVoIP communication is half-duplex and bursty, with a typical call lasting for several seconds to minutes. A specific type of IT, which uses a wireless medium to carry encoded voice information, is called wireless VoIP (WVoIP). Data transmission in wireless media can be accomplished using wireless devices that follow wireless standards, such as IEEE 802.11 [Wla03a], IEEE 802.15 [Wla03b], or IEEE 802.16 [Wla03c]. Therefore, in WVoIP, one or both ends of a VoIP channel can be a notebook computer, handheld computer, or other device equipped with an appropriate wireless network interface card (NIC), such as an IEEE 802.11b or Bluetooth NIC.

Because of the interactive and bidirectional nature of IT, an IT call has stringent requirements for delay, delay variation, and bandwidth. Delay variation is also called jitter. Keagy reports that, for most people, the maximum acceptable delay is 250 ms [Kea00]. This translates to a requirement that the end-to-end delay of a voice packet should be less than 250 ms. The end-to-end delay of a packet is the total time taken by a packet to reach the destination after being sent by the source. Various delays occur due to encoding at the sender, decoding at the receiver, packetization of the compressed bit stream, waiting in queues along the path from sender to receiver, serialization at the interface, transmission of voice packets, and buffering for play out at the receiver. As delays rise over this figure, talkers and listeners become un-synchronized, and often they speak at the same time, or both wait for the other to speak [Cis07a]. This condition is commonly called talker overlap.

Emergency and other personnel in different agencies and other organizations can use LMRVoIP to communicate with each other. Unlike internet telephony, LMRVoIP inherently operates in half-duplex mode and the communication consists mostly of short bursts of conversations, with typical conversations lasting for several seconds to a minute. One of the characteristics of LMRVoIP is that the delay and jitter requirements are less stringent than IT and WVoIP because of the short duration of transmissions. However, reliability and robustness, in general, can be extremely important for public safety and other life-critical or mission-critical applications. LMRVoIP systems are discussed further in Section 2.1.2.

Another interesting type of VoIP application is SoIP. A streaming system is a VoIP system that does not support conversation, i.e., is strictly simplex in nature. It is a VoIP application because the playback of continuous media, with audio being of interest here, must occur in an isochronous fashion [Cur 03]. However, an SoIP application is different from IT and WVoIP due to several unique properties, such as one-way distribution, offline media encoding, and relative insensitivity to delay through the use of a sufficiently large playback buffer. The streaming audio flow is always unidirectional, from the streaming server to the client (in the downlink direction). Normally, the user has limited control over a streaming session and there is not a high level of interactivity between the client and the streaming server. Offline media encoding means that the encoding is done in advance using specific content creation tools such as Microsoft Media Player and Real Player. Because of these properties, an SoIP application is not as sensitive to delay and jitter as IT and WVoIP. Therefore, audio can be streamed and played after an initial latency period. This allows the client to use buffered data to smooth out eventual network jitter without compromising user-perceived audio quality.

2.1.2 LMRVoIP

Land mobile radio systems are used for communication by public safety and other government and commercial organizations [Tsi02]. Crucial service is offered by LMR systems in the day-to-day activities of public safety organizations. LMR systems also provide communications in mission-critical and life-threatening situations. Many commercial businesses, including utility and transportation companies, also rely heavily on LMR systems to mange and co-ordinate daily activities. LMR systems have evolved as autonomous two-way radio systems serving the specific needs and interests of different organizations [DSD01]. This led to the development of “stove pipe” systems that were confined to local scope. Through individual initiatives and requirements, each organization deployed their own independent LMR system that was often driven by both technical and political objectives. As a result, radio communication between emergency personnel from various departments or agencies using different radio systems is sometimes impossible due to lack of radio interoperability [MT02].

It is not cost-effective or feasible in the near term for all departments to each purchase new, compatible radios. Instead, interoperability solutions are being developed using LMRVoIP. LMRVoIP uses a packet-switched infrastructure rather than a conventional circuit-switched network to connect base stations to each other and to dispatchers. Vendors which provide LMRVoIP solutions include Catalyst Communications Technologies, M/A-COM, Motorola, Cisco Systems, EF Johnson, and JPS Communications. One solution developed by Catalyst Communications Technologies is called IP Tone, which is a client/server application [Cat02b]. The server, called the Radio Gateway, connects to a mobile radio. Each client runs on a standard personal computer (PC) running Microsoft Windows and can be located at a participating department [Cat02b]. The server hosts a voice conference that is compatible with the client software running on dispatch stations. The Catalyst system uses standard voice codecs provided with Microsoft Windows, but a proprietary control protocol. Alternatively, such a system could follow a standard, such as the H.323 standard, [Iec03] for all aspects of operation or the Session Initiation Protocol (SIP) [HS+99] for signaling. H.323 is a recommendation from the International Telecommunications Union (ITU) that sets standards for multimedia communications over IP based networks such as the Internet. SIP is an application layer signaling protocol used for creating, managing, and terminating multimedia sessions over packet networks. The M/A-COM's solutions include Network First and P25^{IP} [MAC02a] [MAC02b]. One of the the Motorola's solutions is the ASTRO 25 Trunked Digital Voice and Data Network [Mot04] [Tho03a]. Descriptions of different LMRVoIP solutions are discussed in section 2.2.

Conventionally, circuit-switched infrastructure has been used to connect LMR systems. Circuit-switched networks posses some fundamental disadvantages when compared to packet-switched networks, such as vulnerability to single-points of failure. Moreover, as bandwidth for a voice call is reserved for the duration of the call, network capacity is poorly utilized when there is not continuous conversation as in a typical LMR call. In addition, with the introduction of digital LMR systems, traffic patterns have become more bursty and non-uniform and, thus, less suitable for circuit-switched networks [Tsi02]. There are advantages of deploying a packet-switched LMRVoIP solution. Packet-switched systems easily support features such as mobility management, web browser access, individual calls, group calls, priority calls, text messaging, call-monitoring, and security. These services allow seamless wide-area coverage and resource sharing.

An LMR call generates push-to-talk (PTT) voice traffic, so only one party can speak at a time and the caller "hangs-up" after completing his or her call. Therefore, typical LMRVoIP traffic is not continuous. These characteristics make LMRVoIP service less sensitive to delay and jitter than IT and WVoIP because LMRVoIP transmissions are of short duration and bursty. Thus, to investigate the performance of an LMR voice application, on-off voice traffic scenarios that are representative of a LMR system are considered in this research. Continuous voice traffic scenarios are also considered to study the performance of a LMR system in the most demanding situation.

2.2 LMRVoIP Products

Catalyst Communication Technologies, Inc. (or, simply, Catalyst) provides LMRVoIP products for the land mobile radio industry to allow organizations to connect multiple personal computers

(PCs) to multiple radios for dispatch, help desk, and other office-based operations. One such product that is used in this research is IP Tone [Cat02b]. Other LMRVoIP products such as M/A-COM's NetworkFirst and Motorola's ASTRO 25 Trunked Digital Voice and Data Network are discussed.

2.2.1 Catalyst's IP Tone Product Overview

IP Tone provides a communication and control path between a PC and a two-way radio, providing connectivity via a local area network (LAN) or wide area network (WAN). The product allows office workers to communicate with field personnel who have two-way radios. IP Tone is a client-server application. The server component runs on a Radio Gateway connected to a land mobile radio or base station. The Server controls the radio using standard Electronic Industries Alliance (EIA) tone signaling for the push-to-talk function and converts the audio to packets that can be sent over a WAN or a LAN. The server is connected to one or more clients, called Remote PCs, via a LAN or WAN. The Remote PCs are used by operations center personnel as well as other office workers.

During a voice call, voice messages are sent back and forth between the Remote PC and the radio. The field radio user speaks into his two-way radio, transmitting audio to the fixed radio and the Radio Gateway (server) via the land mobile radio system. The Radio Gateway then converts the speech to a digital message and sends it over the WAN or LAN as one or more IP packets to the Remote PC. The Remote PC converts the digital message to analog voice and plays it through the computer's speakers. The Remote PC user can reply by clicking on the transmit button and speaking into microphone connected to his computer. The Remote PC converts the user's speech to a digital message and sends it over the WAN or LAN to the Radio Gateway. The Radio Gateway converts the digital messages to analog voice. The Radio Gateway routes the audio to the two-way radio for transmission on the radio system.

The product supports numerous other features for controlling talk groups, monitoring multiple radio systems from a single dispatch station, logging calls, controlling access to radio systems, and more [Cat02b]. However, these functions are not germane to the specific research presented here and are not discussed further.

2.2.2 M/A-COM's NetworkFirst Product Overview

Two LMRVoIP solutions provided by M/A-COM are NetworkFirst [Mac03c] and P25^{IP} [MAC02a]. NetworkFirst provides IP packet-switched voice communications among multiple agencies that need to interoperate. NetworkFirst combines a universal analog audio port, the interoperability gateway, with an IP-based voice switch and network administrator to connect multiple radio systems together in a seamless communications web. In conjunction with an optional console, the Network Switching center provides a centralized command point for crisis management.

NetworkFirst is advertised as being able to achieve interoperability across all frequencies and supporting all radio and system level voice types – analog, digital, conventional, and trunked [Mac03c]. NetworkFirst achieves large-scale interoperability by using IP and is supposed to be completely scalable; meaning communication interoperability at all levels – local, regional, state, and national – should be possible. M/A-COM claims that an IP-Based solution, such as NetworkFirst, is the most effective means of achieving interoperability, as it allows agencies to use their existing equipment, between disparate systems in a short amount of time.

2.2.3 Motorola's ASTRO 25 Product Overview

Motorola's primary solution for P25 [USD03] has been the ASTRO 25 Trunked Digital Voice and Data Network [Tho05] [Mot04], which builds on ASTRO [Mot99], its predecessor. ASTRO 25 is advertised as offering advantages such as [Mot04]:

1) Cost savings from combining voice and data into one efficient and flexible solution that allows for easy upgrades and migration as needs evolve. Voice over Packet networking, based on voice-over-IP (VoIP) technology, is supposed to allow the system to carry both voice and data communications efficiently and reliably by combining voice and data into a single dedicated network.
2) Software upgrades are promoted to be less difficult through centralized downloading. It should normally be sufficient to load the software once and then it is automatically distributed throughout one's network to support new features and software patches.
3) Relief from radio frequency congestion via trunked networking and allocating channels between voice and data as needed, thereby supporting more users, more calls, and more information using the same spectrum.
4) Interoperability with other Project-25 compliant solutions [USD03], so the system and personnel can work seamlessly with other departments that have compliant systems even if they utilize another vendor's solutions.

Although Motorola states that there should be no incomptabilities between their P25 solutions and other P25 products, they do imply in their information that there will be proprietary features that exist on their products that will not work on other manufacturer's products.

2.3 Circuit Switching Versus Packet Switching

Networks typically support either circuit switching or packet switching. Voice traffic has normally been carried over circuit-switched networks, such as the public switched telephone network, while packet-switched networks have been designed to carry non-real-time data traffic. However, to achieve cost savings and increase flexibility, voice and other applications are being designed to use packet-switched networks. These two switching schemes and issues related to their use for voice traffic are discussed below.

2.3.1 Circuit Switching

As discussed in Section 2.1, real-time applications such as VoIP have certain requirements for delay, jitter, and bandwidth. In circuit-switched networks, there is end-to-end synchronous transmission. Digitized voice experiences a minimal end-to-end delay as it traverses through the switches in a circuit-switched network which offers reserved bandwidth. Another important property of digitized voice in a circuit-switched network is jitter. The reserved capacity also means that the jitter experienced by digitized voice in a circuit-switched network is minimal, even considering latency introduced by processes such as voice digitization and switching. These two qualities, low delay and minimal jitter represent two important QoS metrics [Hel00]. The third QoS metric that is provided by a circuit-switched network is bandwidth. Traditionally, a pulse coded modulation (PCM) codec is used by telephone systems at a data rate of 64 Kbps.

A uniform and dedicated bandwidth of 64 Kbps is allocated to each voice call. For other codecs, a suitable capacity is also dedicated to the call to eliminate queuing delay and to minimize jitter. The fourth QoS metric is loss and a properly deployed circuit-switched network inherently offers low loss rates. Forward error correction (FEC) can be used to ensure extremely low loss rates.

Even though, circuit-switching provides excellent QoS for real-time communications, its design is relatively inefficient. In a typical conversation, only one person speaks at a time and the other person is silent, i.e. a conversation is normally half-duplex. Even for the speaker, there are times when he or she is silent. Also, highly effective codecs are available that generate variable bit rate traffic, i.e., the voice stream is encoded using the minimal data rate which may vary over time. Since, a fixed bandwidth is allocated for each call in a circuit-switched network, bandwidth is wasted during silent periods or when the codec is able to encode the voice at a rate below some maximum. Also, circuit-switched networks are often vulnerable to single points of failure because it is relatively expensive to provide redundant links and the switch to a backup link must be relatively quick.

2.3.2 Packet Switching

Unlike a circuit-switched network where each user is guaranteed a fixed amount of bandwidth, multiple users share the same network capacity in a packet-switched network. Although packet switching increases the efficiency of network utilization, it introduces several new problems. The present Internet is based on best-effort service, which means traffic that flows through the Internet is serviced based on a first-come, first-served basis with no quality of service guarantees. If real-time applications, such as voice or video applications, are transmitted along with non-real time applications, such as file transfer protocol (FTP) and hypertext transport protocol (HTTP) traffic, and the total network traffic exceeds the capacity of a link, there will be potentially high queuing delays for packets from both the real-time applications as well for the non-real-time applications. If the congestion persists for a long enough period of time, buffers may fill and both types of packets may be dropped. Since the generic Internet is based on best-effort service, packets from both applications have an equal chance of being serviced and of being dropped. Therefore, there is no way to guarantee bounds on delay, jitter, or loss in a packet-switched network. Such QoS guarantees are possible in circuit-switch networks.

2.4 Router Queuing Schemes

Queuing schemes are used to provide priority and more bandwidth to real-time traffic over non-realtime traffic. This section discusses the need for queuing mechanisms. The factors for delays in a packet-switched network are listed and each factor is discussed in detail. Also, different QoS mechanisms are illustrated in detail.

2.4.1 The Need for Queuing Mechanisms at Routers?

A router may handle traffic from many sources and of many types including both real-time and non-real-time services. By default, no priority mechanisms are in place at routers and, as a result, all packets are serviced on a first-in, first-out (FIFO) basis. A worst-case scenario for real-time applications occurs when both real-time and non-real time packets are destined for same output port of a router. For example, assume that a non-real-time packet of 1500 bytes enters the router between two real-time voice packets. Note that the maximum transmission unit (MTU) of an Ethernet frame is 1516 bytes and that the size of a voice packet is, typically, much smaller than a data packet. The second voice packet must wait until the non-real time packet is serviced (transmitted) by the router. The actual waiting time for the second voice packet depends on length of first non-real time data packet and the data rate of outgoing communication link. Also, consider the communication link to be a slow wide area network (WAN) link with a data rate of 64 Kbps. The problem is clear. The transmission delay is the time taken to transmit all of the packet's bits onto the link. In this case, the transmission delay for a 1500-byte packet on a 64-Kbps link is 187.5 ms. This is the amount of time that the second voice packet has to wait before being serviced by the router. Note that the end-to-end delay of a voice packet should not exceed 250 ms for good voice quality [Kea00]. Thus, QoS mechanisms are needed to be deployed to provide priority to packets from real-time applications in a packet-switched network.

When packets for a real-time application are carried in a packet-switched infrastructure that is also transporting packets for other applications, QoS mechanisms are needed to ensure voice quality, even if the amount of contention is fairly small. Packet networks introduce variable delays and packet loss for a variety of reasons, as discussed above. Therefore, network designers need to optimize the following metrics using QoS mechanisms: reliability, delay, jitter, and available bandwidth. In this research, we concentrate on minimizing delay, jitter, and improving throughput for a particular voice application, namely land mobile radio voice over IP. In any packet network, the major sources of delay are due to [Kea00]:

1. processing by the codec at both the sender and the receiver;
2. packet formation (packetization delay);
3. queuing at the output interface;
4. serialization (transmission delay) at the output interface;
5. queuing, serialization, and propagation delay in the network; and
6. time to fill the play-out buffer.

Codec processing delay is introduced when an analog signal is converted into a digital format and compressed form by a codec. Typically, a higher processing delay is associated with a lower bit rate codec to account for the higher degree of compression that is afforded by such codecs. Similarly, delay due to decoding occurs at the receiver when coded digital format is converted back to analog signal.

Packet formation delay, sometimes called packetization delay, occurs when codec frames are encapsulated into UDP/IP packets for transmission. When a single codec frame is transmitted in a packet, the packet formation delay is not significant. However, if multiple codec frames are grouped into a single packet, then the first frame of the group must wait while additional codec frames are generated to complete the packet.

After the codec frames are formed into packets and are ready for transmission, they might need to wait in an interface queue for output. Such delays, which can be substantial if the interface is heavily utilized, depend on the queuing technique being employed at the interface. Hence, to reduce the delay experienced by voice packets, the queuing policy should move voice packets to the front of the interface-queue.

Transmission delay, sometimes called serialization delay, is the time taken to put all of the packet's bits in the link. The amount of serialization delay depends on the length of the packet and the transmission rate of the interface. Hence, large packets or lower link data rates will increase serialization delay.

Queuing, transmission, and propagation delay occurs in the network, too. At each router in the backbone and at the network edge, packets experience queuing and transmission delay. Packets also experience propagation delay while traversing the links between routers.

Delay also occurs when a play-out buffer is filled before decoding begins. The function of the play-out buffer is to supply voice packets (after headers are removed) to the voice decoder in a smooth fashion so as to achieve a continuous audio stream. The buffer size is set so that it never overflows nor, ideally, remains completely empty. Jitter is a more serious concern for packet voice networks than absolute delay. Jitter is caused when packets experience unequal delays while transiting a network. Play-out buffers are employed to help reduce jitter in packet voice systems. Jitter is an inherent characteristic of a packet-switched network, because there is no end-to-end synchronous transmission as in circuit-switched networks. The primary source of jitter in packet-switched networks are queuing delays at the source host's interface and in the network.

There are certain techniques that can be employed to minimize bandwidth utilization. One approach is to optimize data throughput efficiency by prioritizing traffic, minimizing routing updates, and decreasing the amount of packet overhead. Using a voice codec with a large compression ratio obviously reduces bandwidth utilization. Also, monitoring network utilization trends allows network designers to plan bandwidth upgrades.

Having discussed QoS metrics such as delay, jitter, and bandwidth utilization, there are numerous QoS mechanisms for IP networks that can be used to optimize bandwidth and prioritize packet transmissions for real-time applications:

1) Resource Reservation Protocol (RSVP) [MMF01];
2) queuing policies [MMF01];
3) traffic policing and shaping [Kea01];
4) header compression [Kea01];
5) fragmentation and interleaving [Kea01];
6) dual FIFO transmit buffers [Kea01]; and

7) Mapping IP QoS requirements to Asynchronous Transfer Mode (ATM) service classes [Kea01].

This section discussed the need for QoS, various factors for delays in packet-switched networks, and listed different QoS mechanisms. The next section discusses four QoS schemes whose performance is investigated in this thesis. The final section of this chapter, discusses previous research done in LMRVoIP and its relation to this thesis.

2.4.2 Queuing Schemes

In this thesis, we investigate the performance of a LMRVoIP application under four different queuing schemes that are available in a commercial off-the-shelf router, specifically Cisco's Internetwork Operating System (IOS) [Cis07b]:

1) First-in first-out (FIFO);
2) Priority queuing (PQ);
3) Weighted fair queuing (WFQ); and
4) Class-based weighted fair queuing (CBWFQ).

A router's interface has a queue for holding packets awaiting transmission. Queuing schemes give a network administrator control over what happens to queued packets. Normally, efforts to improve QoS in a network start by optimizing interface-queuing policies. There are two core issues that are addressed by queuing policies. First, the policies should provide the QoS required for identified applications. Secondly, the policies should provide an equitable or fair distribution of bandwidth resources, within the constraints of meeting QoS requirements for specified applications. By addressing these two issues, QoS mechanisms must focus on managing delay and delay variation for selected applications and achieving overall fairness for other applications sharing the bandwidth. A queuing scheme has capabilities to manage queue depth and schedule the order of packet transmission to provide required QoS and ensure fairness.

A FIFO queue is a simple buffer that holds outbound packets in a single queue until the transmitting interface can send them. Packets are sent out of the interface in the order in which they are arrived in the buffer, as shown in Figure 2.1. This queuing scheme does not provide QoS for specified flows or equitable bandwidth distribution among flows sharing a link. During periods of congestion, buffers are filled and packets are dropped without regard for the packet type or associated application requirements. When FIFO is used, ill-behaved sources can consume almost all the bandwidth, bursty sources can cause delays for time-sensitive or important traffic, and packets from high-priority flows can be dropped because low-priority traffic fills the queue. The key advantage of FIFO queuing is that it requires the least amount of router processor and memory resources among all queuing schemes. However, the simplistic nature of FIFO queuing is also its key disadvantage.

PQ is an approach to give strict priority to designated traffic over other traffic. Packets arriving at an interface for transmission are separated into four queues, low, normal, medium, and high, as shown in Figure 2.2 [Cis06]. Packets are always serviced from the high priority queue first. If packets are waiting in the high priority queue, they will be sent to the transmission buffer. If the high priority queue is empty, then any packet in the medium priority queue is sent to the transmission buffer. If the higher priority queue and medium priority queue are empty, then packets from the normal queue are sent to the transmission buffer, and so on. High priority traffic incurs the least possible delay and jitter with this queuing technique, but there are no provisions for distributing bandwidth among traffic with equal priorities. In Cisco IOS, the default queue sizes are 80, 60, 40, and 20 packets for low, normal, medium, and high priority queues, respectively. The disadvantage of this scheme is that it could potentially force the packets in lower priority queues to remain un-serviced for a long time, i.e., lower priority traffic could experience "starvation."

WFQ is used as the default queuing mode on most serial interfaces configured to run at or below E1 speeds (2.048 Mbps) in Cisco IOS [Cis06]. Flows can be classified based on packet header fields such as source and destination IP addresses and TCP or UDP port number. WFQ creates a separate queue for each traffic flow and uses a queue depth for each flow as shown in Figure 2.3. In Cisco IOS, the default queue size used by WFQ is 64 packets. The objective of WFQ is to ensure that low bandwidth flows receive preferential treatment in gaining access to an interface, while permitting large bandwidth flows to use the remaining bandwidth in proportion to their weights. WFQ classifies traffic into flows or "conversations." A weight is applied to each class of flow. The weight determines how much bandwidth each flow is allowed relative to other flows. WFQ specifies weights using the IP Precedence value in the Type of Service (TOS) field in the IP header. As the precedence or weight increases, WFQ allocates more bandwidth to the flow during periods of congestion.

In CBWFQ, a separate custom queue, called a class is identified for each flow or conversation that needs to be policed or prioritized. Unclassified traffic is automatically placed in the default class. The maximum number of classes supported by CBWFQ is 64 [Kea01]. In WFQ, it is difficult to precisely specify the amount of bandwidth allocated to a flow and bandwidth for each flow is determined by the number of flows, which can constantly change. CBWFQ overcomes this limitation of WFQ by specifying the exact amount of bandwidth that is to be allocated for each class. CBWFQ also inherits proprieties of priority queuing by configuring a priority queue for the traffic class that is delay-sensitive. Each class has a default queue size of 64 packets.

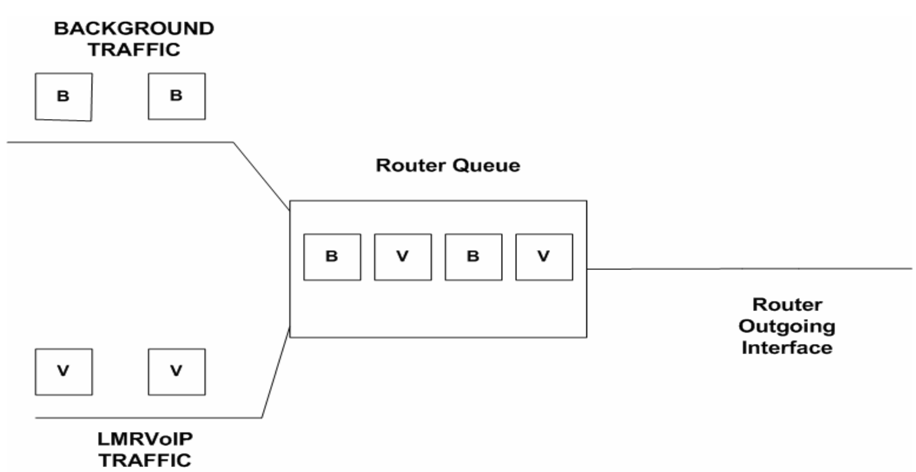

Figure 2.1. Illustration of first-in first-out queuing.

2.5 Prior Studies and Relation to this Thesis

There has been limited research in the area of packet-switched LMR voice traffic or LMRVoIP. Standardization bodies in North America and Europe have developed digital LMR communication systems based on open standards which are the Association of Public and Communication officials (APCO) Project 25 and the Terrestrial Trunked Radio (TETRA), respectively. The inter-system interface (ISI) is responsible for the internetworking of different TETRA and APCO networks.

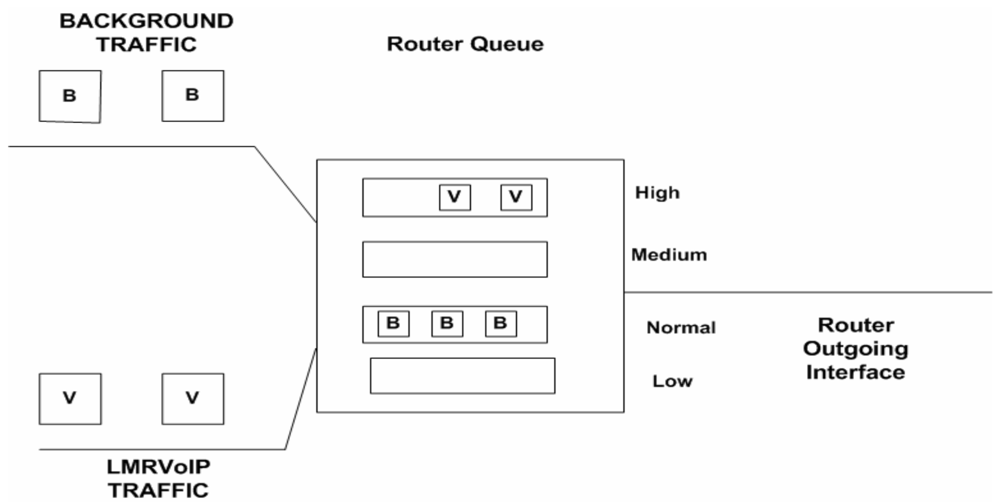

Figure 2.2. Illustration of priority queuing.

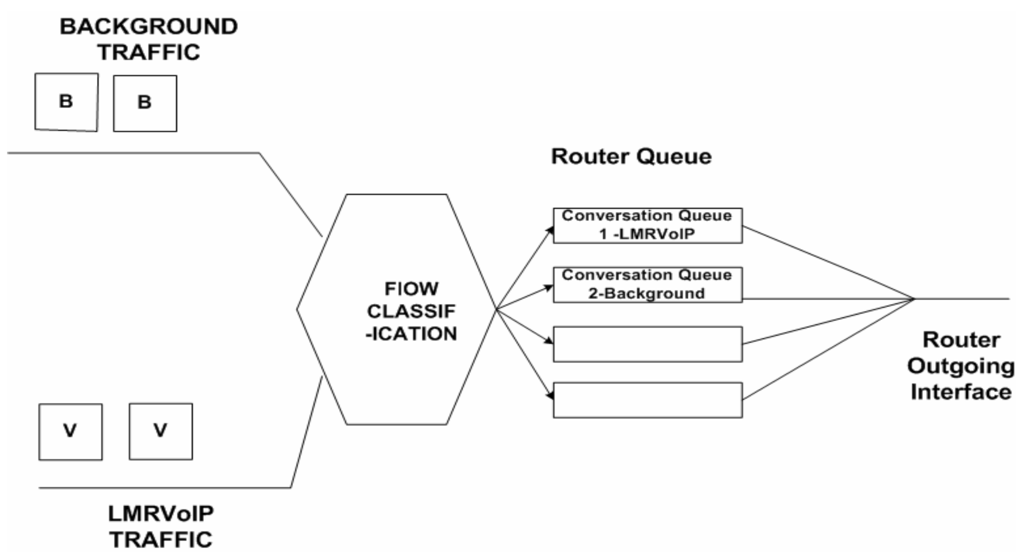

Figure 2.3. Illustration of weighted fair queuing.

Tsiakkouris [Tsi02] proposes a framework for an LMR packet-switched network to interconnect base stations across an extended coverage area. This thesis develops a simulation model to characterize the loading effects on the LMR network and also investigates the performance of a packet-switched ISI. Different underlying technologies, including IP, Asynchronous Transfer Mode (ATM), and Frame Relay (FR), are considered for the backhaul of voice packets across the LMR network. This thesis does not discuss the performance of an LMRVoIP application under different router queuing schemes and with the use of a Virtual Private Network (VPN) mechanism.

Mock and Miller [MT02] propose a solution for establishing a state-wide VoIP network, which consists of a single central server and clients located at participating departments. This solution was proposed for communication between emergency personnel from various departments using different radios operating at different frequencies. Preliminary testing of a prototype system was performed for basic audio functionality and for subjective voice quality under limited bandwidth conditions. This paper does not talk about VPN nor does it study the performance of an LMR packet-switched network under different router queuing schemes.

Dekeris, Adomkus, and Budnikas [DAB06] combine Weighted Fair Queuing (WFQ) and Low Latency Queueing (LLQ) to ensure Quality of Service for high priority bursty video conferencing traffic. This papers claims that Weighted Fair Queueing (WFQ) which is suitable for providing QoS requirements to all kinds of traffic on the network in most cases, but there are exceptions, when it can‘t assure high QoS requirements for the video conferencing traffic, which need high network resources, especially when the network load is high. Results show that loss of packets from WFQ can be eliminated when using WFQ combined with LLQ and also delay can be reduced two times. This research does'nt evaluate the performance of a LMRVoIP application using PQ, WFQ, or CBWFQ.

Georges, Divoux, and Ronddeau, [GDR05] show that the Weighted Fair Queuing balances the allocation of the network resources to the different traffics regarding the time contraints they have to respect. This paper shows WFQ overcomes the main drawback of the Strict Priority Queuing (SP) algorithm, which is, that it can lead to the impossibility for the lowest priority queues to be forwarded. The major contribution of this paper is to study the worst delays when switches are using Weighted Fair Queueing scheduling algorithm and to determine the minimum service that is necessary for the time-constrained frames, in order to ensure a better service to other traffics. Results show that WFQ is more fair than SP, since the access to the output is balanced based on predetermined weights and there is no famine possible for a traffic with poor priority. Moreover, since WFQ is more fair than SP for traffic with a poor priority, delays upper-bounds for traffic with higher priority will be longer, but other traffics will have shorter bounds. This paper does not study the performance of a time-sensitive LMRVoIP system under different queueing schemes.

Tiglao [Tig07] propose the use of Value Based Utility (VBU) to improve the Priority Queuing (PRIQ) mechanism, which has inherent problem of starvation in the lower priority classes. The proposed frame work VBU, models the perceived knowledge of the state and degree of user satisfaction in managing router resources and functions. This enhanced scheduler uses dynamic adaptation that can be used to improve the network performance in the delivery of real time traffic in a diffserv environment. This scheme alleviates the problem of starvation of static PRIQ by elevating the lowest priority packets to the higher priority queues using packet marking at the ingress node of the network. PRIQ-VBU provides lower average delay, lowest average jitter and lower packet loss compared to the standard PRIQ. This paper does'nt deal with performance of an LMRVoIP system with different queuing schemes for QoS and with the use of VPN

2.6 Summary

Three types of VoIP applications were discussed. The LMRVoIP application was discussed in detail. There is a discussion about LMRVoIP products from Catalyst, M/A-COM, and Motorola. Two types of switching technologies, circuit-switching and packet-switching, were discussed. Four types of QoS mechanisms, previous research, and its relation to this thesis were also discussed.

Chapter 3. Problem Statement and Methodology

This chapter discusses in detail the LMRVoIP problem. The specific research questions addressed in this thesis are outlined. The approach followed in this thesis is explained. A high-level overview of different router queuing configurations, VPN configurations, and configurations applied to an LMRVoIP system, a variation of the Catalyst IP Tone product, to generate continuous and ON-OFF voice traffic are discussed. Finally, this chapter concludes with a description of the test bed used for this research, including the test bed components, testing tools, and network configuration.

3.1 LMRVoIP Problem

As discussed in Sections 2.1 and 2.3, there are certain problems, such as jitter and bandwidth utilization, that need to be addressed when deploying LMRVoIP. This section outlines the specific research questions answered by this thesis, after investigating the performance of a LMRVoIP application under different router queuing schemes and with the use of VPN. The focus is on a particular LMRVoIP application, IP Tone from Catalyst Communications Technologies [Cat02b]. The two key research questions are listed below.

1) What is the effect of deploying different router queuing schemes at the intermediate routers on the performance of the LMRVoIP application in terms of jitter, maximum jitter, throughput, and signaling overhead with different forms of LMRVoIP traffic and different levels of background traffic competing with LMRVoIP traffic for network resources?

2) Does the LMRVoIP application perform well over a VPN?

3.2 Approach

As discussed in Section 2.2, the QoS metrics used to evaluate the performance of a real-time packet-switch application like LMRVoIP are delay, jitter, bandwidth, and reliability. Router queuing schemes can decrease jitter and delay and improve application-level throughput for a real-time application that is designated for high-priority treatment. This section provides a high-level overview of classification, packet-marking, and other related router configuration procedures. Details about the configuration using Cisco's IOS for the four queuing schemes, the specific tests conducted, and results obtained are provided in Chapter 4. In a FIFO queue, no classification of traffic and packet marking is done, as FIFO queuing treats all packets the same. In PQ, four levels of queues are supported. Since queues are serviced in the order of their priority, the delay-sensitive LMRVoIP traffic is placed in the highest priority queue. Once the entire PQ configuration is done, PQ has to be enabled on an interface. In WFQ, it is necessary to assign the maximum possible IP precedence value to packets carrying voice traffic, so that these packets are allocated highest level of bandwidth access among all the flows. A flow having the highest IP Precedence value will have the least weight, so the highest bandwidth will be allocated by WFQ to this flow. In WFQ, we cannot specify the required amount of bandwidth for voice traffic and, in most implementations, an IP Precedence value of 5 is configured for voice traffic. Once the required bandwidth is provided to voice traffic, WFQ will allocate bandwidth for other flows, including background traffic. WFQ can be enabled on an interface by using the `fair-queue` command in Cisco IOS. The first step in configuring CBWFQ is to create a class using the IOS `class-map` command for any traffic that needs to be policed. Then, properties of each class can be set. The final step is to enable the CBWFQ policy on an interface.

The VPN configuration includes a VPN server, a VPN client, an application remote or client, and an application server. A single PC acts both as an application server and a VPN client. The VPN server uses the VPN service included in the Windows 2000 Server operating system [Mic04]. In this configuration, Generic Routing Encapsulation (GRE) [Cis06b] and the Point-to-Point Tunneling Protocol (PPTP) [Mic98] are used for encryption.

Continuous and ON-OFF voice traffic are generated by a special version of the Catalyst IP Tone product by enabling and disabling the “Enable Auto TX Setting Below” option in the application. The Catalyst IP Tone software generates an approximately uniform distribution of ON and OFF times based on the minimum and maximum values specified.

3.3 Test Bed

This section discusses the test bed that was created to perform various experiments. Figure 3.1 illustrates the end-to-end LMRVoIP system for testing QoS queuing schemes.

3.3.1 Test Bed Components

The major system components included in the test system are described below.

- The client (Catalyst remote) and server (Catalyst server): The client and server hosts run the LMRVoIP application, in this case a variation of the Catalyst IP Tone application. The client generates voice traffic at a rate determined by the selected codec and sends it to the server.

- Iperf server and Iperf client: The Iperf client sends packets to the Iperf server to generate background traffic to compete with LMRVoIP traffic. The background traffic generated by the Iperf client is intended to emulate traffic from non-LMRVoIP sources, such as electronic mail, file transfer, and web traffic. The Iperf tool is discussed in Section 3.3.2.

- Three Cisco 2514 routers: Three Cisco 2514 routers constitute an IP “cloud” that represents an intermediate or backbone network. The routers are configured with different router queuing schemes, as discussed in Section 3.2 above and in Chapter 4.

- Control host: The control host is used to run Expect scripts and to access all routers and computers via the network. The Expect tool is discussed in Section 3.3.2.

3.3.2 Testing Tools

The various testing tools used are described below.

Iperf: Iperf is a free tool that can be used to generate TCP or UDP traffic of different rates and different packet sizes and to measure the performance of those traffic flows [Ipe05]. We use Iperf primarily as a traffic generation tool to create background traffic in a controlled manner. Iperf allows the tuning of various parameters, such as TCP window size and maximum transmission unit (MTU). Iperf is used to measure maximum TCP bandwidth and UDP characteristics. Iperf reports bandwidth, delay, jitter, and datagram loss.

Ethereal: Ethereal (now called "Wire Shark") is free network "sniffer" for Microsoft Windows or Linux that can be used to capture and analyze network traffic [Eth05]. A text-based version of this tool called Tethereal is used in this research. In this research, Tethereal is used to capture the traffic at the server. The captured traffic is used to determine performance results such as application throughput, delay, jitter, and signaling overhead..

TCL: Tool Command Language (TCL), supported by free tools, is a simple scripting language for controlling and extending applications [Ous94] [Wel03][Tcl04]. It provides generic programming facilities, such as variables, loops, and procedures. We primarily used TCL to analyze the trace files created by Tethereal to determine performance results.

Expect: Expect is a tool for automating interactive applications such as telnet, ftp, and rlogin [Exp05]. Expect is also useful for testing these same applications. In this research, Expect scripts have been developed to configure the network for experiments. The expect scripts run on the control host.

3.3.3 Test Bed Configuration

Figure 3.1 shows the test bed used for testing the queuing mechanisms. It consists of three routers – Router One, Router Two, and Router Three – that are connected by 38,400 bps serial links. Router Three has two inputs from 10-Mbps Ethernet local area networks (LANs). One LAN contains an LMRVoIP client (Catalyst remote) running the Catalyst IP Tone application. The client, using a special version of IP Tone, generates traffic at a rate determined by the voice codec and, also, can generate traffic according to an ON/OFF pattern that can be specified via control settings in the special version of the IP Tone application. The other LAN contains an Iperf client, which, together with an Iperf server, allows controlled generation of background traffic to compete with LMRVoIP traffic for network resources. Traffic flows from the Catalyst remote and the Iperf client move through Router Three, to Router Two, and on to Router One. Beyond Router One, the traffic is delivered to the 10-Mbps Ethernet LAN containing the LMRVoIP server (Catalyst server), Iperf server, and control host. Traffic is captured using Tethereal on this LAN. The captured traffic is analyzed after an experiment using a TCL script to determine performance results including application throughput, delay, jitter, and signaling overhead.

The intermediate or backbone network is constructed from three Cisco 2514 routers. Router Three connects to Router Two via a 38,400-bps serial connection. Router Two connects to Router One via a second 38,400-bps serial connection. While 38,400 bps is a low data rate, it is advantageous as it allows us to easily create controlled congestion conditions. And, some networks do use such low-capacity links.

3.4 Summary

The LMRVoIP problem was discussed in detail. The two key research questions addressed in this thesis were outlined. A high-level overview of different router configurations was given. Details about different router configurations, experiments performed, and results obtained are provided in Chapter 4. The testbed created to perform various experiments was discussed.

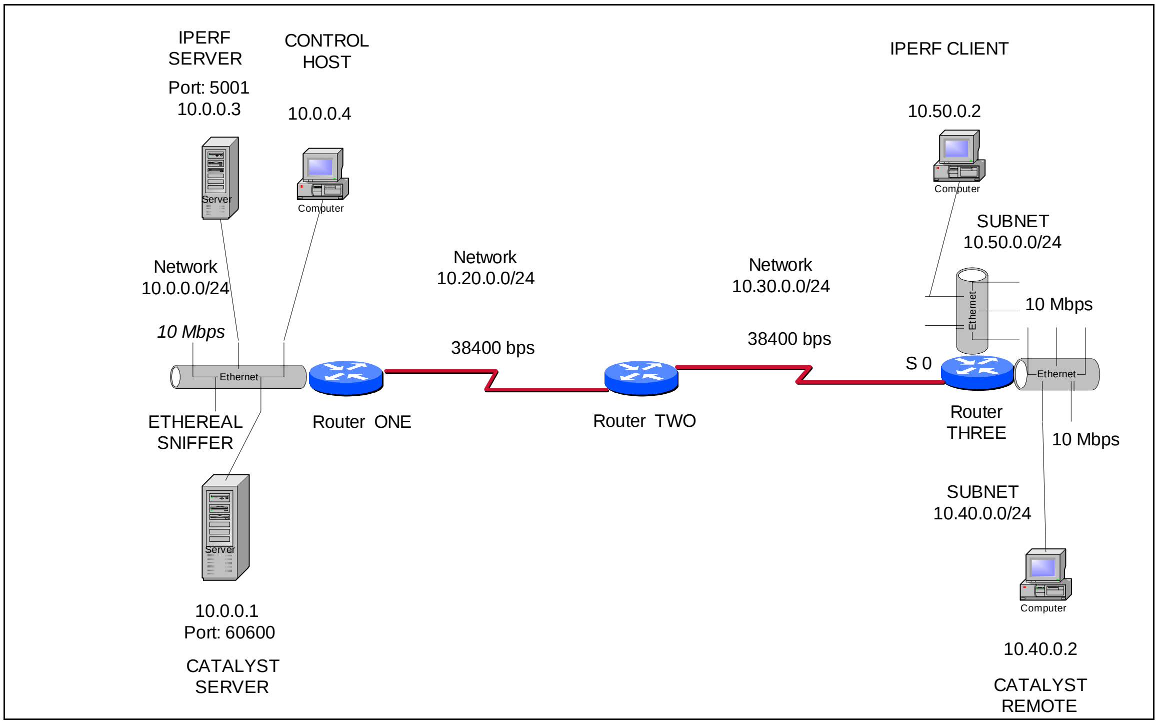

Figure 3.1. Network configuration for testing queuing mechanisms.

Chapter 4. Experiments and Results

This chapter discusses in detail the key assumptions made in this research, the two types of traffic generation patterns used to test QoS schemes, the test scripts developed to perform various experiments, and the test procedures followed to evaluate four different QoS schemes. There are separate sections for first-in, first-out (FIFO), priority queuing (PQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ) queuing schemes, which explain configuration procedures for Cisco IOS and the associated code. These sections also discuss performance of the voice application in terms of four performance metrics. Finally, this chapter concludes with a discussion of the investigation of an LMRVoIP application over a virtual private network (VPN).

4.1 Key Assumptions

The following assumptions were made in this research.

Codec: The Catalyst gateway application and Catalyst remote application were configured to use the GSM codec [Sch03]. It generates a 13-kbp/s constant bit stream in compressed form. By considering one codec, we were able to improve accuracy by conducting more experiments with the same parameter sets. Experiments done using other codecs would likely lead to similar results. The other codecs may have higher or lower bitrate and the baseline test would give us a reference value of application throughput, which can be used to compare results from different queuing schemes.

Size of background packets: The packet size of a background packet is set to same size as that of a voice packet. The packet size of a GSM application payload is 342 bytes and its size after including UDP, IP, and link layer headers is 384 bytes. In a network, the packet size of a background packet depends on the nature of the application generating a particular packet and the maximum transmission unit (MTU) of the link. Since, in this research, the objective is to compare the four different queuing schemes, the packet size of a background packet is set to 384 bytes (including all of the headers) using Iperf in all experiments.

Static routing: All of the Cisco routers are configured with static routing. Static routing can be achieved by the use of static routes. The Cisco IOS *`IP Route`* command is used to configure static routes. An alternative is to run a dynamic routing protocol, such as Open Shortest Path First (OSPF) [Tho03]. Experiments conducted using static routing or dynamic routing protocols in a stable network with no topology changes will likely lead to comparable or similar results. After the routing tables of the routers stabilize, it should'nt affect the performance of the LMRVoIP application, as LMRVoIP application operates in layer 4 and layer 5 of the TCP/IP model, whereas routing protocols operate at layer 3 of the TCP/IP model.

4.2 Investigation of the Performance of Queuing Schemes

This section discusses about two types of traffic patterns used to perform various tests, test scripts developed to automate the configuration of the network and to analyze the data captured by Tethereal, and test procedures followed.

4.2.1 Traffic Generation

The variation of the Catalyst IP Tone application is configured to produce each of the following two patterns.

Continuous voice traffic: Continuous voice traffic is generated at the LMRVoIP client (Catalyst remote) connected to Router THREE and sent to the LMRVoIP server (Catalyst server), which is connected to Router ONE, as shown in Figure 3.1.

ON-OFF voice traffic: Four parameters that are used to specify ON-OFF voice traffic in the Catalyst IP Tone application: maximum ON time, minimum ON time, maximum OFF time, and minimum OFF time. For our testing, all these four parameters are set in such a way they are somewhat representative of push-to-talk voice traffic as captured from live use in an operational LMR system. The parameters set are maximum ON time = 18 s, minimum ON time = 1 s, maximum OFF time = 75 s, and minimum OFF time = 1 s. These parameters are set in the special variation of the Catalyst IP Tone product that is running at the LMRVoIP client connected to Router THREE. The LMRVoIP client generates ON-OFF voice traffic according to the specified parameters.

4.2.2 Expect and TCL Test Scripts

Test scripts for the Expect tool have been developed to configure the network for experiments. The Expect scripts run on the control host as shown in Figure 3.1, which can access all routers and computers via the network. Note that this configuration traffic is carried over the network prior to beginning the actual test and, thus, does not contribute to the background traffic in the network. The expect scripts perform the following functions.

1) Configure Cisco router configurations through Cisco's IOS. Expect provides IOS commands interactively in response to prompts from the router. Different Expect scripts configure different settings on the routers during various tests, such as changing the QoS queuing scheme and configuring associated parameters.

2) Configure and start the Iperf client and server to generate about 38 kbps of UDP application traffic, which emulates background traffic.

TCL scripts have also been developed to analyze data captured by Tethereal. The scripts determine voice throughput, average jitter, maximum jitter, and signaling overhead. The number of TCP packets used by the LMRVoIP application for signaling is a measure of signaling overhead. The scripts are documented in more detail in the appendix.

The status of an ongoing test can be determined by examining information on individual routers. This can be done interactively at the control host.

4.2.3 Test Procedure

For each of the four QoS queuing schemes, tests, as described below, were conducted using both continuous and ON-OFF voice traffic.

4.2.3.1 Continuous Voice Traffic Case

Each of the following tests was run for 10 minutes using continuous voice traffic.

1) **Baseline testing:** In baseline tests, only LMRVoIP traffic is sent and no other traffic is sent. These tests are performed to get reference parameters under ideal conditions when there is no competition for network resources from background traffic. The reference parameters obtained from these tests are used to compare results with different QoS queuing schemes with background traffic. Traffic from the LMRVoIP application is generated and transmitted from the LMRVoIP client to the LMRVoIP server and no other traffic is carried in the network. These tests are carried out with FIFO configuration, but all configurations will lead to similar results since there is no competing background traffic.

2) **Testing beyond saturation:** To ensure link saturation, all tests were carried with the total rate of offered traffic being about 150% of the serial link's capacity. This loading is achieved by generating UDP background traffic at the rate of about 38 Kbps as measured at the application layer. This results in about 43 Kbps of background traffic after UDP and IP headers are considered. There is about 16 Kbps of voice traffic generated by the GSM codec, after including IP and UDP or TCP transport layer headers. The four queuing schemes described in Section 2.4 were tested with this load. The queuing scheme, FIFO, PQ, WFQ, or CBWFQ, is configured on all router interfaces. WFQ is tested with one and two different background traffic flows.

4.2.3.2 ON-OFF Voice Traffic Case

To represent the range of values observed in push-to-talk traffic as captured from a live LMR operation, the ON duration was chosen to vary from 1 to 18 seconds and the OFF duration was chosen to vary from 1 to 75 seconds. The ON and OFF duration is configured on the LMRVoIP client running a special build of the Catalyst IP Tone application. Each of the five tests performed for continuous traffic case, namely FIFO queuing, PQ, WFQ, and CBWFQ, is repeated using ON-OFF traffic instead of continuous traffic. Background traffic is generated by Iperf at 38 Kbps at the application layer. All tests were performed over a longer period to allow router queues to stabilize, but the data is collected for 10 minutes after the initial "warm up" period.

An additional test with ON-OFF traffic was performed for CBWFQ. This test was run for 20 minutes to determine if variability in results for the different trials of CBWFQ is due to variability in the amount of traffic generated by the special version of Catalyst IP Tone, which might be present in a 10-minute test.

4.3 Configuration and Results for First-In-First-Out (FIFO) Queuing

This section provides configuration details for the FIFO queuing scheme and results obtained from the experiments.

4.3.1 First-In-First-Out Configuration

On high bandwidth interfaces, the routers enable FIFO queuing by default. On interfaces with less than 2 Mbps of bandwidth, namely the serial interfaces, executing the following Cisco IOS command on all three routers enables FIFO.

```
Interface serial 0
No fair-queue
```

By default in a Cisco router, the outbound FIFO queue can hold 40 packets and the inbound FIFO queue can hold 75 packets.

4.3.2 Results for Baseline Case (Only LMRVoIP Traffic)

To ensure that results are representative, three trials are run for each experiment. Results are provided for each trial and aggregate statistics for all three trials are also presented. Table 4-1 shows the legend used to indicate the parameters measured in the tests.

Table 4-1. Legend for Labels used in the Tables

Parameters	Legend	Units
UDP Throughput (voice)	UTP	kbps
Average Jitter	AJ	ms
Maximum Jitter	MJ	ms
Number of Packets (Signaling Overhead)	N	packets
Time Taken	TM	ms
Mean	M	
Standard Deviation	SD	
95% Confidence Interval (Lower Value)	M-X	
95% Confidence Interval (Upper Value)	M+X	
Maximum Value	Max	
Trials 1, 2, and 3	T1, T2, and T3	

Results from baseline tests carried out with FIFO configuration are discussed in this section. Results from these tests in which only LMRVoIP traffic is sent and no other traffic is sent are given in Table 4-2. The baseline test provides values to compare to the results using different queuing schemes. For the case of continuous traffic, the mean throughput is 15.4 Kbps, which includes IP and TCP or UDP transport layer headers. For the case of ON-OFF traffic, the mean throughput is 3.6 Kbps because LMRVoIP traffic is not transmitted continuously. The average jitter and the number of TCP packets used by the LMRVoIP application for signaling are shown in the table.

Table 4-2. Results for Baseline Case with only LMRVoIP Traffic

	Continuous Traffic					ON-OFF Traffic			
	UTP	AJ	MJ	N		UTP	AJ	MJ	N
T1	15.394	5.399	187	13		3.494	5.642	212	115
T2	15.261	5.407	187	11		3.293	6.135	187	98
T3	15.599	5.471	187	11		3.978	6.263	434	120
M	15.418	5.425667	187	11.666		3.588	6.0133	277.667	111
SD	0.170	0.039	0	1.154		0.352	0.3279	135.965	11.532
Max			187					434	
M-X	15.225	5.38		10		3.19	5.639		98
M+X	15.61	5.469		13		3.98	6.38		24

The maximum jitter between any two successive voice packets was found for the three trials performed and maximum of the maximum jitter values among all three trials is 187 ms for continuous voice traffic case and 434 ms for ON-OFF voice traffic case. The 95% confidence intervals for throughput, average jitter, and number of TCP packets are shown.

4.3.3 Results for First-In-First-Out Queuing

With FIFO queuing, the buffer fills due to congestion and there is an equal chance that packets from the voice traffic flow and the background traffic flow will be dropped. FIFO queuing is not desirable as a standalone method to manage interface congestion. The results from tests using FIFO queuing are shown in Table 4-3. The problems with FIFO queuing are discussed in Section 2.4.

Table 4-3. Results for FIFO Queuing

	Continuous Traffic					ON-OFF Traffic			
	UTP	AJ	MJ	N		UTP	AJ	MJ	N
T1	6.869	228	985	11		0.847	223	837	38
T2	6.747	230	985	15		0.743	204	664	26
T3	7.288	233	985	10		1.521	214	669	28
M	6.968	230.333	985	12		0.795	213.5	750.5	32
SD	0.2838	2.516	0	2.645		0.0735	13.435	122.329	8.485
Max			985					837	
M-X	6.648	227.487		9		0.56	198.304		24
M+X	7.288	233.179		15		1.514	228.7		38

As seen in Table 4-3 for the case of continuous traffic, the mean throughput of voice traffic is 6.9 Kbps, which indicates a loss of more than 50% of voice packets when compared to the baseline value of 15.4 Kbps. The same conclusion can also be drawn for the case of ON-OFF traffic where the mean voice packet throughput is 0.8 Kbps, compared to 3.6 Kbps in the baseline case, due to loss of voice packets. The mean jitter increased to 230 ms and 213 ms from baseline values for continuous traffic and ON-OFF traffic cases, respectively, because voice traffic was not given priority over background traffic. The maximum jitter is 985 ms and 837 ms for continuous traffic and ON-OFF traffic cases, respectively, which would cause noticeable gaps in the received audio at the server resulting in poor voice quality. Graphs for throughput, average jitter, maximum jitter, and number of TCP packets are shown in Section 4.7.

4.4 Configuration and Results for Priority Queuing (PQ)

This section provides configuration details for the PQ scheme and results obtained from the experiments.

4.4.1 Priority Queuing Configuration

The following steps are required to enable Priority Queuing on a router interface.

1) Identify traffic types for each of the four priority queues.
2) Assign a maximum queue depth to each of the priority queues.
3) Assign the priority queues to an interface.

Traffic is assigned to one of four priority queues which are high, medium, normal, and low, based on network protocol, packet size, originating interface, and access-list that identifies specific addresses or higher-layer protocols. Here, the Cisco IOS `priority-list` command is used to match LMRVoIP signaling and application packets with specific TCP and UDP port numbers and place the identified packets in the high priority queue. The default values for queue depth are used, which are 20 packets for the high priority queue, 40 packets for the medium priority queue, 60 packets for the normal priority queue, and 80 packets for the low priority queue. Since packets in lower priority queues must wait more often, the lower-priority queues accommodate more packets by default. Finally, the `priority-group` command is used to enable priority queuing on an interface. Configuration of priority queuing for the serial interfaces of all the routers is performed as shown below using Cisco IOS. To configure the clock rate for the hardware connections on serial interfaces such as network interface modules (NIMs) and interface processors to an acceptable bit rate, Cisco IOS `clock rate` interface configuration command is used [Cis07c].

```
priority-list 2 protocol ip high tcp 8896
priority-list 2 protocol ip high udp 60600
interface Serial0
ip address 10.20.0.2 255.255.255.0
priority-group 2
clockrate 38400
```

4.4.2 Results for Priority Queuing

In this configuration, LMRVoIP traffic is placed in the high priority queue and background traffic is placed in the normal queue. As a result, LMRVoIP traffic is given preferential treatment over background traffic. LMRVoIP traffic benefits greatly from this configuration as is evident from the results in Table 4-4. Throughput for the continuous traffic case was 14.4 Kbps, which is nearly equal to the baseline value and the throughput for the ON-OFF traffic case also improved from the value obtained for FIFO queuing. Also, LMRVoIP traffic incurs the least possible latency and jitter with PQ as it is placed in the high queue. For example, the mean jitter for the continuous traffic case and ON-OFF traffic is 40.2 ms and 39.042 ms, respectively, a substantial improvement over the FIFO configuration. The maximum jitter for the ON-OFF traffic case is 847ms, which is greater than the 837 ms observed for the FIFO configuration. This may be due to a series of other packets, such as Address Resolution Protocol (ARP) [KR03] packets and Cisco Discovery Protocol (CDP) [Huc03] packets, which may be present between two successive voice packets resulting in high jitter. Thus, maximum jitter tells us how large jitter may be and should not be individually used to judge the overall performance. Graphs for throughput, average jitter, maximum jitter, and number of TCP packets are shown in Section 4.7.

Table 4-4. Results for PQ

	Continuous Traffic					ON-OFF Traffic			
	UTP	AJ	MJ	N		UTP	AJ	MJ	N
T1	14.368	40.21	186	11		2.318	37.22	187	97
T2	14.57	40.22	187	11		2.608	39.518	514	96
T3	14.421	40.24	187	12		2.702	40.39	847	110
M	14.453	40.223	186.666	11.333		2.542	39.042	516	101
SD	0.104733	0.0152	0.577	0.577		0.200	1.637	330.004	7.810
Max			187					847	
M-X	14.335	40.205		10		2.316	37.19		93
M+X	14.57	40.24		13		2.769	40.894		109

4.5 Configuration and Results for Weighted Fair Queuing (WFQ)

This section provides configuration details for the WFQ scheme and results obtained from the experiments.

4.5.1 Weighted Fair Queuing Configuration

The Cisco IOS `fair-queue` command is used to enable fair queuing on serial interfaces of all three routers as shown below.

```
interface Serial0
ip address 10.20.0.2 255.255.255.0
fair-queue
clockrate 38400
```

LMRVoIP traffic needs to be assigned the maximum allowed IP precedence value of 5 so that WFQ will allocate LMRVoIP traffic a greater proportion of available bandwidth than background traffic. The "weighted" part of WFQ comes into play when used with the IP Precedence field. Note that this marking is only done on the ingress router, which is Router THREE for the test configuration, as shown in Figure 3.1. The `route-map` feature supported by Cisco IOS is used for marking packets received by interface Ethernet0 at Router THREE, which is connected to the LMRVoIP client (Catalyst remote). The configuration performed on Router THREE using Cisco IOS is shown below.

```
route-map setprecedence permit 10
match ip address 101
set ip precedence critical
interface Ethernet0
ip address 10.50.0.1 255.255.255.0
ip policy route-map setprecedence
```

Between flows of different priority levels, WFQ allocates bandwidth based on weights associated with the flows. The following formula is used by WFQ to assign weights to flows with different IP Precedence bit values [Kea00]:

$$\textit{Weight} = 4096 / (1 + \textit{IP-Precedence}) \tag{4.1}$$

When there are two flows in the network (the LMRVoIP flow and a single background traffic flow), the bandwidth allocated by WFQ is given by Table 4-5.

Table 4-5. WFQ Bandwidth Allocation to Different Flows

Flow	IP Precedence	Weight	Ratio	Bandwidth%
LMRVoIP	5	683	6	85.7%
Background	0	4096	1	14.3%

As indicated in Table 4.5, LMRVoIP traffic is allocated a bandwidth percentage of 85.7, which accounts for 32.9 Kbps out of 38.4 Kbps serial link. Typically, voice flow including LMRVoIP flow will consume less bandwidth than background traffic. Once the LMRVoIP flow consumes its required bandwidth, WFQ will start allocating bandwidth for the background traffic flow.

4.5.2 Results for Weighted Fair Queuing

The results for the tests with WFQ are shown in Table 4-6. LMRVoIP packets are marked with an IP precedence value of 5, while background packets have default IP precedence value of 0. The greater allocation of capacity to LMRVoIP traffic is evident from the results. For example, the mean throughput for the continuous traffic case is 14.5 Kbps, which is near the baseline value. The mean jitter is 40.76 ms, a huge improvement over the FIFO configuration. We can also see improvement in throughput for the ON-OFF traffic case compared to results for FIFO queuing. Throughput for the ON-OFF case is 2.9 Kbps and the mean jitter is 39.65 ms. Graphs for throughput, average jitter, maximum jitter, and number of TCP packets are shown in Section 4.7.

Table 4-6. Results for WFQ (Single Background Traffic Flow)

	Continuous Traffic					ON-OFF Traffic			
	UTP	AJ	MJ	N		UTP	AJ	MJ	N
T1	14.527	40.82	197	13		3.034	38.6	197	88
T2	14.499	40.57	197	13		2.895	40.82	707	93
T3	14.721	40.89	197	11		2.702	39.53	364	90
M	14.582	40.76	197	12.333		2.877	39.65	422.666	90.333
SD	0.120	0.168	0	1.1547		0.166	1.114	260.012	2.516
Max			197					707	
M-X	14.445	40.57		11		2.686	38.39		88
M+X	14.715	40.95		14		3.065	40.76		93

The performance with WFQ is affected by the number of competing flows, not just the amount of competing traffic. To observe the effect of additional background traffic flows on LMRVoIP application performance, the background traffic was divided among two flows, i.e., the total amount of background traffic was kept constant but it was sent as two separate flows. Results for this test are shown in Table 4-7. As seen in the table, increasing the number of background traffic flows adversely affects the performance of the LMRVoIP application traffic. The most significant difference is the increase in average jitter. For example, the average jitter for continuous traffic with one competing flow is about 41 ms, while the average jitter with two competing flows is about 145 ms. Jitter is increased because voice packets may have to wait for the additional flow to be serviced. Similar results are also obtained for ON-OFF traffic case, in which average jitter is about 40 ms and 120 ms for one and two competing flows, respectively.

Table 4-7. Results for WFQ (Two Background Traffic Flows)

	Continuous Traffic					ON-OFF Traffic			
	UTP	AJ	MJ	N		UTP	AJ	MJ	N
T1	13.472	144.82	638	12		2.929	128	309	99
T2	13.697	144.98	480	12		2.786	115	309	91
T3	13.711	145.1	480	12		2.8	116	309	96
M	13.62667	144.9667	532.6667	12		2.838333	119.6667	309	95.33333
SD	0.134128	0.140475	91.22134	0		0.078831	7.234178	0	4.041452
Max			638					309	
M-X	13.475	144.808		12		2.749	111.48		91
M+X	13.77	145.154		12		2.927	127.847		100
X	0.151699	0.158878	103.1713	0		0.089158	8.181855	0	4.570882

4.6 Configuration and Results for Class Based Weighted Fair Queuing (CBWFQ)

This section provides configuration details for the CBWFQ scheme and results obtained from the experiments.

4.6.1 Configuration for Class Based Weighted Fair Queuing

Three steps are needed to configure router interfaces to use CBWFQ:

1) Sort traffic into classes.
2) Apply policies to classes.
3) Assign a service policy to an interface.

For any traffic that needs to be policed with certain QoS parameters, a separate class has to be created with the Cisco IOS `class-map` command. Class "voice" and class "default" are created to correspond to LMRVoIP traffic and background traffic, respectively. We can assign traffic to a class using the Cisco IOS `access-group` command. The access-group and access-list features offered by Cisco IOS are used to match LMRVoIP traffic. Class voice is configured as a priority queue using the IOS `priority` command and bandwidth of 17 kbps is allocated. Traffic which is not matched for QoS, i.e., background traffic in this case, is matched with class default. The default class can be made to operate in WFQ mode by the `fair-queue` command. CBWFQ is enabled on serial interfaces using the `service-policy output catalyst` command. The procedure to configure CBWFQ, which is done on all three routers, is given below.

```
class-map match-all voice
  match access-group 101
policy-map catalyst
  class voice
    priority 17
  class class-default
   fair-queue

interface Ethernet0
ip address 10.50.0.1 255.255.255.0
service-policy output catalyst
```

4.6.2 Results for Class Based Weighted Fair Queuing

Results from the CBWFQ test are shown in Table 4-8. With CBWFQ, LMRVoIP traffic is placed in the "voice" class, allocated a bandwidth of 17 Kbps, and configured as a priority queue. As expected, results show substantial improvement over those for FIFO queuing.

Table 4-8. Results for CBWFQ

	Continuous Traffic					ON-OFF Traffic			
	UTP	AJ	MJ	N		UTP	AJ	MJ	N
T1	14.427	40.56	197	11		2.688	41.71	466	82
T2	14.481	40.54	197	11		2.756	40.5	540	99
T3	14.789	40.59	197	13		2.46	38.758	197	116
M	14.565	40.563	197	11.666		2.634	40.322	401	99
SD	0.1952	0.0251	0	1.154		0.155	1.483	180.502	17
Max			197					540	
M-X	14.345	40.534		10		2.459	38.379		81
M+X	14.785	40.591		13		2.809	42.265		118

The throughput for the continuous traffic case is 14.4 Kbps, which is a significant improvement over results for FIFO queuing. Similarly, the mean jitter is 40.5 ms, a huge improvement over mean jitter for FIFO queuing. Results also indicate improvement in throughput and average jitter for the ON-OFF traffic case. While the router is providing background traffic with its proportion of bandwidth, there is a chance that the buffer maintained for LMRVoIP traffic may overflow and, hence, LMRVoIP packets may be lost.

Given the variability seen in results for CBWFQ, especially for the number of TCP packets (*N*), an additional test was run. Three additional trials were run for 20 minutes each. Results are shown in Table 4-9. The total ON time and the total number of calls were fairly consistent across all three trials, but there was still variability in the number of TCP packets. The results tend to indicate that the variability in TCP traffic is due to the normal variability in TCP's operation and not the difference in offered load.

Table 4-9. ON-OFF Traffic Results for CBWFQ with 20-minute Trials

	Total ON Time	Number of Calls		ON OFF Traffic			
				UTP	AJ	MJ	N
T1	24	274		2.87	41.08	286	170
T2	24	280		2.945	41.84	224	181
T3	24	275		2.931	41.86	207	181
M				2.915333	41.59333	239	177.3333
SD				0.039879	0.444672	41.58125	6.350853
Max						286	
M-X				2.862	41.011		169
M+X				2.967	42.172		186

Based on the observations and analysis after running a few tests for long periods, we can conclude that this variability is likely due to the manner in which the sockets and associated timers are implemented. However, the variability in the number of TCP packets and the extra capacity consumed by TCP packets due to variability is not excessive and the fact that the number of TCP packets varies is not really a performance concern. Therefore, the number of TCP packets should not be individually used to determine the performance of a LMRVoIP application.

4.7 Graphs Showing Performance Metrics Versus Queuing Schemes

Graphs for throughput, average jitter, maximum jitter, and signaling overhead for different queuing schemes for both continuous and ON-OFF traffic cases are show in Figure 4.1 through Figure 4.8. The error bar for each value indicates the 95% confidence interval.

4.8 Investigation of LMRVoIP with Virtual Private Networks

The performance of the LMRVoIP application was investigated with the use of VPN taking into consideration packet encryption, packet decryption, and latency introduced by extra packet processing at the LMRVoIP client

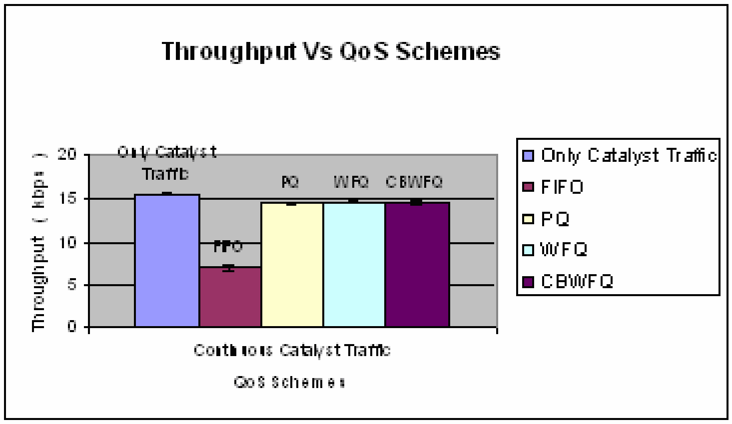

Figure 4.1. Throughput for continuous LMRVoIP traffic.

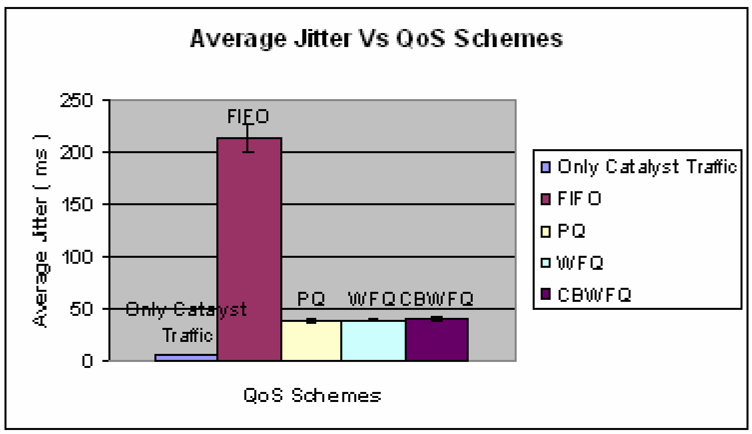

Figure 4.2. Average jitter for continuous LMRVoIP traffic.

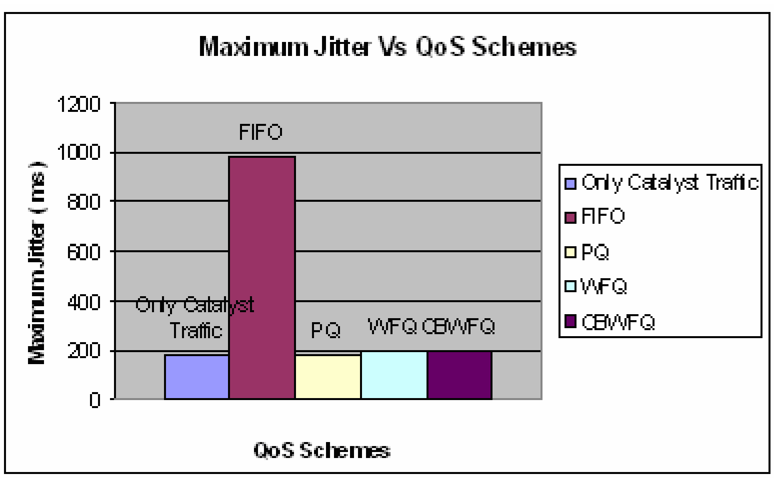

Figure 4.3. Maximum jitter for continuous LMRVoIP traffic.

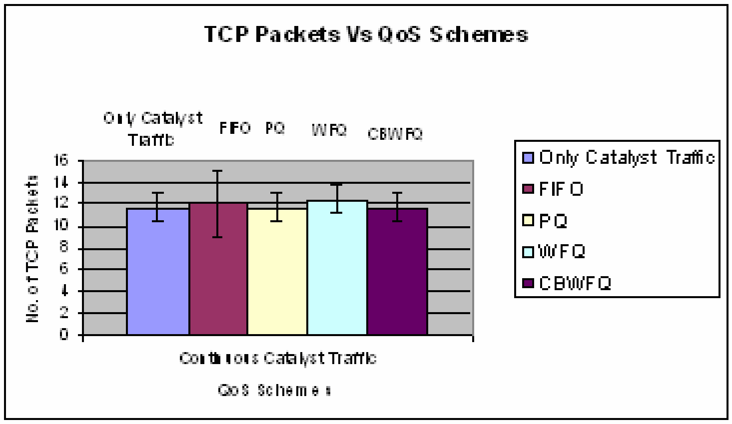

Figure 4.4. Signaling overhead for continuous LMRVoIP traffic.

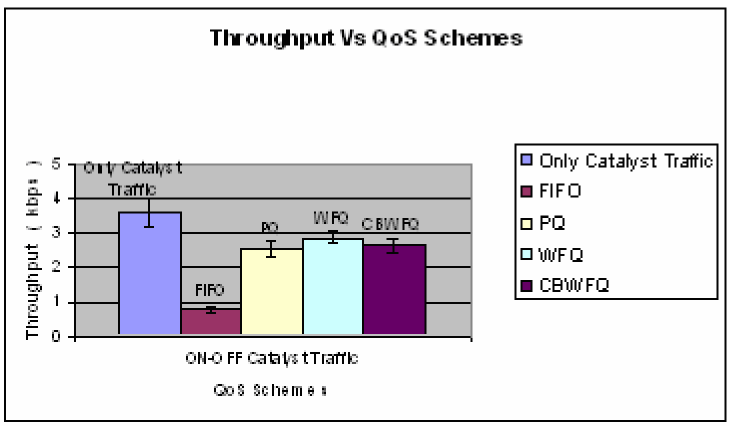

Figure 4.5. Throughput for ON-OFF LMRVoIP traffic.

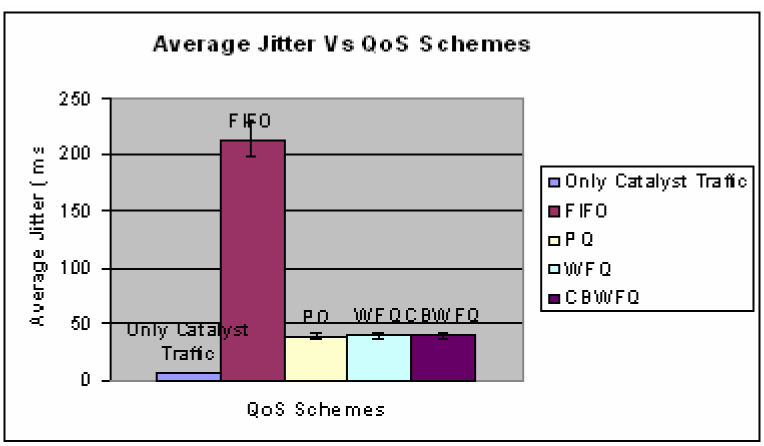

Figure 4.6. Average jitter for ON-OFF LMRVoIP traffic.

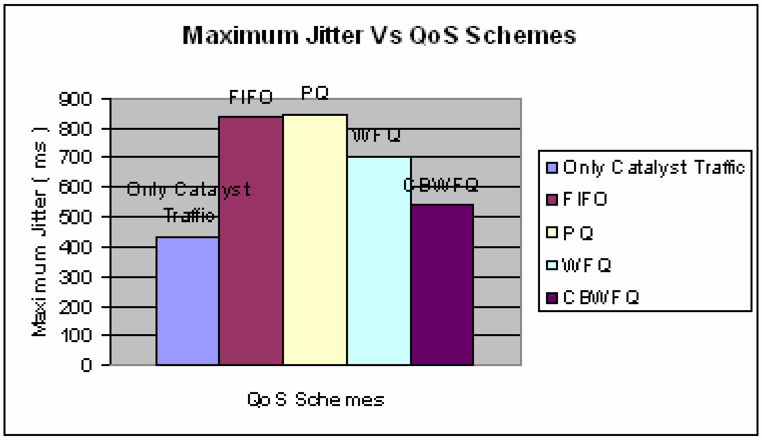

Figure 4.7. Maximum jitter for ON-OFF LMRVoIP traffic.

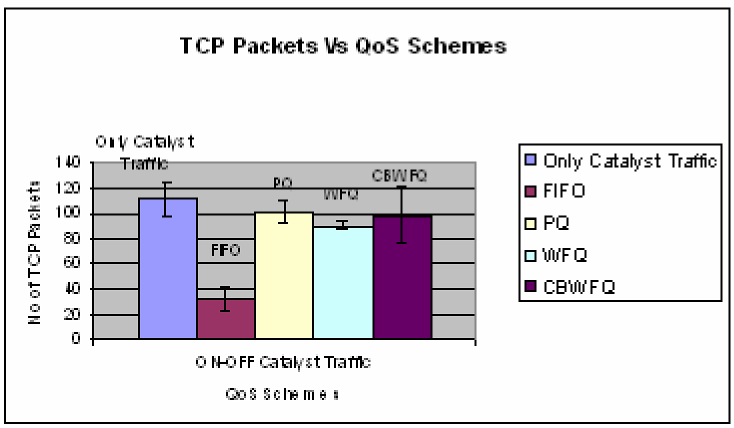

Figure 4.8. Signaling overhead for ON-OFF LMRVoIP traffic.

4.8.1 Network Configuration for VPN Testing

Figure 4.9 shows the network configuration used for VPN testing. The LMRVoIP client is on a remote network. It establishes a VPN connection to a VPN server. The VPN server for used in the testing used the VPN service in the Windows 2000 Server operating system. In an operational scenario, the VPN server would be inside the protected network or at the border of the protected network and the public network. The VPN can then send unencrypted packets to the LMRVoIP server. In an operational scenario, the LMRVoIP server would be inside the protected network.

4.8.2 Observed Results for VPN Testing

The packet flow from the LMRVoIP client (Catalyst remote) to the LMRVoIP server (Catalyst server) was observed to be as follows. Packets from the LMRVoIP server to the LMRVoIP client take the reverse path.

Packets from the LMRVoIP client (the source IP address is 10.40.0.2) are sent to the VPN server at its public interface (the destination IP address is 10.60.0.4). The packets are encapsulated and encrypted. In our particular configuration General Record Encapsulation (GRE) and the Point-to-Point Tunneling Protocol (PPTP) are used. Figure 4.10 shows packets captured between the LMRVoIP client and the VPN server. Ethereal was used to capture the packets. Packets from the LMRVoIP client to the VPN server are marked as being PPTP compressed packets and are encrypted. Packets from the VPN server to the LMRVoIP client are marked as being GRE packets and are also encrypted.

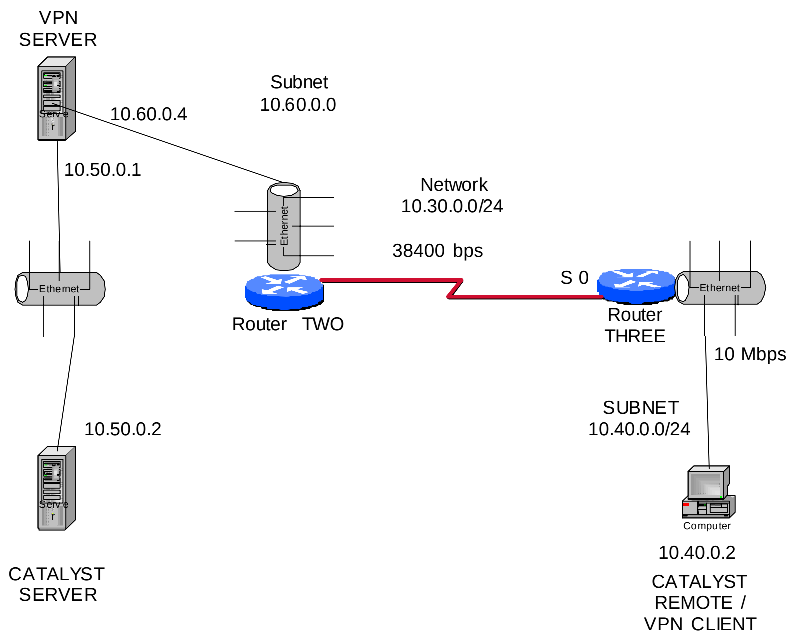

Figure 4.9. Network configuration for VPN testing.

The VPN server decrypts packets destined for the LMRVoIP server and forwards the packets to the LMRVoIP server (the destination IP address is 10.50.0.2). The packets are sent from the VPN server using a private address assigned to the LMRVoIP client (the IP address is 10.50.0.11). Figure 4.11 shows packets captured between the VPN server and the LMRVoIP server. Note that packets are sent "in the clear" and appear to be standard LMRVoIP application packets. Packets are received at the LMRVoIP server as though they were sent by the LMRVoIP client, but with the VPN's IP address as the source IP address. Thus, packets sent form the LMRVoIP server to the LMRVoIP client will be addressed back to the VPN server.

The LMRVoIP application performed correctly with the client using a virtual private network. Note that some modest latency is introduced by the extra packet processing at the LMRVoIP client and the VPN server and that a modest increase in bandwidth consumption occurs due to the overhead of packet encapsulation between the LMRVoIP client and the VPN server.

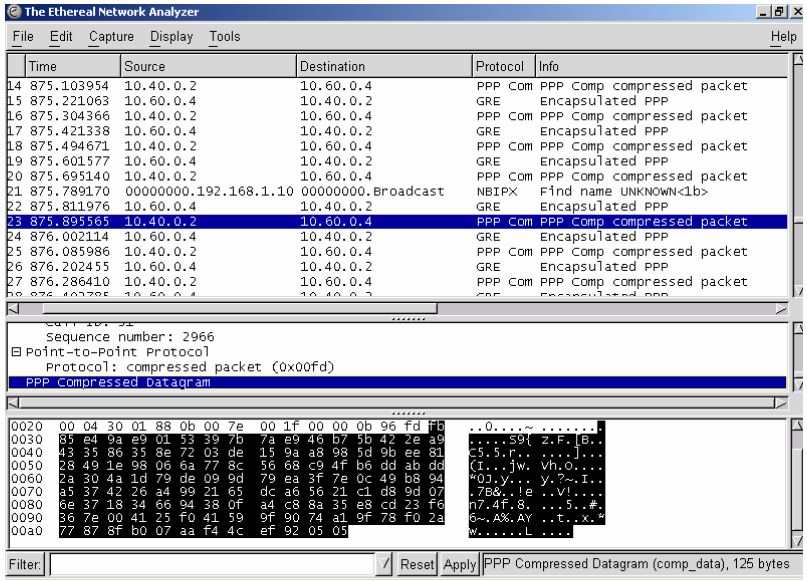

Figure 4.10. Packets captured leaving the client showing encapsulation for VPN**.**

4.9 Summary

This chapter presented key assumptions made in this research, types of traffic used to perform various experiments, test scripts developed, and test procedures. The configurations for FIFO, PQ, WFQ, CBWFQ, and VPN were presented and the corresponding results were presented and discussed. Chapter 5 provides conclusions and suggests future work.

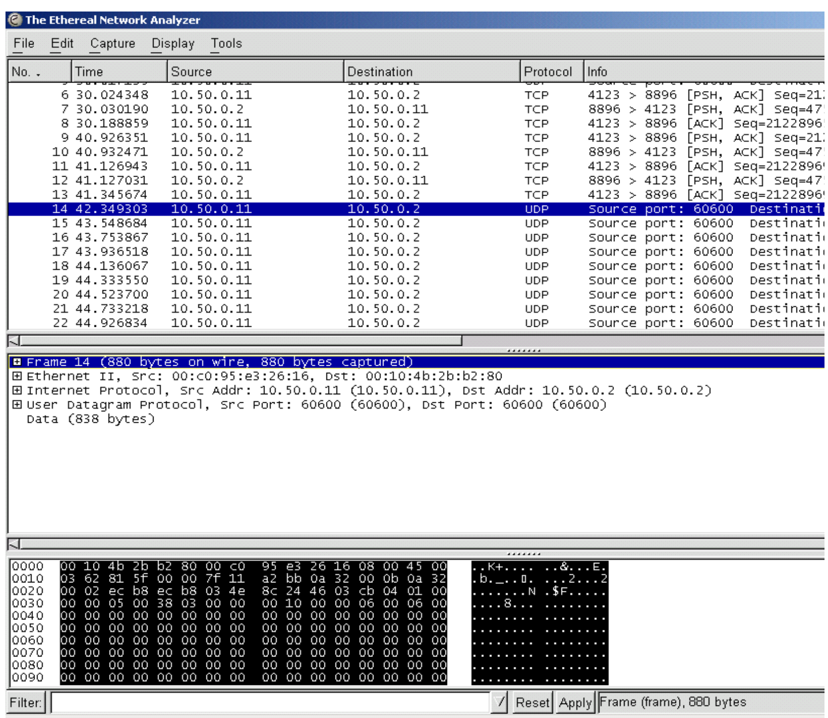

Figure 4.11. Packets captured at the server network showing packets in the "clear"**.**

Chapter 5. Conclusions and Future Work

This concluding chapter first briefly summarizes the salient features of this thesis. The next section then discusses the relative advantages and disadvantages of the four different router queuing schemes based on the results from this research. This section also suggests possible situations where these queuing schemes can be used and not used. The final section cites potential topics for future research.

5.1 Summary

The performance of a typical LMRVoIP application was studied in this thesis under both continuous and ON-OFF voice traffic cases. The ON-OFF voice traffic represents characteristics of a typical LMRVoIP call, as communication is mostly bursty and half-duplex. Jittter and application throughput are important QoS metrics for a real-time application. The LMRVoIP application's performance was investigated under four different router queuing schemes, first-in first-out queuing, priority queuing, weighted fair queuing, and class-based weighted fair queuing. Performance was based on four performance metrics, average voice application throughput, jitter, maximum jitter, and signaling overhead. The procedure for configuring FIFO, PQ, WFQ, and CBWFQ router queuing schemes and the associated code were presented in Chapter 4. Chapter 4 also discussed results for the four different QoS router queuing schemes. Graphs plotting the four performance metrics for the four queuing schemes were shown in Section 4.7. Finally, the performance of the LMRVoIP application was studied with the use of a virtual private network, taking into consideration packet encryption, packet decryption, and the latency introduced by extra packet processing at the LMRVoIP client.

5.2 Conclusions

In first-in first-out queuing, we are unable to prioritize LMRVoIP traffic. With FIFO queuing, the first packet placed in the outbound queue is transmitted first, regardless of the importance of the packet. This method is obviously detrimental to LMRVoIP traffic because voice traffic needs to be given priority over other, delay-tolerant traffic and should move to the front of the queue for transmission. As a result, LMRVoIP traffic experiences low throughput because of high packet loss, high average jitter, and high maximum jitter. Therefore, FIFO provides poor performance in heavily loaded networks.

Priority queuing has the best overall performance in terms of throughput, average jitter, and maximum jitter for both continuous and ON-OFF traffic cases because it provides exclusive priority for voice traffic. However, if there are a sufficient number of voice flows, voice traffic can consume too much link capacity. This situation can lead to starvation for background traffic, as PQ provides strict priority to voice traffic. The goal of deploying a queuing scheme is to be fair to all the flows and to give voice traffic some priority, though not exclusive priority. Therefore, PQ may not an acceptable solution for environments which may have a large amount of voice traffic, because PQ tends to starve non-voice applications in such situations. But, PQ can be an acceptable solution for LMRVoIP if the level of voice traffic, relative to link capacity, is not too high. For networks carrying multiple types of delay-intolerant traffic, for example if we have LMRVoIP traffic and video traffic sharing the same links, PQ can be used as long as the amount of delay-intolerant traffic is not too high. Priorities should be set to match application requirements. Often, LMRVoIP might be considered to be mission critical and, thus, would be assigned to the highest priority, other delay-intolerant traffic could be assigned to a lower priority, and non-real-time traffic could be assigned to the lowest priority level.

In weighted fair queuing, the bandwidth allocated to a flow depends on the number of flows and their IP Precedence values. Thus, we cannot allocate a fixed amount of bandwidth to LMRVoIP traffic with WFQ configuration. Under the assumptions of this study, 16 Kbps is required for continuous voice traffic case and 3.6 Kbps is required for ON-OFF voice traffic case. In this research performance of LMRVoIP application under WFQ was studied with one and two background flows. It was found that an increase in number of background flows adversely affects the LMRVoIP application, leading to a reduction in application throughput and an increase in average jitter. In a live network, there can be many flows and the number of flows may vary, so WFQ should not be used as we cannot guarantee a certain amount of bandwidth to the LMRVoIP traffic. In a network, where the number of flows is constant or if we can determine the number of flows in advance, then WFQ can be used, with LMRVoIP traffic being allocated its required bandwidth.

Class-based weighted fair queuing has properties of both PQ and WFQ. We can assign LMRVoIP traffic to a priority queue and, also, configure the bandwidth required for LMRVoIP traffic to the class designated for it. If the LMRVoIP traffic exceeds the configured bandwidth, then the next configured class is served. After this class is serviced, CBWFQ checks the priority class for any packet waiting for transmission before servicing other classes. Thus, CBWFQ overcomes the limitations of PQ by not starving background traffic and it overcomes the limitations of WFQ and by offering the ability to configure bandwidth for a prioritized class. If we know in advance the maximum number of sources, i.e., LMRVoIP servers and clients, that will generate voice traffic, then we can configure the network capacity with CBWFQ. Thus, for different network situations, both PQ and CBWFQ could perform almost equally well. Therefore, CBWFQ is attractive for LMRVoIP in many situations because it provides priority to important traffic and, at the same time, avoids starvation and provides appropriate fairness to all other traffic.

Virtual private networks are used to make communication secure through authentication and, potentially, encryption for privacy. With the use of a VPN, there may be a modest increase in latency due to extra packet processing at the LMRVoIP client and VPN server and, also, an increase in bandwidth due to the overhead of packet encapsulation. The LMRVoIP application was tested over a VPN and it was found to perform well.

5.3 Contributions

This research provided results for the performance of a time-sensitive LMRVoIP system. Results for the performance evaluation of a LMRVoIP application was provided for four different Quality of Service queuing schemes, which are First-In First-Out Queuing, Priority Queuing, Weighted Fair Queuing, and Class Based Weighted Fair Queuing. Results are provided for both ON-OFF traffic case and continuous traffic case. CBWFQ is attractive to

LMRVoIP traffic in many situations, because it overcomes the limitations of both PQ and WFQ. In CBWFQ, we can assign LMRVoIP class to a priority queue and, also, configured the bandwidth required for LMRVoIP traffic. CBWFQ overcomes the limitations of PQ by not starving the background traffic and it overcomes the limitation of WFQ by offering ability to configure bandwidth for the LMRVoIP class.

Different tools such as ethereal, iperf, TCL, and expect have been integrated so that different experiments can be performed efficiently. Expect scripts have been developed for automatic configuration of the network for different queuing schemes so that regression testing can be performed quickly and efficiently. TCL scripts have been developed to analyze the data from regression tests to get performance metrics, which are, UDP application throughput, jitter, maximum jitter, and number of TCP packets.

5.4 Future Work

There are several directions in which research can take place in future. Wireless networks have properties of high bit error rate and relatively lower bandwidth than wired media. Performance comparison of different low-bit rate codecs in a wireless network can be studied. Measurement of voice quality is an important issue for packet-switched VoIP networks because it uses non-linear low-bit rate codecs. Perceptual speech quality measurement (PSQM) is an objective voice quality algorithm defined by ITU-T recommendation P.861 [ITU98]. PSQM is used primarily to test networks that use speech compression, digital speech interpolation, and packetization. Experiments to measure voice quality using PSQM can be conducted. Echo is a phenomenon related to delay which exists in PSTN and VoIP networks, but is more noticeable in the latter because of its levels of delay [PH01]. Investigation of different echo cancellation schemes can be done in future. Another possible future research topic is differential treatment of different flows of LMRVoIP application traffic. One variant of this would consider the priority of different users. The source would send packets in such a manner that intermediate routers could apply appropriate QoS for traffic from only critical and high-priority users.

In a more complex network supporting multiple applications and multiple flows with quality of service, queuing schemes such as WFQ and CBWFQ may also still allow some voice packets to be lost. Error recovery schemes that will eliminate or reduce this problem can be investigated. Potential fundamental approaches include forward error correction (FEC) [KR03], automatic repeat request (ARQ) [KR03], or some hybrid of FEC and ARQ. FEC implemented at the application allows information in lost packets to be recovered based on information in other packets that are received. Coding must be applied across multiple packets, which will introduce delay at the sender. FEC will also introduce computational complexity and increase network utilization even in error-free environments. ARQ can introduce delay, but likely would work well under moderate load conditions without continuously increasing network utilization like FEC. Use of both FEC and ARQ could increase jitter, which could be accommodated through the use of a jitter (or elastic) buffer at the receiver. The objective of the research would be to find the best match of existing error correction schemes to the properties of land mobile radio codecs and user requirements.

Bibliography

[Cat02a] Catalyst Communication Technologies, *Network Access Radio Product Description*, Forest, VA, July 2002. Available at http://www.catcomtec.com/content/view/47/79

[Cat02b] Catalyst Communication Technologies, *IP Tone Product Description*, Forest, VA, December 2002. Available at http://www.catcomtec.com/content/view/44/78

[Cat02c] Catalyst Communication Technologies, *IP Link Product Description*, Forest, VA, January 2002. Available at http://www.catcomtec.com/content/view/42/74

[Cis01] Cisco Systems, *Quality of Service for Voice Over IP*, San Jose, CA. Available at http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800d6b73.shtml#xtocid0

[Cis06a] Cisco Systems, *Congestion* Management Overview, San Jose, CA. Available at http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800b75a9.html

[Cis06b] Cisco Systems, *Generic Routing Encapsulation*, San Jose, CA. Available at http://www.cisco.com/en/US/tech/tk827/tk369/tk287/tsd_technology_support_sub-protocol_home.html

[Cis07a] Cisco Systems, *Understanding Delay in Packet Voice networks*, San Jose, CA. Available at http://www.cisco.com/warp/public/788/voip/delay-details.html

[Cis07b] Cisco Systems, *Cisco IOS Sofware*, San Jose, CA. Available at http://www.cisco.com/en/US/products/sw/iosswrel/products_ios_cisco_ios_software_category_home.html

[Cis07c] Cisco Systems, *Cisco IOS Interface Command Reference*, San Jose, CA. Available at http://www.cisco.com/en/US/partner/products/sw/iosswrel/ps1831/ products _command_reference_chapter09186a00800880c2.html

[Cur03] I. D. D. Curcio, "Multimedia Streaming Over Mobile Networks: European Perspective," *Wireless Internet Handbook: Technologies, Standards, and Applications*, CRC Press, Portland, OR, 2003.

[DAB06] B. Dekeris, T. Adomkus, A. Budnikas, "Analysis of QoS Assurance Using Weighted Fair Queueing (WFQ) Scheduling Discipline with Low Latency Queue (LLQ)," *28 th International Conference on Information Technology Interfaces*, pp 507-512, Cavtat, Croatia, June, 2006

[DB03] S. K. Das and K. Basu, "VoIP Services in Wireless Networks," *Wireless Internet Handbook: Technologies, Standards, and Applications*, CRC Press, Portland, OR, 2003.

[DPT03] J. C. Dvorak, C. Pirillo, and W. Taylor, *Online, The Book*, Pearson Education, Upper Saddle River, NJ, October 2003.

[DSD01] R. I. Desourdis, D. R. Smith, and R. J. Dewey, *Emerging Public Safety Wireless Communication Systems*, Artech House, Norwood, MA, 2001.

[Eth05] Ethereal Software, *Ethereal*, September, 2005. Available at http://www.ethereal.com/

[Exp05] National Institute of Standards and Technology, *The Expect Home Page*, Feb 2005. Available at, http://expect.nist.gov/#windows

[Goo02] B. Goode, "Voice Over Internet Protocol," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1495-1517, September 2002.

[GDR05] J. Georges, T. Divoux, E. Rondeau, "Strict Priority versusWeighted Fair Queueing in Switched Ethernet networks for time critical applications," *Proceedings of the 19th IEEE Intenational Parallel and Distributed Processsing Symposium*, pp 141-141, Université Henri Poincaré, France, 2005

[Hel02] G. Held, *Quality of Service in a Cisco Networking Environment*, John Wiley & Sons, Ltd., Chichester, West Sussex, UK, May 2002.

[HS+99] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," Internet Engineering Task Force RFC 2543, March 1999.

[Huc03] D. Hucaby, *CCNP BCMSN Exam Certification Guide*, Cisco Press, Indianapolis, IN, September 2003.

[Iec03] International Engineering Consortium, *H.323: Definitions and Overview*, Chicago, IL. Available at http://www.iec.org/online/tutorials/h323

[Ipe05] National Laboratory for Applied Network Research, *IPERF*, May, 2005. Available at http://dast.nlanr.net/projects/Iperf/

[ITU96a] Recommendation G.723.1, "Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s," International Telecommunications Union, Geneva, March 1996.

[ITU96b] Recommendation G.729, "Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Llinear- Prediction(CS-ACELP)," International Telecommunications Union, Geneva, March 1996.

[ITU98] Recommendation ITU-T P.861, “Objective Quality Measurement of Telephone Band (300-3400 Hz) Speech Codecs,” International Telecommunications Union, Geneva, February 1998.

[Kea00] S. Keagy, *Integrating Voice and Data Networks*, Cisco Press, Indianapolis, IN, October 2000.

[KK01] B. Klepec and A. Kos, “Performance of VoIP Applications in a Simple Differentiated Services Network Architecture,” *Proceedings Of International Conference on Trends in Communication*, Eurocon 2001, vol.1, pp. 214-217, Bratislava, Slovak Republic, July 2001.

[KR03] J. F. Kurose, K.W. Ross, *Computer Networking, A Top-Down Approach Featuring the Internet*, Addison Wesley, Boston, MA, July 2002.

[MAC02a] M/A-COM, “M/A-COM Launches New Line of Project 25 Compliant Conventional Products,” April 2002. Available at http://www.macom-wireless.com/news/pressdetail.asp?id=28

[MAC02b] M/A-COM, “M/A-COM Unveils P25IP: First IP-based Conventional P25 Mobile Radio System,” August 2002. Available at http://www.macom-wireless.com/news/pressdetail.asp?id=38

[MAC03a] M/A-COM, “MASTR III P25 Station UHF,” Lynchburg, VA, November 2003. Available at http://www.macomwireless.com/products/p25/datasheets/7106.pd

[MAC03b] M/A-COM, “MASTR III P25 Station VHF,” Lynchburg, VA, July 2004. Available at http://www.macomwireless.com/products/p25/datasheets/6030.pdf

[MAC03c] M/A-COM, “NetworkFirst,” Lynchburg, VA, August 2003. Available at http://www.networkfirst.com/ features/solution

[MAC03d] M/A-COM, “P25 SitePro Controller,” Lynchburg, VA, March 2003. Available at http://www.macomwireless.com/products/p25/datasheets/7109.pdf

[Mic98] Microsoft Corporation, *Installing, Configuring, and Using PPTP with Microsoft Clients and Server*, Redmond, WA, March 1998. Available at http://msdn2.microsoft.com/en-us/library/ms811078.aspx

[Mic04] Microsoft Corporation, *Windows 2000 Home Page*, Redmond, WA, 2004. Available at http://www.microsoft.com/windows2000/default.mspx

[MMF01] S. McQuerry, K. McGrew, and S. Foy, *Cisco Voice over Frame Relay, ATM, and IP*, Cisco Press, Indianapolis, IN, April 2001.

[Mot04] Motorola, “Astro 25 Network,” 2004. Available at http://www.motorola.com/governmentandenterprise/northamerica/enus/public/functions/browseproduct/productdetailpage.aspx?navigationpath=id_804i/id_1174i.

[Mot99] Motorola, “Astro Network,” 1999. Available at http://www.motorola.com//governmentandenterprise/northamerica/en-us/public/functions/browseproduct us/public/functions/browseproduct /id_1173i

[MT02] J. H. Mock and W. T. Miller, “A Voice Over IP Solution for Mobile Radio Interoperability,” *Proceedings of Vehicular Technology Conference*, vol. 3, pp. 1338-1341, September 2002.

[MU01] P. C. Mehta and S. U. Dhani, “Overview of Voice Over IP,” Technical Report, MS-CIS-01-31, University of Pennsylvania, 2001. Available at http://www.cis.upenn.edu/~udani/papers/OverviewVoIP.pd

[Ous94] J. K. Ousterhout, *Tcl and the Tk Tollkit*, Addison-Wesley, Boston, MA, March 1994.

[SL+03] H. Sanneck, N. T. L. Lee, M. Haardt, and W. Mohr, “Selective Packet Prioritization for Wireless Voice over IP,” *Proceedings of Fourth International Symposium of Wireless Personal Multimedia Communications*, pp. 621-630, September 2001.

[Sch03] J. Schiller, *Mobile Communications*, Addison Wesley, Boston, MA, 2nd edition, August 2003.

[PH01] S. Pracht and D. Hardman, “Voice Quality in Converging Telephony and IP Networks,” White Paper, Agilent Technologies, October 2001. Available at http://whitepapers.tmcnet.com/detail/RES/971973879_4.html&src=FEATURE_SPOTLIGHT

[Tcl04] *TCL Developer Exchange*, Oct 2004. Available at http://www.tcl.tk/

[Tho05] H. O. R. Thomschutz, “Security in Packet-Switched Land Mobile Radio Backbone Networks,” Master’s Thesis, Virginia Polytechnic Institute and State University, May 2005. Available at http://scholar.lib.vt.edu/theses/available/etd-06152005-221301

[Tho03] T. Thomas, *OSPF Network Design Solutions*, Cisco Press, Indianapolis, IN, April 2003.

[Tig07] N.M. Tiglao, “Utility-Based Enhanced Priority Scheduler for Differentiated Services,” *Proceedings of the First Asia International Conference on Modelling and Simulation*, pp 187-192, Phyket, Thailand, 2007

[Tsi02] S. A. Tsiakkouris, “Investigation of a Packet-Switched Inter-System Interface for Land Mobile Radio Systems,” Master’s Thesis, Virginia Polytechnic Institute and State University, July 2002. Available at *http://scholar.lib.vt.edu/theses/available/etd-07312002-135257/*

[USD03] USDA Forest Service, “P25 Frequently Asked Questions,” July 2003. Available at http://www.fs.fed.us/fire/niicd/main_page_files/p25_faq.pdf

[Wel03] B. B. Welch, *Practical Programming in Tcl and Tk*, Prentice-Hall, Upper Saddle River, NJ, June 2003.

[Wla03a] IEEE 802.11 Wireless Local Area Networks Working Group, The Institute of Electrical and Electronics Engineers, New York, NY, 2007.

[Wla03b] IEEE 802.15 Wireless Personal Area Networks Working Group, The Institute of Electrical and Electronics Engineers, New York, NY, 2007.

[Wla03c] IEEE 802.16 Broadband Wireless Access Standards Working Group, The Institute of Electrical and Electronics Engineers, New York, NY, 2007.

Appendix A: Router Configurations

This appendix documents the router configurations used in the experiments for different queuing schemes. The network configuration shown in Figure 3.1 is assumed. Routers can be configured using the scripts in Appendix B.

A.1. Router ONE Configuration

```
hostname ONE

enable password cisco

class-map match-all voice
  match access-group 101

policy-map catalyst
  class voice
    priority 17
  class class-default
   fair-queue

interface Ethernet0
 ip address 10.0.0.2 255.255.255.0

interface Ethernet1
 ip address 10.10.0.1 255.255.255.0

interface Serial0
 ip address 10.20.0.1 255.255.255.0
 no fair-queue

ip classless
ip route 10.30.0.0 255.255.255.0 10.20.0.2
ip route 10.40.0.0 255.255.255.0 10.20.0.2
ip route 10.50.0.0 255.255.255.0 10.20.0.2

access-list 101 permit udp any any eq 60600
access-list 101 permit tcp any any eq 8896
priority-list 2 protocol ip high tcp 8896
priority-list 2 protocol ip high udp 60600
no cdp run

route-map setprecedence permit 10
 match ip address 101
 set ip precedence critical
 end
```

A.2. Router TWO Configuration

```
hostname TWO

enable password cisco

class-map match-all voice
```

```
  match access-group 101

policy-map catalyst
  class voice
    priority 17
  class class-default
   fair-queue

interface Serial0
 ip address 10.20.0.2 255.255.255.0
 no fair-queue
 clockrate 38400

interface Serial1
 ip address 10.30.0.1 255.255.255.0

ip route 10.0.0.0 255.255.255.0 10.20.0.1
ip route 10.10.0.0 255.255.255.0 10.20.0.1
ip route 10.40.0.0 255.255.255.0 10.30.0.2
ip route 10.50.0.0 255.255.255.0 10.30.0.2

access-list 101 permit udp any any eq 60600
access-list 101 permit tcp any any eq 8896
priority-list 2 protocol ip high tcp 8896
priority-list 2 protocol ip high udp 60600

route-map setprecedence permit 10
 match ip address 101
 set ip precedence
 end
```

A.3. Router THREE Configuration

```
hostname THREE

enable password cisco

class-map match-all voice
  match access-group 101

policy-map catalyst
  class voice
    priority 17
  class class-default
   fair-queue

interface Ethernet0
ip address 10.50.0.1 255.255.255.0

interface Ethernet1
ip address 10.40.0.1 255.255.255.0

interface Serial0
ip address 10.30.0.2 255.255.255.0
no fair-queue
clockrate 38400
```

```
ip route 10.0.0.0 255.255.255.0 10.30.0.1
ip route 10.10.0.0 255.255.255.0 10.30.0.1
ip route 10.20.0.0 255.255.255.0 10.30.0.1

access-list 101 permit udp any any eq 60600
access-list 101 permit tcp any any eq 8896
priority-list 2 protocol ip high tcp 8896
priority-list 2 protocol ip high udp 60600

route-map setprecedence permit 10
match ip address 101
set ip precedence critical
end
```

Appendix B: Expect Test Scripts

This appendix specifies scripts used with Expect to enable FIFO, PQ, WFQ, and CBWFQ queuing mechanisms on Router ONE, Router TWO, and Router THREE, as shown in Figure 3.1.

B.1. First-In First-Out (FIFO) Queuing

B.1.1. Script to Configure Router ONE for FIFO Queuing

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# FIFO Configuration at Router 1 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 1
# Telnetting into router 1
spawn telnet 10.10.0.1
expect "ONE>"
send "enab\r"

# enable password is "cisco"
# Change if configured password is different from “Cisco”
expect "Password:"
send "cisco\r"
expect "ONE#"
send "config term\r"
expect "ONE(config)#"
puts "\n"
send "interface s 0\r"
expect "ONE(config-if)#"

# Note that this script enables FIFO and
# Disables PQ, WFQ and CBWFQ
send "no priority-group 2\r"
send "no service-policy output catalyst\r"
send "no fair-queue\r"
send "no shut\r"
expect "ONE(config-if)#"
puts "\n"
puts "Done with All the Configuration"
send "end\r"
expect "ONE#"
```

B.1.2. Script to Configure Router TWO for FIFO Queuing

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# FIFO Configuration at Router 2 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100
```

```
# Note that this script configures router 2
# Telnetting into router 2
spawn telnet 10.10.0.2
expect "TWO>"
send "enab\r"

# enable password is "cisco"
# Change if configured password is different from “Cisco
expect "Password:"
send "cisco\r"
expect "TWO#"
send "config term\r"
expect "TWO(config)#"
puts "\n"
send "interface s 0\r"
expect "TWO(config-if)#"

# Note that this script enables FIFO and
# Disables PQ, WFQ and CBWFQ
send "no priority-group 2\r"
send "no service-policy output catalyst\r"
send "no fair-queue\r"
send "no shut\r"
expect "TWO(config-if)#"
puts "\n"
puts "Done with All the Configuration"
send "end\r"
expect "TWO#"
```

B.1.3. Script to Configure Router THREE for FIFO Queuing

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# FIFO Configuration at Router 3 Fig 3.1
# usage is :        tclsh80  <script name>


# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 3
# Telnetting into router 3
spawn telnet 10.30.0.2
expect "THREE>"
send "enab\r"

# enable password is "cisco"
# Change if configured password is different from “Cisco”
expect "Password:"
send "cisco\r"
expect "THREE#"
send "config term\r"
expect "THREE(config)#"
send "interface e 1\r"
expect "THREE(config-if)#"
```

```
# Disable route-map
send "no ip policy route-map setprecedence\r"

# Note that this script enables FIFO and
# Disables PQ, WFQ and CBWFQ
send "interface s 0\r"
expect "THREE(config-if)#"
send "no priority-group 2\r"
send "no service-policy output catalyst\r"
send "no fair-queue\r"
send "no shut\r"
expect "THREE(config-if)#"
puts "\n"
puts "Done with All the Configuration"
send "end\r"
expect "THREE#"
```

B.2. Priority Queuing (PQ)

B.2.1. Script to Configure Router ONE for PQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# PQ Configuration at Router 1 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 1
# Telnetting into router 1
spawn telnet 10.20.0.1
expect "ONE>"
send "enab\r"
expect "Password:"

# enable password configured is "cisco"
# Change if configured password is different from “Cisco”
send "cisco\r"
expect "ONE#"
send "config term\r"
expect "ONE(config)#"
puts "\n"
send "interface s 0\r"
expect "ONE(config-if)#"

# Note that this script enables PQ and
# Disables FIFO, WFQ and CBWFQ
send "no fair-queue\r"
send "no service-policy output catalyst\r"
send "priority-group 2\r"
send "no shut\r"
expect "ONE(config-if)#"
send "end\r"
expect "ONE#"
```

B.2.2. Script to Configure Router TWO for PQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# PQ Configuration at Router 2 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 2
# Telnetting into router 2
spawn telnet 10.20.0.2

expect "TWO>"
send "enab\r"
expect "Password:"

# enable password is "cisco"
# Change if configured password is different from "Cisco"
send "cisco\r"
expect "TWO#"
send "config term\r"
expect "TWO(config)#"
puts "\n"
send "interface s 0\r"
expect "TWO(config-if)#"

# Note that this script enables PQ and
# Disables FIFO, WFQ and CBWFQ
send "no fair-queue\r"
send "no service-policy output catalyst\r"
send "priority-group 2\r"
send "no shut\r"
expect "TWO(config-if)#"
send "end\r"
expect "TWO#"
```

B.2.3. Script to Configure Router THREE for PQ

```
# Expect Router Configuration Script
# PQ Configuration at Router 3 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 3
# Telnetting into router 3
spawn telnet 10.30.0.2

expect "THREE>"
send "enab\r"
expect "Password:"

# enable password is "cisco"
# Change if configured password is different from "Cisco"
```

```
send "cisco\r"
expect "THREE#"
send "config term\r"
expect "THREE(config)#"
send "interface e 1\r"
expect "THREE(config-if)#"

# Disable route-map
send "no ip policy route-map setprecedence\r"

expect "THREE(config-if)#"
puts "\n"
send "interface s 0\r"

# Note that this script enables PQ and
# Disables FIFO, WFQ and CBWFQ
expect "THREE(config-if)#"
send "no fair-queue\r"
send "no service-policy output catalyst\r"
send "priority-group 2\r"
send "no shut\r"
expect "THREE(config-if)#"
send "end\r"
expect "THREE#"
```

B.3. Weighted Fair Queuing (WFQ)

B.3.1. Script to Configure Router ONE for WFQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# WFQ Configuration at Router 1 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100


# Note that this script configures router 1
# Telnetting into router 1
spawn telnet 10.10.0.1

expect "ONE>"
send "enab\r"
expect "Password:"

# enable password is "cisco"
# Change if configured password is different from “Cisco”
send "cisco\r"
expect "ONE#"
send "config term\r"
expect "ONE(config)#"
puts "\n"
send "interface s 0\r"
expect "ONE(config-if)#"
```

```
# Note that this script enables WFQ and
# Disables PQ, FIFO and CBWFQ
send "no priority-group 2\r"
send "no service-policy output catalyst\r"
send "fair-queue\r"
send "no shut\r"
expect "ONE(config-if)#"
puts "\n"
puts "Done with All the Configuration"
send "end\r"
expect "ONE#"
```

B.3.2. Script to Configure Router TWO for WFQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# WFQ Configuration at Router 2 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 2
# Telnetting into router 2
spawn telnet 10.10.0.2

expect "TWO>"
send "enab\r"
expect "Password:"

# enable password is "cisco"
# Change if configured password is different from "Cisco"
send "cisco\r"
expect "TWO#"
send "config term\r"
expect "TWO(config)#"
puts "\n"

# Note that this script enables WFQ and
# Disables PQ, FIFO and CBWFQ
send "interface s 0\r"
expect "TWO(config-if)#"
send "no priority-group 2\r"
send "no service-policy output catalyst\r"
send "fair-queue\r"
send "no shut\r"
expect "TWO(config-if)#"
puts "\n"
puts "Done with All the Configuration"
send "end\r"
expect "TWO#"
```

B.3.3. Script to Configure Router THREE for WFQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# WFQ Configuration at Router 3 Fig 3.1
```

```
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 3
# Telnetting into router 3
spawn telnet 10.20.0.2
expect "THREE>"
send "enab\r"
expect "Password:"
send "cisco\r"
expect "THREE#"
send "config term\r"
expect "THREE(config)#"
puts "\n"
send "interface e 1\r"
expect "THREE(config-if)#"

# Enable route-map to mark Catalyst packets
send "ip policy route-map setprecedence\r"

# Note that this script enables WFQ and
# Disables PQ, FIFO and CBWFQ
send "interface s 0\r"
expect "THREE(config-if)#"
send "no priority-group 2\r"
send "no service-policy output catalyst\r"
send "fair-queue\r"
send "no shut\r"
expect "THREE(config-if)#"
puts "\n"
puts "Done with All the Configuration"
send "end\r"
expect "THREE#"
```

B.4. Class-Based Weighted Fair Queuing (CBWFQ)

B.4.1. Script to Configure Router ONE for CBWFQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# CBWFQ Configuration at Router 1 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times out after 100secs of inactivity
set timeout 100

# Note that this script configures router 1
# Telnetting into router 1
spawn telnet 10.20.0.1
expect "ONE>"
send "enab\r"
expect "Password:"

# enable password is "cisco"
# Change if configured password is different from “Cisco”
```

```
send "cisco\r"
expect "ONE#"
send "config term\r"
expect "ONE(config)#"
puts "\n"
send "interface s 0\r"
expect "ONE(config-if)#"

# Note that this script enables CBWFQ and
# Disables PQ, FIFO and WFQ
send "no fair-queue\r"
send "no priority-group 2\r"
send "service-policy output catalyst\r"
send "no shut\r"
expect "ONE(config-if)#"
send "end\r"
expect "ONE#"
```

B.4.2. Script to Configure Router TWO for CBWFQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# CBWFQ Configuration at Router 2 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times after 100secs of inactivity
set timeout 100

# Note that this script configures router 2
# Telnetting into router 2
spawn telnet 10.20.0.2
expect "TWO>"
send "enab\r"
expect "Password:"


# enable password is "cisco"
# Change if configured password is different from “Cisco”
send "cisco\r"
expect "TWO#"
send "config term\r"
expect "TWO(config)#"
puts "\n"
send "interface s 0\r"
expect "TWO(config-if)#"

# Note that this script enables CBWFQ and
# Disables PQ, FIFO and WFQ
send "no fair-queue\r"
send "no priority-group 2\r"
send "service-policy output catalyst\r"
send "no shut\r"
expect "TWO(config-if)#"
send "end\r"
expect "TWO#"
```

B.4.3. Script to Configure Router THREE for CBWFQ

```
# Expect Router Configuration Script
# Expect version 5.21, tcl version 8.0
# CBWFQ Configuration at Router 3 Fig 3.1
# usage is :        tclsh80  <script name>

# Script times after 100secs of inactivity
set timeout 100

# Note that this script configures router 3
# Telnetting into router 3
spawn telnet 10.30.0.2
expect "THREE>"
send "enab\r"
expect "Password:"

# enable password is "cisco"
# Change if configured password is different from “Cisco”
send "cisco\r"
expect "THREE#"
send "config term\r"
expect "THREE(config)#"
send "interface e 1\r"
expect "THREE(config-if)#"

# Disable route-map
send "no ip policy route-map setprecedence\r"

# Note that this script enables CBWFQ and
# Disables PQ, FIFO and WFQ
expect "THREE(config-if)#"
puts "\n"
send "interface s 0\r"
expect "THREE(config-if)#"
send "no fair-queue\r"
send "no priority-group 2\r"
send "service-policy output catalyst\r"
send "no shut\r"
expect "THREE(config-if)#"
send "end\r"
expect "THREE#"
```

Appendix C. TCL Data Analysis Script

This appendix provides the script used to extract numerical results from a trace file collected by Ethereal.

```
# tcl version is 8.3
# usage is:  tclsh83  <script name>  <ethereal trace filename>

# The format of "tethereal" file is:
# 0.000000 10.0.0.1 -> 10.50.0.4    TCP 2003 > 8896 [SYN] Seq=674621270 Ack=0 Win=16384 Len=0
# 0.012147 10.0.0.1 -> 10.50.0.4    TCP 2004 > 8896 [SYN] Seq=674664491 Ack=0 Win=16384 Len=0
# 0.038552 10.0.0.1 -> 10.50.0.4    TCP 2003 > 8896 [ACK] Seq=674621271 Ack=3416506456 Win=17520
# Len=0
# 0.130701     10.0.0.1 -> 10.50.0.4    TCP 2003 > 8896 [PSH, ACK] Seq=674621271 Ack=3416506456
# Win=17520 Len=284
# 0.009868     10.0.0.1 -> 10.50.0.4    TCP 2004 > 8896 [ACK] Seq=674664492 Ack=3416546200
# Win=17520 Len=0

# Variables to store number of Voice (vcount), background (bcount) and TCP (tcount) packets
 set vcount 0
 set bcount 0
 set tcount 0

# Variables to store total TCP bytes(tcpby) and bits(tcpbi)
 set tcpby 0
 set tcpbi 0
 set sum 0
 set totabsj 0
 set totdel 0
 set ocount 0

# The variable 'voicejitter' is the time difference between successive voice packets
 set voicejitter 0

 # The variable 'delta' is the average jitter between 2 successive
 # voice packets measured at the source
 # variable 'delta' is used in calculation of jitter at the destination
 # used value of 0.197 for QoS tests and value of 0.0982 for IP Link tests

 set delta 0.0982
 set djitter 0

# The variable 'absjitter' is the actual Voice jitter between voice successive voice UDP packets
 set absjitter 0
 set great 0
 set large 0
 set largeb 0
 set maxjitter 0

# Directory where the ethereal trace files are placed
cd C:/Vijay/Catalyst/iplink/1219/proc2/server2/WFQH

set file [lindex $argv 0]
puts " filename is $file"

# Open the ethereal trace file in read mode using 'open' and resultant file identifier is set to
# variable 'input'
set input [open $file r]

# 'gets' command reads lines in the trace file one by one
# stores the read line in the variable 'line'
# returns a count of the no of the characters read in each line

while {[gets $input line]>=0} {

# UDP parameters such as number of voice packets and total jitter are calculated
# The script looks for lines with "UDP" in tethereal trace file
```

```
        if {[regexp {([0-9.]+)\s+[0-9.]+ -> [0-9.]+\s+(UDP)\s+Source port: (\d+)\s+Destination
port: (\d+)} $line - jitter protocol sourceport destport]} {

         # Catalyst Voice packets use port 60600 as destination UDP port

              if {$destport==60600} {
                                      incr vcount 1
                                      # The var 'sum' find the total time of the test
                                      set sum [expr {$sum+$jitter}]
                                      set voicejitter [expr {$voicejitter+$jitter}]
                                      set djitter [expr {$voicejitter-$delta}]
                                      set absjitter [expr abs($djitter)]

                                      if {$absjitter>1} {
                                      # puts "absolute jitter is greater than 1.00"
                                                         set voicejitter 0
                                                         set absjitter 0
                                                         incr great 1
                                                         continue
                                                        }

                                 # To calculate Total difference between voice packets
                                 set totdel [expr {$voicejitter+$totdel}]

                                 # To calculate total voice jitter
                                 set totabsj [expr {$absjitter+$totabsj}]

                                 # place the current voice difference and voice jitter
                                 # in a list
                                 lappend listabs $absjitter
                                 lappend vj $voicejitter

                                 # set the var voicejitter to zero in order to calculate
                                 # jitter between successive voice packets
                                 set voicejitter 0
                                 set absjitter 0

    # Background packets generated by 'Iperf' use 5001 as default destination port number
                                 } elseif {$destport==5001} {
                                                            set sum [expr {$sum+$jitter}]
                                                            incr bcount 1
                                            set voicejitter [expr {$voicejitter+$jitter}]
                                                            }

                              continue
                              }


# TCP parameters such as number of TCP packets and total TCP bytes are calculated
# The script looks for lines with "TCP" in tethereal trace file
# TCP packets have ACK, SYN, PSH and FIN flag bits set

        if {[regexp {([0-9.]+)\s+([0-9.]+) -> ([0-9.]+)\s+([A-Z]+)\s+[0-9]+ > [0-9]+\s+\[([A-
Z]+)(?:,\s*([A-Z]+))?\]\s+Seq=[0-9]+\s+Ack=[0 -9]+\s+Win=[0-9]+\s+Len=([0-9]+)} $line - jitter
src dest prot t1 t2 leng]} {
       # puts " jitter=$jitter src=$src dest=$dest prot=$prot t1=$t1 t2=$t2 leng=$leng"

        if { $prot=="TCP" } {
                                 incr tcount
              # below lines calculate the Total TCP bytes
              # Varibale 'tcpby' to store Total TCP bytes
                                 if { $t1=="SYN" } {
                                                 # puts "SYN"
                                                  set tcpby [expr {$tcpby+62}]
                                                    }
                                 if { $t1=="ACK" } {
                                                  set tcpby [expr {$tcpby+60}]
                                                 # puts " current TCP bytes are $tcpby bytes"
                                                  }
                                 if { $t1=="FIN" } {
```

```
                                                    set tcpby [expr {$tcpby+60}]
                                                      }
                                    if { $t1=="PSH" } {
                                                   # puts "PSH"
                                                    set tcpby [expr {$tcpby+$leng+54}]
                                                     }
                                     if {$jitter>1} {
                                                  incr large
                                                  #continue
                                                 }
                                   set sum [expr {$sum+$jitter}]
                                   set voicejitter [expr {$voicejitter+$jitter}]
                                   continue
                                   } else {
                                           if {$jitter>1} {
                                             incr largeb
                                             #continue
                                             }
                                   set sum [expr {$sum+$jitter}]
                                   incr ocount 1
                                   set voicejitter [expr {$voicejitter+$jitter}]
                                     }
                       }
      }

puts "The number of Voice packtes are $vcount"
puts "The number of TCP packtes are $tcount"
puts "The number of background packets are $bcount"
puts "The number of other packets are $ocount"
puts "The value of total jitter are $sum secs"
puts "The value of total absolute jitter are $totabsj secs"
puts " The value of total voice delay $totdel secs"
puts "The total time of this test is $sum secs"


# UDP parameters such as Average jitter, Maximum jitter and Voice throughput are
# calculated and printed

if { $vcount > 0 } {
    set avgjitter [expr {double($totabsj)/double($vcount)}]
    puts " The average jitter is $avgjitter secs"
    # Size of UDP packet using GSM codec is 384 bytes
    set vbytes [expr {$vcount*384}]
    puts "The total voice bytes is $vbytes bytes"
    set vbits [expr {$vbytes*8}]
    puts "The total voice bits is $vbits bits"
    set  vthru [expr {double($vbits)/double($sum)}]
    puts "The average voice throughput is $vthru bits/sec"
    puts "Number of packets greater than delay of 1.00 is $great"

            if { $totabsj > 0 } {
            # Arrange the list 'listabs' (contains jitter of voice packets) in descending order
            set absdesc [lsort -real -decreasing $listabs]
            set maxjitter [ lindex $absdesc 0 ]
            puts "The maximum Voice jitter is $maxjitter secs"
             }
 }

# TCP parameters such as TCP throughput and no. of TCP packets are calculated and printed

if { $tcount > 0}  {
    puts " Total TCP bytes is $tcpby bytes"
    set tbits [expr {$tcpby*8}]
    puts "The total TCP bits is $tbits bits"
    set  tthru [expr {double($tbits)/double($sum)}]
    puts "The average TCP throughput is $tthru bits/sec"
    puts "The LARGE JITTER TCP packets $large"
}
    puts "The LARGE JITTER BACKGROUND packets $largeb"
    puts " last jitter is $jitter"
    puts " The metrics Throughput, Average jitter, Maximum jitter and TCP Packets are:"
```

```
 puts "\n"
 puts "The number of TCP packtes are $tcount"
 puts "The average TCP throughput is $tthru bits/sec"

if { $vcount > 0 } {
             puts "The average voice throughput is $vthru bits/sec"
             puts " The average jitter is $avgjitter secs"
             puts "The maximum Voice jitter is $maxjitter secs"
             }
 exit
```

Vita

Vijayanand S. Ballapuram was born in Chennai, the capital of the state of Tamil Nadu, India. He received his bachelor's degree in Electrical and Electronics Engineering from Satyabama Engineering College (now called "Sathyabama University") Chennai, India in August 1998. Then, he earned his Master's degree in Electrical Engineering from Wichita State University, Wichita, Kansas in December 2001. He is currently working as a Technical Assistance Center Engineer for Cisco Systems in Richardson, Texas. His areas of interests include routing, switching, IP telephony, and wireless networking. In his spare time he likes to play table tennis and cricket and watch movies.