

Assessment of Voice over IP as a Solution for Voice over ADSL

Abhishek Ram

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 Science

Dr. Srinidhi Varadarajan, Chairman
Dr. Luiz DaSilva, Co-Chairman
Dr. Eunice Santos

22 May 2002
Blacksburg, Virginia

Keywords: DSL, VoATM, VoIP, QoS, Admission Control

Copyright© 2002, Abhishek Ram

Assessment of Voice over IP as a Solution for Voice over ADSL

Abhishek Ram

(ABSTRACT)

Voice over DSL (VoDSL) is a technology that enables the transport of data and multiple voice calls over a single copper-pair. VoDSL employs packet voice technology instead of the traditional circuit switched voice. Voice over ATM (VoATM) and Voice over IP (VoIP) are the two main alternatives for carrying voice packets over DSL. ATM is currently the preferred technology, since it offers the advantage of ATM's built-in Quality of Service (QoS) mechanisms. IP, on the other hand, cannot provide QoS guarantees in its traditional form. IP QoS mechanisms have been evolved only in the recent years. VoIP has gained popularity in the core networks. If it could replace VoATM in the access networks, it would open the door for end-to-end IP telephony that would result in major cost savings. In this thesis, we propose a VoIP-based VoDSL architecture that provides QoS guarantees comparable to those offered by ATM in the DSL access network.

Our QoS architecture supports Premium and Regular service categories for voice traffic and the Best-Effort service category for data traffic. Voice and data packets are placed in separate output queues at the bottleneck link. The Weighted Fair Queuing algorithm is used to schedule voice and data packets for transmission over the bottleneck link. Fragmentation of large data packets reduces the waiting time for voice packets in the link. We also propose a new admission control mechanism called Admission Control by Implicit Signaling. This mechanism takes advantage of application layer signaling by mapping it to the IP header. The router can infer the resource requirements for the connection by looking at certain field in the IP header of the application layer signaling packets. This eliminates the need for an explicit signaling protocol.

We evaluate the performance of our QoS architecture by means of a simulation study. Our primary metrics are the end-to-end delay of voice packets across the access network and the bandwidth consumed by a voice call. Our results show that the end-to-end delays of voice packets in our VoIP architecture are comparable to that in the VoATM architecture. ACIS limits the number of voice calls admitted into the premium service class and provides guaranteed service to those calls under all loads. It also provides acceptable service to regular calls under light loads. We also show that PPP is a better choice than ATM as a Layer 2 protocol for our VoIP architecture. PPP offers the advantages of low bandwidth requirement and interleaving of voice packets in between fragments of large data packets during transmission over the bottleneck link. We conclude that our VoIP architecture would be suitable for future VoDSL deployments.

Acknowledgements

I would like to thank my advisor Dr. Luiz DaSilva for his able guidance during each stage of this research. His encouragement was a great source of motivation for me. A special word of thanks goes to Dr. Srinidhi Varadarajan for his expert advice. I am also grateful to Dr. Eunice Santos for her valuable suggestions.

I would like to thank Kaustubh Phanse and Vikram Dham, both students of Dr. DaSilva, for their assistance.

And finally, I would like to express my heartfelt gratitude to family and friends for their support and encouragement.

Table of Contents

Chapter 1	1
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Contributions	2
1.3 Organization of the Thesis	3
Chapter 2	4
2 Background	4
2.1 DSL Basics.....	6
2.2 Voice over DSL.....	6
2.3 Voice over Packet Networks	8
2.4 VoATM Concepts.....	10
2.4.1 ATM Basics.....	10
2.4.2 ATM Traffic Parameters.....	12
2.4.3 ATM QoS Attributes.....	12
2.4.4 ATM Service Classes.....	13
2.4.5 ATM QoS Mechanisms.....	14
2.5 VoIP Concepts	15
2.5.1 Protocols	15
2.5.2 Protocol Overhead in VoIP.....	16
2.5.3 QoS for VoIP.....	17
2.5.3.1 Integrated Services.....	17
2.5.3.2 Differentiated Services.....	18
2.6 Chapter Summary.....	19
Chapter 3	20
3 An IP QoS Architecture for VoDSL	20
3.1 Service Classes.....	20
3.2 Queuing and Scheduling.....	21
3.3 Fragmentation.....	22
3.4 Admission Control by Implicit Signaling (ACIS)	24
3.4.1 Encoding Application Layer Signaling Information.....	24
3.4.2 Hash Table Description.....	25
3.4.3 ACIS Operation.....	26
3.5 Related Work	30
3.6 Chapter Summary	33
Chapter 4	35
4 Simulation Methodology.....	35
4.1 Objective.....	35
4.2 System Boundaries.....	35
4.3 Performance Metrics.....	35

4.4	Simulation Model.....	36
4.5	Implementation of the Simulation Model.....	39
4.6	Validation and Verification.....	41
4.6.1	Validation.....	41
4.6.2	Verification	42
4.7	Experiments.....	42
4.8	Chapter Summary	43
Chapter 5		44
5	Performance Evaluation	44
5.1	VoATM Model	44
5.2	VoIP Model.....	46
5.2.1	Weighted Fair Queuing.....	46
5.2.2	Fragmentation.....	47
5.2.3	Comparison of Layer 2 Protocols.....	49
5.2.4	ACIS.....	52
5.2.5	Effect of Traffic Loads and Traffic Mixes.....	56
5.3	Chapter Summary	59
Chapter 6		60
6	Conclusions and Future Work.....	60
6.1	Conclusions.....	60
6.2	Future Work.....	61
References		62
Glossary of Acronyms		68
Vitae		70

List of Tables

Table 2.1:	Bit rates of some standard voice encoders.....	9
Table 5.1:	ETE Delays – VoATM with UBR voice and data.....	44
Table 5.2:	ETE Delays – VoATM with CBR voice and UBR data	45
Table 5.3:	ETE Delays – VoIP with no service differentiation.....	46
Table 5.4:	ETE Delays – VoIP with WFQ.....	47
Table 5.5:	ETE Delays – VoIP with WFQ and Fragmentation.....	48
Table 5.6:	ETE Delays – VoIP over SLIP with WFQ and Fragmentation	51
Table 5.7:	ETE Delays – VoIP over SLIP Overloaded.....	52
Table 5.8:	ETE Delays – VoIP over SLIP with ACIS	53

List of Figures

Figure 2.1: Topology of DSL Access Network.....	4
Figure 2.2: VoATM for DSL.....	6
Figure 2.3: Protocol stacks for voice and data in the VoATM scenario.....	6
Figure 2.4: VoIP for DSL.....	7
Figure 2.5: AAL2 PDU format.....	11
Figure 2.6: AAL5 PDU format.....	11
Figure 2.7: ATM Protocol Stack.....	12
Figure 2.8: Probability density function for CTD.....	13
Figure 2.9: IPv4 Packet Header.....	15
Figure 2.10: The DiffServ field.....	18
Figure 3.1: Queuing and Scheduling at the CPE	21
Figure 3.2: Multilink PPP based Fragmentation and Interleaving	23
Figure 3.3: Additional IP header fields for ACIS mechanism.....	24
Figure 3.4: CPE operation on receiving “Invite” message.....	26
Figure 3.5: CPE operation on receiving voice packet	28
Figure 3.6: CPE operation on receiving “Bye” message.....	29
Figure 4.1: Simulation Model for VoATM scenarios.....	37
Figure 4.2: Simulation Model for VoIP scenarios.....	38
Figure 4.3: CPE model in the VoATM scenario.....	39
Figure 5.1: Comparison of ETE delays for UBR voice and CBR voice.....	45
Figure 5.2: Improvement in VoIP Performance.....	48
Figure 5.3: Bandwidth requirements for VoIP over ATM and VoIP over SLIP	49
Figure 5.4: Comparison of ETE delays for VoATM and VoIP.....	51
Figure 5.5: Performance of ACIS.....	53
Figure 5.6: Upgrade of Regular calls to Premium calls.....	55
Figure 5.7: Soft State in ACIS.....	56
Figure 5.8: Effect of different sizes of file uploaded via FTP on regular voice.....	57
Figure 5.9: Effect of different traffic mixes on regular voice.....	58

Chapter 1: Introduction

Digital Subscriber Line (DSL) refers to a collection of technologies used for the transmission of high-speed data over copper twisted-pair lines. It is used to connect the network service providers (NSP) and the customers, which are usually residences or small-to-medium sized businesses. Asymmetric Digital Subscriber Line (ADSL) is the most widely deployed DSL technology. ADSL provides higher bandwidth from the NSP to the customer (downstream) than from the customer to the NSP (upstream). It enables high data-rate services such as high-speed Internet access and streaming audio/video for the customers.

Traditionally, residences and small-to-medium sized businesses use the Public Switched Telephone Network (PSTN) for voice services. The voice services are circuit-switched and a dedicated copper pair is needed for each telephone line. This implies that customers requiring multiple lines would need a copper pair for each of those lines. This proves to be expensive for both the NSPs and the customers, due to the added cost of provisioning additional copper pairs. Voice over DSL (VoDSL) is a technology that provides a convenient and cost-effective solution to this problem.

VoDSL uses the existing DSL access network to provide voice services in addition to data services. Voice is packetized at the customer premises and the packet-switched DSL access network is used to deliver the voice packets to a voice gateway. The voice gateway converts packetized voice into circuit-switched voice traffic and sends it to the PSTN. Thus, a single copper pair can be used to provide data services and also one or more voice lines to the customer. This eliminates the cost of provisioning separate copper pairs for voice and data and additional copper pairs for multiple voice lines. Also, customers have the convenience of having a single NSP cater to both their voice and data needs. Another advantage of having packet-based voice is that voice calls consume bandwidth only when they are active.

1.1 Motivation

The current trend is to use Asynchronous Transfer Mode (ATM) as the transport technology for VoDSL. This is also called the Voice over ATM (VoATM) approach. VoATM takes advantage of ATM's built-in Quality of Service (QoS) mechanisms to guarantee low end-to-end delays for voice packets. ATM also uses a lightweight protocol for carrying voice, thereby ensuring high bandwidth efficiency.

An alternative approach is to use the Internet Protocol (IP) as the transport technology for packet voice. This is also called the Voice over IP (VoIP) approach. Compared to VoATM, VoIP suffers from several potential problems. The performance of voice traffic degrades in the presence of competing data traffic. End-to-end IP QoS mechanisms that prioritize voice traffic are not yet standardized. Bandwidth efficiency is another issue in VoIP. The VoIP protocol stack adds a number of headers to the voice packets, which

impose a considerable overhead on the voice packets. These difficulties have hampered the deployment of VoIP for VoDSL and made ATM the preferred protocol.

However, there is considerable interest in using IP-based data networks to replace PSTNs as carriers of voice in core networks, since it is cheaper to packetize voice and carry it over a data network. IP is certainly the preferred protocol in the core, owing to its universality. IP can run over any kind of core network, such as those based on ATM, SONET, Gigabit Ethernet or Frame Relay. Also, data traffic is growing much faster than voice traffic and hence it makes sense to use IP in the core, as it the predominant protocol for carrying data traffic over the Internet.

If the performance of VoIP in the DSL access network can be improved using IP QoS mechanisms, then the deployment of VoIP for VoDSL would yield several other benefits. First, the voice packet format used in the DSL access network would be compatible with that used in the core. This eliminates the need for a voice gateway and paves the way for end-to-end IP telephony. Also, VoIP can run over any kind of DSL network – ATM based, frame-based or PPP-based. Since IP is already being used in DSL to carry data traffic, it would be easy for voice traffic to run over IP too. Thus, a solution that achieves VoIP performance comparable to that provided by VoATM would be of enormous benefit in the DSL access network.

1.2 Thesis Contributions

This thesis aims at assessing VoIP as a solution for VoDSL. For VoIP deployment to be feasible, the IP QoS mechanisms used must be able to guarantee a service similar to that guaranteed by ATM's built-in QoS mechanisms. This involves prioritization of voice packets and protection of voice traffic from competing data traffic. In addition to this, techniques to alleviate the bandwidth overhead imposed by the VoIP protocol stack must be employed. We evaluate the effectiveness of existing techniques in achieving these objectives and attempt to identify areas where there is scope for improvement. We then propose a solution that augments the existing techniques and strengthens the case for using VoIP in future deployments of VoDSL. This research is based on simulation studies of the performance of voice traffic in the DSL access network.

The main contributions of this thesis are –

- Simulating the deployment of VoATM for VoDSL and understanding the effectiveness of ATM's built-in QoS mechanisms.
- Simulating the deployment of VoIP for VoDSL and assessing the shortcomings of this approach.
- Proposing an IP QoS architecture to support voice and data convergence in DSL.
- Devising an admission control mechanism that limits the number of voice calls entering the network and provides service guarantees to the admitted calls. This admission control mechanism is incorporated into our QoS architecture.
- Assessing the performance improvement obtained from classification, queuing and scheduling mechanisms incorporated in our architecture.

- Assessing the improvement in performance of voice traffic when large data packets, which compete for bandwidth with voice packets, are fragmented.
- Assessing the performance of our admission control mechanism in providing service guarantees to voice calls.
- Studying the performance of the above-mentioned QoS mechanisms under different loads and different mixes of traffic.
- Comparing the suitability various Layer 2 protocol options for VoIP deployment.

1.3 Organization of the Thesis

The thesis is organized as follows. Chapter 2 explains the basic concepts of DSL, packet voice, VoATM and VoIP. Chapter 3 describes our IP QoS architecture and compares it with QoS mechanisms that already exist. Chapter 4 describes the simulation methodology used in our research. Chapter 5 discusses the experiments performed to evaluate our QoS architecture. Chapter 6 states our conclusions and also discusses the scope for future work.

Chapter 2: Background

2.1 DSL Basics

Digital Subscriber Lines ([DSL], [Paradyne]) connect the NSPs and the customers. At the customer's home or office, a device called the Customer Premises Equipment (CPE) provides access to the NSP's network. The CPE connects to a DSL Access Multiplexer (DSLAM) located in the Central Office (CO) of the NSP. The DSLAM aggregates traffic from different customers and sends it over a high-speed uplink towards the core of the network. Figure 2.1 illustrates the network topology of the DSL access network. DSL has several variants ([DSL], [Ibe], [Paradyne]) that use different transmission technologies, offer different data rates and support different types of services.

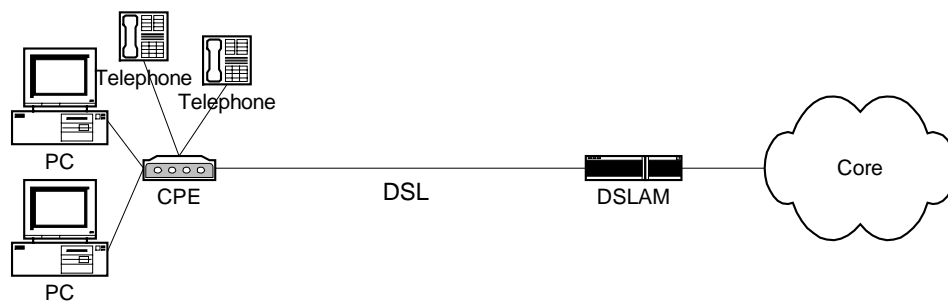


Figure 2.1: Topology of DSL Access Network

ADSL is the most common DSL technology. The downstream bandwidth (network to user) provided by ADSL is greater than the upstream bandwidth (user to network). The upstream bandwidths range from 16 Kbps to 640 Kbps. Downstream bandwidths depend upon the length of the DSL link and range from 1.544 Mbps (18000 feet) to 8.448 Mbps (9000 feet). This suits applications such as high-speed Internet access, online shopping and video-on-demand that require a high downstream data rate and a relatively low upstream data rate. ADSL allows both voice and data to be sent simultaneously over the same line by using a splitter that partitions the voice and data channels. ADSL customers are mainly small to medium sized enterprises and homes.

G.Lite (also called Splitterless ADSL or ADSL Lite) is a version of ADSL that provides lower bandwidths, but is easier to deploy. G.Lite eliminates the need for a splitter by allowing the CPE to be plugged in directly to the existing telephone socket. The absence of a filter results in increased interference between voice and data channels, thereby leading to additional noise and reduced capacity of G.Lite. G.Lite can provide downstream data rates of up to 1.5 Mbps and upstream data rates of up to 512 Kbps. The performance of G.Lite may be improved using a microfilter. It is cheaper and easier to install than a splitter and can provide separation of voice and data channels.

High-bit-rate DSL (HDSL) uses two or three twisted-pair lines operating in full-duplex mode and provides data rates up to 2.3 Mbps in both upstream and downstream directions. It offers an inexpensive means of providing T1/E1 service without the use of

repeaters, which T1 and E1 links need. It can operate over distances of up to 15000 feet. HDSL does not provide circuit switched voice service over the same line. However, it may be used to carry packetized voice. Since it requires two or three twisted pairs its deployment is difficult in areas where there is a shortage of twisted pairs. HDSL usually finds application in corporate environments.

HDSL2 is a newer version of HDSL that uses a single twisted pair line and provides symmetric data rates of 1.544 Mbps in both directions. It can operate over distances of up to 12000 feet. Since HDSL2 requires only a single twisted pair it is preferred in areas where there is a shortage of twisted pairs. It also finds application in small offices and homes. Like HDSL, it cannot carry data and circuit switched voice over the same line.

Symmetric DSL (SDSL) uses a single twisted pair to provide symmetric data rates of 128 Kbps to 2.3 Mbps in both directions. SDSL is actually a general term for a number of variations of DSL that provide symmetric data rates in the range mentioned above. SDSL usually operates at distances up to 14000 feet.

G.shdsl is standard version of SDSL that provides symmetric bandwidths between 192 Kbps and 2.3 Mbps in both directions. It enables operation over 20 to 30 percent longer distances than SDSL. It permits multi-vendor interoperability and compatibility with other DSL variants. The standard specifies the use of a single twisted pair for normal operation and two twisted pairs for operation over longer distances.

Very-high-bit-rate DSL (VDSL) provides a high downstream bandwidth that depends on the distances covered by the DSL link. It ranges from 52 Mbps (1000 feet) to 13 Mbps (4500 feet). Upstream bandwidths range from 1.5 to 2.3 Mbps. Since it operates over small distances it can be used in scenarios where an optical fiber connects the central office to the neighborhood. This is called a fiber-to-the-curb (FTTC) network. A neighborhood cabinet houses an optical network unit that serves as a distribution point close to the customer premises. VDSL runs from the neighborhood cabinet to the customer's office or home. VDSL can also be configured to operate in the symmetrical mode.

Integrated Services Digital Network DSL (IDSL) provides symmetric data rates of 128 Kbps over a single twisted pair. It can operate over distances of up to 18000 feet. IDSL is similar to Integrated Services Digital Network (ISDN) in that it can use the same modem and provides the same data rate. However, unlike ISDN, IDSL is an "always on" service and it does not support native, circuit switched voice. However packet voice can be carried over IDSL.

Rate Adaptive DSL (RADSL) is a variation of ADSL in which the data-rate is adjusted to according to the needs of the application, the length of the link and the effect of the environment on loop conditions. The RADSL modem automatically and dynamically adapts the data rates according to the line conditions. RADSL provides downstream bandwidths ranging from 600 Kbps to 7 Mbps and upstream bandwidths ranging from 128 Kbps to 1 Mbps.

Multirate DSL (MDSL) is a variation of SDSL that adapts its data rate based on the length of the link and other physical conditions. It provides symmetric bandwidths that range from 272 Kbps to 2.3 Mbps.

2.2 Voice over DSL

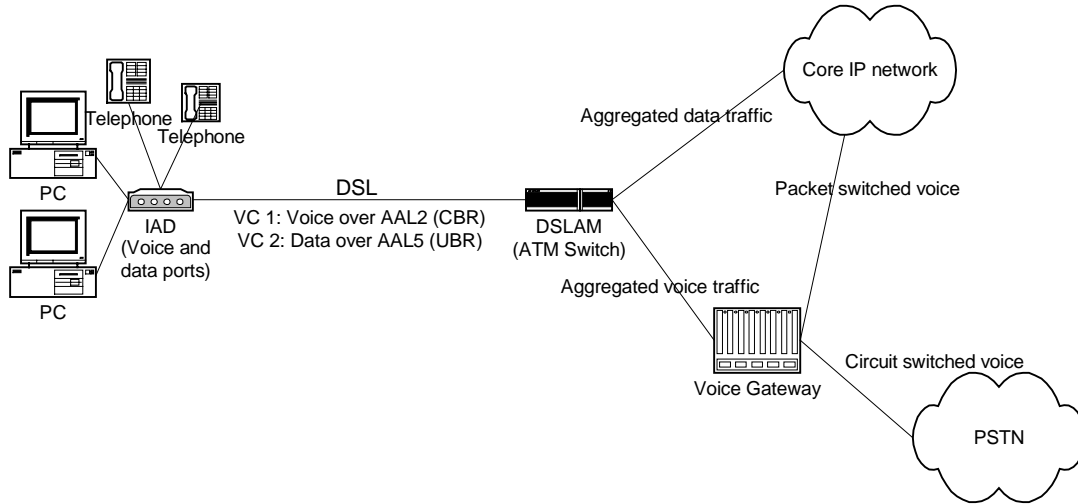


Figure 2.2: VoATM for DSL

The current practice is to use VoATM as the VoDSL technology ([Coppercom1], [CopperCom2], [CopperMount], [Lucent], [Alcatel]). Figure 2.2 represents VoATM deployment for DSL and Figure 2.3 shows the protocol stacks for this scenario. In the VoATM architecture, the CPE is an Integrated Access Device (IAD) that provides integrated voice and data access. The IAD consists of an Ethernet port for the customer's data traffic and one or more voice ports for the customer's voice traffic. Voice and data are carried over different virtual circuits (VCs) from the IAD to the DSLAM, which is basically an ATM switch. Voice is carried directly over ATM Adaptation Layer 2 (AAL2) using Constant Bit Rate (CBR) service. Data is carried over IP over ATM Adaptation Layer 5 (AAL5) using Unspecified Bit Rate (UBR) service. At the DSLAM, AAL2 voice traffic is switched to a voice gateway and AAL5 data traffic is switched towards the core IP network. The DSLAM aggregates voice VCs from different customers into a single voice VC towards the voice gateway. The voice gateway converts the voice packets into the appropriate format and sends them to their destination, depending on whether PSTN or IP telephony is used in the core. Similarly, it aggregates data VCs from different customers before sending it to the core IP network.

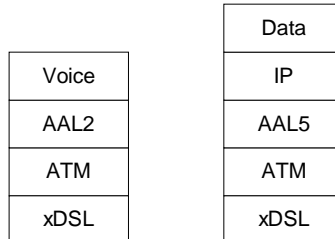
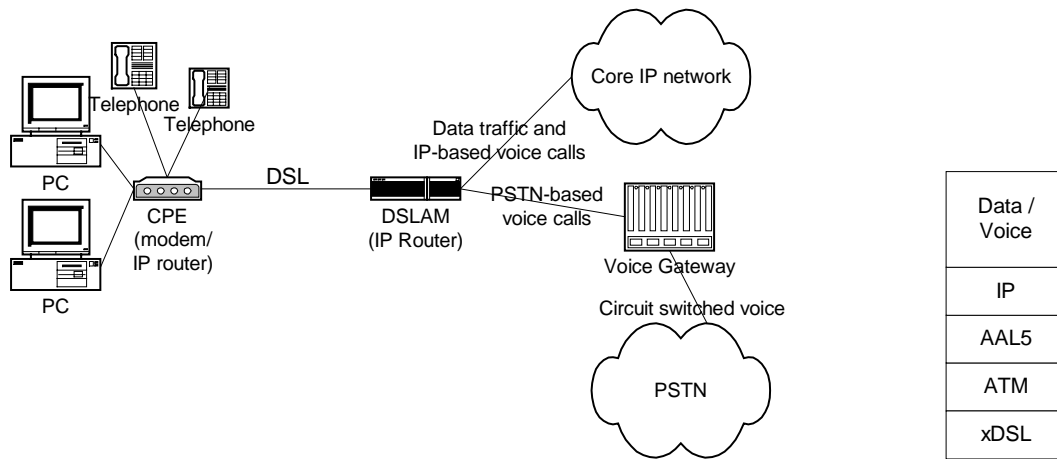


Figure 2.3: Protocol stacks for voice and data in the VoATM scenario

AAL2 imposes very low protocol overhead on voice packets and it also supports multiplexing of different voice flows over a single VC. For voice packets, AAL2 defines a header that contains an identifier indicating which flow it belongs to. The header also has fields for length and encoding scheme. Using ATM's signaling mechanism, each voice flow requests CBR QoS and specifies the peak data rate in its traffic contract. Any node along the path that cannot support the data rate specified in the traffic contract rejects the call. The voice flows that get admitted are guaranteed a portion of the bandwidth and hence they experience low delays across the network. Thus, VoATM with its low protocol overhead and built-in QoS support has been a strong contender to provide VoDSL.

Another technology for carrying voice traffic over DSL is Voice over IP (VoIP) ([CopperCom1], [CopperCom2], [TollBridge]). The CPE in this case is simply a DSL modem or router. Both voice and data traffic is carried over IP from the CPE to the DSLAM. The DSLAM is also an IP router, which routes data packets towards their core. Voice packets are also sent over the packet network towards the core if IP telephony is used for long-distance calls. On the other hand, if PSTN is used for long-distance telephony, then the DSLAM routes packets towards a voice gateway that converts packet voice into circuit-switched voice to be carried over the PSTN. The Layer 2 protocol may be ATM, Frame Relay or SLIP. In the VoIP scenario, the Layer 2 protocol treats voice and data packets alike and usually does not perform any QoS-related functions. Figure 2.4(a) represents VoATM deployment for DSL and Figure 2.4(b) shows the protocol stack for this scenario.



(a) Network Topology

(b) Protocol stack

Figure 2.4: VoIP for DSL

VoIP suffers from poor bandwidth efficiency. To reduce packetization delay and transmission delays to a minimum, it is best to keep the payload size of voice packets small. However the VoIP protocol stack has several layers, each which adds a header to the voice payload. These headers impose a significant protocol overhead on the small voice packets. Solutions such as header compression ([Jacobson1], [Casner], [Caputo]),

and IP trunking ([El-Khatib], [CopperCom1]) (also known as RTP multiplexing) help alleviate the bandwidth problem.

Competition from data traffic poses yet another problem for VoIP. During congestion, small voice packets could get queued behind several large data packets at the routers. They would then experience significant end-to-end delays across the network as they wait for long periods in the queue. Prioritizing voice packets and placing them in a separate queue is a possible solution to this problem. However, this is by no means a complete solution. Even with prioritization of voice packets, a voice packet that arrives while the data packet is being transmitted has to wait for the transmission to be completed. Large data packets have long transmission times over slow links. The waiting time for voice packets could be anywhere between zero and the transmission time for the entire packet. Thus, voice traffic may experience significant variation in end-to-end delay (sometimes referred to as jitter). Fragmentation of large IP data packets is a possible solution for this problem.

VoATM is currently the preferred method for VoDSL, due to its superior performance. However, as discussed in Section 1.1 of Chapter 1, the deployment of VoIP would offer some clear advantages, due to the ubiquity of IP. It would pave the way for end-to-end IP telephony. Packet voice can be carried inexpensively over data networks, with a uniform packet format across the Internet. These advantages strongly motivate research in providing IP QoS for real-time voice traffic over the Internet.

2.3 Voice over Packet Networks

Voice signals are analog waves that must be converted to digital format before being transported over a data network. This is done using a voice encoder. Circuit switched networks establish a connection between two end-points and send digitized voice over this connection. Packet switched networks encapsulate digitized voice in packets with headers that contain the destination address. The packets are then forwarded hop-by-hop over the packet network towards the destination. The destination extracts the digitized voice from the packet and uses a decoder to convert the digitized voice back into analog form.

The stream of digitized voice is usually treated as a sequence of constant sized frames. Each frame is thus a collection of bits and its size is expressed in terms of the duration of speech that its bits encode. Packetization delay is the delay incurred in accumulating voice frames so that there are enough frames to make a big enough voice packet. There is a tradeoff between bandwidth efficiency and packetization delay. If the packet size is too small, the bandwidth efficiency is reduced because of the protocol headers added to each packet. If the packet size is too large, the packetization delay may become unacceptably high.

The voice stream can be encoded in ways that produce voice streams with low bit rates, thereby reducing the bandwidth requirement for transporting the voice stream over a data

network. These techniques include Adaptive Differential Pulse Code Modulation (ADPCM) and Code Excited Linear Predictor (CELP). Most standardized voice encoding schemes use the above-mentioned techniques or their variants, as well as silence suppression.

Table 2.1 lists the bit rates and frame sizes of voice streams generated by the standard encoders.

Encoder	Bit Rate (Kbps)	Frame Size (ms)
G.711	64	0.125
G.726	32	0.125
G.729	8	10
G.723	5.3	30

Table 2.1: Bit rates of some standard voice encoders

About 60 % of a voice conversation usually contains periods of silence. Long periods of silence occur when one of the speakers is listening to the other. Even when a person is speaking, there are short periods of silence in between sentences and words. Silence suppression is a technique in which the periods of silence are not transmitted to the other party in the conversation. This can save a significant amount of bandwidth. Silence suppression can be achieved by digital signal processors (DSP), using a technique called voice activity detection (VAD). The principle behind VAD is to monitor track changes in the power and frequency of a voice signal. This enables the DSP to detect when a speaker has started speaking and when he or she has stopped.

The end-to-end delay experienced by voice as it travels from source to destination has a significant effect on voice quality. Most users find it inconvenient to carry out a conversation if the end-to-end delay is greater than 150 ms. Delay is caused by several factors -:

- Digitization of analog voice
- Compression of digital voice bit stream
- Packetization delay
- Delay introduced by the operating system in the end-host as it shares processing time between several applications
- Queuing delay and transmission delay at the end-host and at the routers/switches along the path from source to destination
- Delay introduced by the jitter buffer (described below) at the destination
- Depacketization at the destination
- Decompression of digital voice
- Conversion of digital voice to analog voice

Another factor influencing voice quality is the variation in delay (also called jitter). The source may send out voice packets in a uniform manner, but they may experience different delays across the network. Thus, the inter-arrival times of voice packets at the destination is variable, rather than constant. This delay variation results in unevenness in

the reconstructed speech if the voice packets are delivered to the user as soon as they arrive. This problem can be solved by storing the arriving voice packets in a buffer at the destination and playing them out at a constant rate. Packets arriving early are held in the buffer long enough to make their delays equal to a constant value. Packets arriving with delays greater than this constant value are considered lost. This value must be chosen keeping in mind that too low a value will result in many lost packets, and too high a value will cause too much end-to-end delay.

The fraction of voice packets lost during transit also affects the voice quality. Retransmission of voice packets is not feasible for real-time applications like voice, since they have very tight delay-bounds. As mentioned earlier, a packet that arrives too late is also considered lost. Several measures have been proposed to alleviate the effects of packet loss. The simplest solution is to replay the previous packet. Some decoders can also guess the missing signal based on the previous packets. Another option is to send a low bit-rate version of each packet in the subsequent packet, so that the receiver can obtain the sample from the subsequent packet. However, if several packets in succession are lost, then the decoder simply inserts silence periods.

2.4 VoATM Concepts

2.4.1 ATM Basics

ATM is a connection-oriented, packet-switched technology that transports data in fixed length packets called cells. Packets are transported over a VC that is established by signaling between the connection end-points. The ATM protocol stack consists of three layers – the physical layer, the ATM layer and the ATM Adaptation Layer (AAL).

The physical layer deals with the physical medium issues such as voltages, bit timing and framing. It is divided into two sublayers – the Physical Medium Dependent Sublayer (PMD) and the Transmission Convergence (TC) sublayer. The PMD sublayer is dependent on the actual physical medium. Its responsibility is putting bits on wire and taking bits off the wire. It thus handles the bit timing. At the sender, the TC sublayer converts an ATM cell into a stream of bits and sends it to the PMD sublayer. At the receiver, it converts the pure bit stream arriving from the PMD sublayer into ATM cells for the ATM layer. It also performs header error correction (HEC) for the ATM cells.

The ATM layer deals with cells. It defines the ATM cell format and assigns values to the cell header fields. An ATM cell is 53 bytes in size. The first 5 bytes form the header and the remaining 48 bytes form the payload. The ATM layer is also responsible for finding a route from source to destination, setup and teardown of VCs and switching of cells.

AAL allows applications and other protocols to run over ATM. It consists of two sublayers – The Convergence Sublayer (CS) and the Segmentation and Reassembly (SAR) sublayer. The CS provides the interface to the higher layer protocol. CS encapsulates the higher layer Protocol Data Unit (PDU) into a Common Part

Convergence Sublayer PDU (CPCS-PDU). This may be done by adding a header and/or a trailer. The CPCS PDU is usually too large to fit as payload of a single ATM cell. At the sender, the SAR segments the CPCS PDU into cells and sends them to the ATM layer for transmission. Similarly, at the receiver, SAR reassembles the ATM cells into the CPCS PDU. The stand AAL types are AAL1, AAL2, AAL3/4 and AAL5. Of these AAL2 and AAL5 are of relevance to our study.

AAL2 is used for connections that require real-time, variable bit rate services such as voice traffic with silence suppression. A voice packet is usually much smaller than the maximum payload size of an ATM cell, so it is inefficient to have an ATM cell carry just one voice packet. Carrying several voice packets from the same voice call in an ATM cell would increase the packetization delay. AAL2 facilitates the sharing of a single ATM cell among several voice calls. Each voice packet has a 3-byte AAL2 header that has the format shown in Figure 2.5.

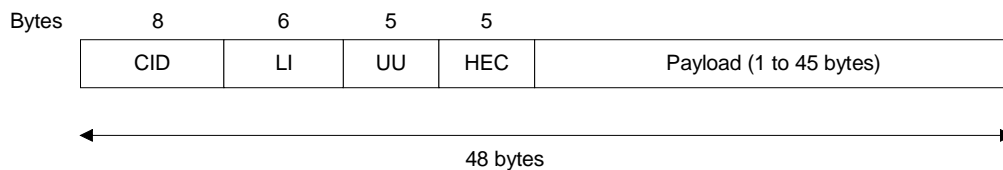


Figure 2.5: AAL2 PDU format

The Channel Identifier (CID) field identifies the channel to which the voice packet belongs. This enables voice packets belonging to different calls to be mapped on to a single ATM VC. Thus, a single ATM cell can be used to carry voice packets belonging to different calls. This eliminates the disadvantage of increased packetization delay and also improves bandwidth utilization. The Length Indicator (LI) field indicates the length of the payload and the Header Error Control (HEC) field is a checksum over the header. The User-to-User Indication (UUI) field is for use by the higher layers.

AAL5 is used for connections that carry data, often over some other higher layer protocol such as IP. The AAL5 PDU is shown in Figure 2.6



Figure 2.6: AAL5 PDU format

The CPCS adds an 8-byte trailer to the higher layer PDU. The User-to-User (UU) field is not used by AAL. Instead it is available for use by the higher layers to identify the user payload. The Common Part Indicator (CPI) field has not yet been defined in the standards. The Length field contains the length of the payload. A pad field is used to

make the AAL5 PDU size a multiple of 48 bytes, so that the PDU fits exactly into an integral number of ATM cells. The CRC is a checksum over the PDU.

Figure 2.7 shows the ATM protocol stack.

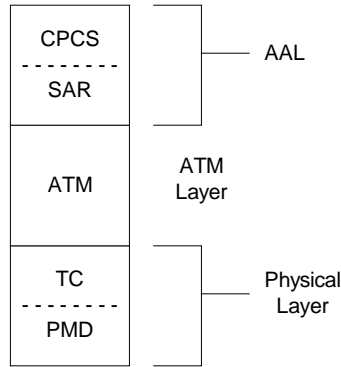


Figure 2.7: ATM Protocol Stack

2.4.2 ATM Traffic Parameters

The characteristics of an ATM traffic source are described by a set of traffic parameters, collectively known as the traffic descriptor. The important traffic parameters are –

1. Peak Cell Rate (PCR) – This is the maximum rate at which the source sends cells into the ATM network.
2. Sustained Cell Rate (SCR) – This is the average rate at which the sender sends cells into the ATM network over a long period of time.
3. Maximum Burst Size (MBS) – This is the maximum number of cells the sender can send at the PCR when the SCR has been specified in the traffic descriptor.
4. Minimum Cell Rate (MCR) – This is the cell rate that will always be guaranteed to the sender over the ATM network.

2.4.3 ATM QoS Attributes

The ATM QoS attributes measure the performance of the network in transporting cells from the source to the destination. We discuss three important QoS attributes below -

1. Cell Transfer Delay (CTD) – This is the time taken for a cell to be transported from source to destination.
2. Cell Delay Variation (CDV) – This is the variation in the inter-arrival times of the cells (of a particular connection) at their destination.
3. Cell Loss Ratio (CLR) – This is the fraction of the transmitted cells that are not delivered to the destination, or are delivered too late to be of use.

Figure 2.8 represents the relationship between these QoS characteristics.

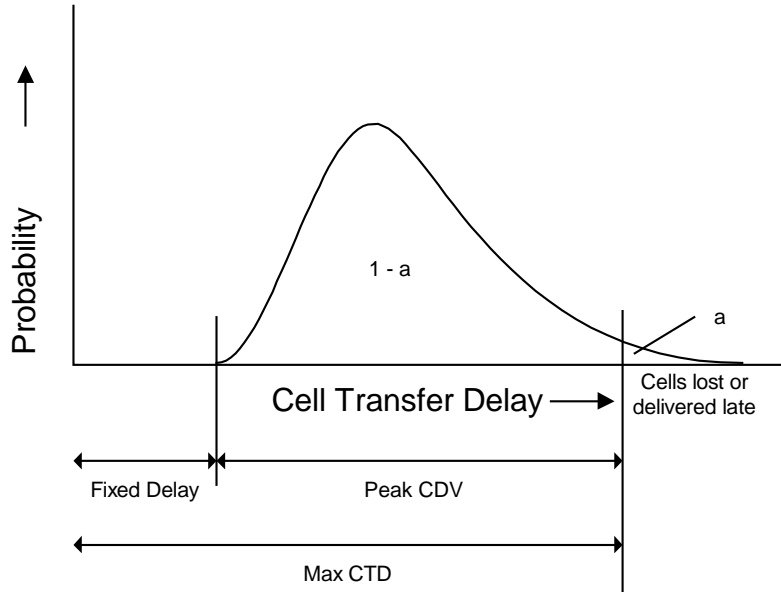


Figure 2.8: Probability density function for CTD (Adapted from [Tanenbaum])

2.4.4 ATM Service Classes

The ATM Forum has specified five service classes for ATM traffic –

1. **Constant Bit Rate (CBR)** – This service is used by connections that require a constant bandwidth for the entire duration of the connection. The required bandwidth is specified by the PCR of the traffic source. If the connection is accepted, the source can be guaranteed the requested level of service as long as it transmits cells at a rate less than or equal to the PCR. The network is expected to keep end-to-end delay and delay variation to a minimum. No guarantees are provided for traffic transmitted above the PCR.
2. **Real-Time Variable Bit Rate (rt-VBR)** – This service is used by connections that have strict requirements of delay and delay variation, but transmit at variable data rates. rt-VBR service is requested by specifying the PCR, SCR and the MBS of the traffic source. If connection is accepted and the traffic conforms to the specifications included in the traffic descriptor, then the connection receives a guaranteed level of service from the network. No guarantees are provided for traffic that does not conform to the specifications.
3. **Non-Real-Time Variable Bit Rate (nrt-VBR)** – This service is used by connections that transmit a variable data rates, but do not have strict delay and delay variation requirements. The network is expected to keep the cell loss rate of the connection low. nrt-VBR service is requested by specifying the PCR, SCR and MBS. No guarantee on the cell loss rate is provided to non-conforming traffic.
4. **Unspecified Bit Rate (UBR)** – This service is used by connections that have no delay, delay variation, loss rate and bandwidth requirements. The network provides no service guarantees for this class of traffic.

5. Available Bit Rate (ABR) – This service is used by connections that have no strict delay or delay variation requirements and can adapt their transmission rates to the changing network bandwidth. The connection is requested by specifying the MCR and the PCR. If accepted, the source can transmit at a rate at least equal to the MCR and expect a low CLR from the network. If more bandwidth becomes available, then the network can signal this to the endpoints, which can then increase their transmission rates. Similarly, if the bandwidth availability decreases, the network signals the endpoints to reduce the transmission rate. The transmission rate must never exceed the PCR and must never fall below the MCR.

2.4.5 ATM QoS mechanisms

ATM QoS is defined at the time of VC establishment by means of a signaling messages exchanged during this process. Once the VC has been successfully established, its QoS characteristics are guaranteed. The guarantees are provided using the following mechanisms –

- Connection Admission Control (CAC) to decide whether a not the QoS requested by a new connection can be supported by the network. The connection is admitted if there are sufficient resources to guarantee the level of service. Otherwise, it is rejected.
- Usage Parameter Control (UPC) to ensure that the traffic injected into the network by a particular connection does not exceed the limits specified in the traffic contract negotiated during connection setup.
- Queuing and scheduling mechanisms at each node to ensure that each connection gets its share of bandwidth.

The CAC function obtains the following information from the traffic contract –

- (i) The values of the traffic attributes specified in the traffic descriptor – i.e. PCR, SCR, MBS, and MCR etc.
- (ii) The requested QoS class and values of the requested QoS attributes in either direction. The QoS attributes include maximum CTD, maximum CDV, and CLR etc.
- (iii) The requested conformance definition for traffic of that connection. This includes PCR and SCR values used to determine whether or not a cell is conforming.

Using this information, the CAC mechanism determines whether to accept the connection. It also determines the traffic parameters that would be needed for performing UPC. CAC also allocates resources for the connection at each node along the route.

The UPC function is responsible for monitoring a connection to determine whether the traffic injected into the network is in compliance with the traffic contract. Appropriate actions are taken based on whether the traffic is compliant or not. UPC must ensure that non-compliant traffic of a connection does not affect the QoS provided to other connections that comply with the traffic contract. UPC also checks if the VPI and VCI values of the incoming traffic are valid.

Bandwidth allocation is accomplished by setting up queues within an ATM node and servicing these queues in accordance with a scheduling algorithm. Assigning priorities to the queues can provide different levels of service. A scheduling algorithm that allocates bandwidth to the queues based on priority is employed. The ATM specifications do not define the scheduling algorithm or how these queues are set up.

2.5 VoIP Concepts

2.5.1 Protocols

IP is a connectionless, packet-switched technology for carrying data across the Internet. It forms the network layer of the Internet and its primary function is to route packets from their source to their destination. For each packet, IP (Version 4, IPv4) adds a header with the format shown in Figure 2.9

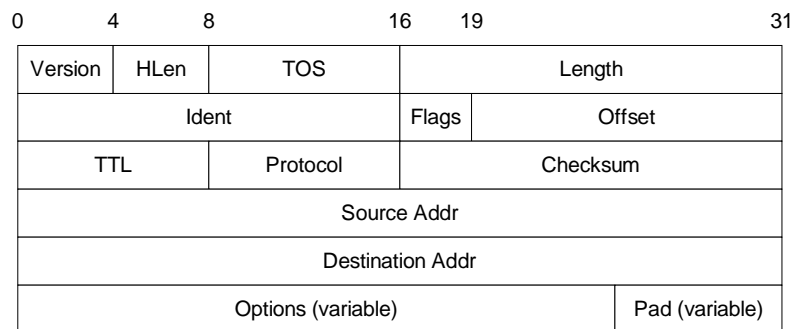


Figure 2.9: IPv4 Packet Header (Adapted from [Peterson])

The Version field specifies the version number of IP. The HLen field specifies the length of the header in 32 bit words. Usually, the header length is 20 bytes. In cases where the Options field is used, it may be greater. The Type Of Service (TOS) field allows packets to be classified and treated according to the class to which they belong. The Length field specifies the size of the payload in bytes. The Time To Live (TTL) field is actually a counter that counts the number of hops traversed by the packet. The Protocol field identifies the higher layer protocol whose packet is encapsulated within this packet. The Checksum field provides a checksum over the header. Fields containing the source and destination addresses of the packet are also included in the header. The Options field is used to provide some additional options and the Pad field is used to make the header size equal to an integral number of 32-bit words.

User Datagram Protocol (UDP) is a transport layer protocol that runs on top of IP. It provides for process-to-process communication. Since there could be several processes running on a host, UDP enables the processes to share the network by multiplexing and demultiplexing of packets belonging to different process over the network interface. This is achieved by means of port numbers. Each process uses a unique port number within the host and UDP includes that port number in the packets that belong to that process. UDP

uses a checksum to ensure the correctness of the message, but does not implement flow control, congestion control, reliable data transfer or in-order delivery of packets. It is more suited for real-time applications that are tolerant to losses but have stringent delay requirements.

Transmission Control Protocol (TCP) is a more versatile transport layer protocol that finds widespread use in the Internet. TCP also enables multiplexing and ensures correctness the same way as UDP does. In addition, it also provides other features such as flow control, congestion control, and reliable, in-order delivery of packets from sender to receiver. It is suited for non real-time applications that require guaranteed delivery of data. It is not suitable for real-time applications.

The Real-time Transport Protocol (RTP) is an application layer protocol for multimedia traffic. RTP is actually used with another protocol called Real-time Transport Control Protocol (RTCP). RTP and RTCP both run over UDP. RTP is used to carry multimedia data, and RTCP is used to carry control information associated with the data.

Session Initiation Protocol (SIP) ([Schulzrinne]) is a signaling protocol that establishes a VoIP call between the participating end-hosts. Some of the important SIP messages are –

- INVITE – Sent by the calling party to the called party to initiate a call.
- ACK – Response sent by the called party to the calling party confirming participation in the call.
- BYE – Sent by either the calling or the call party to indicate that it wishes to end the call.
- CANCEL – Used to cancel a request to establish a call.
- OPTIONS – Used to request information about the capabilities of the called party.

2.5.2 Protocol Overhead in VoIP

The VoIP protocol stack consists of RTP, UDP and IP, in addition to the data link protocol. RTP, UDP and IP add headers of length 12, 8 and 20 bytes respectively to the voice payload. Thus, a voice packet has 40 bytes of header information even before it is passed to the data link layer. As mentioned earlier, these headers result in a significant protocol overhead being added to the voice packets, which must be kept small in size in order to minimize the packetization delay and the transmission delay over slow links.

RTP header compression ([Casner]) is a technique used to reduce the size of the RTP header in the voice packets. This scheme takes advantage of the fact that for a given RTP session, some RTP header fields change at a constant rate and the others are constant. Also, the source and destination UDP port numbers and the source and destination IP addresses are constant for a given RTP session. If RTP header compression is enabled at both the sending router and the receiving router, then the bandwidth efficiency can be greatly improved by eliminating the transmission of some RTP header fields. For the fields that change at a constant rate, the receiving router needs to store just the original value and the rate of change. From this information, it can compute the header fields of

the subsequent packets. This eliminates the need to transmit these fields in each packet. For the RTP, UDP and IP fields that have constant values, the receiving router can use a hash table to store these values. The hashing function will yield a small integer that can be transmitted instead of all those fields. Thus, RTP header compression results in significant bandwidth savings.

Another technique that can be employed to mitigate the bandwidth overhead is IP trunking, also called RTP multiplexing ([El-Khatib]). This can be employed when several voice connections share a common path. In such a situation, the RTP payload can carry multiple voice packets belonging to different connections, with each packet being preceded by a small header that identifies the length of the packet and the connection to which it belongs. Thus, when there is more than one active call, the overhead of the IP, UDP and RTP header is shared by packets belonging to different voice calls. This helps improve bandwidth efficiency.

2.5.3 QoS for VoIP

IP networks were originally designed to provide only best-effort service. However, the best effort approach does not satisfy the requirements of many applications, which require a guaranteed level of service from the network. This led to significant amount of research aimed at enabling QoS support in IP networks. Approaches to QoS can be classified as –

- Fine-grained approaches, which provide QoS to individual flows. The Integrated Services (IntServ) architecture ([Braden1]) proposed by the Internet Engineering Task Force (IETF) is based on this approach.
- Coarse-grained approaches, which provide QoS to aggregates of flows. The Differentiated Services (DiffServ) architecture ([Blake]) proposed by IETF adopts this approach.

2.5.3.1 Integrated Services

An IntServ flow is made up of packets that require a common QoS and that are transmitted by a common source towards one or more destinations. IntServ specifies two main service classes –

- (i) Guaranteed Service (GS) ([Shenker]) – This is meant for real-time applications that have strict bandwidth and delay requirements.
- (ii) Controlled Load (CL) ([Wroclawski]) – This is meant for non real-time applications that require performance similar to what would be obtained from a lightly loaded best-effort network, irrespective of what the actual load on the network is. CL service emulates a lightly loaded network.

In the IntServ architecture, applications use the Resource reSerVation Protocol (RSVP) ([Braden2]) as the signaling mechanism to reserve resources at each router between the source and the destination. At each router, the application's request to reserve resources is accepted or rejected, based on resource availability. Once all the routers along the path from source to destination accept the request, the application is guaranteed the level of service it requests. The IntServ model requires that the routers process per-flow

reservation messages, maintain per-flow state and perform per-flow classification, queuing and scheduling.

This approach works well when the number of flows in the network is small. However, there are severe scalability problems with the IntServ, since the complexity of per-flow operations at each router increases as a function of the number of flows in the network. Hence, IntServ is considered unsuitable for large-scale deployment in the Internet core.

2.5.3.2 Differentiated Services

The DiffServ architecture for IP QoS proposes the aggregation of traffic flows into different classes of traffic, also called behavior aggregates (BA). The BA of a packet can be identified from the value of the DiffServ Code Point (DSCP) carried in its header. The DiffServ architecture redefines the IPv4 TOS octet as the DiffServ field, as shown in Figure 2.10.

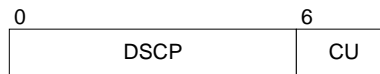


Figure 2.10: The DiffServ field

Six bits of the DiffServ field make up the DSCP and two bits are currently unused (CU). DiffServ-capable routers handle packets based on the BA to which the packet belongs. The externally observable forwarding behavior of a DiffServ router is called Per Hop Behavior (PHB). The PHB to be applied to a packet is determined by the DSCP value in its header.

A DiffServ domain is a network consisting of a contiguous set of DiffServ-capable nodes, all of which use a particular set of DiffServ mechanisms to provide QoS within that network. The DiffServ domain has a set of well-defined rules for classifying and conditioning ingress traffic and supports a particular set of PHBs. The nodes at the edge of the DiffServ domain are called DiffServ boundary nodes - they connect the DiffServ domain to other DiffServ or non-DiffServ domains. DiffServ interior nodes connect only to other nodes within the same DiffServ domain. Boundary nodes act as ingress nodes for traffic entering the DiffServ domain and as egress nodes for traffic leaving the DiffServ domain. The DiffServ architecture pushes the complexity towards the boundary nodes, while the interior nodes are kept simple. The boundary nodes perform complex packet classification and traffic conditioning functions. The interior nodes simply forward packets, based on the marking of the packet.

A packet classifier classifies packets based on the value of one or more header fields. The DiffServ architecture defines two types of classifiers. The Behavior Aggregate classifier, which classifies packets based on the DSCP only, and the Multi-Field classifier, which classifies packets based on the values of a combination of header fields such as the DiffServ field, source IP address, destination IP address, source port, destination port and transport protocol.

A traffic conditioner usually consists of the following components- meter, marker, shaper and dropper. A meter compares a traffic stream against a traffic profile. A marker sets the value of the TOS octet. Marking may be done by the operating system of the traffic source (host-marking) or by a router in the source domain (router marking). A shaper delays packets belonging to a traffic stream, in order to make the stream comply with a traffic profile. A dropper discards packets that are out-of-profile in a traffic stream.

Each router allocates buffer and bandwidth resources to different classes of traffic based on the PHB. PHBs are implemented using buffer management and packet scheduling mechanisms. The PHB describes only the externally observable treatment that a particular class of traffic receives at a node. It does not describe the implementation used for this purpose. Two of the popular PHBs are Expedited Forwarding (EF) ([Jacobson2]) and Assured Forwarding (AF) ([Heinanen]). Packets receiving EF service are forwarded at least as fast as the rate at which they arrive. The rate at which EF traffic arrives into the network is limited through policing or shaping. The router is configured to service the EF packets faster than the maximum rate at which they can arrive. EF service thus ensures low packet loss rates, low latencies and low jitter. On the other hand, AF PHB service guarantees are relative rather than absolute. AF specifies four service classes and packets are queued in one of four queues based on their service class. The packet scheduler allocates a share of the available bandwidth to each class. Packets within a service class are assigned one of the three drop-precedence levels.

The DiffServ approach is scalable and hence suitable for deployment in core networks, where the number of flows is large. Since the functions of a core router are simple under this architecture, these routers operate faster. Also, they do not need to maintain per-flow state information. On the other hand, QoS guarantees for individual flows are not easily achieved with this approach.

2.6 Chapter Summary

This chapter was meant to bring the reader up to speed on some basic concepts related to DSL, ATM, IP, QoS and packet-voice. The first section described the DSL technology. The components of a DSL network and the different variants of DSL were discussed. In the next section, we discussed technologies for transporting voice over DSL. We touched upon the VoATM and VoIP approaches and highlighted the relative merits and demerits of both. Then, we included a section that described the fundamentals of carrying voice over packet networks. It discussed how voice is packetized and what factors affect the quality of voice calls routed over packet networks. The next section covered the principles of ATM and ATM QoS that are relevant to understanding the VoATM technology. The last section of this chapter was devoted to principles underlying VoIP. It described the various protocols used for VoIP and the possible approaches for providing QoS to voice traffic in an IP network.

Chapter 3: An IP QoS Architecture for VoDSL

In this chapter, we describe our IP QoS architecture for voice and data convergence over DSL lines. Our goal is to ensure low delay and jitter for voice packets as they travel across the access network, from the end-hosts in the CPE to the DSLAM in the CO. In particular, we are concerned with bandwidth management techniques for the upstream link from the CPE to the DSLAM, since this link is often the bottleneck in the access network, especially in the case of ADSL. The mechanisms that we propose in this chapter are to be implemented on the CPE.

3.1 Service Classes

Our architecture is designed to support four service classes –

1. **Premium Voice** – This service class is meant for voice calls that require excellent QoS. Each customer is guaranteed a certain amount of bandwidth for premium voice calls. A percentage of the upstream bandwidth is reserved for premium voice calls. The number of premium calls that can be supported by that bandwidth would depend on the codec used by the customer. Admission control is strictly enforced for calls belonging to this category. Premium calls are not allowed to borrow unused bandwidth that is reserved for other classes of service. The reason is that when the load offered by traffic belonging to other classes of service increases, each class must get the share of bandwidth reserved for it. Any bandwidth borrowed by the premium class must be immediately returned at this point. Since a portion of the bandwidth being used by premium calls is taken away from them, the quality of the premium calls in progress will suffer. This is referred to as the stolen bandwidth problem and is discussed in [Breslau] and [Bernet]. The moral of the story is that premium traffic load must never be allowed to exceed the limit that can be supported by the bandwidth reserved by it.
2. **Regular Voice** – This service class is meant for voice calls that do not get admitted to the premium category. If we were to have only the premium category for voice calls, then all voice calls would get rejected if all the bandwidth reserved for the premium class were in use. Even if the bandwidth reserved for other classes of traffic were unused, we would be unable to take advantage of it, since premium traffic cannot borrow bandwidth from other classes. Such a strict fragmentation of resources leads to inefficient utilization. To address this issue, we introduce the concept of regular voice calls. Voice calls that cannot be admitted as premium can still be allowed to go through as regular calls. However, the regular calls are not given any QoS guarantees. Depending on the implementation, a portion of the upstream bandwidth may be reserved for them, or they may just share the bandwidth along with best-effort traffic (described next). There is no admission control for the regular voice traffic. Under conditions of light load from best-effort traffic, the regular calls will receive acceptable performance. In the case of ADSL, the typical customer runs applications such as

web browsing, email, remote login and streaming multimedia, which offer low load on the upstream link. Besides, the downstream bandwidth is usually quite high, so there is a reasonable chance of a regular call experiencing acceptable quality in a lightly loaded network.

3. ACK class – This service class is meant for TCP ACKs traveling in the upstream direction. In asymmetric links, the downstream throughput of TCP connections is limited by the rate at which the ACKs arrive over the upstream link. Thus, to get a good performance from TCP applications, it is necessary to guarantee a certain portion of the upstream bandwidth for ACKs. A detailed discussion of these concepts can be found in [Balakrishnan], [Kalampoukas] and [Phanse]. A QoS architecture for voice and data convergence in ADSL would benefit greatly from inclusion of the ACK service class. Since this research is primarily focused on voice traffic, we do not discuss these concepts further.
4. Best Effort – This service category is for non real-time traffic. Applications such as web browsing, email, FTP, remote login, database access etc fall under this category. In our architecture we reserve a portion of bandwidth for Best Effort traffic in order to prevent starvation of these flows. In addition to the reserved bandwidth, these applications are allowed to use any additional bandwidth that is left unused by the premium and regular voice traffic.

3.2 Queuing and Scheduling

Figure 3.1 represents queuing and scheduling at the interface of the CPE that connects to the DSLAM via ADSL.

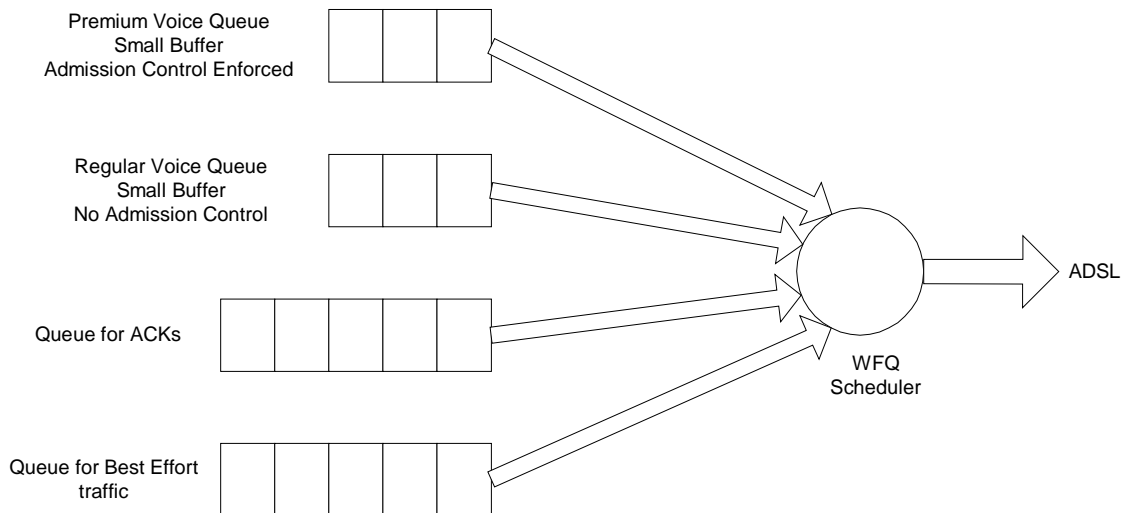


Figure 3.1: Queuing and Scheduling at the CPE

As shown in the figure, we have four queues, one for each of the service classes discussed in Section 3.1. Since real time applications such as voice are inelastic with respect to delay requirements, the buffer sizes for the premium and regular voice queues are kept small. For such applications, a packet that arrives late is considered lost. So a packet with a large queuing delay might as well be discarded and hence we keep small buffers for these. The arrival of packets to the premium voice queue is controlled by an admission control mechanism described later in this chapter.

The weighted fair queuing (WFQ) ([Demers]) algorithm is used to service these queues. This algorithm allows assignment of weights to the different queues and divides the link bandwidth among these queues in ratio of the weights. This algorithm tries to imitate the behavior of time-division multiplexing system that multiplexes data from multiple queues over a single link. Ideally, this would be done by servicing the queues in a round-robin fashion, with the number of bits transmitted from each queue being in the ratio of the assigned weights. However, in packet networks, transmission occurs on a packet-by-packet basis, rather than a bit-by-bit basis. Hence the WFQ algorithm makes an approximation by scheduling packets in such a way that the order in which they are received at the other end of the link would be the same as that in an equivalent TDM system. This approximation may result in short-term inaccuracies in dividing the bandwidth, but the long-term throughput for all the queues would be in the ratio of the assigned weights.

3.3 Fragmentation

The scheduling mechanisms described above are able to guarantee a portion of the upstream link bandwidth to each class. However, as we mentioned earlier, transmission occurs on a packet-by-packet basis. This implies that once a packet has begun transmission, it cannot be preempted by another packet. The transmission of the entire packet must be completed before the transmission of the next packet can begin. This can pose a problem for real-time traffic on low-speed links. Low-speed links take a long time to transmit big data packets, e.g. a 1500 byte packet takes around 90 ms to be transmitted over a 128 Kbps link. A voice packet that arrives just after a large data packet has begun transmission will have to wait until the link has finished transmitting the data packet. This delay can be unacceptable for real-time applications. Also note that depending on when they arrive, the waiting time for voice packets can vary between zero and the time taken to transmit the largest possible packet. Hence, the delay variation (or jitter) of the voice packets is also increased due to large data packets.

The simplest solution is to fragment large data packets. This can be done at the IP layer by reducing the Maximum Transmission Unit of the interface over which the packets need to be sent. The solution is effective enough in reducing the delay and jitter of voice packets, but there are some undesirable side effects. The IP header is replicated in each of the fragments and this increases the protocol overhead. Also, fragmentation at the IP layer has an end-to-end effect, because after the packet is fragmented, it will be reassembled only at the destination. Another disadvantage is that applications that use

PathMTU Discovery ([Mogul], [McCann]) will not send packets that are larger than the MTU. Hence, if the MTU is made very small, then application packets will be able to carry hardly any payload.

A better solution for this problem is obtained by using the Point-to-Point Protocol (PPP) ([Simpson]). PPP is an auto-configuring protocol that is used for exchanging packets over a serial, full duplex, point-to-point link. In our case, PPP can be used over the ADSL line connecting the CPE and the DSLAM. Multilink-PPP (MP) ([Sklower]) is a PPP variant that fragments PPP packets before sending them over the link. [Caputo], [Armitage] suggest a solution based on MP fragmentation to reduce the delay and jitter of voice packets.

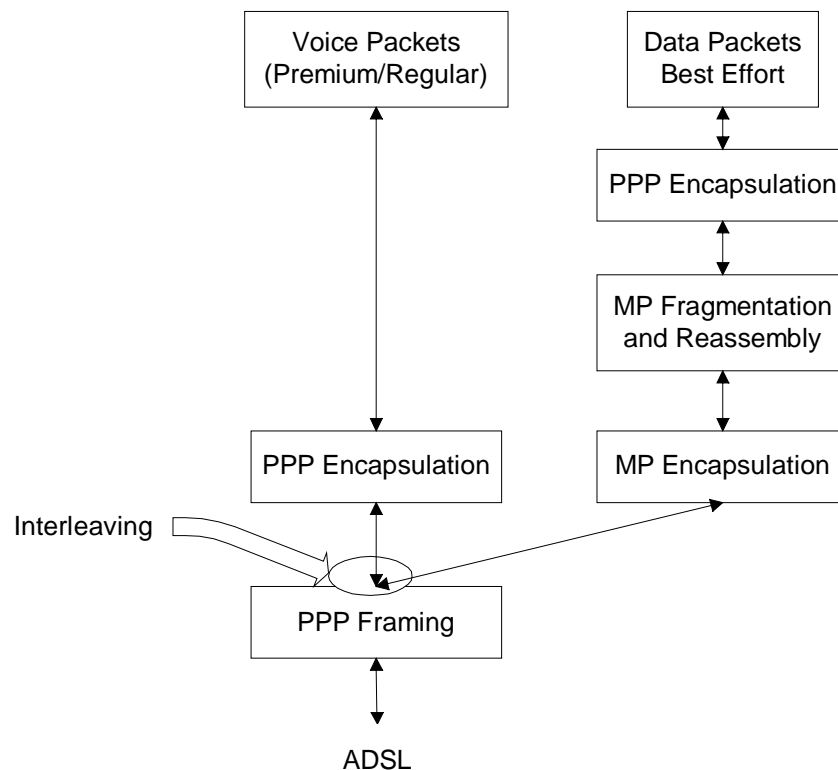


Figure 3.2: MP based Fragmentation and Interleaving (Adapted from [Armitage])

The idea is to have the upstream link shared by ‘IP-over-MP’ and ‘IP-over-PPP’ running simultaneously. The low priority data packets (belonging to the best-effort class) are forwarded using ‘IP-over-MP’ and the high priority voice packets are forwarded over ‘IP-over-PPP’. This is shown in Figure 3.2. Since MP forces fragmentation of the packets, the transmission of voice packets could be interleaved between the transmissions of the MP fragments for a single data packet. This interleaving of voice packets in between data packets eliminates the delay and jitter problem described earlier and ensures that voice packets are serviced in a timely manner. Since fragmentation is done at the data link layer, the IP header is not replicated in each packet. Also, the fragmentation does not have an end-to-end effect. The fragments are reassembled at the other end of the link. We incorporate Multi-link PPP based fragmentation in our QoS architecture.

3.4 Admission Control by Implicit Signaling (ACIS)

As part of our QoS architecture, we have developed an admission control mechanism based on an implicit signaling protocol. We call our mechanism Admission Control by Implicit Signaling (ACIS). ACIS is implemented in the CPE to limit the number of voice calls that are admitted to the premium category. This mechanism takes advantage of signaling done at the application layer of the end-hosts to establish a voice call. Application layer signaling is done using protocols such as the Session Initiation Protocol (SIP) or H.323. In this document we use SIP as an example to explain our idea. The same principles can be applied to other signaling protocols as well. The ACIS mechanism does not reject any voice calls as such. Instead, it admits only a limited number of voice calls as premium voice calls and admits the rest as regular voice calls. In this manner, we prevent the arrival rate from exceeding the service rate for the premium class of traffic. This guarantees low ETE delays for premium calls. On the other hand, there is no limit to the number of voice calls that can be admitted as regular calls. Hence, these calls may receive satisfactory service at light loads, but the quality degrades as the load increases.

3.4.1 Encoding Application Layer Signaling Information

The key idea in ACIS is to map the application layer signaling protocol on to the IP header, so that the CPE can infer the setup and teardown of voice connections. Some connection information can also be encoded in the IP header. For the working of ACIS, the IP header must have the additional fields shown in Figure 2.3. We call these fields collectively as the ACIS fields.



Figure 3.3: Additional IP header fields for ACIS mechanism

The Packet Type (PT) field indicates whether the packet contains an application layer signaling message, or just voice payload. If the packet contains an application layer signaling message, then type of message is also encoded in the PT field. The Encoder (ENC) field represents the voice encoding scheme used by the application. Finally, the Number of Frames (NFR) field contains the number of voice frames contained in the voice packet.

The lengths of the above fields are implementation dependent. The TOS octet in the IP header can contain these fields. However, depending on the implementation, it is possible that the eight bits of the TOS field would be insufficient to encode all the information. In such cases, one or more of the ACIS fields can be moved to the Offset field of the IP header. A similar approach is adopted in the DPS technique proposed in [Stoica]. The Offset field is actually meant for use only when the IP packets need to be fragmented. But voice packets are small and it is highly unlikely that they would ever be fragmented. Hence, the Offset field can be used to store some of the above information, if necessary.

The bandwidth required by voice connection can be inferred from the ENC and NFR fields, if a standard encoder is used. The bit-rates and frame sizes of the standard encoders are well known. The NFR field contains the number of frames included per voice packet. Using the frame size and the number of frames per packet, we can compute the protocol overhead incurred by the voice connection. We add the protocol overhead to the bit rate to get the actual bandwidth requirement of the connection.

For example, the G.729 encoder produces a voice stream at the rate 8 Kbps. The frame size of G.729 is 10 ms. When expressed in bits, the frame size is $(8 \times 1000) \times (10 / 1000) = 80$ bits. If we have two voice frames per packet, then the voice payload in a packet is $2 \times 80 = 160$ bits. Assume a protocol stack in which the protocols together add a header of size 40 bytes (320 bits). Then, the total packet size becomes $160 + 320 = 480$ bits. Thus, for every 160 bits of voice payload, there are 320 bits of header. Having obtained this ratio, we figure out that for an 8 Kbps voice stream, the headers would require 16 Kbps. The total bandwidth requirement, in this case, is $8 + 16 = 24$ Kbps.

If silence suppression were to be used by the encoders, then the actual bandwidth requirement would be less than what is computed above. However, the above computation at least gives an upper bound on the bandwidth requirement. A more sophisticated algorithm may use a scaling factor that takes into account the effect of silence suppression. We do not explore the idea further in this document.

3.4.2 Hash Table Description

The CPE maintains connection information for each premium call that it admits. It uses a hash table for this purpose. The flow ID of each call is used as the key for the hashing function. The flow ID of a call is the 4-tuple \langle source address, destination address, source port, destination port \rangle . (If we assume that only one voice call runs between a pair of end-hosts, then we do not even need port numbers. ACIS can then operate without snooping into the IP packets for higher layer information). The hash table has the following fields –

- Premium Flag – This is a flag that indicates whether the call is admitted as premium. This flag determines the validity of a hash table entry. If the flag is set, the hash table entry is valid.
- Bandwidth – This stores the bandwidth requirements of the call.
- Attempts – This stores the number of connection attempts made by this call.
- Last Used – This stores the time when a packet belonging to this call last arrived at the CPE.

The role played by each of these fields will become clear when we explain ACIS operation in the next section. The number of entries in the hashing table is bounded by the number of premium calls admitted by the CPE. We expect this number to be small for a single customer. Hence, maintaining per-flow information would not pose significant storage or processing overheads in our case.

3.4.3 ACIS Operation

When a SIP-based application wishes to establish a VoIP call, it sends an “Invite” message to the intended receiver. The ACIS fields in the IP header of the “Invite” message are used to alert the CPE to the establishment of a new connection. They also convey the connection information to the CPE, as described above. The CPE keeps track of the total bandwidth reserved for premium calls and the bandwidth that is currently available for incoming calls. The CPE tries to accommodate every incoming call as a premium call and downgrades calls to the regular category only when sufficient bandwidth is not available. This classification by the CPE is transparent to the end-hosts. The end-hosts are not explicitly informed whether their call has been accepted as premium or regular.

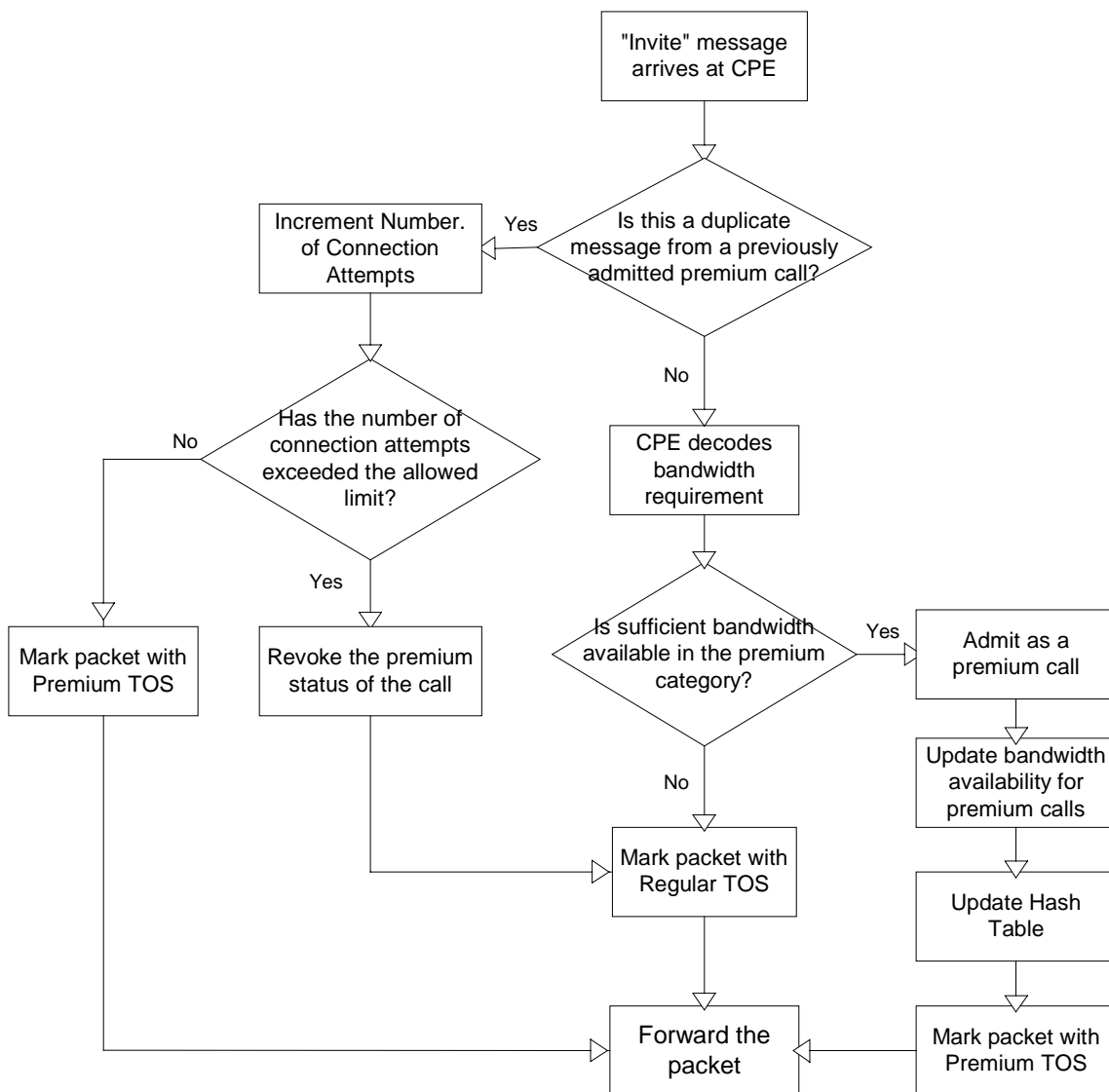


Figure 3.4: CPE operation on receiving “Invite” message

Upon receiving the “Invite” message, the CPE first checks whether this is the first connection attempt being made by the end-host. This is done by checking whether the hash table has a valid entry for this call. If this is the first attempt, then the CPE computes the bandwidth requirements for this call and compares it against the bandwidth availability for premium calls. (The CPE may use a lookup table to speed up the process of determining the bandwidth requirements from the ACIS fields). If sufficient bandwidth is available, it admits this call as a premium voice call. It then decreases its bandwidth availability to reflect the admission of the new call. In the hash table entry for that call, it makes the following updates –

- The ‘Premium’ flag is set
- The bandwidth requirement is stored in the ‘Bandwidth’ field
- Since this is the first connection attempt, the value in the ‘Attempts’ field is set to 1.
- The ‘Last Used’ field is set to the current time.

Finally, the CPE marks the TOS field in the packet’s IP header to indicate that the packet belongs to the premium category and forwards it. On the other hand, if there is no more bandwidth available for premium calls, the TOS field of the packet is marked to indicate that it belongs to the regular category and the packet is forwarded. Regular calls do not have any entry in the hashing table. At this point, we make some comments about remarking of packets at the CPE. Note that any previous TOS markings made at the end-host are overwritten at the CPE. Overwriting the TOS markings will not have any effect on ACIS operation, even if some of the ACIS fields are located in the TOS octet. The reason for this is that these ACIS fields have significance only within the customer premises and not in the rest of the network. Hence, the CPE should remark packets in accordance with some scheme that holds significance in the entire access network (and perhaps, beyond). In our discussion, we have simply assumed that premium and regular categories are supported in the rest of the network as well. Hence, we perform remarking at the CPE to indicate whether the voice packet belongs to a premium call or a regular call. However, in a situation where the rest of the network uses some other set of service classes (or none at all), remarking at the CPE would be done differently.

It is possible that the CPE receives duplicate “Invite” messages belonging to the same connection. This may be caused by retransmission due to packet loss or it may be a retry if the connection was rejected on a previous attempt. Thus, each time the CPE receives an “Invite” message, it checks the hash table to determine whether the call has already been admitted as a premium call. If it is already a premium call, then the CPE does not attempt to allocate more bandwidth for that call. Instead, it just increments the number of connection attempts, stored in the ‘Attempts’ field of the hash table. If the number of connection attempts exceeds a fixed maximum, then the CPE revokes the premium status of this call. It resets the ‘Premium’ flag in the hash table and marks the “Invite” packet as a regular category packet. This prevents a call from occupying the premium list for too long without being able to establish a connection. But if the number of connection attempts is less than the maximum allowed for a connection, then the “Invite” message is forwarded as a premium category packet. Figure 2.4 shows the handling of “Invite” messages by the CPE.

The “Invite” message is forwarded as usual through the network to the receiver, which would then confirm the acceptance of the request by sending of an “ACK” message. The session is established once the sender gets the “ACK” message. The same operations performed by the sender’s CPE on intercepting the “Invite” message could also be performed by the receiver’s CPE on intercepting the “ACK” message.

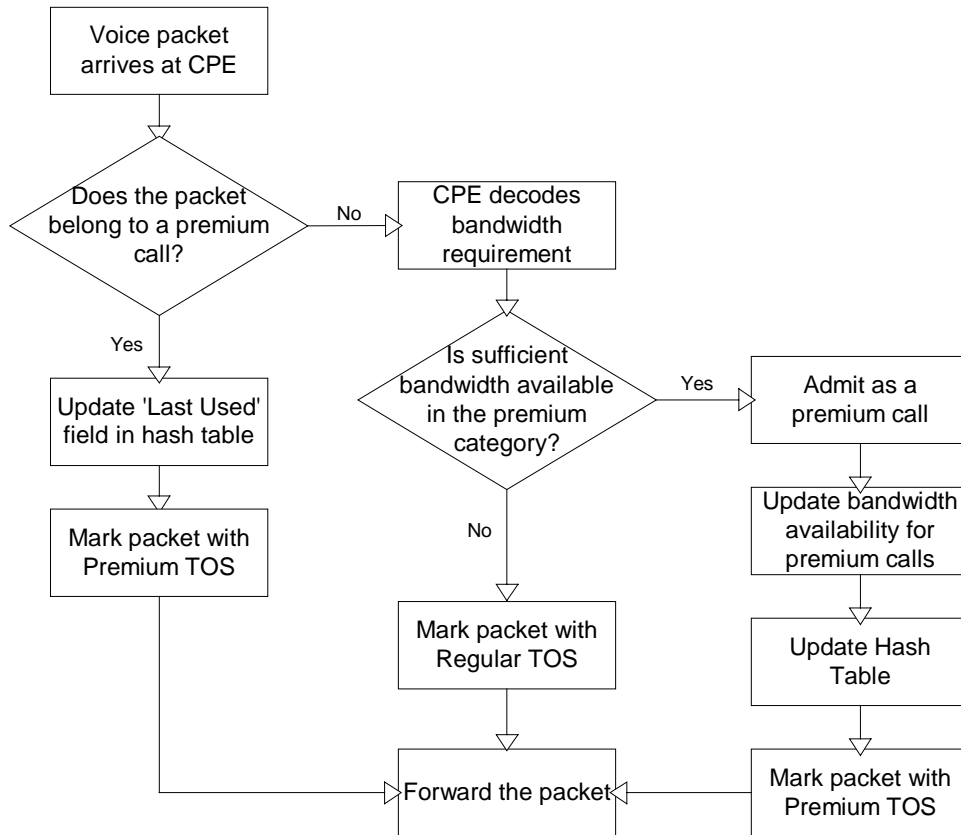


Figure 3.5: CPE operation on receiving voice packet

After the session has been established, end-host starts sending voice packets. The PT field, defined in the IP header for ACIS, is marked to indicate that this packet contains voice payload. The ENC and NFR fields are also marked with the appropriate values for the encoder and the number of frames per packet. Figure 2.5 represents the handling of regular voice packets by the CPE. When the CPE receives a voice packet, it first verifies whether the packet belongs to a premium call. This is accomplished by checking whether the hash table has a valid entry for this call. If so, then the ‘Last Used’ field of the hash table is updated to the current time. The TOS of the voice packet is then marked to reflect that it belongs to the premium category. The packet is then forwarded towards the DSLAM. However, if the voice packet does not belong to the premium call, the CPE checks whether the call can be upgraded from regular to premium category at this stage. This might be possible, if some premium calls have terminated, thereby freeing some premium bandwidth, after this particular call was established. Hence, the CPE determines the bandwidth requirement of the call from the ACIS fields and checks its bandwidth availability. If sufficient bandwidth is available, the call is upgraded to the premium

category. The premium bandwidth availability is decreased to reflect this admission and a hash table entry is created for this call as described earlier. Note that if several regular calls are in progress and there is enough bandwidth to upgrade only one of those, then the call that gets upgraded is the one whose voice packet reaches the CPE first after the additional bandwidth becomes available.

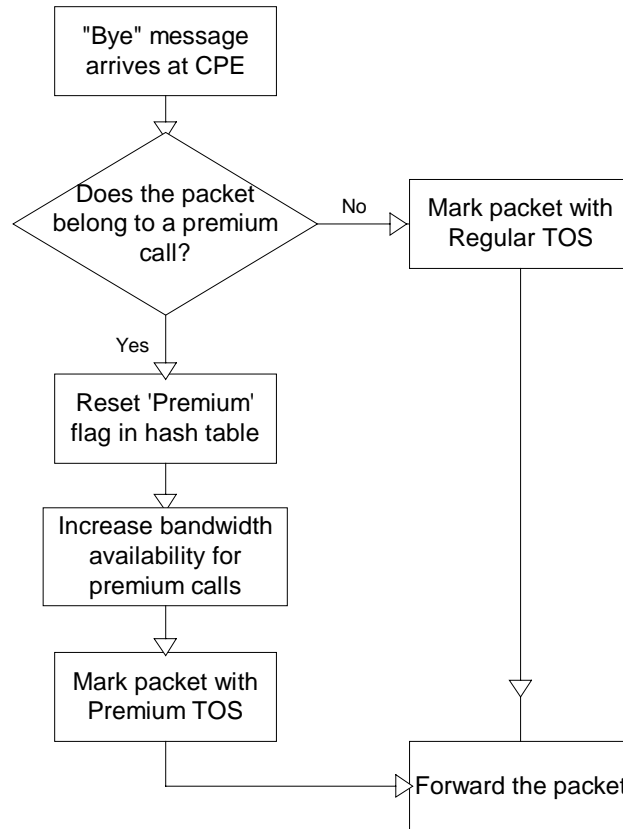


Figure 3.6: CPE operation on receiving “Bye” message

When the end-host wishes to end the voice call, SIP sends a “Bye” message to indicate the end of session to the other party. As usual, the PT field defined by ACIS is marked to indicate that the packet contains a “Bye” message and the ENC and NFR fields indicate the encoder used and the number of frames per packet respectively. When the CPE intercepts this message, it checks the hash table to determine whether the packet belongs to a premium call. If so, the ‘Premium’ flag in the hash table is reset to indicate that the call has terminated. The bandwidth availability for premium calls is increased to reflect the closing of this session. The TOS value of the packet is marked to indicate that it belongs to a premium call and the packet is then forwarded. If the “Bye” message belonged to a regular call, then the CPE simply marks the packet as a regular packet and forwards it. Duplicate “Bye” messages, generated due to retransmission (caused perhaps by packet loss) by the end-host, do not affect the correctness of ACIS. In such cases, the CPE would not mistakenly increase its bandwidth availability for each duplicate “Bye” message received, since the first “Bye” message would revoke the premium status of the

call. Duplicate “Bye” messages will thus be treated as packets belonging to regular calls. Figure 2.6 shows the actions taken by CPE on receiving a “Bye” message.

The mechanism described above needs to be further refined to address robustness issues. It is possible for a premium voice application running on the end-host to crash in the middle of a call. In this case, the application would never send a "Bye" message to indicate the end of the call. The CPE needs some mechanism to de-allocate the bandwidth that it had reserved for this application. To deal with this, we use the concept of soft state. For each premium call, the CPE keeps track of the time when a packet from the call last arrived. This is stored in the ‘Last Used’ field in the hash table entry. ACIS defines a maximum silence period for premium calls. If a premium call has not sent a packet for a period longer than the maximum silence period, then the premium status of the call can be revoked. This could be done by the CPE on a periodic basis. Or, it could be done on a need basis, i.e. if a new connection request arrives and there is no bandwidth available for it. A possible implementation would involve a combination of the two choices. That is, the CPE could revoke the premium status of idle connections at longer intervals, while new connection requests could trigger off an immediate check for idle connections whose premium status can be revoked. In either case, the CPE scans through the list of premium calls to see if there exist one or more calls that have not sent a packet for a period longer than the maximum silence period. Such calls are removed from the premium category in order to free some bandwidth and to give a chance to new calls.

3.5 Related Work

IP QoS has been an area of active research in the recent years. The research has led to the emergence of several mechanisms for providing different levels of service to different kinds of applications. Having described our QoS architecture, we compare and contrast our work with IP QoS mechanisms that exist in the literature.

Several publications have studied buffer management and scheduling algorithms under the assumption that the network is overprovisioned. Overprovisioning implies that sufficient bandwidth has already been provisioned for classes of traffic that require a high level of service. In this scenario, the offered load for these classes would never exceed the bandwidth provisioned. [Klepec] studies the effectiveness of priority queuing in providing QoS to voice traffic, while [Tyagi] evaluates Class-Based Queuing for the same purpose. [Lee] and [Hai] propose and evaluate their own buffer management and scheduling algorithms. All these schemes yield good results in situations with over – provisioning. But very often, the resources provisioned for traffic classes requiring strict service guarantees are limited. Hence, admission control must be applied to limit the number flows (of these classes) entering the network. Otherwise, all flows belonging to a class will be equally degraded when the offered load exceeds the limit that can be supported by the network for that class. Keeping this in mind, we designed our QoS architecture to include not only buffer management and scheduling, but also admission control.

In the past, researchers have come up with several ways to enforce admission control. Endpoint admission control is an admission control technique in which the end-hosts themselves make the admission control decision, without any support from the routers. The basic idea is to have the end-host send probe packets into the network at the rate at which it intends to transmit the actual data packets, if admitted. The probe packets are sent to the receiver with which the end-host wishes to communicate. The receiver makes measurements on the arriving probe packets in order to estimate the QoS received by them. The results are reported to the sender, which makes the admission control decision based on the report. [Breslau] and [Bianchi1] study the performance of endpoint admission control algorithms in general. [Borgonovo], [Elek] and [Mase] propose and evaluate their own endpoint admission control schemes. Endpoint admission control mechanisms fit nicely into the DiffServ architecture. Since they do not require any explicit support from the routers, they are highly scalable. However, these algorithms rely on measurements made by the end-hosts and hence they might be inaccurate (as discussed in [Breslau] and [Bianchi1]). Hence, endpoint admission control can provide only a Controlled Load type of service. The applications receiving this service can be assured that the aggregate load is kept within reasonable limits. However, no hard QoS guarantees can be provided to these applications. ACIS differs from endpoint admission control algorithms in that the admission decision is not taken by the end-hosts in ACIS. The end-hosts support ACIS operation by marking application layer signaling protocols appropriately, but the admission decision is taken by the router (CPE). The QoS architecture has been developed for the DSL access network and the end-to-end operation of endpoint admission control is not suitable for this. Besides, ACIS can provide better QoS guarantees since the admission decision is made by the network and not by the end-host.

Measurement-Based Admission Control (MBAC) is a concept in admission control where the decision to admit a flow into the network is made based on traffic measurements. Endpoint admission control algorithms are actually a special case of MBAC and are also called Edge-to-Edge MBAC (EMBAC) algorithms. Unlike endpoint admission control algorithms, the MBAC mechanisms do require router support. The involvement of routers offers MBAC a distinct advantage, since each router can make a more accurate estimate of local congestion. Consequently, the admission decisions in MBAC are based on accurate measurements made at each hop, rather than measurements made solely at the end-hosts. [Rhee] and [Bianchi2] propose their own MBAC schemes. [Bianchi3] proves that the scheme proposed in [Bianchi2] can be implemented in a DiffServ network that uses service differentiation mechanisms compliant with AF standards. The admission control schemes proposed in [Hill] and [Almesberger] are rather similar to MBAC schemes, though the authors themselves do not classify their schemes as MBAC schemes. [Jamin] and [Qi] focus on measurement techniques and the admission criteria employed by routers, rather than MBAC protocols. We feel that ACIS can be used in conjunction with an MBAC scheme. Currently, ACIS assumes that the CPE has a fixed amount of bandwidth reserved for premium calls. However, the assignment of bandwidth to classes of calls could also be varied dynamically based on measurements of traffic load.

[Cetinkaya] introduces the concept of egress admission control, where the admission control decisions are made at the egress routers. This is done by mapping the network path to a black-box model, which consists of arrivals that can be measured at the ingress and the services that can be measured at the egress. The egress node uses measurements made with the help of this black-box model to perform admission control decision. ACIS would be interoperable with egress admission control if the implicit signaling messages used by ACIS could be forwarded transparently through the access network to the egress router. The egress router would then perform admission control functions that are being performed by the CPE in the current ACIS design.

Some QoS architectures like those proposed in [Jia], [Wang], [Hou] and [Tajima] use a resource managing entity called a bandwidth broker (BB) to manage bandwidth resources within an administrative domain. The BB has information about the resources allocated within the domain and can thus make admission control decisions for the entire domain. A BB is a suitable solution for DiffServ networks where the core routers do maintain per-flow state. The BB maintains the per-flow state for all routers across the network. ACIS can be operated in conjunction with BBs also. Rather than have the CPE maintain a fixed bandwidth reservation for premium traffic, we can use a bandwidth broker (located perhaps at the DSLAM) to keep track of the bandwidth available for premium traffic in the network service provider's entire domain. The CPE would contact the BB whenever requests for new connections arrive. A signaling protocol would necessary for communication between the CPE and the BB. [Detti] and [Mameli] propose signaling protocols that can be used for this purpose.

We now turn our attention to techniques that rely on information stored in packet headers to provide QoS. [Bakiras] proposes a packet scheduling algorithm and an admission control algorithm that do not require the router to maintain any per-flow state. The packet scheduling algorithm requires scheduling parameters to be stored in the IP header of the packets. The admission control algorithm describes only the admission criteria, and needs to be supported by a signaling protocol that conveys the end-host requirements to the routers. Also, robustness issues that may arise from duplicate signaling messages are not considered in this paper. [Stoica] proposes a packet scheduling algorithm and an admission control algorithm, both of which are based on the concept of Dynamic Packet States (DPS). DPS involves using the packet header to carry state information that coordinates the actions among routers. It eliminates the need for routers to maintain any state information. The admission control technique described in this paper is robust in the presence of duplicate signaling messages. While ACIS does store information in packet headers, it is different from the above approaches because the information is initialized at the end-host and is not modified by routers. In our case, the purpose of storing information in the header is to communicate application requirements to the router. But in [Bakiras] and [Stoica], the purpose of storing information in the header is to coordinate the actions of the routers.

The use of end-to-end signaling protocols is another approach for providing IP QoS. The Resource Reservation Protocol (RSVP) ([Braden2]), Sender-initiated Resource Reservation Protocol (SRRP) ([Zhang]) and DiffServ PHB Reservation Protocols

(DPRP) ([Qian]) are examples of this approach. Since these are end-to-end protocols, they require several (if not all) routers along the path to be modified to support these protocols. On the other hand, ACIS focuses only on the QoS within the DSL access network. Hence, it requires modification of just the CPE. Moreover, ACIS relies on an implicit signaling protocol that takes advantage of application layer signaling by mapping it to the network layer header. [Mortier] proposes a similar admission control scheme that uses the concept of a connection at a higher layer to perform admission control. The paper describes how this idea can be used to perform admission control for TCP flows by snooping into the TCP header to identify SYN and SYN/ACK packets, which are used for connection establishment in TCP. Once the router identifies these packets, they can accept the connection by forwarding the packet as usual. Else, they can reject or defer the connection by dropping the packet or deferring its transmission. ACIS is different from this scheme in that it does not involve delving into the IP packet to get higher layer information. Instead, it maps application layer signaling information on to the IP header. Moreover, ACIS also explores the idea of classifying the connections into different classes of traffic rather than merely accepting or rejecting them.

Some researchers have also discussed the use of the IntServ architecture to provide QoS ([Wu] and [Baldi]). Since the IntServ architecture proposes reservation of resources on a per-flow basis, it is flexible and capable of providing guaranteed QoS to the individual flows. The per-flow approach also ensures higher resource utilization. However, the IntServ architecture requires routers to maintain per-flow state and this poses a scalability problem when the number of flows is large. [Vutukury] identifies this scalability problem and proposes a flow aggregation technique to address this scalability problem. The idea is to aggregate flows into a smaller number of flow classes and maintain reservation state information for each flow class, rather than for each flow. However, techniques such as these only alleviate, and not completely solve the scalability problem. Hence, the DiffServ architecture, with its undoubtedly superior scalability, is gaining popularity over IntServ. ACIS does require the CPE to maintain per-flow state information, but we do not expect the number of flows in the customer premises to be very large. Scalability is not much of an issue in this case.

Attempts have been made to design an end-to-end QoS model that combines the flexibility and levels of assurance provided by IntServ with the simplicity and scalability of DiffServ. [Bernet] proposes a framework for end-to-end Integrated Services operation over a network consisting of one or more DiffServ domains. IntServ treats the DiffServ domains as virtual links that connect IntServ-aware routers. The IntServ aware routers participate in the end-to-end RSVP signaling for resource reservation. A subset of routers in the DiffServ domains may also participate in RSVP signaling. An RSVP-aware ingress router performs admission control and resource reservation within the DiffServ domain. The DiffServ domains implement certain PHBs to support IntServ-style QoS within their domain. [Fineberg], [Tang], [Buchli], [Detti], [Mameli] and [Zhang] are all examples of proposals based on this framework.

While our QoS architecture also draws several ideas from both the IntServ and the DiffServ model, we do not claim strict compliance with either of these.

3.6 Chapter Summary

In this chapter, we discussed the IP QoS architecture that we propose a solution for voice and data convergence in the DSL access network. We first defined the different service classes that form part of our architecture. Next, we described the scheduling and buffer management mechanisms that we use in our solution. We also explained the need for fragmentation of large data packets and described how fragmentation based on Multilink PPP suits our purpose. Then, we proposed an admission control mechanism called ACIS (Admission Control by Implicit Signaling) to restrict the number of voice calls that can be admitted to our premium service class. Finally, we compared our work with QoS mechanisms already proposed by other researchers.

Chapter 4: Simulation Methodology

4.1 Objective

The objective of our simulations is to evaluate the performance of our IP QoS architecture. In particular, we investigate the feasibility of replacing the existing VoATM architecture with our VoIP architecture in future DSL access networks. If the mechanisms proposed by us can provide QoS guarantees that are at least reasonably close to what VoATM currently offers, then deploying our VoIP architecture would enable VoDSL to reap the benefits of compatibility between core and access network packet formats.

4.2 System Boundaries

In our study, the system of interest is the DSL access network. Recall that the objective of our research is to study the feasibility of replacing VoATM with VoIP in future VoDSL deployments. Hence, to compare VoATM and VoIP, we confine our simulations to the DSL access network. Voice communication in the core, based either on circuit-switched PSTN or packet-switched IP networks, is beyond the scope of our research. While in reality traffic from the access network enters the core network, in our study we assume that all traffic leaving the DSL access network simply terminates in a traffic sink. Our system thus ranges from the end-hosts in the customer premises to the traffic sink located just beyond the DSLAM. The key components of this system are the end-hosts, the CPE, the DSL, the DSLAM and the traffic sink.

4.3 Performance Metrics

The quality of a voice call depends on the delay experienced by voice packets as they travel across the network from sender to receiver. If this delay is too high, then the users find it difficult to carry out the conversation, since speech is delivered to the other party after a noticeable pause. Another important factor that affects the quality of packet voice is the variation in delay, also called jitter. If the jitter is too high, a larger jitter buffer must be used, i.e. the packets arriving early must be held for longer periods before being played out. This effectively increases the end-to-end delay as perceived by the users.

Our primary interest is in the QoS received by voice traffic as they traverse the DSL access network. The metrics that we use to characterize QoS for voice are the end-to-end delay (ETE delay) experienced by voice packets across the DSL access network and the variation in this ETE delay. Note that when we say ETE delay, we mean the delay experienced in traveling from one end of OUR SYSTEM to the other. This does not include the delays incurred in the core network and in the access network of the receiver. Thus, in the rest of this document, ETE delay must be interpreted to mean ETE delay across the DSL access network, unless we explicitly mention otherwise.

Another metric of interest is the bandwidth requirement of voice calls. Bandwidth is often a precious resource in data communication over low speed links such as the upstream link in ADSL. Several voice encoding schemes produce voice streams with low bit rates. However, since the voice packets are small, the headers added by each layer of the protocol stack impose a huge overhead on these packets. Consequently the bandwidth requirement of the voice stream is much higher than its raw bit rate.

4.4 Simulation Model

The ADSL link in our simulations runs for a distance of about 16000 feet between the CPE and the DSLAM. It has an upstream bandwidth of 128Kbps and a downstream bandwidth of 1.5 Mbps. The upstream link is the bottleneck in our system. All other link bandwidths are sufficient to support the load offered.

The simulation model for our VoATM scenarios is shown in Figure 4.1. The end hosts in the customer premises consist of Ethernet-based clients that generate data traffic and AAL2-based clients that generate voice traffic. The CPE has an Ethernet interface for data traffic, and some ATM interfaces for voice traffic. The interface connecting the CPE to the DSLAM is ATM-based. The CPE acts like an ATM switch, sending voice and data packets from the customer premises to the DSLAM. The CPE also performs the functions of an Ethernet to ATM gateway, since the CPE interface for the customer's data traffic is Ethernet-based and the uplink to the DSLAM is ATM-based. The gateway function converts the IP-over-Ethernet data packets into IP-over-AAL5 data packets before sending it over the upstream link. The gateway functions of the CPE are not needed for voice packets, since they are sent directly over AAL2. The DSLAM is an ATM switch that switches voice and data packets to their respective destinations. In our model, these destinations are traffic sinks that are connected directly to the DSLAM. We have used ATM-based end-hosts to model our traffic sinks. All ATM interfaces in our model have a separate output queue for each ATM service class (CBR, UBR, rt-VBR, nrt-VBR and ABR). A scheduler that uses the WRR scheme services these queues.

The simulation model for our VoIP scenarios is shown in Figure 4.2. The end-hosts in the customer premises are all Ethernet based clients that can generate both voice and data traffic. The CPE is an IP router, which has an Ethernet interface for the all the traffic (both voice and data) originating from the end-hosts. The interface connecting the CPE to the DSLAM could be ATM or Serial Line Internet Protocol (SLIP) [Romkey] or Frame Relay. In our experiments, we have considered ATM and SLIP only. The DSLAM is another IP router that routes the voice and data packets towards the traffic sinks. The traffic sinks are modeled using Ethernet-based end-hosts. By default, the IP interfaces in our model have a common output queue for all types of traffic. However, for the CPE interface that connects to the DSLAM, we have set up different output queues for different service classes (Premium, Regular, ACK, BE). A scheduler using WFQ services these queues. The ACIS mechanism is implemented on the CPE to limit the amount of premium traffic on the upstream link.

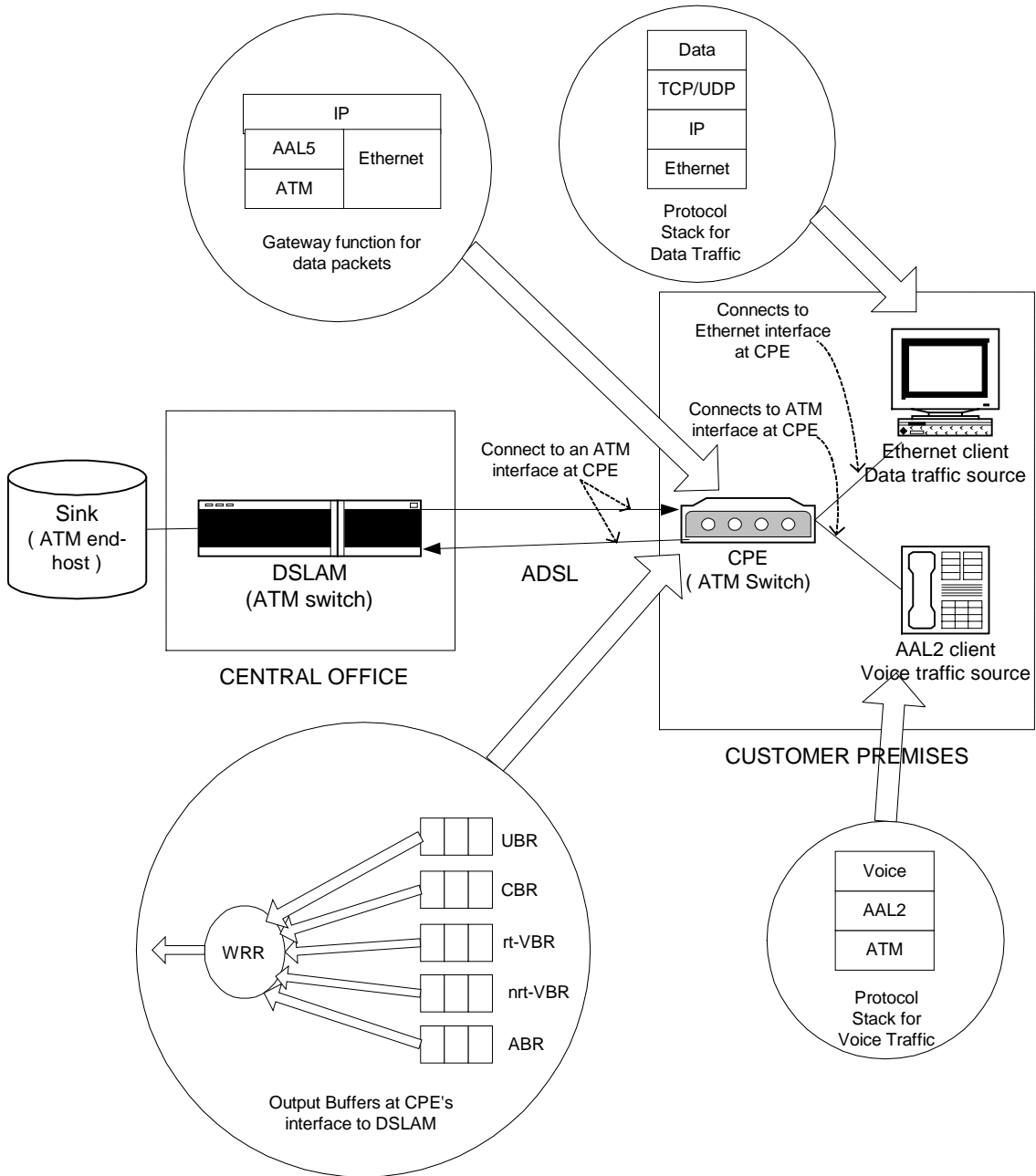


Figure 4.1: Simulation Model for VoATM scenarios

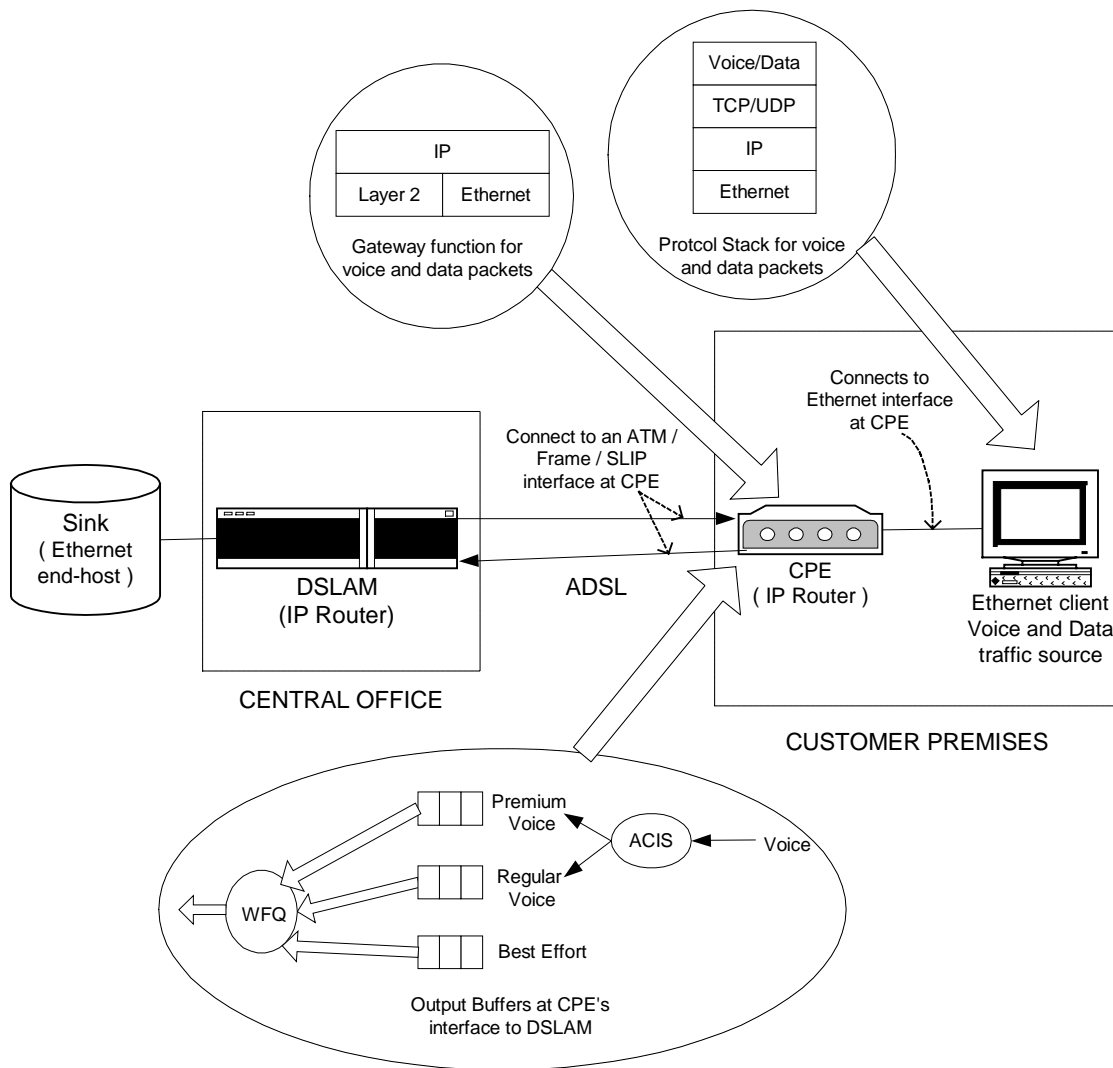


Figure 4.2: Simulation Model for VoIP scenarios

The G.729 encoder is used for all voice traffic. Recall that the G.729 encoder produces a voice stream at the rate of 8 Kbps. The frame size is 10 ms (10 bytes of data). Data traffic has been modeled using standard applications such as FTP, HTTP, E-mail, and Remote Login. In our VoATM models, the protocol stack for voice is AAL2 over ATM and for data it is TCP/IP over AAL5 over ATM. Voice and data are assigned the CBR and UBR service classes respectively, and are sent over different VCs from the CPE to the DSLAM. In our VoIP models, the protocol stack for voice is UDP over IP over Layer 2 (ATM or SLIP) and for data it is TCP over IP over Layer 2 (again, ATM or SLIP). Voice is assigned to either Premium or Regular service class and data is assigned to the Best Effort Service class. We fragment large IP packets before transmitting them over the slow uplink. We do this by setting the MTU of the CPE's uplink interface to an appropriate value.

In our VoIP simulation model, we have made two simplifications. First, we have not modeled the RTP protocol for encapsulating voice traffic. Instead, we send the voice packets directly over UDP. As a result of this simplification, the values that we get for bandwidth requirement of voice calls would be lesser than the values in the real world scenarios, which would include RTP in their protocol stack. However, in our research we are interested only in the general principles that govern bandwidth requirements of voice calls. Precise numbers are not really the focus here. Hence we could afford to make the simplification.

Second, while our VoIP QoS architecture actually proposes the use of Multilink PPP for fragmentation of large IP packets, our simulation model merely reduces the MTU of the CPE's interface to the DSLAM for this purpose. As a result of this simplification, the protocol overhead added to the data packets will increase, since the IP header is replicated in each fragment. However, this will not affect the performance of voice traffic, since the size of voice packets is well below the MTU. Since we are primarily focusing on the performance of voice traffic, we can afford this simplification as well.

We make these simplifications because the simulation tool that we have used does not have support for either RTP or Multilink PPP.

4.5 Implementation of the Simulation Model

We implemented our simulation model on OPNET Modeler™ ([Opnet]). *In building our model, we used several models that are already provided by OPNET Modeler™. In the discussion that follows, these models are shown in italics.*

In our VoATM model, we used *Ethernet workstations* and *ATM UNI clients* to model the end-hosts that generate data traffic and voice traffic respectively. We set the Adaptation Layer of the *ATM UNI clients* as AAL2. We used high speed *ATM links* to connect the *ATM UNI clients* to the ATM interfaces of the CPE. The *Ethernet workstations* are connected using *10BaseT links* to the Ethernet interface of the CPE. The CPE has only a single Ethernet interface. So, when there is more than one *Ethernet workstation* in the customer premises, we connect these to the CPE via an *Ethernet switch*. The CPE was modeled using an *Ethernet-to-ATM gateway* and an *ATM switch*. This is shown in Figure 4.3.

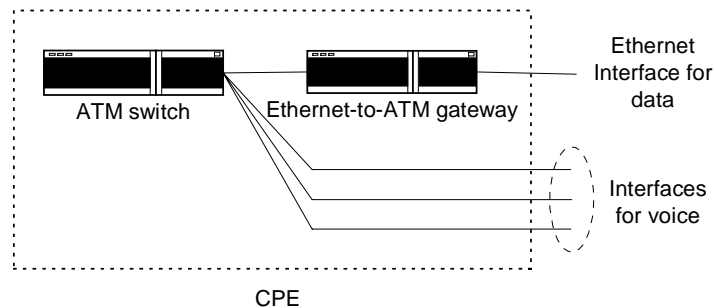


Figure 4.3: CPE model in the VoATM scenario

The ATM interface of the *Ethernet-to-ATM gateway* connects to one of the ports of the *ATM switch* using a high-speed *ATM link*. The remaining ports of the *ATM switch* are for voice connections. The Ethernet interface of the *Ethernet-to-ATM gateway* is for the data traffic connections. The ADSL link was modeled using two simplex *ATM links*, one each for upstream and downstream. The DSLAM was modeled using an *ATM switch*. Our model has several traffic sinks, one for each connection. Statistics for some of the metrics are gathered at the sink. So, to gather results for each individual connection, it is convenient to have a separate sink for each connection. The sinks for voice connections were modeled using *ATM UNI clients*, while those for data connections were modeled using *ATM servers*. OPNET Modeler™ also provides support for output queuing based on service class and servicing these queues using WRR scheme.

In our VoIP model, *Ethernet workstations* were used to model end-hosts that could generate both voice and data traffic. The CPE was modeled using an *IP router* with an Ethernet interface for the customer's voice and data traffic. The *Ethernet workstations* are connected to the Ethernet interface using *10BaseT* links. When several workstations need to share the Ethernet interface, an *Ethernet switch* is used to connect to the CPE. The interface connecting to the CPE to the DSLAM is ATM in some experiments (ATM at Layer 2) and SLIP in others (SLIP at Layer 2). Correspondingly, the ADSL link is modeled using two simplex *ATM links* in the 'ATM at Layer 2' experiments and two simplex *PPP links* in the 'SLIP at Layer 2' experiments. The DSLAM is also modeled using an *IP router*. The traffic sinks are modeled using *Ethernet workstations* for voice connects and *Ethernet servers* for data connections. Like in the VoATM model, we have one sink for each connection.

OPNET Modeler™ provides support for TOS based queuing and for servicing these queues using the WFQ scheme. However, in OPNET Modeler™'s implementation of WFQ, the entire link bandwidth is split up among the different queues. No spare bandwidth is left for transmitting the header bits that will be added by the Layer 2 protocol. Hence, if the protocol overhead imposed by Layer 2 is high enough, then the packets will experience considerable queuing delays even if the traffic arrival rate at the queue is equal to the bandwidth assigned for the queue. We solved this problem by scaling down the link bandwidth before splitting it among the different queues. The scaling factor chosen is the ratio of the packet size before adding the Layer 2 header to the packet size after adding the Layer 2 header.

For fragmenting large IP packets, we set the MTU attribute of the CPE's interface the DSLAM to the value of the largest packet size we wish to permit.

Finally, we implemented our ACIS mechanism in OPNET Modeler. We added our code for admission control at the IP layer in the CPE model. Whenever a packet arrives at the interface of the CPE that connects to the customer premises, we check its header to see if it is marked as a voice packet. If it is, then the admission control procedures are executed before forwarding the packet and placing it in the appropriate output queue. The code for marking of packets by the end-host was added at the interface between the application

and transport layers in the end-host model. This interface conveys the application layer requirements (including the desired TOS marking) to the transport layer, which then passes it down to the IP layer.

4.6 Validation and Verification

4.6.1 Validation

Validation is the process of determining whether the model actually simulates the system of interest to a reasonable level of accuracy. To validate our model, we compared the results obtained during a few preliminary simulations with some of the results that could be verified analytically.

We simulated a scenario in which the voice traffic load is less than the data rate of the upstream ADSL link bandwidth. We did not introduce any competing data traffic into our system. Hence the queuing delay at the upstream link was zero. The other links in the system are all high-speed links and hence we assume that the queuing and transmission delay at those links is negligible. In this case, the total ETE delay of voice packets should be equal to the transmission delay of the packet at the upstream link, since this is the only significant delay in the system. We computed this value analytically and found it to match with the simulation result. We repeated the experiment by varying the link capacity and also by varying the size of the voice packets.

Next, we simulated a scenario where the offered traffic load is greater than the ADSL upstream link bandwidth. We also introduced some competing data traffic into our system. In this case, the voice traffic gets its share of bandwidth, but the queuing delays at the CPE become more significant. We theoretically computed the transmission delay of voice packets over the upstream link. We also measured values for queuing delay at the CPE. Once again, we assume that the other links are high-speed links and the queuing and transmission delays are negligible at those. Hence, the ETE delay should be equal to the sum of the queuing delay and the transmission delay at the upstream link. Our simulation results for ETE delay confirmed this.

Also, depending on when the voice packet arrives at the CPE, its queuing delay can vary between zero and the transmission time of the largest possible packet. We found this to be true in our simulations. We experimented with different values of maximum packet size and each time, observations are in accordance with the theoretical predictions.

Industry standards recommend that typical access network delays for voice packets must be less than 30 ms. Depending on our network model and the load on the network, we obtain values between 5 and 27 ms.

We also computed the bandwidth requirements of voice calls theoretically. Using the encoding scheme, the number of frames included in a voice packet and the protocol

header sizes, we arrived at theoretical numbers for bandwidth. Our simulation results for the bandwidth consumed by voice calls are found to match the theoretical values exactly.

The close match between our theoretical computations and simulation results increases our confidence in the validity of our simulation model.

4.6.2 Verification

Verification is the process of determining whether the implementation of the model is correct. For verification purposes, we tested our implementation under various conditions and observed whether the behavior of the system was along the expected lines.

In our VoATM model, the ETE delays for voice packets were high when both voice and data received UBR service. But when voice received CBR service and data received UBR service, the ETE delays dropped sharply. Also, when the combined reservation requested by CBR traffic exceeded the maximum bandwidth reserved for CBR traffic at a particular ATM switch, new CBR calls got rejected.

In our VoIP model, the voice packets suffered huge ETE delays in the absence of any QoS mechanisms. We observed that these delays reduced when we incorporated TOS-based WFQ and fragmented large data packets. Next, we tested our ACIS implementation. We verified that the CPE correctly updates the value for premium bandwidth availability each time a premium call joins or leaves the system. We also verified that the CPE stops admitting premium calls when premium bandwidth is no longer available. We observed that the ETE delays of premium voice calls are low under all traffic loads, but the ETE delays of the regular voice calls increase with increasing load. We tested the soft-state implementation and the 'call upgrade' feature of ACIS and found the results to be along expected lines.

Finally, we measured the bandwidth requirements of voice calls using different encoders. We also varied the number of voice frames included in a packet. The results obtained were accurate for all encoders. When we increased the number of voice frames per packet, the voice packets became larger. Hence the protocol overhead reduced, resulting in reduced bandwidth requirements.

During the intensive testing of our system under various conditions, we always found the behavior along expected lines. We were thus convinced that our implementation is correct.

4.7 Experiments

We first performed experiments on the VoATM model to understand the effectiveness of ATM QoS mechanisms. We then started with the best-effort IP model and incorporated IP QoS mechanisms one by one to observe the increments in performance. The IP QoS mechanisms that we investigated are TOS-based WFQ, fragmentation of large data

packets and ACIS. We also compared ATM and SLIP as Layer 2 protocol options for VoIP. We conducted experiments to test our IP QoS mechanisms under different loads and different mixes of competing data traffic. We finally compared our VoIP results with our VoATM results.

Our simulation time was chosen such that the system was in a steady state for more than 90% of the simulation time. After a few experimental runs, we determined that a simulation time of 4 minutes was adequate for our purposes. We discarded the data points that were collected when the system was in a transient state.

We collected statistics for the ETE delays of the voice packets. We computed the average delay from the data points collected for each voice call from each simulation run. We thus have (#voice calls) * (#simulation runs) values for average delay. We compute 95% confidence intervals for ETE delay using the following formula ([Garcia])-:

$$(M_n - z * (V_n / \sqrt{n}) , M_n + z * (V_n / \sqrt{n}))$$

Where -

- n, the number of samples of ETE delay, is equal to (#voice calls) * (#simulation runs)
- M_n and V_n are the sample mean and variance
- z is a constant whose value depends upon the percentage of confidence and the number of samples

In order to measure the jitter experienced by voice calls, we computed the variance of the ETE delay samples taken during each run.

We also obtained results for bandwidth requirements of voice calls using different protocol stacks. The values obtained for bandwidth requirement are deterministic, rather than stochastic.

Our results are presented in the next chapter.

4.8 Chapter Summary

This chapter described the simulation methodology used for evaluating the performance our proposed QoS architecture. We stated our objective, identified the system of interest and defined our performance metrics. We then described our simulation models for VoATM and VoIP scenarios and discussed how these were implemented in OPNET Modeler™. Next, we discussed the validation and verification of our model. Finally, we stated in brief the experiments performed, and the methods employed for data collection and data processing.

Chapter 5: Performance Evaluation

In this chapter we describe the experiments performed to evaluate the performance of our system. *In all experiments that we discuss, the ADSL link has upstream bandwidth of 128 Kbps and downstream bandwidth of 1.5 Mbps.* The upstream ADSL link is the bottleneck in our system; all other links have sufficient capacity to serve the arriving packets with minimum queuing and transmission delays.

5.1 VoATM Model

To evaluate our ATM model, we first perform an experiment with both voice and data receiving UBR service. We then study the improvement in performance when voice receives CBR service and data receives UBR service.

In our experimental setup, we have four end-hosts in the customer premises engaged in voice calls. They use the G.729 encoder, and include 2 voice frames in each packet. Each call requires 21.2 Kbps, including the protocol headers. One end-host on the customer premises generates data traffic by uploading a 2 MB file via FTP.

In our first experiment, both voice and data receive UBR service. Bandwidth guarantees are not given to any of the flows. The FTP application generates a huge amount of data traffic that competes with voice traffic for the upstream bandwidth. Both voice and data traffic share the UBR queue. Since the voice calls are not guaranteed any portion of the bandwidth, the average ETE delay of the voice packets is high. The results are shown in Table 5.1.

Run	ETE Delay (s)			
	Call 1	Call 2	Call 3	Call 4
1	0.7661	0.7702	0.7605	0.7714
2	0.7515	0.7535	0.7495	0.7502
3	0.7410	0.7420	0.7453	0.7383
4	0.7421	0.7435	0.7520	0.7431
5	0.7330	0.7293	0.7317	0.7294

Mean ETE Delay: 0.747 s

95% Confidence Interval for ETE Delay: (0.741, 0.753) s

Delay Variation: (0.00414, 0.00724) s²

Table 5.1: ETE Delays – VoATM with UBR voice and data

We repeat the above experiment with voice traffic receiving CBR service and data traffic receiving UBR service. Using ATM's call setup mechanism, each voice call requests CBR service and specifies a bandwidth requirement of 21.2 Kbps. Thus, a voice connection that is accepted has 21.2 Kbps upstream bandwidth reserved for it. Voice

packets are sent to the CBR queue and data packets are sent to the UBR queue. The CBR queue has 50% of the upstream link bandwidth (50% of 128 Kbps = 64 Kbps) reserved for it. In this experiment, the 64 Kbps reserved in the upstream link for voice can support a maximum of three voice calls ($3 * 21.2 \text{ Kbps} = 63.6 \text{ Kbps}$). Hence, the CPE rejects one of the four voice calls during call setup. However, the three accepted calls experience low ETE delays because they sufficient upstream bandwidth reserved for them. Since the ATM cells are all small and equal in size, a cell that arrives in the CBR queue while a UBR cell is being transmitted will not have to wait too long. Hence, the delay variation is low. The results are shown in Table 5.2. Figure 5.1 shows the decrease in ETE delay for voice packets when they receive CBR service, as compared to UBR service.

Run	ETE Delay (s)			
	Call 1	Call 2	Call 3	Call 4
1	0.0258	0.0259	0.0259	Rejected
2	0.0261	0.0259	0.0259	Rejected
3	0.0258	0.0257	0.0260	Rejected
4	0.0259	0.0260	0.0258	Rejected
5	0.0260	0.0257	0.0258	Rejected

Mean ETE Delay: 0.0259 s

95% Confidence Interval for ETE Delay: (0.0258,0.0259) s

Delay Variation: (1.03×10^{-6} , 2.6×10^{-6}) s^2

Table 5.2: ETE Delays – VoATM with CBR voice and UBR data

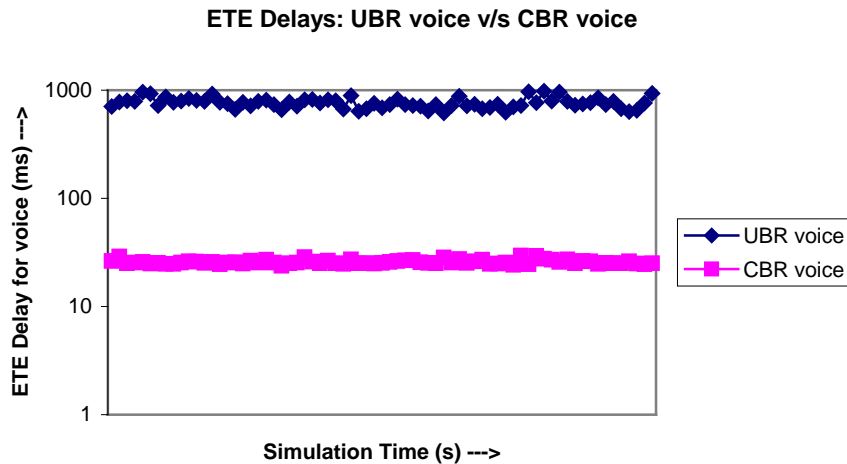


Figure 5.1: Comparison of ETE delays for UBR voice and CBR voice

5.2 VoIP Model

5.2.1 Weighted Fair Queuing

We first simulate our VoIP model for the traditional best-effort service offered by IP. Most of the current DSL deployments use ATM over DSL. Hence, in our initial experiments, we use ATM as the Layer 2 protocol. In our experimental setup, we have two end-hosts in the customer premises participating in voice calls. The G.729 encoder is used, with 1 frame per voice packet. The bandwidth requirement of a voice call is 42.4 Kbps, including the protocol headers. Data traffic is generated by an end-host uploading a 2 MB file via FTP. In the traditional best-effort IP model, there is no service differentiation between voice and data packets, both of which share the same queue. The FTP application generates a large amount of data traffic that severely degrades the performance of voice traffic. Voice packets experience long queuing delays as they are trapped behind several large data packets in the queue. The mean ETE delays of the voice packets are shown in Table 5.3. It is clear that these are unacceptably high for real-time applications.

Run	ETE Delay (s)	
	Call 1	Call 2
1	6.7499	6.7498
2	6.4483	6.4602
3	6.4599	6.4737
4	6.5245	6.5209
5	6.5351	6.5365

Mean ETE Delay: 6.546 s

95% Confidence Interval for ETE delay: (6.465, 6.626) s

Delay Variation: (0.1013, 0.2363) s²

Table 5.3: ETE Delays – VoIP with no service differentiation

Next, we incorporate the TOS-based WFQ mechanism at the CPE and repeat the above experiment. Voice and data packets are identified by different TOS values and placed in different queues. (*In this experiment, we have just one category for voice packets. Later, when we describe our experiments with ACIS, we have premium voice and regular voice*). Since we have two voice calls, each requiring 42.4 Kbps, we must reserve at least $2 * 42.4 = 84.8$ Kbps upstream bandwidth for voice traffic. In our experiment, we assign weights to the queues such that 70% of the upstream bandwidth (70% of 128 Kbps = 89.6 Kbps) is reserved for voice traffic. The ETE delays for the voice packets drop considerably, because the voice packets now have a separate queue and are served in a timely fashion. They no longer are stuck behind long queues of data packets. Table 5.4 shows the results.

Run	ETE Delay (s)	
	Call 1	Call 2
1	0.0778	0.0778
2	0.0779	0.0775
3	0.0763	0.0783
4	0.0779	0.0795
5	0.0772	0.0763

Mean ETE Delay: 0.0777 s

95% Confidence Interval for ETE Delay: (0.0770, 0.0783) s

Delay Variation: $(4.35 \times 10^{-5}, 13.7 \times 10^{-5}) \text{ s}^2$

Table 5.4: ETE Delays – VoIP with WFQ

5.2.2 Fragmentation

Although WFQ succeeded in bringing down the ETE delay of voice packets, WFQ alone is not good enough. Note that the ETE delay obtained in the WFQ experiment is about 78 ms, but typical access network delays for voice calls ought to be less than 30 ms. Also, the delay variation, although small, is still higher than what we observed in the VoATM with CBR voice experiment. The reason is that even though voice packets have their share of bandwidth, they cannot preempt a data packet that is already being transmitted when they arrive. The end-hosts on the customer premises run on Ethernet, which supports packet sizes of up to 1500 bytes. A 1500 byte packet takes roughly 94 ms, to be transmitted over a 128 Kbps link. Thus, an arriving voice packet may have to wait anywhere from 0 ms to 94 ms in the queue, depending on whether a data packet is already in transmission and how much of that transmission is remaining. It is easy to see that in this scenario, very often the waiting time could be unacceptably high for voice packets.

To investigate the performance improvement obtained by fragmentation of large data packets, we reduce the MTU of the upstream interface to a value below 1500 bytes. We choose a value such that the data packets are not too severely fragmented. At the same time, we wish to limit the queuing delay of arriving voice packets to acceptable limits. [Caputo] suggests a figure of 20 ms as the upper limit for transmission time of packets on a low-speed link. We select a value close to 20 ms, since it suits our purposes too. With a maximum waiting time of 20 ms and a minimum waiting time of 0 ms, the average queuing delay for voice packets would be 10 ms. This is well below our 30 ms ETE delay budget for the access network. A 128 Kbps link can transmit $(128000 \times 20) / (8 \times 1000) = 320$ bytes of data in 20 ms. Hence, we select a value close to 320 bytes as the MTU of the upstream interface. Since the Layer 2 protocol is ATM, we choose the MTU such that the packet would fit exactly into an integral number of ATM cells. We thus eliminate any overhead caused by padding. In our experiment, we chose an MTU of 328 bytes. After adding an AAL5 trailer of 8 bytes, we get a total of 336 bytes, which then fits exactly into

7 ATM cells. Along with their headers, these 7 ATM cells make up $7 \times 53 = 371$ bytes of data. Thus, the maximum transmission time for a packet is $(371 \times 8) / 12800 = 23$ ms.

We repeat the experiment for VoIP with WFQ after setting the MTU of the CPE's upstream interface to 328 bytes. The fragmentation of large data packets into smaller packets results in lower delay and lower variations in delay. The results are shown in Table 5.5. Observe that the overall ETE delays are now less than 30 ms, which was what we had aimed for. The values for ETE delay obtained in this experiment are comparable to the values in the VoATM with CBR voice experiment. Figure 5.2 shows the improvement in VoIP performance resulting from WFQ and fragmentation.

Run	ETE Delay (s)	
	Call 1	Call 2
1	0.0239	0.0240
2	0.0238	0.0237
3	0.0241	0.0240
4	0.0239	0.0241
5	0.0240	0.0238

Mean ETE Delay: 0.0239 s

95% Confidence Interval for ETE Delay: (0.0238, 0.0240) s

Delay Variation: $(5.27 \times 10^{-7}, 17.5 \times 10^{-7})$ s²

Table 5.5: ETE Delays – VoIP with WFQ and Fragmentation

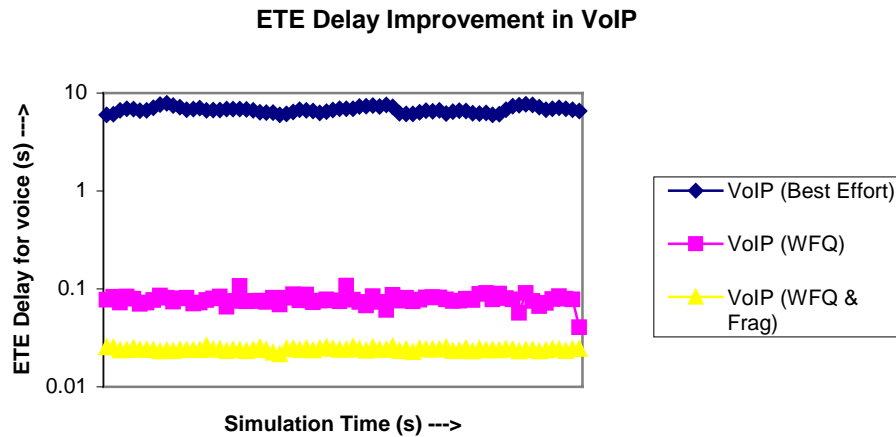


Figure 5.2: Improvement in VoIP Performance

5.2.3 Comparison of Layer 2 Protocols

We now turn our attention to the bandwidth requirements for voice calls. Recall that in our VoIP experiments, each voice call used the G.729 encoder and required a bandwidth of 42.4 Kbps. The G.729 encoder produces a raw voice bit stream at the rate of 8 Kbps. The rest of the bandwidth ($42.4 - 8 = 34.4$ Kbps) is for the protocol headers. VoIP is obviously very extravagant in terms of bandwidth utilization.

The problem with the IP over ATM protocol stack for voice is that voice packets should be small in order to minimize the packetization delay and the transmission delay (especially on low speed links). In the IP over ATM (AAL5) model, AAL5 adds an 8-byte trailer to the voice payload and the ATM layer adds a 5-byte header for every 48 bytes of payload. Since the cell sizes are fixed, the ATM layer may also add a pad as large as 47 bytes to make the payload fit exactly into an integral number of cells. All this is in addition to the UDP and IP headers. (Recall that for the sake of simplicity, our model does not include RTP. Having RTP would make things even worse!) . As we discussed in Section 2.5.1, techniques like header compression and IP trunking alleviate the bandwidth problem, but an investigation of these techniques is beyond the scope of our research. Instead, we look at SLIP as an alternative to ATM in Layer 2. SLIP is used for exchange of data over point-to-point links (such as our ADSL link). SLIP has a very small header and consequently, it imposes a low protocol overhead. Moreover, the payload size of a SLIP frame is not fixed and hence there is no need to add padding. Figure 5.3 compares the bandwidth requirement of VoIP over ATM with that of VoIP over SLIP, when G.729 is used as the encoder.

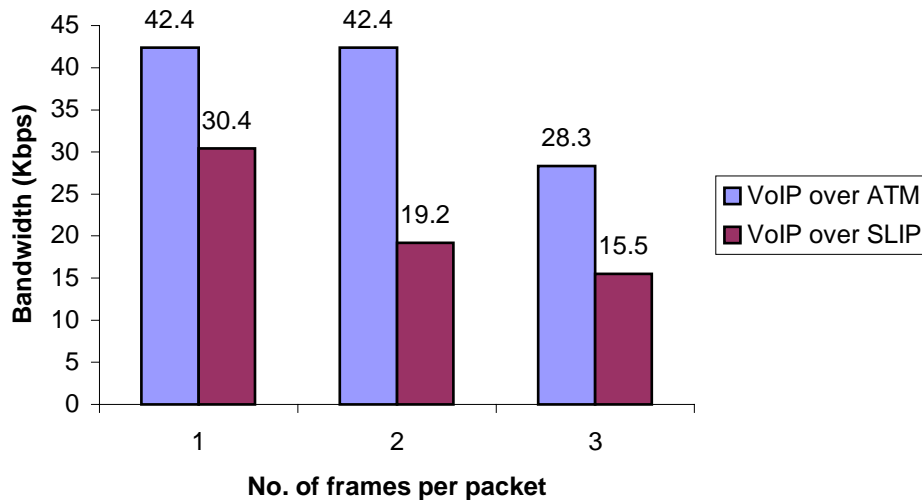


Figure 5.3: Bandwidth requirements for VoIP over ATM and VoIP over SLIP

Observe that the bandwidth requirement for VoIP over SLIP is consistently less than that of VoIP over ATM. When SLIP is used at Layer 2, increasing the number of voice frames per packet decreases the bandwidth requirement. However, this also increases the packetization delay. Thus, we need to make a tradeoff between bandwidth requirement and packetization delay. In the case where ATM is used at Layer 2, we cannot always make this tradeoff. From Figure 5.3, we see that for VoIP over ATM, there is no change in bandwidth requirement per call when we increase the number of frames per packet from 1 to 2. The reason is that the payload size exceeds 48 bytes on including the second frame. Hence, two ATM cells are now required to hold that payload, instead of just one. In other words, in the earlier case one cell is able to carry one voice frame and in the latter case two cells are needed to carry two voice frames. Hence, there is no reduction in the bandwidth requirement by packing two voice frames in a packet. There is only the penalty of a higher packetization delay. For this reason, we included only 1 voice frame per packet in our VoIP over ATM experiments described earlier. We could have included 3 voice frames per packet to improve the efficiency, but then the packetization delay would have been unacceptably high.

For the rest of the experiments described in this chapter, we replace ATM with SLIP in Layer 2 because of the superior bandwidth efficiency yielded by SLIP.

Currently, SLIP is being replaced by PPP, which is a more sophisticated protocol that accomplishes the same task. We used SLIP in our experiments since OPNET Modeler™ 7.0 does not provide support for PPP. However, like SLIP, PPP also imposes a low protocol overhead. Hence, the bandwidth requirements for a voice call using VoIP with PPP at Layer 2 protocol will be similar to that of SLIP. In addition, PPP also allows us to take advantage of the fragmentation by Multi-link PPP, as described in section 3.3.

We would also like to mention that Frame Relay (FR) provides another good option as a Layer 2 protocol in the VoIP stack. The protocol overhead in FR is slightly higher than that in SLIP, but it is much lower than in ATM. Also, the frame size in FR is not fixed, hence padding is not added.

We now describe an experiment that uses VoIP over SLIP. Three end-hosts in the customer premises are participating in voice calls and one end-host is uploading a 2 MB file via FTP. The end-hosts use the G.729 encoder and pack 2 frames in each voice packet. The bit rate of a voice call is 19.2 Kbps. Hence, we need to reserve at least $19.2 * 3 = 57.6$ Kbps upstream bandwidth for the voice traffic. Note that when compared to the 'VoIP over ATM' experiment, we need to reserve less bandwidth in this case, even though we support an additional voice call. For this experiment, we reserve 50% of the upstream bandwidth (50% of 128 Kbps = 64 Kbps) for voice packets. The MTU of the upstream interface is set to 328 bytes. Table 5.6 shows the mean ETE delay of voice packets in this experiment.

Run	ETE Delay (s)		
	Call 1	Call 2	Call 3
1	0.0144	0.0148	0.0146
2	0.0146	0.0150	0.0148
3	0.0146	0.0143	0.0145
4	0.0145	0.0146	0.0145
5	0.0147	0.0142	0.0145

Mean ETE Delay: 0.0146 s

95% Confidence Interval for ETE Delay: (0.0145, 0.0147) s

Delay Variation: $(1.15 \times 10^{-6}, 2.81 \times 10^{-6}) s^2$

Table 5.6: ETE Delays – VoIP over SLIP with WFQ and Fragmentation

Interestingly, the ETE delays of the voice packets for ‘VoIP over SLIP’ are the lowest that we have observed in our experiments. Recall that in VoATM and in ‘VoIP over ATM’, the ETE delays were approximately 25 ms. We attribute this difference to the relative complexity of the ATM protocol stack when compared to SLIP. The packet processing functions at the various layers of the ATM protocol stack cause the additional delay in the VoATM and ‘VoIP over ATM’ experiments. To verify this, we collected results for the ETE delay of the ATM cells and also the ETE delay of the AAL PDUs. We found that the ETE delays of the AAL PDUs were much higher than those of the ATM cells. This indicates the delay imposed by the ATM stack. Thus, the simplicity of SLIP not only offers greater bandwidth efficiency, but also lowers ETE delay.

We conclude this section with a graph (Figure 5.4) that compares the ETE delays for the QoS solutions we have investigated so far.

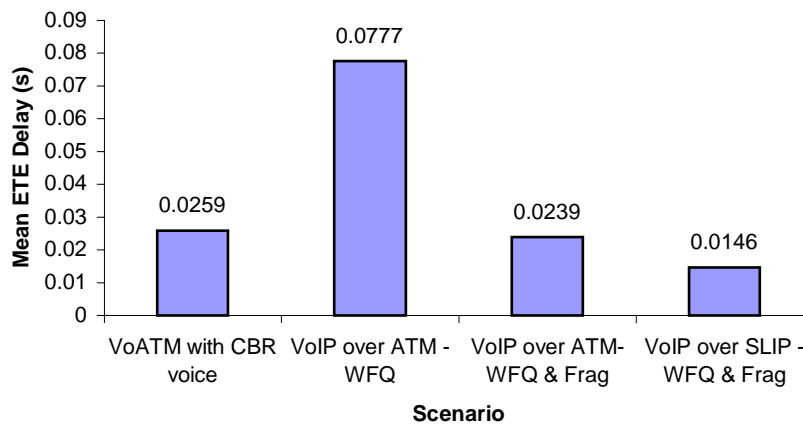


Figure 5.4: Comparison of ETE delays for VoATM and VoIP

5.2.4 ACIS

In all our VoIP experiments so far, we have computed the total bandwidth requirement for all the voice calls and reserved sufficient upstream bandwidth at the CPE. But the number of voice calls may not always be known in advance. Hence, it is possible for offered voice traffic load to exceed the bandwidth reserved for voice. If this were to happen, then the performance of all the voice calls will degrade. This is illustrated in our next experiment.

Four end-hosts in the customer premises are engaged in voice calls. They use the G.729 encoder with 2 voice frames per packet. The protocol stack used is VoIP over SLIP. The resulting bit rate of a voice call is 19.2 Kbps. Another end-host uploads a 2 MB file via FTP. The CPE implements both WFQ and fragmentation of large packets. Assume that the number of calls is not known in advance and that 35 % of the upstream bandwidth (35 % of 128 Kbps = 44.8 Kbps) is reserved for voice traffic. The bandwidth requirement of the four voice calls is much higher $19.2 * 4 = 76.8$ Kbps. As shown in Table 5.7, the mean ETE delays of voice packets are unacceptably high in this scenario, since the FTP application generates a heavy load of competing data traffic and the bandwidth reserved for voice is insufficient.

Run	ETE Delay (s)			
	Call 1	Call 2	Call 3	Call 4
1	0.0544	0.0569	0.0564	0.0541
2	0.0512	0.0490	0.0466	0.0445
3	0.0607	0.0553	0.0570	0.0595
4	0.0440	0.0451	0.0438	0.0430
5	0.0478	0.0493	0.0512	0.0490

Mean ETE Delay: 0.0509 s

95% Confidence Interval for ETE Delay: (0.0483, 0.0535) s

Delay Variation: (0.00189, 0.00576) s²

Table 5.7: ETE Delays – VoIP over SLIP: Overloaded

To prevent such a situation from occurring, we implement the ACIS mechanism at the CPE. ACIS classifies voice calls into two categories – premium and regular. Both have a certain amount of upstream bandwidth reserved for them. However, the difference is that ACIS does not admit any voice calls as premium calls after all the premium bandwidth has been reserved by previous calls. We repeat the above experiment with 35 % of the upstream bandwidth (35 % of 128 Kbps = 44.8 Kbps) reserved for premium voice traffic and 15 % of the upstream bandwidth (15 % of 128 Kbps = 19.2 Kbps) reserved for regular voice traffic. Since only 44.8 Kbps is reserved for premium voice, the CPE can admit only two voice calls, which will reserve $19.2 * 2 = 38.4$ Kbps. The remaining 6.4

Kbps is not sufficient to accommodate a third call. Hence the remaining two calls are admitted as regular calls. The results are shown in Table 5.8.

Run	ETE Delay (s)			
	Call 1	Call 2	Call 3	Call 4
1	0.0136	0.0135	0.0673	0.0651
2	0.0136	0.0135	0.0657	0.0645
3	0.0136	0.0135	0.0839	0.0747
4	0.0135	0.0137	0.0606	0.0560
5	0.0138	0.0135	0.0615	0.0589

Mean ETE Delay: 0.0136 s (*Premium*) and 0.0658 s (*Regular*)
95% Confidence Interval for ETE Delay: (0.0135, 0.0136) s (*Premium*)
(0.0600, 0.0716) s (*Regular*)
Delay Variation: $(8.26 \times 10^{-7}, 13.93 \times 10^{-7}) s^2$ (*Premium*)
 $(0.00156, 0.00915) s^2$ (*Regular*)

Table 5.8: ETE Delays – ‘VoIP over SLIP’ with ACIS

Call 1 and Call 2 get admitted as premium calls and are ensured low ETE delays. Call 3 and Call 4 are admitted as regular calls and their packets experience unacceptably high ETE delays in the presence of competing FTP traffic. The total of 19.2 Kbps reserved for regular voice calls is insufficient to support two regular calls, since each of them requires 19.2 Kbps bandwidth. Figure 5.5 illustrates the improvement in performance obtained by implementing ACIS.

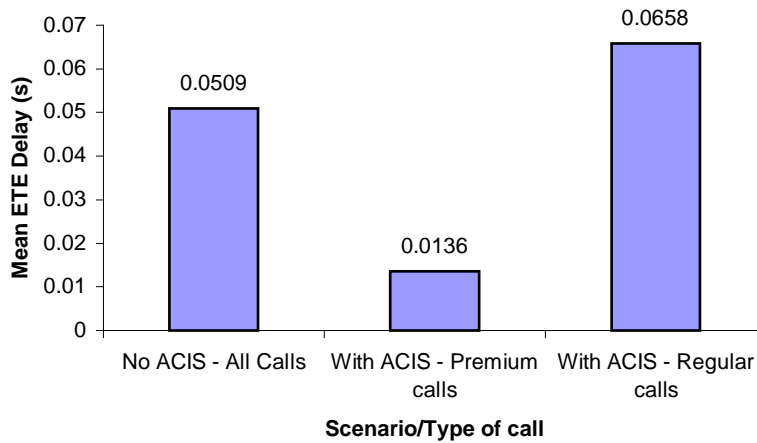
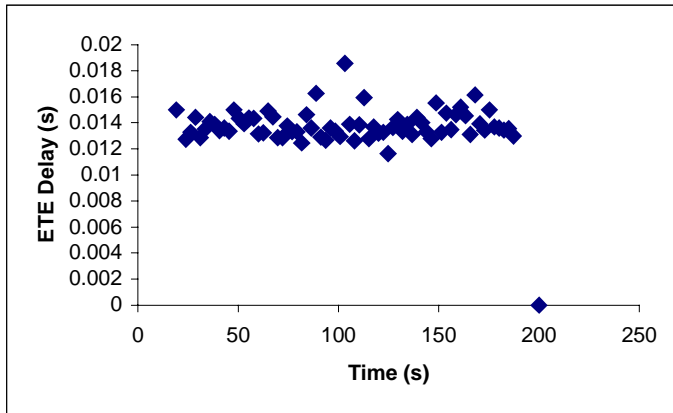


Figure 5.5: Performance of ACIS

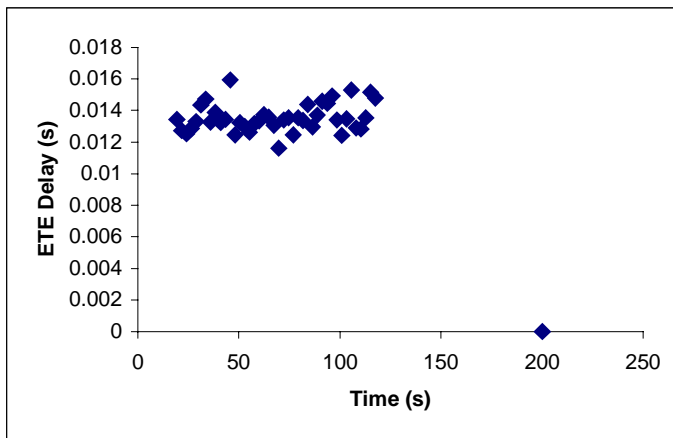
Next, we illustrate the ability of ACIS to upgrade ongoing calls from regular to premium of premium bandwidth becomes available while a regular call is in progress. Four end-hosts in the customer premises are engaged in voice calls. All use the same encoder and

the same number of frames per packet. There is sufficient premium bandwidth for only two premium calls. Figure 5.6 shows the ETE delays of the voice packets from each call.



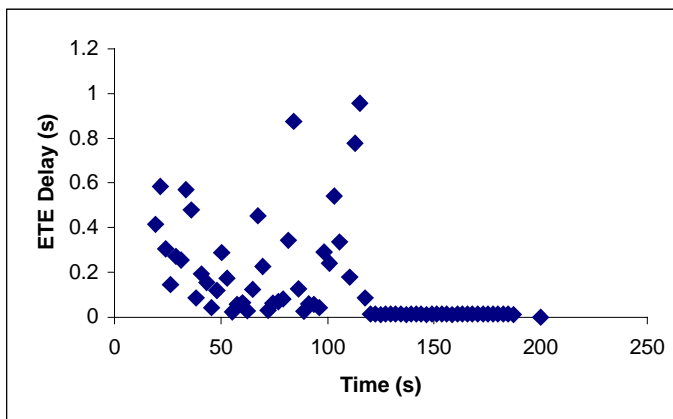
(a) Call 1

Call 1 is admitted as a premium call and it remains a premium call for its entire duration. The ETE delay is low throughout.



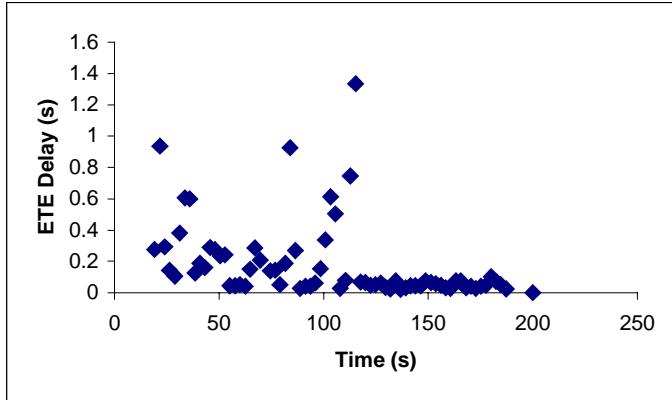
(b) Call 2

Call 2 is admitted as a premium call and it remains a premium call for its entire duration. The ETE delay is low throughout. However, Call 2 terminates earlier than the others.



(c) Call 3

Call 3 is admitted as a regular call, but it is upgraded to a premium call after Call 2 terminates. Note the sharp drop in ETE delay after 120 s, when the upgrade occurs.

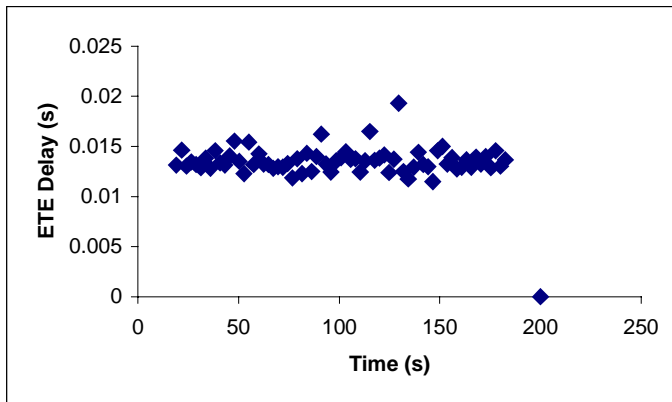


(d) Call 4

Call 4 is admitted as a regular call and remains a regular call for its entire duration. The ETE delay drops after Call 3 is upgraded because Call 4 has the entire regular call bandwidth to itself after the upgrade. However, the ETE delays are still somewhat higher than that for premium calls

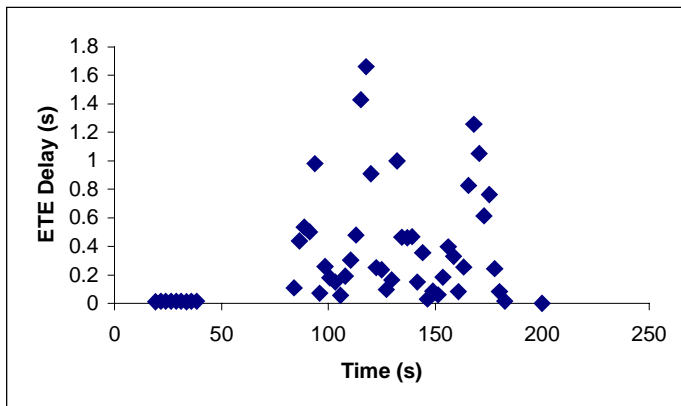
Figure 5.6: Upgrade of Regular calls to Premium calls

Finally, we illustrate the soft state maintained by ACIS. If a premium application crashes without de-allocating the reserved bandwidth, it can be preempted by new requests that arrive after a ‘maximum silence period’. Like our previous experiment, we have four end-hosts participating in voice calls with the same bit rate. There is sufficient bandwidth for only two premium calls. Figure 5.7 shows the ETE delays of the four voice calls.



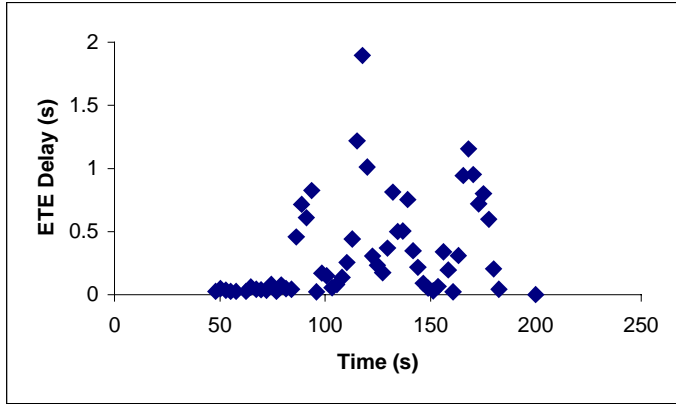
(a) Call 1

Call 1 is admitted as a premium call and it remains a premium call for its entire duration. The ETE delay is low throughout.



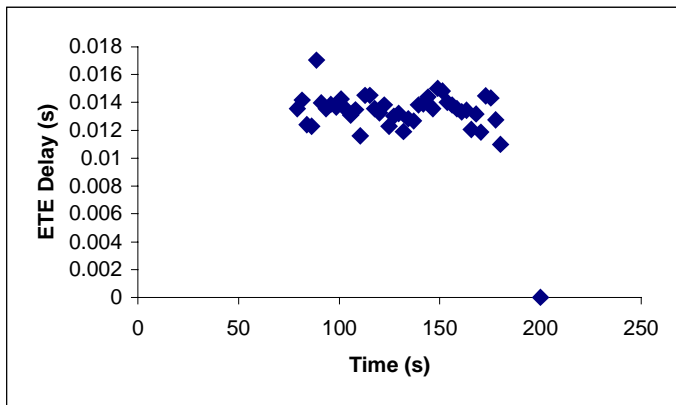
(b) Call 2

Call 2 is admitted as a premium call. However, at about 40 s, the application crashes. It recovers again at 85 s. Note the silence period during the time the application is down. The call’s premium status gets revoked when the application is down. On recovery, the voice call is classified as a regular call.



(c) Call 3

Call 3 is admitted as a regular call. The request for Call 3 arrives after Call 2 has crashed, but the maximum silence period for Call 2 has not yet expired. Hence Call 3 cannot preempt Call 2 from the premium category. Call 4 preempts Call 2, as we shall see next. ETE delays for Call 3 are reasonably low until Call 2 recovers. After that, the delays increase due to competition from Call 2.



(d) Call 4

Call 4 is admitted as a premium call. The request for Call 4 arrives after Call 2 has crashed and the maximum silence period for Call 2 has expired. Hence Call 4 preempts Call 2 from the premium category. Call 4 experiences low ETE delays.

Figure 5.7: Soft State in ACIS

5.2.5 Effect of Traffic Loads and Traffic Mixes

In our experiments so far, we have used the upload of a constant sized (2 MB) file, via FTP, to model our competing data traffic. The FTP application offers a very heavy load of data in this case. We now investigate the performance of our QoS mechanism in the presence of different loads of and different mixes data traffic.

We repeat our experiment consisting of four end-hosts engaged in voice calls with identical characteristics. There is sufficient bandwidth to accommodate only two premium calls. The other two calls are admitted as regular calls. Data traffic is generated by an end-host uploading a file via FTP. We now study the performance of voice traffic under various loads of data traffic. To accomplish this, the file size used by the FTP application is varied from 1 MB to 100 KB. Our results show that the premium calls are not affected by variations in load, since they always have sufficient bandwidth reserved for them. Even for regular calls, the ETE delay decreases only slightly as we reduce the file size. Of course, the time taken for completing the file transfer decreases considerably as we reduce the file sizes. Hence, with small files, the regular calls are disrupted for

shorter periods of time. When the file sizes are large, the regular calls are disrupted for longer periods. This can be seen in Figure 5.8, which shows the ETE delays of regular voice calls when competing with an FTP application that uploads files.

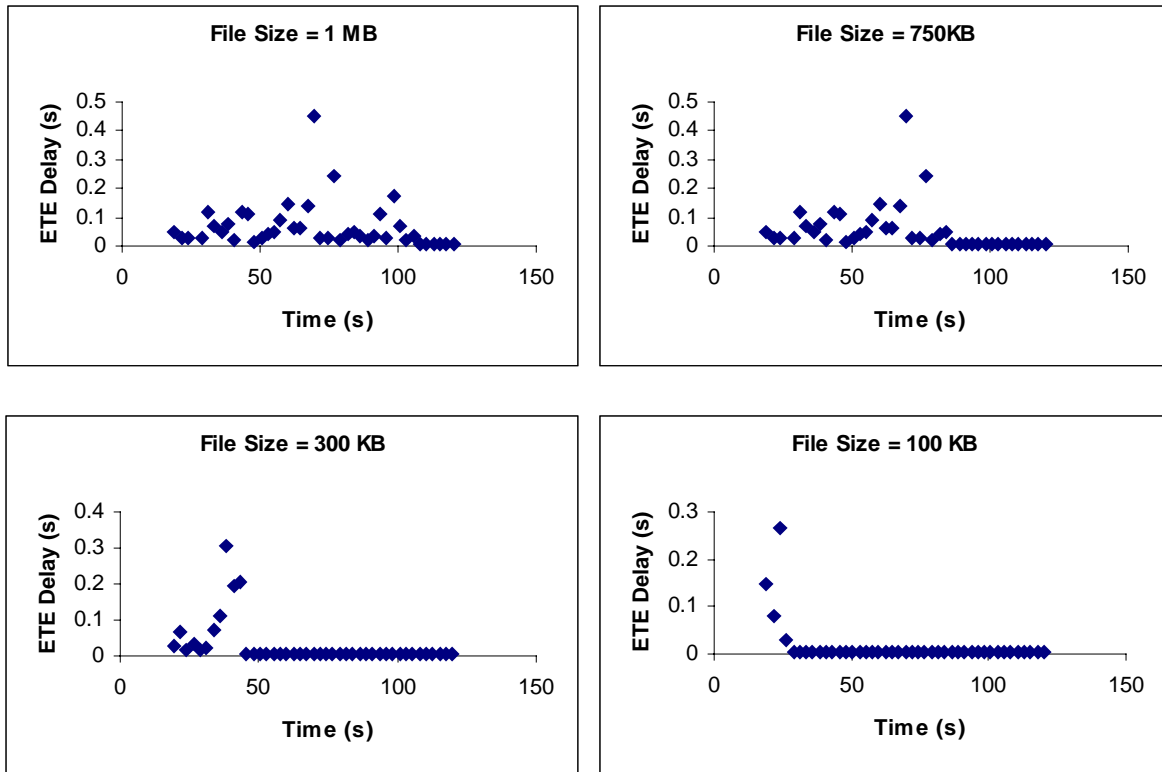
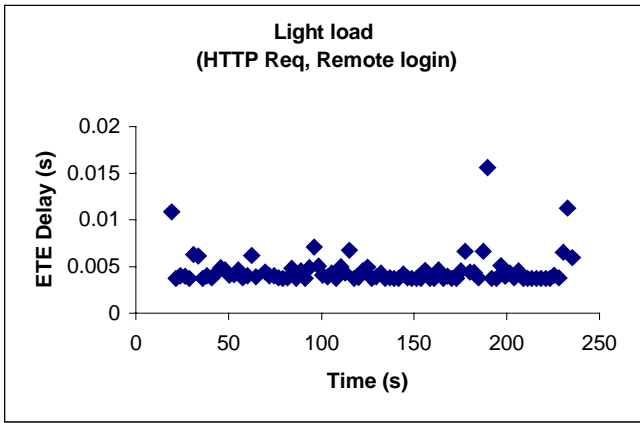


Figure 5.8: Effect of different sizes of file uploaded via FTP on regular voice

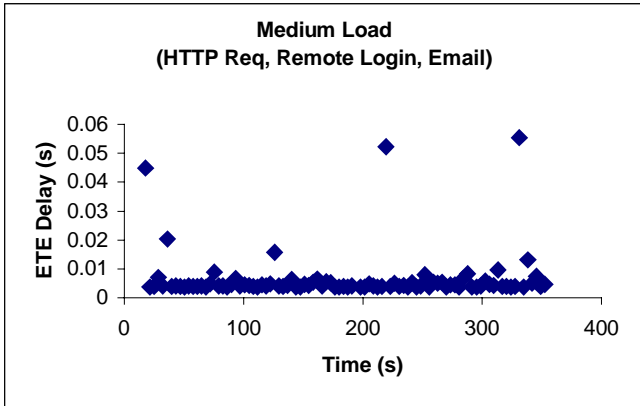
Finally, we perform an experiment to evaluate our QoS mechanism under different mixes of traffic. As usual, we have four end-hosts participating in voice calls with identical characteristics. The bandwidth reserved for premium traffic can support just two voice calls. The other two voice calls are admitted as regular calls. We evaluate our system under the following scenarios-

- i) Light traffic load – This is generated by a mix of HTTP and Remote Login applications. (Note that the upstream link mainly carries HTTP requests, which offer only a light load. The load offered by the responses may be heavier, but that would be in the downstream direction.)
- ii) Medium traffic load – This is generated by a mix of HTTP, Remote Login and Email traffic.
- iii) Heavy traffic load – This is generated by a mix of HTTP, Remote Login, Email and FTP traffic. The FTP traffic is generated by uploading a file in the upstream direction.

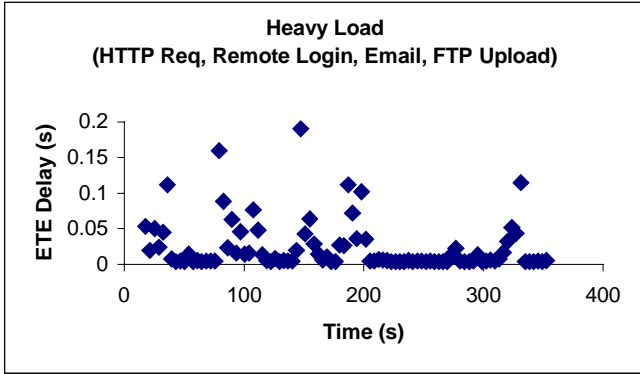
Our results show that premium calls experience low ETE delays for all types of traffic mixes, since they always have sufficient bandwidth reserved for them. The ETE delays for regular calls are shown in Figure 5.9.



Under light loads, regular calls perform pretty well. The ETE delays are low most of the time. HTTP request packets and remote login packets generate very light loads of competing data traffic.



Under medium loads too, the performance of regular call proves to be acceptable. The few spikes in the ETE delay represent times when an email is sent out.



Under heavy loads the ETE delays of the regular voice packets go up considerably. FTP uploads generate a large amount of data traffic that competes with and severely degrades the quality of regular voice calls.

Figure 5.9: Effect of different traffic mixes on regular voice

5.3 Chapter Summary

In this chapter, we presented experiments to evaluate the performance of our QoS mechanisms and analyzed the results of our experiments. We first investigated the ATM QoS model to understand the effectiveness of ATM QoS. Next we studied the VoIP model, starting with the best effort model and adding QoS mechanisms one by one. We observe the improvement in performance of our VoIP model resulting from the use of WFQ, fragmentation of large data packets and ACIS. Next, we perform experiments to illustrate the call upgrading capability and soft state features of ACIS. Finally, we investigate the performance of our traffic model under different loads and different mixes of traffic.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

Our objective in this thesis was to evaluate the feasibility of using VoIP instead of VoATM in future DSL deployments. VoIP in the DSL access network paves the way for end-to-end IP telephony, which is expected to be the technology of the future. We proposed and evaluated a VoIP architecture suitable for deployment in the DSL access network. Our results show that the performance of our architecture is comparable to that of the VoATM architecture. Hence, our VoIP architecture could replace the existing architecture to deliver comparable voice quality. In addition, it also offers the benefit of voice packet compatibility with the core network, which is already employing VoIP to carry voice inexpensively over data networks. Thus, the idea of end-to-end IP telephony is conceivable in the future.

As part of our architecture, we proposed the use of Weighted Fair Queuing to schedule voice and data packets over the upstream ADSL link. We also emphasized the importance of fragmenting large data packets before transmitting them over the upstream link, so as to avoid long packet transmission times. In addition, we highlighted the need for an admission algorithm for voice calls going over the upstream link, in order to prevent the load from exceeding the resource availability. With this aim we proposed a new admission control mechanism called Admission Control by Implicit Signaling (ACIS). We advocate the use of ACIS as it eliminates the need for the end-hosts to use an explicit signaling protocol for admission control at the CPE. Instead, it takes advantage of existing application layer signaling protocols by mapping them to the network layer. ACIS requires the CPE to maintain per-flow state information. While this characteristic is a scalability concern in the core network, we are concerned only with the customer premises. The number of flows is not likely to be large in this case and hence the performance of ACIS is not affected by this requirement. In fact, maintaining per-flow information for the premium calls enables us to provide excellent QoS and also ensures robustness in the event of failures and duplicate requests.

Our results show that ACIS is capable of providing excellent QoS to premium calls at all loads and acceptable QoS to regular calls at light or medium loads. Most ADSL users run applications such as web browsing, E-mail and streaming multimedia. These typically have low bit-rates in the upstream direction. For this reason, we do not reject any voice calls. Instead, when all the premium bandwidth has been reserved, we classify calls as regular calls. Since the typical ADSL user runs applications with low loads in the upstream direction, there is a fair chance that the regular calls receive an acceptable performance. So, when the voice traffic load is high, we attempt to take advantage of unused bandwidth that is reserved for the data. Even if the data traffic were to reclaim the borrowed bandwidth at some point, then only the regular calls would suffer. The customer would still be guaranteed excellent QoS for his or her premium calls.

We recommend the use of a lightweight protocol like PPP at Layer 2, instead of ATM. Due to the widespread existence of ATM in today's DSL networks, some existing VoIP implementations for DSL use ATM at Layer 2. We showed that using ATM at Layer 2 imposes an extremely high protocol overhead on voice packets. Moreover, VoIP does not rely on the use of ATM QoS. Thus we propose that future deployments of VoIP eliminate ATM altogether in favor of a lightweight protocol like PPP. The use of PPP at Layer 2 would also enable us to take advantage of fragmentation by Multilink PPP, which we showed is a better technique than fragmentation by IP.

In conclusion, we would like to say that though IP initially started out as a best-effort service, IP QoS mechanisms are now reaching maturity. In this scenario, VoATM no longer offers a huge advantage over VoIP in terms of providing QoS guarantees to voice. This being the case, it makes sense to use VoIP in future deployments of DSL, since IP is the universal protocol used throughout the Internet.

6.2 Future Work

In this section we provide some possible directions for future research in this area. It would be interesting to look at the performance of data traffic in our VoIP architecture. This would involve a study of TCP's ability to utilize the residual bandwidth (bandwidth unused by voice traffic) in the link. For connections with high round-trip times (RTT), TCP would react slowly to congestion at the CPE. The reason for this is that CPE is the first hop from the source and hence the source would be notified of a packet drop only after almost a whole RTT has elapsed. Until then, it would keep pumping data and adding to the congestion. As a result of the added congestion, the CPE would drop more packets and TCP's congestion window and bandwidth utilization will fall drastically. It is worth studying this problem in greater detail.

Another possibility is for extending the ideas employed by ACIS to suit a wide variety of situations. Currently, ACIS is designed for the specific task of admission control of voice calls over the upstream ADSL link. ACIS can be extended to work in conjunction with MBAC or a bandwidth broker. In its current implementation, ACIS reserves a fixed amount of bandwidth for premium voice traffic. However, this amount can be varied dynamically by using some MBAC scheme at the CPE to measure the load on the network. At any give time, the CPE would then be able to estimate whether the network can support any more calls. Alternatively, the CPE could contact a BB located somewhere in the service provider's network. The BB would keep track of resource availability and make admission control decisions based on it. The CPE would merely enforce the admission control decision made by the BB. These extensions to ACIS would make its deployment in larger networks more feasible.

The introduction of interactive video traffic over DSL opens another possibility for research. This would create a situation where voice, video and data traffic would coexist over DSL. Both voice and video are real-time applications and designing a QoS architecture to satisfy the requirements of the two would be an interesting task.

References

- [Alcatel] Alcatel White Paper, "Multiservice DLC/DSLAM for VoDSL".
- [Almesberger] T. Ferrari, W. Almesberger, J.L. Boudec, "SRP: a Scalable, Resource Reservation Protocol for the Internet," *Proc. Int'l Workshop on Quality of Service (IWQoS '98)*, Napa, CA, May 1998, pp. 107-116.
- [Armitage] G. Armitage, *Quality of Service in IP Networks*, MacMillan, 2000, pp. 169-183.
- [Bakiras] S. Bakiras, V.O.K. Li, "Quality of Service Support in Differentiated Service Packet Networks," *Proc. IEEE Intn'l Conf. Comm. 2001 (ICC 2001)*, vol. 8, 2001, pp. 2370-2374.
- [Balakrishnan] H. Balakrishnan, V.N. Padmanabhan, R.H. Katz, "The Effects of Asymmetry on TCP Performance," *Proc. ACM Mobicom '97*, September 1997.
- [Baldi] M. Baldi, F. Risso, "Efficiency of Packet Voice with Deterministic Delay," *IEEE Comm. Magazine*, vol. 38, Issue: 5, May 2000, pp. 170-177.
- [Bernet] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, "Integrated Services Operation over DiffServ Networks," IETF Internet Draft, June 1999.
- [Bianchi1] G. Bianchi, A. Capone, C. Petrioli, "Throughput Analysis of End-to-End Measurement-Based Admission Control in IP," *Proc. 19th Ann. Joint Conf. of the IEEE Computer and Comm. Soc. (INFOCOMM 2000)*, vol. 3, 2000, pp. 1461-1470.
- [Bianchi2] G. Bianchi, N. Blefari-Melazzi, M. Femminella, F. Pugini, "Performance Evaluation of a Measurement-Based Algorithm for Distributed Admission Control in a DiffServ Framework," *Proc. IEEE Global Telecommunications Conf. (GLOBECOM '01)*, vol. 3, 2001, pp. 1886-1891.
- [Bianchi3] G. Bianchi, N. Blefari-Melazzi, "Admission Control over Assured Forwarding PHBs: a Way to Provide Service Accuracy in a DiffServ Framework," *Proc. IEEE Global Telecommunications Conf. (GLOBECOM '01)*, vol. 4, 2001, pp. 2561-2565.
- [Blake] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," IETF RFC 2475, December 1998.
- [Borgonovo] F. Borgonovo, A. Capone, L. Fratta, M. Marchese, C. Petrioli, "PCP: A Bandwidth Guaranteed Transport Service for IP networks," *Proc. IEEE Int'l Conf. Comm. (ICC 1999)*, vol.1, 1999, pp. 671-675.

[Braden1] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: An Overview," IETF RFC 1633, June 1994.

[Braden2] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," IETF RFC 2205, September 1997.

[Breslau] L. Breslau, E.W. Knightly, S. Shenker, I. Stoica, H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," *Proc. ACM SIGCOMM 2000*, pp. 57-69.

[Büchli] M. Büchli, D. De Vleeschauwer, J. Janssen, A. Van Moffaert, G.H. Petit, "Resource Allocation and Management in Diffserv Networks for IP Telephony," *11th Int'l workshop on Network and Operating Systems support for digital audio and video*, pp. 33-39, January 2001.

[Caputo] R. Caputo, *Cisco Packetized Voice and Data Integration*, McGraw Hill, 1999, pp. 64-109.

[Casner] S. Casner, V. Jacobson, "Compressing IP/UDP/RTP headers for Low-Speed Serial Links," IETF Internet Draft, August 1998.

[Cetinkaya] C. Cetinkaya, V. Kanodia, E. Knightly, "Scalable Services via Egress Admission Control," *IEEE Trans. on Multimedia*, vol. 3, Issue: 1, March 2001, pp: 69-81.

[Coppercom1] CopperCom Technology White Paper, "Mastering Voice Over DSL: Network Architecture".

[Coppercom2] CopperCom Technology White Paper, "Seven Essentials of a Local Exchange Softswitch System, Part 4: Packet Voice over Broadband".

[CopperMount] Copper Mountain Networks White Paper, "Voice Over DSL".

[Detti] A. Detti, M. Listanti, S. Salsano, L. Veltri, "Supporting RSVP in a Differentiated Service Domain: an Architectural Framework and a Scalability Analysis," *Proc. IEEE Int'l Conf. Comm. (ICC '99)*, vol.1, 1999, pp. 204-210.

[Demers] A. Demers, S. Keshav, S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," *Proc. ACM SIGCOMM '89*, September 1989.

[DSL] DSL Forum Website: www.adsl.com

[Elek] V. Elek, G. Karlsson, R. Ronngren, "Admission Control Based on End-to-End Measurements," *Proc. 19th Ann. Joint Conf. of the IEEE Computer and Comm. Soc. (INFOCOM 2000)*, vol. 2, 2000, pp. 623-630.

[El-Khatib] K. El-Khatib, G. Luo, G. Bochmann, "Multiplexing Scheme for RTP Flows between Access Routers," IETF Internet Draft, July 1999.

[Fineberg] V. Fineberg, "A Practical Architecture for Implementing End-to-End QoS in an IP Network," *IEEE Comm. Magazine*, vol. 40 Issue: 1, Jan. 2002, pp. 122-130.

[Garcia] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, 1994.

[Hai] D. Hai, S. Vuong, "Dynamic-Distributed Differentiated Service for Multimedia Applications," *Proc. Int'l Conf. Dependable Systems and Networks (DSN 2000)*, 2000, pp. 586-594.

[Heinanen] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "An Assured Forwarding PHB Group," IETF RFC 2597, June 1999.

[Hill] R. Hill, H.T. Kung, "A Diff-Serv Enhanced Admission Control Scheme," *Proc. IEEE Global Telecommunications Conf.*, (GLOBECOM '01), vol. 4, 2001, pp. 2549-2555.

[Hou] Y. Hou, Z. Duan, Z. Zhang, T. Chujo, "Providing Scalable Support for Multiple QoS Guarantees: Architecture and Mechanisms," *Proc. IEEE Int'l Conf. Comm.* (ICC 2001) vol.7, 2001, pp. 2115-2122.

[Ibe] O. Ibe, *Converged Network Architectures*, John Wiley & Sons Inc., 2002, pp. 292-301

[Jacobson1] V. Jacobson, "Compressing TCP/IP Headers for low speed serial links," IETF RFC 1144, February 1990.

[Jacobson2] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB," IETF RFC 2598, June 1999.

[Jamin] S. Jamin, P. Danzig, S. Shenker, L. Zhang, "A Measurement-Based Admission Control Algorithm for Integrated Service Packet Networks," *IEEE/ACM Trans. on Networking*, vol. 5, Issue: 1, Feb. 1997, pp: 56 -70.

[Jia] Y. Jia, M. Chen, "A Novel Architecture of Providing Guaranteed Services for Differentiated Services Network," *Proc. Int'l Conf. Trends in Comm.* (EUROCON '2001), vol.2, Volume: 2, 2001, pp. 492-495.

[Kalampoukas] L. Kalampoukas, A. Varma, K.K. Ramakrishnan, "Improving TCP Throughput over Two-Way Asymmetric Links: Analysis and Solutions," *Proc. Sigmetrics '98*, June 1998.

[Klepec] B. Klepec, A. Kos, "Performance of VoIP Applications in a Simple Differentiated Service Network," *Proc. Int'l Conf. Trends in Comm.* (EUROCON '2001), vol. 1, 2001, pp. 241-217.

[Lee] H. Lee, H. Kwon, Y. Nemoto, "Guaranteeing Multiple QoSs in Differentiated Services Internet", *7th Int'l Conf. Parallel and Distributed Systems: Workshops*, 2000, pp. 233-238.

[Lucent] Lucent Technologies White Paper, "Transforming Copper to Gold, High Revenue Voice-over-DSL Services".

[Mameli] R. Mameli, S. Salsano, "Use of COPS for Intserv operations over DiffServ: Architectural issues, Protocol design and Test-bed implementation," *Proc. IEEE Int'l Conf. Comm.* (ICC 2001), vol.10, 2001, pp. 3265-3270.

[Mase] K. Mase, Y. Toyama, A. A. Bilhaj, Y. Suda, "QoS Management for VoIP Networks with Edge-to-Edge Admission Control," *Proc. IEEE Global Telecommunications Conf.* (GLOBECOM '01), vol. 4, 2001, pp. 2556-2560.

[McCann] J. McCann, S. Deering, J. Mogul, "Path MTU Discovery for IP version 6," IETF RFC 1981, August 1996.

[Mogul] J. Mogul, S. Deering, "Path MTU Discovery," IETF RFC 1191, November 1990.

[Mortier] R. Mortier, I. Pratt, C. Clark, S. Crosby, "Implicit Admission Control," *IEEE J. on Selected Areas in Comm.*, vol. 18, Issue: 12, Dec. 2000, pp. 2629-2639.

[Opnet] Online Documentation, OPNET Modeler 7.0.

[Paradyne] Paradyne Networks document, "The DSL Sourcebook".

[Peterson] L. Peterson, B. Davie, *Computer Networks*, Morgan Kaufmann, 2000, pp. 252

[Phanse] K. Phanse, L. A. DaSilva, K. Kidambi, "Characterization of Performance of TCP/IP over PPP and ATM over Asymmetric Links," *Proc. 9th Int'l Conf. on Computer Comm. and Networks* (ICCCN 2000), October 2000, Las Vegas, NV, pp. 334-339.

[Qi] Z. Qi, D. Zhicheng, G. Hong, Z. Xianghua, "A Novel Admission Control Scheme for Internet Applications," *Proc. Int'l Conf. on Comm. Technology* (WCC - ICCT 2000), vol.1, 2000, pp. 683-686.

[Qian] W. Qian, W. Jing, C. Shiduan, M. Jian, "A PHB Reservation Mechanism in Diffserv Network", *Proc. Int'l Conf. Comm. Technology*, (WCC - ICCT 2000), vol.1, 2000, pp. 424-427.

[Rhee] W. Rhee, H. Kim, K. Park, K. Kim, S. Kim, "Two Phase Edge-to-Edge Distributed Measurement Based Admission Control Mechanism in large IP networks," *IEEE Global Telecommunications Conf. (GLOBECOM '01)*, vol. 4, 2001, pp. 2571-2575.

[Romkey] J. Romkey, "A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP," IETF RFC 1055, June 1988.

[Schulzrinne] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol," IETF RFC 2543, March 1999.

[Shenker] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service," IETF RFC 2212, September 1997.

[Simpson] W. Simpson, "The Point-to-Point Protocol (PPP)," STD 51. IETF RFC 1661, July 1994.

[Sklower] K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti, "The PPP Multilink Protocol (MP)," IETF RFC 1990, August 1996.

[Stoica] I. Stoica, H. Zhang, "Providing Guaranteed Services Without Per Flow Management," *Proc. ACM SIGCOMM '99*, Cambridge, Massachusetts, September 1999, pp. 81-94.

[Tajima] K. Tajima, M. Hashmani, M. Yoshida, "A Resource Management Architecture over Differentiated Services Domains for Guarantee of Bandwidth, Delay and Jitter," *Proc. IEEE/AFCEA Information Systems for Enhanced Public Safety and Security (EUROCOMM 2000)*, 2000, pp. 242-249.

[Tanenbaum] A. Tanenbaum, *Computer Networks*, Prentice Hall 1996, pp. 463

[Tang] Y. Tang, C. Mao, H. Ou, X. Yang, X. Li, "Research on Adaptive IP QoS Management Framework," *Proc. 6th Int'l Conf. on Computer Supported Cooperative Work in Design*, 2001, pp. 149-152.

[TollBridge] TollBridge Technologies White Paper, "TollVoice Architecture and Local Exchange Gateways: Enabling Technologies for Voice over Broadband".

[Tyagi] A. Tyagi, J.K. Muppala, H. De Meer, "VoIP support on Differentiated Services using Expedited Forwarding," *Proc. IEEE Int'l Conf. Performance, Computing, and Comm. (IPCCC '00)*, 2000, pp. 574-580.

[Vutukury] S. Vutukury, J. Garcia-Luna-Aceves, "A Scalable Architecture for Providing Deterministic Guarantees," *Proc. 8th Int'l Conf. Computer Comm. and Networks (ICCCN 1999)*, 1999, pp. 91-96.

[Wang] F. Wang, P. Mohapatra, "An Efficient Bandwidth Management Scheme for Real-Time Internet Applications," *Proc. Int'l Symp. Intelligent Multimedia, Video and Speech Processing*, 2001, pp. 469-472.

[Wroclawski] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," IETF RFC 2211, September 1997.

[Wu] D. Wu, Y. Hou, Z. Zhang, H. Chao, T. Hamada, T. Taniguchi, "On Implementation Architecture for Achieving QoS Provisioning in Integrated Services Networks," *Proc. IEEE Int'l Conf. on Comm. (ICC '99)*, vol.1, 1999, pp. 461-468.

[Zhang] G. Zhang and H.T. Mouftah, "End-to-End QoS Guarantees Over Diffserv," *Proc. 6th IEEE Symp. Networks, Computers and Comm.*, 2001, pp. 302-309.

Glossary of Acronyms

AAL – ATM Adaptation Layer
ABR – Available Bit Rate
ACIS – Admission Control by Implicit Signaling
ADPCM – Adaptive Differential Pulse Code Modulation
ADSL – Asymmetric Digital Subscriber Line
AF – Assured Forwarding
ATM – Asynchronous Transfer Mode
BA – Behavior Aggregates
BB – Bandwidth Broker
CAC – Call Admission Control
CBR – Constant Bit Rate
CDV – Cell Delay Variation
CELP – Code Excited Linear Predictor
CID – Channel Identifier
CL – Controlled Load
CLR – Cell Loss Ratio
CO – Central Office
CPCS – Common Part Convergence Sublayer
CPE – Customer Premises Equipment
CPI – Common Part Indicator
CS – Convergence Sublayer
CTD – Cell Transfer Delay
CU – Currently Unused
DPRP – DiffServ PHB Reservation Protocol
DPS – Dynamic Packet States
DSCP – Differentiated Services Code Point
DSL – Digital Subscriber Line
DSLAM – DSL Access Multiplexer
DSP – Digital Signal Processor
EF – Expedited Forwarding
EMBAC – Edge-to-Edge Measurement Based Admission Control
ETE – End to end
FR – Frame Relay
FTTC – Fiber To The Curb
GS – Guaranteed Service
HDSL – High-bit-rate DSL
HEC – Header Error Control
IAD – Integrated Access Device
IDSL – ISDN DSL
IP – Internet Protocol
ISDN – Integrated Services Digital Network
LI – Length Identifier
MBAC – Measurement Based Admission Control
MBS – Maximum Burst Size

MCR – Mean Cell Rate
MDSL – Multi-rate DSL
MP – Multilink PPP
NFR – Number of Frames
nrt-VBR – non real-time Variable Bit Rate
NSP – Network Service Provider
PCR – Peak Cell Rate
PDU – Protocol Data Unit
PHB – Per Hop Behavior
PMD – Physical Medium Dependent
PPP – Point-to-Point Protocol
PSTN – Public Switched Telephone Network
PT – Payload Type
QoS – Quality of Service
RADSL – Rate Adaptive DSL
RSVP – Resource reSerVation Protocol
RTCP – Real-time Transfer Control Protocol
RTP – Real-time Transfer Protocol
RTT – Round Trip Time
rt-VBR – real-time Variable Bit Rate
SAR – Segmentation and Reassembly
SCR – Sustained Cell Rate
SDSL – Symmetric DSL
SIP – Session Initiation Protocol
SLIP – Serial Line Internet Protocol
SRRP – Sender-initiated Resource Reservation Protocol
TC – Transmission Convergence
TCP – Transmission Control Protocol
TOS – Type of Service
TTL – Time To Live
UBR – Unspecified Bit Rate
UDP – User Datagram Protocol
UPC – Usage Parameter Control
UUI – User to User Indicator
VAD – Voice Activity Detection
VC – Virtual Circuit
VDSL – Very-high-data-rate DSL
VoATM – Voice over ATM
VoDSL – Voice over DSL
VoIP – Voice over IP
WFQ –Weighted Fair Queuing
WRR – Weighted Round Robin

Vitae

Abhishek Ram hails from Chennai, India. He earned his Bachelor's degree in Computer Engineering from Karnataka Regional Engineering College, Surathkal, India. He then joined the Master's program in Computer Science at Virginia Tech in Blacksburg, Virginia. After his graduation, he shall be working for Microsoft Corporation at Redmond, Washington.