ASSESSING THE QUALITY OF MULTIMEDIA COMMUNICATIONS OVER INTERNET BACKBONE NETWORKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Athina P. Markopoulou
October 2002

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Fouad Tobagi
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

David Cheriton

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Bernd Girod

Approved for the University Committee on Graduate Studies:

Abstract

The Internet is evolving to become the universal network infrastructure that supports all traffic types and communications needs. Among them, interactive voice and video applications, such as Voice over IP (VoIP) and Video Conferencing, are of great interest. As these applications become increasingly popular, users expect higher quality. However, they both have strict requirements both in terms of loss (for good speech and video quality) and delay (for interactivity). If the Internet is to eventually replace traditional networks, such as the telephone network for voice communications, it has to provide service at the same quality levels. Our objective in this study is to assess to what extent today's Internet meets this expectation. In the process, we identify sources of impairments and we comment on the possible causes and improvements.

We first study loss and delay measurements collected for many days, over the wide-area backbone networks of several service providers across the US. We then study the performance of voice communications over these networks, in terms of user perceived quality measures. While backbone networks are generally believed to be sufficiently provisioned for data applications, we find that this is not always the case for voice and video traffic. Loss and delay characteristics are not consistent across all backbone networks. Some backbone networks exhibit fairly good characteristics and may offer communication at good quality levels, leading to a confirmation that packet voice is a sound approach. Other backbone networks exhibit undesirable characteristics, such as high delay jitter, periodic delay patterns and long loss periods. At the receiving end, the playout scheduling algorithm plays an important role in

smoothing out the delay jitter caused by the network. We examine existing playout scheduling algorithms and we propose new algorithms that take into account the delay variability observed in the meausurements and the loss - delay tradeoff in the overall VoIP performance. We finally study the performance of video traffic with low delay constraints and we find it to share fate with voice over the same backbone networks.

Most of the problems identified on these backbone networks seem more related to reliability, network protocols and router operation, rather than to traffic load. Therefore, they should be better understood and prevented from occuring. As long as such problems exist and as long as they remain below a certain magnitude, measures can be taken at the end-systems to mitigate their effect, such as playout scheduling and multipath streaming.

Acknowledgments

This thesis would not have been possible without the support and guidance of many people.

First, I would like to thank my adviser Fouad Tobagi, for giving me the opportunity to work with him, for guiding me toward interesting projects and for his involvement and interest in my progress. I also learned a lot from his methodic and critical approach to research problems. I would like to thank my secondary adviser David Cheriton, for his influence toward problems that eventually became my thesis topic and for his constructive feedback. I also value the opportunity he gave me to work with him as in intern in Cisco and the arrangement of a research grant that supported me for a year. I would also like to thank Bernd Girod for motivating me to learn more about multimedia through his class, talks and interaction with his group as well as for his concrete feedback. I would like to thank all three members of my committee not only for having the patience to read this thesis, but also for having already contributed to its content.

This work would not have been possible without the measurements provided to us by RouteScience Technologies Inc. The measurements have been the main asset of this study and I appreciate the opportunity I was given, by both RouteScience and my adviser, to work with them. I would also like to thank the RouteScience team for their research grant that partially supported this work. Mansour Karam has been my mentor and friend, since he was a senior student in the group and later in the project.

I would also like to thank my fellow graduate students in my research group at Stanford. Amit Vyas worked with me with great enthusiasm in the early stage of the measurement project. Chuck Fraleigh devoted much of his time taking care of system administration and making the work for all of us possible. I also want to thank Chuck, Cristina Hristea, Wael Noureddine, Benjamin Chen, Jose-Miguel Pulido for valuable discussions and their feedback. Also, Charlie Orgish has often (been) volunteered to solve urgent problems, including systems problems.

I have learned a lot about voice and video communications, ranging from specific tools to research directions, from the guidance provided by many people. Sangeun Han, John Apostolopoulos, Stephen Wolf, Stephan Wenger and Alan Clark were eager to share with me their experience. I am particularly grateful to Jacob Chakareski and Yi Liang for their help on the part concerning path diversity.

On the personal side, I would first like to thank my parents: my mother for her vision and support and my father for his influence through his love for mathematics. Looking back, I now appreciate even more his structured way of thinking, which is at least comparable to many people with higher degrees I have worked with, myself included. I would also like to thank Kristian Jessen for his support during the last year. Also Katerina Argyraki, Maria Gkatziani, Christina Fragouli, Persefoni Kyritsi, Valeria Bertacco and many other fellow graduate students especially from the greek community, for their friendship throughout the years. Overall, Stanford has been an amazing place that had definitely an impact on my personal growth. Academic competence is only one of the qualities of the people that I had the luck to meet here. The cultural diversity made the experience even more valuable.

As this work comes closer to completion, I start looking forward to some good work in the years to come, that could hopefully be both useful and intellectually pleasing. Therefore, I see this dissertation more as a valuable learning process rather than as the end of my efforts.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Network Convergence

In the past decades we witnessed a significant increase in the size of the Internet (in terms of number of hosts and users, geographical reach and connectivity) as well as in the richness of applications. While initially, only data applications, such as email and ftp, were supported, in the last decade, we have seen the creation and use of various new services such as multimedia interactive applications and streaming, content distribution, games and business transactions. Many new services were enabled by technological advances such as the increase in bandwidth, which is in the order of Gbps in the backbone, hundreds of Mbps in the local area networks and Mbps in the last hop to the user. The new services were also facilitated by the introduction of the World Wide Web (WWW) which made the Internet widely popular. Also the increase in connectivity, geographical reach and the advances in access networks, including wireless, provided global access and made the use of the network even more attractive.

Today, the Internet is evolving into the universal, integrated communication infrastructure that supports all types of applications and communication needs. It already serves a large part of our daily activities, including information, entertainment, telecommunications, commercial and bureaucratic transactions.

There are two reasons for this evolution: one has to do with the user, the other has to do with the network provider. From a user perspective, the Internet is attractive due to its global reach and the relatively low fee for access to a variety of services. Today, Internet services are free and thus sometimes preferred even at low quality. However, as we are getting more and more used to those services, our expectations increase. Furthermore, as the Internet evolves from an experimental research project to a commercial network, new business models arise and fees may be associated with some services. Even in that case, the fees will be relatively low, compared, for example, to today's long distance calls. From a network provider point's of view, the network convergence is attractive due to the bandwidth efficiency of packet-switching and the simplicity and scalability of the IP architecture. Therefore, there is a long term motivation for creating a single, efficient, integrated communication infrastructure that will provide all existing services, and even new services by combining different traffic types.

However, many new applications have stricter requirements than the traditional best-effort data applications, in terms of availability, quality and security. For example, business transactions require reliability, security and low latency. Real-time multimedia applications require low latency and reasonably good quality. Traditional telephony services require very high availability, good speech quality and low latency, at levels comparable to those provided by the telephone network. In order for the Internet to constitute an attractive alternative to the traditional networks, or even to work in conjunction with them, it is critical that it meets the above requirements at similar or better quality levels.

Interactive Multimedia Applications

The focus of this work is on interactive multimedia applications, such as telephony, video telephony and video conferencing. Internet telephony is an application with a huge potential market, currently served by the traditional telephone companies. Video conferencing is also becoming increasingly popular for reporting news from remote locations, for business meetings with participants in remotes sites and for virtual classrooms.

Voice characteristics. Packet voice has been studied since the early 1980s in the context of local area networks. Many of the issues addressed then (e.g. multiplexing of voice sources or voice with data, tradeoff between the multiplexing gain and the loss or delay jitter induced to voice) are still relevant today. In the context of IP networks, telephone services are known as "Voice over IP" (or "VoIP"). The ITU-T standards used for compression are G.711, G.723.1 and G.729, leading to data rates from 5.33 to 64 Kbps. The encoded speech is packetized using the Real Time Protocol (or RTP, [85]). The resulting VoIP flow consists of packets of small fixed size sent at fixed intervals.

Voice requirements. In order to maintain a conversation at good quality levels, there are requirements in terms of loss, delay and delay variability. First, low loss is needed in order to achieve good speech quality and intelligibility. However, some amount of loss can be tolerated, especially if the decoder uses a concealment technique to produce replacement for lost packets. In that case, loss rates up to 10% can be tolerated, as reported in [38, 64]. Second, in order to maintain an interactive conversation, the total end-to-end delay should remain below 150 ms or even lower for highly interactive conversations, as reported in [58]. The ITU-T standard G.114, [39], recommends a maximum end-to-end delay below 150 ms for acceptable quality; the delay range 150-400 ms is considered acceptable although the annoyance is perceived; delay more than 400 ms is considered intolerable for effective communication. Another effect of large end-to-end delays is the annoyance caused by echoes when no echo cancellation is present in the system. Finally, low or no delay variation (also referred to as jitter) is needed in order to provide a continuous speech playout.

Video characteristics. Video Telephony and Video Conferencing use the H.261, H.263, H.263v.2 and H.264 (until recently known as H.26L) standards that are designed specifically to meet the delay constraints and result to relatively low data rates, i.e. in the order of tens or hundreds of Kbps. Encoding is done at real time. Low frame rates (e.g. 10 fps) and small images (QCIF or CIF) are typically used. The encoded video sequence is packetized using RTP and the appropriate RTP profile. The resulting video stream is, in general, more variable than a voice flow.

Video requirements. The requirements for high quality interactive video applications, are similar to those for voice applications. First, low or no loss is desirable to maintain good video quality. Furthermore, the temporal and spatial dependencies in a compressed video stream amplify the effect of loss through error propagation and increase the sensitivity to the loss of critical information. However, the receiver is often able to conceal missing parts from the received information. The tolerable amount of packet loss varies with the content, encoding and error control and recovery mechanisms used. Second, in order to maintain good interactivity as well as lip synchronization between audio and video, the delay for the video stream should be at similar levels as for voice. Finally, low or no delay jitter is required to provide a continuous video playout.

In summary, interactive multimedia applications need: (i) low or no loss for good speech and video quality (ii) low delay for interactive communication and (iii) low or no delay variability for continuous playout. However, the Internet today cannot guarantee any of the above requirements. Figure 1.1 shows the components of the Voice/Video over IP system under study and summarizes the impairments introduced.

Impairments Introduced by the Network

Loss may be due to congestion in the network leading to packets getting dropped at switches and routers, or failure of network components leading to a reconfiguration of the network. In particular bursty loss, which is the common case in the Internet, is difficult to conceal and has a major effect on perceived quality. As far as delay is concerned, we distinguish the fixed part from the variable part of the delay. The fixed part comprises packet transmission time over, and propagation time across the links in the path, and any fixed transit delay that may

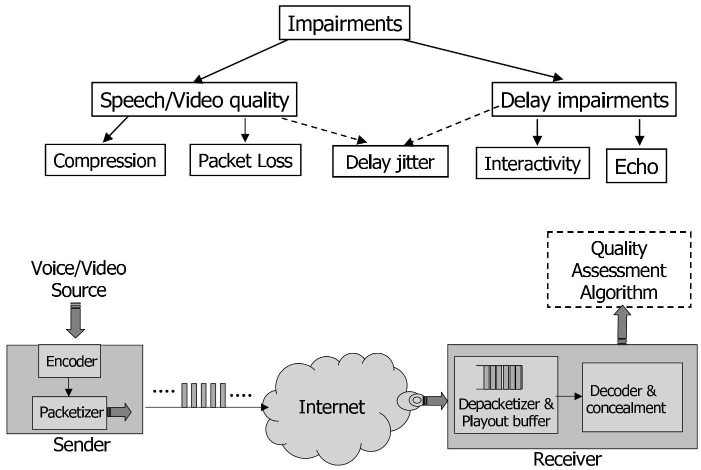


Figure 1.1: Voice/Video over IP System considered and Quality Impairments

be incurred through network elements encountered in the path. The variable part of the delay comprises queuing delays incurred within network elements and other possible delays introduced by the operation of the network elements (e.g. delay components in a router are discussed in [72]). Even without variations in the end-to-end delay, the magnitude of the latter is important because of its effect on interactivity and echo. As mentioned above, the total end-to-end delay should remain below 150 ms or even lower for highly interactive conversations. Unfortunately, there is little that one can do about the fixed part of the delay. On the other hand, delay jitter can be dealt with at the receiver, up to some extent, by means of playout scheduling. Even with playout scheduling, delay jitter results in some impairment, which is either additional delay or additional speech/video distortion.

Question

Given the incentive for providing VoIP/VIP services over the Internet and given the quality impairments introduced today by the Internet, the following questions become then important. What level of quality is actually provided today in the Internet to those interactive multimedia applications? How extensive are the loss, delay and delay jitter impairments and how bad is their effect? Can these impairments be dealt with in the network? Can they be dealt with at the destination or with collaboration between the source and the destination? What measures can one take, if any, in order to make this possible?

Dealing with Impairments

The responsibility for achieving good quality is shared between the network and the end-systems. On one hand, the impairments are introduced in the network, therefore they should be prevented from happening there. On the other hand, there are measures that can be taken to overcome these problems at the end-systems, i.e. at the source, at the destination, at a proxy server or with a collaboration between them. The effectiveness of these measures is limited to certain ranges of network conditions. Furthermore, many problems can be handled efficiently by making the applications network aware.

Measures in the network. Network architectures have been proposed in the recent years for providing Quality of Service (QoS) in the Internet. The Integrated Services, proposed in [86], provides mechanisms for adding circuit-functionality to the packet-switched Internet. The Differentiated Services architecture, proposed in [3], enables service differentiation between different traffic classes. Voice traffic, perhaps the most important and sensitive traffic class, could have made use of those mechanisms. However, the proposed architectures introduce significant complexity and are difficult to configure, thus they are not widely used. In fact, in high bandwidth environments, (over)provisioning approaches work sufficiently well, as explained in [24]. In limited bandwidth environments, simple mechanisms such as priority-based schemes may be sufficient for voice, as argued in [55].

Providing Quality-of-Service (QoS) to video streams is more difficult due to the higher data rates. Unlike voice, giving strict priority to video traffic could starve the rest of the traffic. Instead, another network differentiation mechanism has been proposed, in order to protect the most important parts of a video stream during congestion periods: priority dropping combined with scalable coding. Example of studies on this topic are [31, 57, 71]. However intuitive such an approach may be, it did not get widely used in practice.

Measures at the end-systems. In practice, much of the responsibility for mitigating the effect of loss, delay and delay jitter is delegated to the end-systems. The issues are similar for both voice and video, although video is a larger problem space, due to the spatial and temporal dependencies in the bitstream.

In order to deal with packet loss in the case of voice applications, error control and recovery mechanisms have been proposed. They include error concealment, forward error correction (FEC) and retransmissions when allowed by the delay budget. A survey of such mechanisms for voice can be found in [73].

In order to deal with loss for video applications, a large number of error control and recovery mechanisms has been proposed in the standards including transport-level mechanisms (such as various packetization schemes, feedback, retransmissions, forward error correction), error-resilient encoding, decoder error concealment and encoder-decoder interactive error control. The tight delay constraint of video conferencing limits the choice of the applicable mechanisms. For example, retransmissions or long receiver buffers are typically not allowed. A good survey of error control and concealment mechanisms for video communications can be found in [101].

As far as absolute end-to-end delay is concerned, there is little one can do for high end-to-end delay, apart from keeping all delay components low. There is little control in today's Internet that one in general has about the routes packets take to reach their destinations. Proposals arguing in favor of some kind of route control available to the source include [17, 12, 79]. Other approaches rely on relay servers at the application level for route control, and as an extension, for implementing path diversity, [2, 63].

In order to absorb delay jitter and achieve a smooth playout of speech, extensive work has been conducted on voice playout scheduling. The scheduling of packet playout may be fixed, whereby a constant end-to-end delay target is enforced on all packets. Packets that exceed the target delay are dropped. Alternatively, the scheduling of packet playout may be adaptive, whereby the target delay is allowed to vary over time. Studies on adaptive playout scheduling include [22, 64, 67, 69, 76, 77, 78, 88]. Extending the same problem to include speech, audio and video, work has been conducted on adaptive media playout, [53, 54, 95]. Although adaptive playout is relatively straightforward for video (by varying the rate of displaying frames), in the context of video conferencing, it is constrained by the lip synchronized voice, for which variable playout rate leads to a pitch change, as discussed in [64, 53].

Finally, there is ongoing work at the interface between application and network that makes the application network-adaptive. Such techniques include (i) adaptive media playout that was mentioned above (ii) rate-distortion optimized packet scheduling at the sender, at the receiver or at an intermediate proxy, [13] (iii) path diversity to be exploited for a number of decisions at the end-systems and other techniques. A good survey of advances on network-adaptive mechanisms can be found in [27].

1.2 Approach

In order to answer our original question, with regards to the quality of multimedia communications over the Internet today, we take two steps. First, we study measurements collected over Internet backbone networks. Then, we simulate the transmission of voice and video using these traces and we obtain results in terms of perceived quality. Let us now discuss the key elements of our approach, namely the use of measurements and of perceived quality measures. Let us also define the loss and delay control mechanisms, implemented at the end-systems, that we consider as part of the VoIP/VIP system under study.

Network Measurements

In order to capture the behavior of today's Internet, we use measurements collected over wide-area backbone networks. The measurements were collected and provided to us by RouteScience Technologies Inc., [79]. Measurement facilities were installed in five cities across the US and they were connected to the backbone networks of seven major ISPs across the US. This a large enough data set, covering a wide range of networks (43 paths in total belonging to 7 ISPs) and time (2.5 days), thus capturing sufficiently well the characteristics of today's Internet backbones in the US.

These measurements are limited to wide-area backbone networks. This choice was primarily driven by our access to extensive measurement data collected for such networks by RouteScience Technologies, Inc. There are many reasons why a focus on wide area backbone networks is of interest. These networks are an important part of the end-to-end path for all long distance VoIP calls, including calls that are serviced by a combination of a switched telephone network in the local area and the Internet for the long haul. Performance problems in these networks will be experienced by all such calls; therefore, they need to be well understood and fixed, regardless of what takes place elsewhere in the path.

In practice the end-to-end path may go through many networks. The Internet is a large system with a hierarchical structure. The levels of this hierarchy are the following from the bottom to the top: residential or wireless access networks, campus and metropolitan area networks and wide-area backbones (that provide global connectivity). Characteristics vary considerably between levels. For example, wide area backbone networks are generally well provisioned, while regional networks that handle a much higher degree of variability in traffic may exhibit congestion. Therefore, to study the characteristics of the Internet and derive any conclusions as to its performance with regards to VoIP, it is more appropriate to focus on one level of the hierarchy at a time. Even when focusing on a single level in the hierarchy, we find that the Internet comprises many separate domains, each administered by a different organization. Each such organization is responsible for the deployment and operations of the networks within that domain. These networks can differ considerably in their provisioning and operations, and as a result, their performance. Therefore, to get a realistic assessment of their loss and delay characteristics, it is important to study a good sample of these networks. This is also important, given that, in general, Internet users (including VoIP service providers) do not have control over the routes taken by packets. Packets transmitted between two hosts may not take the same route in both directions, and these routes may fall in different domains.

The study of these measurements are useful in the following ways. First, they reveal the loss and delay characteristics of Internet backbones today in the US. These characteristics may change in the future, hopefully toward the direction of the good paths. Second, we used these traces to simulate the transmission of voice and video and assess their quality. We specifically chose the size and frequency of probes to simulate stream traffic. The delay and loss experienced by the probes was assigned to the voice and video packets in our simulation. This implies that we consider voice and video traffic sent using the same best-effort service as the probe packets (i.e. sharing the same queues with data traffic), which is mostly the case today. Third, the measurements characterization can be used to simulate the behavior of backbone networks, for example, for the purpose of studying the performance of end-to-end mechanisms, such as loss concealment and playout scheduling algorithms.

Quality Measures

Loss and delay measurements give a rough idea of the performance of multimedia traffic over those paths. However, they do not directly translate to user perceived quality. Ideally, a quality assessment would consist of people actually having phonecalls or video conferencing sessions and giving their opinion. However, this is not feasible for assessing a large data set and it is clearly inappropriate for online monitoring.

Typically, the network community has been assessing the quality of multimedia traffic in terms of loss and delay percentiles. The speech and video research communities have been studying the impairments introduced by encoders and by transmission loss using subjective tests. A mapping of measurable network parameters to perceived multimedia quality would be very useful for network planning. In the case of voice, the Emodel is an ITU-T standard that provides such a mapping, [35, 36]. Objective measures, which compare the original to the distorted signal and provide a rating that correlates well with voice subjective quality, have been proposed and standardized. Examples include the following measures, listed in chronological order: Perceptual Analysis Measurement System (PAMS) in [45], Modulated Normalizing Blocks (MSN) and Perceptual Speech Quality Measure (PSQM) in ITU-T P.861 [46], Perceptual Evaluation of Speech Quality (PESQ) in ITU-T P.862 [47].

In the case of video, similar objective measures have been developed (such as PDM [104] by EPFL) and work in this are is still ongoing. For example, Institute for Telecommunication Sciences (ITS, [33]) is currently working on the Video Quality Metric (VQM) that can be used to assess various aspects of perceptual distortion of a video sequence after transmission over a network. A recent study used VQM to relate IP network parameters to video quality, [1]. Also, an analytical model has been developed in [90], that captures the inter-frame error propagation and predicts the distortion given the loss rate and the video sequence, and assuming uncorrelated and stationary loss. Mapping of measurable network parameters to perceived quality is currently under study for computer games.

In our assessment of VoIP performance, we study the different aspects of voice quality and we use an assessment method that predicts the overall rating that a human would give at the end of a phonecall. In the absence of a mature Emodel-like standard for video, we use the widely used peak signal-to-noise ratio (PSNR) as our video quality measure.

Control Mechanisms Considered

The application has no control over the behavior of the network. However, there are numerous mechanisms that can be used at the end-systems to prevent or mitigate the effect of network impairments. The choice of such mechanisms strongly affects the overall performance. Our assessment of the Internet backbones performance takes into account some of them, commonly used in the context of interactive applications.

First, we consider that loss concealment is implemented at the receiver to mitigate the effect of packet loss for both voice and video. For the voice study, concealment is taken into account when we translate packet loss to speech degradation, by using the appropriate loss-to-degradation curves that consider loss concealment (see Section 3.1.2.3). For the video simulations, the non-normative concealment of the H.264 decoder [52] is used. The second mechanism that we consider, is playout scheduling at the receiver to mitigate the effect of delay jitter in the network. In Section 3.2, we study the performance of existing algorithms and we propose new algorithms that perform well over these traces. Third, in Section 3.4, we study the benefit from using path diversity. The idea is that sending a stream over multiple paths increases the options and allows for optimized choices at the end-systems.

Assessing all available error control and recovery mechanisms over these paths would be a huge problem space, as one can see in the related surveys of such mechanisms for voice and video, [73, 101]. Furthermore, not all combinations are appropriate for all networks and application requirements. For example, the strict delay requirement combined with the large coast-to-coast delay rules out time-consuming mechanisms such as retransmissions or packetization schemes that introduce additional delay. Scalable video coding is not applicable in our case either, because we assume best effort service provided to multimedia traffic under study, the same service that is provided to the measurement probes. Scalable video coding is not useful unless combined with some network differentiation mechanism.

This study did not intend to advance the state of the art in multimedia error control and recovery. It rather considers a limited set of mechanisms, commonly used for the interactive applications, in order to provide a fair assessment of the performance of backbone networks.

1.3 Thesis Contributions and Outline

This thesis is an Assessment of the Quality of Multimedia Communications over today's Internet. The thesis contributions and outline are the following.

Thesis Contributions

First, we characterize delay and loss measurements collected over the backbone networks of Internet Service Providers in the US. On one hand, the characterization of this measurement set is revealing for the state of Internet backbones today. We identify delay and loss patterns, consistent per path or even per provider, which seem more related to network control protocols and routers' operation, rather than to traffic load. On the other hand, the characterization and modeling can be used to capture the behavior of these networks, for example for the purpose of evaluating the performance of application level mechanisms over them.

Our second contribution is the assessment of VoIP performance over these backbone networks. We find that performance varies significantly, depending on the provider. A large number of the networks studied are ready to support VoIP at high quality levels, which is a confirmation that packet voice is a sound approach. The rest of the networks experience problems that seem mostly related to network reliability. Therefore, in order to improve backbone networks, more effort should be put on understanding the network operation rather than on devising Quality-of-Service mechanisms. As parts of the assessment we take two important steps. The first step is the development of a methodology for assessing subjective quality based on the loss and delay measurements. The second step is the study of application-level mechanisms that can help mitigate the impairments introduced by the network. An important such mechanism is playout scheduling that absorbs the delay variability. We evaluate existing algorithms and develop new ones that perform well for the networks under study. We also confirm the value of path diversity over these networks.

Finally, we consider video-conferencing traffic with tight delay constraints and assess its performance over Internet backbone networks, following similar steps as for VoIP. We find video and voice traffic to share fate over these networks. In the process, we deal with a number of issues, including path diversity and video quality assessment.

Thesis Outline

In Chapter 2, we present the measurements that constitute the basis of this study. In Section 2.1, we review related work on network measurements. In Section 2.2, we describe the measurement setup. In Section 2.3 and 2.4, we describe the loss and delay characteristics observed in the traces. For each type of event or characteristic, we provide representative examples and descriptive statistics. Section 2.5 provides evidence for the consistency of loss and delay patterns on the same path and, to a large extent, on the same provider. Section 2.6 concludes the chapter, with a summary of the findings and a discussion on the possible causes and remedies of the problems.

In Chapter 3, we study the performance of VoIP over these backbone networks. In Section 3.1, we discuss the various aspects of voice quality, collect results from studies that have assessed the effect of individual impairments, combine them and develop a methodology for predicting the subjective rating that a human would give at the end of a phone call. In Section 3.2, we study playout scheduling for VoIP. First, we evaluate existing algorithms and identify their weaknesses. Then, based on the patterns observed in the traces and our understanding of the voice quality aspects, we propose new algorithms and demonstrate that they perform well over these backbone networks. In Section 3.3, we apply the assessment methodology on representative traces and we obtain statistics in terms of perceived quality. In Section 3.4, we discuss the benefit of exploiting the path diversity. In particular, we confirm the benefit of combining multipath streaming with playout scheduling.

In Chapter 4, we study the quality of video traffic with tight delay constraints over Internet backbones. In Section 4.1, we describe the Video over IP (VIP) system under consideration. In Section 4.2, we simulate the transmission of video traffic over representative paths and we present statistical results.We also confirm the benefit of rate-distortion optimized video streaming over multiple paths. In Section 4.3, we discuss to what extent these observations hold for different VIP scenarios. In Section 4.4, we compare the simulations results to the quality degradation predicted by the analytical model in [90].

In Chapter 5, we conclude, summarize our findings and discuss future directions.

1.4 Glossary

The following table summarizes terms and abbreviations, frequently used throughout the text. The terms are listed alphabetically, although they may be used in different chapters.

Table 1.1: Symbols and abbreviations

Symbol / Abbreviation	Explanation
AND	Andover, Massachussets
ASH	Ashburn, Virginia
'assisted max fixed'	playout scheduling algorithm defined in section 3.2.5
'assisted 99% fixed'	playout scheduling algorithm defined in section 3.2.5
'baseline'	the 'spike-det' scheduling algorithm, using the default
	parameters: $\alpha = 0.998002$ and $ENTER = 100ms$
CCDF	Complementary Cumulative Distribution Function (% of samples above a certain value)
CDF	Cumulative Distribution Function (% of samples up to a certain value)
clip or elementary loss event	consecutive packets lost
cluster of loss or complex loss event	a short period (tens of seconds) with high loss rate i.e. composed of many elementary loss events
EWR	Newark, New Jersey
Fast Increase - Exp Decrease	playout scheduling algorithm defined in section 3.2.6
fps	frame rate (fps)
ISP	Internet Service Provider
loss duration	number of consecutive packets (or ms) lost
loss free duration	period between two packets lost (also “loss distance“)
'maximize-MOS'	playout scheduling algorithm defined in section 3.2.7
mouth-to-ear or 'm2e'	end-to-end one way delay
outage	a long duration (in the order of tens of seconds) during which all packets are lost. A long elementary loss.
PDF	Probability Density Function (PDF) (% of samples with a value within a certain range)
P_1, P_2, ... P_7	Provider 1, Provider 2, ..., Provider 7
PSNR	Peak Signal-to-Noise ratio
QoS	Quality of Service
SJC	San Jose, California
'spike-det'	the 'spike-detection' playout scheduling algorithm proposed in [77] and evaluated in section 3.2.3
THR	Thornton, Colorado
Thu	Thursday, 06/28/01
UTC	Coordinated Universal Time (Greenwich time)
VoIP	Voice over IP
VIP	Video over IP
Wed	Wednesday, 06/27/01

Chapter 2

Measurements of Internet Backbone Networks

2.1 Related Work and Outline

Related Work

There has been extensive work on measurements and characterization of delay and loss in the Internet. This is a research topic that is continuously evolving along with the evolution of the Internet as well as with the evolution of the applications. Some measurement studies characterize the aggregate traffic, e.g. to help the providers in their network provisioning. Some others characterize the loss and delay incurred by an individual flow, e.g. to help the application become network adaptive. Different studies may focus on different parts of the Internet hierarchy, such as backbone, regional or access networks. Active or passive measurements can be used. Active measurements can be encapsulated in various protocols, such as UDP, TCP or ICMP; probes can be generated at various packet sizes and frequencies. In some cases, privileged access to the routers is available; other times conclusions have to be made based exclusively on end-to-end measurements. Finally, different measurement studies try to characterize different attributes of the network, related to the performance of a specific application or network protocol.

As far as the network is concerned, our measurements are limited to backbone networks of Internet Service Providers in the US. Backbone networks are in general sufficiently provisioned, so they are typically believed not to introduce any impairments. It is interesting to note that we identify delay and loss patterns on those networks that seem mostly related to the network and router operation, as opposed to the traffic load. Similar patterns on backbone networks have been observed, in 1993 by S. Sanghi, A. Agrawala et.al. in [80, 81]. A more recent study in 1999 by C. Labovitz, [60], investigated the stability and the failures of wide-area backbones due to the underlying switching system as well as due to the software and hardware components specific to the Internet's packet-switched forwarding and routing architecture. Recent work on Sprint's backbone network, [7], focused on link failures and their impact on voice traffic. The same group studied the delay components in a router used in a backbone network, [72], and identified periods during which the routers were taking vacations from serving packets. [24] and many others have also monitored backbone links, for the purpose of characterizing the aggregate traffic.

End-to-end techniques for measuring ISP topologies at the router-level has been recently developed by N. Spring et.al. in [87] and have been used to provide accurate maps of the topologies of 10 major ISPs. However, note that the studies on topology discovery aim at discovering the topology of the network (via traceroute-based probing or via monitoring or routing protocols messages) rather than at calculating the loss and delay introduced at each link or end-to-end. It is expected that end-to-end techniques to study a network without privileged access, will become increasingly popular in the next years. An example is the recent paper [30] by K. Gummadi et.al., which estimates end-to-end latency between arbitrary Internet hosts.

An older end-to-end study of interest is [70] by A. Mukherjee, back in 1994. He studied loss and delay on regional, backbone and cross country paths, by sending infrequent probes.

He found (i) dominant low-frequency delay components which he modeled successfully using a Gamma distribution and (ii) a wide range in the correlation between loss and delay. Although the characteristics of regional and backbone networks have changed since then, his model has been recently used for helping network-adaptive applications in their estimation of network delay, [27].

Also different applications are interested in different network characteristics. There have been TCP-oriented studies (a good survey can be found in [71]), stream-based measurements (that use continuous media sources to sample the network) and generic purpose studies (sampling the network using Poisson flows, such as [105] which studied the "constancy", in time, of delay and loss properties of Internet paths). Because our end-goal is to assess the quality of multimedia traffic over these networks, of particular interest to us are the stream-oriented measurements studies. An early such study [4] was conducted in 1993 by J. Bolot, who sent audio traffic and measured the delay and loss incurred. The delay variability was found to have the form of spikes and was modeled as the result of multiplexing into a single queue, the audio flow with an Internet interfering flow. From the inter-arrivals of the audio packets, the characteristics of this interfering flow were inferred. In [93], multicast measurements were used to study the correlation of packet loss in the MBONE, in time as well as in space, exploiting the multicast tree topology. In [68, 93], audio traffic was also studied over the MBONE, and loss rates, burstiness and correlation between loss and delay were characterized. In [68], delay variability was found to have the form of spikes and playout scheduling algorithms were proposed to deal with these spikes. A recent study, [65], conducted a very large scale experiment where they streamed MPEG-4 low rate video to clients located in more than 600 cities and provided statistics for the quality of the video-streaming sessions.

From all these options, our study has the following elements: (i) we focus on *backbone* networks of major ISPs in the US (ii) we send probes with packet sizes and inter-packet distances similar to that of *voice* traffic (iii) we measure and characterize the *end-to-end* loss, delay and delay jitter for each flow of probes. Its value lies in the combination of the above elements in great accuracy and length (e.g. diversity of backbones, long measurement period, fine granularity of the probes sent, accurately synchronized timestamps) and to its relevance to *today's* Internet. The contribution lies in the characterization of this rich data set, capturing the loss, delay and delay jitter properties, that can be further used to study the performance of application traffic over these networks.

Outline

In this chapter, we study the measurements that constitute the basis of our work. In Section 2.2, we describe the measurement setup and collection. In Section 2.3, we describe the loss characteristics observed in the traces. We group the observations by the type of loss event, and for each type, we provide examples and statistics. In Section 2.4, we describe the delay characteristics observed in the traces, following a similar approach. In Section 2.5, we identify the characteristics that are common between paths of the same provider. We also provide additional details for completeness. In Section 2.6, we summarize those characteristics that cause impairments to voice traffic and discuss their possible explanations. The reader is referred to Table 1.1 for a list of the frequently used terms throughout this chapter and their definition.

As a part of the traces characterization, empirical distributions are provided that can be used to generate a delay or loss trace with the same distribution.[1] In a few cases, we also provide fits to distributions, to assist the reproduction of loss and delay characteristics of interest. The loss and delay characteristics can then be used to simulate the network and to evaluate network-adaptive applications. Modeling loss and delay in the Internet is a wide research area by itself. This study tries to (i) characterize the empirically observed loss and delay patterns (ii) identify correlations between events (iii) reason on possible explanations.[2]

[1]There are well-known computer methods for generating random variables with a given cumulative distribution function (CDF), like those described in [62].

[2]Without privileged access to the measured networks, it is difficult to be sure about the causes of the

We are particularly interested in those characteristics that affect VoIP quality, such as loss durations and delay spikes.

As far as loss is concerned, we characterize the loss and the loss-free durations. It turns out that when single packets are lost, the loss-free durations follow an exponential distribution. For special loss events that consist of a small number of loss durations, we provide the exact loss durations and the spacing between them. As far as delay is concerned, we observe that the delay pattern consists of spikes of different shapes, heights and distances from each other. We provide delay percentiles across the day, as well as statistics for the height and distance of delay spikes for selected representative periods. It turns out that lower delays usually follow random patterns (roughly exponential spike heights at roughly exponential distances). On the other hand, regular delay patterns (i.e. periodically repeated spikes of the same height) are more probable to occur (or at least to distinguish from the rest) when delays are higher. Low-frequency delay components are limited to a few paths; we discuss the appropriateness of the gamma distribution for modeling these components.

2.2 Measurement Set

Our study is based on delay and loss measurements provided by RouteScience Technologies Inc.. Measurement facilities have been installed in five major US cities: San Jose in California (SJC), Ashburn in Virginia (ASH), Newark in New Jersey (EWR), Thornton in Colorado (THR) and Andover in Massachusetts (AND). These measurement facilities have been connected directly to the backbone networks of seven different providers, through T1 or T3 links. We refer to the seven different providers as $P_1, P_2, ..., P_7$ for anonymity purposes. Multiple providers may connect a given pair of cities, resulting to 43 paths in total. The measurement setup is shown in Fig. 2.1. E.g. the arrow drawn from SJC to AND with a label "P_3, P_6" means that probes were sent from SJC to AND using providers P_3 and P_6.

observed events. However, it is still useful to identify their nature and discuss directions for improvement.

All paths are two ways, except for those shown in parenthesis.

Probes of 50 Bytes long each were sent every 10 ms between the measurement facilities. Probes were sent from Tuesday 06/26/2001 19:22:00 until Friday 06/29/2001 00:50:00 UTC, i.e. a continuous period covering a little over two full days. "UTC" stands for Coordinated Universal Time which corresponds to Greenwich Mean Time (GMT). GPS was used to synchronize senders and receivers and the network delays were inferred by subtracting the sender's from the receiver's timestamp. The data rate of the probes (40kbps) is a very small fraction of the links used in the backbone network; therefore it could not affect the delay and loss characteristics of these networks. The size of each probe was chosen to be 50 Bytes in order to simulate a G.729 frame generated every 10 ms at 8 Kbps rate: 10B for the payload and 40B for the IP/UDP/RTP header. By taking into account the access bandwidth of the providers, we are able to compute the transmission time and infer delays for any voice packet size from the probe delays.[3] Furthermore, the 10ms sending interval is small enough to simulate the highest rate a VoIP encoder/packetizer might send packets at. By appropriately omitting some probes we can simulate lower packet rates or silence periods. For example, by omitting 100 consecutive probes, we simulate a silence period of 1 *sec*. Also, by omitting every other probe, we can simulate voice packets sent every 20ms.

We have also studied a similar data set, collected by RouteScience, for 14 days (from 04:53:08 on 12/1/2000 until 23:59:59 UTC on 12/14/2000) using the same providers and three of the measurement facilities, namely SJC, EWR and ASH. The advantage of the earlier over the current measurement set is that it covers a longer time period. Its main drawback is that probes were sent at 100ms intervals, which are larger than the typically shorter (10ms-30ms) intervals between voice packets. All the results we present are based on the current, fine granular data set. In the context of this thesis, the earlier set of

[3] A G.711 packet sent every 10ms at 64 Kbps, contains 80B (payload) + 40B (header) = 120B. This is longer than the probe by 70B. The transmission of 70B takes $0.012ms$ and $0.038ms$ over a T3 and a T1 access link, respectively. We did take into account these delays, which are anyway negligible compared to the network delays. In the backbone, the bandwidth is in the order Gbps and the difference in transmission times is even shorter and thus ignored.

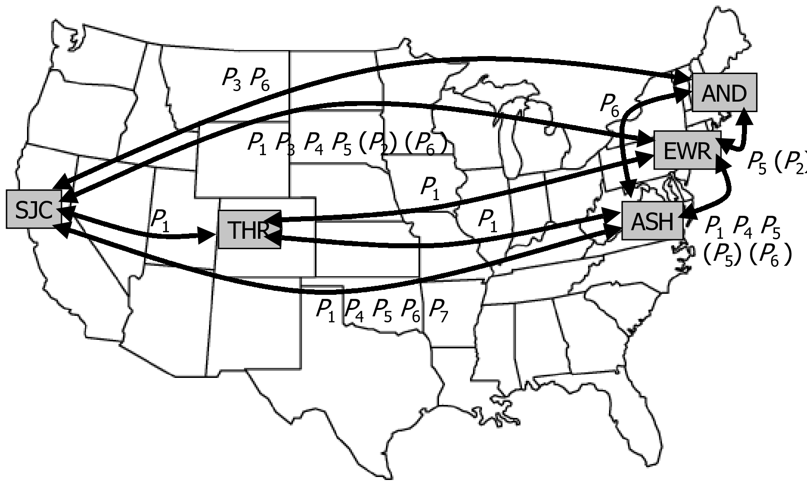


Figure 2.1: Measurements over ISP backbones

measurements was only useful to validate that our current findings, obtained over the 2.5 days period, are true over a longer time period.

2.3 Loss Characteristics

2.3.1 Summary

There was one path, namely SJC-AND for provider P3, with no loss at all during the entire measurement period. For all other paths, packet loss events of various characteristics occur. For four paths of provider P_3, loss occurred regularly for the entire measurement period, and is described in a separate section (2.3.5).

Perhaps, the most important observation is that for the remaining 38 out of the 43 paths, loss is sporadic. This means that, in general, there is no loss in the traces, except for relatively short time periods, during which, loss happens in a single or multiple loss durations. Therefore, it does not make sense to compute loss rates over large time periods. Indeed, no more than 0.26% of all packets are lost in any path, over the entire measurement period. However, the loss rate can be from 10 to 100% over short time periods. For the purpose of accurate description of packet loss characteristics, we identify two types of events:

elementary loss events which consist of consecutive probes getting lost (comprising one or more packets) separated by relatively long periods of time, and *complex loss events* which correspond to the occurrence of several elementary loss events concentrated over a short period of time.

Table 2.1 summarizes the loss on each path during the 48 hours period (Wed 06/27/01 and Thu 06/28/01). The first three columns define the path. Column 4 gives the total number of packets lost during the 48 hours period. Columns 5 and 6 give the number of loss events (isolated or part of a complex event) and the longest loss duration (in number of packets), respectively. Columns 7 and 8 give the number of single packets lost either in elementary or in complex loss events, and the percentage of the total loss (column 4) that is due to such single packets lost. Columns 9 and 10 give the number of isolated elementary events (i.e. that are not part of a complex event) and the percentage of total loss (column 4) that is due to such isolated events. Finally, columns 11 and 12 give the number of complex events and the percentage of the total loss due to complex loss events.[4]

Loss characteristics vary among different providers and sometimes also between paths of the same provider. Loss characteristics vary among different providers and sometimes also between paths of the same provider. Some providers (P_1, P_4, P_7) experience mostly elementary loss events. Some others (P_2, P_6) experience mostly complex events. The smallest number of packets lost occurs on some paths of providers P_1, P_5 and P_6 as indicated by the small number of total packets lost (column 4). Paths of provider P_5 have not only small number of packets lost, but also small number (see column 5) and length (column 6) of loss durations. The highest amount of loss (i.e. total number of packets lost in column 4) happens on path EWR-P_1-ASH and it is due to long loss durations. Also, a large number of single packets are lost on the four paths of provider P3, as described in section 2.3.5.

[4]Note that the paths of provider P_3 are excluded from this table and are described separately in section 2.3.5. The reason is that loss on provider P_3 is recurrent (at random intervals, on average every 5 seconds), instead of sporadic and the distinction between elementary and complex loss event becomes irrelevant. For the purpose of classification in Table 2.1, we considered two elementary events as separate, if they are are more than $\tau = 2sec$ apart, otherwise we consider them as part of a complex event.

Table 2.1: Loss statistics in 48 hours period, for all paths except provider P_3

Path			all loss events			single loss		elementary		complex	
1	2	3	4	5	6	7	8	9	10	11	12
From	Provider	To	packets	durations	max	number	% loss	number	%loss	number	%loss
THR	P_1	ASH	986	100	224	73	7.4 %	64	78%	9	22%
ASH	P_1	THR	2881	405	1110	348	12.1%	326	90%	29	10%
EWR	P_1	ASH	41059	269	2880	180	0.39%	187	49%	27	51%
ASH	P_1	EWR	19939	234	2329	111	0.55%	57	42%	29	58%
SJC	P_1	ASH	4559	81	1395	44	1.03%	32	4.36%	12	95.6%
ASH	P_1	SJC	2041	278	582	209	10.24%	88	81%	23	19%
SJC	P_1	EWR	17874	90	1600	4	0.02%	37	39%	20	61%
EWR	P_1	SJC	39247	85	3696	16	0.04%	49	66%	15	34%
SJC	P_1	THR	433	212	90	195	45.03%	195	89%	3	11%
THR	P_1	SJC	701	57	197	30	4.39%	17	81%	8	19%
EWR	P_2	SJC	4957	434	570	364	7.3%	35	37%	8	63%
EWR	P_2	ASH	2922	132	145	65	1.91%	42	11%	20	89%
SJC	P_4	ASH	3807	200	221	56	1.47%	45	11%	39	89%
ASH	P_4	SJC	20277	126	1582	34	0.17%	29	57%	35	43%
SJC	P_4	EWR	28890	56	2779	9	0.03%	24	90%	6	10%
EWR	P_4	SJC	15905	119	3305	19	0.05%	38	78%	32	22%
ASH	P_4	EWR	30103	57	11899	11	0.03%	29	90%	6	10%
EWR	P_4	ASH	42463	408	3326	245	0.57%	120	91%	79	9%
SJC	P_5	ASH	494	4	250	0	0	1	23%	1	77%
ASH	P_5	SJC	3611	1188	99	767	21.24%	378	38%	249	62%
SJC	P_5	EWR	256	5	139	1	0.39%	5	100%	0	0
EWR	P_5	SJC	3618	1143	100	736	20.3%	405	39%	231	61%
ASH	P_5	EWR	395	18	149	5	1.26%	11	48%	3	52%
EWR	P_5	ASH	699	57	270	28	4.18%	47	33%	5	67%
SJC	P_6	ASH	55	20	12	13	23.64%	7	73%	3	27%
ASH	P_6	SJC	277	142	23	125	45%	9	48%	3	52%
SJC	P_6	AND	496	323	45	240	48%	4	14%	6	86%
AND	P_6	SJC	1390	546	46	242	17%	219	69%	8	31%
ASH	P_6	AND	1685	38	533	17	1%	13	66%	7	34%
AND	P_6	ASH	2206	255	456	59	2.07%	183	35%	9	65%
EWR	P_6	SJC	2904	58	150	15	0.51%	32	15%	6	85%
EWR	P_6	ASH	4198	120	454	56	1.33%	36	18%	11	82%
EWR	P_6	AND	2598	40	145	10	0.38%	16	4%	5	96%
SJC	P_7	ASH	29291	6	16618	3	0.01%	4	98.99%	2	0.01%
ASH	P_7	SJC	11413	21	11190	9	0.08%	19	99.98%	2	0.02%

In the following sections, we give concrete examples of each type of loss event.

2.3.2 Elementary Loss Events

Elementary loss events come in various durations. Single packet loss events are a large percentage of all loss events but they contribute little to the total amount of loss, as it can bee seen in columns 5 and 7 of Table 2.1. There are other elementary loss durations for which we provide distributions for their durations and identify correlations with other events. However, let us first discuss two interesting loss durations: (i) 19-25 consecutive packets lost and (ii) longer periods, also called *outages*, lasting for tens of seconds up to two minutes, during which all packets are lost.

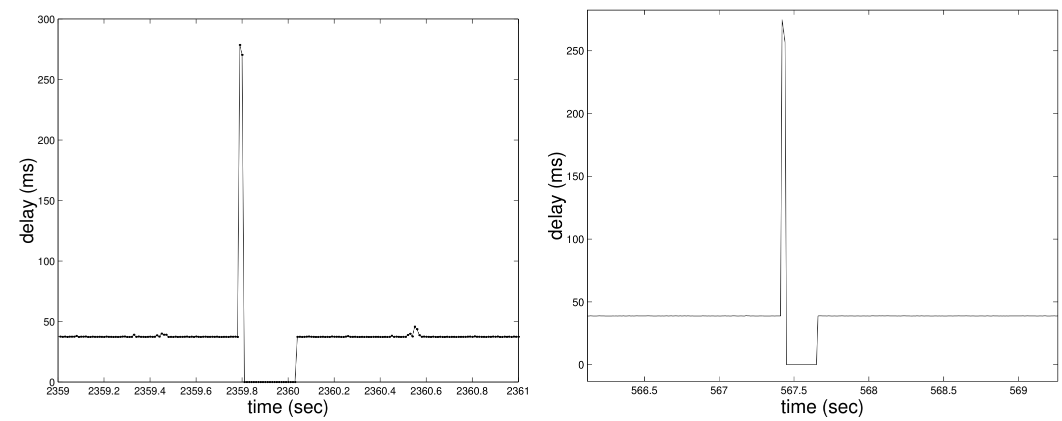
19-25 consecutive packets lost

It is interesting to note the frequent occurrence of 19-25 consecutive packets lost that typically follow high delay values. An example is shown in Figure 2.2(a), in which we plot the delay incurred by probes as a function of the probe's send time; for probes that are lost, we show a delay of zero. Another example of the same type of event on a different path is shown in Figure 2.2(b). We have no good explanation for this loss event. However, it appears frequently in the loss distributions of providers P_2, P_3 and P_5 (notice the knee around 200ms loss duration in the later Figures 2.20, 2.48, 2.49).

Outages

Let us now present examples of outages and show their correlation with other events, such as outages on other paths of the same provider and changes in the minimum delay.

The longest elementary loss event (166.18 sec) happened on the path SJC-ASH of provider P7 and it is shown in Figure 2.3(a). It is interesting to note that this long loss period accompanied a change in the fixed part of the delay. Also, the reverse path (from ASH to SJC) of the same provider incurred similar loss (111.9 sec) at the exact same time.



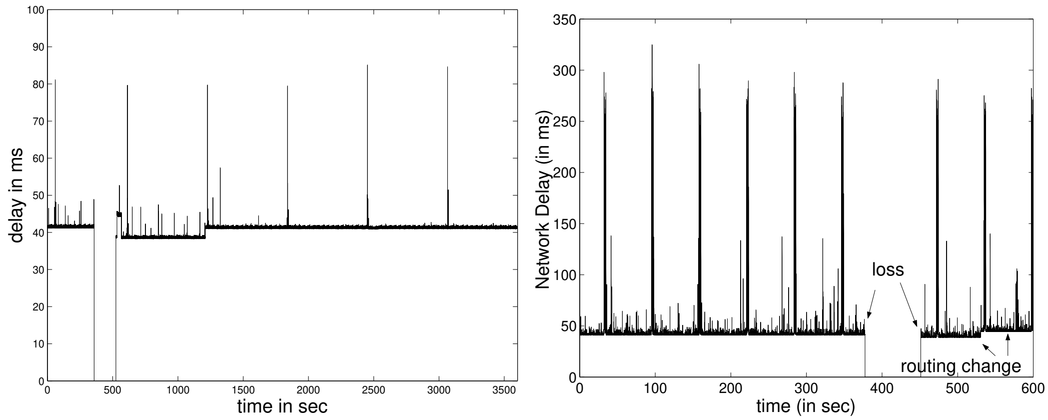
(a) 23 packets on EWR-P2-SJC, at Thu 13:00

(b) ASH-P6-SJC, zooming in 3sec on Wed 7:50

Figure 2.2: Examples of elementary loss of 19-25 consecutive packets

The event was repeated the following day at 3:20 with loss 12.639 sec on the path SJC-ASH. Such long loss periods occurred on 6 out of 7 providers at least 1-2 times per day. For two of these providers, these outages were correlated with a change in the fixed part of the delay. The change in delay was in the order of 1-2 milliseconds, which by itself is not significant, but it indicates a reconfiguration (e.g. a routing change) that may be responsible for the long loss duration. For provider P_4, outages accompanying changes in the fixed part of the delay, was a recurrent event. An example is shown in Figure 2.3(b).

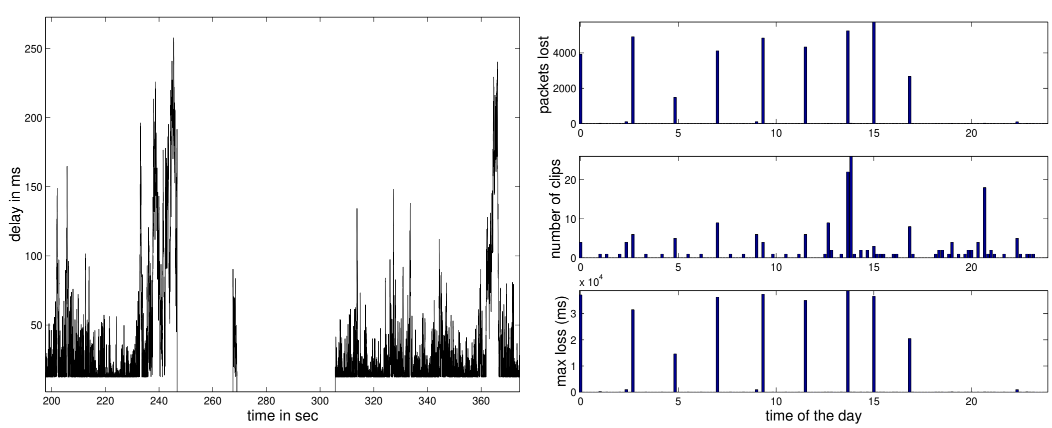
Examples of two outages without such a correlation are shown in Figure 2.4(a) for provider P_1. However, these outages can be particularly high (up to 50 seconds) and follow a sustained increase in delay. Figure 2.4(b) shows the number of packets lost, the number of loss durations and the maximum loss duration in 10 minute intervals for 24 hours on this path (EWR-P_1-ASH). We can see that nine outages from 10 to 50 sec happen during the day. Finally, an example outage 20 seconds long, not correlated with any other event can be found in Figure 2.16(a) for provider P_3 .



(a) A 166.18sec outage, on SJC-P7-ASH, on Wed 4:00-5:00

(b) A 77.3sec outage on EWR-P4-SJC, on Wed 21:10-21:20

Figure 2.3: Example of outages correlated with changes in the fixed part of the delay



(a) Zooming in on 3 minutes, Wed 15:00

(b) Loss behavior on Wed 06/27/01

Figure 2.4: Outages on EWR-P_1-ASH

Paths with elementary loss events

There are certain paths that experience consistently elementary loss events for the entire measurement period. In other words, loss on these paths happens in the form of isolated loss durations, as opposed to periods during which a large number of shorter loss durations is lost. When these elementary loss events happen less than 2 sec apart, then they may appear as complex loss events for the purpose of classification. However, a closer look into the traces reveals that loss happens in the form of long elementary loss events.

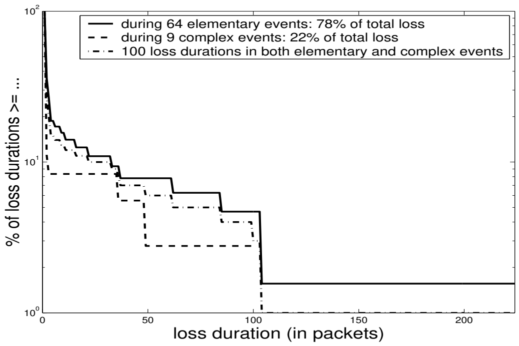
For example, all paths of provider P_1 experience isolated elementary loss events that contribute most to the total loss. Figure 2.5 shows the complementary cumulative distribution function (CCDF) of the loss durations for six paths of provider P_1. The first two paths experience medium loss duration; the second two paths experience large outages; the last two paths experience small loss durations. In all paths, loss is sporadic. Even what appears as complex loss (due to the $\tau = 2sec$ interval), consists in fact of two elementary losses a few seconds apart, as in Figure 2.4(a).

Loss is also sporadic on paths of provider P_4. Most loss on path EWR-P_4-SJC is due to isolated loss events that are quite long (10 to 100 seconds) and amount to 78% of the total loss, as shown in Figure 2.6(a). These are the outages during changes in the fixed part of the delay. Such outages are also mainly responsible for the loss on all other paths of P_4, except for SJC-ASH. Table 2.1 indicates that 57-91% of the total loss on these paths is due to elementary loss events. Figure 2.6(b) shows that many of these elementary events are quite long, i.e. above 500 consecutive packets or, equivalently, 5 seconds.

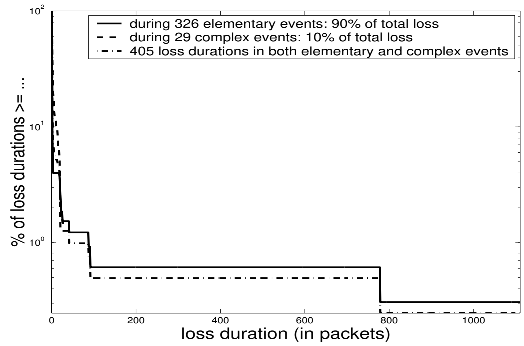
Finally, loss on the two paths of provider P_7 happens only during two long outages described above, in the order of 2 minutes.

2.3.3 Complex Loss Events

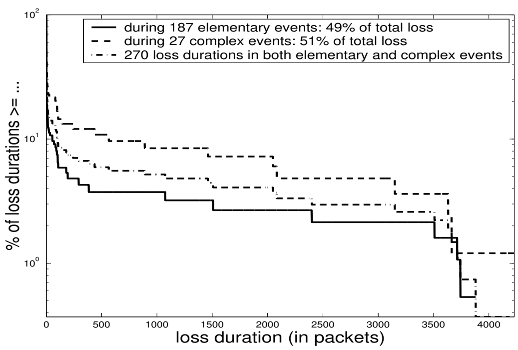
Complex loss events consist of multiple elementary loss events (single packets or longer durations) over a relatively short period of time (up to 50 seconds), during which the loss



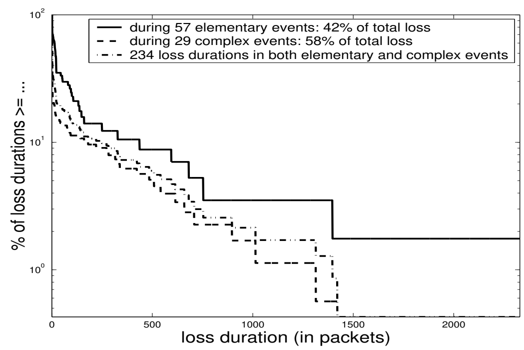
(a) THR-P1-ASH



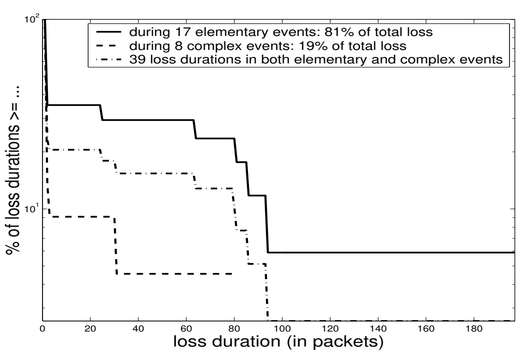
(b) ASH-P1-THR



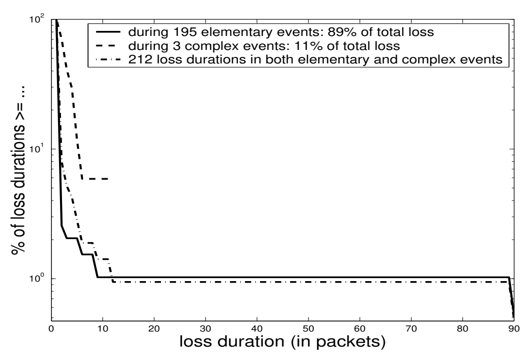
(c) EWR-P1-ASH



(d) ASH-P1-EWR

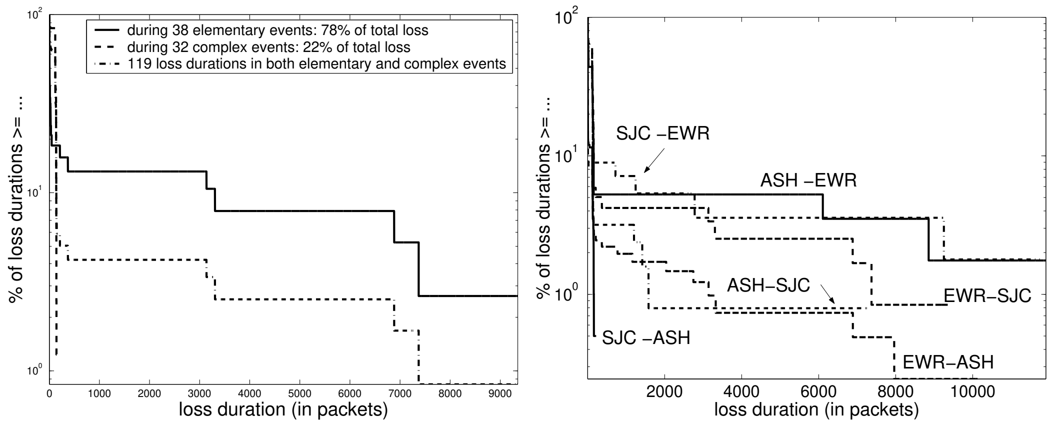


(e) THR-P1-SJC



(f) SJC-P1-THR

Figure 2.5: Loss on paths of six paths of provider P_1. Complementary Cumulative Distribution Function (CCDF) of loss durations in 48 hours.



(a) Path EWR-P4-SJC

(b) All 6 paths of P4

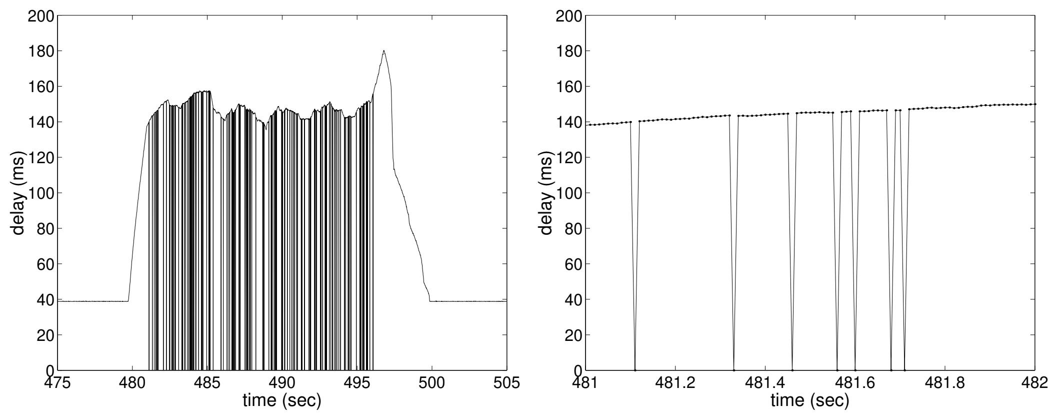
Figure 2.6: Loss durations on provider P4.

rate is 10-80%. They happen mainly on providers P_2, P_5 and P_6. Let us now see two characteristic types of complex loss events. The first type consists of single packets lost. The second type consists of fewer and longer loss durations (up to 1.5sec). In both cases, there are correlations with loss on other paths.

Example 1: complex loss events consisting of single loss

As a concrete example, we consider the path ASH-SJC of provider P6 and a single complex event that lasted 15 sec during which packet loss occurred in the form of single packets leading to a loss rate of 9.4%, see Figure 2.7(a). In Figure 2.7(b), we show a blow up of a portion of the graph shown in Figure 2.7(a). The loss-free intervals between consecutive losses have durations which are exponentially distributed, as it is shown in the cumulative distribution (CCDF) in Figure 2.8(a). Indeed the CCDF of the exponential distribution would be a straight line (in a x-logy plot). The autocorrelation function for the loss-free durations also decreases fast in Figure 2.8(b), indicating independence.

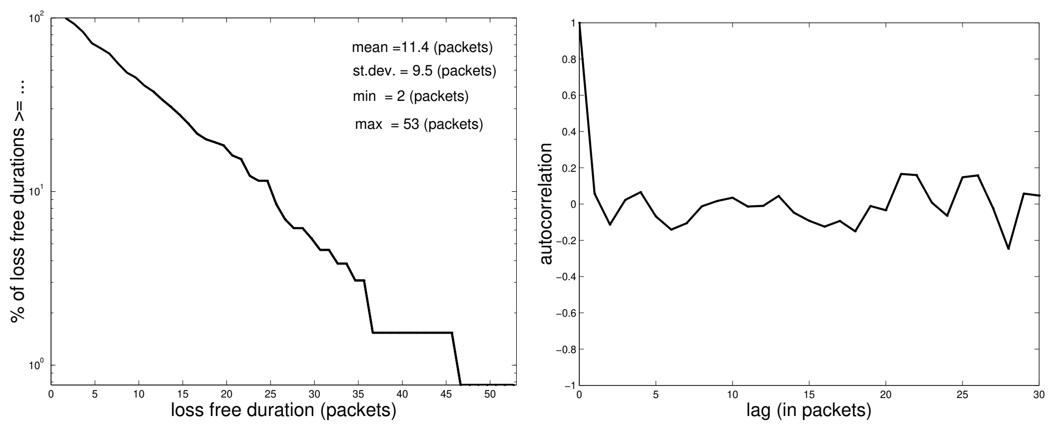
This roughly exponential CCDF shape was also observed in the traces with regular loss (see Section 2.3.5) as well as in other complex events consisting of individual loss. This



(a) 15 sec of single loss

(b) Zooming in

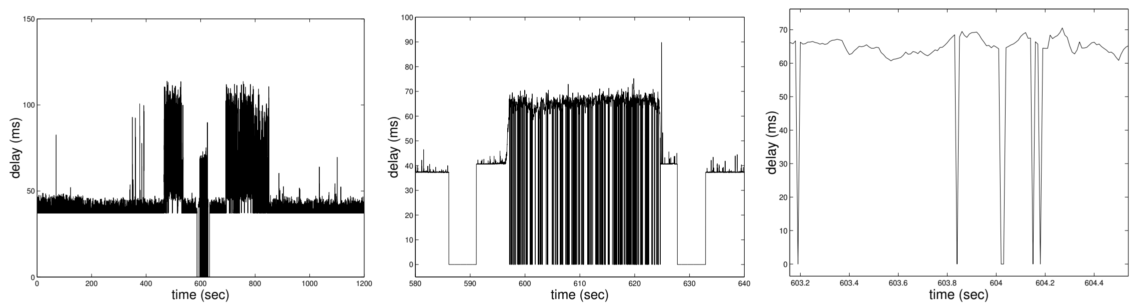
Figure 2.7: Example "1" of complex loss event. A complex loss event on path ASH-P_6-SJC, on Wed 06/27/01 at 3:20 (UTC). 141 packets were lost during 15 seconds: 131 single packets and 5 times two consecutive packets.



(a) Complementary Cumulative Distribution Function (CCDF) of loss-free durations

(b) Autocorrelation function of loss-free durations

Figure 2.8: Loss-free durations, i.e. times between two (elementary) losses, during the complex loss event on path ASH-P_6-P1, on Wed 06/27/01 at 3:30 (UTC)



(a) 20 minutes (b) Zooming in on 1 minute (c) Zooming in on 1.5 second

Figure 2.9: Path EWR-P_2-SJC. Complex loss event, on Wed 06/27/01 at 3:30-3:50 (UTC)

finding agrees with the finding of [105], that the loss process (arrivals of elementary loss events) can be modeled well as Poisson. The correlation in loss observed in earlier studies, [4, 93], was due to consecutive packets being lost. What we call here elementary loss event corresponds to an event in the loss process in [105].

Another similar example is the event on EWR-P_2-SJC, on Wed at 3:30 shown in Figure 2.9. It comprises a period of single packet loss events spanning a period of 30 seconds, between two longer elementary loss events lasting 502 and 512 packets each. The entire duration of the complex loss event is 50 seconds, and the packet loss rate during that period is 24.6%. During the middle period, there are 205 loss durations, 194 of which are single packets lost and 11 consist of 2 packets. The time between two consecutive losses, i.e. the "distance" or loss-free duration, has a CCDF that is shown in Figure 2.10.

Example 2: complex loss events consisting of longer loss durations. Some of the complex loss events often coincide with changes in the delay pattern. An example of a loss preceding a change in the delay pattern, is shown in Figure 2.11(a). In Figure 2.11(b), we blow up a portion of the graph. The event consists of longer loss durations (ranging from 10 to 143 packets, with an mean and standard deviation of 103 and 41 packets respectively). The loss-free durations range from 12 to 1891 packets, with a mean and standard deviation

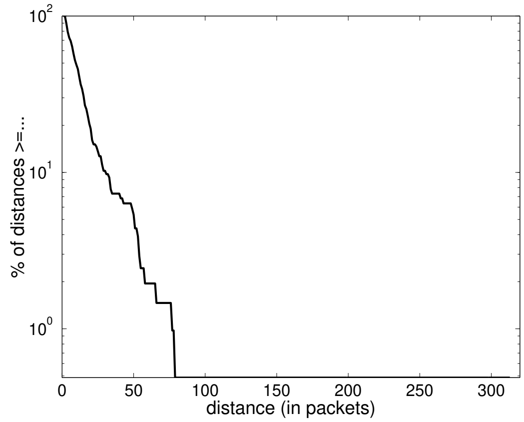
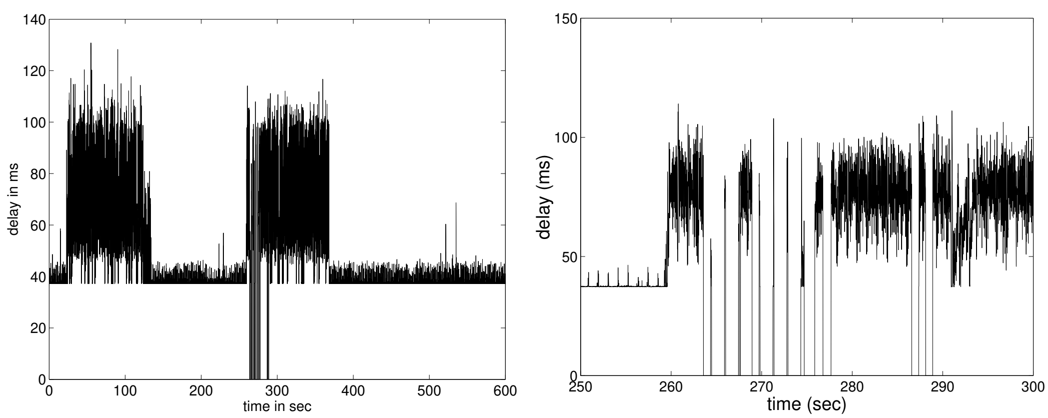


Figure 2.10: Distribution (CCDF) of the loss-free durations (distances between losses), during the single packet loss period, on P_2, on Wed at 3:30-3:50.



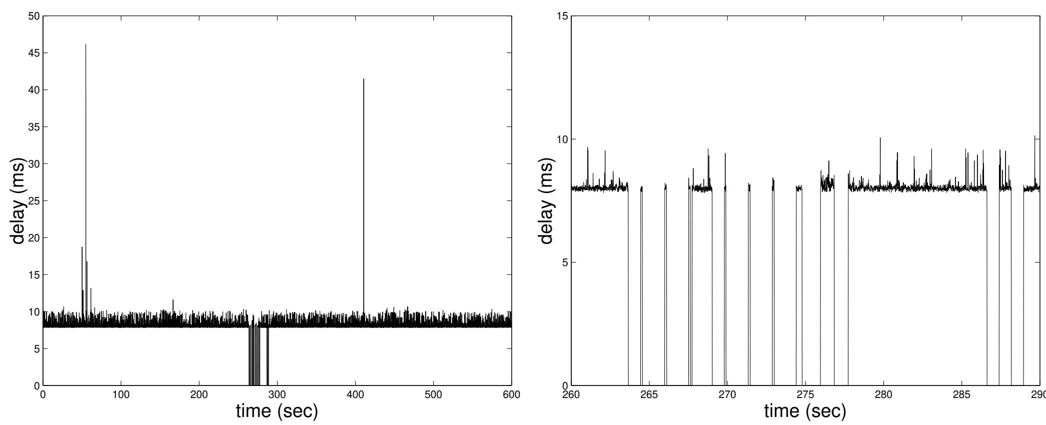
(a) A complex loss event, correlated with a delay change

(b) Zooming in on the complex loss event

Figure 2.11: Example "2" of complex loss event. Path EWR-P_2-SJC. Thu 06/28/01 at 20:10.

Table 2.2: Description of the loss event on Thu at 20:10, on both paths of P_2

counting loss durations	1	2	3	4	5	6	7	8	9	10	11	12
	path EWR-P_2-SJC, event starting at packet sequence=26356											
loss duration (in packets)	79	143	142	10	78	142	142	142	117	90	78	79
distance from next (packets)	12	13	13	129	12	13	13	38	91	891	77	-
	path EWR-P_2-SJC, event starting at sequence=26365											
loss duration (in packets)	79	143	141	11	78	143	142	141	118	90	78	79
distance from next (packets)	12	13	12	130	11	13	13	38	89	889	78	-



(a) 10 minutes (b) 30 seconds

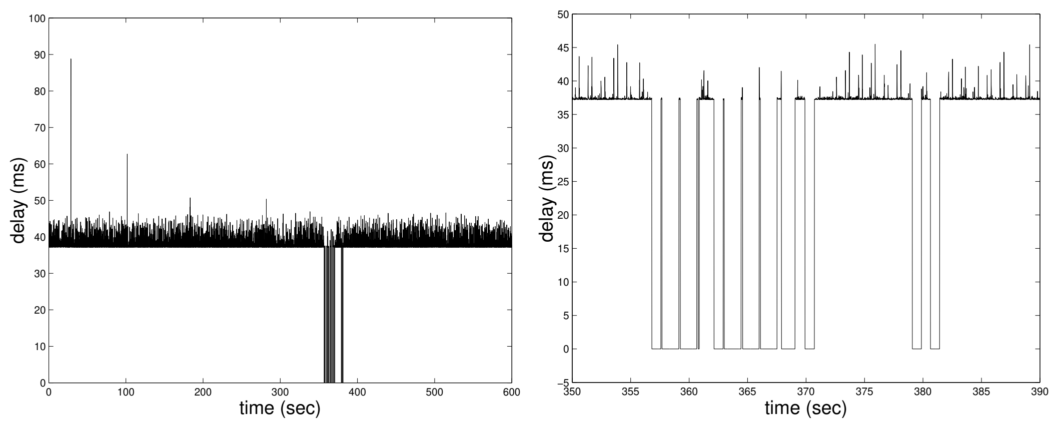
Figure 2.12: Path EWR-P_2-ASH. Complex loss event on Thu at 20:10, synchronized with the event on EWR-P_2-SJC.

of 118 and 259 packets respectively. The whole event lasts a total duration of 30 seconds and leading to a packet loss rate of 41.4%. Because there are only 12 loss durations in this event, it does not make sense to provide a distribution for it; instead, we provide the exact values for the loss durations and distances, in Table 2.2. It is interesting to note that the exact same loss pattern (without the change in delay) happens on the second path of this provider at the exact same time, see Figure 2.12.

Another event that happened on both paths on Wed at 6:20, was identical to the previous one: 1234 packets were lost over 30 seconds, leading to loss rate of 41.1%. This event is described in Table 2.3: the 12 loss durations and distances are identical or very close to

Table 2.3: Description of the loss event on Wed at 6:20, on both paths of P_2

counting loss durations	1	2	3	4	5	6	7	8	9	10	11	12
	path EWR-P_2-SJC, event starting at packet sequence=35680											
loss duration (in packets)	78	145	141	9	79	145	141	142	116	81	78	79
distance from next (packets)	10	13	13	127	9	14	12	39	86	838	78	-
	path EWR-P_2-SJC, event starting at sequence=35662											
loss duration (in packets)	78	145	142	9	79	145	140	143	116	81	79	77
distance from next (packets)	10	13	13	127	9	14	12	39	86	837	79	-



(a) 10 minutes

(b) Zooming in on 30 seconds

Figure 2.13: Complex loss event on path EWR-P_2-SJC on Wed at 6:20.

those in Table 2.2. Notice the similarity with the previous events. Also, the exact same event happens at the same time on the second path of provider P2.

In addition, loss events similar to the above, happen on all three paths of provider P_6 that depart from EWR, at the same time (Thu 20:10, Wed 6:20). Table 2.4 shows the loss durations and distances on the three P_6 paths departing from EWR during the loss event on Wed at 6:20. Once again, notice the striking similarity in the loss durations and distances between the three paths as well as and with the previous Tables 2.2 and 2.3.

Table 2.4: Description of the loss event on Wed at 6:20, on the three paths of P_6, departing from EWR.

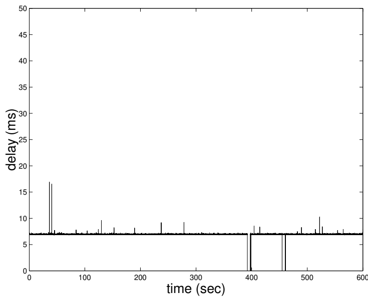
counting loss durations	1	2	3	4	5	6	7	8	9	10	11	12
	path EWR-P_6-SJC, event starting at packet sequence=35667											
loss duration (in packets)	78	144	141	9	79	144	142	141	116	81	78	79
distance from next (packets)	11	14	13	127	10	13	14	39	86	836	78	-
	path EWR-P_6-ASH, event starting at sequence=35679											
loss duration (in packets)	79	143	142	10	78	145	142	141	116	82	78	79
distance from next (packets)	12	13	13	127	10	12	13	39	86	836	78	-
	path EWR-P_6-AND, event starting at sequence=35685											
loss duration (in packets)	79	144	141	10	79	145	141	140	118	81	78	78
distance from next (packets)	10	14	13	127	9	13	14	37	86	837	78	-

2.3.4 Simultaneous Loss Events on Many Paths

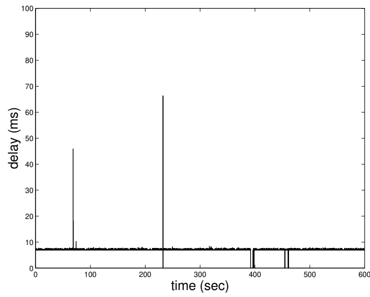
We saw examples of both elementary and complex loss events that happen simultaneously on more than one path of the same or of more than one providers. An example of a 2 minutes outage that happened simultaneously on both paths of provider P_7 was shown in Figure 2.3(a). All complex loss events discussed in the section above, happened during one of three periods (either on Thu 20:10, or on Wed 6:20 or on Wed 3:20), they started at the same time and they had similar loss pattern. And these are not the only occurrences.

On Wed 3:20-3:30, three paths of provider P_6 incur the same loss durations, shown in Figure 2.14. In the tables of Section 2.5 that summarize each provider, we can see many loss events that happen at the same time and day and have the same duration.

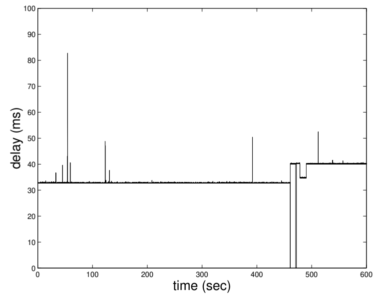
Synchronization of loss events also happens on provider P_4. Loss on all six paths of this provider happens in long outages correlated with changes in the fixed delay. Looking at the summary Table 2.11 for this provider, we see that these events often happen simultaneously on more than one paths. For example, on Tue from 21:00 until 21:20, on Wed at 21:00 and on Wed at 6:20, changes in delay and long outages happen at the same time on all six paths of the provider. Another such event happens on Thu from 7:30 until 7:50, on both directions of ASH-P_4-SJC. Actually, all outages on the path ASH-P_4-SJC happen simultaneously both



(a) ASH-AND



(b) AND-ASH



(c) EWR-SJC

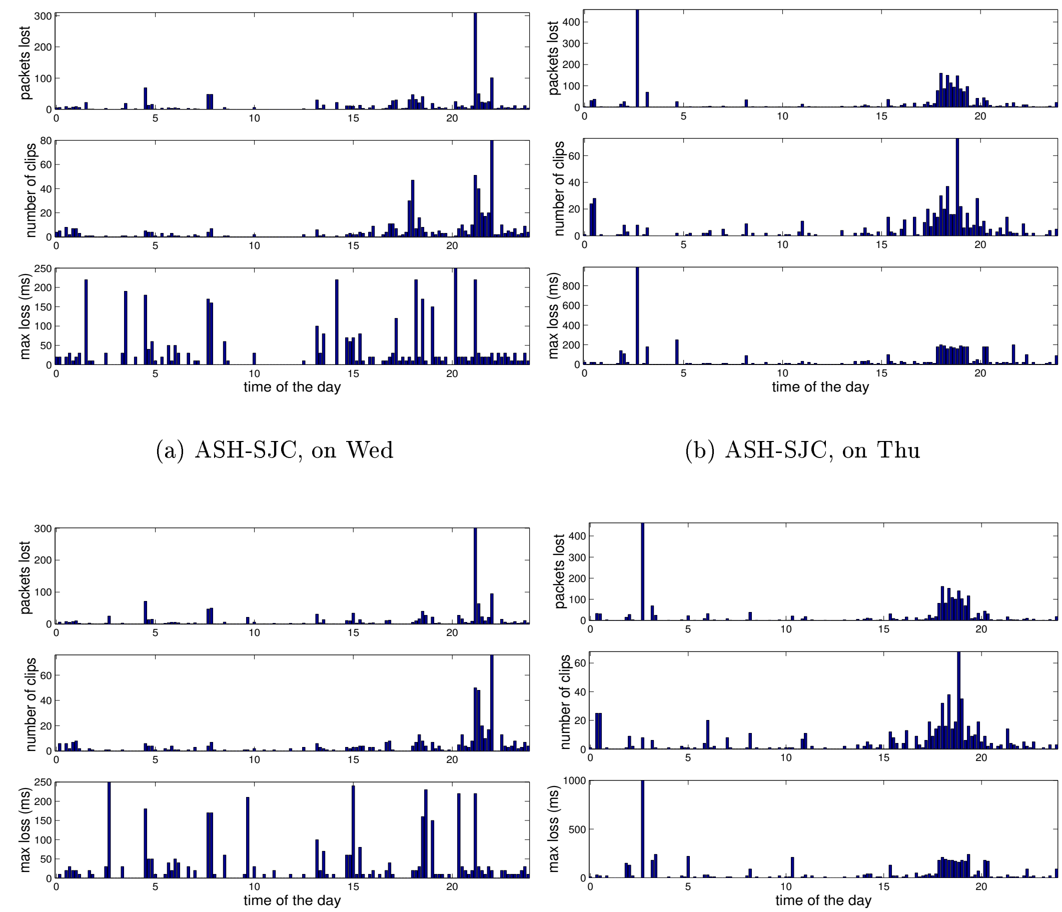
Figure 2.14: Synchronized loss events on three paths of provider P_6, on Wed, 3:20-3:30 (UTC)

ways: on Wed at 6:20, on Wed at 21:00, on Thu at 7:30, on Thu at 8:10, on Thu at 20:00.

An example of synchronization of loss events during the entire measurement period is shown in Figure 2.15. These paths belong to the same provider and they have the same destination: ASH-P_5-SJC and EWR-P_5-SJC. The figure plots the number of packets lost, the number of elementary events (or “clips”) and the maximum loss duration for each 10-minute interval, for two paths of the same provider (P_5). The two paths have almost identical loss behavior for the entire 48 hours period: almost the same number and duration of losses, at the same time of the day. We also zoomed in closer and confirmed that similar loss durations happen simultaneously.

For example, the loss on path ASH-SJC on Thu at 2:40, is shown in the later Figure 2.38(g). It consists of 8 clips of which the longest was 99 packets. Loss happens also at the exact same time on four other paths: SJC-P_5-ASH (with longest loss duration 250 packets), SJC-P_5-EWR (with longest duration 139 packets), ASH-P_5-EWR (with longest duration 149 packets) and EWR-P_5-ASH (with longest duration 270 packets).

The synchronization of loss events on many different paths indicates that these paths share a network element. Failures or a congestion periods on a shared link or the idiosyncratic behavior of a shared router can affect all paths. The repetition of loss events with almost



(a) ASH-SJC, on Wed

(b) ASH-SJC, on Thu

(c) EWR-SJC, on Wed

(d) EWR-SJC, on Thu

Figure 2.15: Identical loss behavior during the entire 48-hours period, on two paths of provider P_5: EWR-P_5-SJC and ASH-P_5-SJC.

identical characteristics at different times on the same path, could be due to the operation of a router on this path, some maintenance or network control procedure.

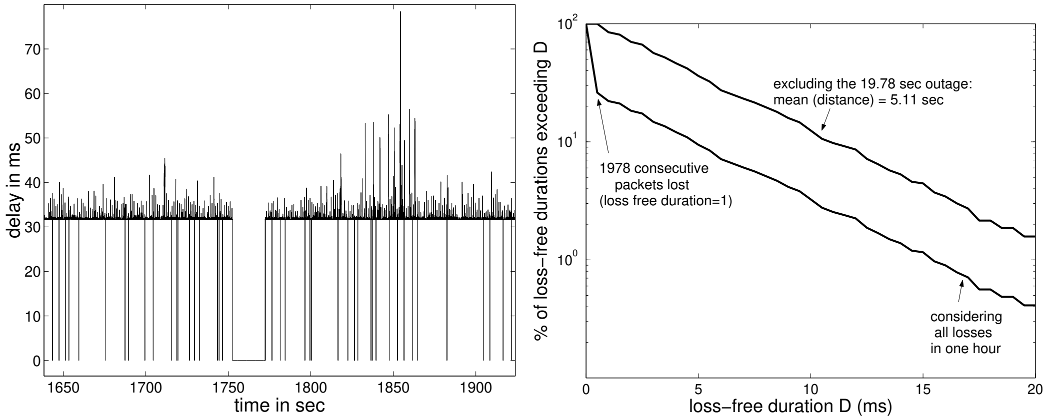
2.3.5 Regular Loss of Single Packets

We mentioned that loss is sporadic for all but four paths, that belong to provider P_3, and for which loss occurs regularly for the entire measurement period. Loss on these paths consists mostly of single packets lost, separated by loss-free intervals, that are roughly exponentially distributed with an average of 5 sec (looked at short and long time scale).

Let us look at an example path, EWR-P_3-SJC, in detail. Figure 2.16 shows the complementary cumulative distribution function (CCDF) of the loss-free durations, for an one-hour period on path EWR-P_3-SJC. Loss during that hour, consists of single packets lost, and an outage (1978 consecutive packets lost). The intervals between single losses seem to match very well the exponential shape with mean 5.11sec (indeed the exponential distribution has a CCDF which is a straight line in a x-logy plot) and the autocorrelation function decreased fast from the first samples (thus indicating independence). The overall loss rate is low (0.2%).

Another 10 minute period on the same path is shown in Figure 2.17(a); a blow-up of the first 100 seconds is shown in Figure 2.17(b). During this 10 minute period, 119 single packets were lost, resulting to 0.198% average loss rate. Lost packets were separated by loss-free intervals that follow a distribution with the shape of an exponential (with a mean of 5 seconds), shown in Figure 2.18(a). Note that there is a small number (118) of such intervals in the 10 minutes considered, which explains the coarse steps in Figure 2.18(a).

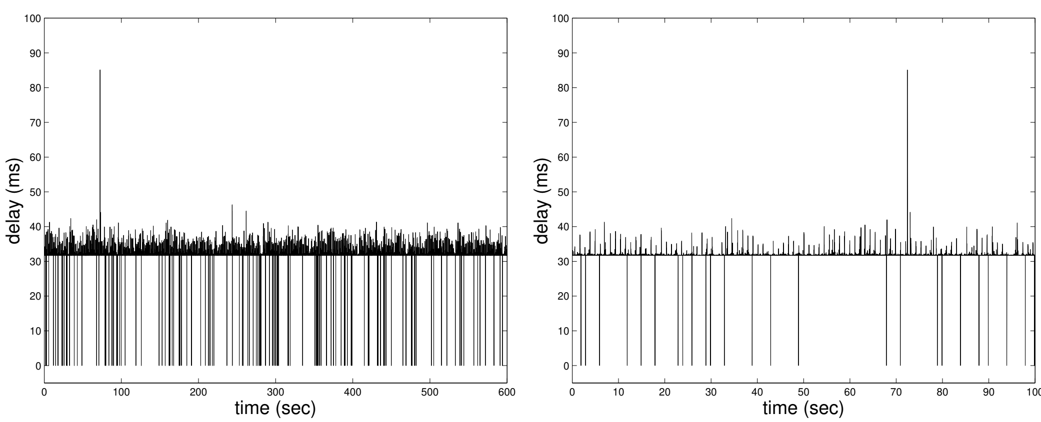
Let us now look at the entire 48 hours for this path (EWR-P_3-SJC). Figure 2.19 shows loss statistics for all 10 minute intervals of the measurement period. Clearly, the loss behavior of this path, remains the same in time. Single packet loss at roughly 0.25% rate happens in all 10 minute intervals. It is interesting to observe that the loss-free intervals between successive single losses across the entire 48 hours period, also follows a CCDF with the shape



(a) Part of one-hour period, where the outage happens (1978 consecutive packets)

(b) Complementary Cumulative Distribution Function (CCDF) of loss-free durations

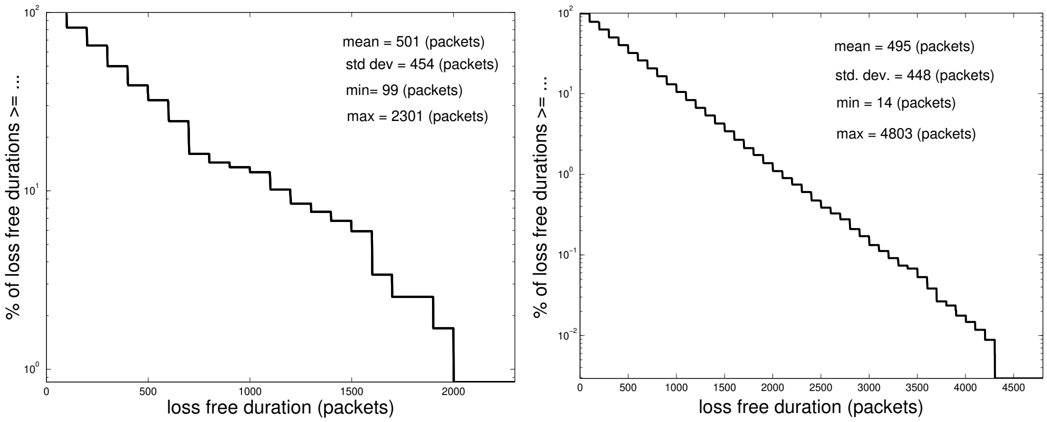
Figure 2.16: One-hour period (Thu 10:00-11:00) on path EWR-P_3-SJC.



(a) 10 minutes trace

(b) Zooming in on the first 100 seconds

Figure 2.17: Example trace from path EWR-P3-SJC, on Thu 06/28/01, from 7:20 until 7:30 (UTC).

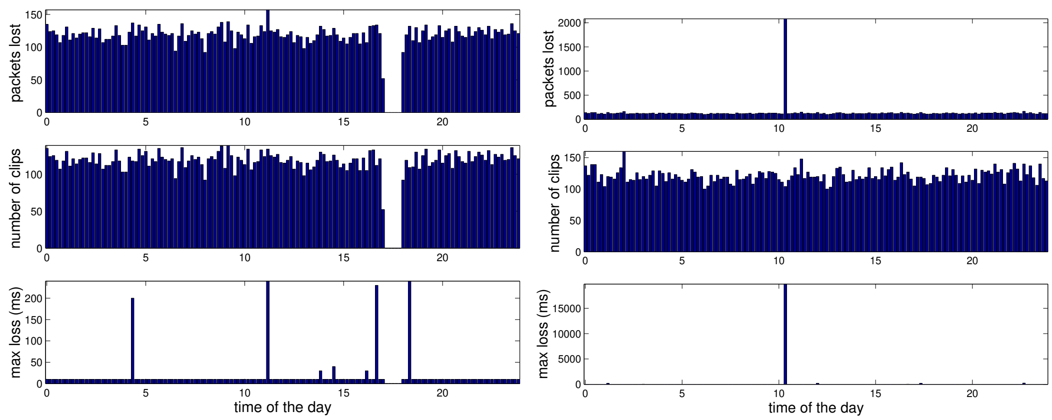


(a) 10 minutes: 7:20-7:30 on Thu 06/28/01 (b) The entire 48 hours period

Figure 2.18: The time between consecutive losses on path EWR-P_3-SJC has a complementary cumulative distribution function (CCDF) with the shape of an exponential, computed in a short and in a long time scale.

of an exponential and almost the same average as for the 10 minutes, see Figure 2.18(b).

The same observations hold for all 4 paths of provider P_3 and for the entire measurement period. Table 2.5 summarizes the loss on provider P_3 and Figure 2.20 shows the complementary cumulative distribution function (CCDF) of loss durations. One path, namely SJC-P_3-AND, has no loss at all. A second path, namely SJC-P_3-EWR, has no loss other than three elementary loss events that last for 599, 594 and 107 packets each. The rest four paths incur regular loss of single packets during the entire measurement period. 100-180 packets are lost in every 10 minute interval, resulting to a loss rate of 0.16-0.28%. The loss durations are mostly single as we can see in Figure 2.20: only 0.04% of loss durations exceed one packet. In addition, there are a few longer loss durations (see columns 9 and 6 in Table 2.5). In Figure 2.20, it can also be seen that there are some longer loss events, either 2-5 or 19-25 packets long. Finally, there is the outage on Thu at 10:20 (UTC) 19.78 sec long, which is synchronized with an outage 18.04 sec long, on the reverse path from SJC to EWR.



(a) Wed 06/27/01 (UTC)

(b) Thu 06/28/01 (UTC)

Figure 2.19: Loss statistics for the path EWR-P_3-SJC per 10 minute intervals for the entire 48 hours measurement period.

Table 2.5: Loss statistics in 48 hours period, for all paths of provider P3

Path			all loss events			single loss		longer loss durations	
From	Prov.	To	packets	elem.	max	number	% loss	number	%loss
SJC	P_3	EWR	1300	3	599	0	0	3	100%
EWR	P_3	SJC	36148	33937	1978	33923	93.9%	14	6.1%
EWR	P_3	AND	34106	33750	181	33736	98.9%	14	0
AND	P_3	EWR	42629	42076	28	41911	98.3%	165	1.7%
SJC	P_3	AND	0	0	0	0	0	0	0
AND	P_3	SJC	44232	41948	1804	41776	94.4%	172	5.6%

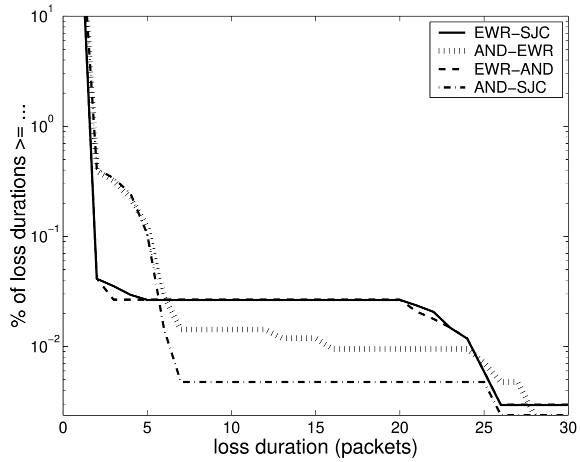


Figure 2.20: CCDF of loss durations for the 48 hours period on four paths of provider P_3 (outages not shown).

2.4 Delay and Delay Jitter Characteristics

As far as delay is concerned, there are two components of interest: the *fixed part* and the *variable part.*

2.4.1 Fixed Delay

The *fixed component* of the delay consists of propagation and transmission delay and, as shown in Table 2.6, it is low on the backbone networks under study. Indeed, transmission delay is negligible on high speed backbone links. The transmission of a 50 Bytes probe takes 0.266 ms on a T1 and 0.009 ms on a T3 access link. Propagation delay clearly depends on the distance of the end-points. Overall, fixed delay is below 12 milliseconds for communication on the same coast and in the range of 32-47 milliseconds for coast-to-coast. Surprisingly enough, there are paths for which the fixed delay was as high as 78 ms, indicating that the shortest route was not followed.

These delay ranges by themselves are below the 150 ms delay, where the effect starts being perceived by the VoIP end-user (as it is discussed in detail in the next chapter, section 3.1.2.4). However, the total end-to-end delay may include additional contributions coming

Table 2.6: Fixed part of the delay for short/medium/long distance paths

Path connecting cities	Fixed Delay Range
in the east coast	3-12 ms
coast-to-coast	31 -47 ms
from/ to THR	28 -78 ms

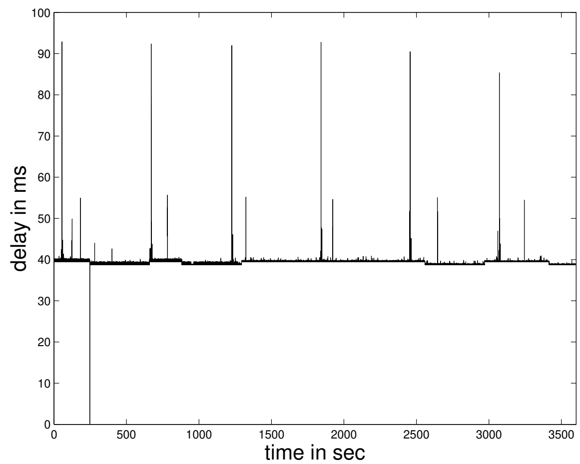


Figure 2.21: Example of changes in the fixed part of the delay, that are not accompanied by outages. Path ASH-P_6-SJC. Wed 4:00-5:00.

from slow access links and regional networks at the edge, from packetization and from playout delay. The higher the fixed delay, the smaller the margin remaining for the other delay components.

We have also observed changes in the fixed part of the delay on paths of providers P_1, P_4, P_6, P_7 in the order of a few (1-3) milliseconds, which by itself is negligible. More frequently than not, these changes are accompanied by long loss periods (outages). Evidence has been provided in section 2.3.2 about elementary loss and outages. Such changes may be due to routing changes and the outages may be due to the time it takes for network reconfiguration. However, the correlation between changes in the fixed part of the delay and the occurrence of an outage is not necessary. Section 2.3.2 provided examples of outages that are not accompanied by a changes in the fixed part. Conversely, Figure 2.21 shows one hour during which the fixed part of the delay changes frequently, without being accompanied by outages.

2.4.2 Delay Variability

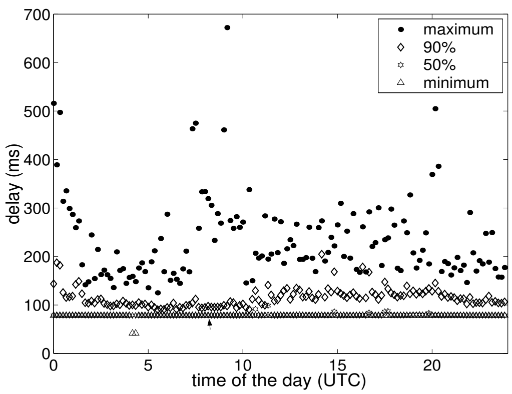
On top of the fixed delay, there is also delay variability for long and short time scales. For long time scales, the delay distribution on some paths changes during the day, as the load changes during working hours. However, most paths are not affected by the time of the day.

To aid in the analysis of delay for such a large set of measurement data, we begin by examining the statistics of delays incurred by probes over 10 minute intervals. We record for each such interval the minimum and maximum delays, and various delay percentiles (primarily the 50^{th} and 99^{th} percentiles). We then plot these for all 10 minute intervals for a 24 hour period. We show in Figure 2.22 such a plot for four different paths.

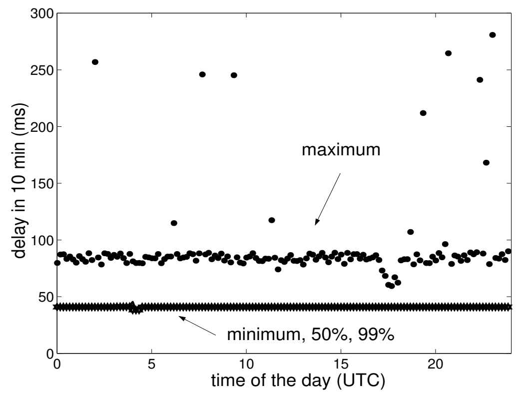
The minimum delay corresponds to the fixed part of the delay, discussed above.[5] The maximum delay and delay percentiles are important to identify intervals during which probes have experienced delays that are large compared to the minimum. If in a one 10-minute interval we observe a high maximum accompanied by increased values of the percentiles, then it means that the interval is of interest for further study. The delay statistics exhibited in Figure 2.22 are also useful to give an indication of the effect of time of day on measured delay. It also aids us in comparing paths; for example, from Figure 2.22 we see that the path THR-P_1-ASH is a path that exhibits high peaks as well as high percentiles most of the day, while at the other extreme the path SJC-P_7-ASH is a path that exhibits rather low delays. The path SJC-P_2-ASH is a path that is usually good (similar to P_7) for most of the day, but does incur higher delays over a certain period of the day. The path EWR-P_4-SJC has a periodic pattern that leads to the percentiles observed.

We are primarily interested in analyzing the delay variations in short time scale, also called delay jitter, identifying the various possible delay jitter patterns and characterizing them. This requires that we plot the delay of individual probes versus their respective send times. An example is shown in Figure 2.23. The delay variations that we see show that

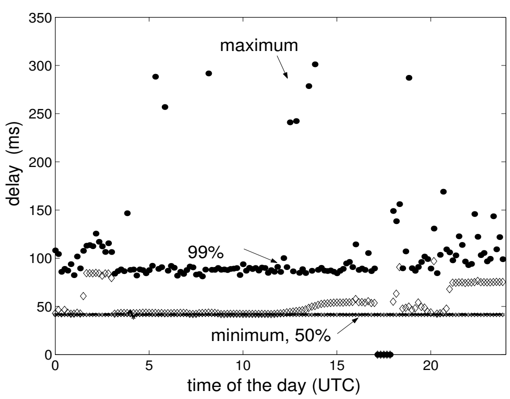
[5] There is a 50 minutes period in some plots, during which all delay percentiles are zero, including the minimum. This corresponds to a 50-minutes interruption of the EWR measurement facility during that period.



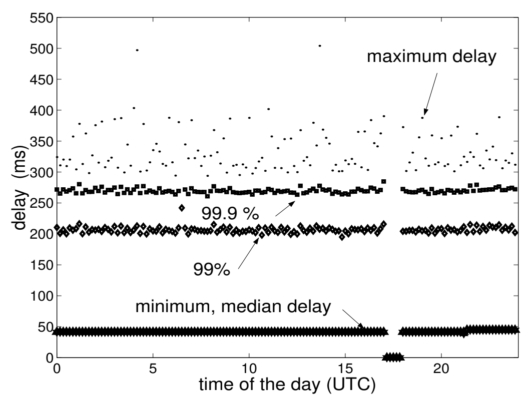
(a) THR-P1-ASH on Wed 06/27/01. A path with high delay and high delay variability



(b) SJC-P7-ASH on Wed 06/27/01. A path with low delay variability, except for infrequent spikes



(c) EWR-P2-SJC on Thu 06/28/01. A path with a high and a low delay pattern.



(d) EWR-P4-SJC on Wed 06/27/01. A path with a periodic pattern

Figure 2.22: Delay percentiles per 10 minutes intervals for a 24 hours period and four different paths

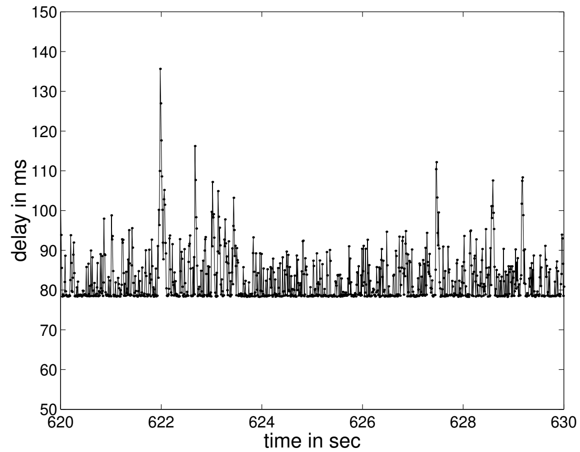


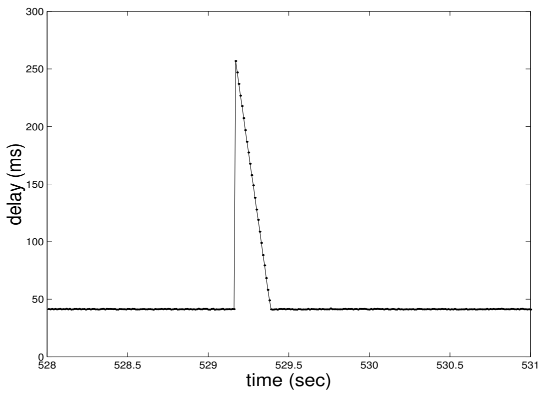
Figure 2.23: Delay of individual probes on path THR-P1-ASH, on Wed 06/27/01 at 2:10

the delay is constantly varying within a certain relatively small range above the minimum. There are frequent visits to the minimum, indicating that the corresponding paths are lightly loaded. This type of delay variation prevails and corresponds to what one might call the normal pattern.

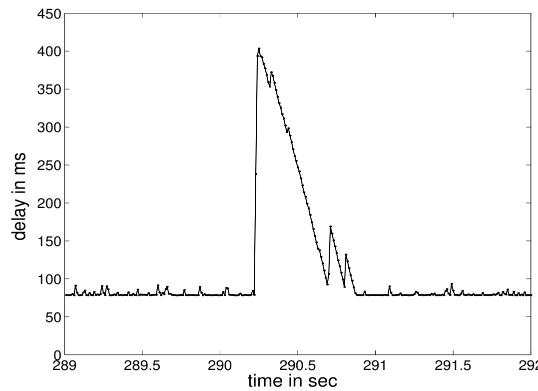
Most of the time and for most backbone paths, the delay variability was within a few milliseconds of the fixed part. This is expected as backbone networks are usually overprovisioned with enough bandwidth to have empty queues most of the time. The lowest jitter is incurred by providers P6 and P7, for which the 99^{th} jitter percentile is from 0.1 to 0.7 milliseconds. However, there are higher delay variations that occur in the form of spikes (as opposed to a slow changing component).

By *spike* we refer to a number of packets that have significantly higher delays than the rest and they follow roughly the triangular shape shown in Figure 2.24(a). There is a sudden sizable jump in delay for a probe, followed by a succession of probes delays decreasing by 10 ms each. We note that since probes are sent deterministically one every 10 ms, the delays of probes succeeding the peak follow a line with a slope of -1. This shape indicates that packets arrive bunched up at the receiver.

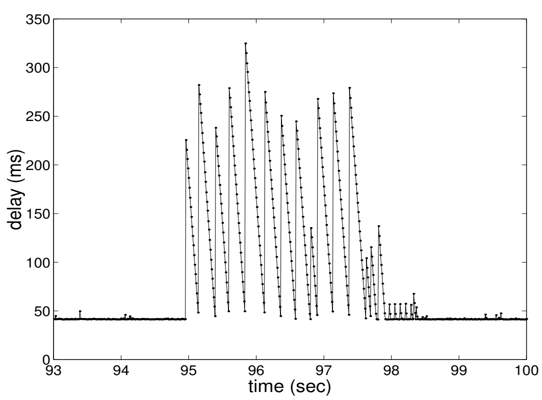
The simplest spike is the one with the perfectly triangular shape, shown Figure 2.24(a):



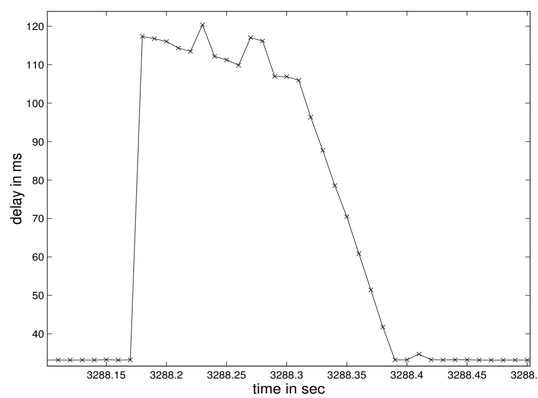
(a) "Model" spike, on path SJC-P7-ASH, on Wed 06/27/01 at 2:00 (UTC)



(b) High spike, on path THR-P1-ASH, on Wed at 0:00



(c) Clustered spikes, on path EWR-P4-SJC, on Wed at 21:00



(d) Spike (an exception to the triangular shape), on path SJC-P5-EWR, on Wed at 17:00.

Figure 2.24: Example spikes

a sudden sizable increase in delay, followed by a 45 degrees slope linear decrease. The only parameter characterizing such a spike is the magnitude of the jump, or equivalently the peak delay. The width of the spike is almost equal to the jump up to the peak delay. The spike shown in Figure 2.24(b) is not as simple: there is some jitter in the decreasing slope and there are several smaller peaks that follow the first and tallest peak. In this case, the entire event may be characterized by the magnitude of the first (highest) peak, the width of the spike and the height of the smaller peaks. There are yet other situations that differ from the above description. An example is shown in Figure 2.24(c). It consists of a rapid succession of spikes of similar heights lasting over three seconds. Another example is shown in Figure 2.24(d): following the sudden jump in delay, a number of probes incur roughly the same delay as the peak, before the linear decrease in delay is observed. This is an exception to the triangular spike shape. However, the large majority of spikes in the traces follow the triangular shape of 2.24(a).

The characteristics of spikes and the specific pattern vary from path to path and over time. We illustrate this fact by examining the four example paths: THR-P_1ASH, SJC-P_7-ASH, EWR-P_2-SJC and EWR-P_4-SJC. We are guided by their delay statistics for 10 minutes intervals, shown in Figure 2.24 above, to select periods of time on which to study in greater detail. We will see that lower delays follow random patterns (consisting of spikes with random peaks at random distances) while higher delays follow periodic patterns (the same shape of spike or cluster of spikes repeated periodically). We also identify similar behaviors in other paths. We finally provide a discussion on the low-frequency delay components.

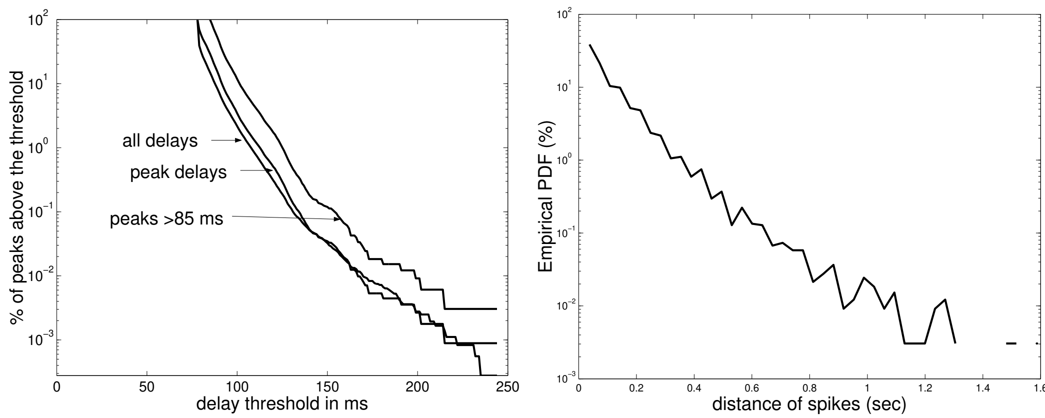
2.4.3 High Delay Variability

Example Path THR-P_1-ASH

The path from THR to ASH belonging to provider P_1, shown Figure 2.22(a), is a typical example of path with high delay and high delay variability. In general, there are no strong low-frequency delay components on these path (apart from a few cases discussed separately in section 2.4.6). For most of the time, delay happens in the form of spikes of various heights spaced at various intervals from each other. Although the characteristics of the spikes vary during the day, when we zoomed in on different parts of the day, we found that delay follows one of three distinct patterns.

The first pattern is what we call *random delay pattern* ; it holds for most of the day, when delays are relatively low. The second and the third pattern happen when delay is high, are associated with an increase in delay percentiles in Figures 2.22(a) and have some structure in them. The second pattern happens when the maximum value increases considerably and we call it *very high peaks* pattern. The third pattern, which we call *block pattern*, is associated with an increase in the 50^{th} and 99^{th} delay percentiles and it happens 9 times during the measurement period. We now discuss the three patterns in detail.

Random Delay Pattern. Most of the time, delay is low (roughly below 150 ms) and follows a random pattern, consisting of spikes with random peaks that happen at random intervals. For example, the pattern shown in Figure 2.23 corresponds to such a normal period (from 2:00 - 3:00 UTC on Wed 06/27/01). Figure 2.25(a) shows the complementary cumulative distribution function (CCDF) for all probe delays and for the peak delays in particular. It is interesting to note that the distribution of all probe delays is very close to the distribution of the peak delays, which can be justified by the triangular shape of the spikes. The shape of this CCDF is almost a straight line, which indicates that the exponential distribution (with a CCDF that is exactly a straight line) is a good fit. We consider peak delays of a considerable size to be those above 85 ms and we observe that their distribution also follows an exponential shape (with a mean of 92 ms). The period of time separating these spikes (above 85 ms) also follows a roughly exponential distribution, as shown in Figure 2.25(b). The same observations hold for most of the day, when delays are small. It is worth mentioning that a recent measurement study by AT&T, [105], followed a similar approach and came up with similar findings: (i) spikes above a certain magnitude were considered, although no clear cutoff point could be identified, and (ii) exponential



(a) CCDF of peak delays

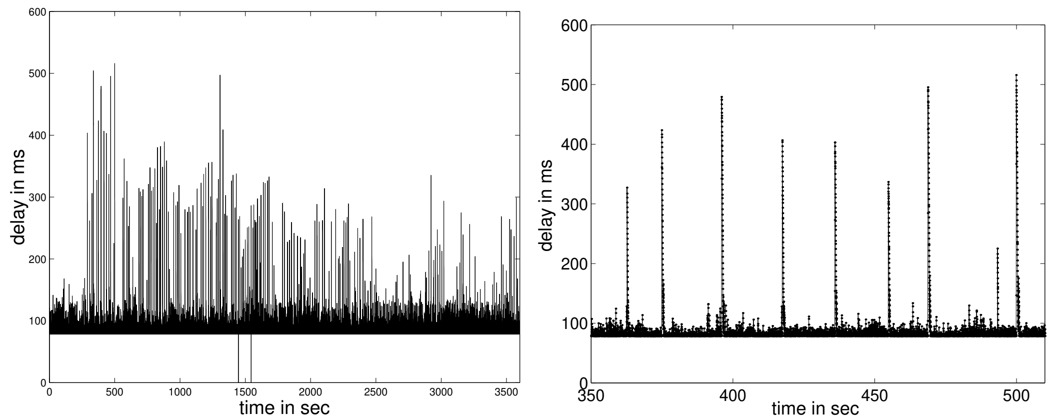
(b) PDF of distance between consecutive spikes

Figure 2.25: "Random delay pattern" on path THR-P1-ASH, Wed 2:00-3:00

interarrival times between spikes were observed.

Very High Delays. This pattern happens when maximum delay reaches the highest values observed (400-700 ms in Figure 2.45(a) and (b)), e.g. during the periods 0:00-1:00, 6:00-10:00 and 20:00-21:00, 23:00-00:00. An example of such an hour is shown in Figure 2.26(a). Figure 2.26(b) zooms in 50 seconds; we can see that these high peaks happen every 10-20 ms. Figure 2.24(b) shows the details of one of these spikes: a high peak is followed by many smaller ones. When we zoomed in on remaining high spikes, we noticed that they have the same structure.

Block Pattern. The second regular pattern consists of a cluster of spikes repeated periodically. The first spike is of a higher fixed height and is followed by many spikes half as high. Cluster of spikes of the same shape and height are repeated periodically. Figure 2.27 shows two examples of this block pattern. The first example is shown in the top two plots. It happened at 14:00 on Wed 06/27/01 and it lasted for 5 minutes; the spikes were 250 ms high and the cluster was repeated every 2-3 sec. The second example is shown in the bottom two plots. It happened at 3:00 on Thu 06/28/01 and lasted for 9 minutes; the

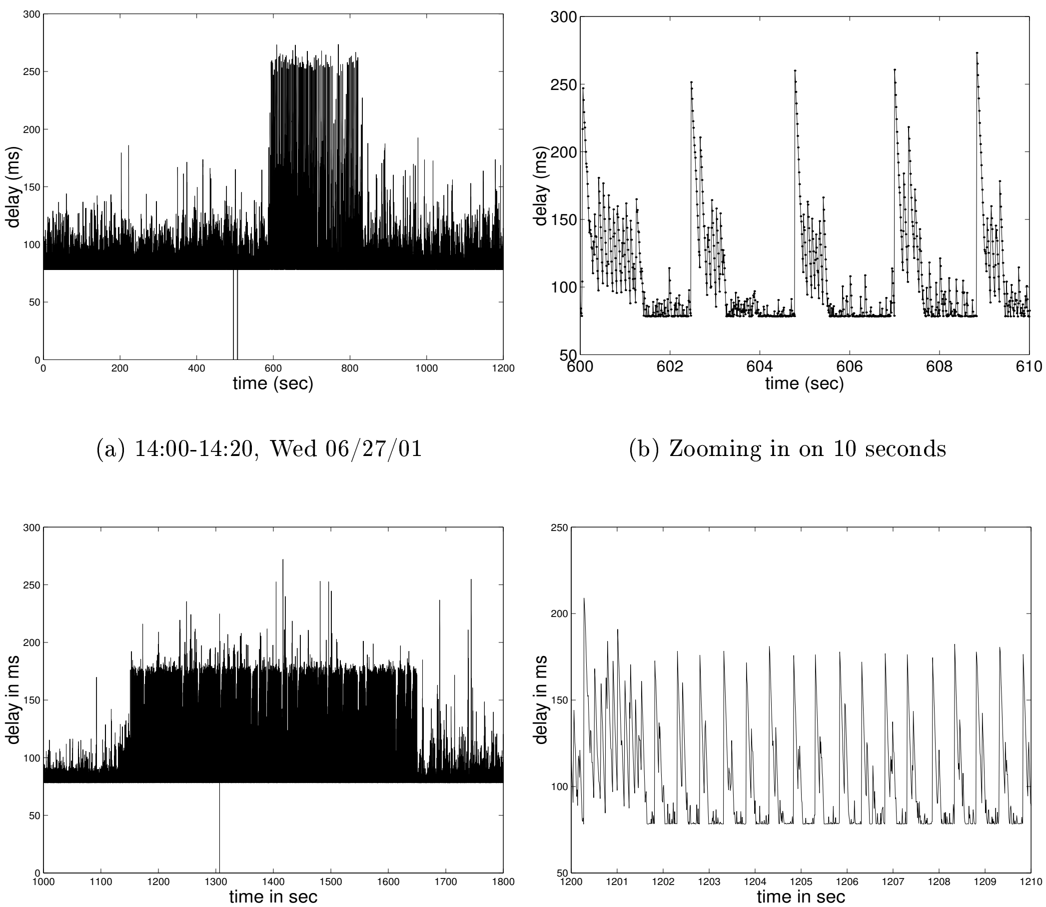


(a) Wed 0:00-1:00 (b) Zooming in

Figure 2.26: Example of the "very high delay" pattern on path THR-P1-ASH.

spikes were 180 ms high and they were repeated every 0.5 sec. We call this pattern "the block pattern", due to the box shape in the left two graphs. It occurs 9 times in the entire measurement period and it leads to an increase in the 50^{th} and 99^{th} delay percentiles in Figure 2.22 (or in the later ones 2.45(a) and (b)).

Delay Characterization in the Presence of Many Patterns. The default delay pattern is the random one. In addition, when delays are high, one of the other patterns may also take place. In order to model such a trace we consider sets of peaks above a certain magnitude, starting from higher and proceeding with smaller magnitudes. We characterize each set of delays by describing the pattern and giving the distributions for the peaks and for the distances between peaks. One can then generate a set of peaks with the same distribution. The rest of the delays can be found from the peak delays by using the triangular spike shape. We find that lower delays usually follow the random pattern and have peaks roughly exponential (as in Figure 2.25(a)) at roughly exponential distances (as in Figure 2.25(b)). We also characterize the higher delays by describing their pattern and by giving the distribution for the peaks (that ends up being exponential, constant or bimodal) and the distance between two consecutive spike clusters (it turns out to be roughly periodic).



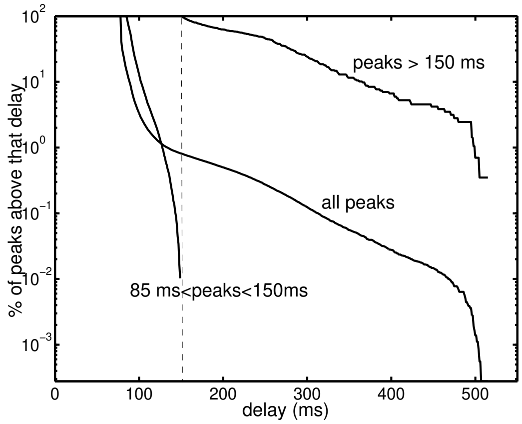
(a) 14:00-14:20, Wed 06/27/01

(b) Zooming in on 10 seconds

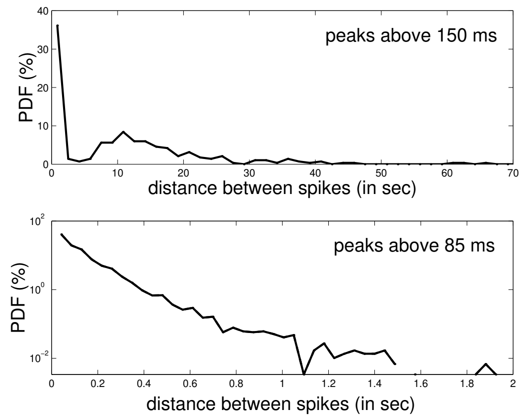
(c) 3:10-3:30 (UTC), Thu 06/28/01

(d) Zooming in on 10 seconds

Figure 2.27: Examples of the "block pattern" on THR-P1-ASH



(a) CCDF of peaks of the spikes

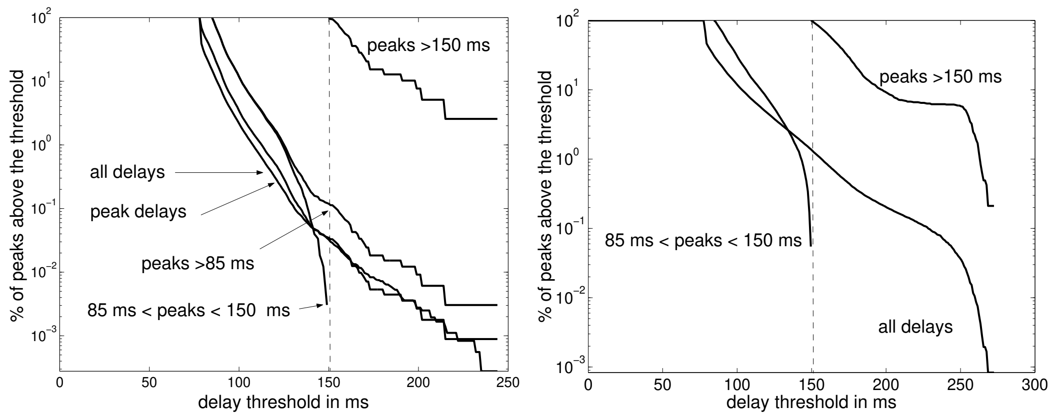


(b) PDF of time (distance) between spikes

Figure 2.28: Path THR-P_1-ASH. Delay characterization for one hour (Wed 0:00-1:00) with random pattern and very high peaks

Let us take the example of trace in Figure 2.26(a) that has a random pattern and very high peaks. The CCDF of all peaks, in Figure 2.28, has a knee around 150 ms. In addition, 150 ms is an acceptable delay value for voice purposes. Therefore, we choose to characterize separately the delays below and above 150 ms. In Figure 2.28(a), we see that the distribution (CCDF) of peaks above 150 ms has a roughly exponential shape. In the top graph of Figure 2.28(b), we see that the distribution (PDF) of distances between these higher spikes has a maximum around 10 seconds, and can be as high as 70 sec. As for the lower delays, we see that the CCDF of their peaks seems roughly truncated exponential CCDF and the distance between them has a PDF that has also the exponential shape (with a mean of 0.12 seconds or 12 packets). Note that we model only spikes of significant size, i.e. above 85 ms, although 85% of all packet delays are in the [78 ms, 85 ms] range. If we considered all peaks, then the large majority of spikes would be small with distances of 1-2 packets from each other, thus hiding the higher spike patterns.

One can apply the exact same modeling steps on any trace. Figure 2.29(a) refers to the example hour with random pattern only; see Figure 2.23. Notice that Figure 2.29(a)



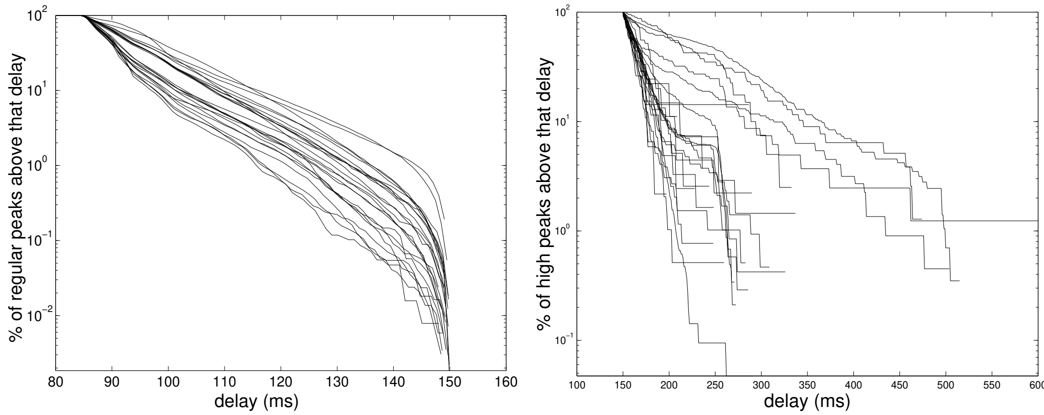
(a) Wed 2:00-3:00 (b) Wed 14:00-15:00

Figure 2.29: Path THR-P_1-ASH. CCDF of delay for two more hours: (a) Wed 2:00-3:00 has only random pattern (b) Wed 14:00-15:00 has random, block, and low-frequency delay patterns

is a refinement of the distribution provided in Figure 2.25 for all delays. The few samples exceeding 150 ms seem to follow a roughly exponential CCDF. The delay distribution for another example hour is shown in Figure 2.29: it is the hour with the block pattern (Wed 14:00-15:00, see Figure 2.27 (a) and (b)). The knee in the distribution of higher delays is due to the constant height of the spikes in the block pattern. Finally, Figure 2.30 gives the distributions for the lower and higher delays for all 24 hours on Wed 06/27/01. We see that the CCDF of lower delays have the shape of a truncated exponential, in the log scale. There is some small spread in the slope during the day. The higher lines correspond to hours with additional patterns and high delay percentiles. These high delays may follow an exponential or bimodal distribution with a larger spread.

2.4.4 Low Delay Variability

In the previous section, we studied a path with high delay variability. In this section, we study paths at the other extreme.



(a) 85 ms < peaks < 150 ms

(b) Peaks > 150 ms

Figure 2.30: Path THR-P1-ASH. Delay characterization for 24 hours on Wed 06/29/01 (UTC).

Example Path SJC-P_7-ASH

SJC-P7-ASH is a path in a very well provisioned network that exhibits very low delay variations. Its delay percentiles have been shown in Figure 2.22(b) and an example 10 minutes period is shown in Figure 2.31. We can see that delay is practically constant most of the time: delay lies between 40.5 and 42 ms and the 99$^{\text{th}}$ jitter percentile lies between 0.59 and 0.715 ms.

However, there are occasionally higher spikes, that have the perfect triangular shape. Some of them, with peaks no more than 80 ms are shown in Figure 2.31. An example of another perfectly triangular spike on this path was also shown in Figure 2.24(a). Another example was shown in Figure 2.2(b): 80 ms high spikes were repeated every 10 minutes. The maximum values in 2.22(b) indicate that spikes can occasionally be 250-300 ms high.

Figure 2.32(a) shows another example hour on this path: 80ms high spikes happen once every 10 minutes. The complementary cumulative distribution function (CCDF) of the peak delays is shown in Figure 2.32(b). This distribution has a sharp knee around 42 ms. Indeed, out of the 360000 packets in an hour, 119869 packets have delays above 41 ms, 43621 packets

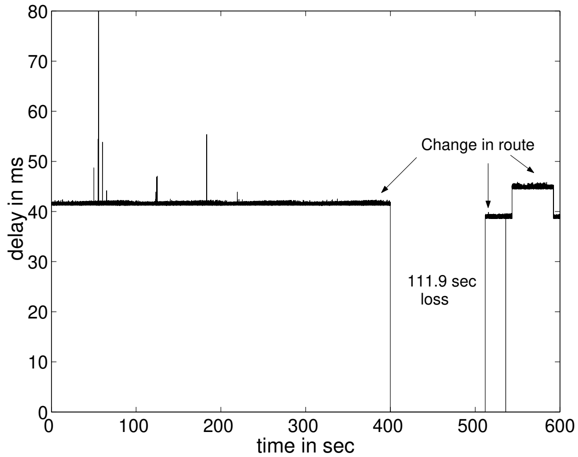
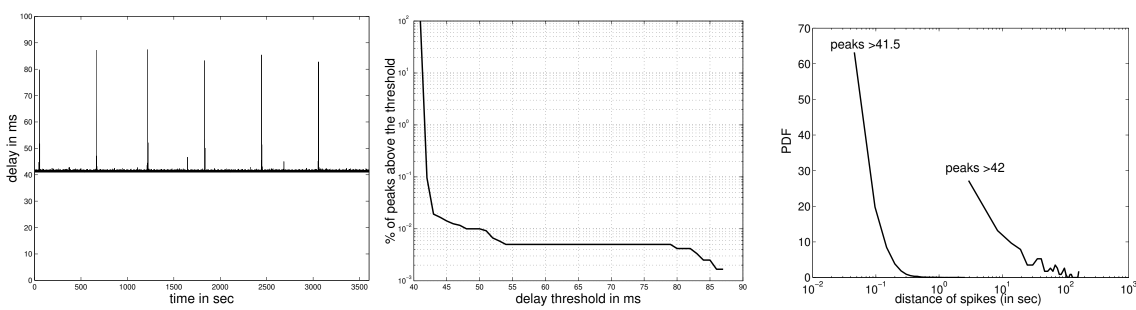


Figure 2.31: 10 minutes of path SJC-P7-ASH, Wed 4:00-4:10



(a) One hour trace (b) CCDF of peak delays (c) PDF of distances

Figure 2.32: Delay on path SJC-P_7-ASH, Wed 0:00-1:00.

above 41.5 ms, only 115 packets above 42 ms, 12 packets above 50 ms and only 5 above 70 ms. The distribution (PDF) of the distances between spikes is shown in Figure 2.32(c).

We speculate that the 10 minutes periodicity has to do with the measurements themselves: we collected and stored probe measurements in a single file every 10 minutes. Similar spikes in other traces, are probably hidden by the larger delay variability in those traces.

Other Paths with Low Delay Variability

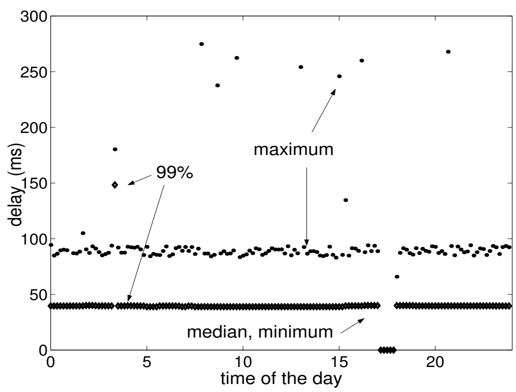
There are more paths with similar behavior, i.e. with very low delay variability and with infrequent (every 10 minutes) spikes.

Such are ***all 9 paths of provider*** P_6**,** consistently for the entire measurement period. They have low delay (32.3-39.1 for the long distance and 3.25-7.1 ms for the paths in the east coast) and the lowest delay variability (e.g. 99^{th} jitter percentile was from 0.1 to 0.6 ms) among all 43 paths. Let us look at an example path, namely ASH-P_6-SJC. Figures 2.33(a) and (b) show the delay percentiles in 10 minute intervals for 48 hours. Figures 2.33(c) and (d) show in detail two hours on Wed. We can see that delay lies in a 2 ms range above the minimum value, which results in the 50^{th} and 99^{th} percentiles being very close to the minimum for the entire period. There are also 90 ms spikes (followed by a smaller 55ms spike) every 10 minutes, which are responsible for the maximum values being around 90ms.

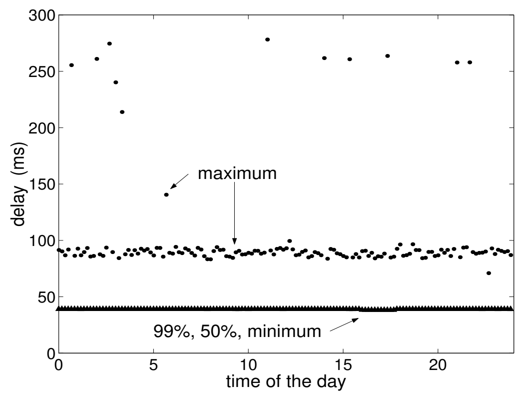
In both the percentiles and the time-trace figures, we can observe frequent changes in the minimum delay by a few milliseconds; unlike all other providers, these changes are not accompanied by outages. Occasionally, the maximum becomes as high as 250-300ms, usually accompanied by some loss. For example, at the end of the 7:50 interval in Figure 2.33(d), a high spike happened, accompanied by 21 packets loss. Between 3:20 until 3:30 in Figure 2.33(c) there is increased delay and loss. A closer look shows that this is actually the complex loss event of Figure 2.7 that has been discussed in Section 2.3.

There are yet other paths with good delay behavior, such as ***four paths of provider*** P_1**.** An example is path THR-P_1-SJC. The minimum delay is 24 ms and the maximum delay is 70 ms during the night (note that the local time is 8 hours behind the UTC time used in the plots) and it can be higher during business hours as we can see in Figure 2.34(a). However variability is significantly lower than the other paths of P_1 (e.g. THR-P_1-ASH in section 2.4.3).

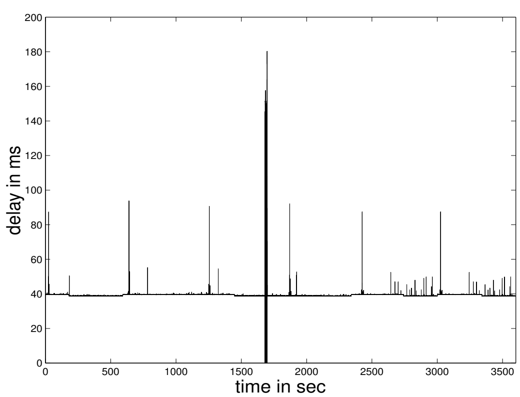
Similarly, the ***six paths of provider*** P_3 have also low delay variability consistently across the entire measurement period. Delay percentiles for path EWR-P_3-SJC are shown



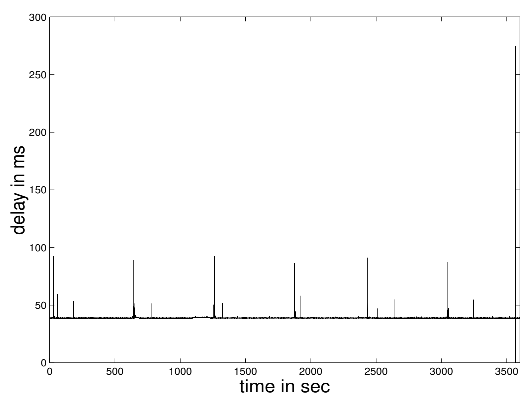
(a) Wed 06/27/01 (UTC). Delay percentiles per 10 minutes intervals.



(b) Thu 06/28/01 (UTC). Delay percentiles per 10 minutes intervals

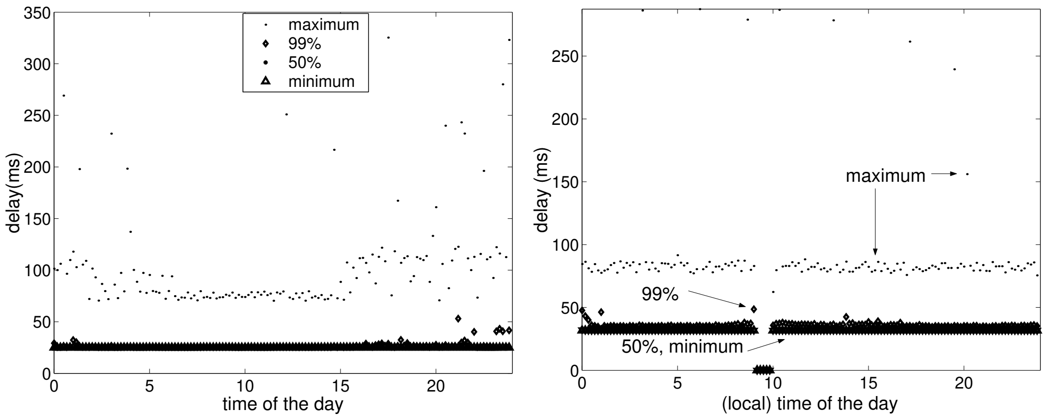


(c) One-hour period: 3:00-4:00



(d) One hour period: 7:00-8:00

Figure 2.33: Delay on path ASH-P6-SJC.



(a) THR-P1-SJC, on Wed

(b) EWR-P3-SJC, Thu

Figure 2.34: More example paths (from P_1 and P_3) with low delay variability

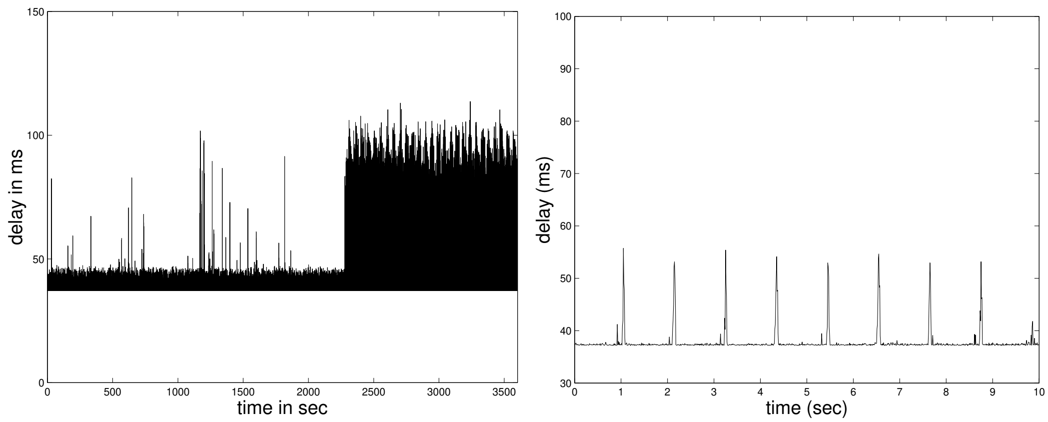
in Figure 2.34(b).

2.4.5 Mixed Characteristics

The paths described in the above section, had consistently low delay variability for the entire day. The paths described in this section have also low delay variability (in the order of 10-20 ms). However, there are periods when the delay pattern changes to higher ranges. Such transitions are usually accompanied by high loss periods. Correlation between loss and delay increase has been also observed in measurements used in another study, [69]. The examples of complex loss events in Section 2.3.3 are examples of such transitions between delay states that are accompanied by loss. Let us see more examples of paths with such mixed characteristics; they mainly belong to providers P_2 and P_5.

Example Path EWR-P_2-SJC

The path EWR-P_2- SJC exhibits mixed delay characteristics. The delay pattern consists of spikes at almost constant height, repeated periodically every 1-1.5 sec. The height of these spikes varies during the day, as the pattern alternates between a low and a high delay



(a) One hour (Thu 1:00-2:00) with a change in the delay state

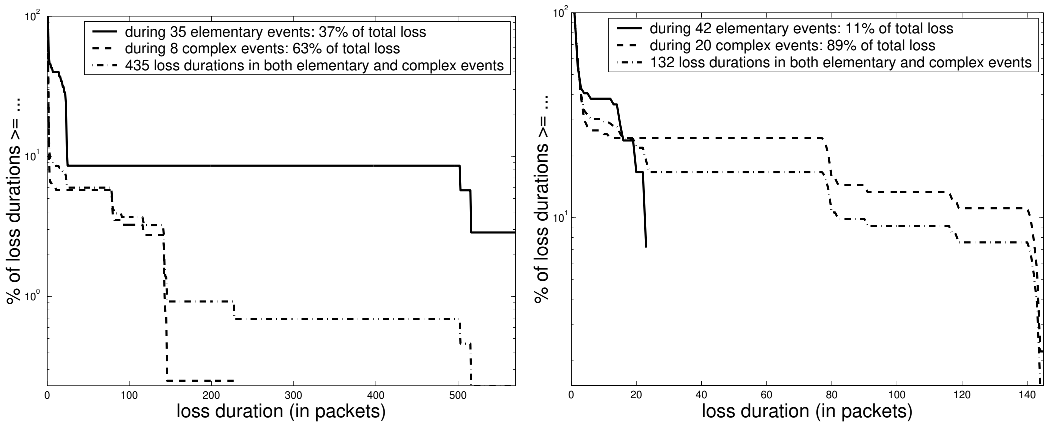
(b) 10sec from Thu 14:00-15:00 - periodic small spikes

Figure 2.35: Path EWR-P2-SJC. Zooming in on delay patterns.

state. Figure 2.35(a) shows an hour where a transition between states occurs. Figure 2.35(b) shows 10 seconds from another hour, when the delay pattern consists also of periodic spikes of fixed height repeated every 1 sec.

Figure 2.22(c) showed the delay percentiles in 10 minute intervals for one day. For most of the period, the 50^{th} and 99^{th} percentiles are very close to the minimum, corresponding to the low delay state. However, the maximum delay is around 90 ms due to higher spikes, like those in the middle of Figure 2.35(a). Occasionally the maximum delay is as high as 300 ms, usually accompanied by loss durations 19-25 packets long; an example has been shown in earlier section in Figure 2.2(a). Periods of time during which the 99^{th} percentile increases correspond to the high delay state.

An important observation is that loss happens during the transitions to higher delay. For example, the complex loss events on Thu 20:10 and Wed 3:30 (studied in detail in section 2.3.3 - Figures 2.9 and 2.11 respectively) accompanied changes in the delay pattern. Figure 2.48 shows the distribution of loss durations on the two paths of this provider. We can see that complex loss events constitute most (63% and 89%) of the loss on these paths. Even



(a) EWR-P2-SJC (b) EWR-P2-ASH

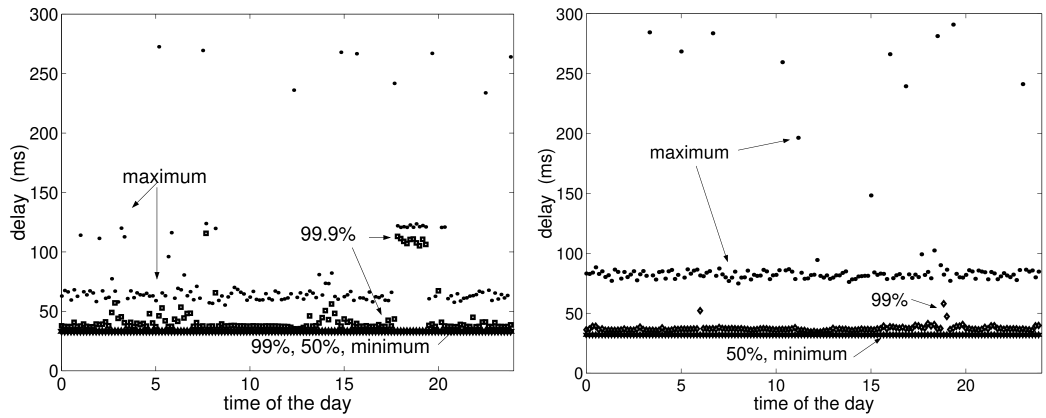
Figure 2.36: The two paths of provider P_2. Distribution (CCDF) of loss durations.

the loss that appears as elementary loss event, is due to long loss durations a few seconds (more than $\tau = 2sec$) far from the rest of the complex loss events, as in Figure 2.9.

More Example Paths: Provider P_5

The six paths of provider P_5 have also mixed delay characteristics. They have low fixed delay (32-33 ms for the long distance and 6-7 ms for the short distance paths) and in general low delay variability (spikes up to 50-90 ms every 1 second) except for some periods of the day that we now discuss in detail. Figure 2.37(a) and (b) show delay percentiles on paths SJC-EWR and EWR-SJC respectively, for a 24 hours period.

Let us look at the first path, SJC-P_5-EWR and examine what happens during the hours that the 99.9$^{\text{th}}$ delay percentile increases. Figure 2.38(a) shows in more detail such a period: Wed 17:00-18:00. The first part of the trace is the typical behavior of this path. Toward the end of this hour, there are higher spikes at almost constant height 110-120ms. One of them has been shown in detail in Figure 2.24(d) and a second one is shown in 2.38(b). This shape is an exception to the triangular spike shape. These spikes appear for the next one and a half hour (until 19:30). They also appear whenever there is an increase in the 99.9$^{\text{th}}$



(a) SJC-P5-EWR, on Wed 06/27/01 (UTC) (b) EWR-P5-SJC, on Thu 06/28/01 (UTC)

Figure 2.37: Delay percentiles per 10 minute intervals for two paths of provider P5.

percentile in Figure 2.37(a).

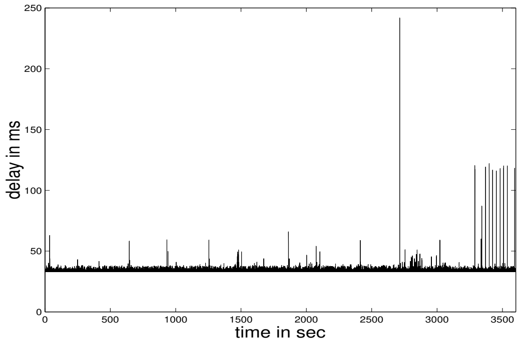
Let us now look at the second path, EWR-P_5-SJC, which seems, from its percentiles in Figure 2.37(b), to have even less delay variability. However, there are four ten minute intervals where there is an increase in the 99.9$^{\text{th}}$ percentile that is due to a sustained increase in delay. Figures 2.38(c) and (d) show an example of this pattern. Notice that this change in delay state is also accompanied by some loss. Figures 2.38(e) and (g) show two other examples on paths of provider P_5, with simultaneously higher delay and loss.

2.4.6 Low-Frequency Delay Components

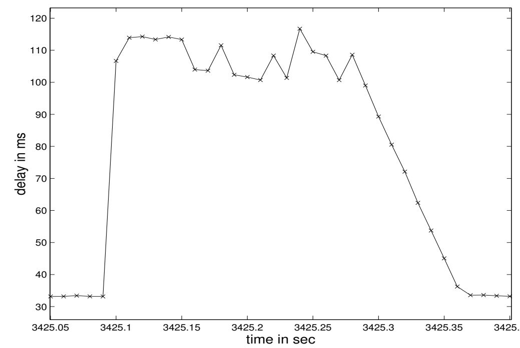
Low-Frequency Delay Components in the Traces.

For most of the traces and for most of the time, delay happens in the form of spikes that start from and return to the minimum delay. For these cases, the most natural characterization of delay is in terms of statistics for the height and the distance of spikes. In this section, we focus on the low-frequency delay components.

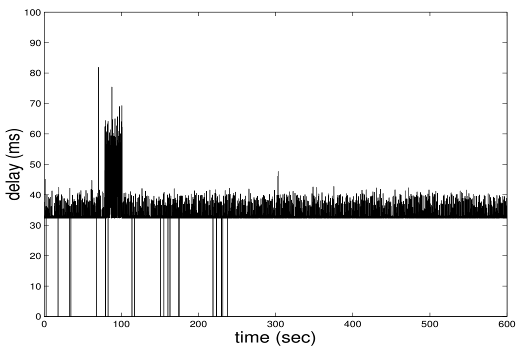
There are only a few cases where there is a sustained increase in delay. An example of



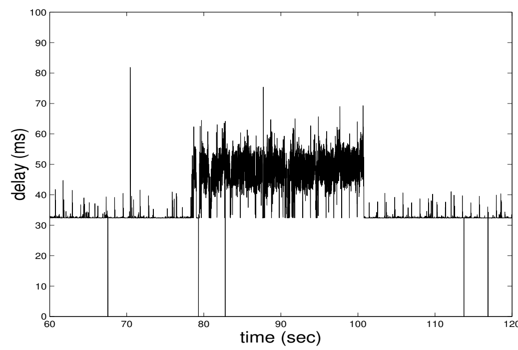
(a) SJC-P5-EWR, Wed 17:00-18:00



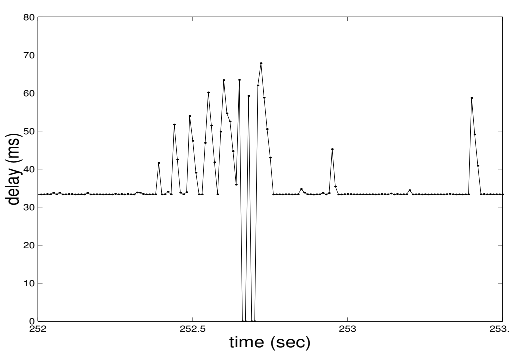
(b) Zooming in on one of the last spikes on Wed 17:00-18:00 (figure at left)



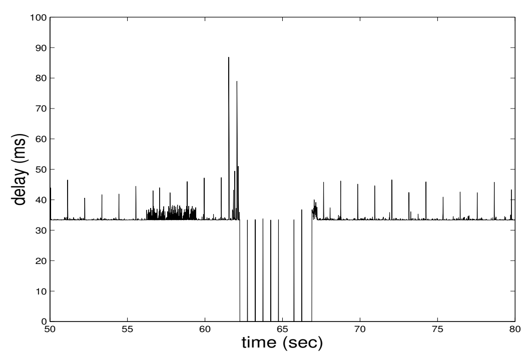
(c) EWR-P5-SJC, Thu: 6:00-6:10



(d) Zooming in the high delay period on Thu 6:00-6:10 (figure at left)

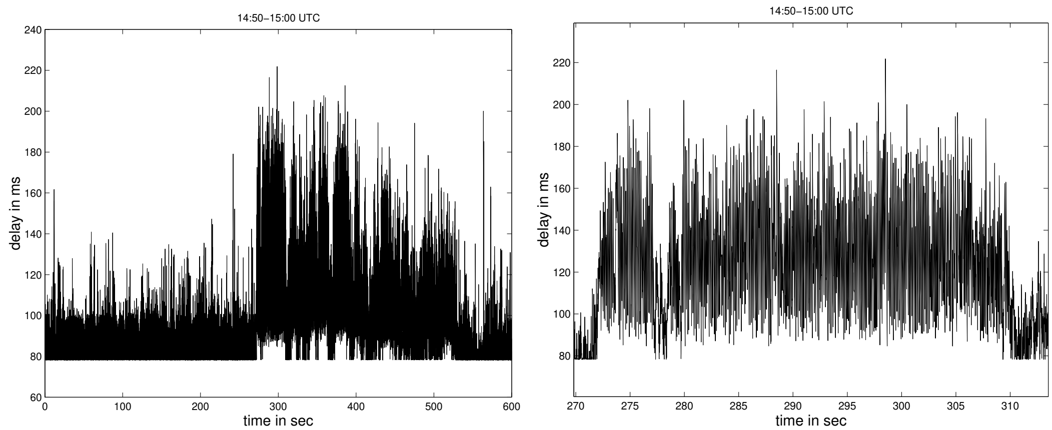


(e) ASH-P5-SJC, 10 sec on Thu 2:10



(f) ASH-P5-SJC, 30 sec on Thu 2:40

Figure 2.38: Example of delay and loss patterns on provider P_5



(a) 14:50-15:00 (UTC), Wed 06/27/01

(b) Zooming in on 45 seconds

Figure 2.39: Example of sustained increase in delay on THR-P1-ASH.

a 10 minutes period with a sustained increase in delay, on a loaded path of provider P_1, is shown in Figure 2.39(a). The spikes are more frequent and there is a sustained increase in delay lasting for hundreds of seconds. A blow-up of 35 seconds is shown in Figure 2.39(b). This example indicates that there is a low-frequency delay component, on top of which spikes are super-imposed.

This pattern also leads to an increase in the median delay in Figure 2.45. It happened 18 times in 48hours on this path. It also happened on four more paths of provider P_1. Increases in the median delay happen also on paths with mixed delay variability of providers P_2 and P_5, and they last for periods ranging from seconds to hours. Examples have been shown in the previous section. However, in those cases the change in the median delay is due to a different spike pattern (e.g. higher spikes), and can be still characterized by the height and the distance of the spikes.

On Modeling Network Delay using a Gamma Distribution.

A. Mukherjee, in his [70] paper in 1993, studied loss and delay on regional, backbone and cross-country paths. He sent probes infrequently (i.e. clusters of 20 probes, spaced 1 second apart, every 1 minute) to avoid increasing the load in the network. A spectral decomposition indicated the presence of dominant low-frequency delay components. He further smoothed out the average delay using a low-pass filter. He found that the distribution of the smoothed delay was very well approximated by a shifted gamma distribution. There was no intuitive reason why a gamma distribution was chosen at that point, other than the facts that (i) the PDF resembled that of a gamma distribution (ii) the gamma distribution is very flexible and can model a wide range of empirical distributions. The parameters of the gamma distribution, namely the shape and scale, varied during the day for each path, depending on the load. Thus, they had to be estimated for a short time period. The shift corresponded to the fixed delay on the path, i.e. propagation and transmission.

Later on, e.g. in [13] by Chou et.al., the gamma distribution was interpreted as a sum of exponentials, that represents the queuing delay incurred by a packet at consecutive routers along the end-to-end path. Recently, a shifted gamma distribution has been widely used by network-adaptive applications to model network delay.

We now test whether the delay on the example trace of Figure 2.39, can be modeled using a gamma distribution. We consider two parts of the trace. The first part is shown in Figure 2.40(a) and corresponds to 10 seconds, from 70 to 80 sec, during which the smoothed delay does not vary much. The second part is shown in Figure 2.41(a) and corresponds to 100sec, from 300sec to 400sec, during which the smoothed delay is more pronounced.

For the first part of the trace of Figure 2.39, we show the actual probe delays, as well as the smoothed delay (using an averaging filter over 200ms), in Figure 2.40(a). In Figure 2.40(b), we show the empirical PDF for both. The QQplot test for the smoothed delays is shown in Figure 2.40(c); it is very close to the straight line. This indicates that the smoothed delays can be modeled well by a shifted gamma distribution, whose parameters can be estimated. The scale can be estimated from the slope of the QQplot, the shape can be found from the mean and variance of the delays, the shift can be approximated from the shift on the delay axis of the QQplot. However, a similar test fails for the instantaneous delays.

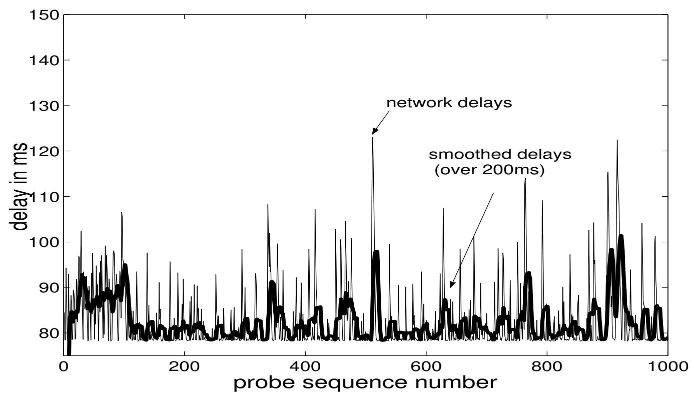
The above statements are true for the first 150sec of the example trace, as well for other traces with similar characteristics. However, modeling the low-frequency delay component on this part of the trace is of limited importance, as this slow-varying component does not seem to have significant magnitude.

On the other hand, it would be more valuable to model the low-frequency delay component on the second part of the trace shown in Figure 2.39. As shown in Figure 2.41(a), the slow-varying component is more pronounced. Unfortunately, we were not able to achieve a good fit to a gamma distribution in none of the following cases: (i) considering all probe delays (ii) considering smoothed versions using different averaging intervals (iii) considering various durations of the trace, namely 10 seconds, 1 minute and 100 seconds. Determining appropriate intervals for delay modeling, as well as the transitions between them, is a difficult problem, (as discussed in the paper "On the Constancy of Internet Path Properties" [105]). The fit was also bad for other traces with high load.

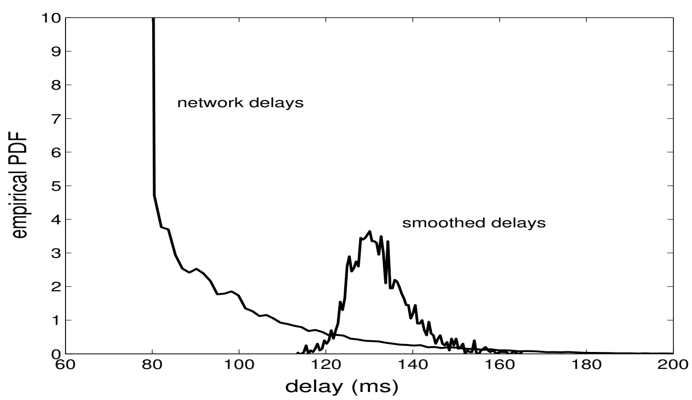
In conclusion, we found the use of a gamma distribution (i) inappropriate for the network delays themselves at the granularity of 10 ms (ii) appropriate for smoothed delays for traces with low load (iii) very difficult or inappropriate for traces with high load.

2.4.7 Periodic Delay Patterns

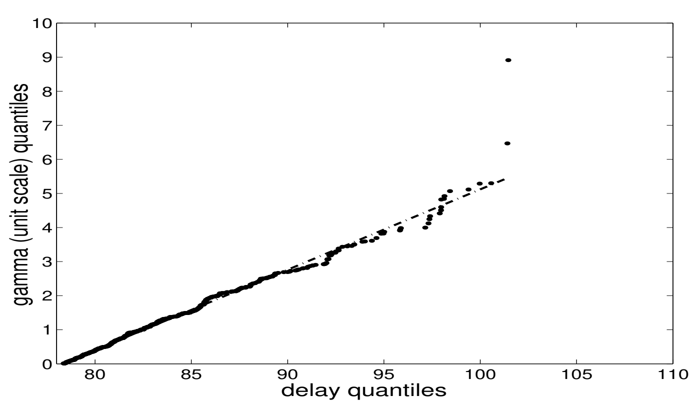
We have identified many delay patterns that exhibit some kind of structure. By "structure" we mean a specific shape of spike cluster that happens recurrently, sometimes even periodically. Two such patterns (namely the block, the very high delays) have been identified in high delay variability paths and discussed in Section 2.4.3. Another pattern (high spikes repeated every 10 minutes) has been identified in the lower variability paths and discussed in Section 2.4.4. Structure has also been identified in paths with mixed characteristics in Section 2.4.5, especially during the periods of high delay. The regularity in all these occurrences hints more toward the routers' operation or a network control protocol, rather than to multiplexing with cross-traffic. The most perfectly periodic pattern, was observed on



(a) 10 seconds of the example trace (from 70 to 80 sec)

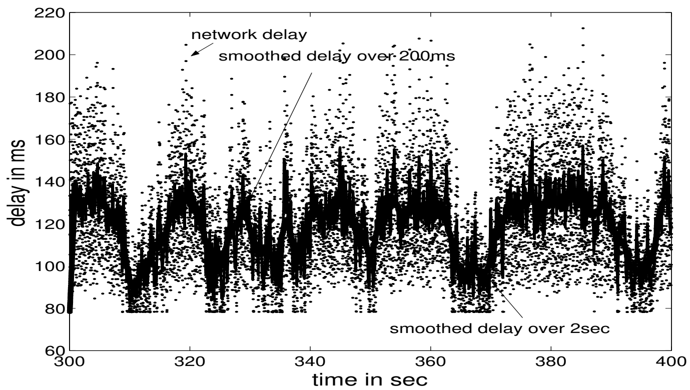


(b) Empirical PDF of instantaneous and smoothed delays

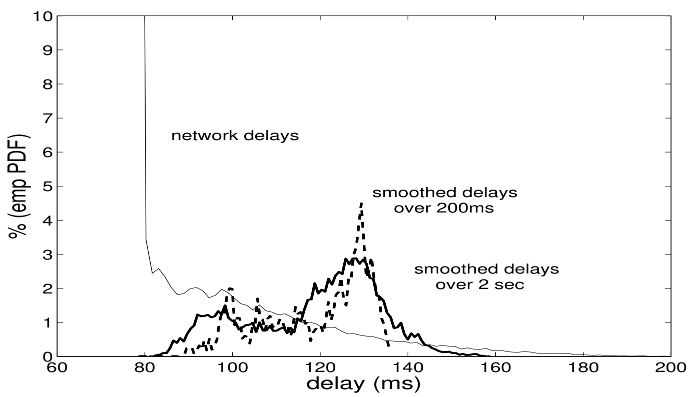


(c) QQplot test between the smoothed delay and gamma

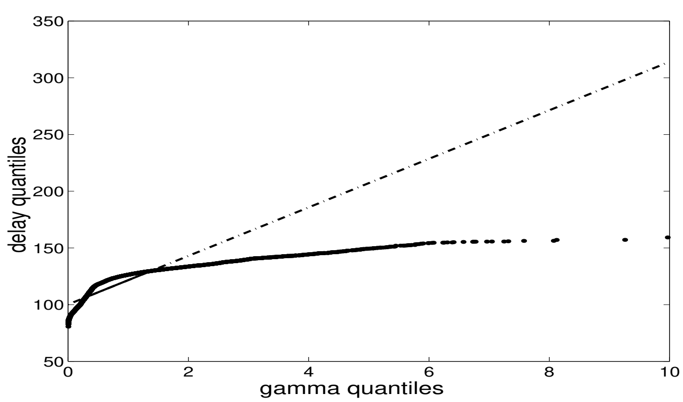
Figure 2.40: Example trace for which the Gamma distribution is a good fit



(a) 100sec from the example trace (from 300 to 400sec)



(b) Empirical PDF of instantaeous and smoothed delays



(c) QQplot between smoothed delay and gamma

Figure 2.41: Example trace for which the Gamma distribution is a bad fit

provider P_4. It was consistent across all six paths and the entire measurements period. Let us present this pattern.

Path EWR-P_4-SJC

The last example of delay percentiles shown in Figure 2.22(d), belongs to path EWR-P_4-SJC. That figure showed that the maximum, 99.9^{th} and 99^{th} have the same high value for the entire day. A closer look reveals that these percentiles are due to the following pattern.

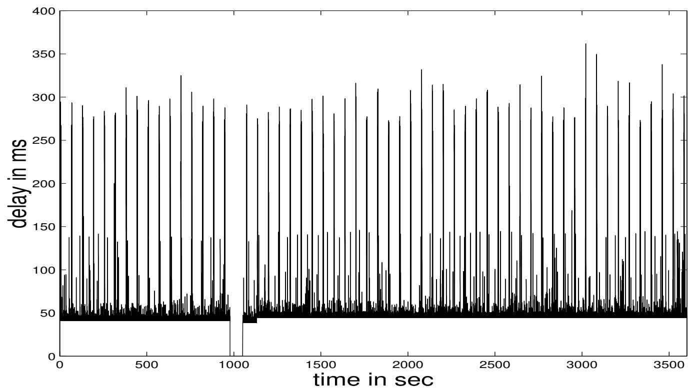
Figure 2.42(a) shows a typical hour on the path EWR-P4-SJC. Figure 2.42(b) shows in detail 200 sec and Figure 2.42(c) shows 7 sec. We can visually see that delay increases significantly every 60-70 sec. This increase is due to clusters of spikes 250-300 ms high, lasting for 3 seconds and repeated every 60-70 sec. In addition, there are some smaller spikes (100-150 ms high).

The periodicity in the high delay clusters is so perfect and so consistent (across six paths and 2.5 days) that cannot be explained as a result of multiplexing with other traffic. It is more possible to be due to network control traffic or some periodic process in the routers. In favor of this argument, are earlier studies that observed similar periodicities in delay, [80, 81]. In their case, the higher delays were caused by (i) a debugging option turned on in the gateways and (ii) by a synchronization in routing updates due to faulty Ethernet interfaces. There are many more control protocols and router operations that can potentially be responsible for this periodicity.

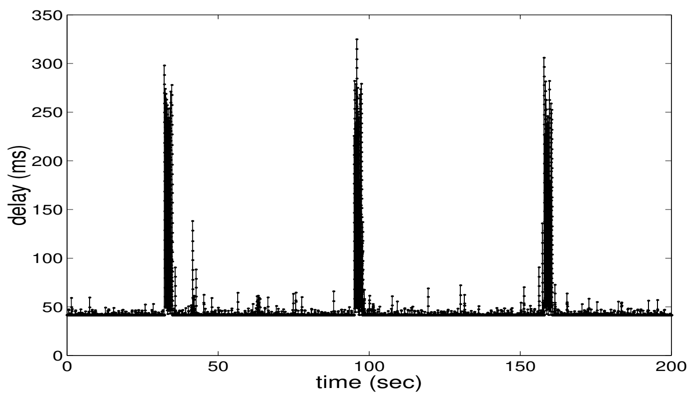
Our goal in this subsection is to provide a characterization that will allow the interested reader to reproduce the above delay pattern. As for provider P_1, it suffices to characterize the peak delays. The rest of the delays follow the triangular shape with slope -1.[6]

Similarly to what we did for provider P1, we first model higher delays (i.e. above 150 ms) and then we proceed with lower delays (above 100ms and above 50ms). The reason

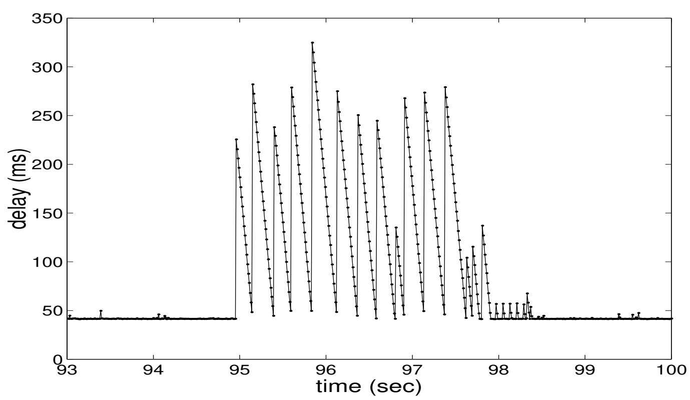
[6] In order to confirm this triangular shape, we plot in Figure 2.43(b) the height vs. the duration of all spikes above 50 ms. The fact that they lie close to the 45 degrees line, means that spikes last roughly as much as their height, without too much jitter. This is especially true for higher spikes, which we are particular interested in modeling.



(a) One hour: Wed, 21:00-22:00

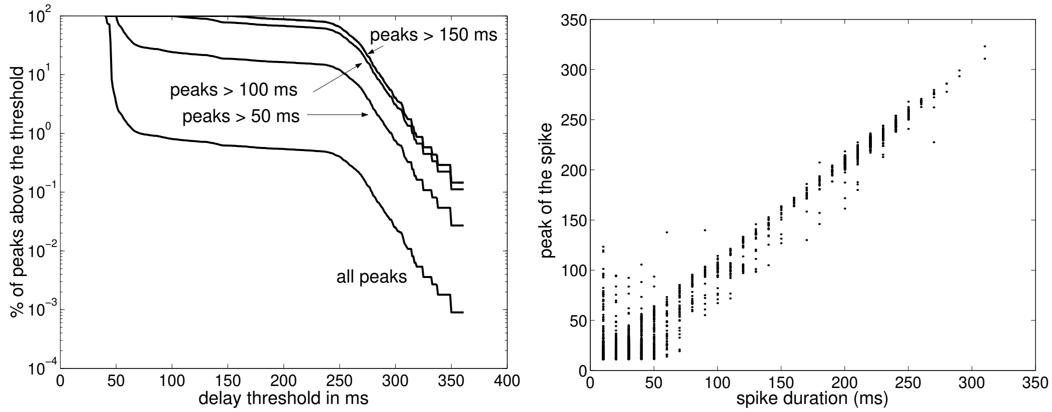


(b) Zooming in on 200 sec, starting at 21:10



(c) Zooming in on 7 sec

Figure 2.42: Perfectly periodic delay pattern on path EWR-P_4-SJC.

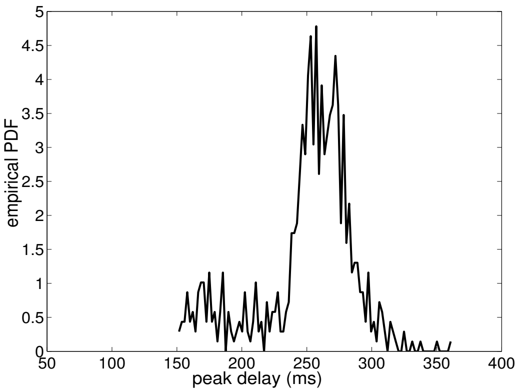


(a) CCDF of peak delays (b) Confirming the triangular spike shape

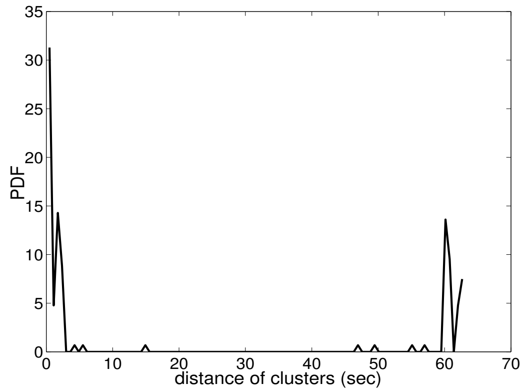
Figure 2.43: Path EWR-P4-SJC. Delay characterization for one hour (Wed 21:00 - 22:00).

for this separation is that the special patterns happen for higher delays which are a small percentage of the entire data set and therefore they would get diluted if the entire data set were examined. Furthermore, the interested user can choose one of these sets, depending on the application (e.g. playout above a certain value). Figure 2.43(a) shows the CCDF of all peak delays and of delays above 150ms, 100ms and 50 ms separately. We can see that no more than 1% of delays exceed 100ms; however these are the ones responsible for all the interesting patterns.

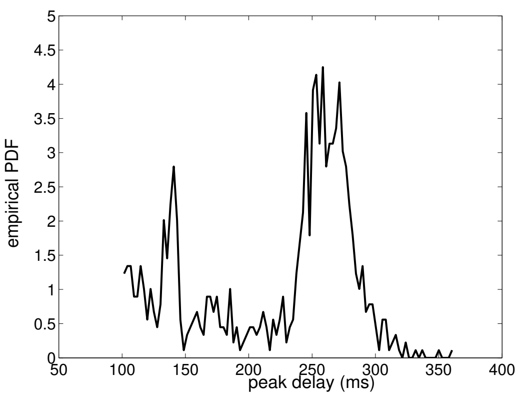
Figures 2.44(a) and (b) show that delays above 150 ms have heights around 250 ms and form clusters appearing every 60 seconds. The leftmost peak in Figure 2.44(b) corresponds to spikes in the same cluster; the rightmost peak corresponds to clusters 60 seconds apart. If we consider delays above 100ms, there is in addition a second set of spikes with peaks around 120 ms (the leftmost peak in Figure 2.44(c)), frequently enough to affect the distribution of distances (Figure 2.44(d)). If we consider delays above 50 ms, then the majority of spikes has a peak around 50 ms and they occur very frequently, as shown in Figures 2.44(e) and (f) respectively.



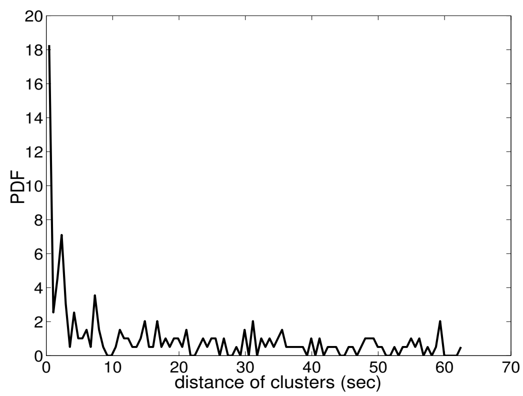
(a) Peak delays above 150 ms



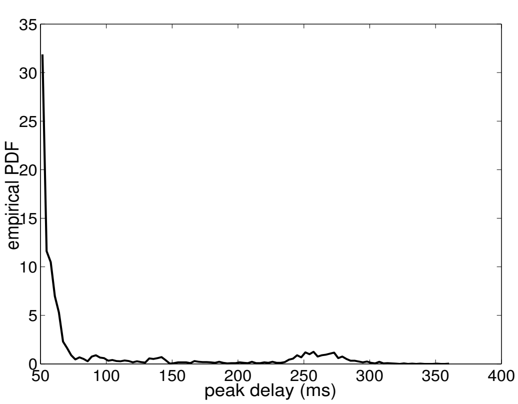
(b) Distances, for spikes above 150ms



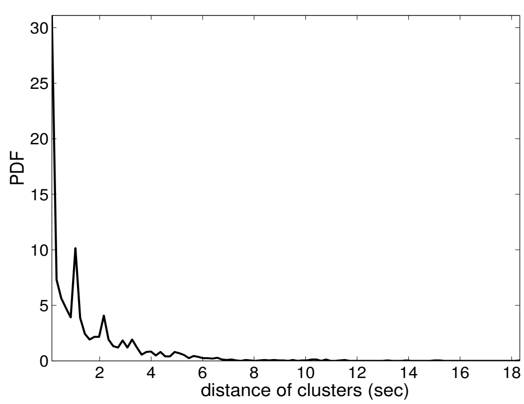
(c) Peak delays above 100 ms



(d) Distances, for spikes above100ms



(e) Peak delays above 50 ms



(f) Distances, for spikes above 50 ms

Figure 2.44: Characterization of three sets of spikes: above 150 ms, 100 ms and 50 ms.

2.5 Characteristics of Each Provider

In the above section, we described representative loss and delay patterns that occurred frequently in many different paths. However, only a limited combinations of loss and delay patterns happen on a specific path, depending, to a large extent, on the provider. We observed that paths of the same provider have similar delay and loss, patterns and ranges, whether they are short or long distance. This is intuitively expected as paths of the same provider may share network elements. For example, a failure on a shared link will affect all paths going through it; a shared link with increased load may become the bottleneck for many flows going through it and may affect their delay pattern. A periodic operation of a network protocol may affect all paths on the same backbone.

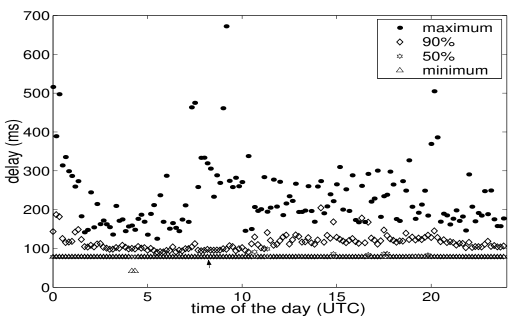
In this section, we go through each provider and mention the characteristics that are common in its paths, thus constituting the "signature" of the provider. Table 2.7, summarizes the most distinct loss and delay characteristics of each provider. Along the way, we also provide some additional information on each provider, for completeness.

Provider P_1

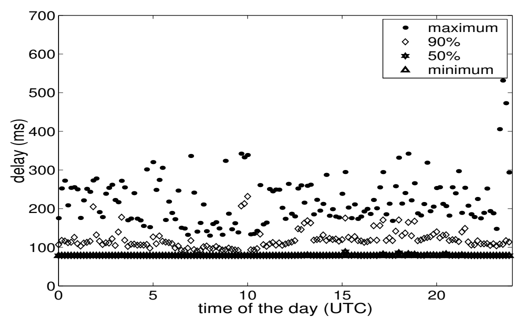
The larger amount of paths available (12) belong to provider. They can be divided in three groups with different characteristics.

One group has high delay variability and sporadic loss. An example path is THR-P_1-ASH, whose delay behavior has been studied in detail in Section 2.4.3. The loss on this path consists of medium size elementary durations, shown in Figure 2.5(a). The delay and loss percentiles for the entire 2 days period are shown in Figure 2.45.

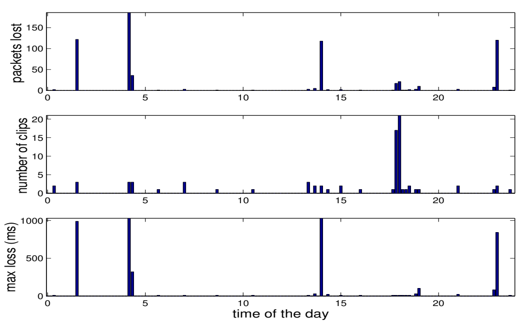
The second group has the largest delay and delay variability and many long outages, up to 50 seconds. A representative of this group is path EWR-P_1-ASH, whose delay percentiles are shown in Figure 2.46. We see that delay can be as high as 1.2 seconds, which is one of the two highest observed. The loss durations distribution and examples of outages have



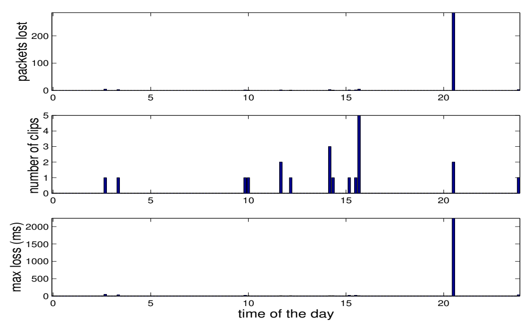
(a) Delay on Wed 06/27/01



(b) Delay on Thu 06/28/01



(c) Loss on Wed 06/27/01



(d) Loss on Thu 06/28/01

Figure 2.45: Path THR-P_1-ASH. Delay and loss statistics in 10 minute intervals.

Table 2.7: Consistent Characteristics of Paths per Provider

Provider	Number of paths	Distance	Observations on delay	Observations on loss
P_1	8	long	A group of paths has the highest delay and jitter	Only elementary loss (in the order of 1-2sec)
	4	long		
P_2	1	short	mixed: 2 states (high, low)	Mostly complex, synchronized on many paths of P_2 and P_6
	1	long		
P_3	2	short	Low-medium jitter	Single packets lost regularly on average every 5sec
	4	long		
P_4	2	short	Clusters of 250-350ms high spikes, periodic every 60 sec Frequent changes in minimum	Outages during changes in the minimum delay
	4	long		
P_5	2	short	Mixed	Small total number and clips. No outages. Synchronization of loss on different paths.
	4	long		
P_6	4	short	No jitter (2ms) except spikes every 10min.	Small total number. Mostly complex loss events, synchronized on different paths.
	5	long		
P_7	2	long	No jitter (2ms) except spikes every 10min	No loss, except 2 outages (with change in fixed delay)

been shown in Figures 2.5(c) and 2.4 respectively. There are 11 outages which are so long (up to 50 seconds) that result in the highest amount of loss experienced among all paths.

A third group has smaller delay variability and sporadic single packet loss. A representative path is THR-P_1-SJC, for which delay and loss percentiles are shown in Section 2.4.4 and Figure 2.5(c) respectively.

The remaining 9 paths of P1 fall in one of the above three categories. For example, Figure 2.5(b), (d) and (g) shows the loss durations distributions for the reverse of the three example paths, which have indeed similar characteristics. Path ASH-P1-THR has medium loss durations, ASH-P1-EWR has long outages and SJC-P1-THR has mostly small loss durations. The paths between EWR and SJC fall in the second category in terms of loss (many long outages). The paths between SJC and ASH, as well as between EWR and THR fall in the third category (low delay variability).

Table 2.8 summarizes all 12 paths. The most distinct characteristic of this provider is its

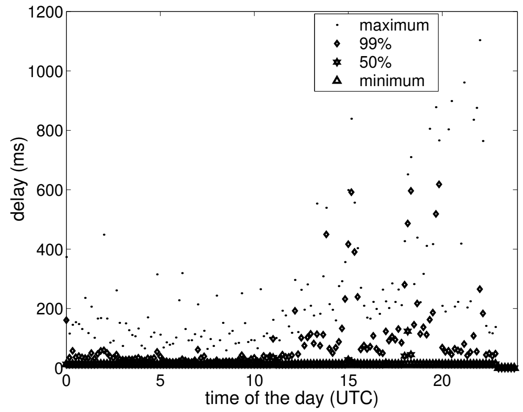


Figure 2.46: Delay percentiles in 10 minute intervals for path EWR-P_1-ASH, on Wed 06/27/01. Very high delay and delay variability

delay behavior. The first two groups of P_1 paths have the higher delay and delay variability than all other providers. Also the behavior of most paths varies during the day. Loss is sporadic: there are mostly elementary loss events that contribute to most of the total loss.

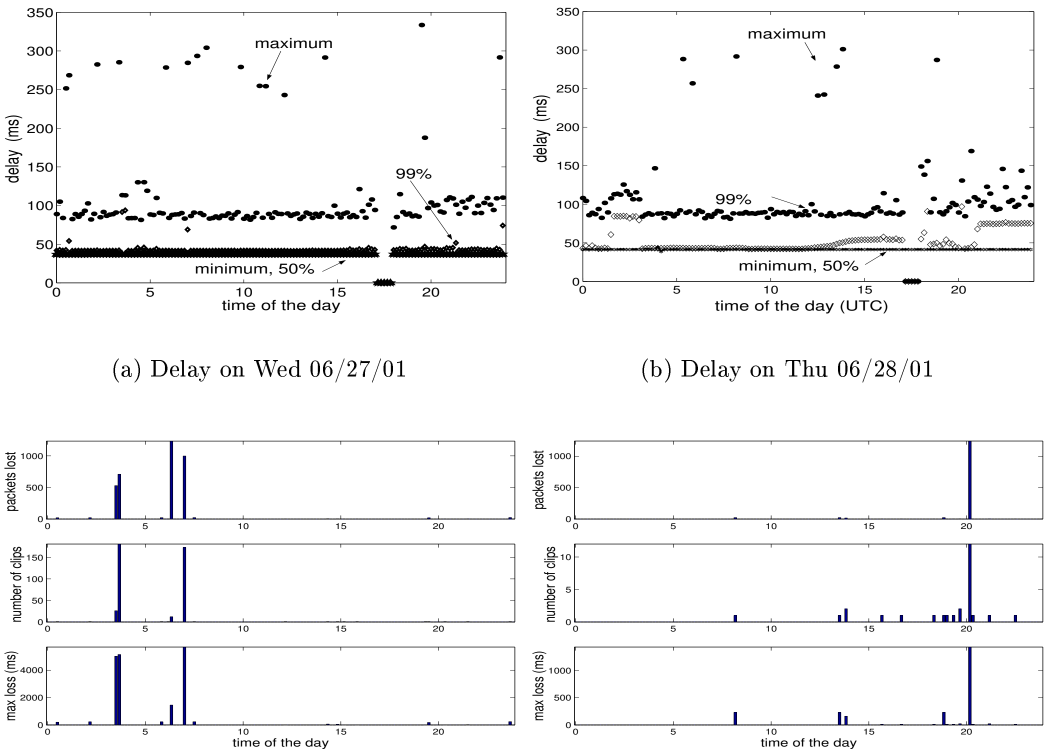
Provider P_2

There are only two paths that belong to provider P_2: one from EWR to SJC and one from EWR to ASH. These paths have two distinct characteristics. First, delay alternates between 2 possible states, as discussed in Section 2.4.5. Figure 2.47(a) and (b) shows delay percentiles for 2 days on EWR-P_2-SJC. Second, loss happens in complex loss events that occur during transitions between delay states, and often simultaneously on more than one paths, as discussed in great detail in Section 2.3.3.

Table 2.9 summarizes the 2 paths. Figure 2.47(c) and (d) show loss in 10 minute intervals for the entire 48 hours period. Figure 2.48 shows the distribution of loss durations for the two paths. It confirms (i) the concentration of loss duration around 19-25 consecutive packets and (ii) the fact that complex loss events contribute most to the total loss on these paths.

Table 2.8: Summary of paths on provider P_1

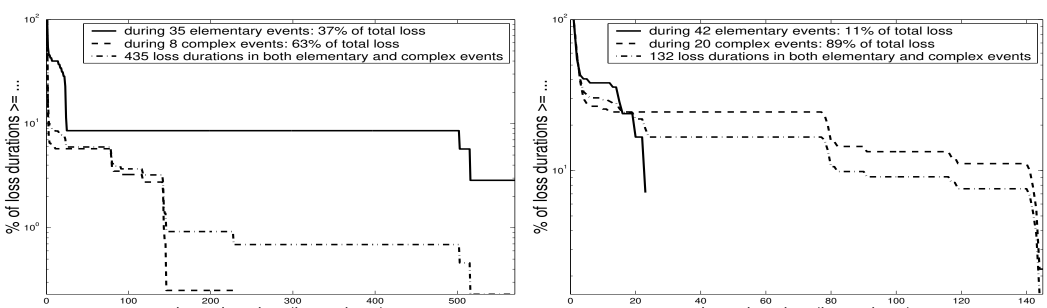
Path		Delay (in ms) (in 10 min intervals)			Loss events (in 10 minutes intervals)				out of order (packets) in a day
From	To	min	max best	max worst	day	time (UTC)	num. of clips	max clip (sec)	
THR	**ASH**	77.8	124.7	671.7	usually		~0	0.01-1	8
					Thu	20:30	2	2.24	
		41.3	change in fixed		Wed	4:00	3	0.32	
ASH	THR	68.5	110.9	397.6	usually		~0	0.1	9
							4	11.1	
EWR	**ASH**	11.9	58.8	1103.3	9 (Wed), 2 (Thu)		1-25	25-40	7
ASH	EWR		117.8	701.6	9 (Wed), 2 (Thu)		1-10	5-25	26
ASH	SJC	47	96.1	451.2	usually		~0	0.05-0.4	5
					Wed	4:00	5	1.67, 5.82	
SJC	ASH	55.8	105.8	802.5	usually		~0		
SJC	EWR	48.6	72.8	438.2	9 times on Wed		1-15	5-15	
EWR	SJC	45.5	90.6	340	9 times on Wed		1-10	10-35	8
THR	SJC	24.7	67.9	821.8	usually		~0	1	0
SJC	THR	24.5	28.1	924.8	usually		0-100	0.01-0.02	9
EWR	THR	26.9	34.1	296.2	always		1-10	0.2, 0.01	11
					Thu	14:10	1	7.53	
						14:20	2	5.89	
THR	EWR	28.4	54.4	1666	usually		~0	0	2
					Wed	4:00-4:10	8	111, 77.8	
					Wed	14:10	2	0.92	
					Thu	14:10	4	1.06	
					Thu	14:20	6	1.94	



(a) Delay on Wed 06/27/01 (b) Delay on Thu 06/28/01

(c) Loss on Wed 06/26/01 (d) Loss on Thu 06/28/01

Figure 2.47: Path EWR-P2-SJC. Delay and loss statistics in 10 minute intervals



(a) EWR-P2-SJC (b) EWR-P2-ASH

Figure 2.48: Provider P2. Distribution (CCDF) of loss durations

Table 2.9: Summary of paths on provider P_2

Path		Delay in ms			Loss events				out of order
		(in 10 min intervals)			(in 10 minutes intervals)				
From	To	min	min max	max max	day	time (UTC)	num. of clips	max clip (sec)	(in a day)
EWR	SJC	36.6	71.7	333.5	Wed	6:20	12	1.45	0
					Thu	20:10	12	1.43	
					Wed	3:30	26	5.02	
					Wed	3:40	151	5.15	
					Wed	7:00	143	0.57	
EWR	ASH	7.4	43.8	277.5	Wed	4:40	3	0.23	28
					Wed	6:20	13	1.45	
					Thu	20:10	12	1.43	
						usually		0.01-0.2	

Provider P_3

The distinct characteristic of provider P_3 is that it is the only one whose paths have regular single packet loss, at a rate 0.16-0.28%, as discussed in detail in Section 2.3.5. This loss behavior is consistent for four out of six paths. A possible explanation for the regular single loss, is that Random Early Drop (RED) is turned on, on the routers of this backbone.

The delay behavior of provider P_3 is not special compared to other providers. The fixed delay is low and there are small spikes. The delay pattern on trace EWR-P_3-SJC was shown in Figure 2.17. Delay percentiles across the entire day for this path are shown in Figure 2.34(b). The high maximum values every few hours correspond to rare high spikes. This delay behavior is consistent for all six paths.

Table 2.10 summarizes the 6 paths.

Provider P_4

Provider P_4 has two distinct characteristics, which are consistent across the entire measurement period and all six paths. First, delay follows the unique periodic pattern, studied in

Table 2.10: Summary of paths on provider P_3

Path		Delay (in ms) (in 10 min intervals)			Loss events (in 10 minutes intervals)				out of order (packets) in a day
From	To	min	min max	max max	day	time (UTC)	num. of clips	max clip (ms)	
EWR	SJC	31.2	61	300	usually		110-140	10-200	0
					4 times on Wed		1	200-250	
					Thu	10:20	104	19780	
SJC	EWR	32.2	118	421	all other intervals		0	0	10
					Thu	10:20	3	5990	
AND	EWR	7.6	22	393	usually		130-170	10-200	26
EWR	AND	6.9	17	304	usually		100-140	10	48
AND	SJC	35.5	65	338	usually		130-190	10-200	14
SJC	AND	35.7	122	334	always		0	0	41

detail in Section 2.4.7 about periodic delay behavior. Second, there are frequent (more frequent than in any other provider) changes in the fixed delay accompanied by long outages, studied in detail in Section 2.3.2 about outages. Table 2.11 lists all 10-minutes intervals, during which the fixed part of delay changed and/or an outage happened. It shows the time and day of occurrence, the value of the fixed delay and the number and duration of the elementary loss events.

Provider P_5

There are 6 paths belonging to provider P5, summarized in Table 2.12 . They all have mixed delay variability as shown in Section 2.4.5 and a good behavior in terms of loss.

Loss is not a problem on this path. As we can see in Table 2.1, the number of packets lost is small, among the smallest of all paths. Unlike other providers, there are no long outages, and the longest loss duration is 1-2.7 sec in all six paths. Two paths have very little loss. The first one, SJC-ASH, had only one elementary loss duration 1.14 sec long and a complex event consisting of clips 47, 250 and 83 packets each separated by a single received packet. The second path, ASH-SJC, had only 5 isolated (elementary) loss durations 139, 107, 7, 1

Table 2.11: Summary of paths on provider P_4

Path		Delay in ms (in 10 min intervals)			Loss events (in 10 minutes intervals)				Out of order packets (in a day)
From	To	min	min max	max max	day	time (UTC)	num. of clips	max clip (sec)	
ASH	EWR	6.8	262.1	628.1	Tue	21:00	1	23.94	>100
					Wed	6:20	12	1.57	(Thu@9)
					Thu	7:50	363	0.03	
					Thu	9:30	1	118	
					Thu	20:10	1	15.25	
EWR	ASH	7.8	270.9	618.4	Tue	21:00	1	16.07	5
		8.5			Wed	21:00	1	11.63	
		7.2			Thu	8:00	157	27.48	
ASH	SJC	41.7	292.8	609.8	Tue	21:00	3	33.7	11
		49.8			Wed	21:10	1	14.9	
		36.4			Thu	7:50	1	15.82	
		41.7			Thu	8:00	6	1	
SJC	ASH	45.2	309.9	621.7	Tue	21:00	1	30	12
		45.2			Wed	6:20	12	1.63	
		35.1			Thu	7:20	10	0.06	
		46.8			Thu	20:00	52	1.1	
SJC	EWR	47.8	300.3	630.3	Tue	21:00	0	0	0
		52.6			Tue	21:10	0	0	
		48.8			Tue	21:20	0	0	
		48.8			Wed	6:20	12	1.57	
		52			Wed	18:00	0	0	
		48.8			Thu	8:00	1	7.23	
		46.8			Thu	8:10	1	12.46	
		48.8			Thu	8:20	0	0	
		46.3			Thu	9:30	1	27.79	
		44.6			Thu	14:20	0	0	
EWR	SJC	42.3	289.8	609	usually			0.01-0.2	17
		40.7			Tue	21:00	0	0	
		38.3			Wed	21:10	0	0	
		44.1			Tue	21:20	0	0	
		49			Thu	9:30	0	0	
		49			Thu	9:40	3136	0.01	
		42.3			Thu	10:30	0	0	
		42.3			Thu	20:20	11	1.35	
		42.3			Thu	20:30	5938	49.15	

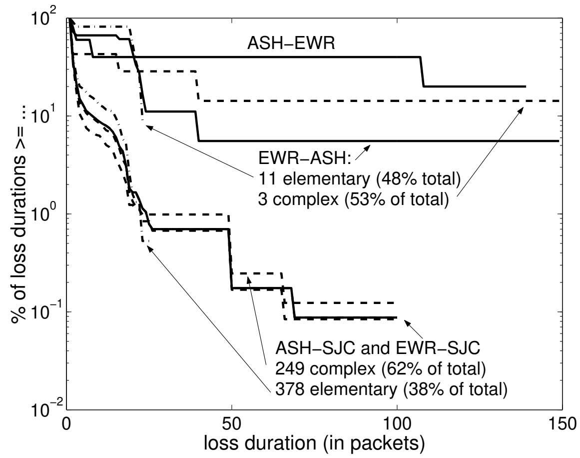


Figure 2.49: Complementary Cumulative Distribution Function (CCDF) of the loss durations on the four (lossier) paths of provider P_5

and 2 packets each. Figure 2.49 shows the distribution (CCDF) of the loss durations for the remaining four paths. It confirms that (i) loss durations are small (ii) there is again some concentration of loss durations around 19-25 packets (iii) paths ASH-SJC and EWR-SJC have almost identical distributions (we have already seen the identical loss behavior over 48 hours in Figure 2.15).

Provider P_6

There are 9 paths pertaining to provider P6, summarized in Table 2.13. They all have the same low delay variability as the example path shown in Section 2.4.4. Many of them also incur frequent changes in the minimum delay, as in Figure 2.21.

Figure 2.50(a) shows the distribution of loss durations for the entire 48 hours period on path ASH-P6-SJC. The total number of packets lost (277) is among the lowest of all paths. There are 142 loss durations, 125 of which are single packets lost and the longest of which is only 23 consecutive packets. There are 3 complex loss events that have been discussed in detail in Section 2.3.3.

Figure 2.50(b) shows the distribution of loss durations for all 9 paths of provider P_6.

Table 2.12: Summary of paths on provider P_5

Path		Delay (in ms)			Loss events				out of
		(in 10 min intervals)			(in 10 minutes intervals)				order
From	To	min	max	max	day	time	num. of	max	(packets)
			best	worst		(UTC)	clips	clip (sec)	in a day
SJC	ASH	32.8	51.9	275.1	usually		0	0	11
					Thu	2:40	4	2.5	
ASH	SJC	32.9	62.3	303.3	usually		<342	0.15	3
					Tue	21:50	50	0.03	
EWR	SJC	32	47.8	420.7	usually		10-100	0.01	
					Tue	21:50	50	0.03	
SJC	EWR	32.7	74.8	295.1	usually		0	0	0
					Thu	2:40	2	1.39	
ASH	EWR	6.9	21.8	396.9	usually			0.2	4
					Thu	2:40	7	1.49	
EWR	ASH	6.2	44.4	247.4	usually			0.2	20
					Thu	2:40	5	2.7	

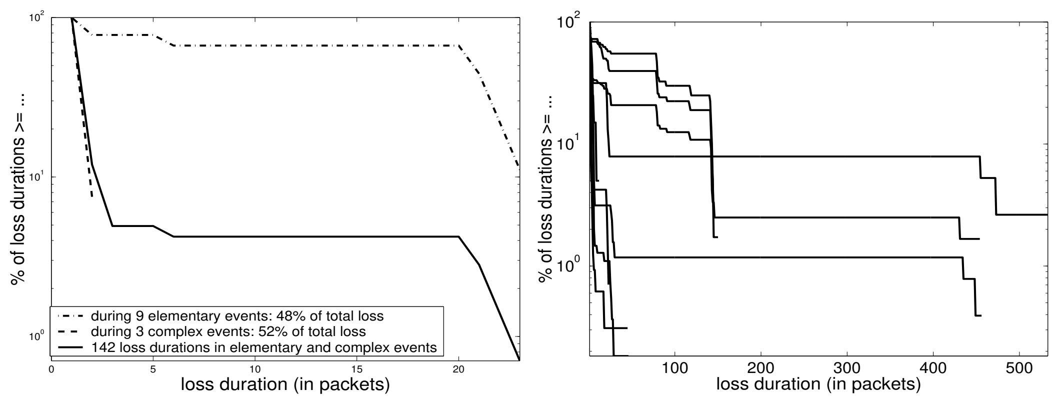
We can see that there are longer loss durations (the longest being 5.33 sec and many of them being around 1.4sec). A closer look reveals that these 1.4 sec durations happen also during the complex loss events on Wed 6:20 and on Thu 20:10 . Also notice once again, the frequency of loss durations around 20 packets.

Provider P_7

There are only two paths pertaining to this last provider, connecting SJC and ASH. They have the exact same behavior. They have very low delay variability, studied in Section 2.4.4. Jitter is the second lowest after P_6: the 99^{th} jitter percentile of all 10 minutes intervals lies in the range [0.59ms, 0.715 ms] and [0.589ms, 0.751 ms] for the two paths. The highest delay values are due to spikes every 10 minutes. The only loss on these paths is the outage happening simultaneously on both paths, accompanying a change in the fixed part of the delay, repeated at the same time both days. These outages have also been discussed in Section 2.3.2 about outages. Table 2.14 gives the time and day, the value of the minimum delay and the loss duration during those events.

Table 2.13: Summary of paths on provider P_6

Path		Delay (in ms)			Loss events				out of
		(in 10 min intervals)			(in 10 minutes intervals)				order
From	To	min	min	max	day	time	num. of	max	(packets)
			max	max		(UTC)	clips	clip (sec)	in a day
SJC	ASH	37.7	55.9	363.8	usually		0	0	11
ASH	SJC	38.4	64.7	361.8	usually		0	0	7
		39.1			Wed	3:20	140	0.02	
		change in the fixed part			very often		no loss		
EWR	SJC	32.3	56.8	297.1	usually		0	0.02	0
					Wed	3:20	140	0.02	
					Wed	6:20	12	1.44	
AND	SJC	32.3	62.4	337.5	usually		5-10	0.1	0
SJC	AND	37.8	40.8	288.2	usually		0	0.03-0.45	5
ASH	AND	6.7	9.8	247.1	usually		0	0.2	0
		6.9			Tue	20:10	6	5.45	
		7			Wed	3:10	13	5.3	
		7.1			Wed	3:20	54	4.34	
					Wed	4:50	7	4.72	
AND	ASH	6.7	27.6	330.4	usually		0	0.1	0
		6.9			Tue	20:10	6	5.31	
		7			Wed	3:10	10	4.56	
		7.1			Wed	3:20	54	4.34	
EWR	ASH	3.48	22.2	325.9	usually		0	0.01-5	0
					Wed	6:20	12	1.44	
EWR	AND	3.25	9.9	275.7	usually		0	0.25	0
					Wed	6:20	12	1.45	
					Thu	20:10	12	1.42	



(a) Path ASH-P6-SJC

(b) All nine paths of P6

Figure 2.50: Complementary Cumulative Distribution Function (CCDF) of the loss durations on provider P_6.

Table 2.14: Summary of paths on provider P_7

Path		Delay (in ms)				Loss events				out of order (packets) in a day
		(in 10 min intervals)				(in 10 minutes intervals)				
From	To	min	99%	max best	max worst	day	time (UTC)	num. of clips	max clip (sec)	
SJC	ASH	40.5	41.5	58.9	289.8	usually		0	0	36
		40.7				Wed	3:10			
		37.8				Wed	4:00	1	166.18	
		40.5				Wed	4:30			
		40.5				Thu	3:20-			
		37.8				Thu	3:20	1	12.639	
		40.5				Thu	3:50			
ASH	SJC	40.8	41.6	64.9	342.3	usually		0	0.2	5
		38.1				Wed	4:00	1	111.9	
		40.8				Wed	4:30			
		38.1				Thu	3:20	2	0.25	
		40.8				Thu	3:50			

Table 2.15: Summary and discussion of the events observed in the traces

Impairment	Event observed in the traces	Possible cause	Perceived effect on VoIP	Possible Remedy
Loss	short loss duration	drop in the buffer (overflow or early drop)	clipped speech	concealment
	~20 packets lost	buffer drop due to ?	clipped speech	-
	loss clusters	reconfiguration, link failures	loss of connectivity	improve network reliability
	Outages			
Delay	high e2 delay	routing, jitter, other delay components	bad interactivity	live with it
			amplified echo	cancel echo
Delay Jitter	high spikes	routers operation (debug options, “vacation“)	{gap, clip or pitch change} or {additional e2e delay}	{fix the patterns in the network} or {playout scheduling}
	periodic spike patterns	control traffic (e.g. routing protocols)		

2.6 Summary of Events and Possible Causes

In this chapter, we studied the loss and delay characteristics of 43 backbone paths. Among them, there were paths with low delay and low jitter, as well as negligible or no loss at all. The paths with these good characteristics belong to over-provisioned backbone networks, with practically constant delay and without loss. However, there are paths that exhibit worse loss and delay behavior. Even on the good paths, there are occasional loss and delay events, which may introduce impairments to multimedia or even to TCP-based traffic. In this section, we revisit the “bad events” and try to infer their causes by observing their patterns and comparing them to similar findings in the literature. Furthermore, we give a preview of how these events are perceived by VoIP users and how they can be handled by the network or the application. Table 2.15 summarizes the events and their causes. Columns describing the perceived effect of the impairments and their possible remedies are also given for completeness.

First, in terms of loss we observed the following. There are many *short loss durations*that happen sporadically, as we can see in Table 2.1 and in the distribution of loss durations for every provider. For provider P_3, single packet loss is regular every 5 seconds. This might be due to Random Early Drop (RED) turned on in the routers. The regularity of loss may have an effect on TCP but not on stream traffic, as short loss durations up to 60-90 ms can be adequately concealed at the receiver. In addition, we noticed that providers P_2, P_3, P_5, P_6 experienced frequently *19-25 consecutive packets lost*, usually following high spikes. We have no good explanation for these events. 200 ms of speech lost cannot be concealed without contextual information, as this is a duration longer than the phonemes. There were also *outages* (i.e. periods in the order of tens of seconds or 1-2 minutes), during which no packets at all were received. We attribute these outages, which accompany changes in the minimum delay, to routing changes and the time required by routing protocols to converge. For the reasons behind the rest of the outages, we speculate link failures or maintenance. Evidence for these explanations are the facts that (i) some outages happen at the same time of the day (that could be a maintenance process) and (ii) many outages affect more than one paths (implying failure of a shared link). Recent work in SprintLabs, [7], showed that the main problem in their backbone today, is link failures followed by periods of routing instability, during which packets are forwarded to invalid paths and eventually dropped. Finally, we observed *loss clusters* (or "complex loss events") which were periods of tens of seconds with loss rates 10-80%, which are too high to be concealed. Many of these clusters happened simultaneously on multiple paths and had the exact same loss pattern, hinting again to a failure or congestion of a shared link. A recent study by AT&T, [105], also looked at loss grouped in events with high loss rates, during which loss durations were i.i.d. distributed.

In terms of end-to-end (e2e) delay, we saw that the fixed delay can be as high as 78ms in these backbones, which implies that routing does not always choose the shortest path (e.g. from THR to ASH). On top of the fixed delay, there are additional components that contribute to the total e2e delay, such as the algorithmic and packetization delay, queuing and transmission delay on regional networks and playout delay due to jitter. There is not much we can do about high e2e delay, apart from keeping the contributing components as low as possible. Even if the overall e2e delay is high, people are still able to carry out a conversation by adjusting their conversation speed.

In terms of delay variability, we observed unusually *high spikes*, up to 500 ms or even 1 second. Spikes of smaller size can be due to multiplexing with cross traffic; this was the modeling assumption in a pioneer paper in this area by Bolot, [4]. However, the size of these spikes and the lack of slow varying component in the delay traces, hints more toward an explanation based on "server vacation theory". Indeed, routers often take breaks from serving packets in the queue, during which they perform other internal tasks. This was recently observed by Papagiannaki et.al. in [72]; however the height of those spikes was much smaller than the ones we observed. Earlier experimental work by Sanghi et.al. back in 1993, [80], observed 600ms high spikes every 90 seconds, caused by a debugging option turned on in the gateways. They also identified other periodic patterns, [81], caused by synchronized routing updates due to faulty Ethernet interfaces. Indeed, the *perfect periodicity* of some patterns on providers P_1 (see Section 2.5) and of all traces for the entire measurement period on P_4 (see Section 2.5) cannot be explained by multiplexing with regular traffic. It worths mentioning that in [105], 60-second periodicities (in terms of loss) were identified which were discarded as anomalies in the data set. The periodicity and the height of the spikes are more likely to be explained by network control traffic (such as exchanges of messages by routing protocols) or router specific operations (e.g. debugging options).

Overall, it seems that problems in backbone networks, tend to be caused by the network operation (network protocols, control traffic, maintenance, route changes) and by the routers internal operation (e.g. "vacations", debugging options) rather than by the traffic load and the (lack of) quality-of-service mechanisms. Understanding which protocols result in these patterns as well as understanding the router behavior would help to solve these problems by eliminating the patterns. This is a useful direction to explore for people that have privileged access to routers and network operation. As long as the delay patterns remain, there is value in mechanisms (such playout scheduling) at the receiving end that minimize the effect of these patterns on the application.

Chapter 3

Voice Communications over Internet Backbones

In this chapter, we study the performance of voice communications over Internet backbone networks. Although the loss and delay measurements of Chapter 2 give us a rough idea about this performance, they do not directly translate to voice quality, as perceived by the user.

In Section 3.1, we describe the VoIP system under study and the various aspects of voice perceived quality. We collect results from several studies that have assessed the effect of individual impairments and we combine them to develop a methodology for assessing an entire phonecall. In Section 3.2, we study an important component of the VoIP system that strongly affects the overall perceived performance, namely the playout scheduling. We study existing algorithms and we propose and evaluate new ones, based on our measurement study of Internet backbone networks and on our understanding of voice quality. In Section 3.3, we simulate phonecalls happening for periods of time on representative backbone paths, and we provide statistics for their quality. The performance strongly depends on the path (and in particular on the provider) used, as well as on the playout algorithm applied. In Section 3.4, we confirm that, when multiple paths are available, path diversity can be exploited to provide a better delay and loss behavior to the end-user.

As discussed in Section 2.6, the problems observed in these backbone paths (i.e. the large and regular spikes and the high loss periods) and their possible causes (i.e. control traffic and router operation) indicate that the permanent solution is to better understand these events and to prevent them from occurring. Until the problems are fixed in the network, application-level mechanisms such as playout scheduling and multipath streaming, are needed to provide acceptable VoIP quality.

3.1 VoIP System and Quality Aspects

VoIP refers to voice communication over IP data networks. In this Chapter, we first describe the various components of a VoIP system. We then discuss the impairments introduced by the network and their perceived effect in the quality of voice communications. Extensive work has been conducted on assessing the degradation caused to individual impairments. We review and combine results from different studies, in order to develop a methodology for assessing the VoIP quality over Internet traces.

3.1.1 Components of a VoIP System

Speech is an analog signal that varies slowly in time, with bandwidth not exceeding 4KHz. Furthermore, speech alternates between talkspurts and silence periods, that have been reported to follow approximately exponential distributions with mean 1.2 and 1.8 sec respectively, [9]. For the purpose of transmission over networks, the speech analog signal is converted into a digital signal at the sender; the reverse process is performed at the receiver. In an interactive conversation, the participating parties switch turns in taking the sender and receiver roles.

There are many *encoding schemes* that have been developed and standardized by the ITU. The simplest is the sample-based G.711 which uses Pulse Code Modulation (PCM) and

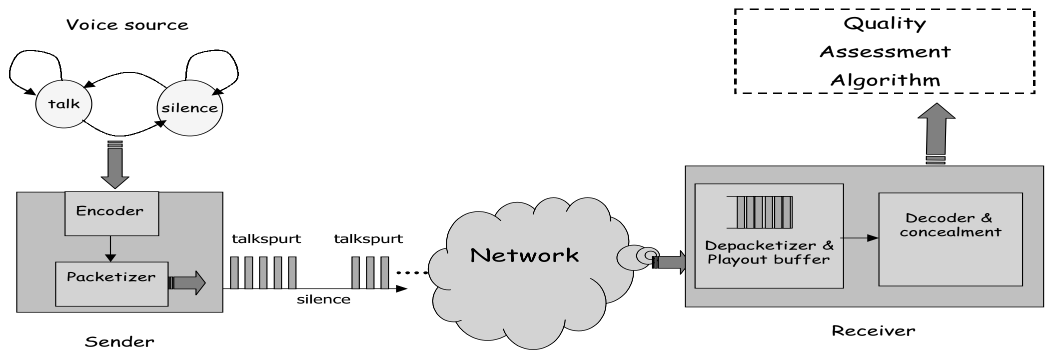


Figure 3.1: VoIP System

produces a digitized signal of 64 Kbps. CELP-based encoders provide rate reduction (i.e. 8 Kbps for G.729, 5.3 and 6.4 Kbps for G.723.1) at the expense of lower quality and additional complexity and encoding delay, as discussed in [20]. Further reduction in the data rate can be achieved if no signal is encoded during the silence periods, a technique known as Voice Activity Detection (*VAD*). VAD tends to elongate talkspurts by the minimum continuous silence period (known as hangover time) required to decide the end of a talkspurt.

The encoded speech is then *packetized* into packets of equal size. Each such packet includes the headers at the various protocol layers (namely, the RTP (12B), UDP (8B) and IP (20B) header as well as Data Link Layer headers) and the payload comprising the encoded speech for a certain duration.

As the voice packets are sent over an *IP network*, they incur variable delay and possibly loss. In order to provide a smooth playout at the receiver despite the variability in delay, a *playout buffer* is used. Packets are held for a later playout time in order to ensure that there are enough packets buffered to be played out continuously. Any packet arriving after its scheduled playout time is discarded. There are two types of playout algorithms: fixed and adaptive. Playout scheduling is discussed in detail in its own separate Section, and thus it is omitted here.

The content of the received voice packets is delivered to the *decode*r which reconstructs the speech signal. Decoders may implement *packet loss concealment (PLC)* methods that produce replacement for lost data packets. Simple PLC schemes simply insert silence, noise or a previously received packet. More sophisticated schemes attempt to find a suitable replacement based on the characteristics of the speech signal in the neighborhood of the lost packet(s). They may be interpolation-based (and try to match the waveform surrounding the lost portion) or regeneration-based (by being aware of the structure of the codec and exploiting the state of the decoder). A good survey of packet-loss recovery techniques can be found in [73].

Although not evaluated in our study, it is worth mentioning that audio tools, such as RAT developed in UCL [96], may include additional error resiliency mechanisms, such as transmission of layered or redundant audio, interleaving frames in packetization, retransmissions (if the end-to-end delay budget permits it), feedback to signal the sender to switch rate or encoder.

3.1.2 VoIP Impairments

The quality of voice communication is affected by a number of impairments that have to do either with speech quality or with interactivity and other delay related aspects.

Speech quality is affected by low bitrate compression, even before any transmission over the network. In addition, the transmission of packet voice over a network is subjected to packet loss in network elements causing degradation in the quality of voice at the receiver. Further loss is incurred in the playout buffer at the receiver, caused by large delay jitter in the network. The perceived degradation due to packet loss can be mitigated by means of packet loss concealment at the receiver.

Another aspect of voice communications is the interactivity between the communicating parties, which is affected by the delays incurred in the course of transmission over the network. Indeed, a large delay may lead to "collisions", whereby participants talk at the same time. In order to avoid such collisions, the participants can take turns as if the connection were half duplex. In that case, it will take longer time to complete their conversation. To achieve a good level of interactivity, the end-to-end delay (from mouth-to-ear or "m2e") should be maintained below a certain maximum delay, typically on the order of 100-150 ms. Longer delays become noticeable; the longer the end-to-end delay is, the lower is the degree of interactivity. The end-to-end delay encompasses: (i) the delay incurred in encoding (referred to as algorithmic delay), (ii) the delay incurred in packetization (function of the amount of speech data included in a packet), (iii) the delay incurred in the path from the sender to the receiver (propagation time, transmission time over network links, and queuing delays in network elements), (iv) the delay incurred in the playout buffer, and finally (v) the delay incurred in the decoder (usually negligible).

Finally, the presence of echo in various situations could represent a major source of quality degradation in voice communication. An informative discussion on echo in VoIP systems, as well as on other VoIP related issues, can be found in [59]. One cause of echo is the reflection of signals at the four-to-two wire hybrids; this type of echo is present when a voice call involves a combination of a VoIP segment in the Internet and a circuit segment in the switched telephone network. Another cause of echo is in PC-based phones (which are typically equipped with a microphone and loud-speakers), whereby the microphone at the remote end picks up the voice played on the loud-speakers and echoes it back to the speaker. Voice echo is not perceptible if the end-to-end delay is very short (below 10 ms.) For larger end-to-end delays, echo becomes perceptible, and the longer the delay is, the more annoying the effect of echo becomes. The effect of echo is mitigated by means of echo cancellation placed close to the cause of echo.

Sometimes users are willing to tolerate some of the above impairments, in exchange for the ease of access (e.g. cellular phones) or for a lower price. However, if the Internet is

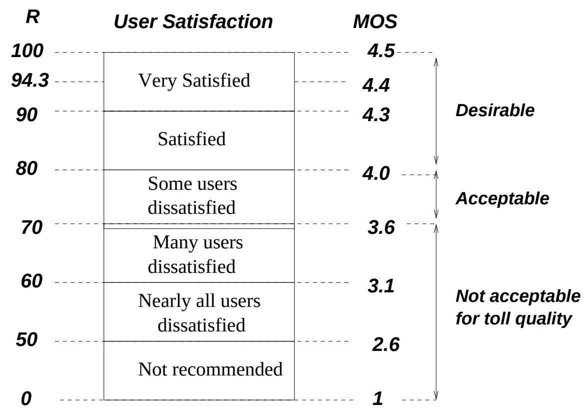


Figure 3.2: The relation between Mean Opinion Score and user satisfaction is given in the "Definition of categories of speech transmission quality" in ITU-T G.109. The correspondence between MOS and the Emodel rating R is given in ITU-T G.107/Annex B.

to eventually replace the telephone network, it should stand up to the same high quality standards and thus eliminate the above impairments.

3.1.2.1 Subjective Quality

In order to assess the quality of voice communication in the presence of impairments, it is necessary to study the individual as well as collective effects of the impairments, and produce quantitative measures that reflect the subjective rating that listeners would give. This subjective quality measure is also referred to as Mean Opinion Score (MOS) and is given on a scale of 1 to 5. A MOS rating above 4.0 matches the level of quality available in the current Public Switched Telephone Network; a rating above 4.3 corresponds to the best quality whereby users are very satisfied; and a rating between 4.0 and 4.3 corresponds to a high quality level, whereby users are satisfied. A MOS rating between 3.6 and 4.0 corresponds to a medium quality level whereby some users are dissatisfied. A MOS rating in the range between 3.1 and 3.6 corresponds to a low level of quality whereby many users are dissatisfied. A MOS rating in the range between 2.6 and 3.1, the level of quality is poor whereby nearly all users are dissatisfied. And finally, a MOS below 2.6 is not recommended. (See Figure 3.2.)

Table 3.1: Standard encoders and their characteristics

Standard	Codec type	Rate (Kbps)	Frame (ms)	Lookahead (ms)	MOS intr.
G.711	PCM	64		0	4.43
G.729	CS-ACELP	8	10	5	4.18
G.723.1	ACELP	5.3	30	7.5	3.83
G.723.1	MP-MLQ	6.3	30	7.5	4.00

Numerous studies have been conducted to assess the effect on voice quality of various impairments under various conditions. Some of them have also been compiled into reports and recommendations published by standards organizations. In the remainder of this Chapter (i) we review and summarize the results obtained therein in order to complete the evaluation space as well as to confirm their consistency and (iii) we identify an approach for assessing the quality of a phonecall.

3.1.2.2 Degradation in Speech Quality due to Compression

The degradation in speech quality due to the encoder has been characterized and is summarized in Table 3.1. The quality after compression, without considering the effect of packet loss, is often referred to as intrinsic quality MOS_{intr}. As can be seen from the table, lower rate encoders result in lower MOS values.

3.1.2.3 Degradation in Speech Quality due to Loss

We now address the effect of packet loss that results in speech clipping, focusing on G.711, G.729 and G.723. Among the earliest work in this area is that by Gruber and Strawczynski back in 1985, [29]. They addressed the effects of speech clipping and variable speech burst delays incurred in dynamically managed voice systems utilizing speech activity detection. Subjective evaluations were conducted with PCM encoded speech for various simulated conditions pertaining to speech clipping and delay variations. Of interest and relevance to this study are the results pertaining to speech clipping (i.e., loss) whereby speech clips of

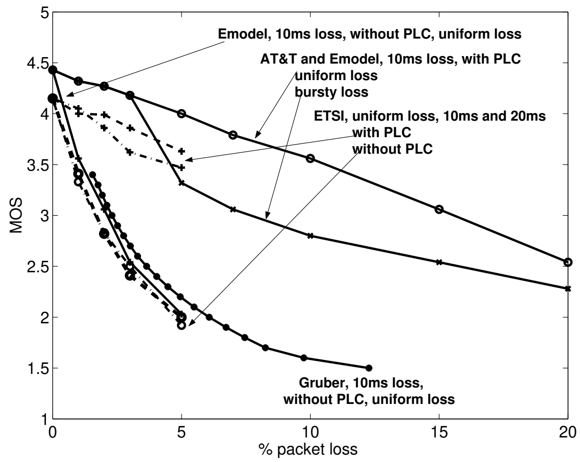


Figure 3.3: G.711 quality under various packet loss conditions. The packet size is 10ms in all cases.

certain fixed durations (ranging from 4 ms to 256 ms) are uniformly distributed across time, with overall loss rates ranging from 0 to 20%. Results for 10ms loss duration are plotted in Figure 3.3.

We note that these tests did not involve loss concealment, as such techniques did not exist for PCM at that time. Later on (in particular in the AT&T contribution T1A1.7/99-012 to committee T1), extensions to G.711 have been proposed to complement PCM with frame erasure concealment and make it appropriate for packet transmission systems. On the other hand, G.729 [42] has been developed for data and cellular networks, and an algorithm has been specified for concealing the loss of a frame. When this happens, all the CELP parameters for the lost frame are interpolated from parameters from the previous frame.

The benefit of error concealment has been studied for G.711 under both uniform and bursty loss conditions, considering packets containing 10 ms of speech, and packet loss rates ranging from 0 to 20%. The model used for the bursty loss was a 2-state model with 100ms maximum allowed loss duration. Results pertaining to G.728 and G.729 with their standard packet loss concealment but for loss rates ranging up to only 5% can also be found in [21]. The results obtained for G.711 and G.729 are reproduced in Figures 3.3 and 3.4 respectively,

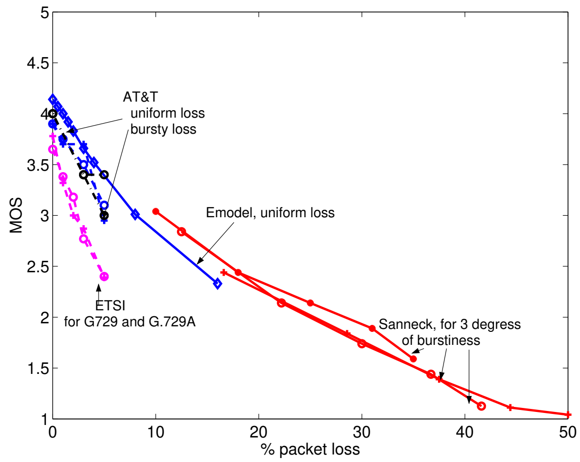


Figure 3.4: G.729 quality under packet loss conditions, as reported by various studies. The packet size is $20ms$ and packet loss concealment was implemented, in all cases.

under the label "AT&T and Emodel". The study by Perkins et al., reported upon in [74], characterized the subjective performance of G.729 in wireless and wired networks under various conditions, including channel bit errors, environmental noise, and frame erasures (up to 3% loss rate). It is of interest to note that the results in [74] pertaining to the effect of frame loss in G.729 agree with those published in [21].

Sanneck et. al. [83] also assessed the effect of loss on G.711 and G.729, using a Gilbert model to simulate bursty loss conditions, and using the unconditional (ulp) and conditional (clp) loss probabilities as the two free parameters of the model. He considered a wide range of loss rates ($ulp \leq 50\%$) and degrees of burstiness ($0 \leq clp \leq 50\%$). The evaluation for G.711 was performed with and without loss concealment. The evaluation for G.729 was performed with the standard concealment as well as with newly proposed (in [82, 83]) concealment schemes. The results pertaining to G.729 and three degrees of burstiness ($ulp =$ 0.1%, 0.3%, 0.5%) with the standard PLC are shown in Figure 3.4. As can be seen from Figure 3.4, contrary to G.711, the degree of impairment in G.729 is not sensitive to the degree of burstiness in speech loss; this is attributed to the robustness of the loss concealment method in G.729.

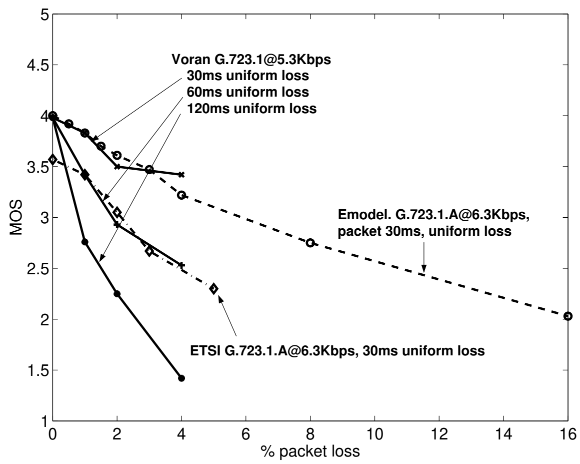


Figure 3.5: G.723.1 quality under various packet loss conditions. Packet Loss Concealment has been used in all experiments.

Voran [98] studied the effect of loss on voice encoded with G.723.1 with VAD at 5.3 Kbps. Uniform loss rates ranging from 0 to 4% were considered, with speech loss durations equal to a single frame (30 ms), two consecutive frames (60 ms) and 4 consecutive frames (120 ms). The results are shown in Figure 3.5. The small deviation (initial MOS = 4 vs. 3.98) is due to the different encoding schemes considered for G.723.1 by the Emodel (MP-MLQ) and by [98](ACELP).

Several among the above-mentioned studies and several other studies have been compiled into documents published by the ITU, [35, 36, 38] and ETSI, [23]. The latter constitute very good references. Work initiated at ETSI, resulted in the development of a group of standards by ITU-T in 1996, known as the "Emodel" and defined in [35, 36, 38]. Recommendation G.107, [35], defines the Emodel. Recommendation G.108, [36], provides guidelines on how to use Emodel for network planning. Recommendation G.113, [38], collected results from studies that applied packet loss to G.711, G.723 and G.729. These results are shown in Figures 3.3, 3.4 and 3.5 respectively using the label "Emodel". The results for G.711 with packet loss concealment, for both uniform and bursty loss, are taken from [21]. In addition, a curve for G.711 without packet loss concealment is provided, which agrees with the results for $10ms$ packet obtained by [29], see Figure 3.3. Work along the same lines is still ongoing in ETSI until today and a recent technical report is [23], dated in 2000. We plot the results provided for G.711, for G.729 and G.729A, and for G.723.1, contributed by Nortel Networks, see [23], in Figures 3.3, 3.4 and 3.5 respectively, under the label "ETSI". The results shown are chosen to have packet loss concealment and packet sizes of 10, 20 and 30 ms, for the purpose of comparison with the similar experiments. The entire set of results provided by this contribution has a MOS lower than the other studies.

Discussion

One can make the following observations, looking at the above results coming from different sources. First, some comparable experiments have slightly *different starting MOS* (i.e. at the absence of loss), due to different conditions of the experiments. However, the rate of MOS degradation with packet loss is similar for comparable experiments, which confirms their consistency. The *slope of MOS degradation* seems to depend mainly on the use of packet loss concealment. In experiments with packet loss concealment and 10 ms loss duration, MOS drops by roughly $1-1.5$ unit every 10% of packet loss; in experiments without packet loss concealment, MOS drops much faster, by roughly 1 unit every 1% of packet loss. Larger loss durations result in increased degradation but the slope does not change drastically. The encoding scheme does not seem to affect much the degradation slope either. Although it is simplifying to decouple the encoder from the effect of packet loss, this seems to be practically the case. Clearly, the starting quality of the encoder determines the maximum allowed packet loss in the network. G.711 that starts at high intrinsic quality can incur up to 15% loss and still sustain an acceptable quality. These high loss rates are unacceptable for G.723 that starts at an already low intrinsic quality. Finally, *bursty loss*, which has been repeatedly reported to be the loss pattern in the Internet and has also been the case for most of the paths in Chapter 2, seems to affect the resilience of G.711 but not that of G.729, attributed to the concealment used.

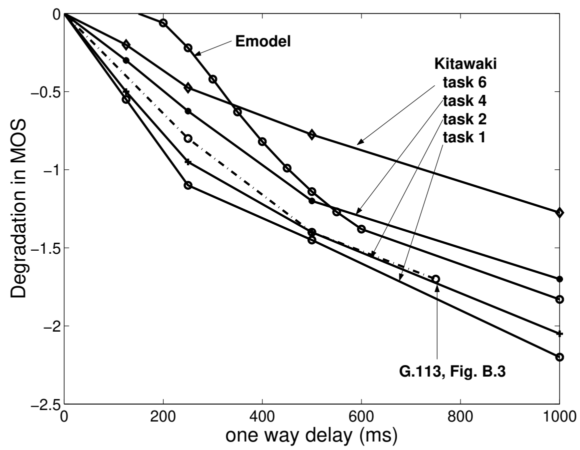


Figure 3.6: Loss of interactivity due to one way delay in echo free environments, as reported by various sources: (i) Kitawaki et.al. in NTT Labs, [58] (ii) Emodel in ITU-T recommendation G.107, [35] (iii) ITU-T recommendation G.113, [38].

3.1.2.4 Degradation in Interactivity due to Delay

In 1991, a study by Kitawaki et.al. in NTT Labs, [58], assessed the loss of interactivity due to large end-to-end delay, in echo free telephone circuits. They introduced various amounts of delay and they studied the resulting Mean Opinion Scores, conversational efficiency (i.e. the additional time needed to complete a conversation) and detectability thresholds (i.e. the ranges of delay that listeners were able to detect the delay introduced), using groups of subjects varying with various degrees of expertise. Six conversational modes (also called "tasks") were considered, each having a different switching speed between the communicating parties and thus a different sensitivity to delay. The most stringent task is Task 1, where people take turns reading random numbers as quickly as possible. On the other extreme, Task 6 is the most relaxed type, free conversation.

Recommendation G.114, published in 2000, also focused on the loss of interactivity due to delay, assuming echo free environments, [39]. Traditionally, a one way delay up to 400 ms was considered acceptable for planning purposes; recommendation G.114 emphasized that this is not the case for highly interactive conversations and declares 150 ms acceptable for

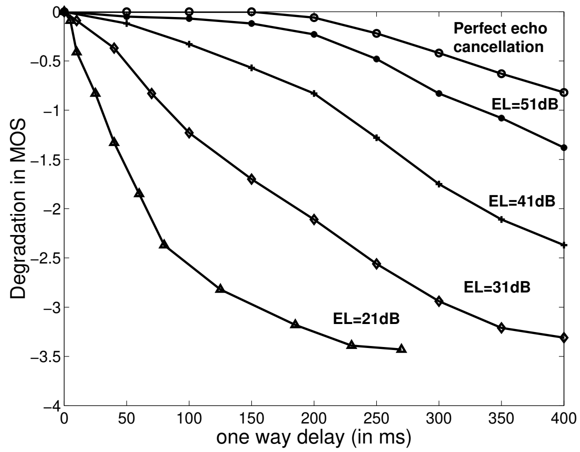


Figure 3.7: Degradation in MOS due to echo, according to the Emodel (G.107).

most applications in echo free environments. In Annex A of G.114, estimates for the delay incurred in various components of circuit and IP networks are provided. In Annex B, results from the above-mentioned [58] and other similar studies are collected.

The Emodel standards also provide a formula for calculating the loss of interactivity as function of the one way delay, in the absence of echo, [35]. The Emodel [35] (and later studies based on it such as that of Cole at.al, [19]) are lenient in a sense that they predict no degradation for delay below 150 ms. A possible explanation is that the Emodel curve does not take into account the aspect of the different conversational modes (or tasks) and the expertise of the subjects that participated in the subjective experiments. Indeed, in [58, 39] it is reported that trained subjects were able to detect delays as low as 50 ms in highly interactive tasks, while delays in free conversation could be detected only at about $350ms$, even by trained subjects.

The degradation in MOS as delay increases, as reported by all three sources, is shown in Figure 3.6.

3.1.2.5 Echo Impairment

As discussed in section 3.1.2, echo can cause major quality degradation, if it is not adequately canceled. Its effect is amplified by large delays. The Emodel provides formulas that allow to calculate the impairment due to talker ($I_{dte}(m2e, EL_2)$)and listener ($I_{dle}(m2e, EL_1)$) echo respectively, given some transmission parameters. $m2e$ stands for the one way or "mouth to ear" delay; EL_1, EL_2 are the echo losses in dB at the points of reflection and their value depends on the echo cancellation used. $EL = \infty$ (infinite echo loss) corresponds to perfect echo cancellation. $EL = 51\,dB$ corresponds to a simple yet efficient echo controller. Figure 3.7 shows the degradation in MOS, due to the combined talker and listener echo.

3.1.2.6 Delay Jitter Impairment

Voice packets are sent at regular intervals as shown in the first line of Figure 3.8. However, they incur variable delays in the network. In particular, as we discussed in length in Chapter 2, delay jitter happens in the form of spikes. Spikes result in voice packets arriving bunched up after a gap as long as the jump of the spike, as shown in the second line of Figure 3.8.

If the packets are played out as they arrive at the receiver, the pitch of the speech will be affected and there may be gaps in the speech. We are not aware of subjective results that systematically quantify the effect of playout jitter in speech quality. Special speech processing techniques to allow for jitter in the speech playout while preserving the pitch, have been proposed since 1980s, e.g. [28, 94]. One such a technique, called "time scale modification", first appeared in [94] and was further exploited in [64]. In [27, 64], it is reported that up to 25-30% variation in the playout was found acceptable in informal subjective tests, when using this technique.

The receiver has no control over the send and receive time of the packets. However, it has control over their playout times, and it can hide the network jitter from the voice application. This decision is called playout scheduling and is the topic of a Section 3.2. Depending on the playout, the spike may have different perceived effects, shown in the last

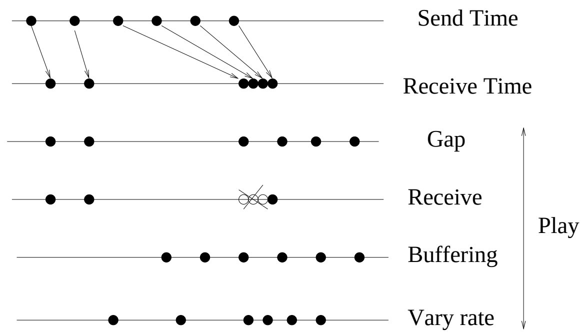


Figure 3.8: Delay Jitter Impairment

four lines of Figure 3.8. The first option is to introduce a gap in the speech. The second option is to play the packets that arrive on time, and consider the ones that arrive late as lost. The third option is to do buffering in anticipation of spikes, resulting in additional delay and occasional loss. The last option is buffering combined with variable playout rate, as suggested in [64].

In all cases, network delay jitter results in some kind of impairment. If it is not eliminated, then it results in speech playout jitter; to the best of our knowledge, its perceived effect has not been systematic quantified. If network delay jitter is eliminated by means of playout scheduling, then it results either in speech loss (quantified in Section 3.1.2.3) or in additional delay (quantified in 3.1.2.4) or in the modification of silence intervals (a study of which can be found in [29]).

3.1.3 Combining all Impairments using Emodel

The above-mentioned Emodel started as a study by ETSI and eventually was standardized by ITU-T in [35][36][38]. Comprehensive studies of the Emodel can be found in [19] and [19, 97]. It is a computational model that uses transmission parameters to predict the subjective quality of voice quality. It gives a rating R for the quality of a speech sample, on a scale from 0 to 100, whose translation to quality and MOS is shown in Fig. 3.2 and Fig. 3.9. The Emodel combines different impairments based on the principle that *"the perceived* *effect of impairments is additive, when converted to the appropriate psycho-acoustic scale (R)".*

$$R = (R_o - I_s) - I_d - I_e + A \text{ (3.1)}$$

The details of equation (1) are as follows. R_0 is the basic signal-to-noise ratio based on send, receive loudness, electrical and background noise. I_s captures impairments that happen simultaneously with the voice signal, such as sidetone and PCM quantizing distortion. Both Ro and Is terms are intrinsic to the transmitted voice signal itself and do not depend on the transmission over the network. Thus, they are irrelevant for the purpose of comparing VoIP to PSTN calls. Their default values suggested in [35], are incorporated in the initial value of R, i.e. before any transmission in the network. I_d captures the delay impairments, i.e. the loss of interactivity (discussed in subsection 3.1.2.4) and the echo (discussed in subsection 3.1.2.5). I_e stands for "equipment factor" and captures the degradation in quality due to compression (discussed in subsection 3.1.2.2) and loss during transmission (discussed in subsection 3.1.2.3). A stands for the advantage factor that captures the fact that users might be lenient in their judgment, willing to accept some degradation in quality in return for the ease of access, e.g. using cellular or satellite phone. For the purpose of comparison to PSTN calls, this factor is set to 0.

The Emodel is important in our study for two reasons. First, it provides data for quantifying the degradation in MOS, due to delay (I_d) and loss (I_e) impairments, many of which are compilations of the above-mentioned studies, into a common framework. Furthermore, the Emodel has the validity of an ITU-T standard. Therefore, we use the Emodel as our main reference to quantify the delay and loss impairments and use the other references to verify and complement it. In addition, the Emodel also models the effect of noise and other speech related impairments, thus allowing us to take them into account without going into detail. Second and most important, the Emodel combines all the impairments, including loss and delay, into a single rating, using the additivity in an appropriate psycho-acoustic scale. Therefore, it allows us to give a single rating (in R or MOS) to a phonecall and

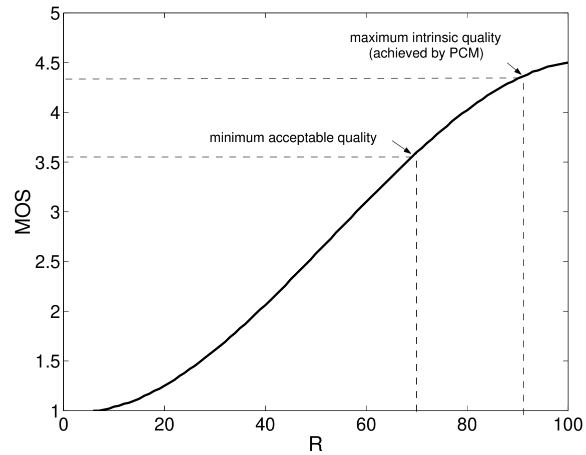
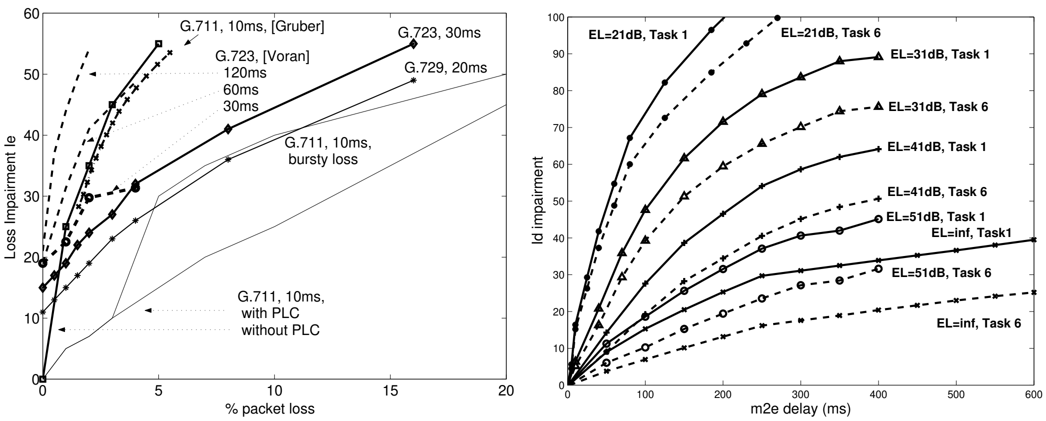


Figure 3.9: Correspondence between Emodel rating (R) and Mean Opinion Score (MOS), provided in G.107.

obtain statistics in terms of this rating. Notice that this single rating comes from averaging subjective results obtained for different speech samples and for different instances of the impairments. Therefore, it is a statistical metric, appropriate for planning purposes, but inaccurate for assessing the quality of a specific speech sample.

G.107, [35], also provides the correspondence from R to MOS, which is shown in Figure 3.9. It is interesting to observe that the curve is almost linear, especially in the area of interest, i.e. in the range of acceptable quality. This observation combined with the additivity of impairments in the R scale means that the impairments are also additive in MOS, for all engineering purposes. In this study, we present results in terms of MOS but the underlying calculations are in the R scale. We use the Emodel terminology interchangeably with the term "degradation in MOS"; I_e will be used interchangeably with "degradation in MOS due to speech distortion"; I_d will be used for "degradation in MOS due to delay". Combining the speech impairments due to compression and packet loss (in sections 3.1.2.2 and 3.1.2.3 respectively) into a single rating results in the I_e curves shown in Figure 3.10(a). Combining the delay impairements due to loss of interactivity (section 3.1.2.4) and echo (section 3.1.2.5) into a single rating results in the I_d curves shown in Figure 3.10(b).



(a) Loss Impairment (Ie) as a function of the packet loss rate. Parameters to be specified fro each curve: (i) compression standard (ii) packet size (iii) use of PLC (iv) source of data

(b) Delay impairment as a function of the mouth-to-ear (m2e) delay. Parameters: (i) task 1, 2, ...6 (ii) echo loss (EL) in dB

Figure 3.10: Loss and Delay Impairments in the R scale

Furthermore, by combining the delay and loss impairments using Equation 3.1, we obtain a single rating R (or equivalently MOS). Figure 3.11 shows contours of quality as a function of delay and loss. There are different pairs of (delay, loss) that lead to the same rating. They might still sound different but on average, they will be rated similarly by users. This figure will be useful in the Playout section 3.2, when we study the loss-delay tradeoff in playout scheduling. If a target overall quality is desirable, the margin between the intrinsic quality of the encoder and the acceptable quality level, can be consumed either to improve speech quality or to improve interactivity.

3.1.4 Rating Phonecalls

We would also like to simulate the rating that a user would give after talking on the phone for several minutes. This is also the goal of commercial systems for online monitoring of VoIP quality, such as those proposed in [14] and [99]. In doing so, one has to deal with the following issues.

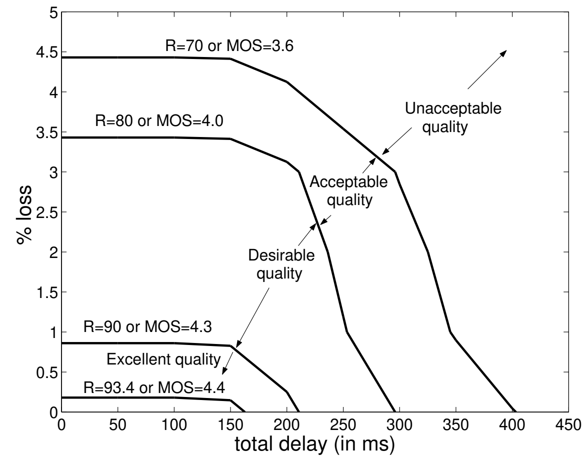


Figure 3.11: Voice Quality contours in terms of R/MOS, for G.711, 10 ms packet, bursty loss pattern, free conversation, echo loss EL=51 dB.

3.1.4.1 Issues in Rating VoIP Phonecalls

Loss in Internet Traces

To appropriately use the above data to assess the performance of VoIP over Internet traces, we have to make sure that we apply them for the same conditions under which the subjective results have been obtained. There are some important conditions underlying those experiments: (i) the specific durations of speech samples used (ii) the pattern of loss applied (iii) the stationarity of applied impairment.

Let us address (i) and (iii) together. Speech samples in the order of 2-3 seconds have been used to assess the effect of loss (I_e) and conversations in the order of 1 minute have been used to assess the delay impairment (I_d). In addition, the statistics of loss or delay under assessment did not change during the experiments. Therefore, one should use carefully the I_d and I_e curves when evaluating phonecalls lasting several minutes, during which impairments vary considerably. A natural approach is to divide the call duration into *fixed time intervals* (in the order of 2 sec for I_e and in the order of 1 minute for I_d) and assess the quality of each interval independently. Another interval that intuitively makes sense to use for assessment is the talkspurt duration.

A second consideration is the burstiness in loss. There is no guarantee that the assumption of uniform loss (or even bursty loss with consecutive loss up to 100 ms, which is the assumption behind the only curve available for bursty loss), holds for Internet traces. As subjective results for long and arbitrarily bursty loss durations (which is the case in the Internet) are not available, performance evaluation in terms of MOS should be supplemented with statistics about the loss durations themselves. In particular, loss durations above 100ms are difficult to conceal at the receiver, lead to loss of entire phonemes. In addition to the network loss, a defining factor for the burstiness of loss, is the delay jitter and how it is handled by the playout at the receiver.

An approach that attempts to address together the burstiness and the non stationarity of Internet impairments is that proposed in [14]. They defined high and low loss periods of variable durations, called "bursts" and "gaps" respectively. If the number of consecutive received packets between two successive losses is less than a minimum value g_{min}, then the sequence of the two lost packets and the intervening received packets are regarded as part of a burst; otherwise, part of a gap. The choice of g_{min} becomes then important. A small g_{min} would give small burst durations with high packet loss rate; on the other hand, a large g_{min} would group neighboring losses into one burst with smaller loss rate averaged over a larger period of time. As the loss rate in a gap is $(100/gmin)\%$, we choose $g_{min} \geq 1sec$ that leads to less than 1% loss in gaps and to meaningful durations in the order of a few seconds. The use of variable intervals appropriately addresses the burstiness in the following ways. First, the loss during gaps is enforced to be uniform by the definition of a gap. During burst periods, we use the I_e curves for bursty loss. Second, by dynamically partitioning each trace into its own gaps and bursts, we emphasize the periods of high loss, as opposed to calculating the loss rates over arbitrarily long intervals and smoothing them out.

The above approach is in the right direction for addressing the loss burstiness and the time variability of impairments. Furthermore, we use the idea of bursts and gaps avoiding computational simplifications, that are often used to meet low processing time constraints.

On the contrary, in [14], only the average gap and burst (instead of all the gaps and bursts) was recorded and updated. However, even with this approach, one cannot exactly match the loss conditions in the Internet with the loss conditions in the assessment studies. As a first example, loss durations can exceed in practice 100ms during bursts. However, this typically happens when loss rates are also high (i.e. above 20%) and MOS is low anyway. In that case, the exact loss duration does not change the -already very low MOS- result. A second example is the choice of g_{min}: (i) we experimented with many values and we found the end-result to be relatively insensitive (ii) the range $[160ms, 1.28sec]$ has been used in [14] and reported to correlate well with subjective listening, in [15, 16].

Perceived Quality Aspects for Long Phonecalls

A long phonecall consists of multiple short intervals. Independent MOS rating of each short interval has been shown to correlate well with the continuous instantaneous rating of the call, [25]. Evaluating each interval leads to transitions between plateaus of quality, as represented by the dashed line in Figure 3.12. However, transitions between periods of high and low loss are perceived with some delay by the listener, as opposed to abrupt changes between plateaus. For example, a human would perceive and rate the changes in quality following the smooth solid line instead of the dashed one in Figure 3.12. This effect is known as *recency effect* and was studied first in [26]. Therefore, a model for predicting the instantaneous user perceived quality should take into account time constants. Instantaneously perceived I_e is considered by [14] to converge toward the $I_e(loss)$ for a gap or burst, following an exponential curve with time constants $T_{bad} = 5\,sec$ for the high loss and $T_{good} = 15\,sec$ for the low loss periods. The constants have been calculated in [14, 16] to match the delays in noticing a transition, as reported by [26]. It has been observed that it takes longer for a subject to forget transitions to bad than to good quality.

A second observation has to do with the overall rating of a long call. It has been shown, that the rating an individual would give at the end of a call is captured at a first

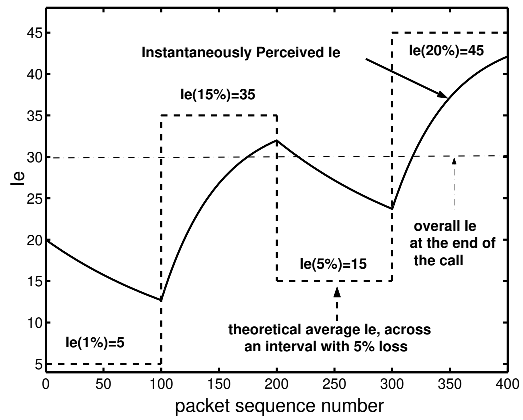
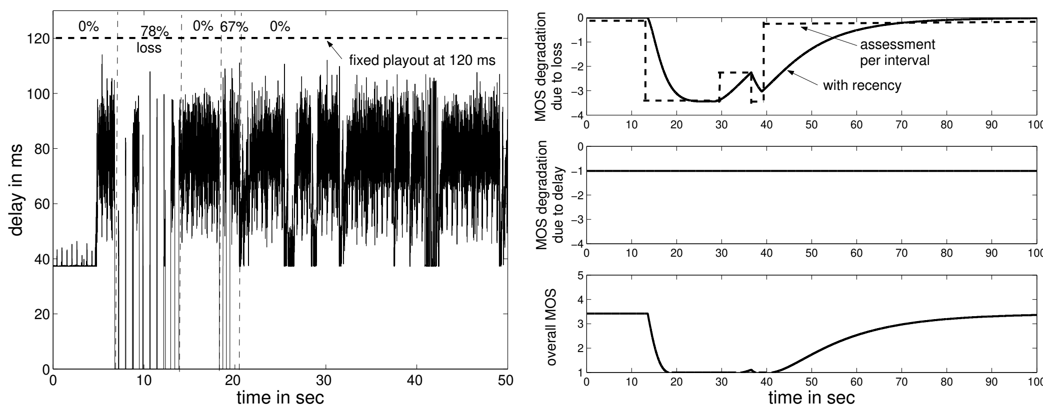


Figure 3.12: Example of a call partitioned into 4 shorter periods. The theoretical I_e for each short interval is shown in dashed line. The instantaneously perceived I_e (taking into account the recency effect) is shown in solid line.

approximation by the time average of the instantaneously perceived MOS, [25]. In [14], the final rating is further adjusted to include the effect of the last significant burst and had good correlation with subjective results, [15], [16]. Notice however, that an individual might forget some bad moments in the middle of the call, that a network provider might be interested in monitoring and eliminating. Therefore, in our assessment of an entire call, we use not only the rating described in [14] to simulate the opinion of an individual, but also the worst quality experienced during a call, in order to highlight bad events. For example, for the call of Figure 3.12, we would report both the overall $I_e = 30$ and the minimum $I_e = 13$ that happens at the 100th packet.

3.1.4.2 Example of Call Assessment

Let us consider a phonecall taking place during some minutes over an Internet trace. In summary, we can obtain a MOS rating for this call, as follows. First, we partition the trace into segments of variable size (gaps and bursts), as described in Section 3.1.4.1. Then, we assess each segment in three steps. First, we assess the degradation in speech quality (at the encoder and due to packet loss in the network and in the playout buffer) using the



(a) 100sec trace, EWR-P2-SJC, Thu at 20:10

(b) Assessment of a 100 sec over the trace

Figure 3.13: Example of assessing a call over 100 sec trace.

curves for G.711, G.729 and G.723 corresponding to the Emodel, in Figures 3.3, 3.4 and 3.5, respectively. We are particularly interested in the bursty loss which is the case in the Internet traces. In Emodel terminology, this first step means that we calculate the I_e factor. Second, we assess the loss of interactivity using the NTT study and the strict (task 1) and lenient (task 6 or free conversation) tasks in Figure 3.6. We assess the degradation due to echo, if any, using Figure 3.7. In Emodel terminology, this second step means that we calculate the I_d factor as $I_d = I_{d,echo}(m2e, EL) + I_{d,interactivity}(m2e)$. Third, we calculate the overall rating R from Equation (3.1) and we translate it into MOS. From the segment-by-segment rating we can obtain the instantaneous rating, by applying the recency effect, as described in Section 3.1.4.1. From the instantaneous quality we can obtain an overall rating, as also described in Section 3.1.4.1.

As a concrete example let us consider 100 sec during our familiar loss event of Figure 2.11 (path EWR-P2-SJC, on Thu at 20:10) and let us assume that a call is taking place over these 100 seconds. This trace exhibits high loss rates due to loss in the network. For simplicity, let us assume a fixed playout at 120 ms, so that there is no additional loss due to late arrivals. Using the gaps and bursts approach and $g_{min} = 1sec$, the call is partitioned in 5 intervals, as shown in Figure 3.13(a): a gap with 0% loss, a burst with 78% loss, a gap with 0% loss, a burst with 67% loss and a gap with 0% loss. The dashed line on the top graph of Figure 3.13(b) shows the impairment due to loss for each of the five intervals. The continuous line is the instantaneous rating, considering the recency effect. The second graph of 3.13(b) shows the delay impairment due to the 120 ms fixed playout, considering Task 1 and EL=51dB. The third graph shows the combined instantaneous MOS as a function of time. The overall rating at the end of the call can be found as the average MOS degraded by the effect of the last burst (i.e. at $time = 40sec$).

3.2 Playout Scheduling for VoIP

An important component of the VoIP system is the playout scheduling at the receiving end. Its purpose is to absorb the variations in network delay and provide a smooth playout for the voice application. The choice of playout algorithm, strongly affects the overall performance.

In Section 3.2.1, we first describe the function of playout scheduling and we review previous work. Then, we study some of the existing algorithms, using the backbone traces. In Section 3.2.2, we discuss the simplest approach, i.e. the fixed playout, which can be useful in many traces, if the choice of fixed value is based on some knowledge of the delay on the path. In Section 3.2.3, we study a widely known algorithm that uses a moving average based estimation; we find that it has a number of problems when applied over these backbone traces.

In Section 3.2.4, motivated by the problems with the previous two approaches, we discuss issues inherent in the design of any playout algorithm and recommend some possible choices. In particular in Section 3.2.4.1, we discuss the process of learning the network delay. Based on our study of the traces in Chapter 2, we recommend that the appropriate delay properties to track are the height and the distance of the delay spikes. In Section 3.2.4.2, we discusss the adaptation of playout to the network delay and we recommend a fast increase of the playout to follow delay spikes and a slow decrease after a spike. In Section 3.2.4.4, we use our understanding of voice quality aspects to show how the loss-delay tradeoff leads to a maximum overall perceived quality.

In the following three sections, we combine the elements of Section 3.2.4, to create three concrete playout algorithms. We refer to these algorithms as "conservative", "configurable", and "intermediate" modes, because they achieve different levels of loss and delay, depending on the relative importance of speech clipping and delay for the user/application. We evaluate their performance using representative traces. In Section 3.2.5, we propose a conservative algorithm that monitors network delay and chooses a fixed value, to be the maximum delay experienced recently; this is like an infrequently adjusted fixed scheme, thus the name *'Assisted Max Fixed'*. It is conservative in the sense that it aims at perfect speech quality even at the cost of high delay. However, if delay is in general low, one might want to decrease the playout in between high spikes. This leads us to the *'Increase Fast-Exp Decay'* algorithm of Section 3.2.6, which leaves the choice between speech quality vs. delay to the user. Depending on that choice, the reaction of the algorithm can range from risky to conservative, thus the term "configurable". In Section 3.2.7, the *'Maximize-MOS'* algorithm is proposed that resolves the loss-delay tradeoff in a way to maximize the overall perceived quality $MOS(loss, delay)$, along the lines of Section 3.2.4.4; thus the name "maximize MOS" for this algorithm. It is an "intermediate" algorithm in the sense that it explicitly includes both delay and loss in its objective function, and thus results in both a low loss rate and a low average delay.

In Section 3.2.8, we compare all algorithms using two representative traces. In Section 3.2.9, we summarize the study of playout algorithms and discuss limitations and possible extensions.

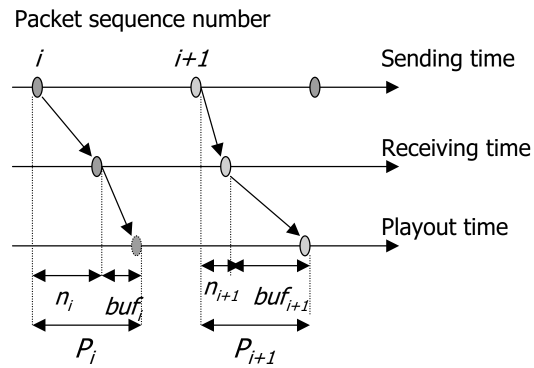


Figure 3.14: Playout scheduling. Packet i incurs network delay n_i and buffering delay buf_i. Its playout time, measured from the time it was sent is $p_i = n_i + buf_i$.

3.2.1 Definitions and Previous Work

Voice packets are produced at fixed intervals. However, the network introduces variable delays, as discussed in Section 3.1.2.6, that need to be removed at the receiving end. This is achieved by holding a packet i arriving with network delay n_i, for an additional buffering delay buf_i, until a later playout time p_i in order to ensure that there are enough packets buffered to be played out continuously; see Figure 3.14. Any packet arriving after its scheduled playout time, p_i, is discarded. The playout buffer can be fixed or adaptive, depending on whether p_i is the same for all packets or can vary from packet to packet.

Much work has been conducted since 1970s on playout of packetized voice; [18] and [67] are among the earliest. Recent work has addressed the problem specifically for the Internet.

A fixed playout buffer scheme schedules the playout of packets so that the end-to-end delay p (including both network and buffering) is the same for all packets. It is important to select the value p so as to maximize the quality of voice communications. Indeed, a large buffering delay decreases packet loss due to late arrivals but hinders interactivity between the communicating parties. Conversely, smaller buffering delay improves interactivity but causes higher packet loss in the playout buffer and degrades the quality of speech. The value of fixed end-to-end delay should be chosen based on some knowledge of the delay in the network. However, such an assessment may not always be possible and/or the statistics of the network delay may change with time. For these reasons, adaptive playout schemes are considered.

Adaptive playout schemes monitor the network delay and its variations and adjust accordingly the playout time of voice packets. Early on in 1994, a number of algorithms were proposed in [77], to monitor network delay, estimate the delay d_{av} and delay variation v using moving averages, adapt the playout time to $p = d_{av} + 4v$ at the beginning of each talkspurt but keep it constant throughout a talkspurt. In addition, it was also proposed to detect delay spikes and adapt p faster during the spike periods, a proposal that became an integral part of most schemes that followed. The scheme proposed in [69] improved over [77] by using a delay percentile rather than a moving average, to estimate the network delay; the improvement achieved came at the expense of increased state and processing. The same paper provided theoretical delay bounds obtained by an offline algorithm, given a trace and a target loss rate. In [22], the prediction of network delays was further improved by minimizing the normalized mean square prediction error. A second group of playout algorithms adapt the value of delay p on a packet-per-packet, instead of a talkspurt-per-talkspurt, basis and thus allows for following delay variations even within a single talkspurt. The scheme in [88] followed such an approach, but it did not take into account the voice signal itself and the pitch of the speech signal was affected by the playout speed. The work in [64] adjusted the playout rate by scaling voice packets while preserving the pitch, using a time-scale modification technique.

All playout schemes can accidentally incur or allow for some loss. (For example, one may choose ignorantly or purposely a low fixed playout. Setting the playout at $p = d_{av} + 4v$ allows for packets with delay at the tail of the distribution to get lost.) However, [76, 78, 69, 64] allow for explicitly specifying a target loss rate. As playout strongly affects both delay and loss, it can be combined with FEC, as in [78], or with loss concealment, as in [64].

Algorithm 1 Generic Adaptive Playout Algorithm

For every packet (i) received:

Estimation: *monitor past and predict future network delays*

- *calculate the network delay* n_i*(*) of the packet*
- *decide whether we are in NORMAL or SPIKE mode*
- *update some expression of the recent delay history*
- *predict the future delays(*$p_{predicted}$*) based on the history and the mode*

Adaptation: *Is it time to adapt the playout?*

- *If yes, then: adjust the playout delay* p *to some function of* $p_{predicted}$ *in order to achieve some goal*
- *If not, then: keep the same playout delay* p *as before*

Playout scheduling algorithms can be summarized by the generic algorithm 1. There are two distinct parts (i) estimation of network delays, which is updated for every received packet and (ii) adaptation, which is called less frequently at appropriate times. Moving averages, histograms and events counting are some of the ways used to capture delay history and predict the future during "normal" times. Delay history is less useful for prediction during delay "spikes", when rapid adaptation should be used. Spikes are detected by looking at jitter thresholds, absolute or relative to the minimum delay. Based on the delay estimation, algorithms can then adjust the playout delay p to follow the estimate. The exact adaptation function $p(p_{predicted})$ depends on the objective function one tries to optimize: a conservative algorithm will try to avoid loss while a "risky" algorithm will allow some loss to save on interactivity. The delay-loss tradeoff is inherent in any playout algorithm. Appropriate times for adaptation are those that are not perceived by the user, as discussed above. The effectiveness of adapting at the beginning of a talkspurt is limited by the length of the talkspurt compared to the frequency of spikes. Silence detection becomes then important; modern compression schemes, such as G.729B, use dynamic hangover that lead to shortened talkspurt and silences and give the algorithm more chances to react. At the limit, talkspurts become too short, the algorithm becomes a continuous adaptation per packet, as in [64].

Synchronization in the context of Playout Scheduling.

Most of the work on playout scheduling, e.g. [77], uses the notion of network delay n_i (the time from when a packet i was sent until the time the packet is received), see Figure 3.14. It can be calculated by subtracting the sender's timestamp from the time the packet arrives at the receiver, given that there is synchronization between sender and receiver. However, these values are accurate only if the clocks at the sender and the receiver are synchronized. Perfect accuracy (in the order of μs) can be achieved using GPS. Satisfactory accuracy (in the order of ms in periods of hours to days) can be achieved using the NTP protocol, or proprietary mechanisms depending on the audio tool. Additional methods for achieving synchronization in the context of VoIP are discussed in [78].

A way to avoid the need for synchronization is to measure and estimate the jitter (instead of delay) which can be measured locally at the receiver. For example, in [69], the virtual timestamps of the NeVoT and VAT tools are used and the variable components of the network delay are considered (by subtracting the minimum calculated network delay). This way, clocks do not need to be synchronized but they should still not drift. Estimation and removal of clock skew is a research topic by itself (e.g. see the thesis by S.Moon [68]). The probes used in this study are actually synchronized using GPS. Using the probes to simulate an application, implies that either there is a good enough synchronization between the sender and the receiver timestamps (a few ms difference does not make a difference for voice) or, alternatively, only variable delays are considered.

3.2.2 Fixed Playout

The simplest approach is fixed playout scheduling. A fixed scheme schedules the playout of packets so that the end-to-end delay p (including both network and buffering) is the

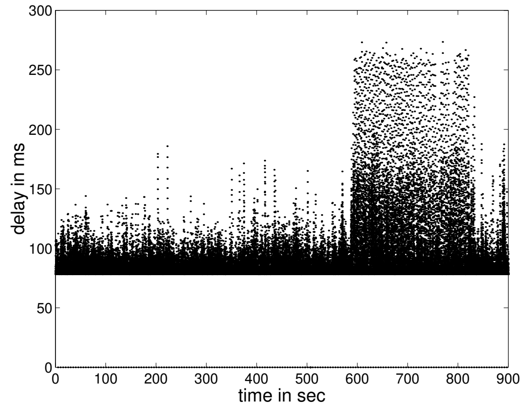
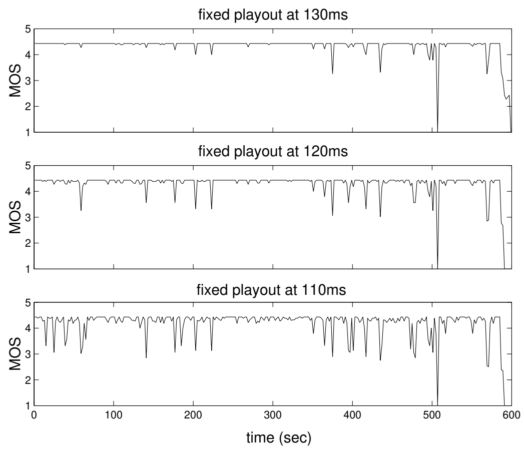


Figure 3.15: Example trace to be used for the sutdy of *Fixed* and *Spike-Det* algorithms. Path THR-P_1-ASH, Wed 14:00-14:15.

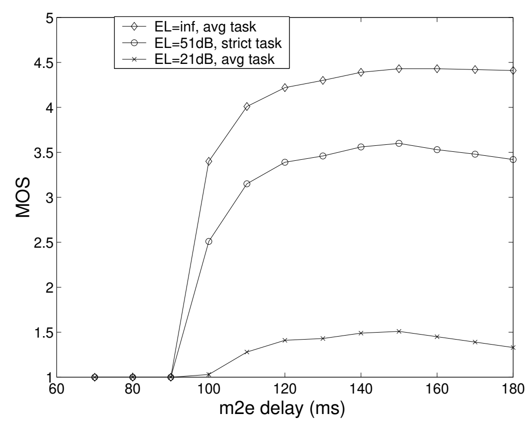
same for all packets. It is important to select the value p so as to maximize the quality of voice communications. Indeed, a large p decreases packet loss due to late arrivals but hinders interactivity between the communicating parties. Conversely, smaller p improves interactivity but causes higher packet (and thus speech) loss due to late arrival. The choice of the playout deadline p becomes then important.

Let us consider the example trace of Figure 3.15 and study the effect of fixed playout over this trace. (Notice that this trace also corresponds to the first 15 minutes of Figure 2.27(a)). Let us first look at the first 10 minutes of this trace, when delay is from 78 ms to 180 ms. Figure 3.16(a) shows that speech quality (MOS) per talkspurt improves as we increase the playout value from 110ms to 130ms, and thus we drop fewer packets due to late arrivals. Figure 3.16(b) shows the combined MOS at the end of a 10 minute call, considering both speech quality and delay penalty. Note that a poor echo cancellation ($EL = 21dB$) and a highly interactive task ("strict") increase the delay penalty. The delay range is low enough (delays around 150 ms are not noticed), to allow for large fixed playout delay: for these 10 minutes the larger the fixed playout value, the better (the MOS is strictly increasing with $m2e$ delay).

Let us now consider the first 15 minutes of the example trace in Figure 2.27(a). Delay



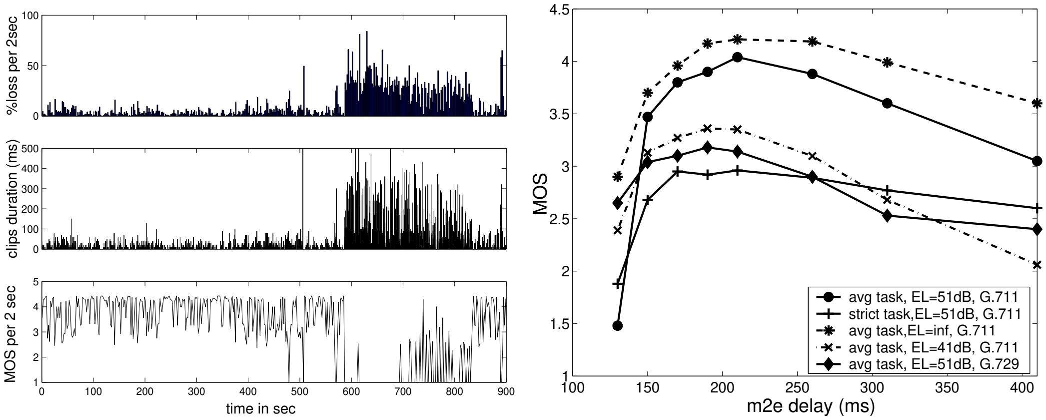
(a) Speech quality (MOS) per talkspurt considering 3 values of fixed playout



(b) Overall MOS (considering loss and delay aspects) for a range of fixed playout values

Figure 3.16: Fixed playout over the 10 minutes (Wed, 14:00-14:10) on path THR-P1-ASH

is considerably higher during the last 5 minutes. Figure 3.17(a) shows a bad speech quality for a low fixed delay (100 ms). Figure 3.17(b) shows the overall rating at the end of a call (considering both delay and loss impairment, as computed in Section 3.1.4.1) for fixed delays from 100 to 400 ms. This delay range is high enough to be noticed. Therefore the $MOS(delay)$ curves have a maximum due to the loss-delay tradeoff, consistently with the discussion of Section 3.2.4.4. As fixed playout delay increases, there is less loss due to late arrivals, and the combined MOS increases. However, above 200ms, the delay penalty is more severe than the gain in speech quality and the overall quality starts dropping again. This tradeoff holds for all compression schemes (G.711, G.729), conversation tasks (strict or average) and echo cancellation (EL). However, the maximum achievable MOS and the delay for which it is achieved, varies with these parameters, as it is the case in Figures 3.16(b) and 3.17(b). For example in Figure 3.17(b), G.729, which starts at a lower intrinsic quality, can achieve a maximum $MOS = 3$ and thus cannot be carried at acceptable quality levels during the 15 minutes period. Similarly, a strict interactivity requirement (e.g. "task 1") or



(a) Speech quality, for fixed playout at 100ms

(b) Rating at the end of the 15 min call, for a range of fixed playout values and VoIP parameters

Figure 3.17: Fixed playout over 15 minutes (Wed 14:00-14:15) on trace THR-P1-ASH

an acute echo (e.g. $EL = 41\,\mathrm{dB}$), would lead to $max\,MOS \cong 3$, which is also unacceptable.

The best choice of fixed playout for the entire 15 minutes is approximately $200\,ms$. However, the first 10 minutes of this trace are different from the last 5 minutes. If one can adjust the fixed value in the middle of a call, then a more appropriate choice is 130 ms for the first 10 minutes and 250 ms for the last five minutes.

A different trace with low delay and low delay variability, like those that belong to providers P_6 and P_7, could use a conservatively high fixed playout (e.g. total delay 150 ms) without any perceived effect. If the end-points can achieve an estimate during the call setup, e.g. using roundtrip time estimates, and realize that delay is below the 150 ms range (for example in the case of on-campus or local calls), then a conservatively high fixed playout is a reasonable choice. However, in cases that delay is higher than 150 ms or the delay range changes due to a route change, a congestion period or any other reason, then monitoring and adjusting of the fixed playout delay is needed throughout the call.

Clearly, the choice of fixed playout value is critical and should be based on some knowledge of the delay in the network. Because such a knowledge may not always be possible to

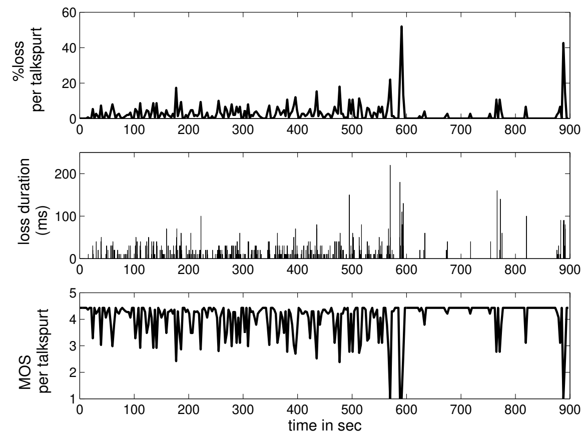


Figure 3.18: Loss impairment per talkspurt caused by the spike-det algorithm, using its default parameters, over the 15 minutes trace of Figure 3.34.

obtain and/or the statistics of the network delay may change with time, adaptive playout schemes are considered. In the following sections we discuss a number of playout algorithms. The algorithm of Section 3.2.5 in particular, called 'Assisted Fixed' , is the variation closest to fixed: a fixed playout with infrequently adjusted value, based on the delay history.

3.2.3 Moving-Average based Adaptive Algorithm: 'Spike-Det'

As a first example of adaptive playout algorithms, we considered the four schemes proposed in [77]. The reason for this choice, was that they are well known schemes and in particular computationally light, which makes them suitable for simple implementations. All four schemes used moving-averages for delay estimation. In particular, we used the fourth algorithm proposed in that paper, called spike-detection or 'spike-det' algorithm, with the default parameters recommended in the paper as our baseline adaptive scheme.

More specifically, the 'spike-det' scheme updates moving averages, for each received packet i, to estimate the network delay ($d_i = a \cdot d_{i-1} + (1-a) \cdot n_i$) and network delay variation ($v_i = a \cdot v_{i-1} + (1-a) \cdot |d_i - n_{i-1}|$). It then adjusts the playout delay to $p = d + 4v$, at the beginning of each talkspurt, similarly to what is done for the TCP round-trip delay estimates. An optimization was the detection of spikes by comparing the increase in delay

Table 3.2: Effect of the parameters on the performance of the *spike-det* playout over the 10 first minutes of Figure 3.34.

weight α	threshold ENTER	% loss	avg delay	avg MOS
0.99802	100ms	3.29%	103 ms	2.7
same	50ms	2.35%	108 ms	2.88
same	30ms	0.8667%	132 ms	3.18
same	20ms	0.17%	166 ms	3.26
0.875	20ms	7.55%	98 ms	1.82
0.90	100ms	6.82%	98 ms	1.96
0.95	same	5.1%	102 ms	1.48
0.98	same	2.6%	113 ms	2.66
0.99	20ms	1.5%	125 ms	2.92
0.99	30ms	2.62%	109 ms	2.67

to a certain threshold ($|n_i - n_{i-1}| > 2 \cdot |v| + ENTER$) and the faster adaptation during a spike ($d_i = d_{i-1} + (n_i - n_{i-1})$).

The scheme does not perform well over the backbone paths under study. For example let us consider the same 15 minutes trace (shown in Figure 3.35(a)) that we used to evaluate the fixed playout. Figure 3.18 shows loss caused and the resulting bad speech quality caused by the 'spike-det' playout when the recommended parameters ($\alpha = 0.998002$ and $ENTER = 100ms$) are used. During the first 10 minutes, the scheme tried to follow the network delays too close during the first 10 minutes, thus leading to significant loss rates and many clips of small durations. The scheme also resulted in long loss durations in the transition from low to high delays around 600sec. Finally, it overestimated the delay during the last 5 minutes, thus leading to no loss but to considerable delay impairment. Overall, it kept the average delay low (122 ms) during the 15 minutes and it gave a rating at the end of the call, $MOS = 3.6$, which is acceptable but not excellent.

Table 3.2 shows the sensitivity to the tuning of α and $ENTER$. Furthermore, there is no obvious way to tune these parameters. A small $ENTER$ threshold makes the scheme detect false spikes and overshoot in delay; conversely, a large threshold does not make use of the spike detection mechanism and leads to loss. It is not clear either, what is the appropriate

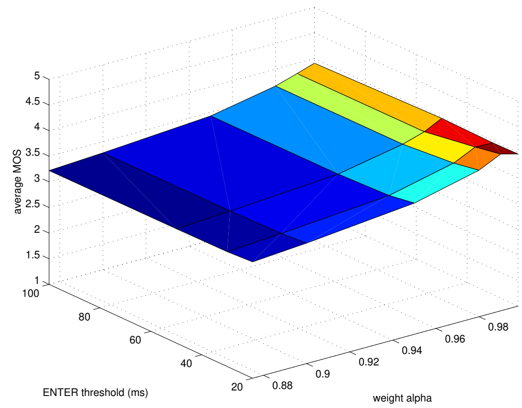


Figure 3.19: Performance of *spike-det* algorithm for a range of its parameters, over the 15 minutes trace of Figure 3.34.

tuning of the weight α: we found that values other than the recommended ones, performed better. Figure 3.19 shows the combined MOS for a range of the parameters over the 15 minutes of Figure 3.34: in the best case, the scheme achieves $MOS \leq 3.9$.

In summary, the problems of the *spike-det* scheme, over our traces, are the following.

1. First and more important, the estimation part of the algorithm does not track the relevant delay components. The backbone paths under study are overprovisioned and thus do not have a slowly moving component to track. As discussed in length in Chapter 2, spikes are not the exception in the delay pattern; they are rather the large majority. As a result, there is sensitivity to the tuning of the parameters, while at the same time there is no good way to tune them.

2. Second, the adaptation part of the algorithm is not meaningful for voice traffic, as also observed in [69]. The scheme leads to loss at the end of a spike, when it tries to track closely the decrease in delay (even when it is not really needed for delays below 150 ms). On the other hand, the TCP-like estimation ($p = 4 + 4v$) overshoots in delay, leading to delay impairment.

3. Finally, the scheme is unable to adjust to spikes that happen during a talkspurt, as it

only adapts at the beginning of each talkspurt.

3.2.4 Considerations in the Design of Playout Scheduling

Having tried the *'Fixed'* and the *'Spike-Det'* algorithms and having experienced limitations and weaknesses with their performance over these backbone networks, let us now revisit the generic adaptive algorithm 1, and define our design objectives that apply to all algorithms. In the next section, we combine these guidelines to design three concrete playout algorithms.[1]

The playout is the interface between the network and the voice application. It hides the delay variability and presents a pattern that the voice application can handle. Therefore there are two considerations to be taken into account. The first is the ***learning*** of the network delay characteristics and the second is the goal that guides the ***adaptation*** (or reaction) part. Learning should be guided by the patterns found in the traces. The adaptation part should be guided by the goal to be achieved, i.e. by the voice quality. The two considerations are closely related, as the knowledge of the trace enables the right adaptation, and vice versa, the goal dictates what characteristics should be monitored and learned.

3.2.4.1 Learning

Why is learning useful?

A playout algorithm should learn the pattern of a trace in order to be able to achieve a certain target level of quality.

Every algorithm has some parameters that require tuning. For example $Fixed$ playout needs to choose the fixed playout delay, based on some knowledge of the trace. The *'Spike-Det'* needs to tune the weight α and the threshold $ENTER$. Similarly, for the three algorithms proposed in later sections. In the absence of any knowledge about the trace, any algorithm has to operate with some arbitrarily chosen parameter. The overall performance

[1] A reader familiar with the design considerations of playout scheduling, could read directly the section on the loss-delay tradeoff 3.2.4.4 and the algorithms sections 3.2.5, 3.2.6, 3.2.7.

is then a result of these parameters and the characteristics of the underlying trace. If the first part of the adaptive algorithm, namely the learning part, monitors and "learns" the delay pattern, then the right parameters can be chosen in order to achieve the goal (e.g. target MOS quality or a target number and size of clips).

What to learn?

We claim that the right delay properties to monitor for the purpose of VoIP playout, are the ***height and frequency of spikes***. The first reason for tracking spike heights and distances is that this is indeed the pattern observed in the traces, as discussed in Chapter 2: spikes of fixed/random heights repeated every fixed/random intervals. Depending on the delay budget, one should only monitor spikes that lead to an end-to-end delay exceeding the perceivable threshold of 150 ms. In other words, it is not useful to monitor spikes much lower than the target playout delay. The second reason why the height of spikes and the distance between them are sufficient properties to learn, is that they allow for explicitly controlling the adaptation in terms of voice-relevant terms. Based on this description, one can specify the acceptable length and frequency of loss durations, and then appropriately tune the algorithm parameters to achieve this target. (This is actually the rationale of Algorithm *'Fast Increase- Exp Decay'* proposed in later section).

A ***moving average***, like the one used in the *'Spike-Det'* algorithm [77], is not appropriate for tracking in these traces due to the simple fact that there is no slow-varying component to track. Even in the rare cases where there is an increase in the mean delay, it can be viewed as a side effect of the change in the spike pattern (height, minimum value and frequency of spikes). Given that spikes are the majority and not the exception in the traces, the spike pattern is the right property to track.

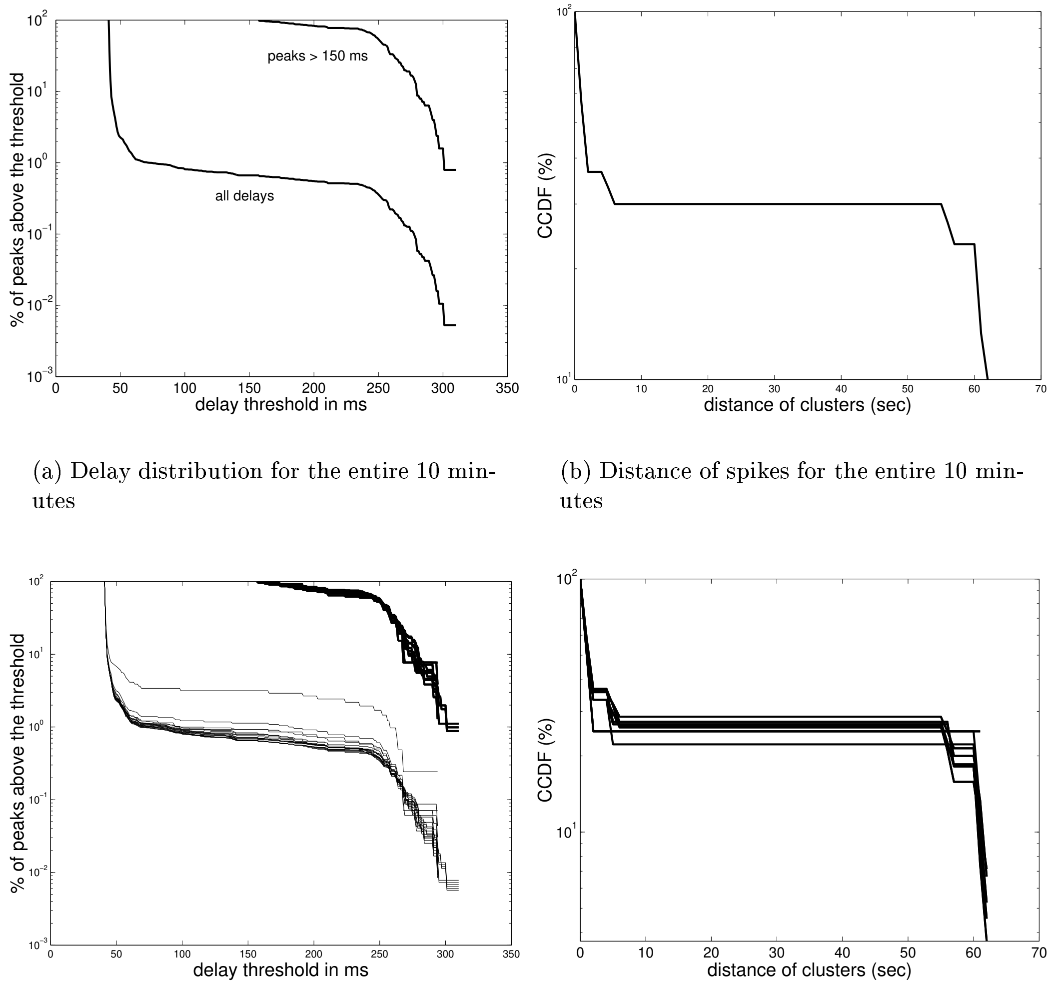
Another commonly used description of the delay behavior is the ***delay distribution***, [69, 64]. There are two problems with this description. First, as we have seen in Chapter 2, the distribution of all delays dilutes the distinct patterns that only higher delays occur. Note also, that for spikes of triangular shape, knowing the peak delays allows us to infer the delays following the peak. Similarly, for spikes with other shapes, characterizing a few parameters of the spike shape is sufficient to infer the remaining delays. Second, the distribution of delays allows for controlling the loss percentage and not the frequency and size of clips in speech; e.g 2% loss evenly spread in small clips over a long period may be imperceptible, while the same number of packets lost consecutively may be unacceptable. That is why, an estimation based on a percentile of the delay distribution, should be accompanied by a short-term mechanism such as the spike detection.

How to learn?

The problem then becomes how to determine the (distributions of) magnitude and frequency of spikes? As an a priori characterization of paths is impossible, learning has to take place as the call progresses. In the beginning a few packets are available; as more packets arrive the distributions can be updated.

If the trace has a ***consistent pattern*** during the entire call, then we can obtain the correct distribution after some period of monitoring. Let us consider the familiar example of a periodic trace, shown in Figure 2.42 (EWR-P_4-SJC, 21:00-21:10). There are cluster of spikes 250-300ms high repeated periodically every 60 seconds. Figure 3.20 (a) and (b) shows the CCDF of the height of spikes above 150 ms and the CCDF of the distance of spike clusters for the entire 10 minutes. Figure 3.20 (c) and (d) show the same distributions considering samples from the beginning of the 10 minute interval until the beginning of each talkspurt. All talkspurts see the same distribution as the entire 10-minutes due to the 60-seconds periodicity of the trace‘. Furthermore, we can also learn a trace with a random pattern by continuously updating the distributions; the distributions will eventually converge to the right ones, provided that the pattern does not change.

However, there are ***traces that change pattern***: delay alternates between states with different magnitude of spikes and distance between them. We saw in Chapter 2 that states



(a) Delay distribution for the entire 10 minutes

(b) Distance of spikes for the entire 10 minutes

(c) Delay as seen at the beginning of each talkspurt

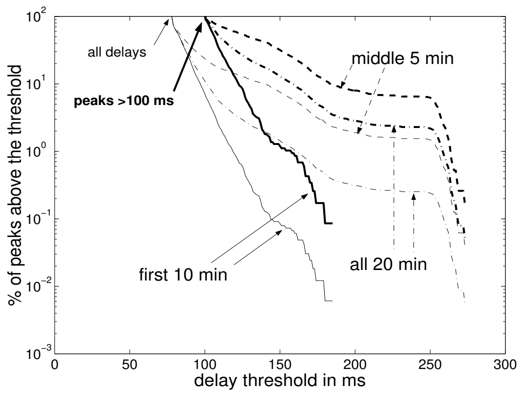
(d) Distance of clusters as seen at the beginning of each talkspurt

Figure 3.20: Learning the periodic pattern of the trace of Figure 3.32 (EWR-P_4-SJC, 21:00-21:10) by continuously updating the distributions for the spikes height and distance.

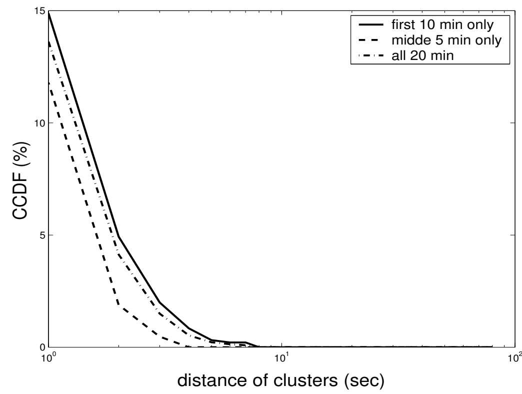
of higher delay last from a few minutes to several hours. For example let us look at the 20 minutes trace of Figure 2.27(a). There are three distinct parts in this trace: 10 minutes with random pattern, 5 minutes with block pattern and 5 minutes with random pattern. Figure 3.21 shows the CCDFs for the height and distance of spikes above 100 ms ((a) and (b)) and above 150 ms ((c) and (d)), considering only the first 10 minutes, only the middle 5 minutes and the entire 20 minutes. In the first part, there are many spikes above 100ms, with roughly exponential height (see Figure 3.21(a)) and close to each other (see Figure 3.21 (b)). There are not many spikes above 150 ms, as reflected in the coarse granularity of Figure 3.21(c) and the large distances in Figure 3.21(d). In the second part of the trace, the block pattern results in many spikes of equal height around 250 ms (in Figures 3.21(a) and (c)), spaced 2-3 seconds apart (see Figures 3.21 (b) and (d)). The third part follows a random pattern similar to the first part, and is not shown in the figure. The distribution for the entire 20 minutes is a mixture of the random and block pattern. For spikes above 150 ms, the behavior of the middle part dominates because there are very few spikes above 150 ms in the first and third part.

In Figure 3.21, knowledge of the entire trace was assumed. Figure 3.22 shows the CCDF of the height and distance of spikes above 150 ms, as the call progresses. Each curve corresponds to delays seen at the beginning of each talkspurt, cumulatively from the beginning of the call. Because the trace changes pattern as the call progresses, the distributions of all samples do not converge, but they are a mixtures of different patterns. Furthermore, it takes time after the actual change of state, for the distribution to change. The ideal behavior of a playout algorithm would be to detect the change in pattern immediately, stop updating the old distributions and start building up new ones. These would result in three separate distributions, one for the first 10 minutes, one for the second 5 and one for the last 5 minutes. Depending on the total end-to-end budget, spikes above 100 ms or spikes above 150 ms might be of interest.

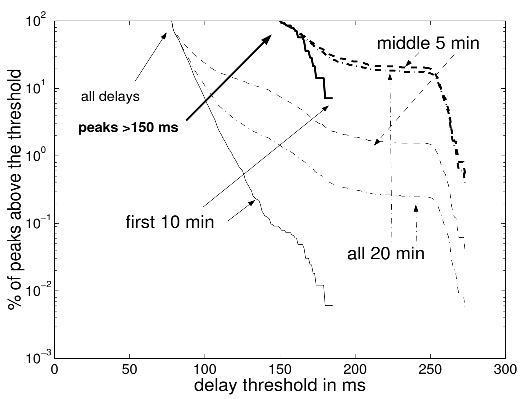
The problem of ***detecting a change*** in a delay pattern is not an easy one to solve in its



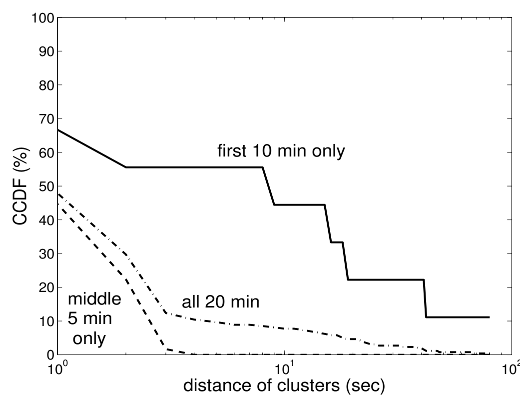
(a) Height of spikes above 100ms



(b) Distance between spikes above 100 ms

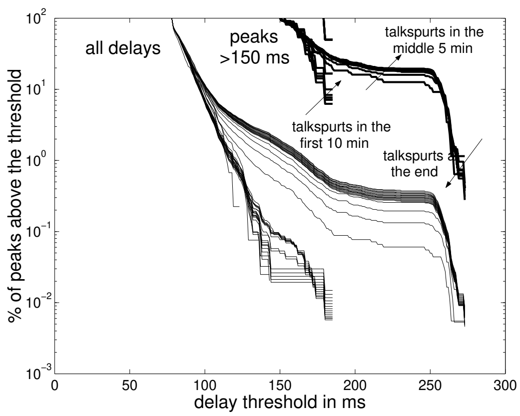


(c) Height of spikes above 150 ms

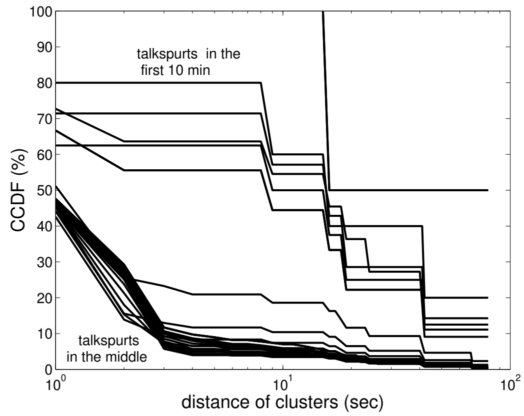


(d) Distance beween spikes above 150 ms

Figure 3.21: Characterizing a trace with changing pattern (THR-P_1-ASH, 14:00-14:20). The first 10 minutes follow a random pattern, the second 5 minutes follow a block pattern and the third 5 minutes follow a random pattern again. For each of the three parts, as well as for the entire 20 minutes, we provide the distribution for the height and distance of spikes above 100 and above 150 ms.



(a) Height of spikes above 150 ms, as seen (cumulatively) at the beginning of each talkspurt



(b) Distance between spikes above 150ms, as seen (cumulatively) at the beginning of each talkspurt

Figure 3.22: Characterizing the example trace with changing pattern (THR-P_1-ASH, 14:00-14:20) at the beginning of each talkspurt.

entirety. Recent work by AT&T Research on the "Constancy of the properties of Internet paths", [105], used statistical tests to detect the change in the median delay, without having a particular application in mind. What would be needed in addition for playout purposes, is a statistical test for detecting changes in the spike heights, distances and shapes. How to define a pattern and how to detect changes in these patterns is a big problem by itself; it has also been studied in other fields, e.g. in pattern recognition. In its general statement, this problem remains open in the scope of this thesis. However, there are heuristics for detecting changes, such as (i) the *spike detection* mechanism and (ii) maintaining a delay history using a *sliding window*. Detecting and following spikes, is a short-term conservative and fast approach. Maintaining a sliding window for the delay history allows for gradually detecting a change in the delay distribution. The larger the window, the longer it takes for the distribution to be affected by new samples, the slower the reaction. That is why the sliding window (long term history) is usually complemented by the faster spike detection mechanism (short term decision). We will see concrete examples in Section 3.2.5.

In summary, learning the loss and delay pattern on a path is a difficult problem in its general statement. From the techniques described above, we use the sliding window mechanism and the spike detection, for the algorithms of sections and 3.2.5 and 3.2.7. Also in the algorithm of section 3.2.6, we assume knowledge of the distribution of the heights and distances between spikes.

3.2.4.2 Adaptation

Goal. Once the estimation part of the algorithm has "learned" the delay in the path, there is a variety of possible reactions that the adaptation part of the algorithm can take. A conservative reaction would consist of adjusting the playout delay upward to always guarantee good speech quality for the remainder of the call, regardless of the effect that this may have on interactivity. A less conservative response would consist of adjusting upward only if the frequency of occurrence of spikes is above a certain threshold. Yet a third approach is to provide the user with the ability to express his or her preference. Unlike the learning part where improvement can be unambiguously quantified by how close the predicted and actual delays are, the adaptation part of an algorithm depends on the goal one tries to achieve. In other words, the appropriate adaptation is not well defined.

Fast Increase to Spikes. As discussed at length in Chapter 2, spikes are the rule and not the exception in the delay patterns in these networks. Every time network delay is higher than the selected playout delay, there is inevitable degradation in speech quality, that can be mitigated to some extend by loss concealment. The algorithm has then to adjust upwards the playout delay. A conservative approach, followed by many algorithms,[64, 69, 77], is to increase the playout delay fast, at the first opportunity after the spike incurs. Indeed, one packet arriving late means that many subsequent packets will also arrive late (bunched up), due to the triangular shape of a delay spike. Furthermore, a spike may be the beginning of a new period of higher delays. For both reasons, a fast increase in p to follow a spike is desirable. Otherwise, a long continuous speech segment may be lost. With spikes ranging from a few ms to 700 ms, just a single spike may cause significant loss.

Note that once a spike higher than the current playout delay occurs, we increase the playout delay at the first opportunity, i.e. at the next talkspurt. However, for packets belonging to the current talkspurt, there is one of the following options: (i) play all packets arriving bunched up and have a gap in the speech (ii) play the packets that arrive in time and have a part of the speech lost (iii) vary the playout rate. The most commonly used option is the second one. We also use this option and consider that late arrival of packets results in loss. The effect of the other two options has not been quantified.

Slow Decay after Spikes. As long as the actual network delays are smaller than the selected playout, one might want to decrease the playout delay, and thus improve interactivity. However, the gain depends on the frequency of the spikes; if spikes happen too often, it might not be worth decreasing the playout delay and soon incur another spike. A conservative approach is to use a hysteresis in decreasing the playout delay, in anticipation of future spikes. This conservative approach has been proved to be useful in the literature (e.g. see the improvement of [69] over [77]) and also useful over the traces under study (e.g. the algorithms performed better for larger decays).

In favor of the slow decaying approach is also the observation that the overall quality ($MOS(loss, delay)$) degrades faster with loss than with delay increase; see Figures 3.10 and 3.11 that show the impairment due to loss and delay separately as well as combined. These graphs capture formally the intuitive observation that good speech quality is the first necessary condition to maintain a good conversation; low interactivity has a second order effect. For example, delays below 150 ms are not even noticed, therefore decreasing the playout below 150 ms brings no benefit at all, at the cost of potential packet loss.

Another reason why a smooth decrease in playout delay is preferable to sudden variations, is the fact that the modification of silence intervals has also its own perceived effect. It has been shown in [29], that the larger the variability introduced in the silence intervals, the more annoying the perceived effect.

In traces with high jitter, there is yet a third reason for using a slow decay. When adaptation happens at the beginning of each talkspurt, the decrease in playout delay, without overwriting previously buffered talkspurts, is limited by the silence intervals. Even in the case that playout delay is continuously adjusted at each packet, there are still limits on the change of playout rate that can be imposed without a perceived effect. A discussion on this topic is provided in the following section.

3.2.4.3 Limits on the Decrease of Playout Delay

In this section, we discuss how the length of silence periods[2] put a limit to how fast we can decrease the playout delay, without "colliding" with previously buffered talkspurts. An increase in the playout delay means that we delay playing a talkspurt, thus increasing its distance from the previous talkspurt; there is no limit in how much we can increase the playout delay. On the other hand, a decrease in the playout delay means that we playout a talkspurt sooner, by shrinking the silence period, i.e. the distance from the previous talkspurt. Therefore, we can decrease the playout delay, without causing loss, at most by the length of the previous silence.

Note that similar arguments hold also in the case that playout delay is continuously adjusted at each packet, as in [64]. There are still limits in how fast one can decrease the delay for each packet without being perceived (e.g. in [64], it is reported that up to 25-30% change in rate is acceptable).

Figure 3.23 demonstrates this problem, which was referred to as "collision", for the first time in [69]. When the decrease in the playout time ($p_1 - p_2$) suggested by the estimation

[2] The durations of talkspurts and silences in VoIP is a topic by itself [50]. Speech has its inherent on/off patterns, reported in [9] to be exponentially distributed with mean 1.2 and 1.8 sec. These patterns are modified by the silence suppression. For example, a long hangover prevents end-clipping but elongates talkspurts; that is why the study in [49] used talkspurts and silence, both 1.5 sec long on average. Modern compression schemes, such as G.729B, adjust dynamically the parameters for silence detection. The resulting patterns are actually more heavy tailed than exponential, as reported early on in [89] (reported average talkspurt 352 ms and average silence 650 ms without hangover) and recently by [50]. ITU-T standard P.59, [44], provides speech samples with talkspurts and silences with average durations 227 ms and 596 ms respectively without hangover, and 1.004 and 1.587 ms with hangover.

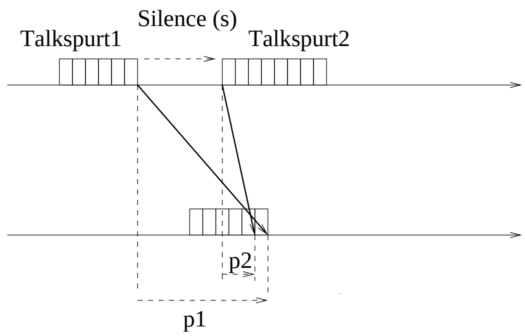
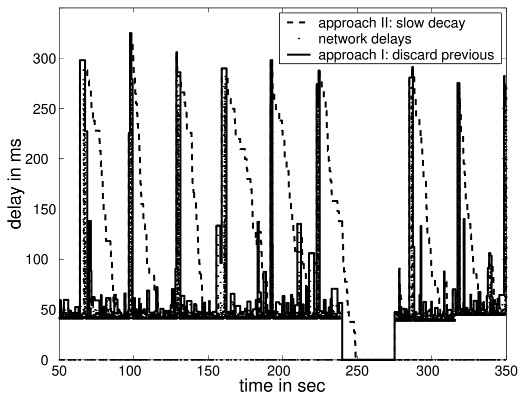


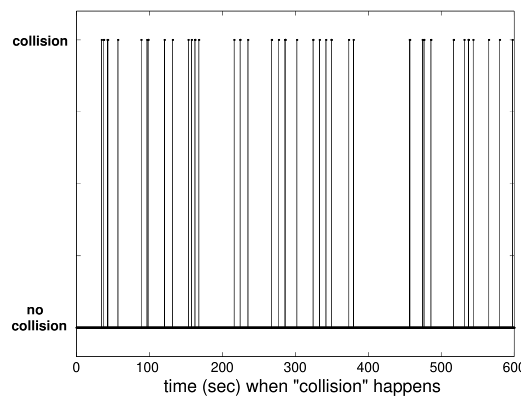
Figure 3.23: Example of "collision": the estimation part of the algorithm suggests a decrease in the playout delay $(p_1 - p_2)$ larger than the silence period (s).

part of the algorithm is larger than the previous silence period s, then there is a "collision" of talkspurts. There are then two options. The first option (I) is to play the new talkspurt $(Talkspurt_2)$ at playout time p_2, discarding the previous overlapping part of talkspurt $(Talkspurt_1)$. This option achieves low delay at the cost of lost speech. The second option (II) is to decrease the playout to $p_2\prime = max\{p_2, p_1 - s\} = p_1 - s$, i.e. no more than what is allowed by the silence interval. Thus, it results in a slow decay p, accumulated delay and a longer duration for completing the same speech sample. An intermediate option would be to sometimes follow approach (I) and sometimes follow approach (II) in a way to optimize some objective function, as it was done in [69] or using another objective function, like the combined $MOS(delay, loss)$.

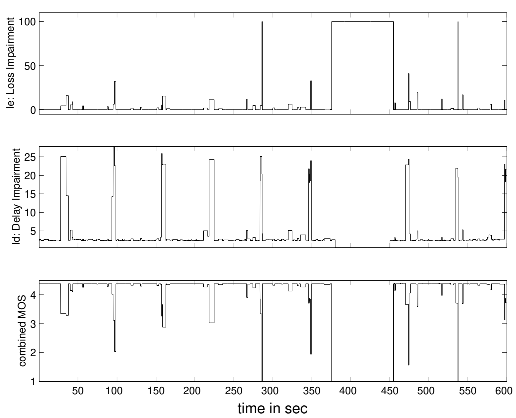
Figure 3.24(a) shows the familiar example trace of 10 minutes on provider P_4, that exhibits the periodic pattern with 250-300ms high spikes every 60 seconds. We artificially considered 1.5sec talkspurts on average and 10 ms fixed size silences and we assumed perfect knowledge of the trace, in order to demonstrate the collisions and the two extreme approaches. Indeed, with so small silence intervals, there are many collisions, as shown in Figure 3.24(b), every time there is an decrease in the delay estimation by 10ms. In reality only, some of these collisions will happen, i.e. whenever the network jitter exceeds the silence interval. Figure 3.24(c) shows the quality achieved by approach (I) which follows exactly the



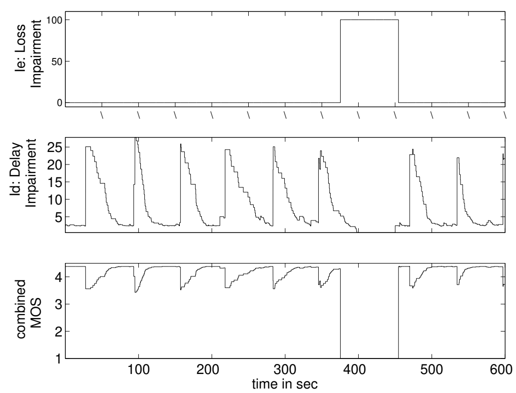
(a) 10 minutes trace and the two extreme approaches for handling collisions



(b) "Collisions" caused by prefect knowledge of the trace and by following it exactly



(c) Instantaneous Quality by approach I:: discarding previous talkspurts



(d) Instantaneous Quality following approach II: accumulating delay

Figure 3.24: Demonstrating the two approaches for handling “collisions” on a 10 minutes of EWR-P_4-ASH (Wed 21:00-21:10)

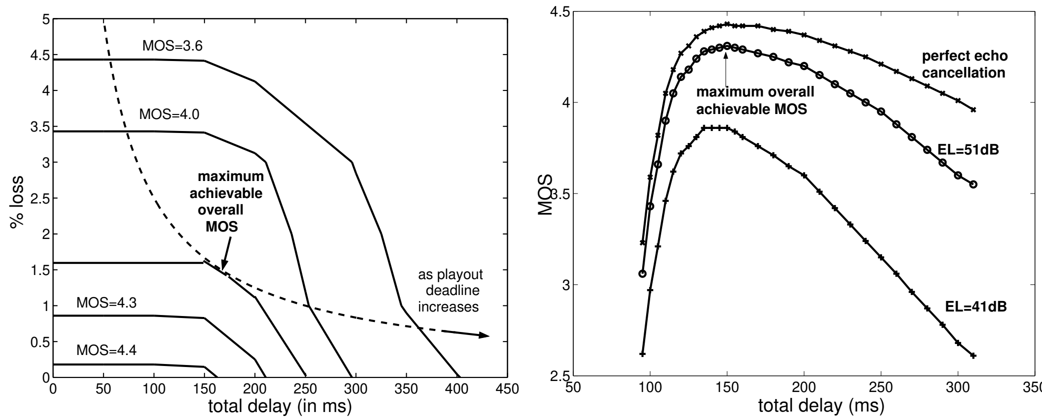
network delay and discards the overlapping parts of previous talkspurts; this way, it keeps average delay low at the cost of loss. Figure 3.24(d) shows the quality achieved by approach (II): no speech at all is lost and collisions result to accumulated delay.

From the possible approaches, we choose to avoid loss due to "collisions" at the cost of additional delay. In other words, whenever a "collision" happens in our simulations, we do not drop any packets and, instead, we play the next talkspurts with accumulated delay. This choice also serves the objectives of the previous section, e.g.. the slow decay of playout delay for conservative estimation.

3.2.4.4 Loss-Delay Tradeoff

As discussed in Section 3.1.3, the overall perceived quality is a combination of delay and loss, see Figure 3.11. Playout scheduling controls both quality aspects. Choosing a playout delay p, determines both the e2e delay and the loss percentage (as packets that arrive after their deadline, i.e. with $n_i > p$, are dropped). As we increase the playout delay for a specific trace, we obtain certain $(loss, delay)$ pairs, shown in Figure 3.25(a) in dashed arrow, which result in a certain overall quality $MOS(loss, delay)$. Larger playout delay means better speech quality (less packets are dropped due to late arrival) at the cost of higher delay. The margin between the intrinsic quality of an encoder and the maximum acceptable degradation, can be used to improve either or both quality aspects. Clearly, there exists a trade-off between loss and delay that leads to a maximum MOS value. In Figure 3.25(a), this happens when the dashed line touches the maximum MOS contour. Further increase in delay decreases the overall MOS, because the delay penalty exceeds the gain in speech quality.

There is only one free control parameter in playout scheduling, namely the playout delay p; the loss rate, due to late arrival, is a side-effect of choosing p. Increasing the playout delay p results in the overall quality shown in dashed arrow in Figure 3.25(a), or equivalently to the $MOS(delay)$ curve in Figure 3.25(b). More specifically, the $MOS(delay)$ curve that corresponds to the dashed line is the middle one, obtained for $EL = 51dB$. The maximum



(a) Voice quality contours

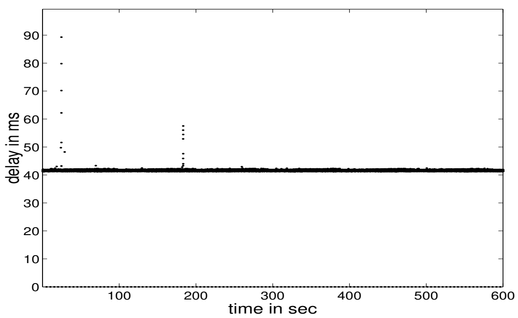
(b) Overall quality as a function of the total end-to-end delay

Figure 3.25: Voice quality and how the playout determines the achievable quality range and maximum value. The contours correspond to G.711 with concealment, 10 ms packet, bursty loss pattern, free conversation, echo loss EL=51 dB. For a specific trace, the choice of playout delay determines the (loss,delay) pair and thus the overall MOS.

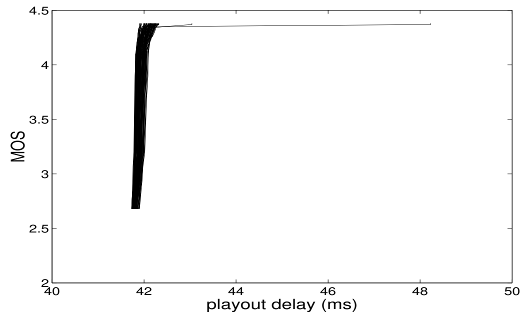
achievable overall $MOS(loss, delay)$ on the dashed line, corresponds to the maximum value of the $MOS(delay)$ curve.

The MOS contours, first appeared in Figure 3.11 and repeated in 3.25(a), do not depend on the trace; they give the combined R or MOS, for all the possible combinations of loss and delay, adding the impairments using Equation (3.1). They only depend on the I_d and I_e curves (see Figure 3.10), i.e. on how severe is the effect of loss on the specific compression scheme and how severe is the effect of delay on interactive conversations or/and in the presence of echo. In Figure 3.25(b), the three curves correspond to different delay impairments due to different echo levels.

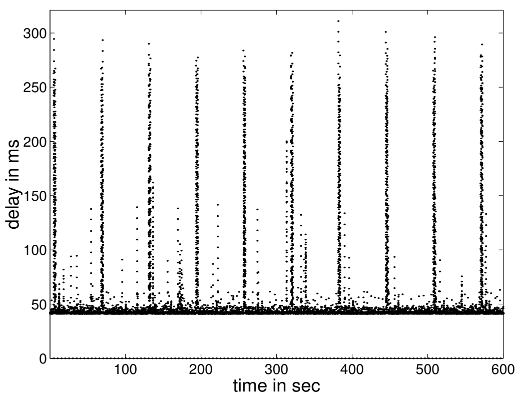
On the other hand, the achievable $(loss, delay)$ pairs, shown by the dashed line in Figure 3.25(a) and translated to the curves of Figure 3.25(b), do depend on the specific trace and its delay distribution. In Figure 3.26, we show the loss-delay tradeoffs for three different traces computed for consecutive five seconds intervals. Every $MOS(delay)$ curve shown at



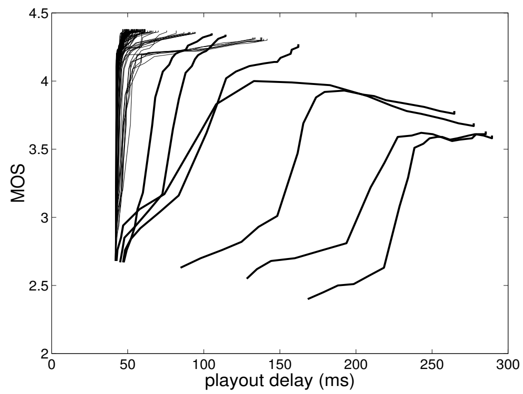
(a) A trace with very low delay variability: ASH-P7-SJC, Wed 3:00-3:10



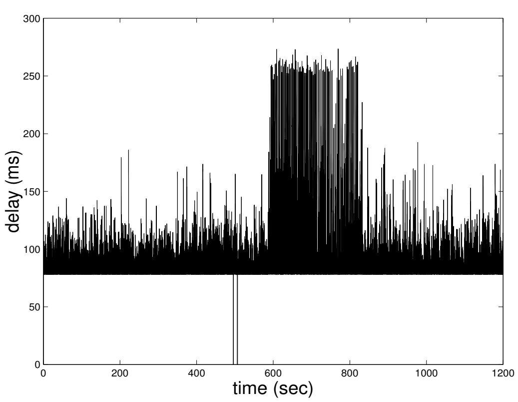
(b) Loss-delay tradeoff for trace ASH-P7-SJC, Wed 3:00-3:10



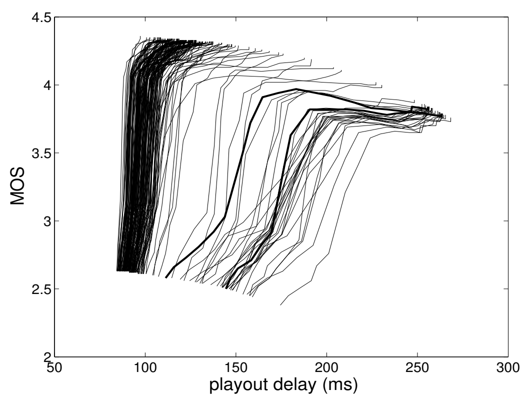
(c) A trace with periodic patern: EWR-P4-SJC, Wed 21:00-21:10



(d) Loss-delay tradeoffs for trace EWR-P4-SJC, Wed 21:00-21:10



(e) A trace with high delay and delay variability: THR-P1-ASH, Wed 14:00-14:20



(f) Loss-delay tradeoffs for trace THR-P1-ASH, Wed 14:00-14:20

Figure 3.26: $MOS(delay)$, computed in 5 seconds intervals, for three different traces

the right, corresponds to a different five seconds interval of the trace shown at its left. In Section 3.2.7, we use these observations and design a playout algorithm that achieves the maximum MOS value of the $MOS(delay)$ curve.

The first trace (Figure 3.26(a)) corresponds to 10 minutes on a path of provider P_7, with low delay (40.5 ms) and very low variability (in the order of 2 ms). All the $MOS(delay)$ curves are strictly increasing, indicating that delay is so low that there is no benefit in following it closely; the playout delay should be set above 50ms, so that no packets are dropped. The second trace corresponds to 10 minutes on a path of provider P_4 and exhibits the periodic pattern discussed in Section 2.4.7 and repeated in Figure 3.26(c). During the periods between the high delay clusters, delay is low, the $MOS(delay)$ curves are strictly increasing, indicating that the higher the playout delay, the higher the overall quality. The few periods with high delay, correspond to the bottom-right $MOS(delay)$ curves in Figure 3.26(d). These have a maximum value which indicates that the optimal choice of playout delay trades loss for delay. The third trace corresponds to 20 minutes of provider P_1, discussed earlier in Figures 2.27(a) and 3.21. High delay (the fixed part is 77.8ms) and high delay variability lead to the trade-offs between loss and delay, shown in Figure 3.26(f), that allow for MOS optimization, especially during the middle 5 minutes that delay is higher.

Having discussed in a general way issues concerning (i) learning the trace (ii) delay adaptation and (iii) the loss-delay tradeoff in the overall perceived quality, let us now apply these principles to design three concrete algorithms, each aiming at a different level of loss and delay.

3.2.5 Conservative Mode: Assisted Max Fixed

Algorithm 2 is in fact the Fixed Playout algorithm, where the choice of fixed value is assisted (adjusted infrequently) by learning the trace. The algorithm maintains a sliding window (W) of the most recent delays and adjusts the playout at the beginning of a talkspurt, conservatively, to the maximum delay experienced during this window. The adaptation to

Algorithm 2 ConservativeAlgorithm: 'Assisted Max Fixed'

Initialization:

- *Choose the window W for delay history and the initial playout delay p.*

For every packet received:

- *calculate the network delay*
- *update the delay history (histogram of last W received packets)*

At the beginning of a talkspurt adjust the delay if needed:

- $p = max_{over\ last\ W\ packets}\{delay\}$
- [optional: $p = max\{p, 150ms\}$]

---

a high delay happens as fast as allowed by our opportunity for adaptation, which is only the beginning of a talkspurt. The effect of a high delay lasts for a long time, i.e. as long as the sliding window W. The larger the window the more conservative the algorithm, as it takes longer to forget a high delay. Thus, large windows achieve low loss at the cost of (perhaps unnecessarily) high delay. In that case, the algorithm behaves like a conservative fixed playout, which infrequently adjusts the fixed value, assisted by learning the trace. It is more appropriate for non-interactive applications because it aims at good speech quality even at the cost of high delay. At the other extreme case where the window becomes very small, the algorithm becomes myopic and follows the short-term variations of delay, with the potential of loss.

In Figure 3.27, we run this algorithm over the example trace that we also used for the fixed and the 'spike-det' algorithm, for a range of windows from 2 to 200 sec. We consider only $W > 2sec$, i.e. at least as long as our adaptation opportunity interval (which is one talkspurt, 1.5 sec long on average).

Figure 3.27(a) shows the 20 minutes of the trace and the playout delays chosen for $W = 10$ and $W = 100$ sec. Figure 3.27(b) shows the loss rates and the burst loss rates,

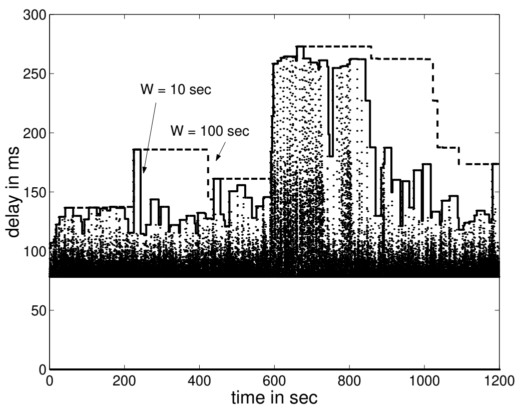
Table 3.3: Algorithm *'Assisted Max Fixed'* over 20 minutes of trace P_1 (THR-P_7-ASH, Wed 06/27/01, 14:00-14:20), for a range of windows W. The statistics presented refer to the entire 20 minutes duration.

W (sec)	% packets lost	number of clips > 60ms	talkspurts affected	average delay	average MOS
2 sec	1.63%	28	22	139 ms	4.03
5 sec	0.59%	8	8	154 ms	4.16
10 sec	0.29%	5	5	164 ms	4.18
20 sec	0.17%	2	2	173 ms	4.17
30 sec	0.14%	1	1	178 ms	4.15
50 sec	0.12%	1	1	186 ms	4.15
100 sec	0.10%	1	1	200 ms	4.10
200 sec	0.09%	1	1	218 ms	4.03

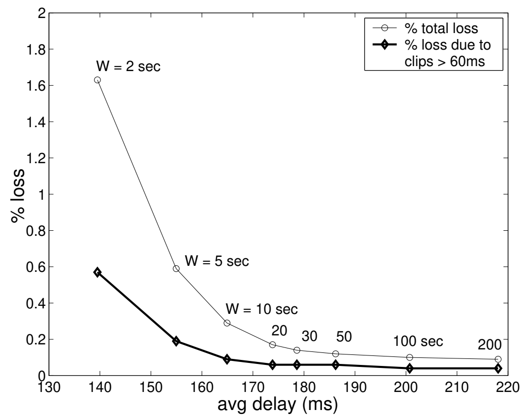
computed over the entire 20 minutes, resulting from the Assisted Max Fixed algorithm when using different values for the window W. We can see that small loss rates are achieved at reasonable delay value, for the entire range of W. Figure 3.27(c) shows how loss is distributed across the 20 minutes, when a window of 10 sec is used. The loss rate, the maximum loss duration and the resulting speech quality (MOS) is plotted per talkspurt, considering 1.5 sec long talkspurts. We see that loss rates are low not only across the 20 minutes but also per talkspurt. Also loss durations above 60ms, which are difficult to conceal, are rare. The resulting speech quality is good for all talkspurts except for a few "transition" talkspurts, during which the playout needs to be adjusted upwards. Figure 3.27(d) shows the same measures as Figure 3.27(c), but for a larger window $W = 100sec$.

Both Figure 3.27(b) and Table 3.3 show that the Assisted Max Fixed algorithm performs well for the entire range of W in terms of loss rate and burst loss (loss durations above 60 ms), average delay and overall MOS. This lack of sensitivity is due to the fast conservative adaptation to spikes: spikes are immediately detected, because they affect the maximum value.

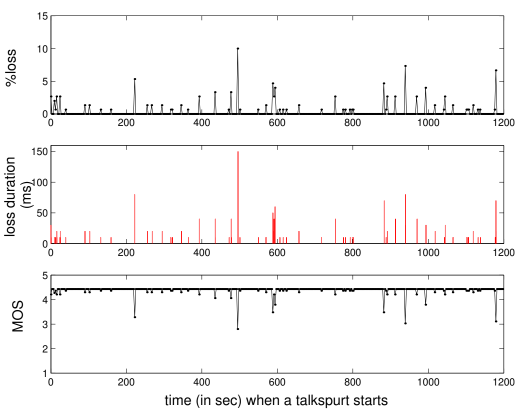
On the contrary, a scheme that adjusts the playout to less than the maximum, would take longer to notice the effect of higher delays. Let us consider a scheme 'Assisted 99%



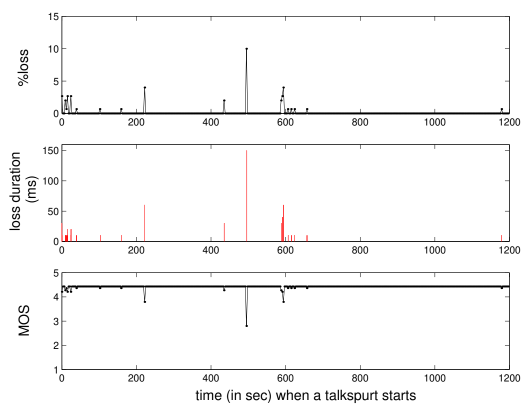
(a) Playout, using 2 different window sizes: 10sec and 100sec



(b) Loss-delay achieved using a range of windows



(c) Speech quality for W = 10sec



(d) Speech quality for W = 100 sec

Figure 3.27: Algorithm 'Assisted Max Fixed' over 20 minutes of THR-P1-ASH, Wed 14:00-14:20, performs well for two different values of the window W.

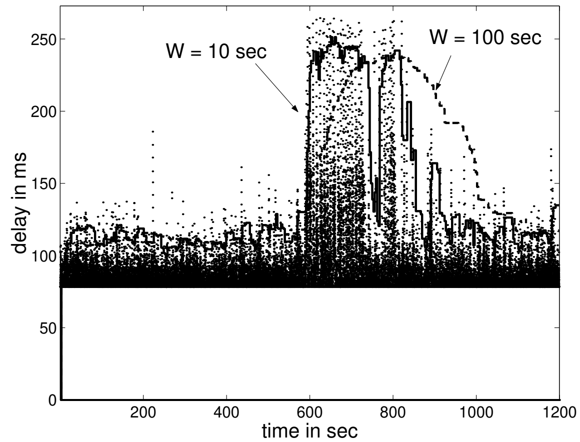


Figure 3.28: Effect of the window length, on the *'Assisted 99% Fixed'* algorithm. The algorithm adjusts p, at the beginning of each talkspurt, to the 99% (instead of the maximum) of the delays over the last W sec (sliding window).

Fixed', that is a scheme which adjusts p to the 99^{th} (as opposed to the maximum) of delays over the sliding window W, at the beginning of each talkspurt. As it is shown in Figure 3.28, it takes a long time before the scheme notices the increase in delay. The larger the window W, the less responsive to changes the scheme becomes. The 'Assisted 98% Fixed' would be even less responsive to changes.

Figure 3.29 shows the loss percentage and the average delay achieved by the variants of the Assisted Fixed algorithm, for a range of windows W from 2 to 200 sec. We can make the following observations based on this figure. First, the 'Assisted Max Fixed' algorithm outperforms all other variants, in the sense that it achieves lower loss at similar, or at least not significantly higher, delay. Second, the curves corresponding to the 98^{th} and 99^{th} percentiles variants have a convex shape. As window increases, spikes are remembered for longer; on the other hand, as window increases, the 98^{th} and 99^{th} percentiles remain low and are not affected by infrequent spikes. The lowest achievable loss rate is still high, compared to the Assisted Max Fixed scheme, and very sensitive to the choice of W. Third, we can see that adding a spike detection (i.e.: "if there is an increase $n > p$ then $p = max$, otherwise $p = 99\%$ of delays in the past W sec") to the 'Assisted 99% Fixed', makes the adaptation

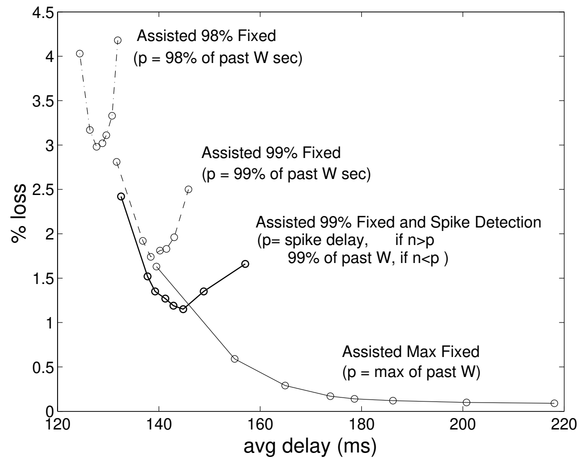
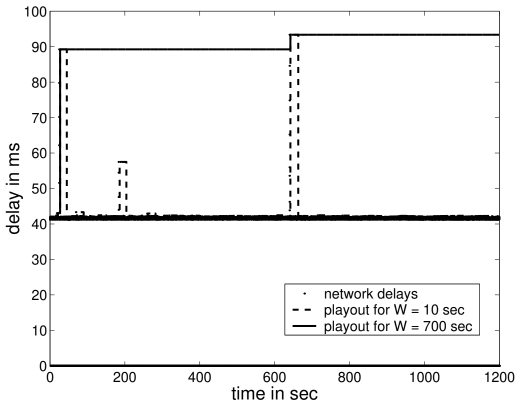


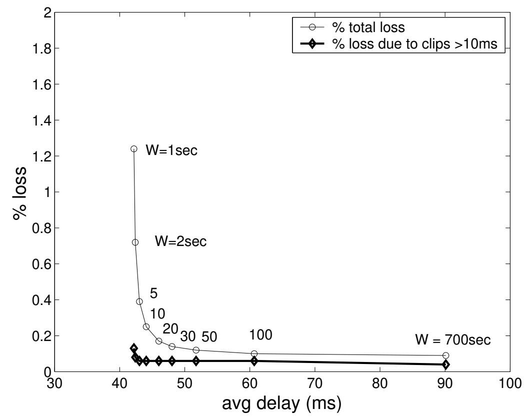
Figure 3.29: Effect of the window size on the variants of the *'Assisted Fixed'* algorithms: (i) Assisted Max Fixed (ii) Assisted 99% Fixed (iii) Assisted 99% Fixed with Spike Detection (iv) Assisted 98% Fixed. Each curve has been obtained by varying the window length (from left to right: 2, 5 ,10, 20, 30, 50, 100, 200 seconds have been considered). The loss percentage and the average delay have been computed over the entire 20 minutes.

faster and improves the performance. However, even then, it is still worse than the 'Assisted Max Fixed'. This is because 'Assisted 99% Fixed' activates the spike detection only at the beginning of a talkspurt, while high delays in the middle of a talkspurt still need time to affect the 99^{th} percentile and be noticed.

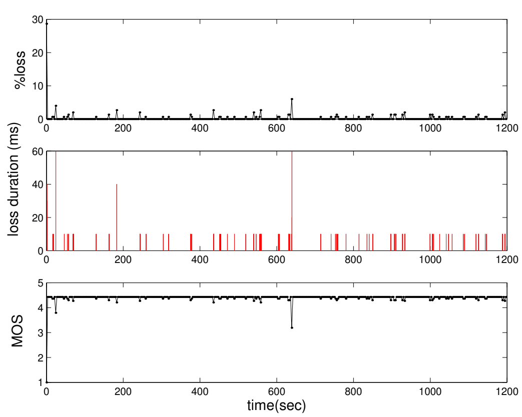
The performance of the 'Assisted Max Fixed' algorithm over a trace with very low delay and delay variability is shown in Figure 3.30. The four graphs show similar measures as Figure 3.27. Note that the trace with low variability is the same example trace of Figure 3.26(a) extended by 10 minutes. The performance of the 'Assisted Max Fixed' algorithm is good for a large range of window W, from 1 sec to 1000 sec. Indeed, negligible loss rates are achieved at very low delay, when computed over both the entire 20 minutes and per talkspurt. The loss durations are mainly 10ms (1 packet only) and thus they can be easily concealed at the receiver. No loss durations exceed 60 ms. The average MOS is high both at the end of the 20 minutes call (e.g. 4.22 for $W = 1sec$ and up to 4.37 for the other values of W) and computed per talkspurt.



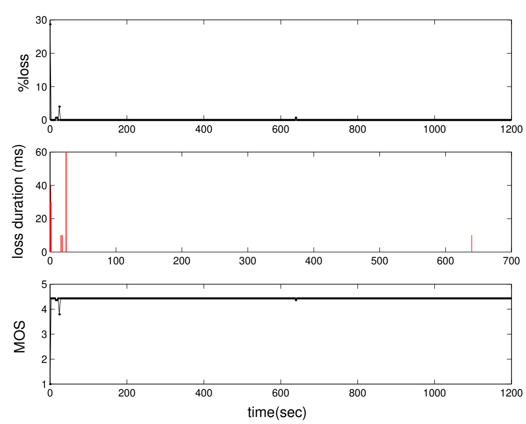
(a) Playout using two different window sizes: 10sec, 700sec



(b) Loss-delay pairs achieved using a range of windows



(c) Speech quality for W=10 sec



(d) Speech quality for W=700sec

Figure 3.30: Performance of 'Assisted Max Fixed' over 20 minutes of ASH-P7-SJC, Wed 3:00-3:20.

Yet a third example is the periodic trace of Figure 3.26(c): clusters of 250-350 ms high delays occur every approximately 60 seconds. Once the window W exceeds the 60 seconds period, there is practically no loss. The performance of the 'Assisted Max Fixed' algorithm, compared to other algorithms over the same trace, will be shown in the later Comparison section; see Figure 3.39.

3.2.6 Configurable Mode: Fast Increase and Exp Decay

Algorithm 3 'Fast Increase - Exp Decay' algorithm

Initially: $p = p_{predicted} = n_1 + safety$
Estimation: *(for every packet i received)*

- *calculate the network delay* n_i
- *if* $n > p_{predicted}$ *(SPIKE mode - increase fast)*
  - $p_{peak} = n_i$
  - $index_{peak} = i$
  - $p_{predicted} = p_{peak} + safety$
- else (NORMAL mode, decrease slowly)
  - $time = i - index_{peak}$
  - $p_{predicted} = p_{peak} \cdot exp(-time/T_{decay})$

Reaction

- *at the beginning of a talkspurt:* $p = p_{predicted}$
- *in the middle: keep the previous p*

The previous algorithm followed a conservative approach: it increased fast at spikes and it remembered the effect of high spikes for a long time, by maintaining a high playout delay. For traces with infrequent spikes, it might be desirable to decrease the playout delay in between. Given a certain trace, there is a range of reactions, from conservative to risky, depending on the relative importance of speech quality and delay for a user/application. The algorithm proposed in this section, follows all the design principles of Section 3.2.4 and, in addition, can provide the entire range of reactions by appropriate tuning of its parameters. Thus the name "configurable mode".

Let us describe the algorithm, called '*Increase Fast - Exp Decay*' using the example trace in Figure 3.31. For every received packet, the algorithm updates an estimate $p_{predicted}$ of the network delay (n), shown in dashed line in Figure 3.31. Whenever there is an increase in network delay ($n > p$), the estimate increases rapidly to $p_{predcicted} = p_{peak} + safety$ in order to prevent long speech clipping. Whenever there is a decrease ($n < p$), the estimate decreases, in order to achieve low delay; however, the decrease is slow in anticipation of future spikes. The shape of the decay function is not critical; we chose an exponential decay (T_{decay}) but we could have chosen other shapes too, e.g. linear. The playout delay (p) is adjusted to reflect the estimate ($p = p_{predicted}$) only at the beginning of a talkspurt (shown in solid line in Figure 3.31); however if techniques like those in [64] are applied, p could be adjusted at arbitrary times.

Let us now discuss the two design considerations, i.e. learning and appropriate reaction, based on the above example. In the absence of any knowledge about the trace, Algorithm 3 has to operate with some arbitrarily chosen parameters T_{decay}, $safety$. The overall performance is then a result of these parameters and the characteristics of the underlying trace. If the pattern is known, then T_{decay} and $safety$ can be tuned to achieve a certain frequency and size of clips in speech.

A first example is the trace of Figure 3.31. Table 3.4 shows the performance of Algorithm 3 for different T_{decay} and $safety$ values. This trace has 250 ms high spikes every 2-3 seconds: the block pattern discussed in Figure 2.27(b). Therefore high playout delay (i.e. high T_{decay} and $safety$) leads to better overall MOS.

A second example is the trace of P_4 that has a perfectly periodic pattern. Figure 3.32 and Table 3.5 show the performance of Algorithm 3 for a range of T_{decay} and for $safety = 0$. We can see that the choice of parameters strongly affects the overall performance; for example

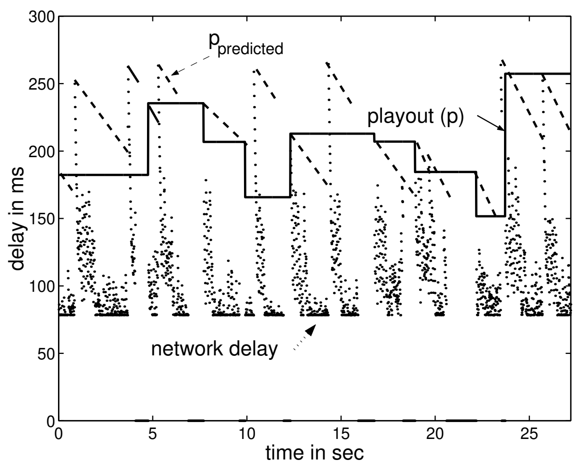


Figure 3.31: Example of Algorithm *'Fast Increase-Exp Decay'* over a 28 sec trace (THR-P1-ASH, Wed 06/27/01, 14:00)

Table 3.4: Algorithm *'Fast Increase - Exp Decay'* over 28 sec of trace P_1 (THR-P_1-ASH, Wed 06/27/01, 14:00 (UTC)

T_{decay}	$safety$	% packets lost	number of clips > 60ms	talkspurts affected	mean delay	mean MOS
0.5 sec	0	25.8%	19	6	135 ms	2.45
1 sec	0	24.9%	19	6	141 ms	2.58
2 sec	0	17.4%	9	5	150 ms	2.79
5 sec	0	13.1%	5	3	167 ms	3.46
10 sec	0	10.9%	3	2	186 ms	3.71
30 sec	0	10.1%	2	1	213 ms	3.78
100 sec	0	9.8%	2	1	226 ms	3.79
0.5 sec	50 ms	11.1%	10	6	153 ms	2.98
1 sec	50 ms	9.8%	10	6	162 ms	3.22
2 sec	50 ms	9.7%	5	4	165 ms	3.24
5 sec	50 ms	6.9%	5	3	176 ms	3.61
10 sec	50 ms	4.8%	3	2	194 ms	3.87
30 sec	50 ms	3.9%	2	1	221 ms	3.94
100 sec	50 ms	3.7%	2	1	234 ms	3.95

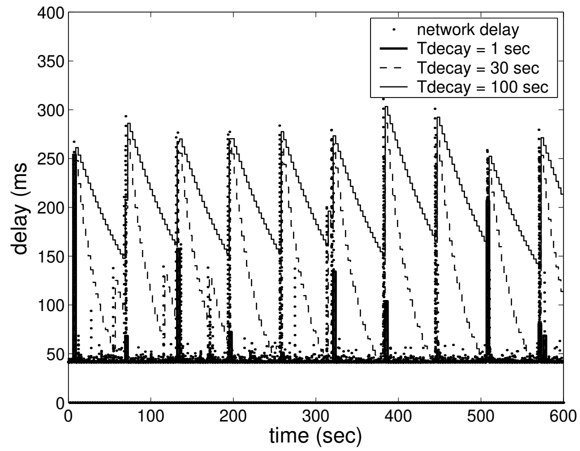


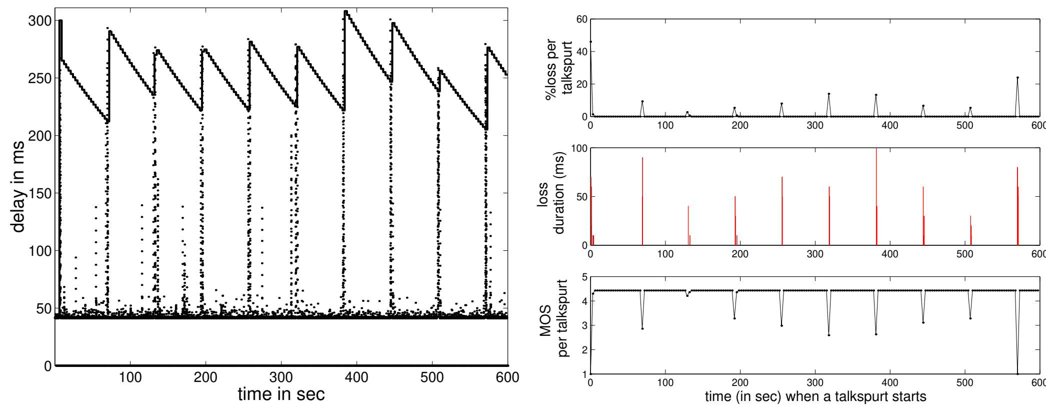
Figure 3.32: Algorithm *'Fast Increase - Exp Decay'* over 10 minutes of trace P_4 (EWR-P_4-SJC, Wed 06/27/01, 21:00-21:10), for various decay functions.

Table 3.5: Algorithm *'Fast Increase - Exp Decay'* over 10 minutes of trace P_4 (EWR-P_4-SJC, Wed 06/27/01, 21:00-21:10), for a range of $Tdecay$ and $safety = 0$

T_{decay}	% packets lost	number of clips > 60ms	talkspurts affected	average delay	average MOS
1 sec	53.3%	546	129	45 ms	1.32
2 sec	49.4%	511	120	48 ms	1.52
5 sec	42.7%	447	234	59 ms	1.96
10 sec	25.1%	234	60	75 ms	2.85
20 sec	7.2%	69	24	103 ms	3.82
30 sec	2.8%	33	13	128 ms	4.04
50 sec	2.1%	32	11	165 ms	4.05
100 sec	1.3%	27	11	210 ms	3.96

the MOS in the last column of Table 3.5 varies from 2.45 to 3.79.

The pattern of this trace is perfectly periodic: 250-300 ms high, lasting for 3 sec and repeated every 60 sec, as we can see visually in Figure 3.32 and as the distributions of Figure 2.43 describe. If the algorithm knew the underlying pattern, it could easily choose the adaptation parameters to achieve a desired quality level. For example, we could choose to increase the delay every 60sec and avoid completely any speech clipping. A second option would be to explicitly control the desired number and size of clips per second at the cost



(a) Playout for Tdecay=267ms

(b) Resulting speech quality

Figure 3.33: Achieving a target clips frequency/durations on the trace of Figure 3.32, by tuning the parameters of *'Fast Increase - Exp Decay'*.

of an average delay, dictated by the trace. For example, by choosing $T_{decay} > 270sec$[3] and by increasing p to $max\{n, 300ms\}$ for large spikes, and to $max\{n, 150ms\}$ for high spikes, we can make sure that no more than 60 ms speech is clipped every 60 sec, as shown in Figure 3.33. Speech quality will then be good (60 ms clipping can be further concealed at the receiver) for an average delay of 250 ms. Yet a third option would be to calculate the overall MOS for all T_{decay} values and choose T_{decay} that leads to the maximum MOS in Table 3.5.

Similarly for a trace with a random pattern, if we know the distributions for the spike heights and frequency, we can appropriately tune the parameters to achieve a target quality level. For example, we could set the delay high enough to achieve no loss at all. If the delay range is below 150 ms, then by choosing fixed playout at 150 ms, we can achieve perfect speech quality at no interactivity cost. Or we could tune T_{decay} and $safety$ to achieve guarantees for a desired speech quality at the cost of a certain average delay, similarly to the periodic trace. The only difference from the periodic trace would be that the guarantees

[3]Indeed, by solving $p_{peak} - p_{peak} \cdot exp(-period/T_{decay}) = 60ms$ where $p_{peak} = 300ms$ and $period = 60sec$ we find $T_{decay} = 268sec$.

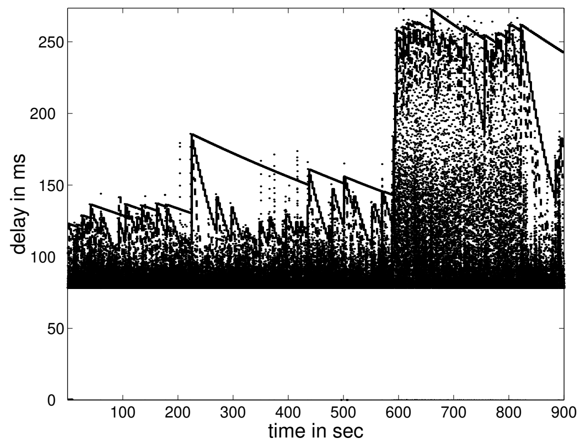
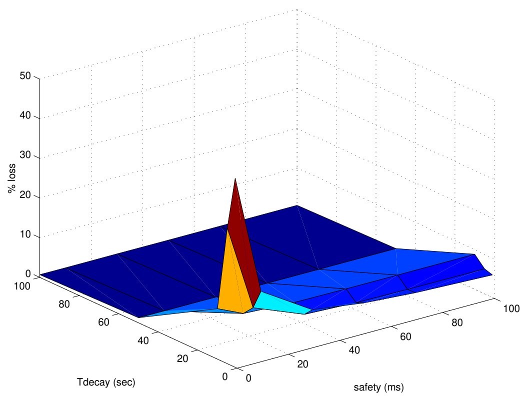


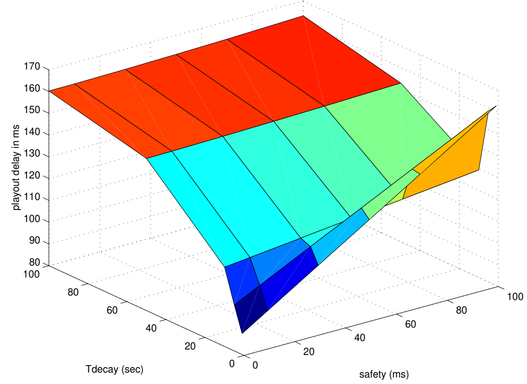
Figure 3.34: Algorithm *'Fast Increase - Exp Decay'* over 10 minutes of trace P_1 (THR-P_1-ASH, Wed 06/27/01, 14:00-14:15 UTC), for different decay functions

would be probabilistic for the random pattern. Figure 3.34 shows 15 minutes of a P1 trace and the playout for 3 different decays. The first 10 minutes have the random pattern described in Figure 2.25. The last 5 minutes have the block pattern of Figure 2.27(b). The 28 sec of Figure 3.31 happen at the beginning of these 5 minutes. Figure 3.35 shows the effect of T_{decay} and $safety$ on the performance. It seems that a high playout delay (resulting from high values for the T_{decay} and $safety$ parameters) leads to better overall performance. Furthermore, the best tuning of these parameters is different for the first ten and the last five minutes.

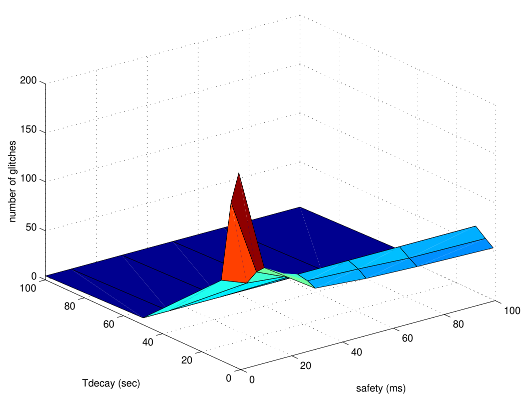
In summary, in this section we experimented with tuning the 'Fast Increase-Exp Decay' algorithm over various example traces. We saw the effect of these parameters on the performance over a P_1 (Figure 3.31 and Table 3.4 for a 30 seconds trace, Figure 3.34 and 3.35 for 15 minutes trace) and over a P_4 (Figure 3.32 and Table 3.5) trace. In Figure 3.33 we tuned the algorithm to achieve a given loss size and frequency over the example P_4 trace. For such a tuning, we first need to learn the trace, in ways discussed in Section 3.2.4.1. In the same section, we discussed why the problem of formally learning an arbitrary trace is a difficult one and it remains an open problem in the context of this thesis. This algorithm uses only

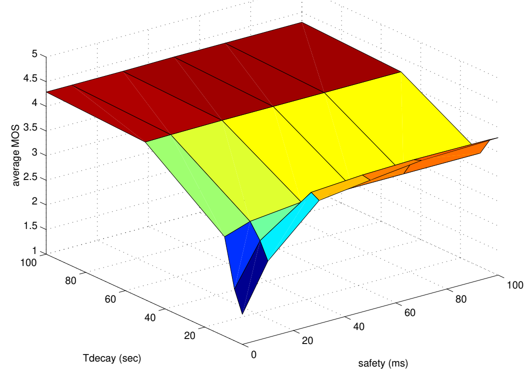


(a) Packet loss rate



(b) mean delay





(c) overall MOS for the entire call

Figure 3.35: Effect of the parameters on the performance of *'Fast Increase - Exp Decay'* algorithm over the 15-minute trace of Figure 3.34

a heuristic of exponentially decaying memory for learning the trace. In the absence of an accurate knowledge of the delay characteristics (i.e. the distribution of spike heights and frequency) a conservative choice is to choose a slow decay function, in which case it becomes like the previous conservative algorithm (Assisted Max Fixed).

3.2.7 Intermediate Mode: Maximize $MOS(loss, delay)$

Algorithm 4 Intermediate Algorithm that maximizes $MOS(loss, delay)$

Initialization:

- *Choose parameters: window W for delay history, candidate percentiles $\{q_1, ...q_k\}$, thresholds for spike detection*
- *Choose initial playout delay p*

For every packet i received:

- *calculate the network delay n_i*
- *decide whether we are in NORMAL or SPIKE mode*
  - *if $n_i/p > ENTER$ then mode=SPIKE, $d_{1st} = n_i$*
  - *if $n_i/d_{1st} < EXIT$ then mode= NORMAL*
- *update the delay history (histogram of last W received packets)*

At the beginning of a talkspurt:

- *If in SPIKE mode, then increase fast:*
  - $p = max\{history, n_i\}$
- *else decrease slowly:*
  - *for every candidate percentile $q \epsilon \{q_1, ..., q_k\}$*
    - $I_d(delay) = I_d(q)$
    - $I_e(loss) = I_e(late\,loss\ (100 - q)\% + network\,loss)$
    - $MOS(q) = R2MOS((R_0 - I_s) - I_d - I_e)$
  - *among them, choose p = \{q% of past W delays to maximize $MOS(loss, delay)$\}*

Description

The Algorithm proposed in this section is called 'Maximize MOS'. It is similar to the previous one (*'Increase Fast-Decay Exp'*), in that it follows the principles discussed in Section 3.2.4. Indeed, the playout increases fast upon spike detection and decreases slowly afterwards. The previous algorithm left the choice of how fast to decay to the "user", i.e. to the tuning of the T_{decay} and $safety$ parameters. This algorithm uses the delay-loss tradeoff discussed in Section 3.2.4.4 to decide on how fast to decay after a spike. In particular, among all possible playout delays, the 'Maximize MOS' algorithm chooses the one that maximizes the combined MOS, see Figure 3.25. This is an intermediate adaptation approach (in the sense that it keeps both delay and loss low, to maximize the overall MOS), better suited for interactive applications, than e.g. *'Assisted Max Fixed'*, as it achieves lower delay.

In order to capture the delay history, a sliding window (i.e. the last W received packets) is used and the histogram of the delay is computed. The algorithm decreases the playout delay p to be equal to the q^{th} delay percentile (computed over the W most recent delays) that maximizes $MOS(delay)$ in Figure 3.25.[4] We use a histogram for the delay history, because it allows for simultaneously capturing both the playout delay (exactly equal to the q^{th} percentile) and the loss percentage ($(100-q)\%$ lost due to late arrival) and thus allows us to calculate $MOS(delay, loss) = MOS(q)$ in a straightforward way.[5] There have been schemes in the past, such as [76, 78, 69, 64], that allow to specify a target loss rate. The difference in this algorithm is that it adjusts this target loss rate (i.e. the delay percentile that results in late arrivals) dynamically throughout the call, in order to maximize $MOS(q)$. For example, the optimal choice for the traces in Figures 3.26 (d) and (f) is different from

[4] Ideally, we would like to perfectly know the delays during the coming talkspurt, at the beginning of which we adapt p. The coming talkspurt is the one affected by the choice of playout. Then, we would choose $q_{opt}\%$ to maximize $MOS(delay)$ over that coming talkspurt. However, in reality we do not know the future delays; we can only believe that they will follow the same distribution as in the recent past. Clearly, the optimal choice over the last W packets, will not be necessarily optimal during the coming talkspurt.

[5] We first calculate $R = (R_o - I_s) - I_e - I_d + A$, which is Equation (3.1). $(R_0 - I_s)$ is the intrinsic quality of the encoder before any transmission to the network. A is a tolerance factor set to $A = 0$. We convert R to MOS using an $R2MOS$ mapping function provided by the standard, see Figure 3.9. This notation has been used in Section 3.1.3.

one 5 seconds interval to the other.

Rationale (why does it perform well?)

We study the performance of the 'Maximize MOS' algorithm over some representative traces and we compare to the rest of the algorithms discussed in this Section. Let us first identify the good properties of this algorithm, that are responsible for the good performance and reduced sensitivity observed in the simulations.

1. The fact that delay is explicitly included in the objective function has two advantages. On one hand, it prevents overshooting in delay estimation (unlike the TCP-based estimation of *'Spike-Det'*) and provides a tight upper bound to network delay (see Figure 3.6(a)). On the other hand it prevents unnecessarily tight playout, as the objective function captures the fact that there is no benefit in reducing delay below 150ms at the cost of even minor loss.

2. The spike detection combined with the sliding window provides a good estimation and reaction (i.e. fast increase to spikes, slow decay after spikes and lack of sensitivity to the window W). This is similar to the *Assisted-Max-Fixed* algorithm (Section 2) and to what existing algorithms, such as [69, 64], have used. The slow adaptation after spikes outperforms schemes that rush into reducing the playout delay after spikes, thus often leading to unnecessary loss.

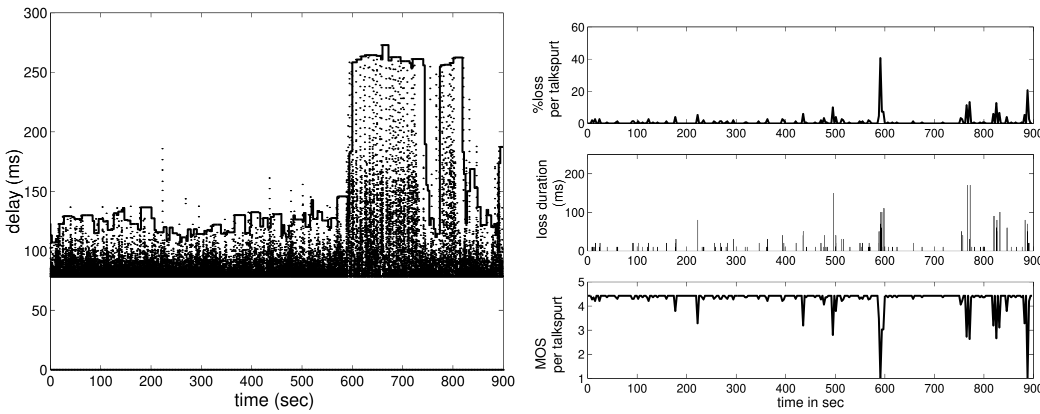
3. In addition to property (2) above, which is common in many schemes, this algorithm has the unique feature that it adjusts the delay percentile (i.e. target loss rate) dynamically with time; thus the algorithm tunes automatically itself to the trace, and drops packets only when reduction in delay is needed. Indeed, there is no need for the target loss rate to be fixed across the entire call. For example in Figure 3.28, the percentile should be larger in the first part of the trace where delay is low anyway and smaller in the middle part where interactivity is hurt due to high delay. Another example where the dynamic choice of loss rate can help, would be a trace with considerable network

loss that leaves no margin for additional loss due to late arrivals; such a case would be captured by the combined $MOS(loss, delay)$.

4. The algorithm is specifically designed to optimize the measure (i.e. combined *MOS(loss, delay)*) used for its performance evaluation. In this sense, it was expected to outperform other schemes when compared using this measure. However, the optimization of MOS turned out to be less important than the three previous observations. The algorithm performs well not only in terms of MOS but also in terms of raw measures, i.e. in terms of $(loss, delay)$ pairs (see comparison in Figure 3.39).

Sensitivity

The most important parameter of Algorithm 4 is the window W of recently received packets, which is used to capture delay history and calculate $q_{opt}\%$ to maximize MOS. On one hand, W should be large enough to collect enough samples to capture the delay distribution and perform the optimization (e.g. we need at least 100 packets (1sec) to be able to choose any percentile $\leq 99\%$). On the other hand, W should be short enough in order for the scheme to be responsive. E.g. Figure 3.28 showed a scheme that predicted p to be 99% of the past W delays; we saw that a long window W made the scheme less responsive, because it took longer for a change to affect the distribution of many samples. However in addition to the window mechanism, this algorithm used the spike detection that helped to react to changes faster and made, to a large extent, the algorithm insensitive to the choice of W. A different consideration is that W should match the durations that are meaningful for the Loss Impairment (I_e) and the Delay Impairment (I_d), in the order of some seconds and up to a minute respectively. Taking into account these considerations, we found that values of W in the order of a few sec to tens of sec to be appropriate. Furthermore, the scheme is insensitive to the tuning of the parameter. The choice of candidate percentiles and thresholds for entering and exiting a spike are not as important. We choose $ENTER = 1$ to detect even the smaller spikes exceeding p.



(a) Playout, using W=10sec

(b) Speech quality per talkspurt

Figure 3.36: Evaluation of Algorithm *'Maximize-MOS'*, with $W = 10sec$, over trace THR-P1-ASH, Wed 14:10-14:15.

Computation

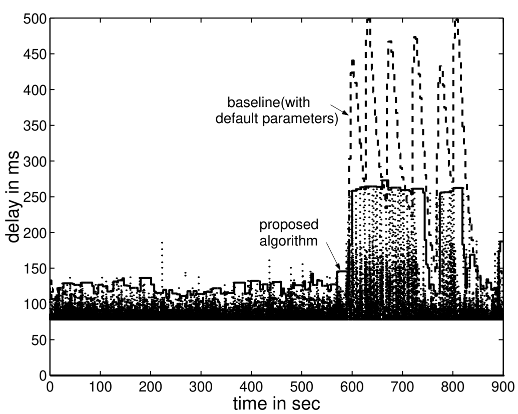
In terms of computation, the following facts reduce the computational complexity. First, we do not consider candidate percentiles from 0 to 100%, as only up to 10% rates can give acceptable quality. We tried {90%, 91%, ...100%} and {90%, 90.5%, ...99.5%, 100%} and obtained similar results. Second, the computation of optimal percentile, is needed only at the beginning of each talkspurts and only if there is a decrease in delay. Third, there are computationally efficient ways to update the delay histogram, e.g. as proposed in [69].

Evaluation over Network Traces

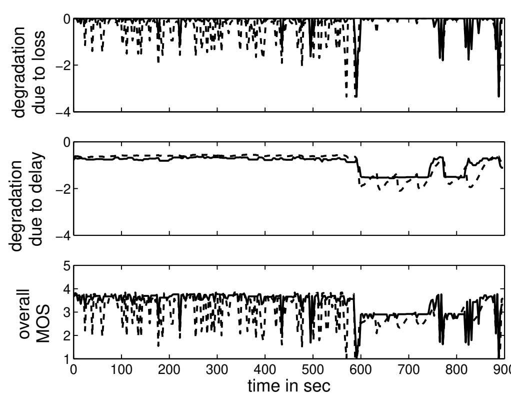
Let us now see the performance of Algorithm *'Maximize-MOS'* over the example P_1 trace that has also been used for evaluating the other algorithms: the first 15 minutes of Figure 3.26(e). Figure 3.36 shows that the algorithm with a window $W = 10sec$ provides a tight upper bound of the network delays (Figure 3.36(a)) with small loss rates and small loss durations (Figure 3.36(b)). Table 3.6 shows that the good performance is consistent for a large range of windows W.

Table 3.6: Sensitivity of Algorithm *'Maximize-MOS'* to the tuning of W, over example trace P_1(THR-P_4-ASH, Wed 14:00-14:15).

W (sec)	% packets lost	number of clips > 60ms	talkspurts affected	average delay	average MOS
1 sec	3.92%	64	38	131ms	3.81
2 sec	1.56%	23	18	145 ms	4.07
5 sec	0.76%	13	13	156 ms	4.27
10 sec	0.56%	16	16	162 ms	4.29
20 sec	0.62%	14	14	163 ms	4.29
30 sec	0.65%	16	16	164 ms	4.29
50 sec	0.71%	22	22	163 ms	4.28
100 sec	0.96%	29	29	160ms	4.28



(a) Network and playout delay



(b) Instantaneously perceived quality

Figure 3.37: Comparison of '*Maximize-MOS*' and *'Spike-Det'* (with default parameters)

Table 3.7: *'Maximize-MOS'* algorithm over the example P_4 trace, (EWR-P_4-SJC, Wed 21:00-21:10).

W (sec)	% packets lost	number of clips > 60ms	talkspurts affected	average delay	average MOS
2 sec	2.58%	36	13	82 ms	4.07
5 sec	2.17%	35	13	107 ms	4.07
10 sec	2.18%	43	16	131 ms	4.05
20 sec	2.37%	41	14	141 ms	4.08
50 sec	1.83%	33	10	154 ms	4.08
100 sec	1.18%	30	26	187 ms	4.01

Figure 3.37 shows that Algorithm *'Maximize-MOS'* outperforms the *'Spike-Det'* algorithm, proposed in [77]. This is due to the window based estimation (properties (2) and (3)) as opposed to the moving average based estimation that is inappropriate for the backbone traces composed by spikes, as discussed in Section 3.2.4.1 (see Figures 3.19 and 3.18 and Table 3.2).

A second example trace is the periodic trace of provider P_4 (Figures 3.26(c) or 2.42). *'Maximize-MOS'* performs well for a range of windows W, as shown in Table 3.7 and Figure 3.39. A third example trace we used was EWR-P_2-SJC, on Thu 20:10; *'Maximize-MOS'* outperformed again the *'Spike-Det'* approach.

It is worth mentioning that for all examples, there is an intermediate value of W that gives a slightly higher MOS. The loss rate does not drop to zero, even for very large W; in exchange the delay range remains narrower than the other schemes. This behavior is expected, as the objective function of the algorithm takes into account both loss and delay.

3.2.8 Comparison of Algorithms

This section compares all playout algorithms discussed using two representative traces (the $P_1$and P_4 examples), using raw performance measures, i.e. $(loss, delay)$ pairs. If a $(loss, delay)$ curve achieved by one algorithm lies entirely below the curve achieved by a second algorithm, then the first algorithm performs better. The $(loss, delay)$ curve is obtained

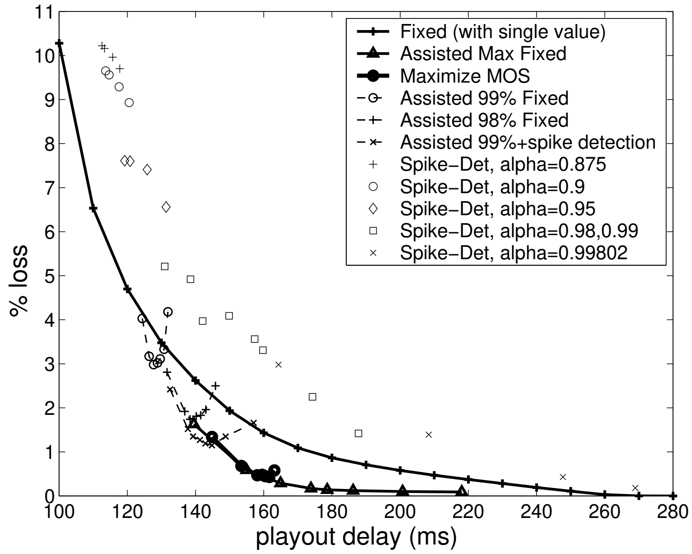


Figure 3.38: Comparison of all considered playout algorithms over the example trace P_1 (THR-P_1-ASH, Wed 14:00-14:15). The loss rates and the average delay are computed over the entire 15 minutes.

for each scheme by considering the entire range of its parameters. For example, the parameter for *'Maximize-MOS'* is the window W; the parameters for *'Spike-Det'* is the weight α for the moving average and /or the threshold for spike detection $ENTER$.

Let us consider 15 minutes of the example trace of Figure 3.26(e) and compare the performance of various playout algorithms over it. Figure 3.38 shows the loss rates and average delay achieved by the algorithms. It is an extension of Figure 3.29 that compared the *"Assisted Max Fixed"* approach to the *"Assisted 98% and 99% Fixed"* approaches. In Figure 3.38, we see that the *"Assisted Max Fixed"* outperforms the *Fixed playout* (with a single fixed value over the 15 minutes). This is expected as the *'Assisted Max Fixed'* can choose a lower delay for the first and a higher delay for the second 10 minutes. The *'Spike-Det'* approach is shown as disconnected $(loss, delay)$ pairs, for different values of its parameters (namely the weight α of the weighted average and the threshold $ENTER$ for entering a spike); it performs worse than all other approaches because the delay pattern of the traces under study consists mainly of spikes and not of slowly varying components.

Finally, the last of the *"Maximize-MOS"* approach performs similarly to the *'Assisted Max Fixed'* approach. In fact, it performs even better because it achieves a lower and narrow delay range, for the same range of window W (from 2 sec to 200 sec). The reason for this low and narrow delay range is that delay is included in the objective function and therefore the playout provides a tight upper bound of the network delay. When instead of the total loss rate, we consider the burst loss or the loss due to durations above 60 ms, we observed that all curves shift lower, without changing their relative position, as in Figure 3.27(b).

The second example trace used for comparison is the one with the periodic spikes every 60sec in Figure 3.26. Figure 3.39 compares the three algorithms of interest over this trace. Due to the unique regular pattern of this trace, the algorithms perform similarly to each other and to the fixed playout. The loss rate decreases almost linearly with the fixed playout delay, which can be explained by a closer look to the pattern of this trace, see Figure 2.42: triangular spikes have their upper part dropped by the fixed playout delay. Algorithm *'Maximize-MOS'* performs slightly better than the others and it achieves again a narrow lower delay range for the same window range than Algorithm *'Assisted-Max-Fixed'* (namely 2, 5, 10, 20, 50 and 100 seconds). *'Assisted-Max-Fixed'* behaves like a conservative fixed once its window exceeds the period of the trace ($W > 50sec$).

3.2.9 Summary and Discussion

Playout scheduling is an important component of the VoIP system that absorbs the delay variability and strongly affects the overall perceived quality. In this section, we first made some general observations concerning the design of playout scheduling, and then we proposed some schemes that make use of these observations.

Summary

In summary, we found the following:

- The delay estimation part of any playout algorithm should learn the delay patterns,

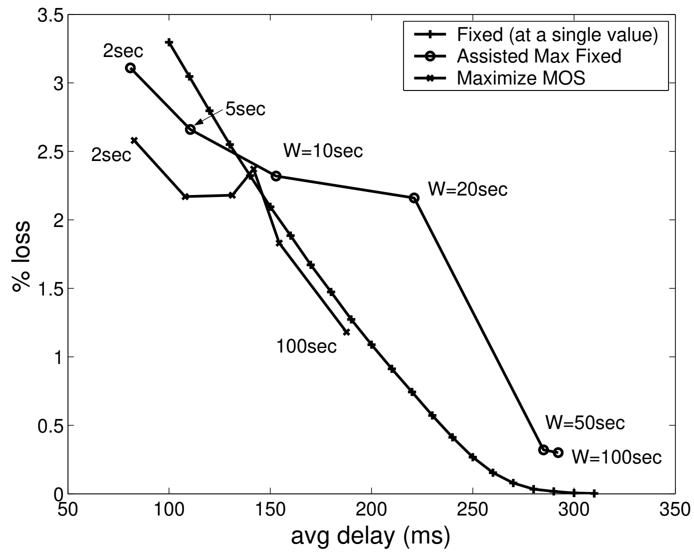


Figure 3.39: Comparison of playout algorithms over example trace P_4 (EWR-P_4-SJC, Thu 21:00-21:10).

before the algorithm is able to adapt to them. From the study of the measurements in Chapter 2, we know that delay in backbone networks follows patterns consisting of spikes of certain heights repeated over certain intervals. We recommend that the playout algorithm learns the distribution of the height and distances of delay spikes above a noticeable magnitude. It is also clear that there is no slowly varying delay component, to be tracked on these backbone networks; this is the reason why moving-average based estimation, as well as estimation that considers spikes as the exception rather as the rule of the delay pattern, fail.

- Even assuming a sufficient knowledge of the delay in the path, the appropriate adaptation depends on the tolerance of the user/application to speech clipping relatively to large end-to-end delay. In this sense, the adaptation part of playout scheduling can not be defined in a unique way. Therefore, it makes sense to design many "modes", that achieve different levels of delay and loss. We proposed a conservative, a configurable and an intermediate mode. Adaptation can be further tuned to achieve a desirable perceived effect, by tuning the parameters of each scheme to match the characteristics of each path.

- A fixed playout algorithm seems to be a good approach when the end-to-end delay is below the perceived threshold of 150 ms. Even in those cases, delay may suddenly change; therefore the delay on the path should be constantly monitored and the (fixed) playout delay should be infrequently adjusted, if needed, along the lines of Assisted Fixed Max algorithm. Following the network delay too closely comes at the risk of unnecessary loss due to failures of the estimation part of the algorithm. Therefore, for traces with low delay, a higher fixed value or the Assisted Max Fixed approach seem to be the appropriate choices. For traces with higher delay, there is value in making the playout delay follow the network delay closely.

- The study of the voice quality, in Section 3.1, indicates that there is a loss-delay tradeoff that can be exploited by the playout scheduling algorithm to maximize the overall perceived quality, along the lines of Sections 3.2.4.4 and 3.2.7.

Worst Case Scenarios

As part of this study, we proposed three algorithms: the *Assisted-Max-Fixed*, the *Fast Increase -Exp Decrease* and the *Maximize-MOS*. Let us now describe the worst case for these three algorithms. This can be thought of as a game between the playout algorithm and the network: the algorithm makes a choice of p at the beginning of each talkspurt, based on observations of network delay so far and estimating that the network will behave in a certain way in the near future. The worst case happens when the network behaves with the exact opposite way than expected, during each talkspurt when the algorithm has no opportunity to react.[6]

[6] In this game, the algorithm is limited by the frequency of its reaction opportunities. If the reaction opportunity is at the beginning each talkspurt, then the algorithm is vulnerable during each talkspurt. If the reaction opportunity is at every packet, then the algorithm is limited by the maximum allowed variation in playout rate (20-30%) that is not perceived.

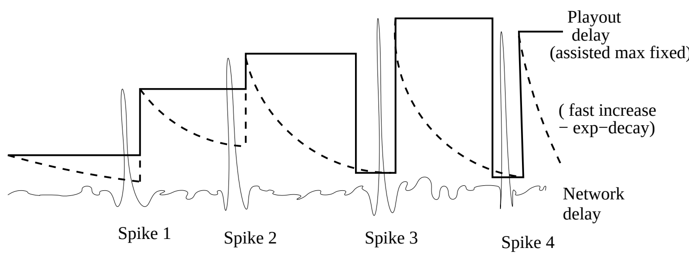


Figure 3.40: Sketch of worst case scenarios for two of the playout algorithms

Let us look at Figure 3.40 which shows a sketch of network delay in a thin solid line. The thick solid line shows the playout delay chosen by the *'Assisted Max Fixed'* algorithm. Let us assume that spike 1 happens in the middle of a talkspurt. Then the algorithm increases to the maximum value at the beginning of the next talkspurt, but is unable to save Spike 1 from being dropped. Spike 2 comes also unexpectedly in the middle of another talkspurt; again, the soonest the algorithm can react is at the beginning of the next talkspurt. After W seconds, no high delays have been observed and the algorithm believes that delays will be low from now on, and thus decides to decrease the playout. The worst case is that spike 3 happens immediately after this decrease. The algorithm increases again the delay, at the beginning of the next talkspurt, but at that point it is useless: the loss of spike 3 is already incurred and no more high delays happen. The delay remains unnecessarily high for the next W seconds. After W sec, the algorithm thinks that network delay finally decreased and decides to decrease the playout. The worst case is that spike 4 happens immediately after this decrease and so on. In summary, there are two worst case scenarios for the *'Assisted Max Fixed'* algorithm. The first is unexpected high delay that leads to loss (as in spikes 1, 2, 3, 4). The second is the low delay immediately after the increase in the playout (as in spikes 3, 4): then a high delay is incurred for no good reason.

The *'Fast Increase- Exp Decay'* algorithm is shown in the thick dashed line and it has similar worst case scenarios with the previous one: an unexpected high spike (that leads to loss) or a spike that does not repeat itself (thus leading to an unnecessary increase in delay). The third algorithm, *'Maximize MOS'*, is not shown in this figure. Similarly to the previous ones, it is also subject to the first bad scenario (unexpected high spike that leads to loss). It is not as strongly subject to the second one: a high delay has to be frequent enough (a certain percentage of the window W) to have a lasting effect on increased playout delay. However, the percentage approach, has another worst case scenario shown in Figure 3.28: it takes some time to realize and react to a change, even when assisted by spike detection.

Limitations and Future Directions

The limitations, as well as the possible extensions, of the schemes studied in this section are the following.

- First, all the schemes discussed here assume that the opportunity for adaptation is at the beginning of each talkspurt. Therefore, their effectiveness is limited by the length of the talkspurt compared to the time scale of delay variation in the network. As talkspurts and silences become shorter (e.g. see G.729B that uses a dynamic hangover), the algorithms have more chances to adapt. At the extreme, there should be an opportunity for adaptation at every packet, as is already done by [64]. The principles concerning the learning of the trace and the loss-delay tradeoff in the overall quality are orthogonal to the continuous time adaptation and could be combined with it. For example, if the effect of variable playout rate is systematically quantified, it can extend the objective function ($MOS(delay, loss, jitter)$) that is used to maximize the overall perceived quality.

- Second, as mentioned in Section 3.2.4.1, learning the delay pattern is a large problem by itself, which includes a model for delay (relevant to VoIP quality) and a way to detect changes in this model. The problem of learning the delay pattern is not solved in its entirety in this thesis. Instead, we were guided by the observation that the delay pattern consists of delay spikes of different shapes, heights and distances which should be characterized; we also used heuristics (such as a sliding window and spike detection) to detect changes. In a future version, the algorithms could also adjust their own parameters throughout the call, based on the perceived quality experienced so far.[7]

- It should be also made clear, that the Emodel provides a rough assessment of voice quality, which is more appropriate for planning than for evaluating a specific speech sample. Again, the principles outlined here can be combined with a more refined voice assessment model, i.e. with a more accurate $MOS(loss)$ function.

If we are to identify the single most important direction for continuing the work on playout, this will be the following. So far, we studied the performance of some existing as well as of our proposed algorithms, for a range of their parameters. However, in practice an actual system will have to face the reverse problem: it will need to tune online its parameters to handle an unknown delay and loss pattern on an unknown Internet path. Therefore, the natural way to continue this study is toward automating the process of learning the trace, tuning the parameters of the algorithms and switching between modes, as the delay and loss characteristics change dynamically.

3.3 Numerical Results

In this section, we consider many phonecalls during certain one-hour periods of various representative paths and providers. We present statistical results for the quality of these phonecalls, that express the probability that a phone call placed during a considered period will experience a certain level of quality.

For each call, we consider talkspurts and silences that follow an exponential distribution

[7] The only one of the three algorithms, that currently adapts one of its own parameters is the 'Maximize MOS' algorithm, that adapts the optimal percentile $q\%$ along the call. Other parameters that could be also adjusted along the call, is the window W and the exponential decay T_{decay}. However, the simulations in this section have been performed with the same tuning of these parameters throughout a call. Adaptation of playout delay happens at every opportunity (e.g. beginning of talkspurt). However, adaptation of the algorithm parameters should happen less frequently.

with mean 1.5 *sec* each, similarly to [50]. We consider fixed playout and the *'Spike-Det'* algorithm over these traces.[8] The quality of the call may be degraded by the combined effect of loss and delay in the network as well as in the playout buffer. We then apply the methodology described in Section 3.1.4.1 to obtain (i) a rating that a human would give at the end of a conversation (as in the example of Section 3.13) and (ii) the minimum MOS experienced during the call. The rating at the end of a call is more lenient as humans tend to forget bad events after some time; however, a network operator might want to capture and eliminate such events, thus the relevance of the minimum MOS rating.

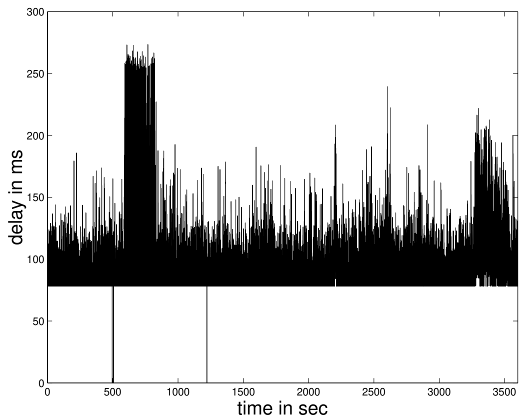
Having discussed one call, we now consider many calls initiated at random times, uniformly spread over each one-hour period. We consider exponentially distributed call durations as in [49]: 150 short (mean duration is 3.5 minutes) and 50 long (mean of 10 minutes each) calls simulate business and residential calls respectively. The quality of each call is varying over time, as shown in the example of Figure 3.41(b).

We provide statistical results for the quality of phonecalls in one hour intervals, for paths of different providers. First, we show results from a high delay variability path (THR-P_1-ASH), on which the results strongly depend on the playout used. Then, we show an example path at the other extreme: ASH-P_6-SJC has very low delay variability. Then, we show results from provider P_4 that has the a periodic pattern.

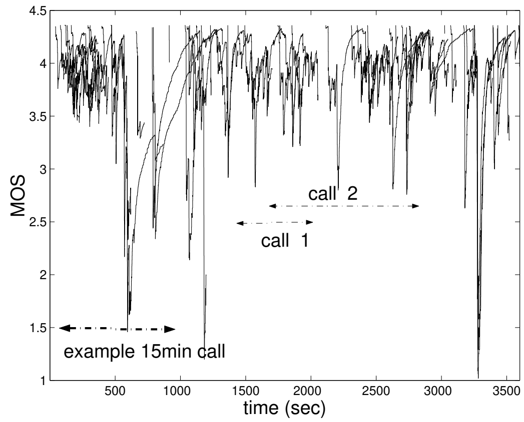
A high delay variability path

Let us consider the one-hour trace (Wed 14:00-15:00) from path **THR-P_1-ASH** (Wed 14:00-14:20) shown in Figure 3.41(a). The reader may be already familiar with this trace, from the Measurements Chapter and the evaluation of the playout algorithms. The selected trace exhibits large delay variations and a period of 1.03sec loss (at 14:10). Figures 3.18, 3.17, 3.27, 3.34 and 3.36 show the playout times chosen and the quality achieved by the *'Spike-Det'*,

[8] In this section, the idea is to consider existing, widely used, playout schemes, as part of the end-to-end VoIP system. We first consider fixed playout for a range of fixed playout delays, as a benchmark for comparison, and then the widely known adaptive scheme, [77].



(a) Network delays

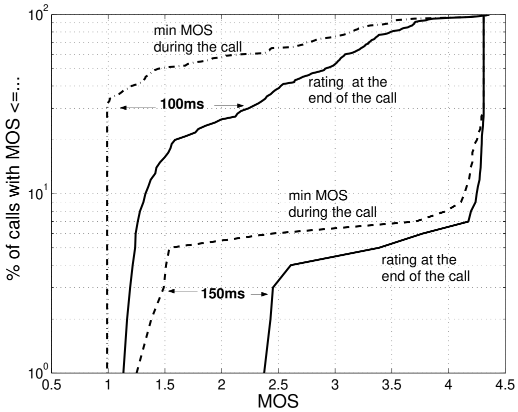


(b) Time varying quality of 50 calls

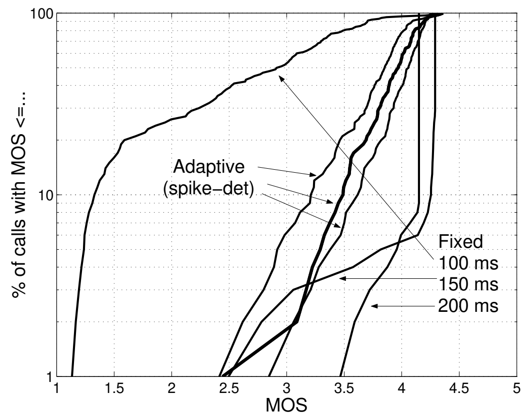
Figure 3.41: Consider one hour period (Wed 14:00-15:00) on path THR-P_1-ASH. Consider many calls with exponential duration, starting at random times, over this hour. Fixed playout at 100ms is applied.

Fixed, *'Assisted-Max-Fixed'*, *'Fast Increase - Exp Decay'* and *'Maximize-MOS'* algorithms respectively. These results were achieved using G.711 encoding, which has a high intrinsic quality, an adequate echo cancellation ($EL = 51dB$) and considering a medium interactivity requirement.

Figure 3.41(b) shows the time varying quality of 50 out of the 200 (150 short and 50 long) considered calls. Figure 3.42 shows the cumulative distribution (CDF) of ratings for the 200 calls. Figure 3.42(a) refers to fixed playout using both measures. Figure 3.42(b) refers to both fixed and adaptive playout (*'Spike-Det'* considered here) using only the rating at the end of a call. For the *'Spike-Det'* algorithm, there are three curves: one with the default parameters (weight α and threshold $ENTER$) and two with better tuned parameters. If fixed playout is used, then the choice of the fixed value becomes critical: $150\,ms$ is acceptable (only 6% of the calls have final rating below 3.6 and only 8% of them experience a period of $MOS < 3.6$) while $100\,ms$ is totally unacceptable (90% of the calls have rating at the end below 3.6). Fixed playout at 200 ms guarantees $MOS > 3.6$ for all calls, although the



(a) Fixed playout. Ratings: (i) minimum MOS (ii) at the end of a call



(b) Fixed and Adaptive ('spike-det'). Rating at the end of a call

Figure 3.42: CDF of call ratings in one-hour period (Wed 14:00-15:00) on path THR-P1-ASH, considering G.711, lenient conversation task and $EL = 51dB$.

delay impairment is larger. The minimum rating is then due to a 1.04 sec loss duration in the network; calls happening during that time receive a lower rating. For the *'Spike-Det'* adaptive playout, we observe the following: (i) the CDF more linearly than for the fixed scheme (ii) this performance is acceptable but still not excellent (10% of the calls have overall rating $MOS < 3.6$ and 50% of them experience a period of $MOS < 3.5$ at least once) and (iii) tuning of the parameters does not lead to significant improvement. This is in accordance with Table 3.2 and Figure 3.19 that showed poor performance of the moving-average based scheme for a large range of its parameters.

Supporting G.729, which has a lower intrinsic quality, at acceptable quality levels is even more difficult. Compare Figure 3.43 to Figure 3.42(b) that showed the performance for G.711. Fixed playout is still the best choice but the CDF is shifted by the difference in intrinsic quality. The *'Spike-Det'* adaptive algorithm is even worse than its shifted version: 98.5% of the calls have unacceptable quality ($MOS < 3.6$). The 'maximize MOS' algorithm, improves over 'spike-det' but still leads to unacceptable performance: 97% of the calls have $MOS < 3.6$.

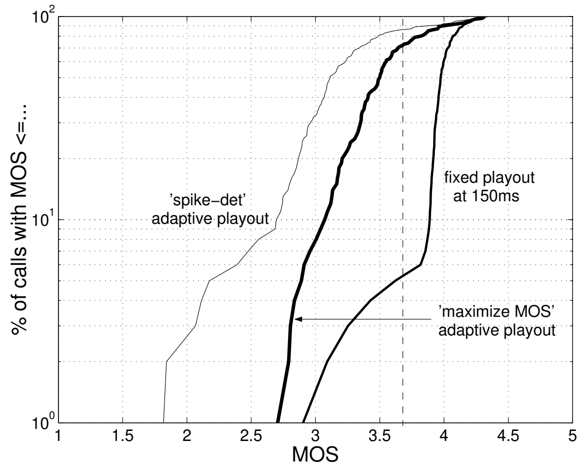


Figure 3.43: CDF of call quality (overall rating at the end of the call) for the on-hour period (Wed 14:00-15:00) on path THR-P_1-ASH, considering $G.711$, lenient conversation task and $EL = 51dB$.

While in Figures 3.42 and 3.43 we plotted the entire CDF for one-hour period, in Figure 3.44 we consider only some percentiles (namely the worst rating, the 10%, 50%, 90% and the best rating among the 200 calls) for each one-hour period of the entire day. For example, the points in Figure 3.44(a) for $Hour = 14$ are consistent with Figure 3.42(a): out of the 200 calls between 14:00 and 15:00, the worst rating is 1.1, 10% of the calls have $MOS \leq 1.4\%$, 50% of the calls have $MOS \leq 3$, 90% of the calls have $MOS \leq 3.75$ and some calls have perfect rating. Figure 3.44(a) shows that a fixed playout at $100\,ms$ is unacceptable when the delays on the path are high, i.e. during the business hours. A fixed value at $150\,ms$, Figure 3.44(a) is a safe choice as no more than 1-2% of the network delays exceed it, see percentiles Figure 2.45. In practice, the choice of the fixed playout value should not be the same for the entire day, but should be infrequently adjusted (as in the *"Assisted Max Fixed"* algorithm). The adaptive playout ('*Spike-De*t') in Figure 3.45(c), had the same performance for the entire day including the business hours, because it was able to monitor the changes in the network delays. However, 10% of the calls in any hour have still $MOS < 3.5$.

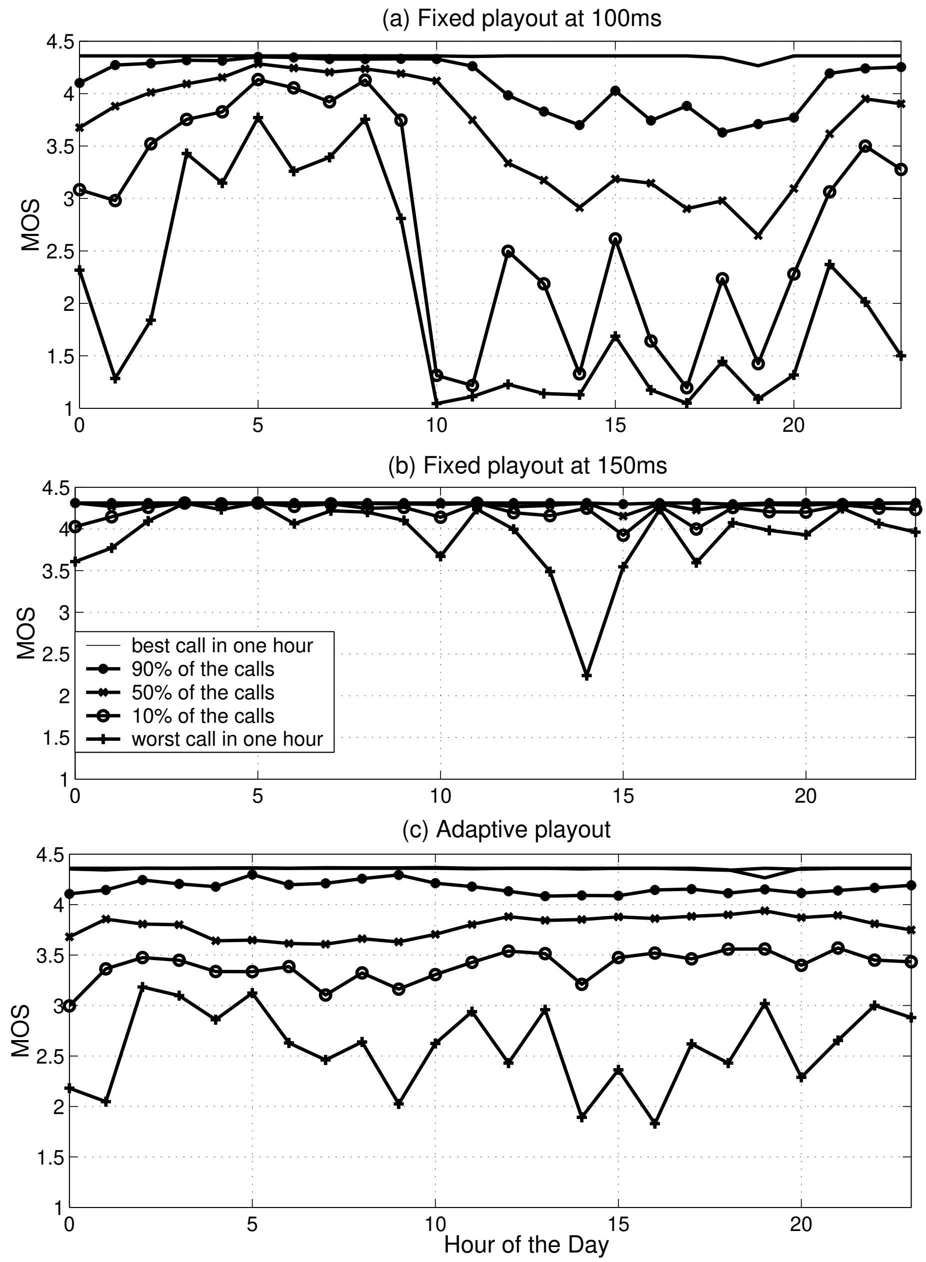
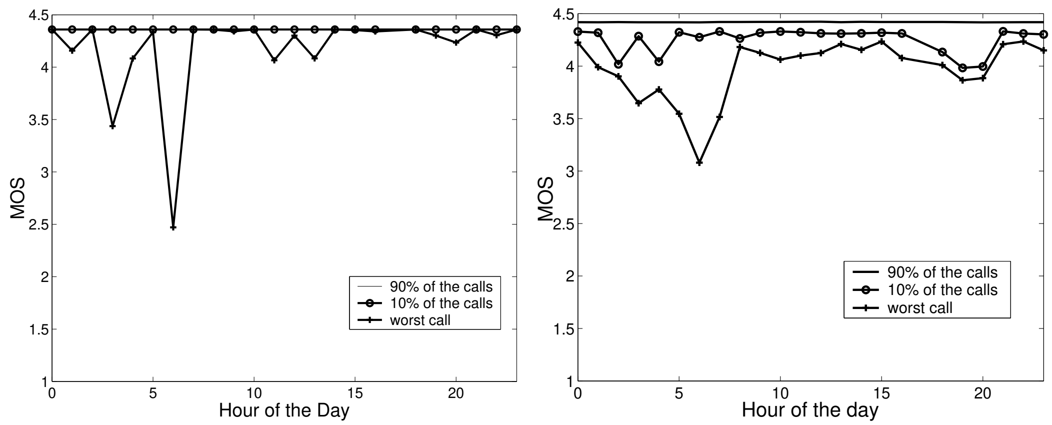


Figure 3.44: Call quality statistics for every hour of an entire day (Wed 06/27/01) on path THR-P_1-ASH. Playout used: (a) Fixed at 100ms (b) Fixed at 150ms (c) Adaptive ('Spike-Det') with default parameters.



(a) Fixed playout at 100ms

(b) Adaptive playout ('spike-det')

Figure 3.45: Call quality statistics (percentiles of call ratings exceeding a certain MOS), on path EWR-P_6-ASH, in one-hour periods on Wed 06/27/01 .

A low delay variability path

On the other extreme, there are paths with very low delay variability, such as those belonging to providers P_6 and P_7. Such paths can achieve an excellent MOS at all times except for the rare cases of long loss durations in the network. Given that the fixed part of the delay on these paths is below 50 ms, a conservatively high fixed playout delay of 100-150 ms is sufficient to yield excellent performance, except for a few very high delay spikes, which are short lived anyway and are perceived as a clip in speech.

An example of such a path is ***EWR-P_6-ASH***. Figure 3.45 shows percentiles for the quality of calls in every hour for 24 hours (Wed 06/27/01 UTC). Figure 3.45(a) shows the quality using a fixed playout at 100ms: 90% of the calls have perfect MOS ($MOS \simeq 4.4$). There are hours when the worst call during that hour can be as low as 2.5. This is due to unavoidable network loss. The low rating during hour 3-4, is due to the complex loss event at 3:20. The low rating during hour 6-7 is due to the complex loss event at 6:30, that happens simultaneously on many paths. The other four low MOS values are due to 20-25 long clips following a high spike.

We also observed that adaptive playout should be used carefully on these paths. Algorithms that try to track closely the delay may cause damage instead of improvement. This happens if the estimation part of the algorithm fails to predict the delay and leads to unnecessary loss, while there was no need for decreasing the delay at the first place. In Figure 3.45(b), we see that the *'Spike-Det'* algorithm causes unnecessary loss. 90% of the calls have no more the perfect quality of Figure 3.45(a).

A path with periodic delay pattern

Another distinct type of paths, are the those belonging to provider P_4, which exhibit periodical clusters of spikes, as high as 250-300ms (see Figure 2.42). These periodic spikes are so high that they cannot be accommodated without loss either in speech quality or in interactivity, thus leading to a low combined MOS. The delay behavior of these paths does not vary across the day; therefore it is sufficient to examine a typical hour of path ***SJC*-P_4 *-ASH***. In Figure 3.46, we show the results using the adaptive playout *'Spike-Det'* with its default parameters: 20% of the calls have overall $MOS < 3.5$ and 80% of the calls experience $MOS < 3.5$ for some period. If a strict interactivity requirement is applied, then the entire CDF degrades by approximately 0.8 unit of MOS. Due to the special periodic delay pattern, an appropriately high fixed delay can work sufficiently well for this path, as shown in Figure 3.39. For example 150 ms fixed playout delay would reduce the number of calls with $MOS < 3.5$ to 10%. In addition to the periodic delay pattern, provider P_4 has the problem of frequent outages, lasting for tens of seconds, described in Section 2.5.

Difference between providers

In Chapter 2, we saw that different providers have different loss and delay behavior, consistent for paths (or subgroups of paths) of the same provider. This is intuitively expected as paths of the same provider may share network resources. In this section, we saw that these patterns translate to a consistently good or bad behavior of VoIP in terms of MOS. Table 3.8

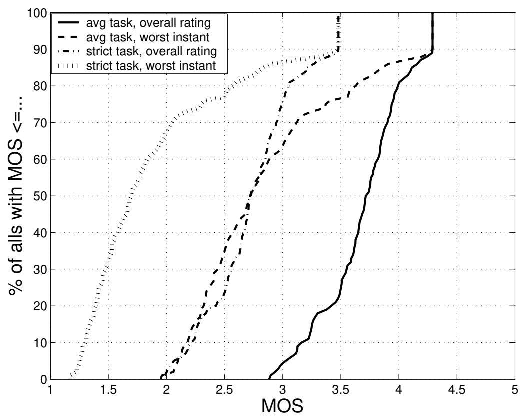
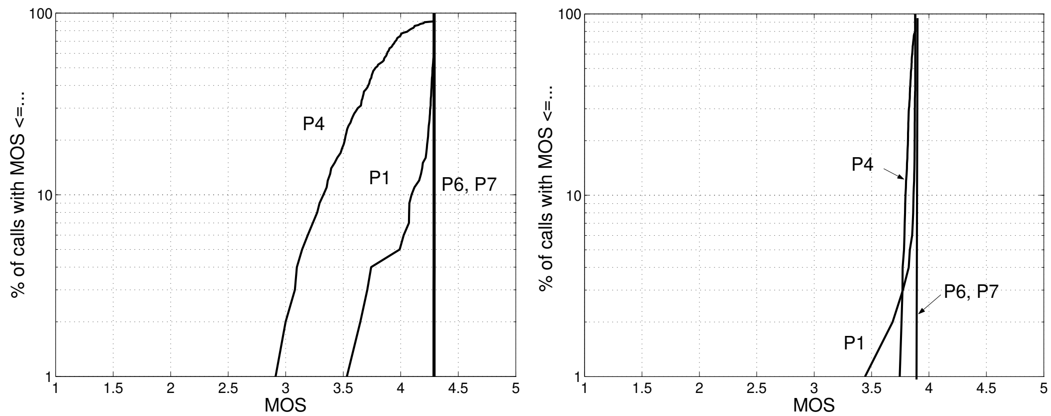


Figure 3.46: Call quality statistics for one hour (Wed 20:00-21:00) on path SJC-P_4-ASH. Adaptive playout (*'Spike-Det'* with default parameters) used.

Table 3.8: Summary of paths

	Number of paths				High loss periods	
delay variability	low	high	occasionally high	periodic pattern	duration (sec)	per day
provider P_1	2	6	4		1-40	1-10
provider P_2	1		1 (for hours)		1-15	2-3
provider P_3	2		4		6-20	1-2
provider P_4				6	15-20	3-5
provider P_5	2		4 (for minutes)		1-3	1
provider P_6	5		4 (for seconds)		1-12	1-2
provider P_7	2				112-160	1
VoIP quality	good	poor	good with appropriate playout	poor	poor during those periods	



(a) Fixed playout at 150ms (b) Fixed playout at 250 ms

Figure 3.47: Paths between the same end-points (from EWR to ASH) through different providers, during the same one-hour period (Wed 06/27/01, 3-4pm EST time)

shows the types of paths per provider. The second column shows the number of paths with very low variability, such as the two paths of provider P_7, the example path EWR-P_6-ASH discussed above and several paths of the other providers. These paths can exhibit excellent performance even with a fixed playout value. The third column includes paths with so high delay and delay variability, such as THR-P_1-ASH and EWR-P_1-ASH, that perform poorly even for a good playout. The forth column, corresponds to path that they have mostly low delay variability, with some periods of higher delay, such as those shown in Figures 2.7 and 2.11. These paths can perform well if the playout captures these traditions. The choice of playout scheme and the tuning of its parameters is critical for the overall perceived performance for all paths. The fifth column corresponds to the unique periodic pattern of provider P_4; the periodic spikes are so high that hurt either speech quality or interactivity. The last two columns show that periods of high loss or outages happen in all kinds of paths.

These observations hold similar for both short and long distance paths. The reason for this, is that most of these backbone paths have delay which is not significantly higher than 150ms. Calls going through multiple backbones or through wireless/access networks would incur even larger delay.

The performance achieved also depends on the system components and the user requirements. Using G.711 encoder with high intrinsic quality, good echo cancellation and low interactivity requirements, paths of column 3 are barely able to provide acceptable ($MOS > 3.6$) VoIP service, far below the guarantees of the telephone network. Performance is even worse for stringent application requirements or less favorable system configurations. For example, strict interactivity requirements decrease MOS by roughly 0.5-1 units. Inadequate echo cancellation has a similar effect. Support of G.729, which has lower intrinsic quality, is possible only on paths of column 2.

The most important observation in this section, as well as in Chapter 2, is the consistent behavior per provider. Therefore, the user experience depends strongly on the backbone provider used. Figure 3.47 shows statistics for the quality of calls placed during the same one-hour period (between 3 and 4 pm EST) between the same cities (EWR and ASH, both at the east coast). The overall MOS depends on the provider used. Providers like P_6 and P_7 with low delay and low delay variability experience perfect quality by just using a fixed playout of 150 ms. This is not the case neither for the loaded provider P_1 nor for P_4. A higher fixed playout at 250ms, improves the quality of many calls on P_1 and P_4 but decreases the maximum quality due to the interactivity impairment.

3.4 Path Diversity

Many measurement studies have demonstrated the wide range of behaviors of Internet paths. For example in [84], it has been showed that 30-80% of the times there is an alternate path between two hosts that performs better than the default path selected for routing. In the previous section as well as in Chapter 2, we showed that there is a large range of behaviors in the backbone paths under study and that this behavior is, to a large extent, consistent per provider. If the user has no control over the path taken, then the voice quality experienced depends on the path used. The only action that can be taken then to mitigate the effect of delay and loss is optimized scheduling of packet transmissions at the sender and/or playout scheduling and concealment at the receiving end. However, if the application has control over the paths taken, then the ***path diversity*** can also be exploited in many ways.

3.4.1 Route Control

The first way is ***route control***, which stands for the selection of the path that exhibits the best characteristics among the available ones, at a specific time. This is what RouteScience Technologies Inc., [79], does: they monitor the performance of multiple paths and they allow their clients (typically enterprises and service providers) to control their layer-3 routing at the network edge, in order to increase the availability and predictability of applications. Of interest to multimedia traffic is the capability of routing traffic in real time (sub-second) to switch to the best ISP path. The approach of route control is justified by the wide range of behaviors observed in the measurements, between paths and in particular between providers. For example, availability can improve dramatically as the number of paths used increases: whereas outages have a non negligible probability of happening in either path, the probability of two outages at the same time is extremely low. Another potential improvement is the decrease in delay by choosing a shortest path, especially for international communications. A concern about the route control approach is the reordering of packets and the disruption of the continuity of a real-time multimedia stream. Another concern is whether instability can be caused when switching routes happens frequently and in a distributed way. The measurements we studied, showed that paths had consistent behavior across the day and the most severe problems (such as long loss periods or outages and periodic spikes) seemed more related to reliability than to traffic load. Switching routes is needed during those high loss periods but not on a continuous time basis. Such infrequent changes of routes are less possible to cause instability. Therefore, based on the loss episodes observed in the traces, route control can help in improving the quality of multimedia traffic.

3.4.2 Multipath Streaming

A second way to exploit path diversity for streaming multimedia is ***multipath streaming***. Transmission through multiple paths combined with multiple description coding, has been proposed for video communication over lossy networks, where odd and even frames of a video sequence are transmitted over different paths, [2]. This approach achieves decrease in loss rate and delay, as the application sees an average path behavior. In particular, burst loss is converted to isolated loss and the outage probability is decreased. The bandwidth overhead, from sending more than one streams, is limited by efficiently encoding.

Playout scheduling of multiple streams

[63] used the multipath approach to improve playout scheduling for voice communication. They proposed that multiple descriptions of the voice stream are used. The basic idea for two streams is the following. The even samples are quantized in finer resolution, the difference between adjacent even and odd samples is quantized in coarser resolution and all the samples are packetized into stream 1. For stream 2, the even and odd samples are quantized in the opposite way. The resulting coding redundancy is 25%.

Sending these descriptions over different network paths, enables the playout at the receiver to take advantage of the largely uncorrelated jitter characteristics of the two paths. If packets of one stream are lost, there is still a good probability that the corresponding packets of the other stream are not lost. There is still distortion due to quantization noise, but having one packet is better than losing both. Furtermore, scheduling the playout of each packet is chosen to minimize a distortion measure that accounts for both delay and loss distortion due to packet loss in either or both streams. This cost function is similar to the one we considered in Section 3.2.7, adding the delay and loss impairments in the appropriate scale, to obtain the overall quality. When switching between the two streams during speech playout, the playout delay needs to be adjusted to the delay statistics of each individual stream. The dynamic setting of each packet's playout is achieved using the packet scaling technique proposed in [64].

Let us now evaluate the proposed in [63],[9] considering representative paths of different providers connecting the same end-points.

The first example is shown in Figure 3.48 and considers two paths connecting ASH and SJC, on Wed at 3:20, over which two descriptions of the voice stream are sent, for 141 sec. One path is through provider P_4 and the other is through provider P_6. The first three plots in Figure 3.48 show the network delay on the first path (ASH-P_4-SJC), the network delay on the second path (ASH-P_6-SJC) and the playout delay at the receiver using both streams. We see that the receiver intelligently chooses among the two streams: in general it prefers the second stream that has lower and almost fixed delay. However, during the high delay and loss event (between packets 4000 and 6000) it switches to the first stream. Note that this is the same complex loss event described in detail in Section 2.3.3. The fourth plot in this figure shows explicitly which stream is chosen at the receiver. The last plot in this Figure shows the improvement when using multipath, as opposed to using only either of the two paths. We can see clearly the reduction of both loss rate and average delay. Although the loss rate is computed as an average over the entire 140 seconds considered, the reduction in the burst loss in particular, shows that the receiver makes the right choice in a short time scale too (burst loss is more difficult to conceal than single packets lost). The different points on the loss-delay curves have been obtained by varying the relative weights of loss and delay in the distortion measure, thus affecting the choices that are considered optimal at the receiver.

A second example is shown in Figure 3.49. It considers two paths connecting EWR and ASH, on Wed at 19:10, over which two descriptions of the voice stream are sent. One path is through provider P_1 and the other is through provider P_4. The first path has high and variable delay. The second path is our familiar (from section 2.4.7) periodic trace; in general it has low delay except for the high delay clusters every 60 seconds. The choice

[9] The author is grateful to Yi Liang for the simulations of playout scheduling of multiple streams, according to the scheme proposed in [63].

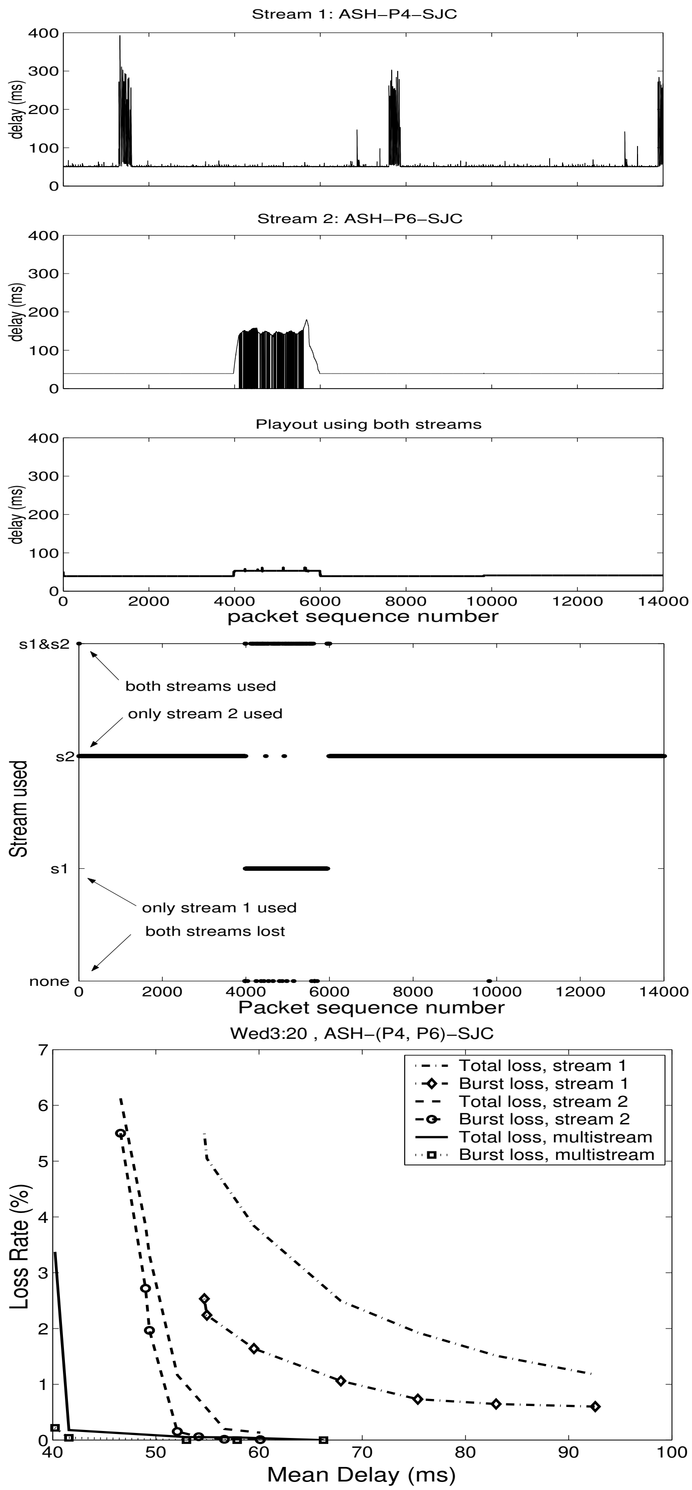


Figure 3.48: Playout scheduling of multiple streams. Two paths used: ASH-P_4-SJC and ASH-P_6-SJC. The last 141 sec of the trace Wed 3:20-3:30 are used (probes 44200-58300).

at the receiver is shown in the third and forth plots in Figure 3.49. Most of the time, the receiver prefers the second stream , except for the periodic clusters periods when it makes use of the first stream. The last plot in this Figure, shows the reduction in loss and delay. Of particular importance is again the reduction of bursty loss, which is difficult to conceal.

The less correlated the paths, the greater the gain from path diversity. The paths considered above belonged to different providers and had clearly different delay patterns, thus leading to significant gain for the multipath scheme. However, in both examples, one path had lower delay, and thus was preferred, most of the time except for some infrequent increases in delay. If the two paths had the same fixed delay or the path with the lowest delay had more network loss, then the distortion-delay optimization at the playout at the receiver would switch more often between the two streams. Finally, if the two paths were among those that incurred the synchronized loss events, as discussed in Section 2.15, the use of multiple paths would not help.

Rate-distortion optimized streaming over multiple paths

In [10, 11], the multipath concept has been applied to the rate-distortion optimized media streaming framework developed in [13]. This framework considers (i) the graph of dependencies between the data units of a video stream (ii) the available transmission opportunities, at the sender, at the receiver or at an intermediate proxy server (according to the extension proposed in [10]) (iii) a bandwidth constraint to be met at minimum distortion. It then chooses optimally the transmission policy of the multimedia data units. In this framework, multipath streaming provides more transmission opportunities than streaming over a single path, and thus a chance for further optimization.

A scheme of rate-distortion optimized streaming over multiple paths is currently under development (see [11]) and is referred to as RaDiO-multipath. The scheme has been evaluated for audio (in [10]) and video (preliminary results are provided in [27]) streaming, by modeling the two independent channels with a two-state Markov model. In Section 4.2.2, we

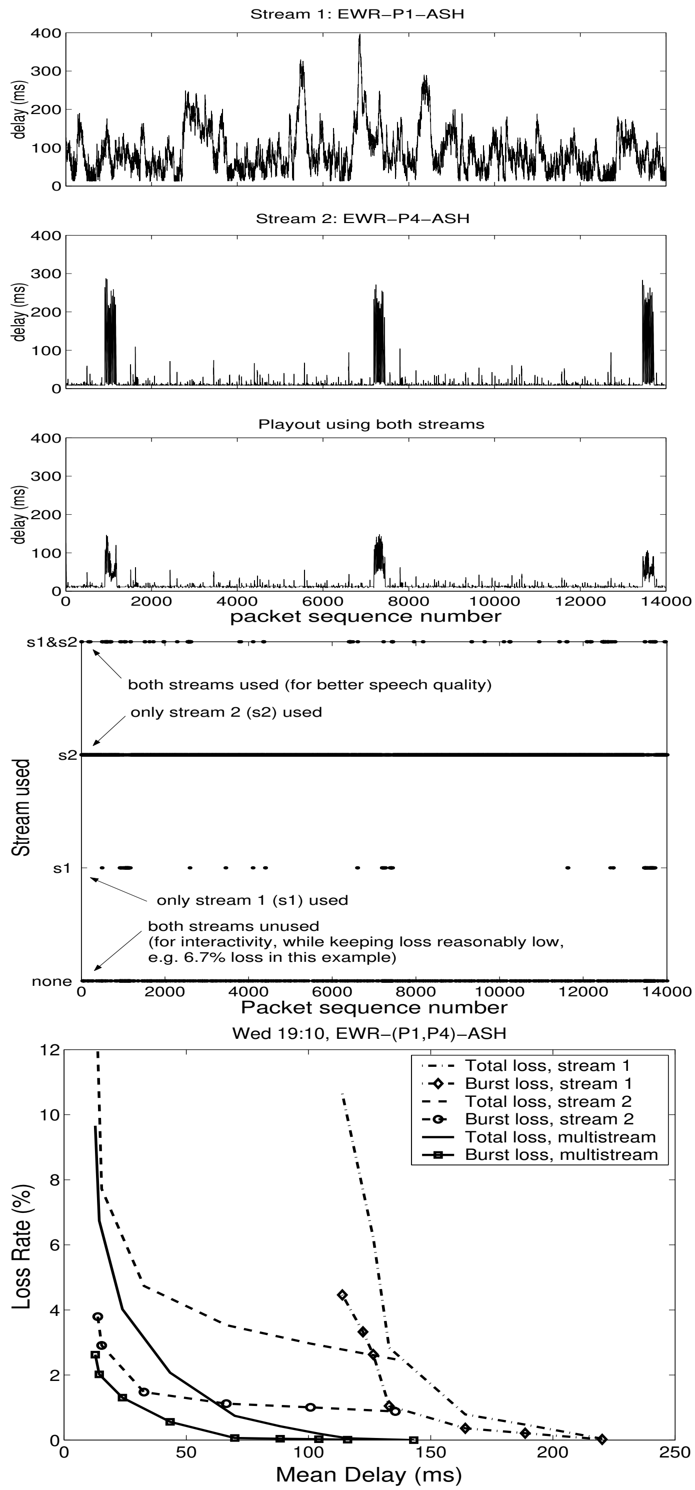


Figure 3.49: Playout scheduling of multiple streams. Two paths used: EWR-P_4-ASH and EWR-P_6-ASH. 150 sec from the trace Wed 19:10-19:20 are used (probes 44000-59000).

evaluate RaDiO-multipath for streaming video with tight delay constraints over representative traces. The tighter the delay constraint imposed by interactivity, the larger the benefit from using multiple paths. The reader is referred to Section 4.2.2 for numerical results. The same idea can be also applied to speech and VoIP.

3.4.3 Discussion

Benefit. The benefit by both route control and multipath streaming depends on the correlation between the paths. The less correlated the paths, the larger the benefit from path diversity. In this section, we verified these facts, by simulating the [63] scheme over representative traces. If the client uses paths from two totally disjoint ISPs (that is, neither is buying bandwidth from the other), then the benefit is potentially large. For a customer far from core Internet networks, all paths follow the same route to the core; in that case, a large portion of the path is shared and the benefit is limited. If/when in the future Internet paths, especially in the backbones, evolve to look like today's good paths, there will be only a small burstiness in time and little (or consistent) difference between paths to be exploited by path diversity.

Concerns. A common criticism against path diversity approaches is that they increase the amount of traffic sent over the network. However, this is not necessarily true if efficient coded multiple descriptions (e.g. complementary instead of duplicate streams) are sent over the two paths, [2, 63]. Redundancy can add further protection from loss but is not necessary for exploiting path diversity. Another question is the number of paths that should be used; in general, the benefit decreases fast with the number of paths. An important issue is the implementation of multipath streaming in the current Internet where the user has no control over the route taken. [63] and [2] propose to implement multipath streaming using relay servers or source routing. Recent proposals, such as [17], are in favor of enabling the source to specify the end-to-end path. [12] also proposes a scheme that allows for loose source-routing through specified administrative domains (AS), via relay servers. Finally, the synchronization of "bad" events that we observed in the measurements (see Section 2.3.4) is also a limitation to the effectiveness of path diversity approaches.

Chapter 4

Video Conferencing over Internet Backbones

There has been a significant increase in the video traffic over the Internet, during the last years. This traffic belongs to a variety of applications such as downloading pre-recorded video, one-way video streaming (e.g. news broadcast) and interactive visual communications (including teleconferencing, video telephony, virtual classrooms). Of interest to this study is the last category, interactive voice and video communications, which we refer to as video conferencing.

Video conferencing has two kinds of quality requirements. First, low delay is needed for effective communication. Also, to maintain lip synchronization when playing audio and video, delay should be at similar levels for both traffic types. Therefore, the delay requirement for interactive video applications is the same as the one needed to maintain an interactive voice conversation. It is because of this strict delay constraint, that interactive video communications are considered among the most demanding of video applications. For such applications, encoding and decoding must be accomplished in real time. To satisfy the low delay requirement, buffers at the encoder and the decoder are typically small and retransmissions are often impossible within the delay budget. Time consuming processing, often needed for error control and recovery, is also limited by the delay budget.

The second requirement is acceptable video quality, i.e. image quality and temporal smoothness. Fortunately, the expectations are less stringent than for other video applications: low to medium temporal and spatial resolution is in general accepted. For example, QCIF at 10 fps is considered acceptable for video phone applications and CIF at 10-20 fps is satisfactory for most video conferencing scenarios, [100]. Furthermore, a moderate amount of compressions and transmission artifacts is often tolerable.

However, even with lower user expectations, error control over the Internet is challenging for video. Transmission of packetized video over the Internet is prone to loss of packets, in the network or at playout time due to delay jitter. Packet loss is not a major problem for TCP-based applications and it is a problem to a certain extent for voice, as discussed in Section 3.1.2.3. However, compressed video streams are particularly vulnerable to loss, due to the dependencies introduced during compression. A lost packet can affect other frames due to the use of temporal prediction; this way, the error propagates temporally, but also spatially due to the use of motion-compensation prediction. A lost packet also affects other parts of the same frame that depend on the lost part (e.g. differential encoding of motion vectors). Finally, if a frame fits into multiple packets, a single lost packet makes the entire part of the frame around it (i.e. between the previous and the next synchronization marks) useless. A large number of error control and recovery mechanisms for video have been proposed to avoid or overcome the effect of loss. A good survey can be found in [101]. Error control and recovery is even more difficult for interactive video applications, because many useful mechanisms (like retransmissions, FEC, interleaving or sophisticated concealment) are not applicable within the delay constraint. Many problems can be efficiently handled at the interface between the network and the applications; a good survey of network-adaptive mechanisms for video streaming can be found in [27].

There have been many studies concerning the performance of video traffic over data networks, using measurements or simulations. In [91], the authors studied H.261 and MPEG-2 over 10Base-T and 100Base-T Ethernets via simulation. [8] studied the effect of packet loss and delay jitter by sending RTP/UDP packetized MPEG video over the public Internet between sites in Europe and the US. In [5], the author experimented with rate and error control mechanisms for video conferencing using measurements. A recent study, [65], conducted very large scale measurements by streaming low-rate MPEG-4 video to clients in more than 600 US cities. In [1], the authors studied the effect of Differentiated Services mechanisms implemented in network testbeds, on video clips encoded using MPEG-1 and Windows Media Encoded Clips. The quality measure used was not in terms of loss statistics but in terms of the Video Quality Measurement (VQM), [75]. Each of the assessment studies have their own value as network environments, transmission scenarios, compression standards, and performance measures continuously evolve in time.

In this chapter, we consider video traffic transmitted over Internet backbones, with the same delay requirement as voice. The results obtained show that video and voice traffic can be carried at similar quality levels over these networks, using a limited -commonly used- subset of the available error resilience mechanisms. In Section 4.1, we describe the components of the simulated Video over IP (VIP) system. In Section 4.2.1, we study its performance, in terms of video quality and delay range, over some representative examples of backbone networks, similarly to what we did for voice. In Section 4.2.2, we discuss the benefit of rate-distortion optimized streaming over multiple paths. In Section 4.3, we discuss to what extent these observations hold for different VIP scenarios. In Section 4.4, we compare the simulations results to the quality degradation predicted by the analytical model in [90]. In Section 4.5, we conclude this chapter.

4.1 The Video over IP (VIP) System

In this section, we describe the simulated Video over IP (VIP) system, shown in Figure 4.1. Transmission of packetized video over the Internet is a large problem space, with

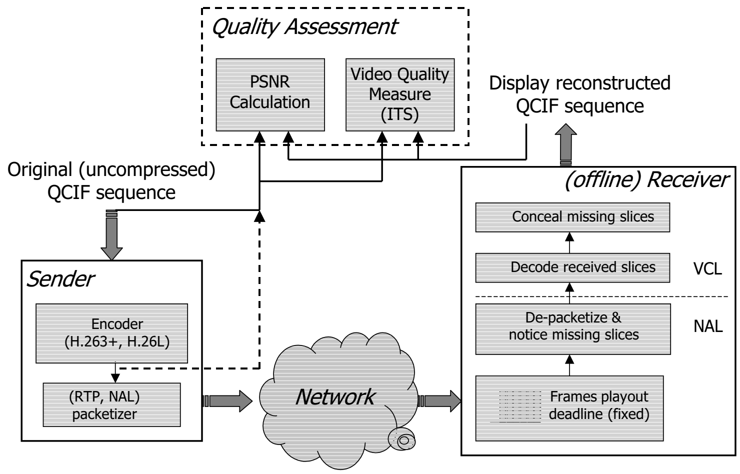


Figure 4.1: Simulated Video over IP (VIP) System.

several options for each component. Table 4.1 lists some of these options. This table is not complete (e.g. many compression standards and error resilience mechanisms are omitted); furthermore, each of the options can be implemented in various ways (e.g. there are many ways to perform concealment or adaptive playout). Different combinations in this problem space are appropriate for different network and application scenarios. We now describe and justify our specific choices, that constitute a simple yet reasonable transmission scenario for videoconferencing.

As already stated, the *application* of interest is interactive visual communication, such as video telephony and video conferencing, which immediately imposes a tight delay constraint, the same as for voice. There are many factors that contribute to the end-to-end delay including (i) digitization and encoding at the real-time encoder, packetization (ii) network delay (i.e. transmission and queuing) (iii) decoding and playout (buffering and display) delay at the receiver. A detailed discussion of the video specific delay components can be found in

Table 4.1: Evaluation space

Component/Issue	Options
Application	interactive, streaming
Standard	H.263+, H.26L
Video Sequence	content, frame rate, quantization, intra rate
Error Resilience	packetization, synchronization marks, concealment
Playout	fixed, adaptive
Network	many traces (Chapter 2)
Quality Measure	avg PSNR, VQM, subjective

[100]. Our focus is on the network and playout delay that depend on the backbone network under study. We assume that the other components are well streamlined and result in a roughly fixed amount of delay, which is the same independently of the network. The delay constraint imposed on the total end-to-end delay (for interactivity and lip synchronization) prevents the use of time demanding error resilience mechanisms.

As for *compression standard,* we started by using H.263 v.2, [34, 43], which is appropriate for video conferencing, and we switched to its successor H.26L, [51, 52]. Since the 1997, the ITU-T's Video Coding Experts Group (VCEG) has been working on a new video coding standard with the internal denomination H.26L. In 2000, H.263 v.3 got finalized and the project became H.26L ("L" for long term) but recently got renamed to H.264[1]. In late 2001, ISO/IEC MPEG's video group and ITU-T's VCEG decided to work together as a Joint Video Team (JVT) and to create common text for the forthcoming ITU-T Recommendation and for a new part of the MPEG-4 standard based on the working draft of H.26L. H.26L is considered today the state of the art both in compression (PSNR analysis superior to MPEG-4 and H.263 v.2) and in network-oriented design. It targets the whole spectrum of video applications, from low-rate video conferencing to video storage, streaming and video cinema. It is particularly designed for transmission over lossy environment, including a variety of error resilience mechanisms and a clear separation of the encoder into

[1]In this Chapter, we refer to the ITU-T H.264 standard as H.26L, to be consistent with the fact that this work is based on the March 2002 versions of the standard and test module, [51, 52], known then as H.26L.

a Video Coding Layer (VCL) and a Network Adaptation Layer (NAL); a good discussion on NAL can be found in [102]. VCL performs the compression using a block-based motion-compensated hybrid transform coder; it introduces some coding features, such as integer transform, arithmetic coding, additional picture formats and others. NAL, on the other hand, is responsible for packaging the bitstream into transport entities, namely packets in the case of IP networks. Error resilience tools include (i) placement of INTRA macroblocks in a loss-aware rate-distortion optimized way (ii) picture segmentation in independently coded slices (iii) packetization flexibility in compound packets (iv) 1-5 reference picture selection with/without feedback, which was optional in H.263 v.2, annex N (v) the parameter set concept to convey important information about the sequence (vi) a simplified data partitioning into Header, Intra and Inter partitions (vii) a non-normative error concealment in the current implementation, [52]. The reader is further referred to [51] for the detailed specification and to [103, 102] for meaningful packetization schemes. The combination of these well-known features (implementation details may differ from previous designs) make H.26L the state of the art today.

As our main *test video sequence* in the simulations, we used Foreman (QCIF, 134 frames at 10 fps frame rate, resulting in a clip duration of 13.4 seconds). We also used the Mother&Daughter sequence that has less motion and scene change. Both sequences are available online at [32]. We used only I and P frames, with ratio (unless otherwise noted) 1:10; we also tried higher Intra-rates. The quantization was Q =15 and 16 for I and P frames respectively, unless otherwise noted. The entropy coding method was CABAC and the number of reference frames was 2. The resulting rate of the Foreman sequence was 98 Kbps. In general, each image can be encoded into independent slices and one or more slices are packetized into packets using RTP. Examples of packetization schemes, which are effective in IP environments, can be found in [101, 102, 6]. We used slices that contain a fixed integer number of rows and we put each slice into a separate RTP packet. Unless otherwise noted, we used 3 slices per frame and one slice per packet. For appropriate Q, slices can fill

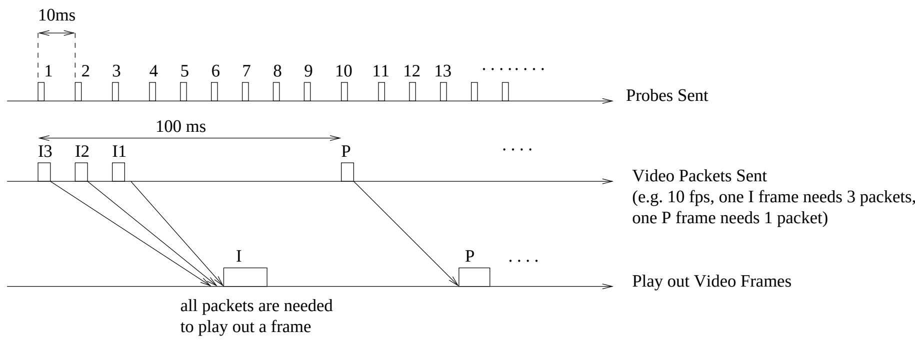


Figure 4.2: Assignment of probes to video packets. We consider that each video packet incurs the loss and delay incurred by the corresponding probe packet.

up entire 1500B packets. However, there are some packets are not entirely filled up. In those cases, we choose to keep the packetization scheme simple and incur the overhead, which is anyway negligible in this high bandwidth environment.

The packetized video sequence transmitted over Internet backbones, is subject to delay and loss. We use the delay and loss experienced by the probes described in Chapter 2, to simulate the delay and loss experienced by the video packets. The probes were sent every 10 ms. We assigned probes to video packets in two steps. For the 10fps sequence, we first assign every tenth probe to a frame, as shown in Figure 4.2. Similarly, for a 30fps sequence, we assign every third probe to a frame. Then, we assign the slices of the frame to consecutive probes. Thus, the finest granularity available in time is 10 ms; this can be interpreted as the time needed to produce a slice, as shaping at the source or as the transmission of a 1500B packet over a T1 link (actually 8ms). For example, a 10 fps sequence with one slice and packet per frame, uses probe numbers {1, 3 , 6, 9, ...} (spaced 30 ms apart, approximating the 33 ms between frames in the 30 fps sequence); a 10 fps sequence, with 3 slices per frame and one packet per slice, uses probes number {1, 2, 3, 10, 11,12, 20, 21, 22 , ... }.

In order to display a frame, we assume that all slices of this frame should be available and displayed at the same time. This has an impact on the delay budget of the packets: if the entire frame is assigned a *playout delay p* (from the time it was sent entirely until the time it was displayed), then, the deadline for the first packet of the frame to arrive is $p_1 = p$, for the second packet is $p_2 = p - 10ms$, and so on. Similarly to the discussion on voice playout, there are many ways one can play out video frames, following a fixed or adaptive algorithm. Adaptive playout for video is an ongoing research topic by itself and different algorithms are appropriate depending on the delay requirement of the applications; e.g. see [53, 54, 95]. Varying the playout rate for video is relatively straightforward by displaying frames faster or slower. However, in the video conferencing context, video playout is dominated by the voice playout, due to the lip synchronization and the pitch change caused by variable playout rate. For this study, we consider only fixed playout deadlines for video, at a range of low playout delays, in the ranges discussed in Section 3.1.2.4, appropriate for interactive voice.

We consider only a few simple *error resilience* mechanisms. First, we used *Intra-coded frames*, which interrupt the error propagation due to temporal dependencies between different frames. No additional Intra-coded macroblocks have been used though. Second, we used *slice* mode and packets that contain one slice each. Independently coded *slices* interrupt error propagation due to dependencies inside the same picture, such as differential coding of motion vectors. Third, we assumed that important information, such as the picture header and higher level information, should never be lost, according to the *Parameter Set Concept*. In practice this can be achieved by transmitting this information out of band, using a reliable protocol, or updating and sending it periodically. In simulation, this can be achieved by never dropping the corresponding packets. Finally, we used *error concealment* at the receiver. Missing packets/slices are marked with an error flag by NAL and trigger concealment by the VCL at the receiver. There are many ways that concealment can be implemented, some of which are discussed in the survey [101], by exploiting the inherent temporal and spatial correlation in any video scene and assisted by information provided by the compression and packetization schemes. Examples include various synchronization marks and repetition of motion vectors from previous packets in the RTP header.

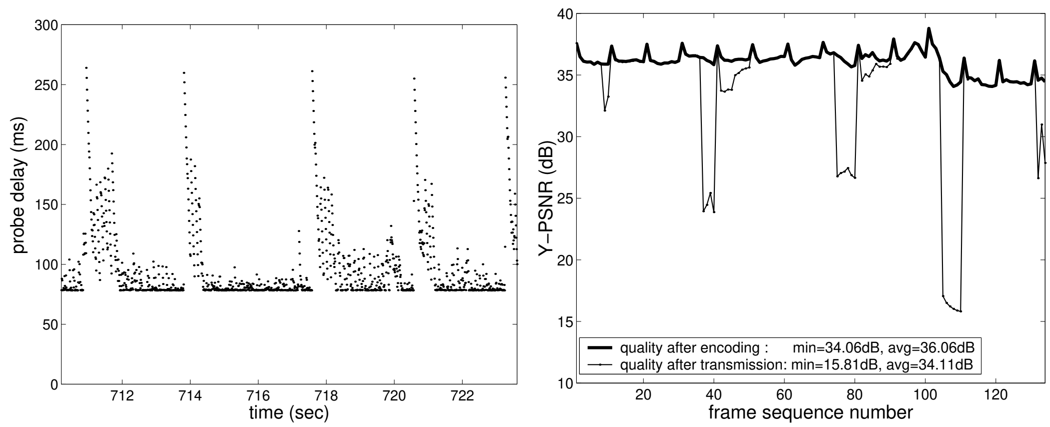
The first reason for considering only the above error resilience is simplicity: in practice these simplest mechanisms are typically used. The second reason is the low delay requirement, imposed by interactivity, which prevents time consuming mechanisms, such as e.g. interleaved packetization. Another constraint comes from the nature of the measurements that we use in simulation. These measurements have been collected over networks that provided best effort service to all probes. This implies that e.g. scalable coding combined with priority dropping cannot be studied using these measurements.

The H.26L software, in particular appendix IV in [52], currently implements a non-normative *error concealment* that we used in simulation and now describe for completeness.

- The concealment algorithms are macroblock (MB) based. A map with the status of all MBs is maintained for each frame. The status of a MB can be "correctly received", "lost" or "concealed". The area of the frame marked as "lost" is concealed MB-by-MB in a column-by-column and border-to-center order (the middle part of a picture is more complex to conceal). After a MB is concealed, it may be further used for concealing its neighbors, if the number of correctly received neighbors is insufficient.

- Lost areas in INTRA-coded frames are concealed spatially, as proposed in [56]. Each pixel value is estimated as a weighted sum of the boundary pixels of the selected adjacent MBs. The weight of each boundary pixel is inversely proportional to the distance from the pixel.

- For lost MBs in INTER-coded frames, it is more efficient to estimate their motion vectors (MVs) from temporal and spatial neighbors, instead of operating in the pixel domain. First, the motion vectors of the received slices are investigated. If the average motion vector is found less than a pre-defined threshold then all lost slices are concealed by copying from co-located positions in the reference frame. Otherwise, the motion vectors of each lost macroblock are predicted using the algorithm proposed in [61]: the MVs of the neighbors are tried and the one that maximizes the spatial smoothness is chosen. The concealed MV is used for motion compensation with respect to the



(a) 13.4 sec on path THR-P1-ASH, on Wed 14-15:00)

(b) Resulting quality (PSNR per frame), using fixed playout at 200 ms

Figure 4.3: The Foreman sequence over an example network trace.

reference frame without additional operations in the pixel domain. When multiple reference frames are used, the same mechanism is used but with respect to the used reference picture.

In the assessment of the VIP system described above, we need an appropriate quality measure. As with voice quality, the ultimate test for video quality is subjective testing and rating provided by human viewing panels. However, the spatial and temporal impairments in a reconstructed video sequence can be quantified in an objective way. Such a measure is the mean square measure (MSE) between two sequences. The most widely used video quality measure is the peak signal-to-noise ratio (PSNR), defined in Equation (5.1). It is appropriate for capturing impairments due to compressions (e.g. due to quantization and motion compensation). In addition, it is also widely used to assess the effect of transmission impairments. For each reconstructed frame, first the difference in luminance (Y) from the original frame, is calculated pixel-by-pixel and then an average over the entire frame is taken:

$$PSNR(dB) = 10 \cdot log \frac{255^2}{E\{(Y_{original} - Y_{decoded})^2\}} \qquad (Eq.5.1)$$

For example, in Figure 4.3(b), the thick line shows the PSNR for each frame of the encoded Foreman, compared to the original uncompressed QCIF. The quality is not perfect due to quantization and motion compensation; depending on the content of each frame, the quality varies slightly. If the sequence is transmitted over the trace shown in Figure 4.3(a) and fixed playout of 200ms is applied, then parts of frames (i.e. entire packets and thus entire slices) are lost and they are partially recovered using error concealment at the receiver. The quality of the reconstructed sequence is shown in thin line in Figure 4.3(b): it can be at best as good as the encoded sequence and it can be as low as 15dB. The quality of the entire sequence is usually captured using the average of PSNR over all frames. The average PSNR is our main quality measure. However sometimes we also report the minimum PSNR across the sequence.

Note that MSE and PSNR are almost exclusively used as quality measures, partly because of their mathematical tractability and partly because of the lack of better alternatives. Designing objective distortion measures that are easy to compute but correlate well with perceptual distortion is still an open research issue, [33].

We also used an objective measure developed by the Institute for Telecommunication Sciences (ITS), [33], called the Video Quality Measure (VQM). It belongs to a family of measures based on feature extraction: (i) temporal and spatial features are extracted (ii) the distance of the original and reconstructed video sequences is computed in an appropriate scale based on these features (iii) this distance is mapped to a subjective score. Compared to PSNR, this measure helps identifying the nature of an impairment; it also has various modes tuned specifically for video-conferencing and the kind of impairments relevant to it. The reader is referred to [33, 75] for details. There is actually a trend for developing objective measures that takes as input the original and distorted video sequence, or a measurable network parameter, and maps it to user perceived quality. We discussed this issue in the introduction, where we mentioned that such measures have been successfully developed for voice quality assessment while they are still in progress for video quality assessment.

PSNR range	Image Quality
40dB < PSNR	Excellent Quality
30dB < PSNR < 40dB	Good Quality
20dB < PSNR < 30dB	Poor Quality
PSNR < 20dB	Unacceptable Quality

Figure 4.4: Rough mapping of PSNR to Image Quality, according to [100]

For a comparison to voice quality, one should bear in mind the voice quality classes defined in Figure 3.2. A similar rule of thumb for mapping PSNR to image quality is given in [100] and is reproduced in Figure 4.4. According to [100], $PSNR > 40dB$ indicates excellent image (i.e. very close to the original); $30DB < PSNR < 40dB$ means a good image (i.e. the distortion is visible but acceptable); $20dB < PSNR < 30dB$ is quite poor; finally, $PSNR < 20dB$ is unacceptable (e.g. when all frames are lost during the outage of 19.78 sec in Figure 4.12, the PSNR is 15 dB in Figure 4.10(a)).

4.2 Numerical Results

In this section, we consider representative hours on paths of four different providers. First, we discuss the quality of video conferencing sessions during those one-hour periods. Second, we discuss the benefit of streaming video over multiple such paths.

4.2.1 Performance of Voice and Video over Typical Paths

We consider the following four traces that have completely different delay and loss characteristics, discussed in Chapter 2.

- Provider P_1, path THR-P_1-ASH, on Wednesday at 14:00-15:00. This trace (shown in

Figure 4.5(a)) has high delay variability; see Sections 2.5.

- Provider P_2, path EWR-P_2-SJC, on Thursday at 20:00-21:00. Ten minutes of this trace are shown in Figure 2.11: there are two periods of higher delay accompanied by loss event; see Section 2.5.

- Provider P_3, path EWR-P_3-SJC, on Thursday at 10:00-11:00. This trace (shown in Figure 4.11) has a small delay variability but it has single packets lost regularly, on average every 5 seconds (see Figure 2.16). This behavior is typical of the entire provider P_3; see Section 2.5.

- Provider P_4, path EWR-P_4-SJC, on Wednesday at 21:00-22:00. This trace, shown in Figure 2.42, exhibits clusters of 250-300 ms high spikes every 60 sec and a long outage. This behavior is consistent across all 6 paths of provider P_4; see Section 2.5.

We consider the 13.4 seconds 10fps Foreman sequence repeated 268 times, to cover one hour. For each of the 268 sequences, we consider both loss in the network and due to late arrival and we compute the average (and minimum) PSNR, as in Figure 4.3. Similarly to the steps we followed for voice, we plot: (i) the quality for each sequence, as in Figure 4.5(c) (ii) the cumulative distribution (CDF) for the quality all 268 sequences in the one-hour period, as in Figure 4.6 and (iii) the loss-delay tradeoff, as in Figure 4.7.

We also compare the video to the speech quality for the same periods, considering 2-seconds intervals (as in Figure 4.8 and 4.9) , and applying the translation from loss to voice-MOS, using the data of Section 3.1.2.3.

One hour on provider P_1

Figure 4.5(a) shows one hour on path THR-P_1-ASH trace; this trace has been studied in the Measurement Chapter and has also been used as an example for evaluating playout algorithms. Figures 4.5(b) and (c) show the percentage of loss and the resulting PSNR per sequence, considering different values of fixed playout delay. Figure 4.6 show the statistics for the quality of these 268 video sequences during this hour: the larger the playout delay, the smaller the loss and the higher the video quality. Figure 4.7 shows the same loss-delay tradeoff, in a way similar to the one for voice (see Figure 3.25). The different curves correspond to percentiles of the 268 video sequences in that hour; with sufficiently high fixed playout (at 200ms) all the video sequences experience good quality (degradation less than 10dB) and 50% of the sequences experience perfect quality.

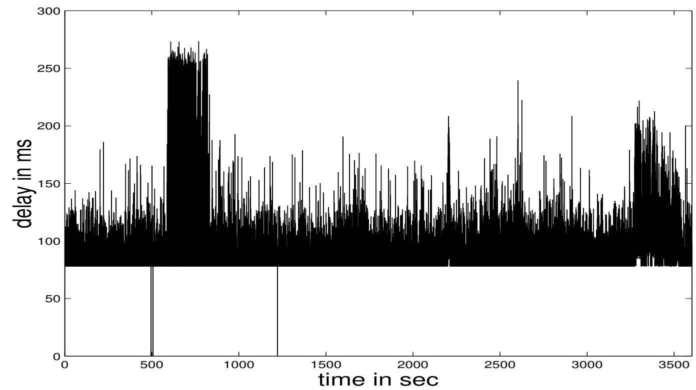
The next two figures, take similar steps for voice over the same trace. The speech quality is calculated for successive 2 second intervals, to resemble the assessment of video quality over 13.4 seconds video sequences. Figure 4.8 shows that voice and video quality degradation is in the same order of magnitude (taking into account the quality classes of Figure 3.2 and 4.4 for video), for the same fixed playout delay. Figure 4.9 is comparable to the statistics for video quality in Figure 4.6.

One hour on provider P_2

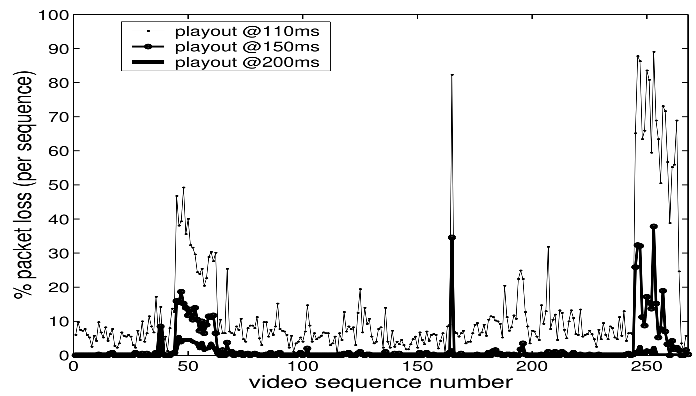
Similarly, in Figure 4.10, we consider various values of fixed playout delay and we show the video quality for each sequence as well as their distribution. The worst part of this hour is during the high delay and loss period of Figure 2.11, resulting in the two first dips of PSNR. For adequately high playout delay (i.e. 120 ms), video quality is almost perfect, except for the 30 second loss period, which coincide with the three worse calls.

One hour on provider P_3

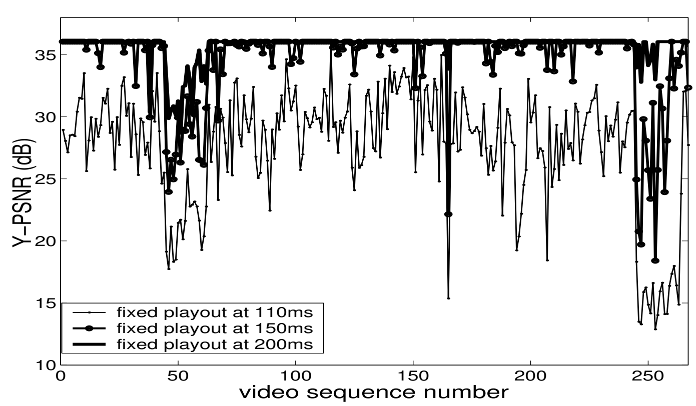
This is an interesting trace in terms of loss: single packets are lost every 5 seconds on average. The loss-free durations follow the exponential distribution shown in Figure 2.16. In addition, there is a long loss duration for 19.78 seconds continuously. Part of the trace is shown in Figure 4.11(a). Statistics for the delay spikes and distances on this trace are shown in Figure 4.11(b) and (c). The delay on this trace varies between 31.6 and 84 ms.



(a) Delay of individual packets



(b) Packet loss rate per sequence



(c) average PSNR (per sequence)

Figure 4.5: One-hour trace of provider P_1 (Wed 14:00-15:00, path THR-P_1-ASH).

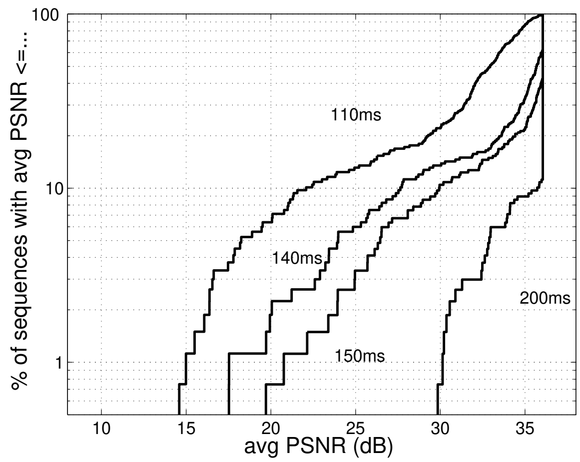


Figure 4.6: Statistics for the quality of video sequences, during one-hour period on provider P_1 (Wed 14:00-15:00, path THR-P_1-ASH), for different playout delays.

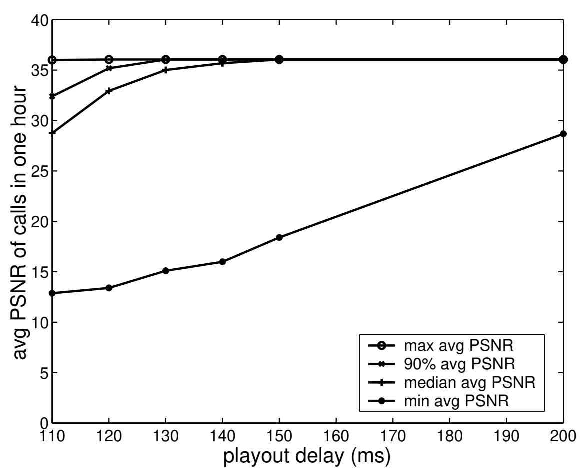
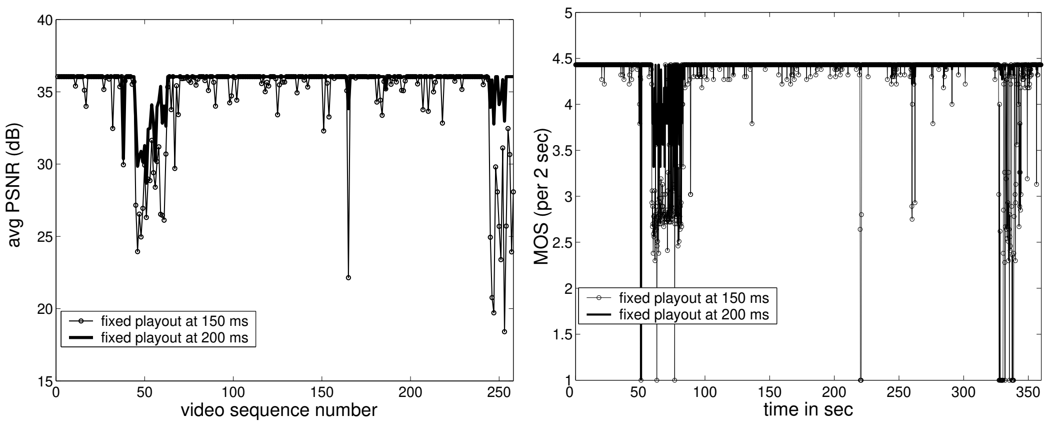


Figure 4.7: Tradeoff between video quality and delay, for the one-hour trace of provider P_1 (Wed 14:00-15:00, path THR-P_1-ASH).



(a) Video quality (average PSNR) for each sequence

(b) Speech quality (MOS) per 2 sec

Figure 4.8: Time varying, voice and video quality, over the on-hour P_1 example trace

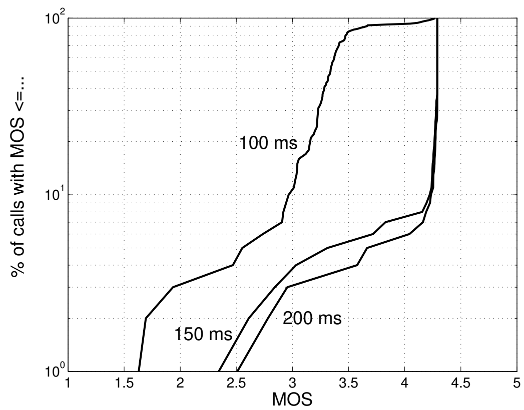
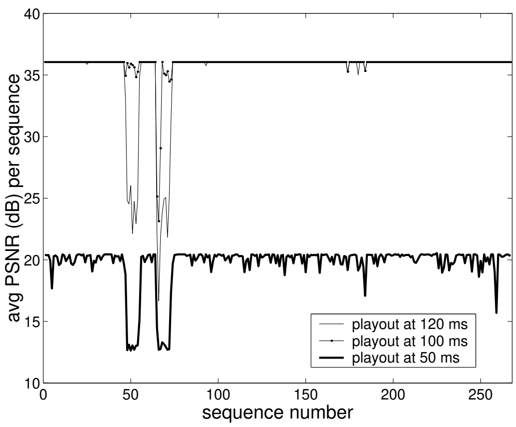
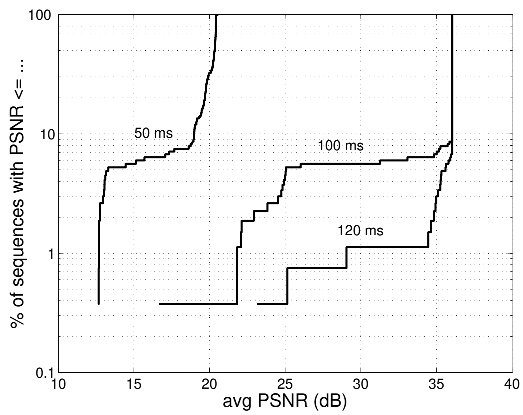


Figure 4.9: Statistics for the voice quality (computed per 2 sec intervals), for various fixed playout delays, over the on-hour example P_1 trace



(a) PSNR per video sequence for different fixed playout delays

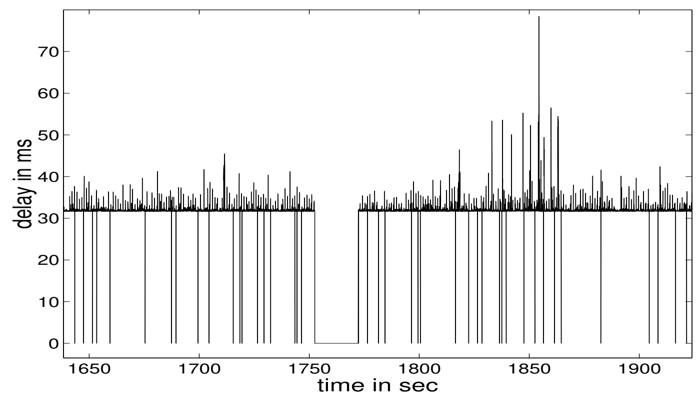


(b) Cumulative distribution function (CDF) for the average PSNR of the 268 video sequences, for different playout delays

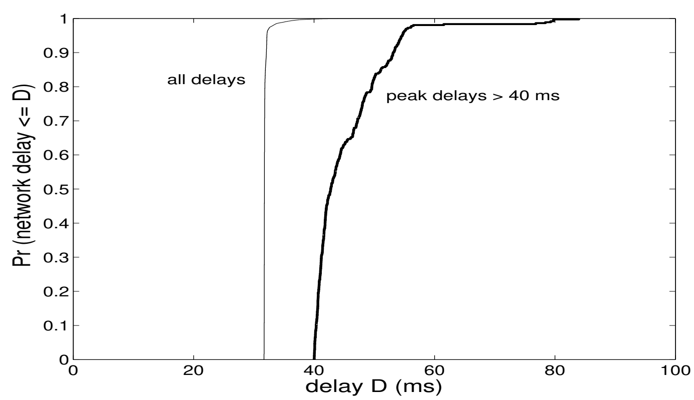
Figure 4.10: Video quality during the one-hour period on provider P_2 (Thu 20:00-21:00, path EWR-P_2-SJC).

Therefore, when a playout delay at 100 ms is applied, there is no loss at all due to late arrivals, and only network loss has an effect. Figures 4.12(a) and (b) show that video and voice quality are good for the entire hour, except for the outage period. Apart from the outage, the lost packets were spread so far apart, that all the 2 second voice intervals had either 0, 1 or 2 packets lost, which is negligible (0-1% loss). The video sequences have an INTRA-coded frame every 1 second, which is shorter than the average distance between losses (one packet lost every 5 seconds); therefore, there is time to refresh between losses. Even the minimum PSNR, corresponding to the frame that incurred the packet loss and its dependents, is high enough. The reason for this is the loss of a single packet results in the loss of only one out of the three slices per frame; in that case the concealment can help significantly to recover.

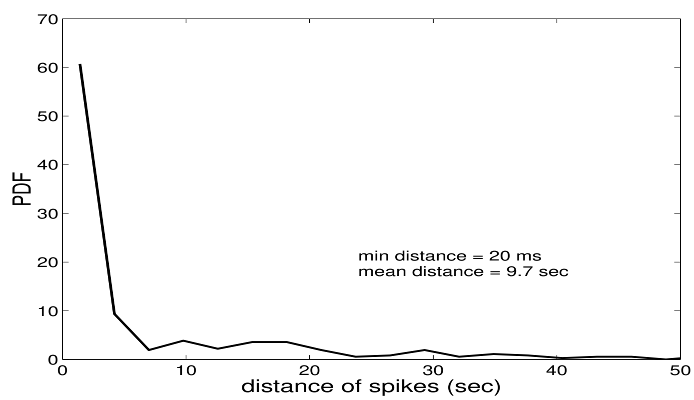
When a tighter fixed playout delay of 40 ms is applied per frame, the average PSNR does not significantly degrades for video although the minimum PSNR is more often lower, see Figure 4.12(c)). There is a more noticeable decrease in the voice quality (see Figure 4.12(d)),



(a) Part of the trace where the outage happens (1978 consecutive packets)

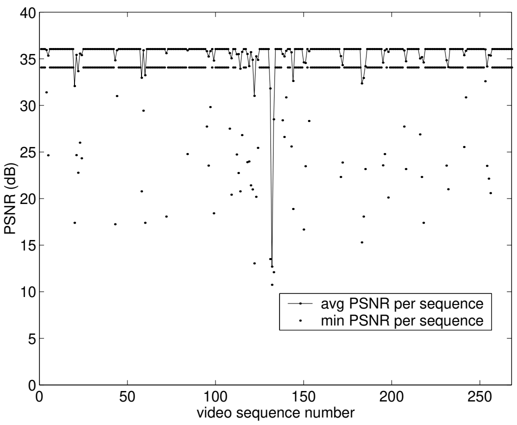


(b) Height of the delay spikes

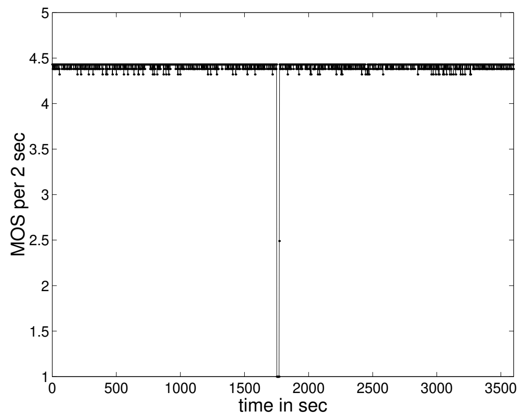


(c) Frequency of the delay spikes

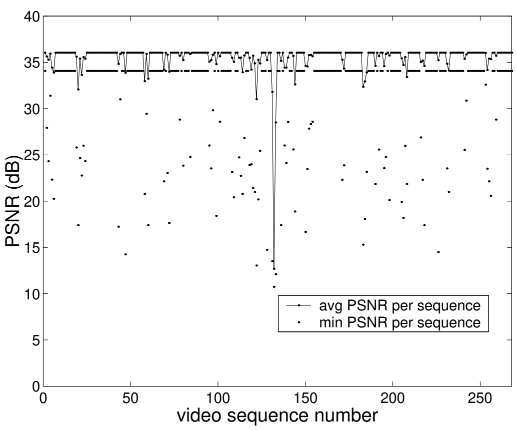
Figure 4.11: One-hour period (Thu 10:00-11:00) of path EWR-P_3-SJC



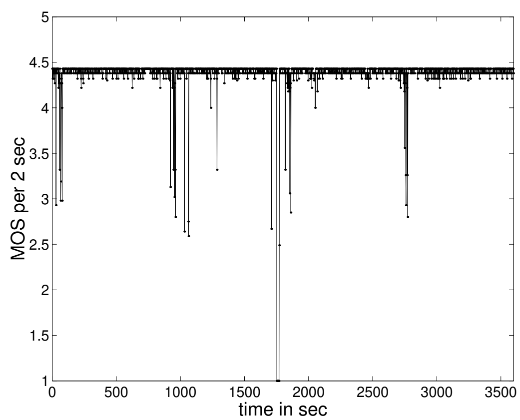
(a) Video quality for playout at 100ms



(b) Voice quality for playout at 100ms



(c) Video quality for playout at 40 ms



(d) Voice quality for playout at 40 ms

Figure 4.12: Voice and video quality over the one-hour P_3 example trace (EWR-P_3-SJC, Thu 10:00-11:00) for playout at 100 and 40 ms.

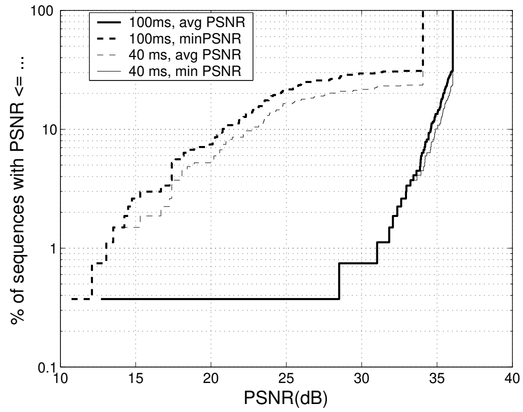
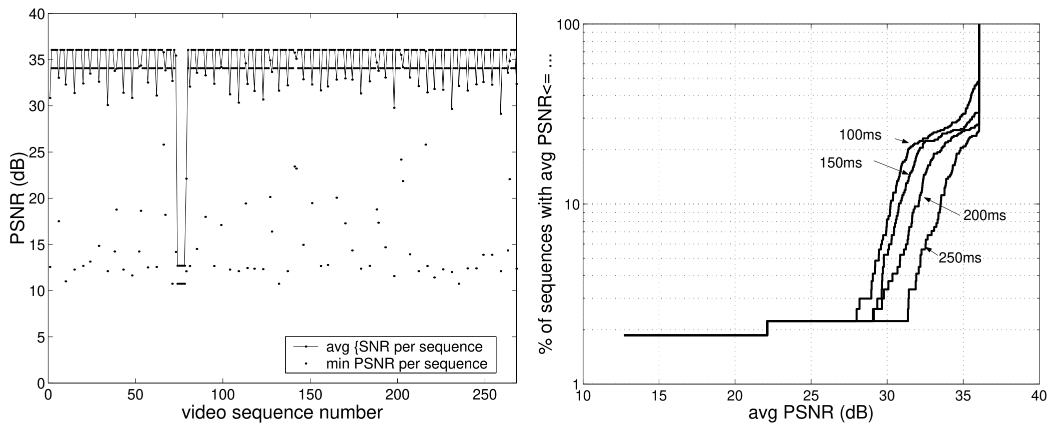


Figure 4.13: Video quality statistics for the one-hour P_3 example trace.

which at first seems counter-intuitive. The difference can be explained by the following facts. First, the video sequence is 13 seconds long, which is a longer period to take a time average than the two seconds intervals considered for voice. Second, there is a voice frame every 10 ms, while there is only a video stream every 100ms. As a result, spikes above 40 ms (which are only 0.94 of all packet loss but can lead to 40 ms loss) are definitely perceived by voice but not necessarily perceived by video. The height and distance of these spikes is shown in Figures 4.11(b)and (c) respectively. Third, all voice packets have the same 40 ms end-to-end delay constraint; however, the two of the three packets of each frame have more relaxed deadlines (40, 50 and 60 ms). Figure 4.13 shows that the statistics for video quality are practically the same for 40 and 100 ms.

One hour on provider P_4

This trace is shown in Figure 2.42 and it is well studied in previous chapters: clusters of 250-300ms are repeated every 60 seconds. Every sequence that coincides with such a cluster has a lower PSNR in Figure 4.14(a). Remember, however, that each sequence is 13.4 seconds long, while the clusters are repeated every 60seconds; therefore there are some sequences with good quality. In addition, there is a 70 seconds outage resulting in the dip in PSNR in Figure 4.14(a). Due to the equal height triangular spikes in a cluster, packets are always



(a) (average and minimum) PSNR per video sequence, for fixed playout at 250 ms

(b) Cumulative distribution function (CDF) for the avg PSNR of the 268 video sequences, for different playout delays

Figure 4.14: Video quality during an one-hour period on provider P_4 (Wed 21:00-22:00, EWR-P_4-SJC).

dropped every 60 seconds, even when we increase the delays from 100 to 250 ms. This bad performance is consistent with the bad performance of voice over this path (compare to the CDF for voice, shown in Figure 3.46).

4.2.2 Rate-Distortion Optimized Streaming over Multiple Paths

So far, we evaluated the quality of video sequences sent over a single representative path. However, better performance can be achieved if more than one paths can be used, and appropriate choices can be made among the available paths.

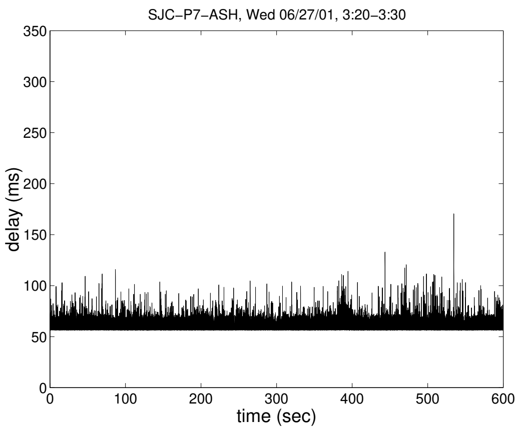
Consider for example rate-distortion optimized video streaming framework developed in [13]. This framework considers (i) the graph of dependencies between the data units of a video stream (ii) the available transmission opportunities, at the sender, at the receiver or at an intermediate proxy server (according to the extension proposed in [10]) (iii) a bandwidth constraint to be met at minimum distortion. It then chooses optimally the transmission policy of the multimedia data units.

In this framework, multipath streaming provides more transmission opportunities than streaming over a single path, and thus a chance for further optimization. [10, 11] applied the multipath concept to the rate-distortion optimized video streaming framework developed in [13]. The rate-distortion optimized streaming over multiple time varying channels is referred to, in [11] and in this section, as RaDiO-multipath.

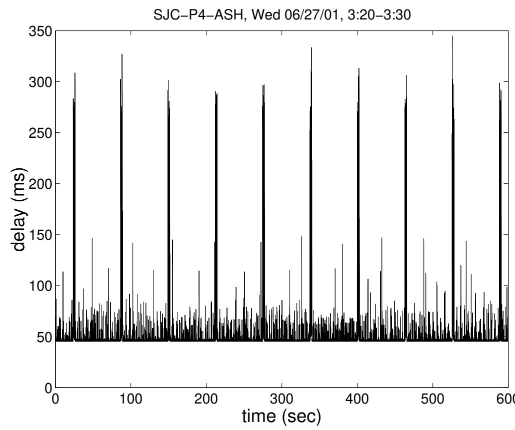
The scheme has been evaluated for audio (in [10]) and video (preliminary results in [27]) streaming, by modeling the two independent channels with a two-state Markov model. We now evaluate[2] the RaDiO-multipath scheme using measurements for two paths between SJC and ASH, shown in Figure 4.15(a) and (b). The Foreman sequence (SNR scalable representation of the first 130 QCIF frames, frame rate at 10 fps and GOP with one I and 9 P-frames) is streamed using the sender-driven RaDiO streaming framework, [13]. Figure 4.15(c) shows the benefit for RaDiO streaming with a startup delay of 100ms. Using both available traces performs better than using only one, in a rate-distortion sense: better quality is achieved for the same bitrate, or the same target quality is achieved at less bitrate. Video quality is improved over the entire range of transmission rates. This graph has been obtained by averaging over many channel realizations, i.e. by starting the Foreman sequence at different times in the 10 minutes trace. Because the behavior of the paths changes with time and the two paths are very different from each other, there is benefit from multipath. However, if a larger startup delay is allowed, e.g. 400ms, then there is no benefit from using multiple paths as it is shown in Figure 4.15(d).

In general, the improvement from RaDiO-multipath is significant for bursty channels and uncorrelated paths. For more on path diversity, the reader is referred to Section 3.4.

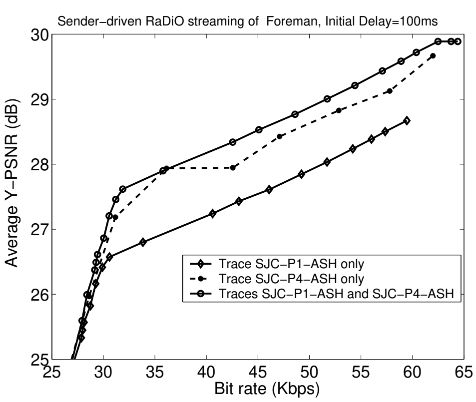
[2]The author is grateful to Jacob Chakareski for the simulations of rate-distortion optimized streaming over multipath paths, according to the scheme proposed in [11], as well as for useful discussions.



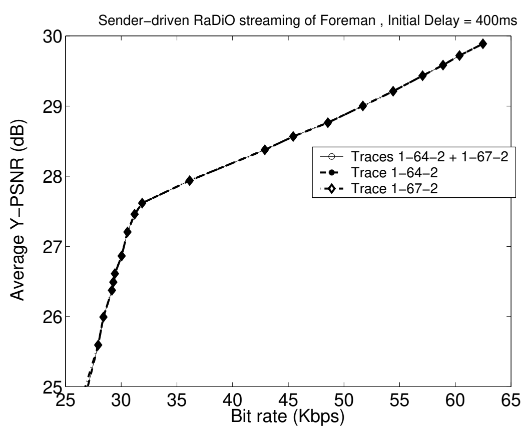
(a) Path: SJC-P1-ASH. Period: Wed 3:20-3:30.



(b) Path: SJC-P4-ASH. Period: Wed 3:20-3:30.



(c) Initial delay 100ms. There is gain from RaDiO-multipath, for the entire range of rates and distortions



(d) Initial delay 400ms. There is no gain from RaDiO-multipath.

Figure 4.15: Sender-driven RaDiO video streaming of the Foreman sequence (QCIF, 130 frames at 10fps) over two different paths connecting SJC and ASH, on Wed 06/27/01: 3:20-3:30.

4.3 Different VIP Scenarios

Transmission of packetized video over the Internet is a large problem space. The results we presented above are obtained for the subset of this space, described in Section 4.1. A natural question to ask ourselves is to what extent, are the above statements true for different scenarios? For example, would the quality of a 6Mbps video stream degrade similarly? What about a sequence with different content, different coding parameters, different error resilience mechanisms or starting at a different point in time?

High-end video: A 6Mbps MPEG-2 vs. a 100Kbps H.26L stream

With respect to the similar degradation of a higher quality video stream, the answer is yes for a 300Kbps H.26L (using the same error resilience modes) and no for a 6Mbps MPEG-2 stream (because of the difference in error resilience modes and not because of the bitrate itself). Better video quality can be achieved by using (i) higher resolution (ii) finer quantization and (iii) higher frame rate. All three improvements imply that more bits are needed to encode the higher quality stream. One could try to get the highest achievable quality using H.26L, by using the largest image size, the finer quantization and 30 instead of 10 fps. In that case we would get the "high end" quality of H.26L resulting in a few hundreds of Kbps. If one wants even higher quality, then he should switch to another standard that is designed for higher quality applications (such as DVD, movies on demand, TV, digital broadcast). MPEG-2 is then the appropriate standard, with bitrates in the order of 2-15 Mbps for video on DVD and even higher rates for HDTV.

However, it is not the difference in data rate by itself, that leads to a difference in quality degradation in a lossy transmission environment. It is the difference in the standard and its error control and recovery mechanisms, that can amplify or limit the effect of loss. An MPEG-2 stream will fail in a lossy environment because it is not designed for it. In general, error resilience mechanisms insert markers and redundant information in the bitstream to assist the receiver in the case of loss, to recover the lost parts using the neighboring (temporally and spatially) received parts. Examples include intra-refresh, layering, forward error correction, reference picture selection, picture segmentation, various packetization and concealment schemes, retransmissions and many others; a good survey of error resilience mechanisms can be found in [101]. Some, but not all, of these mechanisms can be replicated by using RTP and the specific RTP profiles or other proprietary transport-level protocols. For example, redundant repetition of important information, like the motion vectors, can be carried on RTP headers; retransmissions can be used at transport-level. However, the H.26L is inherently a more powerful standard for lossy environments: it specifies more error control mechanisms and a separate Network Adaptation Layer (NAL) that allows for effective packetization.

Let us now assume the same underlying loss process for both H.26L and MPEG-2 streams. The packets of each stream can be thought of as "sampling" this process. The MPEG-2 stream needs more packets: each frame needs more packets than if it were compressed using H.26L, and in addition, there are additional frames at higher frame rate. On one hand, larger number of packets means that the underlying loss process is sampled more often. Bursty loss in the short time scale is more probable to be sampled with frequent samplings. Bursty loss that lasts longer (in the order of tens of seconds which is quite often in the traces) will be noticed similarly by frequent and sparse sampling. On the other hand, larger number of packets (larger, segmented picture and more frames) can potentially contribute to the better concealment of the same loss pattern, as opposed to, for example, having an entire frame lost in one packet. However, the larger number of packets by itself will not help without the segmentation and concealment algorithms that could have been used anyway, even for the lower bitrates. Therefore, the quality degradation depends mainly on the error control mechanism, rather than on the rate.

An example of error resilience mechanism, not applicable in our context, is scalable coding. Let us refer to two studies whose goal was to (i) show that non-scalable streams degrade fast even with a small amount of loss and (ii) advocate the use of scalable video combined with priority dropping to achieve graceful degradation. The results were qualitatively the same for MPEG-2 and data partitioning in [57] and for H.263+ and SNR scalablity in [31].

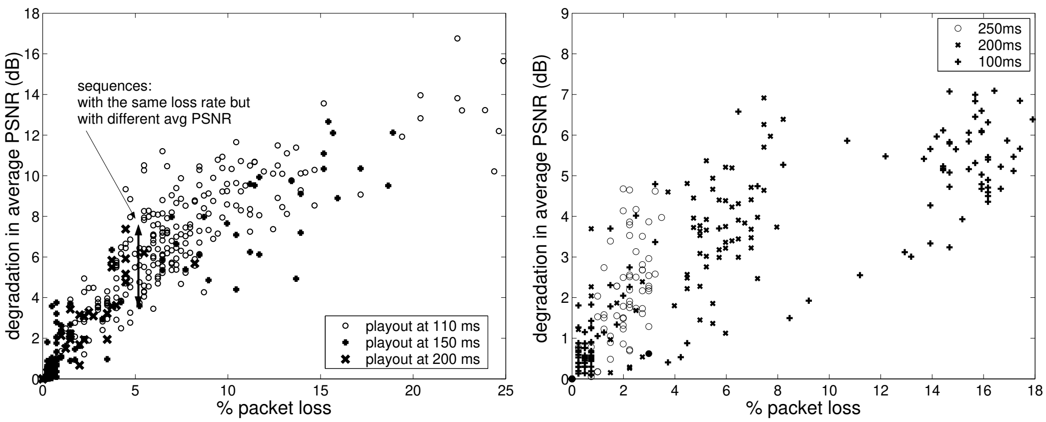
The results presented in this chapter, would be qualitatively similar for higher bitrates, as long as the error resilience mechanisms used are the same. For video-conferencing applications the H.26X (where $X \epsilon \{1, 2, 3, 3+, 3++, L, 4\}$) standards are the appropriate ones, due to the low processing times needed to meet the delay constraints and the error control mechanisms defined specifically for lossy environments.

Other Scenario Variations

We also varied some other parameters of the VIP system. We first varied the starting point of the sequences. The reason is that the same loss pattern may have a different effect on different parts of a sequence; there was no difference on the statistics of average PSNR. Second, we varied the content of the video sequence; "Mother & Daughter" is a sequence with less motion, and better concealment can be achieved in the case of loss; the variations of quality in time were similar, their magnitude was smaller. Third, we varied the ratio of I to P frames from 1:9 to 1:0. There was indeed improvement in quality when using more I frames, because there was no error propagation. However, there was no significant gain, because the noticeable loss periods were longer than the period between I frames.

4.4 Loss-to-PSNR Relation

So far, we simulated the transmission of a video sequence and we calculated the PSNR of the distorted sequence. The second natural question to ask ourselves is whether there is a simple relation between packet loss rate and degradation in quality, similarly to the Emodel curves in section 3.1.2.3. Indeed, [90] calculated analytically the time averaged distortion D_v, for a linear decoder and temporally uncorrelated errors. They provided a linear relation



(a) Trace P1 (b) Trace P4

Figure 4.16: Relation between Packet Loss Rate and PSNR, for two example traces.

between the residual word error rate P_L and the time averaged distortion D_v of a video sequence: $D_v = P_L \cdot function(sequence,\ Intra\,rate)$; the slope depends on the sequence content and on the INTRA rate. They found that this model was accurate for low residual error rates (i.e. $P_L < 10\%$) and for stationary, or “typically” varying, errors .

In Figure 4.16(a) and (b), we plot the packet loss rate (including network loss and loss due to late arrival) and the average degradation (in PSNR) of the 268 sequences transmitted over the example traces P_1 and P_4 respectively, for different values of fixed playout. The loss rates and PSNR values are those obtained with the simulations of Section 4.2.1. Each set of data points corresponds to one fixed playout value. These data sets experience different loss patterns; for example trace P_4 has burstier loss than trace P_1 due to the periodic clusters of high spikes. Even for the same trace, data sets corresponding to lower fixed playout, experience higher and thus more correlated loss.

We observe that the average degradation in PSNR seems to grow linearly with loss rate, as predicted by the analytical model, [90]. However, there is a large range of degradation values corresponding to the same loss rate. For example, let us look at Figure 4.16(a) and the data points for 200 ms. Sequences that have the same loss rate can have a degradation

Table 4.2: Fitting the $(loss, PSNR)$ pairs to a straight line: $PSNR = a_0 + a_1 \cdot loss\,probability$. |error| is the norm of the residual error after the fit.

example trace P_1				example trace P_4			
Data Set	a_1	a_0	$\|error\|$	Data Set	a_1	a_0	$\|error\|$
playout at 110 ms	0.24	5.14	34.3	playout at 100 ms	0.2495	0.5221	16.5713
110ms for $loss < 10\%$	**1.01**	**0.67**	14.01	playout at 150 ms	0.2491	0.5346	18.8638
playout at 150 ms	0.54	0.33	16.82	playout at 200 ms	0.2425	0.5229	18.5558
150 ms for $loss < 10\%$	**1.49**	**0**	1.82	playout at 250 ms	0.2357	0.3844	14.5263
playout at 200 ms	**1.07**	**0.02**	6.12				
(**anyway** $loss < 10\%$)							

in PSNR from a large range (approximately 50% of the maximum value).

If we consider data sets with lower loss rate, then the fit is better, in accordance to [90]. Indeed in Table 4.2, the sequences experiencing loss rate less than 10%, shown in bold, are a better fit to the $PSNR = 0 + a_1 \cdot loss\,probability$ line. The slope depends not only on the sequence but also on the loss pattern. The fit for data sets over the example trace P_4, was bad, because it was far from the assumptions of the model. Indeed, the periodic delay pattern leads to high loss concentrated in a short time period every 60 seconds.

The purpose of this section is to show that simulation is useful in real traces where there are no guarantees for the model of the packet loss process. The linear relation between loss and PSNR as predicted by the analytical model, is confirmed at a first approximation. However, it seems that the burstiness of the loss pattern should also be taken into account.

4.5 Summary and Discussion

In this chapter, we studied the quality of video traffic with tight delay constraints, transmitted over Internet backbones, using a limited, commonly used subset of the available error resilience mechanisms. We compare the video to the speech quality over the same traces, with the same tight delay constraints imposed by interactivity and we find them to incur similar degradations. We also touched upon the rate-distortion optimized streaming over multiple paths; we found it to bring a significant gain, especially when tight delay constraints are imposed, which is the case for the interactive applications of interest. We finally compared the simulation results to the quality degradation predicted by the analytical model in [90], confirmed the model at a first approximation and demonstrated the need to also take the loss pattern into account.

The author would like to note that the assessment presented in this chapter is far from conclusive or easy to generalize. Video communications is a much larger problem space than voice communications, due to the dependencies in the datastream and the numerous sophisticated error resilience and playout mechanisms proposed. A small subset of this problem space has been studied here. However, it is encouraging to find video quality at good or acceptable quality levels, similarly to voice. This is a confirmation that video communications over the Internet is already a sound approach. It is even more encouraging to see that this is the case, even when using a small number of the available mechanisms at the video application level. The performance can improve significantlly (i) if more sophisticated mechanisms are used for video transmission and/or (ii) when the problems identified in today's backbones are fixed by the network operators. In particular, video playout scheduling would help, as most of the degradation was mainly due to the delay patterns combined with the fixed playout.

An interesting area for future research is the topic we touched upon in Section 4.4, i.e. the translation of loss to video quality degradation. Work along these lines has already been done for voice and converged into objective voice quality measures and the Emodel standard, as discussed in length in the VoIP chapter. However, the same issues have not been solved for video yet. The pioneering paper, [90], modeled analytically the error propagation and the relation of loss percentage to video quality degradation. People from the same group are currently refining this model to include the effect of different loss patterns. The evolution of this work to an Emodel type-of-standard for video would be both useful for network planning and technically challenging, due to the complexity of video traffic.

Chapter 5

Conclusions and Future Directions

In this work, we assessed the ability of today's Internet backbones to carry multimedia communications. A key asset in our study is the use of network measurements collected over the backbone networks of major ISPs across the US. In addition to the measurement study, we simulated the transmission of voice and video traffic over these networks, and we assessed their performance, using perceived quality measures (in the case of voice).

In general, backbone networks are sufficiently provisioned and thus expected not to cause significant degradation to data traffic. However, our findings indicate that this is not necessarily the case for multimedia traffic. In fact, packet loss and delay characteristics are not consistent across all backbone networks. There are already some backbone networks that exhibit good characteristics, leading to a confirmation that supporting packet voice and video over an integrated network is a sound approach. Other backbone networks exhibit undesirable characteristics that could not be accommodated with any of the measures introduced today. These characteristics include large delay spikes (in the order of 200 to 700 ms), periodic spike patterns, high loss periods and outage periods correlated with changes in delay, loss events happening simultaneously on many paths.

Most of the problems identified seem more related to reliability (e.g. link failures, outages due to reconfiguration time for a routing change), network protocols (e.g. routing protocol exchanges or other control traffic) and router operation (e.g. debug options, "router vacations"), rather than to traffic load. Section 2.15 discussed in detail the possible causes and remedies for these events. Therefore, for rendering backbone networks ready for multimedia traffic, more effort should be put on reliability and on understanding the network operation, rather than on devising Quality-of-Service (QoS) mechanisms, at least in these high bandwidth environments. This is a useful direction to be explored by people with privileged access to routers and network operation. We had only access to edge-to-edge measurements; by observing the patterns, we commented on their possible causes.

Action for improving voice and video performance can be taken in the network as well as at the end-systems. As long as the problems in the network remain below a certain magnitude, measures at the end-systems can greatly help to mitigate their effect. An important factor for the overall perceived performance turns out to be the network delay variability which can be appropriately handled by playout scheduling at the receiver. Our initial intention was to consider some realistic playout schemes, as part of the end-to-end VoIP system under evaluation. We first considered fixed playout for a range of delay values, as a benchmark for comparison, and then a moving average-based adaptive scheme. In the process, we realized (i) that delay follows spike patterns, consistent per path which should be learned by the adaptive algorithm and (ii) that there is a tradeoff between delay and loss that can be exploited to maximize the overall perceived quality. We proposed three algorithms based on these observations and showed that they perform well over the traces under study.

Another approach for improving the quality of multimedia traffic is to exploit the path diversity by sending multiple streams over different paths and by making appropriate choices at the end-systems. We touched upon this topic, by using representative traces to confirm the improvement from using multipath with (i) optimized scheduling of packet transmissions at the sender and (ii) playout scheduling and concealment at the receiver. In our study, we considered paths between two measurements facilities; it would also be interesting to consider many paths in tandem or access networks attached to these backbone paths.

An area of research that could help to monitor, understand and improve the quality of multimedia traffic transmitted over the Internet, is the development of objective quality measures. The mapping of measurable network parameters to perceived voice and video quality would also help for network planning. There is already progress in this area for voice quality but there is more to be done for video, along the lines of [1, 33, 90] and of section 4.4. Video, in general, is a much larger problem space than voice, due to the temporal and spatial dependencies in a compressed video stream and to the numerous error resilience mechanisms available in the standards.

Our observations are based on a sufficiently representative set of Internet backbones today. Looking forward, the network and traffic characteristics are subject to changes. The Internet will hopefully evolve from a period of rapid growth to a more mature infrastructure, where the loss and delay behavior will be completely understood and characterized, the network operation itself will not introduce impairments, and changes in network size and traffic volume/type will be dealt with, in a more systematic way.

Measurements, network and traffic characterization and modeling is a wide research area that is evolving along with the network and traffic themselves as well as with the applications perspective. The characterization done in this thesis is limited to a formal description of the -otherwise empirically observed- delay and loss patterns. It still sheds light to the nature and the magnitude of impairments that the Internet backbones introduce today.

Work at the interface between the network and the applications (taking into account both, instead of considering one of them as a black box) is effective and efficient today. As the Internet improves to become a mature network infrastructure, there will be less room for such optimizations. In particular, there should be little or no need for QoS or error resilience mechanisms, over high-bandwidth environments like backbone networks. Such mechanisms will still be useful in networks toward the edge and definitely in wireless environments, where the resources are inherently limited.

Bibliography

[1] W. Ashmawi, R. Guerin, S. Wolf, M. Pinson, "On the impact of policing and rate guarantees in DiffServ Networks: A Video Steraming Application Perspective", *in Proc. ACM SIGCOMM 2001,* San Diego, CA, USA, August 2001.

[2] J. Apostolopoulos, "Reliable Video Communication over lossy packet networks using multiple state encoding and path diversity", *in Proc. Visual Communication and Image Processing*, San Jose, CA, USA, January 2001.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z.Wang, W.Weiss, "An Architecture for Differentiation Services", *IETF Request for Comments, RFC 247*5, December 1998.

[4] J.-C. Bolot, "Characterizing end-to-end packet Delay and Loss in the Internet", in *Proc. ACM SIGCOMM 1993,* pp.289-298, San Francisco, CA, USA, September 1993.

[5] J.-C. Bolot, M. Turletti, "Experience with rate control mechanisms for packet video in the Internet", *Computer Communications Review*, January 1998.

[6] C. Borman, L.Cline, G.Deisher, T.Gardos, C.Mariocco, D.Newell, J.Ott, C.Zhu, "RTP payload format for the 1998 version of the ITU-T Recommendation H.263 Video (H.263+)", *IETF Request for comments, RFC 2429*, May 1998.

[7] C. Boutremans, G. Iannacone, C.Diot, "Impact of link failures on VoIP performance", *in Proc. ACM NOSSDAV 2002*, Miami Beach, FL, USA, May 2002.

[8] J. Boyce, R. Gaglianello, “Packet Loss Effects on MPEG Video Sent Over the Public Internet”, *in Proc. ACM MULTIMEDIA 1998*, Bristol, UK, September 1998.

[9] P. Brandy, “A technique for investigating on/off patterns of speech”, *Bell Labs Technical Journal, 44(1): 1-22*, January 1965.

[10] J. Chakareski, P. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," *in Proc. IEEE Workshop on Multimedia Signal Processing*, St.Thomas, US Virgin Islands, December 2002.

[11] J. Chakareski, B. Girod, “Rate-Distortion Optimized Streaming over Multiple Varying Channels”, *to be submitted, 2002.*

[12] D. Cheriton, “WRAP: Wide Area Relaying Protocol”, *http://dsg.stanford.eDU/triad/*

[13] P. Chou, Z. Miao, “Rate-Distotion Optimized Streaming of Packetized Media”, *submitted to IEEE Transactions on Multimedia*, February 2001, submitted.

[14] A. Clark, “Modeling the effects of burst packet loss and recency on subjective voice quality”, *Proc. of IP Telephony Workshop*, New York, NY, USA, March 2001.

[15] A. Clark, R. Liu, “Comparison of TS 101 329-5 Annex E with PAMS and PSQM”, *Temp.Doc. 061 for TIPHON#23*, July 2001.

[16] A. Clark, R.Liu, “Comparison of TS101 329-5 Annex E with Emodel”, *Temp.Doc. 062 for TIPHON#23*, July 2001.

[17] D. Clark, J. Wroclawski, K. Sollins, R. Braden, “Tussle in Cyberspace: Defining Tomorrow’s Internet”, *in Proc. ACM SIGCOMM 2002,* Pittsburgh, PA, USA, August 2002.

[18] D. Cohen, “Issues in transnet packetized voice communications”, *in Proc.* 5^{th} *ACM Data Communications Symposium*, Snowbird, Utah, USA, September 1977.

[19] R. G. Cole, J. Rosenbluth, "Voice over IP performance monitoring", *Computer Communications Review, Vol. 31, Issue 2, pp. 9-24*, April 2001.

[20] R. Cox, "Three new speech coders from the ITU cover a range of applications", *IEEE Communications Magazine*, *pp.74-81*, September 1997.

[21] R. Cox, M. Perkins, "Results of a subjective listening test for G.711 with frame erasure concealment", *AT&T contribution to Committee T1 (Telecommunications) to T1A1.7/99-016*, Boulder, CO, USA, May 4-6, 1999.

[22] P. DeLeon, C. Sreenan, "An adaptive predictor for media playout buffering", in *Proc. of IEEE ICASSP*, *pp. 3097-3100,* Phoenix, Arizona, USA, March 1999.

[23] *ETSI, TIPHON project, Technical Report 101 329-6* "Actual measurements of network and terminal characteristics and performance parameters in TIPHON networks and their influence on voice quality", July 2000.

[24] C. Fraleigh, "Provisioning Internet Backbone Networks to Support Latency Sensitive Applications", *Ph.D. Dissertation*, Stanford University, June 2002.

[25] France Telecom R&D, "Study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences: influence of the recency effect", *ITU Study Group 12*, *contribution D.139*, May 2000.

[26] France Telecom R&D, "Continuous assessment of time-varying subjective vocal quality and its relationship with overall subjective quality", *ITU ST 12*, *Contr. COM 12-94-E*, July 1999.

[27] B. Girod, J. Chakareski, M. Kalman, Y. Liang, E. Setton, R. Zhang "Advances in Network-Adaptive Video Streaming", *in Proc. IWDC 2002*, Capri, Italy, September 2002.

[28] D.J. Goodman, G.B., Lockhart, O.J. Wasem, W.-C. Wong, "Waveform substitution techniques for recovering missing speech segments in packet voice communications", *IEEE Transactions on on Acoustics, Speech and Signal Processing, Vol. ASSP-34, no.6, pp. 1440-1448*, December 1986.

[29] J. Gruber, L. Strawczynski, "Subjective effects of variable delay and speech clipping in dynamically managed voice systems", *IEEE Trans. on Communications*, *Vol. 33, No. 8*, August 1985.

[30] K. Gummadi, S. Saroiu, S. Gribble, "King: Estimating Latency between arbitrary Internet End Hosts", *in Proc. Internet Measurements Workshop 2002*, Marseille, France, October 2002.

[31] S. Han, A. Markopoulou, "Transmitting Scalable Video over a DiffServ Network", *Project Report for EE368C class on "Advanced Topics on Image, Video and Multimedia Communications",* Stanford University, March 2001.

[32] Image Systems Enginering Program, Stanford University, Video Sequences available at *http://ise.stanford.edu.*

[33] Institute for Telecommunication Sciences (ITS), Boulder, Colorado, USA, Video Quality Research, *http://ww.its.bldrdoc.gov/n3/video/*

[34] *ITU-T*, H.263+ public implementation TMN-8, 1997

[35] *ITU-T Recommendation G.107*, "The Emodel, a computational model for use in transmission planning", December 1998.

[36] *ITU-T Recommendation G.108*, "Application of the Emodel: a planning guide", September 1998.

[37] *ITU-T Recommendation G.109*, "Definition of categories of speech transmission quality", September 1999.

[38] *ITU-T Recommendation G.113*, “Transmission impairments due to speech processing”, February 2001.

[39] *ITU-T Recommendation G.114*, “One way transmission time”, May 2000.

[40] *ITU-T Recommendation G.711*, “Pulse Code Modulation of Voice Frequencies”, November 1988.

[41] *ITU-T Recommendation G.723.1*, “Speech Coders: Dual Rate Speech Coder for Multimedia Communications Transmissting at 5.3 and 6.3 Kbps”, March 1996.

[42] *ITU-T Recommendation G.729*, “Coding of speech at 8 kbps using conjugate-structure algebraic-code-excited linear-prediction”, 1996.

[43] *ITU-T Recommendation H.263 v.2*, “ Video Coding for Low Bit Rate Communication“, January 1998.

[44] *ITU-T Recommendation P.59*, “Telephone transmission quality objective measuring apparatus: Artificial Conversational Speech”, March 1993.

[45] *ITU-T Recommendation P.800*, “Methods for subjective determination of transmission quality”, August 1996.

[46] *ITU-T Recommendation P.861,* “Objective quality measurement of telephone-band (30-3400Hz) speech encoders”, February 1998.

[47] *ITU-T Recommendation P.862*, “Perceptual evaluation of speech qaulty (PESQ), an objective method for end-to-end speech qaulity assessment of narrow-band telephone networks and speech codecs”, February 2001.

[48] N. Jayant, S. Cristensen, “Effects of Packet Loss in Waveform Coded Speech and Improvements due to the Odd-Even Sample-Interpolation Procedure”, *IEEE Transactions on Communications*, *Vol. 29, pp. 101-109*, February 1981.

[49] W. Jiang, H.Schulzrinne, "QoS measurement of real time multimedia services in the Internet", *Technical Report CUCS-015-99*, *Columbia University*, December 1999.

[50] W. Jiang, H.Schulzrinne, "Analysis if On/Off patterns in VoIP and their effect in traffic aggregation", *in Proc. ICCCN 2000,* Las Vegas, Nevada, USA, October 2000.

[51] The Joint Video Team (JVT) of the ITU-T Q.6/16 Video Coding Experts (VCEG) and the ISO/IEC JTc1/SC29/WG11 Moving Picture Experts Group (MPEG), "H.26L Working Draft 2 (WD-2)", available at *http://standards.pictel.com/ftp/video-site/h26l/JVT-B118r1.doc*, March 2002.

[52] The Joint Video Team (JVT), *H.26L Test Model Long Term Number 9, (TMN-9, JV1.7)*, March 2002, online available at *ftp://standards.pictel.com/video-site/h26L/*

[53] M. Kalman, E. Steinbach, B. Girod, "Adaptive playout for real time media streaming", *in Proc. of ICIP 2002,* Rochester, NY, USA, September 2002.

[54] M. Kalman, E. Steinbach, B. Girod, "Adaptive media playout for low delay video streaming over error-prone channels", *Technical Report,* Stanford University, October 2001.

[55] M. Karam , F. Tobagi, "Analysis of the Delay and Jitter of Voice Traffic over the Internet", *in Proc. of IEEE INFOCOM 2001,* Anchorage, Alaska, USA, April 2001.

[56] A. Katsaggelos, N. Galatsanos, "Signal Recovery Technique for Image and Video Compression and Transmission", *Chapter 7* in P. Salama, N. Shroff, E. Delp, "Error Concealment in Encoded Video Streams, *Kluwer Academic Publishers*, 1998.

[57] J. Kimura, F. Tobagi, J.-M. Pulido, P. Emstad, "Perceived Quality and Bandwidth Characterization of Layered MPEG-2 Video Encoding", *in Proc. of the SPIE 1999*, Boston, Massachussets, USA, September 1999.

[58] N. Kitawaki, K. Itoh, “Pure delay effects on speech quality in telecommunications”, *IEEE Journal on Selected Areas in Communications*, *Vol . 9, No. 4, pp. 586-593*, May 1991.

[59] T. Kostas, M. Borella, I. Sidhu, G. Schuster, J. Grabiec, “Real-time voice over packet-switched networks”, *IEEE Network 12, 1 (January/February 1998), pp.18-27.*

[60] C. Labovitz, A. Ahuja, “Experimental Study of Internet Stability and Wide-Area Backbone Failures”, *in Proc. Fault-Tolerant Symposium, FTCS99,* Madison, WI, June 1999.

[61] W.-M. Lam, A. Reibman, B. Liu, “Recovery of lost or erroneously received motion vectors”, in *Proc. ICASSP* 1993, Minneapolis, USA, April 1993.

[62] A. Leon-Garcia, “Probability and Random Processes for Electrical Engineering”, *2nd Edition, Addison-Wesley 1994*, Chapter 3.11 “Computer Methods for Generating Random Variables”.

[63] Y. J. Liang, E. G. Steinbach, and B. Girod, "Real-time Voice Communication over the Internet Using Packet Path Diversity", *in Proc. ACM MULTIMEDIA 2001*, Bristol, UK, October 2001.

[64] Y. Liang, N.Färber, B.Girod, “Adaptive Playout Scheduling and Loss Concealment for Voice Communications over the networks”, *IEEE Transactions on Multimedia*, April 2001.

[65] D. Loguinov, H. Radha, “Performance of Low Bitrate Internet Video Streaming”, *in Proc. IEEE INFOCOM 2002,* New York, NY, USA, June 2002.

[66] A. Markopoulou, F. Tobagi, M. Karam, “Assessment of VoIP quality over Internet Backbones”, *in Proc. IEEE INFOCOM 2002,* New York, NY, USA, June 2002.

[67] W. Montgomery, "Techniques for Packet Voice Synchronization" , *IEEE JSAC, Vol. SAC-1, no. 6*, *pp. 1022-1028*, December 1983.

[68] S. Moon, "Measurement and analysis of end-to-end delay and loss in the Internet", *Ph.D. Dissertation, UMASS,* February 2000.

[69] S. Moon, J. Kurose, D. Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms", *ACM/Springer Multimedia Systems*, *Vol. 6, pp.17-28*, January 1998.

[70] A. Mukherjee, "On the Dynamics and Significance of Low-Frequency Components in Internet Load", *Internetworking: Research and Experience, Vol. 5, pp.163-205*, October 1994.

[71] W. Noureddine, "Improving the Performance of TCP Applications Using Network-Assisted Mechanisms", *Ph.D. Dissertation, Stanford*, June 2002.

[72] K. Papagiannaki, S.Moon, C.Fraleigh, P.Thiran, F.Tobagi, C.Diot, "Analysis of Measured Single-Hop delay from Operational Backbone Network", *in Proc. IEEE INFOCOM 2002,* New York, NY, USA, June 2002.

[73] C. Perkins, O. Hodson, V. Hardman, "A survey of packet loss recovery techniques for streaming audio", *IEEE Network 12, 5 (Sept./Oct. 1998), pp. 40-48.*

[74] M. Perkins, K.Evans, D.Pascal, L.Thorpe, "Characterizing the subjective performance of the ITU-T 8 Kbps Speech Coding Algorithm - ITU-T G.729", *IEEE Communications Magazine*, *pp.74-81,* September 1997.

[75] M. Pinson, S. Wolf, "Video Quality Measurement User's Manual", ITS/NTIA *Technical Report 02-392*, Boulder, Colorado, USA, March 2002.

[76] J. Pinto, K.Christensen, “An algorithm for playout of packet voice based on adaptive adjustment of talkspurt and silence periods”, *in Proc. IEEE LCN 1999*, Lowell, Massachuserrs, USA, October 1999.

[77] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks", *in Proc. of IEEE INFOCOM 1994, pp. 680-688*, Toronto, Ontario, Canada, June 1994.

[78] J. Rosenberg, L.Qiu, H.Schulzrinne, “Integrating Packet FEC into ADaptive Voice Playout Buffer Algorithms on the Internet”, *in Proc. IEEE INFOCOM 2000,* Tel Aviv, Israel, March 2000.

[79] RouteScience Technologies Inc., White Papers on Route Control, *http://www.routescience.com/cgi-bin/whitepaper.cgi*

[80] D. Sanghi, A. Agrawala, B. Jain, “Experimental Assessment of end-to-end behavior on Internet”, *in Proc. IEEE INFOCOM 1993,* San Francisco, CA, USA, March 1993.

[81] D. Sanghi, O. Gudmundsson, A. Agrawala, “Study of network dynamics”, in *Proc. of 4th Joint European Network Conference 1993*, Trondheim, Norway, May 1993.

[82] H. Sanneck, “A Packet Loss Recovery and Control for Voice Transmission over the Internet”, *Ph.D. Dissertation*, Berlin, 2000.

[83] H. Sanneck, L. Le, A. Wolisz, “Intra-flow loss recovery and control for VoIP”, *Proc. ACM MULTIMEDIA 2001,* Ottawa, Ontario, Canada, October 2001.

[84] S. Savage, A. Collins, E. Hoffmann, J. Snell, T. Anderson, “The end-to-end effects of Internet path selection”, *Computer Communications Review, ACM SIGCOMM '99, 29(4): 289-299*, October 1999.

[85] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A transport protocol for Real Time Applications”, *IETF Request for comments, RFC 1889*, January 1996.

[86] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", *IETF Request for comments, RFC 2215*, September 1997.

[87] N. Spring, R. Mahajan, D. Wetherall, "Measuring ISP Topologies with RocketFuel", *in Proc. ACM SIGCOMM 2002*, Pittsburgh, PA, USA, August 2002

[88] C. J. Sreenan, J.-C. Chen, P.Agrawal, B.Naderdran, "Delay reduction techniques for playout buffering", *IEEE Transactions on Multimedia*, *Vol. 2, no. 2 pp. 88-100*, June 2000.

[89] K. Sriram, W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data", *IEEE JSAC, Vol. SAC-4, (6): pp. 833-846*, September 1986.

[90] K. Stuhlmüller, N. Färber, B. Girod, "Analysis of Video Transmission over Lossy Channels", *IEEE JSAC*, *Vol. 18, No. 6, pp. 1012-1032*, June 2000.

[91] F. Tobagi, I. Dalgic, "Performance Evaluation of 10Base-Tand 100Base-T Ethernets Carrying Multimedia Traffic", in *IEEE JSAC*, *Vol. 14, No. 7, pp. 1436-1454*, September 1996.

[92] F. Tobagi, A. Markopoulou, M.Karam, "Is the Internet ready for VoIP?", *in Proc. IWDC'02*, Capri, Italy, September 2002.

[93] M. Yajnik, S .Moon, J. Kurose, D. Towsley, "Packet Loss Correlation in the MBONE Multicast Network", *UMASS Technical Report, 1995.*

[94] S. Youngberg, "Rate/pitch modification of speech using constant Q transform", *in Proc. of ICASSP* 1979, Washington, DC, *April 1979, pp. 748-751.*

[95] M. Yuang, S. Liang, Y. Chen, "Dynamic Video Playout Smoothing Method for Multimedia Applications", *Multimedia Tools and Applications*, *Vol. 6, pp. 47-59*, 1998.

[96] University College London (UCL), Department of Computer Science, Robust Audio Tool (RAT), *http://www-mice.cs.ucl.ac.uk/multimedia/software/rat*

[97] V. Vleeschauwer, J. Janssen, G. Petit, F. Poppe, Alcatel Technical Report "Quality bounds for packetized voice transport", *Alcatel Technical Report*, 1st Quarter 2000.

[98] S. Voran, "Speech quality of G.723.1 coding with added temporal discontinuity impairments", *Proc. of ICASSP 2001,* Salk Lake City, Utah, USA, May 2001.

[99] J. Walker, J. Hicks, "Evaluating data networks for VoIP", *NetIQ white paper available online at www.netiq.com, 2001.*

[100] Y. Wang, J. Ostermann, Y.-Q. Zhang, "Video Processing and Communications", *Prentice Hall* 2002.

[101] Y. Wang Q.-F. Zhu, "Error Control and Concealment for Video Communication: A Review", *in Proc. of IEEE, Special Issue on Multimedia Signal Processing*, pp. 974-997, May 1998.

[102] S. Wenger, "H26L over IP: The IP Network Adaptation Layer" , *submitted for publication*, March 2002.

[103] S. Wenger, G. Côté, "Using RFC 2429 and H.263+ at the low to medium bit rates for low latency applications", *in Proc. of Packet Video Workshop 1999*, New York, NY, USA, April 1999.

[104] S. Winkler, "A perceptual distortion metric for digital color video", *in Proc. of SPIE Human Vision and Electronic Imaging, Vol. 3644, pp.175-184,* January 1999.

[105] Y. Zhang, N. Duffield, V. Paxson, S. Shenker, "On the Constancy of Internet Path Properties", *Proc. of ACM SIGCOMM 2001,* San Diego, CA, USA, August 2001.

[106] C. Zhu, "RTP payload format for H.263 video streams, *IETF Request for Comments, RFC 2190*, September 1997.