Packet Loss Recovery and Control for Voice Transmission over the Internet

Vom Fachbereich 12 (Elektrotechnik)
der Technischen Universität Berlin
zur Verleihung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von
Dipl.-Ing. Henning Sanneck

Berlin 2000

D 83

Tag der Einreichung:
7. Juli 2000

Tag der wissenschaftlichen Aussprache:
10. Oktober 2000

Promotionsauschuss:
Vorsitzender: Prof. Dr.-Ing. E. Obermeier
1. Berichter: Prof. Dr.-Ing. A. Wolisz
2. Berichter: Prof. Dr. Dr. h.c. R. Popescu-Zeletin (FB 13)

Abstract

"Best effort" packet-switched networks, like the Internet, do not offer a reliable transmission of packets to applications with real-time constraints such as voice. Thus, the loss of packets impairs the application-level utility. For voice this utility impairment is twofold: on one hand, even short bursts of lost packets may decrease significantly the ability of the receiver to conceal the packet loss and the speech signal play-out is interrupted. On the other hand, some packets may be particular sensitive to loss as they carry more important information in terms of user perception than other packets.

We first develop an end-to-end model based on loss run-lengths with which we can describe the loss distribution within a flow. The packet-level metrics derived from the model are then linked to user-level objective speech quality metrics. Using this framework, we find that for low-compressing sample-based codecs (PCM) with loss concealment isolated packet losses can be concealed well, whereas burst losses have a higher negative perceptual impact. For high-compressing frame-based codecs (G.729) on one hand the impact of loss is amplified through error propagation caused by the decoder filter memories, though on the other hand such coding schemes help to perform loss concealment by extrapolation of decoder state. Contrary to sample-based codecs we show that the concealment performance may "break" at transitions within the speech signal however.

We then propose mechanisms which differentiate between packets within a voice data flow to minimize the impact of packet loss. We designate these methods as "intra-flow" loss recovery and control. At the end-to-end level, identification of packets sensitive to loss (sender) as well as loss concealment (receiver) takes place. Hop-by-hop support schemes then allow trading the loss of one packet, which is considered more important, against another one of the same flow which is of lower importance. As both packets require the same cost in terms of network transmission, a gain in perceived quality is obtainable. We show that significant speech quality improvements can be achieved while still maintaining a network service which is virtually identical to best effort in the long term.

Keywords: Voice over IP, Internet Telephony, Packet Loss, Loss Recovery, Objective Speech Quality Measurement, Queue Management, Differentiated Services

Zusammenfassung

Paket-vermittelnde Netzwerke wie das Internet, die nach dem "best effort"-Prinzip arbeiten bieten keine Möglichkeit die Übertragung von Paketen für Echtzeitdienste wie Sprache zu garantieren. Somit wird durch Paketverluste die Dienstqualität beeinträchtigt. Bei Sprachübertragung treten dabei die folgenden Effekte auf: einerseits können schon kurze Folgen von verlorenen Paketen (Bündelverluste) die Fähigkeit des Empfängers beeinträchtigen die Paketverluste zu verschleiern. Dadurch wird das Sprachsignal als unterbrochen wahrgenommen. Andererseits können einzelne Pakete des Datenstromes besonders anfällig gegenüber einem Verlust sein, da sie Informationen beeinhalten die entscheidend für die wahrgenommene Sprachsignalqualität am Empfänger sind.

Zunächst wird ein Modell entwickelt welches auf der Anzahl der hintereinander verloren gegangenen Pakete basiert. Mit diesem Modell ist es möglich die Verlustverteilung innerhalb eines Datenstromes zu beschreiben. Die von dem Modell abgeleiteten Paketverlustmetriken werden dann mit Methoden der objektiven Sprachqualitätsmessung verbunden. Innerhalb dieses Rahmenwerkes stellen wir fest das schwach-komprimierende Sprachkodierer ("sample-based codecs", PCM) mit Fehlerverschleierung einzelne Paketverluste gut überbrücken können. Bündelverluste haben dagegen einen starken negativen Einfluss auf die Sprachqualität. Bei hochkomprimierenden Kodierern ("frame-based codecs", G.729) ist es einerseits so, das die Auswirkungen von Paketverlusten durch das Gedächtnis der Dekoder-Filter noch verstärkt werden. Andererseits machen es solche Kodiermethoden einfacher eine Fehlerverschleierung durchzuführen, da die Statusinformationen innerhalb des Dekoders extrapoliert werden können. Im Gegensatz zu den schwach-komprimierenden Sprachkodierern ist jedoch festzustellen, das die Qualität der Fehlerverschleierung an Sprachbereichsübergängen zusammenbrechen kann.

Dann werden Mechanismen vorgestellt die zwischen Paketen innerhalb eines Sprachdatenstroms (flow) unterscheiden können, um die Auswirkungen von Paketverlusten zu minimieren. Wir bezeichnen diese Methoden als "intra-flow" Paketverlustbehandlung und -kontrolle. In den Endsystemen (end-to-end) findet dabei die Identifizierung von verlustsensitiven Paketen (am Sender) sowie eine Fehlerverschleierung (am Empfänger) statt. Unterstützungsmechanismen in den Netzwerkknoten (hop-by-hop) erlauben es dann Verluste von als wichtiger identifizierten Paketen auf Kosten von weniger wichtigen Paketen desselben Datenstroms zu vermeiden. Da für beide Paketarten diesselben Netzwerkressourcen aufgewendet werden müssten, ist somit ein Gewinn an Sprachqualität möglich. Es wird gezeigt das dieser Gewinn bedeutend ist, wobei jedoch der Netzwerkdienst, über längere Zeitabschnitte gesehen, praktisch mit einem "best effort"-Dienst gleichgesetzt werden kann.

Stichwörter: Voice over IP, Internettelefonie, Paketverluste, Paketverlustbehandlung, Objektive Sprachqualitätsmessung, Queue Management, Differentiated Services

Acknowledgments

It is with great appreciation that I acknowledge my advisor, Prof. Dr.-Ing. Adam Wolisz, for his encouragement and valuable advice. While giving me a great degree of freedom to choose a topic and pursue my research, it was his insight and guidance which finally made this thesis a reality. Beside his academic excellence, I am also grateful for his caring personality and unique sense of humour.

I would also like to thank Prof. Dr. Dr. h.c. Radu Popescu-Zeletin for taking the time to review the thesis and to give valuable feedback. I am also indebted to him for creating the excellent research environment at GMD Fokus, which made the thesis work so much easier.

Many thanks go to Ass. Prof. Mikhail Smirnov, PhD, for the opportunity to combine my thesis work with my tasks as a researcher in the Global Networking (GloNe) group at GMD Fokus. He also encouraged my thesis work and gave me the possibility to travel and present my work to fellow researchers.

I am very thankful for the time I had the opportunity to spend with my colleague and roommate Dr.-Ing. Georg Carle. Numerous valuable discussions, inspirations and the work on joint research papers have significantly improved the quality of the thesis.

Many thanks are also due to Dr.-Ing. Dorgham Sisalem who did his PhD work at GMD Fokus during the same time period. His excellent research work has been a permanent incentive for me. It has been a pleasure to work with him.

Special thanks go to Dipl.-Ing. Nguyen Tuong Long Le, Dipl.-Ing. Michael Zander, Dipl.-Ing. Andreas Köpsel and Dipl.-Ing. (FH) Davinder Pal Singh who did their Diploma or student thesis work under my supervision and contributed in numerous ways directly to the success of this thesis.

Furthermore, I would like to thank all members of the GloNe and TIP groups at GMD Fokus. Their support in the daily work and the numerous discussions also beyond the scope of my thesis have been a pleasure. Thanks are due also to the system administration group at FOKUS (VST) for their regular file backup schedule, which has once saved me from a nightmare.

I also would like to acknowledge some of the people external to GMD Fokus who helped doing this thesis with valuable discussions and insights, by providing papers, software and computer accounts for Internet measurements: Dr. Yang and Prof. Yantorno, Temple University, Dr. Koodli, Nokia Research Center, Prof. Kleijn, KTH, Mr. Voran, ITS.T, Mr. Jiang, Columbia University and Prof. Noll, TU Berlin.

Deep gratitude goes to my family, particularly to my parents Helga Sanneck and Dr.-Ing. Hugo Sanneck. Starting from early childhood they have been able to convince me of the values of education and always encouraged my pursuit of knowledge.

Finally I would like to express my gratitude to my fiancée Dipl.-Ing. Birgit Königsheim for her continuous love, understanding and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the last three decades we witnessed two fundamental technical evolutions on which this thesis is based. The first consists of using a digital representation for a speech signal ([JN84]), being the basis for efficient further processing, storage and transmission. The second is the development and deployment of packet-switched networks, which started with small experimental islands and grew to the global interconnection we know now as the Internet ([LCC+98]). Both technologies began to converge as early as the 1970s and '80s with research and experiments on *packet voice* ([Coh80], Fig. 1.1).

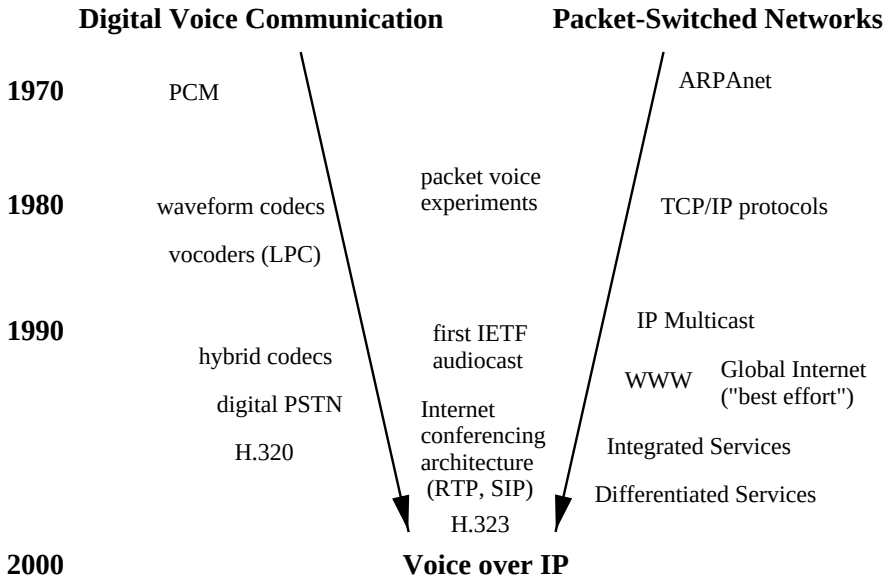


Figure 1.1: Voice over IP history

The 1990s brought both an unprecedented Internet growth, with the WWW as the first (non real-time) "mass" application, as well as the large-scale deployment of digital transmission technology in the conventional (circuit-switched) telephone networks. However, the convergence of packet-switching and voice took place at a

much slower pace. It was not until 1992 that the first large-scale research experiments with voice transmission over the Internet took place ([Cas92]). Only recently, we have seen the development of an Internet conferencing architecture ([HCB96]) including basic protocol support for real-time applications (RTP, [SCFJ96]), support for new communication paradigms like multicast ([SM96]) and call-setup signaling ([HSSR99, Uni96f]). Finally, today we are facing an increasing demand for rapid deployment of real-time services like Internet Telephony ([SR98, SSSK99]), triggered by current economical reasons[1] and the statistical multiplexing gain inherent to a packet-switched network. The ultimate motivation for convergence is, however, the creation of a *single*, integrated communication infrastructure offering manageable advanced services to satisfy the user demand of accessing all services from a unified platform (computer-telephony integration, [Sch97]).

However, when designing such an Integrated Services packet-switched network we are confronted with its fundamental tradeoff: statistical multiplexing at the expense of the reliability of packet delivery which can result in a degradation of the quality of the provided service. Available bandwidth is exploited very efficiently by the multiplexing of packets, yet excessive delays and packet losses cannot be avoided due to unlimited concurrent access of the network by different senders and subsequent congestion at interior network nodes. Packet loss can be compared in its importance to the problem of channel distortion in analogue communications.

The Internet Engineering Task Force (IETF, [Soc]) has worked on these Quality-of-Service (QoS) issues ([CSS+98]) for several years now. Other organizations and standards bodies have just begun to work on QoS issues specific for packet voice ([Con97, Uni99, Ins98]). Still, the Internet today offers only a single "best effort" service, i.e. all traffic is handled in the same way (typically using a single FIFO Drop-Tail queue per interface in a router). Thus the network has no idea about the properties of the traffic it handles, which largely differ between flows[2] of different applications (data, voice, video). Loss recovery is done on an end-to-end basis by higher-layer protocols (TCP) with retransmissions, such that applications can receive a lossless service just using end-to-end means. Obviously, this approach can lead to excessive delays under network congestion. For classical Internet applications like ftp and e-mail, this has been tolerable. However it is not feasible for real-time, interactive services like voice- and video-conferencing.

1.1 Motivation and Scope

Most real-time applications exhibit tolerance against occasional loss of packets. However, considering that a real-time flow experiences a certain constant amount of packet loss, the impact of loss may vary significantly dependent on *which* packets

[1] Except the access fees no costs dependent on the communication path (local, wide area) are (yet) to be paid in the Internet.

[2] Here we define a "flow" informally as a sequence of packets with an application-defined association and finite duration typically in the range as known by human interaction. A formal flow definition for the Internet will be given in section 2.2.3.

are lost within a flow. In the following we distinguish several reasons for such a variable loss sensitivity. For our explanation we consider the packet level as well as the ADU (Application Data Unit) level, where an ADU is the unit of data relevant for the application such as a voice or video frame:

1. *Temporal sensitivity:* Loss of ADUs which is correlated in time may lead to disruptions in the service. For video, a "flickering" or "freezing" image is the result. Note that this effect is further aggravated by some interdependence between ADUs (i.e. that one ADU can only be decoded when a previous ADU before has successfully been received and decoded). For voice, as a single packet contains typically several ADUs (voice frames) this effect is more significant than for video. It translates basically to isolated packet losses versus losses that occur in bursts.

2. *Sensitivity wrt. ADU integrity:* ADU integrity addresses the relationship between the ADU and the packet level (i.e. the specific way in which the ADUs are packetized). For video that means that a loss of 50% of all transmitted frames is annoying, but tolerable, however that the loss of 50% of the packets of every frame (resulting in the same amount of loss) might render the video undecodable and thus completely useless (note that this in turn affects the time sensitivity). For this example we have assumed that every frame is of equal importance and consists of the same number of packets. For speech, ADU integrity is not an issue, due to the fact that for current coding schemes the ADUs (voice frames) are typically smaller than a packet and thus are not split for transmission.

3. *Sensitivity due to ADU heterogeneity:* Certain ADUs might contain parts of the encoded signal which are more important with regard to user perception than others of the same flow. Let us consider a video flow with two frame types of largely different perceptual importance (we assume same size, frequency and no inter-dependence between the frames). Even when under the loss of 50% of the packets, all packets belonging to a certain frame are received (see point 2. above), the perceptual quality varies hugely between the case where the 50% of the frames with high perceptual importance are received and the case where the 50% less important frames are received. For voice traffic this translates to the case where the coding scheme is not able to distribute the information equally between consecutive frames and thus generates packets of variable perceptual importance.

Network support for real-time multimedia flows can on one hand aim at offering a lossless service, which however, to be implemented within a packet-switched network, will be costly for the network provider and thus for the user. On the other hand, within a lossy service, the above sensitivity constraints must be taken into account. It is our strong belief that this needs to be done in a generic way, i.e. no application-specific knowledge (about particular coding schemes e.g.) should be necessary within the network and, vice versa, no knowledge about network specifics should be necessary within an application.

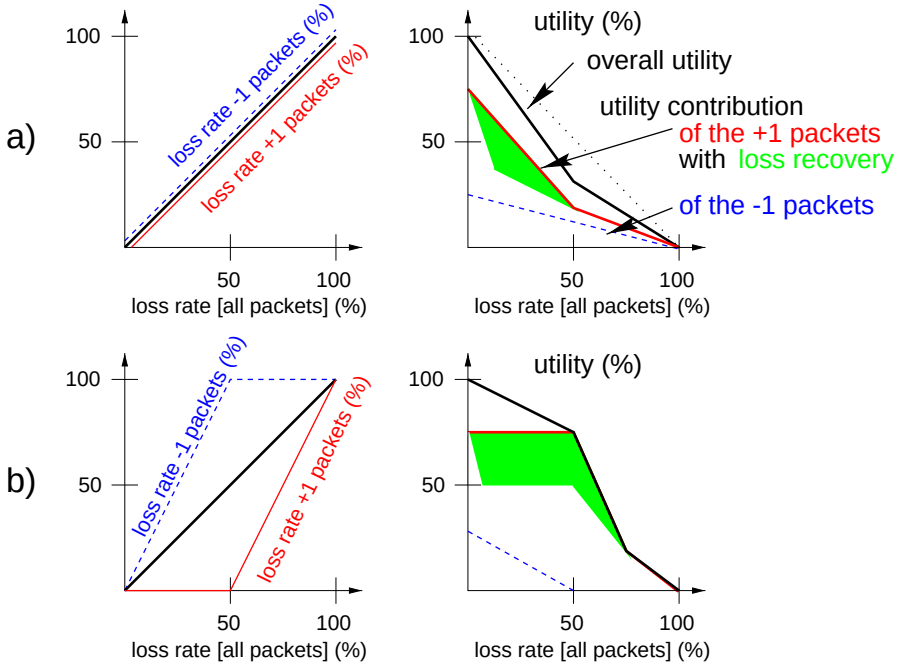


Figure 1.2: Schematic utility functions dependent on the loss of more (+1) and less (-1) important packets: a) "best effort" case b) "best effort" with intra-flow loss control

Let us now consider the case that 50% of packets of a flow are identified as more important (designated by "+1") or less important ("-1") *due to any of the above sensitivity constraints*. Figure 1.2 a) shows a generic *utility* function describing the application-level Quality of Service (QoS) dependent on the percentage of packets lost. For real-time multimedia traffic, such utility should correspond to perceived video/voice quality. If the relative importance of the packets is not known by the transmission system, the loss rates for the +1 and -1 packets are equal. Due to the over-proportional sensitivity of the +1 packets to loss as well as the dependence of the end-to-end loss recovery performance on the +1 packets, the utility function is decreasing significantly in a non-linear way (approximated in the figure by piece-wise linear functions) with an increasing loss rate. Figure 1.2 b) presents the case where all +1 packets are protected at the expense of -1 packets. The decay of the utility function (for loss rates $< 50\%$) is reduced, because the +1 packets are protected and the end-to-end loss recovery can thus operate properly over a wider range of loss rates indicated by the shaded area. This results in a *graceful degradation* of the application's utility. Note that the higher the non-linearity of the utility contribution of the +1 packets is (deviation from the dotted curve in Fig. 1.2 a)), the higher is the potential gain in utility when the protection for +1 packets is enabled. Results for actual perceived quality as utility for multimedia applications exhibit such a non-linear behavior.

QoS	*intra-flow*	*inter-flow*
network nodes (hop-by-hop)	filtering (network adaptation) *packet differentiation*	packet differentiation reservation
end systems (end-to-end)	*selective*/adaptive loss recovery and avoidance	non-adaptive loss recovery

Table 1.1: Taxonomy of QoS enhancement approaches

To describe this effect and provide a taxonomy for different QoS enhancement approaches, we introduce a novel terminology: we designate mechanisms which influence QoS parameters between flows (thus decrease the loss rate of one flow at the expense of other flows) as **inter-**flow QoS. Schemes which, in the presence of loss, differentiate between packets within a flow as demonstrated in Figure 1.2 above, provide **intra-**flow QoS enhancement. As additional mechanisms have to be implemented within the network (hop-by-hop) and/or in the end systems (end-to-end), we have another axis of classification[3].

Now we can group existing QoS enhancement concepts into this framework (Table 1.1). As opposed to the previous paragraphs, we are now only considering the transmission of interactive voice traffic.

End-to-end sender adaptation to the current network load ([SS98a], *end-to-end intra-flow QoS*), i.e. reducing the bit-rate in response to network congestion, is difficult to apply to voice. This difficulty arises when considering the necessary per-flow overhead (fast feedback) together with the usual voice traffic properties (low bit-rate), i.e. the per-flow gain in congestion control through adaptation is small. Often, adaptation is not possible at all due to the fixed output bit-rate of the voice encoder. Thus, Internet voice applications must be augmented by loss recovery mechanisms, which are somewhat isolated from the speech encoding process, to cope with packet loss. This is the case because most standardized codecs were optimized for high compression assuming a transmission "channel" with low error rates (like those available in a circuit-switched network). Due to the given

[3]To describe the location of schemes used for *traffic control*, we use the term "hop-by-hop" as opposed to "network layer". The term "network layer" is conformant to the OSI model layer 3, the layer where routing and forwarding within an internetwork takes place. Generic traffic control mechanisms (i.e. those which are independent of a specific link layer technology) are however typically implemented per link layer interface "below" layer 3 (see Fig. 2.6 in section 2.2.3).

Also, in IP-based networks the definition of the "application layer" location is not in accordance with OSI, but rather driven by current implementation environments, where an application accesses network services via a socket interface (an application can contain some transport layer processing functions, e.g. an application incorporating real-time transport protocol (RTP, [SCFJ96]) processing using an UDP socket). For these reasons we use only "end-to-end" to designate mechanisms operating at the level where the communication relation (sender-receiver) is visible. It should however also be noted that the the terms "end-to-end" and "end system" do not necessarily imply "application-to-application" as other nodes (proxies) can transfer data on behalf of the applications and run "end-to-end" protocols and algorithms.

delay constraints these are open-loop schemes like Forward Error Correction (FEC). When such loss recovery schemes are non-adaptive to network congestion, the flow uses more bandwidth which is then not available for other flows. Therefore such approaches can be classified as *end-to-end inter-flow QoS*. Note that the presented categories for loss sensitivity also apply to applications which are enhanced by end-to-end loss recovery mechanisms. End-to-end mechanisms can reduce and shift such sensitivities but cannot eliminate them.

The QoS can also be improved by exploiting knowledge about a flow within the network which then leads to a graceful degradation under congestion (*hop-by-hop intra-flow QoS*). Typically this is accomplished by filtering application-layer information, which is however both expensive in terms of resources, as well as undesirable with regard to network security. Additionally, specifically for Internet voice, most of these mechanisms are unsuitable, again considering the voice flow properties (high compression, uniform frame structure).

Mechanisms of service differentiation between flows (*hop-by-hop inter-flow QoS*) have been explored extensively (e.g. within the Internet Integrated Services model [BCS94]). However, actual deployment has been delayed, mainly due to complexity reasons (e.g. it is needed to keep per-flow state in every router along the path during the lifetime of a flow). Particularly for voice over IP this leads to high resource consumption (and therefore to high costs) due to the need for conservative characterization of flow requirements, and overhead due to needed signaling and state maintenance. Also, the explicit setup at every hop could take relatively long (in comparison to a session/call initiation). Furthermore, providing inter-flow QoS leads to an immediate need for a complete charging and accounting architecture. Even if such QoS enforcement mechanisms are ubiquitous, it will be necessary to provide alternatives. This can be due to economical reasons, but also e.g. due to user mobility[4].

Thus, considering that hop-by-hop QoS enforcement will not be deployed everywhere, it is important that efficient end-to-end loss recovery schemes are developed which can be complemented (and not replaced) by hop-by-hop support mechanisms. However, it can be stated that only few known approaches consider the presence of loss recovery/control mechanisms at the respective other level (end-to-end / hop-by-hop). We argue that only few cooperation/knowledge between the layers can lead to significant performance improvements. Therefore we adopt a combined approach in this work introducing novel intra-flow QoS mechanisms at both levels. Considering a combined approach is particularly interesting for voice, as scalability is a major concern due to the small per-flow bandwidth.

Our work is targeted mainly at interactive voice. On one hand this is due to the strong demand from the users and the consequently high importance of Voice over IP for the success of the Internet as the ubiquitous packet switching infrastructure. On the other hand due to the simplicity of the voice flow structure, voice is a good candidate to explore simple means of separate and combined operation of QoS

[4]A temporary graceful degradation is needed until the hop-by-hop QoS is reestablished to the new user location during a hand-off.

enhancement mechanisms, which can then be extended to more complex flow types.

We believe that QoS setup for IP telephony and Voice over IP in general should not be tied to the call-setup signaling ([HSSR99, Uni96f]). This allows for QoS provision for aggregated flows in the core of the network (aggregation ([RS96, RS98, JH98, SS98c]) will play an important role due to the small per-packet payload of individual flows). Additionally, the QoS setup should not be limited to a telephone call model (point-to-point), but scale to large multicast groups. Furthermore, deployment of IP telephony signaling and QoS provision can be done incrementally in separate steps. Finally user mobility (see [SR98]) can be supported more efficiently as the data/QoS control and the call signaling path may differ.

1.2 Approach

We propose to employ intra-flow QoS enhancement mechanisms at both the end-to-end and the hop-by-hop level. The end-to-end schemes rely on pre-processing of the speech signal at the sender, which facilitates the concealment/reconstruction of lost speech packets at the receiver. The hop-by-hop schemes shape the loss pattern of the voice packet sequence thus yielding a more predictable service at the end-to-end level. When brought together, the end-to-end mechanism can exploit the increased predictability of the network service. Note that the hop-by-hop mechanisms only constitute a support mechanism with regard to the end-to-end level. We aim at evaluating the benefit by adequate but still simple metrics at both the packet as well as the user level.

We classify the voice coding schemes into either sample-based (where a digital sample of an analog signal is directly encoded) or frame-based (where the evolution of the analog signal over a certain time period is encoded into digital codewords which constitute a frame). The proposed approach covers end-to-end mechanisms for both sample- and frame-based codecs working either independently or with a direct interface to the hop-by-hop level. The availability of a rich existing literature is exploited. At the hop-by-hop level however few directly related work exists, so we rather look at different approaches to inter-flow QoS to identify suitable building blocks which can be adapted to fulfill our goals.

End-to-end level

We design a scheme, where more important parts of the speech signal are better protected in the presence of loss on an end-to-end basis. For sample-based codecs this is achieved by choosing the packet boundaries adaptively. If a packet is lost, the receiver can conceal the loss of information by using adjacent signal segments of which (due to the preprocessing/packetization at the sender) a certain similarity to the lost segment can be assumed. For frame-based codecs the information extracted by the pre-processing is used to identify frames which, in the event of a loss, cannot be easily concealed by the speech decoder itself. Then, these frames are either protected with redundancy or mapped to a higher priority at the hop-by-hop level.

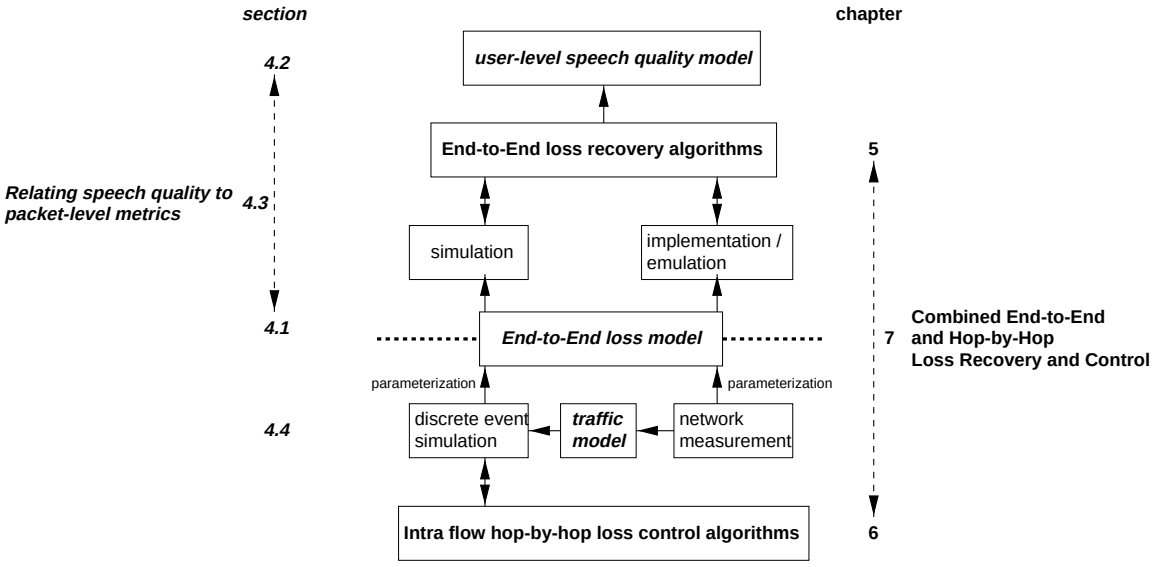


Figure 1.3: Thesis methodology and chapter association

The approach can be grouped into the end-to-end intra-flow QoS category as *selective* (payload adaptive) loss recovery (Table 1.1).

Hop-by-hop level

In this work, active queue management algorithms are developed, which try to give preferred service to certain packets of a flow at the expense of other packets of the same flow. We identify two basic approaches: the first one is based on flow detection, i.e. network nodes have a certain knowledge about flow types and their needs, keep partial per-flow state and influence the packet drop decision using this knowledge. Applications (or proxies acting on behalf an application) then do not need to cooperate explicitly. The second approach relies on packet marking, i.e. the complexity of deriving the relative importance of the packets by the traffic control entities of intermediate routers is pushed to the edges of the network. Routers react dependent on their overall congestion situation. Thus, applications mark the packets of their flow to enable a graceful degradation within the network.

This approach can be described as intra-flow *packet differentiation* (Table 1.1).

1.3 Thesis Outline and Methodology

Figure 1.3 depicts a schematic view on the methodology of this thesis. On one hand we use results of network measurement (own measurements and results available in the literature) to parameterize our own end-to-end loss model according to the measured Internet behavior. On the other hand, based on measurement results, we

construct a traffic model, which allows us to develop loss control algorithms which work at a network node (hop-by-hop) by using discrete event simulation. Then, the same end-to-end loss model can be parameterized according to the simulated modified network behavior.

The parameterized models are subsequently employed for the performance evaluation of end-to-end loss recovery algorithms. Finally we use objective speech quality models to measure the performance at the user level. It should be noted that the end-to-end loss model links separate evaluations at the end-to-end and hop-by-hop level respectively. This separation is done for various reasons: discrete event simulations require a significantly higher effort to yield end-to-end results, therefore they should be confined to developments where the behavior of an individual node needs to be taken into account in detail. Applying numerous loss patterns derived from an end-to-end model to a speech sample is significantly less complex than running the discrete simulations for an equal number of times (using different seeds for the random number generation) while yielding the same statistical relevance. For the end-to-end algorithms we do not use feedback, therefore the simulation of static operating points appears reasonable.

The outline of the thesis is as follows:

Chapter 2 gives a brief introduction to digital voice communication, voice transmission over IP-based networks, the problem of packet loss and basic metrics to describe the packet loss process. We also present a taxonomy of schemes for loss avoidance, recovery and control.

In chapter 3 we then present related work which focuses on end-to-end, hop-by-hop and combined schemes for QoS enhancement: Section 3.1 introduces related work on end-to-end-only QoS enhancement schemes which are applicable to Internet voice. All schemes (except non-adaptive FEC) can be classified as intra-flow QoS enhancement. We analyze receiver-only loss recovery and loss recovery schemes which introduce modifications at both the sender and the receiver. We discuss related work on hop-by-hop mechanisms suitable to give QoS support for interactive voice in section 3.2 covering purely local (intra-flow) as well as distributed approaches which typically improve the inter-flow QoS. Section 3.3 then presents some related work of combined end-to-end/hop-by-hop approaches.

Chapter 4 describes the methodology we employed to evaluate the performance of the QoS enhancement schemes: We identified the need for a thorough analysis of existing packet loss models and metrics and developed a novel model for loss characterization, which is discussed in section 4.1. Section 4.2 then shows how actual application-level QoS (user perception) for voice can be described. We introduce conventional objective and subjective quality metrics as well as novel perceptual metrics for objective speech quality assessment. Section 4.3 describes the relationship between the introduced packet-level and speech quality metrics. To enable the design and performance evaluation of supporting hop-by-hop loss control schemes, section 4.4 presents the traffic model used in the discrete event simulations.

Chapter 5 presents our work on end-to-end loss recovery schemes. Due to the largely differing properties of sample- and frame-based codecs in terms of loss

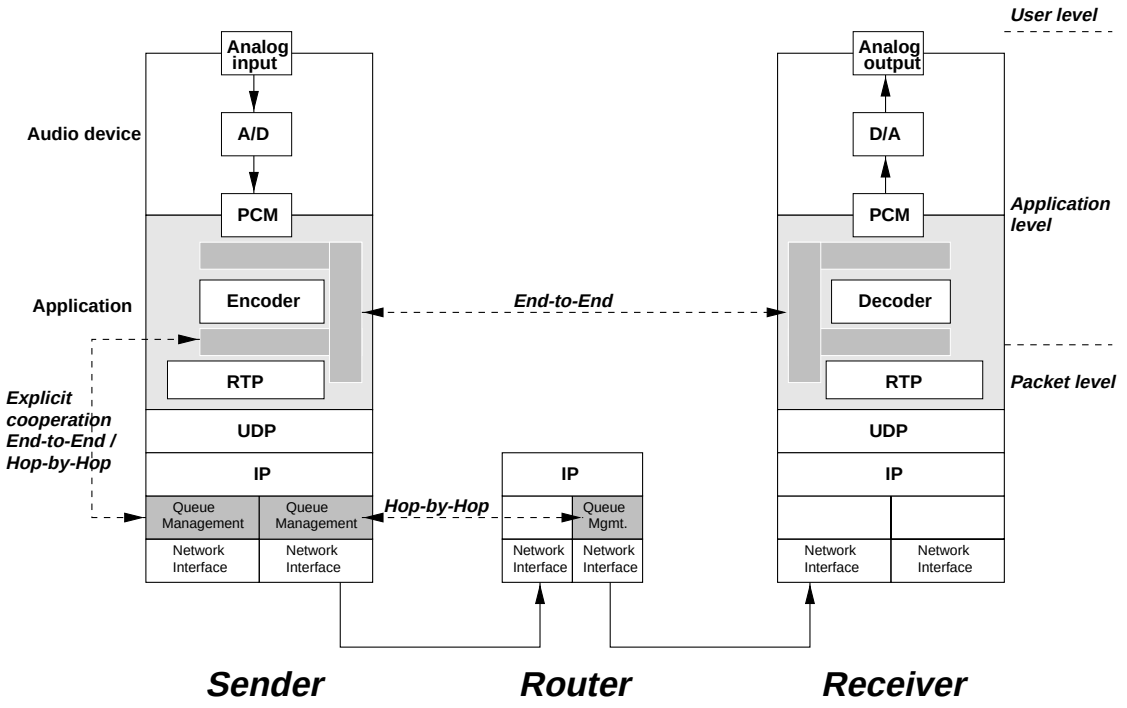


Figure 1.4: Architecture / structure of the thesis

resilience we treat both types of codecs differently: in section 5.1 the development of our proposed scheme for sample-based codecs called Adaptive Packetization/Concealment (AP/C) is explained. AP/C is novel in the sense that it can avoid the basic limitations of receiver-only schemes identified in section 3.1, but at the same time only introduces minor modifications at the sender. The approach is evaluated by objective speech quality measurement and subjective testing as introduced in sections 4.2.1 and 4.2.2 respectively. Additionally, in section 5.2 we present a comparable approach to increase the loss resilience for frame-based codecs (we use the G.729 [Uni96a] voice codec as an example) and apply again methods of objective speech quality assessment for the evaluation.

Chapter 6 then discusses and explores design options for the proposed intra-flow QoS hop-by-hop mechanisms for end-to-end support. We develop two queue management algorithms which fulfill our goals: the first called PLoP (Predictive Loss Pattern) is based on flow detection and selective discarding of queued packets. The second algorithm is Differential RED (DiffRED), a derivative of the well-known RED ([FJ93]) concept of discarding packets adaptively to the congestion state at a network element. We investigate the specific performance of each algorithm by simulation and then compare the two algorithms in section 6.4 using the metrics introduced in section 4.1.

Finally, in chapter 7 we evaluate the performance for combined end-to-end and hop-by-hop schemes. This is done in particular for the explicit mapping of end-to-end knowledge on the hop-by-hop support, as this approach is less separable than

the implicit one.

Figure 1.4 presents an overview over the software architecture for Internet voice transmission. The shaded boxes show the locations in the stack where our proposed mechanisms should be applied. The architectural overview can serve also as a guideline through this thesis, as it reflects the vertical (packet-level versus user-level) and horizontal (hop-by-hop versus end-to-end) nature of the building blocks presented in the individual chapters.

Chapter 2

Basics

In this chapter we will review the basics of digital voice communication which are relevant to our work. Furthermore the necessity of Quality-of-Service enhancement mechanisms is explained and the architecture in which those mechanisms are to be embedded is outlined.

2.1 Digital voice communication

This section presents an overview of production, digitization and coding of speech. We briefly discuss basic coding techniques employed for sample- and frame-based codecs and have a closer look at the G.729 codec as one prominent example for a frame-based codec.

2.1.1 Speech production

In this section, we will briefly discuss some basic properties of speech signals and how they are produced. In particular, we will take a look at the speech properties that are of major importance to our work, especially the characteristics of voiced and unvoiced sounds. See [RS78, Del93] and the references therein for more general and detailed discussions.

Speech signals are non-stationary and at best can be considered as quasi-periodic over a short period of time. Thus, they cannot be exactly predicted. Speech signals can be roughly divided into two categories: voiced and unvoiced sounds. Voiced sounds are produced by pushing air from the lung through the glottis with the shape and the tension of the vocal cords adjusted so that this flow of air causes them to vibrate in a relaxation oscillation. The vibration of the vocal cords results in a sequence of quasi-periodic pulses of air that excites the vocal tract. Thus, voiced sounds can be modeled by exciting a filter modeling the vocal tract with a quasi-periodic signal that reflects the air pulses produced by the vocal cords. The rate of the vibration of the vocal cords' opening and closing are defined as the fundamental frequency of the phonation. It is often used interchangeably with the term *pitch period*. Varying the shape and the tension of the vocal cords can change

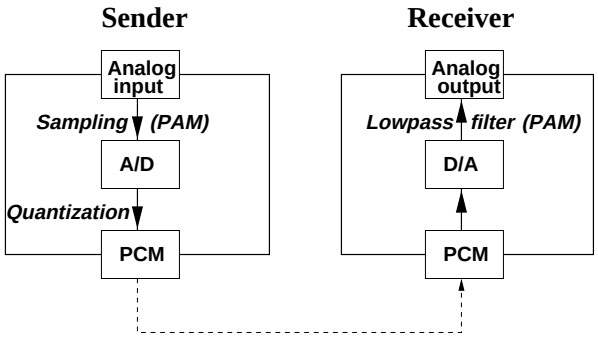


Figure 2.1: Digital voice transmission system using Puls Code Modulation (PCM).

the frequency of the vocal cords' vibration, i.e. the pitch. Another property of voiced speech is that certain frequency ranges are suppressed by resonance within the vocal tract. Thus peaks of the amplitude at the *formant* frequencies appear in the signal's spectra. The properties of voiced sounds can be summarized as follows: they have quasi-periodic characteristics in the time domain. The energy of voiced sounds is generally higher than that of unvoiced sounds. Furthermore, voiced sounds are more important to the perceptual quality than unvoiced sounds.

Unvoiced sounds are generated by forcing a steady flow of air at high velocities through a constriction region in the vocal tract to produce a turbulence. The location of the constriction region determines what unvoiced sound is produced. Unvoiced sounds are similar to random signals and have a broad spectrum in frequency domain. Random signals (white noise) are usually used to model unvoiced sounds.

Within regions of either voice or unvoiced speech regions, smaller units of a size between approximately $40ms$ and $100ms$ can be distinguished. These units are called phonemes. They are the smallest units which convey a linguistic meaning.

2.1.2 Digitization

Figure 2.1 shows the conversion of an analog speech signal to a digital one at the sender, as well as the re-conversion to an analog output at the receiver. At the sender (within a PC or workstation typically within the audio hardware accessible via the audio device of the operating system) the analog signal is first low-pass filtered to avoid aliasing when sampling. Then the *sampling* at a certain sampling frequency takes place, resulting in a PAM (Pulse Amplitude Modulation) signal. A typical sample frequency for voice is $8kHz$; according to the Nyquist theorem this allows to represent frequencies up to $4kHz$ which is sufficient for naturally sounding (telephone quality) speech. This is equivalent to modulating the signal with a pulse train. Then the modulated analog signal (now being a sequence of different amplitude pulses rather than a continuous signal) is converted to a digital representation. This conversion implies *quantization*, i.e. an analog amplitude with infinite resolution within its allowed range is mapped to one value of a discrete set of values (a typical set is e.g. $2^{16} = 65536$ values: 16 bit quantization).

At the receiver the digital representation is decoded back to yield a PAM signal. This signal (which has an infinite number of replicas of the original analog signal's spectrum) is then low-pass filtered with a filter with the same cutoff frequency as at the sender. Thus the original signal, however distorted by the approximation process of the A/D quantization process, is recovered.

2.1.3 Coding / compression

In order to reduce bandwidth consumption in the transmission of speech signals, speech coding is employed to compress the speech signals, i.e. on one hand to use as few bits as possible to represent them and on the other hand to maintain a certain desired level of speech quality. Compression is achieved by exploiting temporal redundancies in the speech signal. Temporal redundancies exist in the correlation between adjacent speech samples (short-term correlation) as well as in the pitch periodicity (long-term correlation). Additionally the different sensitivities of the human hearing system in different frequency bands can be exploited for compression. The actual compression gain is realized by quantization of the relevant samples or coefficients and using predictor filters of limited depth, thus achieving lossy compression of the speech signal with some quality/complexity versus bit-rate tradeoff. In general, speech coding techniques are divided into three categories: waveform codecs, voice codecs (vocoders), and hybrid codecs. In the following we use the term "codec" for the speech encoding/decoding system as a whole and "encoder"/"decoder" for the respective encoding or decoding functionality.

2.1.3.1 Sample-based codecs

Sample-based codecs try to directly encode speech signals in an efficient way by extracting redundancies and exploiting the temporal and/or spectral characteristics of the speech waveform. The simplest waveform codec is Puls Code Modulation (PCM) where the amplitude of the analog signal (section 2.1.2) (or a digital sample with a higher resolution) is (re-)quantized to one of a discrete set of values. PCM is a memoryless (non-adaptive) coding. Therefore the bandwidth needed to transmit a speech signal is high (e.g. 16 bit/sample $\times$ 8000 sample/s = $128kbit/s$). A first step to reduce this bandwidth while maintaining the same output quality is to employ non-uniform quantization (also called "companding"), i.e. the quantization step size varies with the signal value. This improves the quality for two reasons: first, frequently occurring amplitudes can be quantized finer and second, the human hearing exhibits logarithmic sensitivity (i.e. the perception of small amplitudes is more critical and thus they should be quantized finer). Typically companding is employed according to either the μ- and A-law logarithmic curves (for Europe and North America respectively) resulting in a bit-rate of $64kbit/s$ (with 8 bit quantization).

An encoding scheme which exploits the fact that the speech waveform is evolving slowly (i.e. adjacent samples are correlated) is the Differential PCM. In its simplest form the sender encodes the difference between two adjacent samples and the receiver

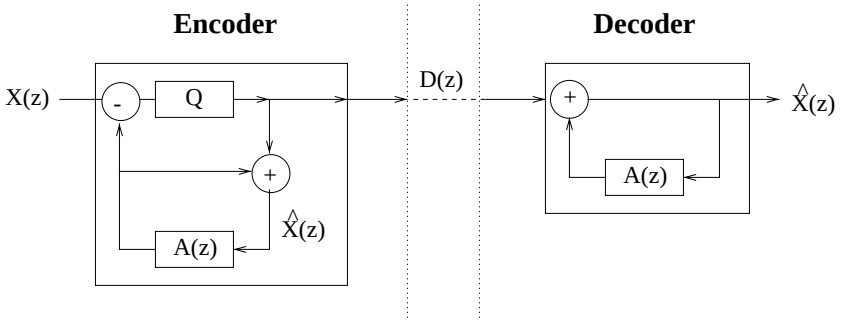


Figure 2.2: Differential Puls Code Modulation (DPCM).

restores the signal by integration. However actual DPCM systems employ a larger predictor filter (with a memory of l samples). The transfer function of the predictor filter in the z-domain can be computed as follows (where a_i are the filter coefficients):

$$A(z) = \sum_{i=1}^{l} a_i z^{-i} \tag{2.1}$$

Figure 2.2 shows the encoder and decoder structure of a DPCM system. At the encoder, the difference between the input speech sample $x(n)$ (represented by its z transform $X(z)$) and its estimate $\hat{x}(n)$ ($\mapsto \hat{X}(z)$) is computed and transmitted to the receiver. There the signal is reconstructed using the same predictor filter loop as in the encoder:

$$\hat{X}(z) = \frac{D(z)}{1 - A(z)}$$

In Adaptive Differential PCM (ADPCM), both the quantizer step size as well as the predictor filter coefficients are varied adaptively to the speech signal content. Typically (using a backward adaptive predictor filter) the predictor filter adaptation is estimated from the received signal. Thus only the quantizer step information has to be transmitted additionally.

2.1.3.2 Frame-based codecs / G.729

Vocoders and hybrid codecs attempt to model speech signals by a set of parameters and then try to efficiently encode these parameters. They usually operate on "frames" which represent a fixed number of speech samples. Hence, they are also called frame-based codecs. Vocoders and hybrid codecs typically operate at a lower bit rate than waveform codecs at the cost of higher complexity.

In frame-based codecs, the vocal tract is modeled by a linear filter (section 2.1.1). That means a speech sample $x(n)$ is estimated by a linear combination of previous speech samples:

$$\hat{x}(n) = \sum_{i=1}^{l} a_i x(n-i)$$

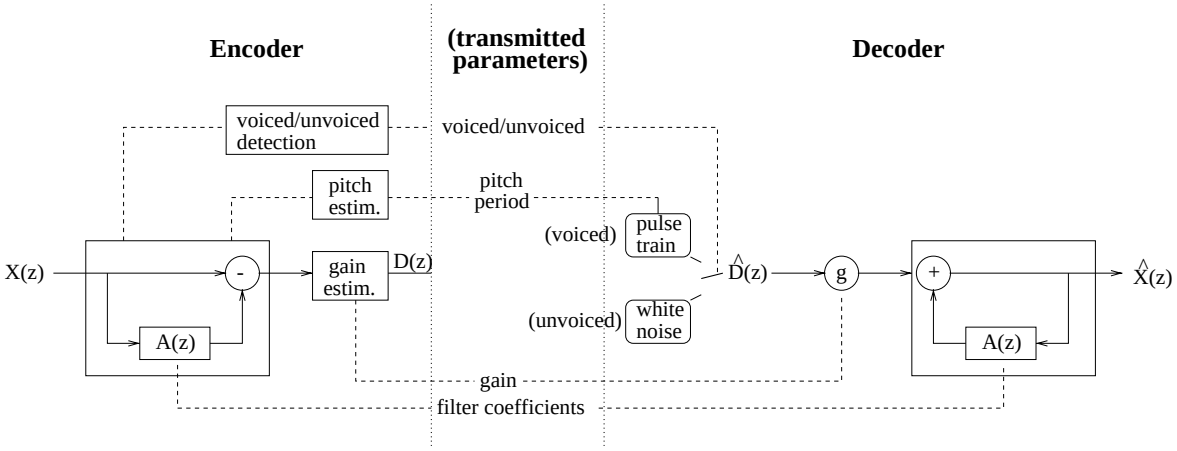


Figure 2.3: Linear Predictive Coding (LPC).

This estimation is referred to as *linear prediction* (LP). It amounts to filtering the signal with a filter (predictor filter) with the transfer function (see Eq. 2.1):

$$A(z) = \sum_{i=1}^{l} a_i z^{-i}$$

There are various approaches for computing the coefficients a_i, e.g. minimization of the mean square of the difference between the original and the estimate $d(n) = x(n) - \hat{x}(n)$. A linear prediction with optimally chosen coefficients yields a decorrelated difference signal $d(n)$ (i.e. the envelope of that difference signal's spectrum is flat). Thus if the speech signal (represented by its z transform $X(z)$) is filtered with the optimal predictor error filter (analysis filter) $1 - A(z)$ we get an output signal with a flat (white) spectrum $D(z) = \frac{1}{g}(1 - A(z))X(z)$ (see Fig. 2.3: encoder; $\frac{1}{g}$ is a scaling factor[1]).

Thus, if the inverse filter $H(z) = \frac{1}{1-A(z)}$ (synthesis filter) is excited with a signal with a white spectrum, the output signal $X(z)$ represents the speech signal for which the predictor coefficients have been optimized (Fig. 2.3: decoder). In vocoders this behavior is approximated by exciting the synthesis filter $H(z)$ with a periodical train of pulses ($d(n) = \delta(n) \mapsto D(z) = 1$) for voiced sounds. The period of the train of pulses is equal to the pitch period. For unvoiced sounds $d(n)$ is a random signal, thus the power density of spectrum $D(z)$ is constant (white noise).

The minimization of the mean square of $d(n)$ for computing the LP coefficients leads to the following linear equations ([RS78, Clu98]):

$$\Phi(i,0) \quad = \quad \sum_{k=1}^{l} a_k \Phi(i,k) \quad i \in [1,l]$$

[1]The scaling factor g is computed from the variance of the difference signal $d(n)$.

Codec	G.723.1 hybrid [Uni96c]	G.729 hybrid [Uni96a]	G.727 waveform [Uni90]
Coding scheme	Algebraic Code Excited Linear Prediction (ACELP) or Multipulse Maximum Likelihood Quantization (MP-MLQ)	Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP)	Adaptive Differential Pulse Code Modulation (ADPCM)
Bit rate (kbit/s)	5.3 or 6.3	8	40, 32, 24, or 16
Complexity (DSP MIPS)	14-20	20	≈ 2

Table 2.1: Properties of common speech codecs

$$\Phi(i,k) = \sum_n x(n-i)x(n-k) \tag{2.2}$$

The summation range of Eq.2.2 is determined by the interval over which a speech signal can be assumed to be stationary and constraints like the desired algorithmic delay of the coder. One way to simplify the computation of this equation is to assume all samples outside of an analysis segment of length N to be zero, resulting in using the autocorrelation function (ACF) of the speech signal r_{xx} ([Clu98]):

$$\Phi(i,k) = r_{xx}(|i-k|) = \sum_{n=0}^{N-1-|i-k|} x(n)x(n+|i-k|) \tag{2.3}$$

The equations (written in matrix form) can then be solved using the Levinson-Durbin recursion ([RS78, Del93]).

Vocoders operate at a bit rate of around 2.4 kbit/s or lower and produce speech that is intelligible but not natural (section 2.1.4). Hence, they are mainly used in military applications where natural sounding is not very important and bandwidth is very scarce.

In hybrid codecs, the excitation signal for the linear filter is chosen in such a way that the perceived distortion is as small as possible. Hybrid codecs deliver a better speech quality than vocoders at the cost of a higher bit rate, because information about the excitation is transmitted as side information. They represent a compromise of different interdependent attributes: bit rate, complexity, and buffer delay. These attributes are traded off against each other, e.g. a very low bit rate could result in high complexity and large buffer delays which are both undesirable. Furthermore, hybrid codecs used for speech transmission over the Internet should also be robust against loss of frames.

Table 2.1 provides an overview over some features of common waveform and hybrid codecs (extracted from [MM98, Spa94]). Particularly the two frame-based

codecs G.723.1 ([Uni96c]) and G.729 ([Uni96a]) are very attractive for speech transmissions over the Internet because they provide toll (telephone) quality speech at much lower bit rates (5.3/6.3 kbit/s and 8 kbit/s respectively) than conventional PCM (64 kbit/s). Thus the network resource requirements for a large scale deployment can be reduced significantly. Their high complexity is not of great concern because speech encoding and decoding can now be performed with inexpensive hardware in the end systems at user premises.

The G.729 speech codec The G.729 codec employs the Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP) coding scheme. It operates at 8 kbit/s. Input data for the coder are 16-bit linear PCM data sampled at 8 kHz. A G.729 speech *frame*[2] is 10 ms in duration, corresponding to 80 PCM speech samples. For each frame, the encoder analyzes the input data and extracts the parameters of the Code Excited Linear Prediction (CELP) model such as linear prediction filter coefficients and excitation vectors. The approach for determining the filter coefficients and the excitation is called analysis by synthesis: The encoder searches through its parameter space, carries out the decode operation in each loop of the search, and compares the output signal of the decode operation (the synthesized signal) with the original speech signal. The parameters that produce the closest match are chosen, encoded, and then transmitted to the receivers. At the receivers, these parameters are used to reconstruct the original speech signal. The reconstructed speech signals are then filtered through a post-processing filter that reduces the perceived noise by emphasizing the spectral peaks (formants, section 2.1.1) and attenuating the spectral valleys ([MM98]).

G.729 encoder and decoder operation For each 10-ms frame, the encoder performs a linear predictive analysis to compute the linear prediction filter coefficients. For the sake of stability and efficiency, the linear-prediction filter coefficients are not directly quantized but are transformed into line spectral pairs (LSP[3]) and quantized using a predictive two-stage vector quantization process . The excitation for the speech signal is computed per 5-ms subframe (corresponding to 40 PCM speech samples) and has two components: fixed and adaptive-codebook. First, an open loop pitch delay is estimated once per 10-ms frame. This estimation is based on the autocorrelation of the weighted speech signal that is derived from filtering the speech signal through a perceptual weighting filter. The adaptive-codebook contribution models the long-term correlation of speech signals and is expressed in a closed-loop pitch delay and a gain. The closed-loop pitch delay is searched for around the open loop pitch delay by minimizing the error between the perceptually weighted input signal and the previous excitation filtered by a weighted linear-prediction synthe-

[2]We use the term *frame* for the unit of the encoding/decoding operation (ADU) and *packet* for the unit of transmission. One packet carries typically several frames.

[3]LSPs are an alternative representation of the LP coefficients with better quantization and interpolation properties ([Del93], chapter 5.4).

sis filter. The difference of the found excitation filtered by the synthesis filter and the original signal is then used to find the fixed-codebook contribution. The fixed-codebook vector and the fixed-codebook gain are searched by minimizing the mean-squared error between the weighted input signal and the weighted reconstructed speech signal, using a pulse train as excitation. The adaptive-codebook gain and the fixed-codebook gain are then jointly vector quantized using a two stage vector quantization process.

The G.729 decoder extracts the following parameters from the arriving bit stream: the line spectral pair coefficients, the two pitch delays, two codewords representing the fixed-codebook vector and the adaptive- and fixed- codebook gains. The line spectral pair coefficients are interpolated and transformed back to the linear prediction filter coefficients for each subframe. Then, for each subframe the following operations are performed:

- The excitation is the sum of the adaptive- and fixed-codebook vectors multiplied by their respective gains.
- The speech signal is obtained by passing the excitation through the linear prediction synthesis filter.
- The reconstructed speech signal is filtered through a post-processing filter that incorporates an adaptive postfilter based on the long-term and short-term synthesis filters, followed by a high-pass filter and scaling operation. These operations reduce the perceived distortion and enhance the speech quality of the synthesized speech signals.

The internal frame loss concealment algorithm of the G.729 will be introduced in section 3.1.3.3.

2.1.4 Speech quality / intelligibility

Speech intelligibility deals with the content (the linguistic meaning): *what* a speaker has said. Speech quality on the other hand can be seen as a superset, i.e. it comprises intelligibility as well as additional perceptions like "naturalness" and speaker recognizability: *how* a speaker has said something. Good quality implies good intelligibility (the converse does not necessarily need to be true). For example, very low-bit-rate vocoders (section 2.1.3.2) produce speech that is intelligible but not natural. The inter-relationship of intelligibility and quality is not well understood currently ([Del93], chapter 9.1.2). This is due in part to the difficulty of isolating properties within the speech signal and associating these properties with either quality or intelligibility.

In this thesis we focus on speech quality as our ultimate metric for the following reasons: Our main target application is telephone-quality speech using low- ($5kbit/s$) to medium-bit-rate ($64kbit/s$) codecs, where assuring intelligibility alone is not sufficient to achieve user satisfaction. In addition to the distortion introduced by the

coding scheme, we consider distortion caused by packet losses. As introduced earlier (section 1) our scope here is to develop end-to-end as well as hop-by-hop mechanisms for loss recovery and control in a well-engineered network which is highly loaded, but has no inter-flow QoS support. If the number of losses introduced is so high that intelligibility is severely impaired, we believe that the scope of research somewhat shifts to proper network/ traffic engineering and load balancing/ QoS routing.

Section 4.2 presents methods to determine speech *quality* using objective and subjective methods. In the next section we first review the impact of voice transmission over packet-switched networks on quality.

2.2 Voice transmission over packet-switched networks

When transmitting interactive voice traffic over a packet-switched network we are confronted with its fundamental tradeoff: we have an efficient bandwidth usage versus a decreased reliability of a packet transmission resulting in a potentially degraded quality of service. This means that packets can be delayed within the network. As packets belonging to a flow might experience different delays due to different states of queues they pass (or even different paths they follow), there can be a substantial variation in the delay called jitter. Finally, in a "best effort" packet-switched network like the Internet there is no guarantee that a packet is delivered at all. Therefore in this section we first discuss the quality impairments on voice traffic over packet-switched networks. Then we present the structure of software tools used for the transmission of voice in such an environment. Finally we discuss the architecture currently proposed for the Internet which allows for the transmission of real-time flows including voice by using specific protocols and loss avoidance, recovery and control mechanisms.

2.2.1 Quality impairments

The component of the *delay* a packet experiences can be described as follows:

- propagation delay (physical layer, Fig. 2.6): the time the physical signals need to travel across the links along the path taken by the data packets. Propagation delay represents a physical limit given by the speed of light that cannot be reduced.

- forwarding delay (network layer): the time the router takes to forward a packet: extraction of the destination address from the packet header, routing lookup and switching the packet over the router's backplane from the input to the output port. Forwarding delay also includes the time needed to send the packet completely out of the output port (and thus is dependent on the outgoing link's speed).

- queuing delay (layer 2/3): the time a packet has to spend in the queues at the input and output ports before it can be processed. Additional queuing time may be caused by specifics of the link layer, e.g. an Ethernet collision or the segmentation/reassembly process between cells and packets in ATM (Asynchronous Transfer Mode).
- packetization/de-packetization delays (all layers): the timed needed to build data packets at the sender (await the arrival of a sufficient amount of data from the application or the upper protocol layer, compute and add headers at the respective layers), as well as to strip off packet headers at the receiver. Packetization and de-packetization time can be kept small by using efficient protocol implementations (avoidance of actual copy operations, proper alignment of header fields, etc.)
- algorithmic delay and lookahead delay (application layer): the time it takes to digitize speech signals and perform voice encoding at the sender. Typically encoding works on a sequence of PCM sample (frames) so that first enough samples have to arrive. Some codecs also need to buffer data in excess of the frame size (look-ahead).
- decoding delay (application layer): the time needed to perform decoding and conversion of digital data into analog signals at the receivers.

There are various recommendations on the maximum end-to-end delay above which a conversation cannot any more be considered interactive. This bound is however highly dependent on human perception. A "mouth-to-ear" (one-way) delay of $< 150ms$ is considered to be just acceptable (see e.g. [Ins98]).

Jitter, also known as the variability of delay (not necessarily being the delay variance), is caused mainly by the queuing delay component. When several packets in a router compete for the same outgoing link, only one of them can be processed and forwarded while the others have to be queued. The result of packet queuing is that packets sent by the sender at equidistant time intervals arrive at the receiver at non-equidistant time intervals. It should be noted that all delay components introduced above (except the propagation delay) may exhibit some variations when the networking software is executed within a non real-time operating system.

At the application layer, the impact of jitter can be reduced by keeping the received packets in a play-out buffer and adding an extra amount of delay before they are played. This extra amount of delay is an engineering trade-off: it must be small enough to have no impact on the interactivity of voice applications and it must be large enough to smooth out the jitter and to enable most of the delayed packets to arrive before their play-out time (packets which arrive after their play-out time have to be considered as lost). Play-out buffer algorithms have been investigated e.g. in [Sch92, RKTS94].

While delay and jitter are important parameters with a direct relation to perceived Quality-of-Service the most fundamental quality impairment with regard to voice traffic is packet loss.

2.2.1.1 Packet loss / loss correlation

Packet loss often occurs in the Internet when a router becomes congested, i.e. it receives more packets to forward than it can process. Large loss bursts (outages) also occur when network pathologies exist, i.e. a router or a link fails. However, this problem is orthogonal to congestion and belongs into the (QoS) routing domain (the routing must re-converge to paths around the point of failure). Another reason for loss can be transmission errors (bit errors) of the underlying medium. Typically the bit error ratio is extremely low however for fixed networks (but it can be significant for wireless networks). For this thesis we only consider losses that are intrinsic to the functional blocks at the end-to-end and hop-by-hop level and thus we use the term packet loss for losses caused by congestion only.

Packet loss in the Internet is a frequent and also the most serious problem that speech transmissions over the Internet have to face. Applications can use message sequence number of transport protocols such as RTP (section 2.2.3.1) to detect a packet loss. In order to provide an acceptable quality, loss recovery / control must be performed.

While coding schemes can exploit the redundancy within the speech efficiently for compression, together with packet loss compression can lead to even more significant degradations of the output speech quality as for PCM speech. When using a backward adaptive waveform coding scheme like ADPCM (section 2.1.3.1), the decoding of the next arriving packets after the loss can lead to significant distortion due to the potentially large changes in the signal amplitude[4]. Vocoders and hybrid coders (section 2.1.3.2) use even more adaptivity for compression, however as the decoder state is not directly coupled to the amplitude as in ADPCM the distortions are less dramatic, however they might persist longer (until the decoder has re-synchronized with the encoder). To summarize we can say that redundancies within a speech signal can be exploited both for compression and loss resilience. The higher the compression of the signal is, the lower is the intrinsic loss resilience. Due to this fact we treat end-to-end loss recovery for sample- and frame-based codecs separately in chapters 5.1 and 5.2.

Obviously the time interval in which the decoder does not receive data from the network is a crucial parameter with regard to user perception (if either a loss is perceived not at all, as a glitch or as a dropout). The time interval at the user level translates to the burstiness of loss (or loss correlation) at the packet level. Several researchers (e.g. [BLHHM95, BVG97, CKS93, MS96, RR95]) have highlighted the importance of loss burstiness as a QoS parameter. Subjective tests using sample-based codecs ([GS85], [MM98]) have shown that it is generally preferable with regard to the resulting speech quality to have a higher number of small length gaps ($\lessapprox 20ms$) rather than the infrequent occurrence of long gaps (which leads to the loss of entire logical speech elements in the signal). Generally, gaps in the signal and the discontinuities at the edges of these gaps have a high impact on the resulting speech

[4]Note that due to this property in current Internet audio tool implementations ([Col98]), an ADPCM packet contains one conventional PCM speech sample. Thus the ADPCM algorithm works rather on a per-packet than on a per-conversation basis.

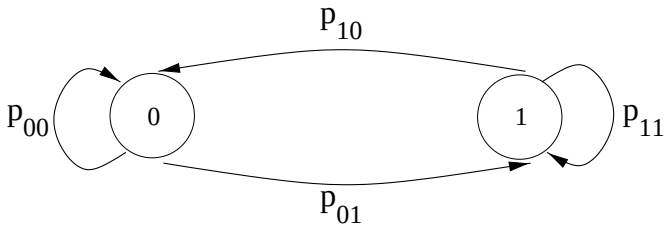


Figure 2.4: Gilbert model

quality.

Basic packet loss metrics For a simple characterization of the behavior of the network as seen by one flow, we use the well-known Gilbert model (Fig. 2.4). The system can be completely described by the probability p_{01} for a transition from state 0 (no loss) to state 1 (loss) and the probability p_{11} to remain in state 1. The probability p_{11} represents the *conditional loss probability clp*. The probability of being in state 1 p_1, representing the mean loss, is called *unconditional loss probability ulp*.

$$p_1 = p_0 p_{01} + p_1 p_{11} \tag{2.4}$$

$$p_0 + p_1 = 1 \tag{2.5}$$

Thus the unconditional loss probability can be computed as follows:

$$ulp = \frac{p_{01}}{1 - p_{11} + p_{01}} \tag{2.6}$$

The Gilbert model implies a geometric distribution of the probability for the number of consecutive packet losses k, $(1-clp)clp^{k-1}$. If losses of one flow are correlated (i.e. the loss probability of an arriving packet is influenced by the contribution to the state of the queue by a previous packet of the same flow and/or both the previous and the current packet see bursty arrivals of other traffic, [SKT92]) we have $p_{01} \leq clp$ and thus $ulp \leq clp$. For $p_{01} = clp$ the Gilbert model is equivalent to a 1-state (Bernoulli) model with $ulp = clp$ (no loss correlation).

The Gilbert model is known to approximate relatively well the head of the loss distribution of actual Internet voice traffic traces. The tail of the distribution is typically dominated by few events, caused e.g. by link outages and route flappings. The Gilbert model thus only provides a limited insight with regard to the correlation of losses (loss burstiness). Several researchers provided additional intra-flow loss models and metrics ([Par92, MFO98, OMF98, KR97, CT97, ZF96, KR00, NKT94, LNT96, KK98, LNT96]). However most of these metrics are neither inter-related nor well motivated.

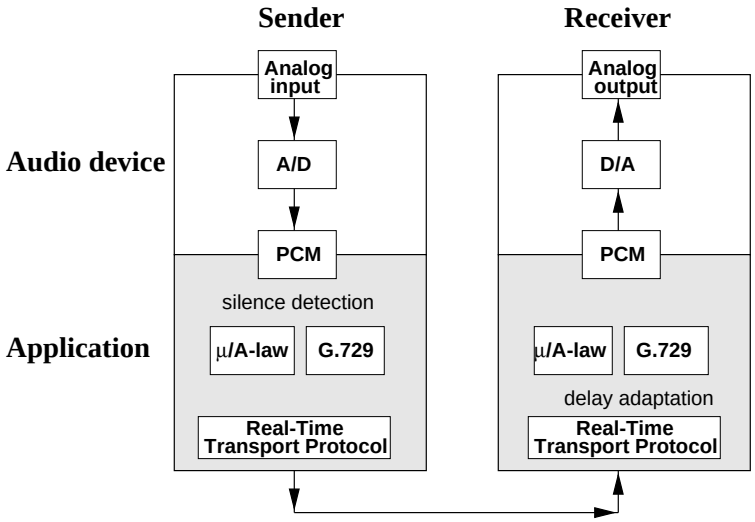


Figure 2.5: Generic structure of an audio tool.

2.2.2 Sender / receiver structure

Figure 2.5 shows the main building blocks of a device/software tool for the transmission of voice over a packet-switched network. In addition to the components introduced earlier (Figure 2.1) we have the following components:

- "silence detection" or "voice activity detection" (VAD): VAD is a method already used when multiplexing voice calls over a circuit-switched network to save bandwidth and exploit the gain of statistical multiplexing: It has been shown that in a typical conversation the activity of a speaker is approximately below 40%. Thus the available bandwidth during silent periods can be used for other calls (in the circuit-switched case) or generally other traffic. The term "talk-spurt" is often used to define a sequence of packets which each have an energy higher than a certain energy threshold. A segment of voice data is defined as a silent segment if its energy is lower than this threshold. Silent segments can thus be suppressed in order to save bandwidth. However, a number of "hangover" silent packets immediately preceding or following a talk-spurt should be transmitted to avoid that perceptually important but low-energy speech material is not transmitted ("clipping", [Sch92, San95, MM98]). The lengths of the talk-spurts vary dependent on the speaker and on the speech material (Dempsey et al. report in [DLW96] that using a length of $400ms$ is in accordance with the measurements they conducted). See also [JS00a] for more details on silence detection.

- encoder/decoder (section 2.1.3)

- delay adaptation (section 2.2.1)

- real-time transport protocol: in addition to conventional transport protocol functions, real-time services need specific protocol support for re-sequencing

of packets/loss detection and play-out point determination/delay adaptation. We will discuss the real-time transport protocol for the Internet in section 2.2.3.1.

2.2.3 The Internet conferencing architecture

Figure 2.6 shows the architecture of the Internet protocols which are relevant to conferencing (i.e. protocols and entities which are necessary to run real-time multimedia application). Protocols providing basic transport (RTP, [SCFJ96]), call-setup signalling (H.323 [Uni96f], SIP [HSSR99]), QoS signalling (RSVP [BZB+97]) and QoS feedback (RTCP, [SCFJ96]) are shown. Additionally to the protocols relevant to QoS, the *enforcement* of the QoS on the data flows which pass through a router (or are emitted from a host) is necessary. For QoS-passive media, i.e. when the link layer does not implement QoS control mechanisms, the enforcement is realized by a *traffic control* entity which is typically located between IP and the network device driver on an outgoing interface. For complex link layers like ATM these mechanisms need either be mapped on or replaced by the respective link layer means (see section 3.2.2.1).

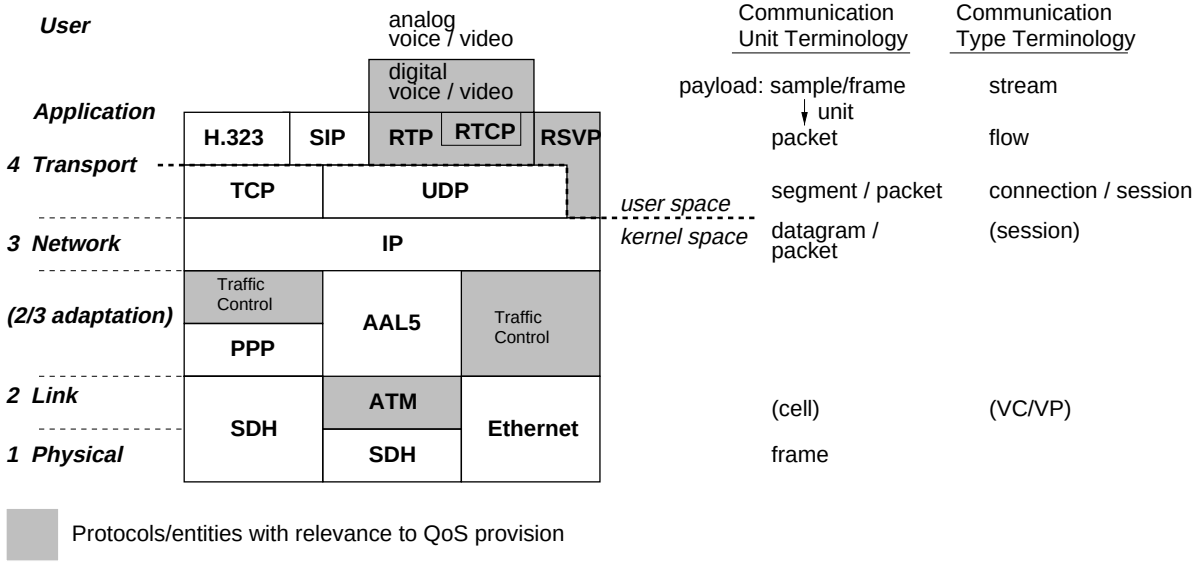


Figure 2.6: The Internet conferencing architecture

Figure 2.6 also gives the terminology for the communication units at the respective layers. We use the generic term "unit" between the application-layer "sample/frame" and the transport layer "packets" because the "samples/frames" may be associated to larger units (section 5.1.1), interleaved (section 3.1.2.1) or combined with additional data (section 3.1.2.2) before packetization. Additionally, we define a terminology for the type of communication, i.e. terms for the conceptual association

of the communication units: "a stream of voice frames", "a flow of packets". Particularly important here is the notion of a flow: For IPv4 a flow can be identified by the tuple *(source address, destination address, protocol ID, source port, destination port)*. A flow contains an application-layer data stream.

2.2.3.1 The Real-Time Transport Protocol (RTP)

Currently, most interactive audio and video applications use the real-time transport protocol (RTP, [SCFJ96]) for data transmission with real-time constraints. RTP itself does not provide Quality of Service (QoS) guarantees or timely delivery of data but relies on lower-layer services to do so. RTP runs on top of existing transport protocols, typically UDP, and provides real-time applications with end-to-end delivery services such as payload type identification and delivery monitoring. RTP provides transport of data with a notion of time to enable the receivers to reconstruct the timing information of the sender. Besides, RTP messages contain a message sequence number to allow applications to detect packet loss, packet duplication, or packet reordering.

RTP is extended by the RTP control protocol (RTCP) that exchanges member information in an on-going session. RTCP monitors the data delivery and provides the users with some statistical functionality. The receivers can use RTCP as a feedback mechanism to notify the sender about the quality of an on-going session.

An RTP message contains an RTP header followed by the RTP payload (e.g., audio data or video data). An RTP message of the current version (version 2) is shown in Figure 2.7. Below is a short explanation for some fields of the RTP message shown in Figure 2.7. More details can be found in [SCFJ96].

- Payload type (PT): 7 bits
  The payload type specifies the format of the RTP payload following the fixed header.

- Sequence number: 16 bits
  The sequence number counts the number of the RTP packets sent by the sender and is incremented by one for each transmitted packet. The sequence number allows the receivers to detect packet loss, packet duplication, out-of-order packet delivery, and to restore the packet sequence.

- Timestamp: 32 bits
  The timestamp reflects the sampling instant of the first data sample contained in the payload of RTP packets and is incremented by one for each data sample, regardless of whether the data samples are transmitted onto the network or are dropped as silent. The timestamp helps the receivers to calculate the arrival jitter of RTP packets and synchronize themselves with the sender.

- Synchronization source identifier (SSRC): 32 bits
  The SSRC field contains a random value that is chosen by a source. It is used by a source as the identifier for each of its data streams and must be unique

within a particular session. RTP specifies a mechanism for resolving collisions in the case that two sources randomly choose the same SSRC.

- Contributing source identifier list (CSRC list): 0 to 15 CSRC items, 32 bits each. The CSRC list contains a list of SSRC identifiers of the sources whose data is combined by an intermediate system to generate the payload of a new RTP packet. The intermediate system is called a mixer and must use its own SSRC identifier for the new RTP packet.

- CSRC count (CC): 4 bits
  The CSRC count specifies the number of CSRC identifiers contained in the CSRC list.

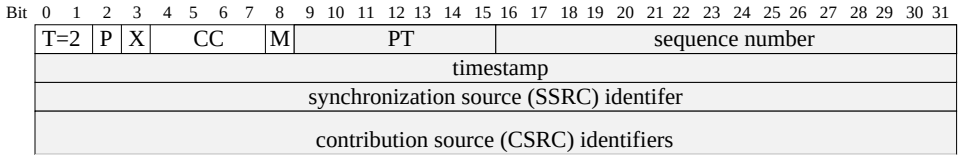


Figure 2.7: RTP header

2.2.3.2 Loss avoidance, recovery and control

We have identified packet loss as an important problem with regard to the deployment of Internet real-time services. In this section we want to briefly introduce (generic) approaches to loss avoidance and recovery in the Internet ([PHH98, KBS+98]).

A large number of different techniques operating either on an end-to-end or on a hop-by-hop basis have been proposed which can be divided as follows:

- Loss avoidance at the application level: sender adaptation (section 3.1.2.3), layered coding/multicasting (section 3.1.2.3)

- Loss avoidance at the network level: per-flow reservation (section 3.2.2.1), per-packet prioritization/aggregate provisioning (section 3.2.2.2), network adaptation (section 3.3.1)

- Loss reconstruction: redundancy mechanisms (section 3.1.2.2)

- Loss alleviation: interleaving (section 3.1.2.1), concealment (section 3.1.3)

Except the first item all methods constitute directly related work which we will build upon in designing our combined approach.

Fig. 2.8 shows a taxonomy with a qualitative classification of the necessary overhead in terms of additional bandwidth and processing at end-systems (or in some

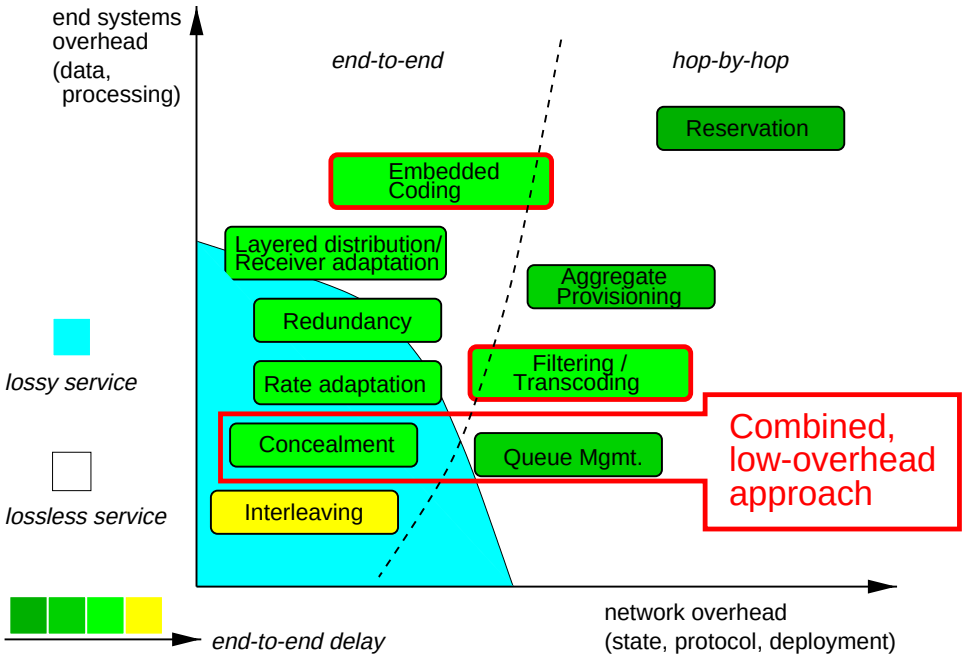


Figure 2.8: Taxonomy of loss treatment schemes for IP-based realtime traffic

cases also within the network), and additional protocol overhead and state which has to be maintained within the network. The end-to-end approaches typically do not actively involve the network, but rely on robust end system protocols and mechanisms. In contrast, the hop-by-hop approaches involve network participation at different levels, thus generally achieving better end-to-end delay properties and (near) lossless service (shadings in Fig. 2.8). Clearly, the associated overhead of both approaches influences overall deployment and *scalability*. Scalability is a major concern, considering a scenario with the presence of numerous, low-bandwidth voice flows in the Internet, because the methods either introduce high per-flow state overhead in Internet routers (reservation), data overhead (redundancy mechanisms) or delay overhead (interleaving, receiver-based concealment).

Chapter 3

Related Work

In section 2.2.3.2 we have briefly introduced a taxonomy of generic approaches to loss avoidance and recovery. In this chapter we will introduce these methods in a more detailed way and present how they are applied for Voice over IP. The first section presents methods to recover losses at the end-to-end level which are typically to be grouped into the "intra"-flow QoS category (Table 1.1). In the second section we discuss hop-by-hop loss control mechanisms which are either purely local or distributed. It should be noted that typically the local methods can be classified as "intra"-flow due to the limited knowledge of the algorithms (the converse is true for the distributed methods). Finally we present the (few) existing approaches which aim at a combination of end-to-end and hop-by-hop mechanisms.

3.1 End-to-End loss recovery

To cope with the packet loss problem on an end-to-end basis, i.e. without modifying the network itself, much research has been done to develop schemes for open-loop error control for voice transmissions over the Internet ([Jay93, PH98, PHH98, CB97a]). Figures 3.1 and 3.2 illustrate the generic structure of audio tools with such mechanisms. In parallel to the conventional encoding and packetization process, analysis modules working before or after the encoder extract redundant information from the signal (another option is that information available during the coding process is used: "encoder-based analysis"). The generated information can then be used to influence the way the packetization is done (interleaving, section 3.1.2.1) or can be added as side information to the data to be transmitted (Forward Error Correction, section 3.1.2.2). The amount of side information ("redundancy") can range from a simple pitch period measurement as for the AP/C scheme presented in chapter 5.1 over information to recover the basic envelope of the speech signal up to running entire speech encoders.

At the receiver (Figure 3.2) the transmitted redundancy is extracted and packet losses are detected. Then as much as possible of the encoded stream is reconstructed[1]

[1]Note that we do not consider the case of a "pre-decoder concealment" here, because it would typically duplicate the internal decoder concealment. However such a function could make sense for

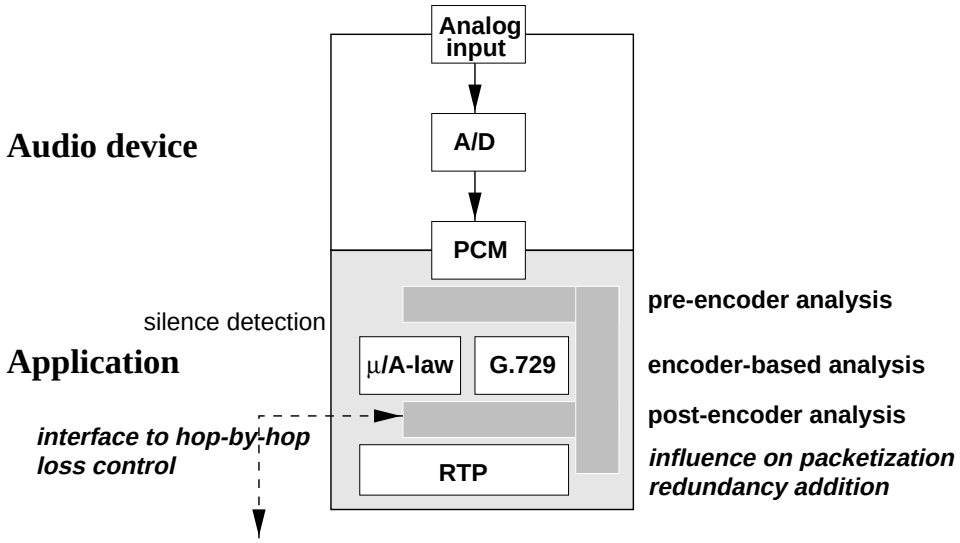


Figure 3.1: Generic structure of an audio tool with loss recovery (sender).

and fed into the decoder. As for the encoder also the decoding process itself can be influenced where possible ("decoder-based concealment") to alleviate the impact of losses. Finally after the decoder the signal can be processed to increase the signal quality further.

Additionally to the loss impact on the decoding process described in section 2.2.1.1, also the performance of end-to-end loss recovery mechanisms like FEC and concealment suffer in the presence of losses, i.e. the number of consecutive packet losses which can be treated is limited. For FEC, the limitation lies in the additional data and delay overhead necessary to detect and recover consecutive losses. For concealment, the limitation in the number of consecutive losses is due to the assumption of quasi-stationarity for speech. This is only valid for a time period typically equivalent to one or two packets. Given these constraints, concealment and forward error recovery approaches become less efficient as the loss burstiness increases (as shown e.g., in [SM90, CKS93]).

3.1.1 Impact of the choice of transmission parameters

Before looking at specific loss recovery mechanisms we discuss the choice of transmission parameters with regard to the impact on the speech quality in the event of a packet loss, which constitutes a loss alleviation option.

3.1.1.1 Packet length

The suitable choice of the speech segment length per packet is a "preventive" measure at the sender.

a distributed operation: some entity (proxy) within the network monitors the stream and conceals / regenerates packets when necessary ([LSCH00]).

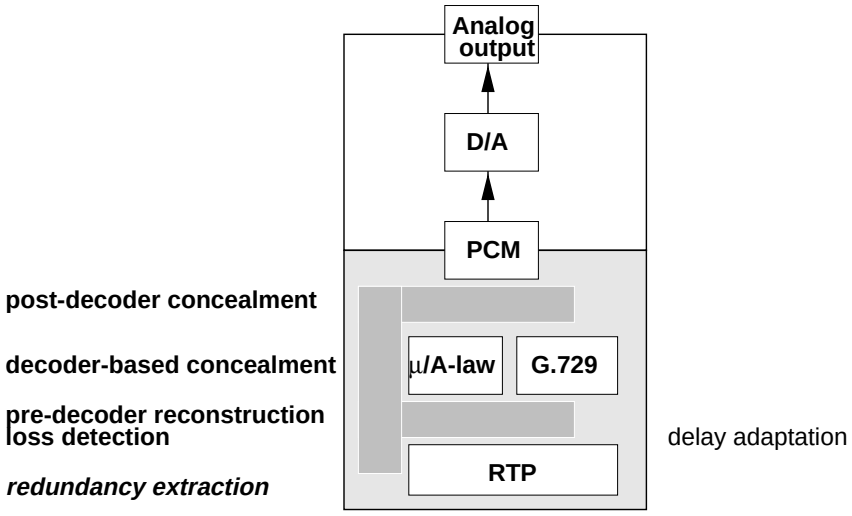


Figure 3.2: Generic structure of an audio tool with loss recovery (receiver).

Speech segment length	*Loss distortion*	*Header Overhead*	*Number of packets per time interval*
$< 2\ ms$	Noise impulses	high	high
$> 32\ ms$	Loss of entire phonemes	low	low

Table 3.1: Choice of the per-packet speech segment duration

The segment length should be chosen to be relatively short, such that the speech signal can be assumed to be stationary for one segment with a high probability. If very small packets are transmitted (Table 3.1), annoying noise impulses will occur. Additionally, the packet header overhead as well as the extreme per-packet processing cost within the network are prohibitive. A large segment length in connection with packet loss may impair the speech intelligibility due to the loss of entire phonemes (see [Min79] for early work on finding an "optimal packet length"). Obviously for frame-based codecs (section 2.1.3) this choice of the segment length is equivalent (and limited) to the choice of the number of frames per packet (see chapter 4.2, p. 79).

In this thesis for all results a packetization duration of $20ms$ is used[2], which may consist of multiple speech frames emitted by a frame-based encoder (chapter 5.2).

3.1.1.2 Compression

Increasing the compression of a speech signal leads to a reduction in the overall amount of data to be sent over the network, i.e. the payload per packet is reduced. Yet the number of packets remains the same (when maintaining the packetization time interval/ play-out delay), thus inducing the same per-packet processing cost

[2]For PCM-encoded speech which has been sampled at $8kHz$ this results in 160 samples per packet and thus (assuming quantization with 8 bit) in 160 octets per packet.

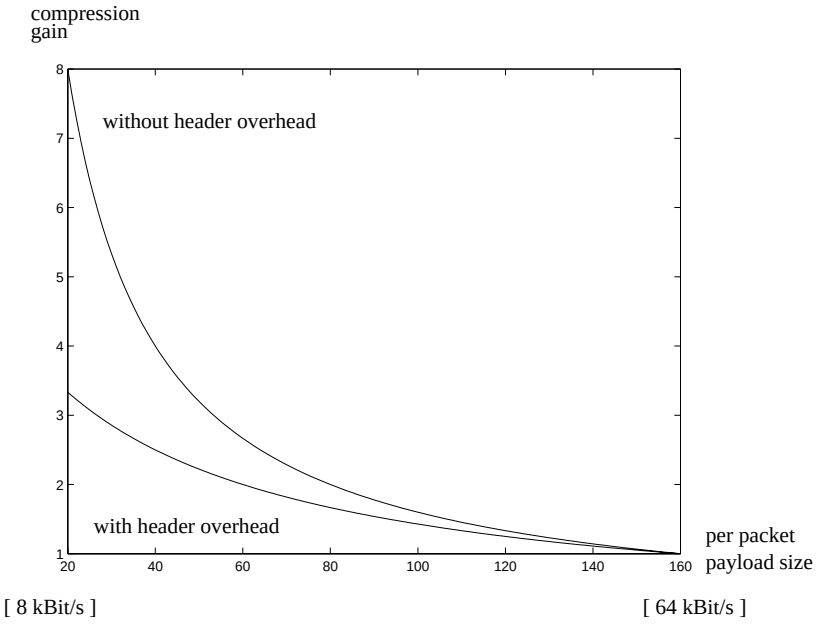


Figure 3.3: Relative compression gain

as before in the network. Additionally, the high per-packet RTP/UDP/IP header overhead diminishes the gain of highly compressed speech, as can be seen in Fig. 3.3. In the figure we plot the compression gain relative to PCM speech quantized with 8 bit, where 20ms speech (sampled at 8 kHz) are contained in a packet and 40 octets per-packet header overhead is assumed. As highly compressed speech is also very sensitive to packet loss (due to encoder/decoder state synchronization), to allow for a reasonable cost/quality tradeoff, multiplexing ([RS96, RS98, JH98, SS98c]) of several voice streams into a single flow (see Figure 2.6) is necessary.

3.1.2 Mechanisms involving sender and receiver

This group of methods is through the involvement of the sender not as flexible and widely applicable as the receiver-only methods (which will be presented in section 3.1.3), however offers much more opportunities to influence the QoS on an end-to-end basis (especially for non-waveform codecs). These schemes allow to perfectly recover at least parts (seen on the time and/or frequency axis) of the original signal. We do not discuss retransmission (ARQ, [CB97a, CSS00]) as an applicable method, as typically the delay constraints (section 2.2.1) together with the delay conditions in the network do not allow to apply this method (though [DLW96] reports the usefulness for voice for some (local area) network scenarios). Recent work in the context of video ([Rhe98]) has shown that retransmission can be used also for real-time transmission to avoid the effect of error propagation (thus a retransmitted packet might not be usable for direct play-out however can be used to update the internal decoder state; see section 2.1.3.2). While such a scheme appears to be less effective for voice it should not be considered impossible.

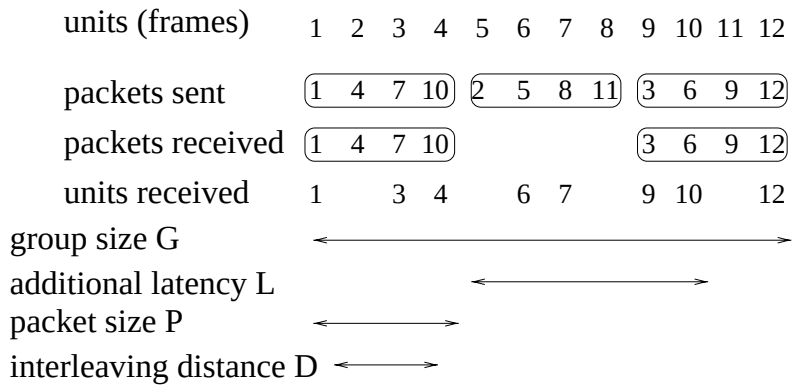


Figure 3.4: Unit interleaving

3.1.2.1 Interleaving

A simple method to increase the audibility of a loss-distorted signal is interleaving ([Ram70, MYT87, VNJ99, Per99] and [PH98], chapter 4.3), i.e. sending parts of the same signal segment in different packets, thus spreading the impact of loss over a longer time period. Particularly for voice this property has been reported to be useful (see "silence substitution", p. 44) in terms of enhanced speech quality due to the long-term correlation property. Interleaving always needs buffering of generated data at the sender and re-sequencing at the receiver, thus introducing a higher latency.

Figure 3.4 shows the interleaving of "units" (e.g. voice frames): a number of units are associated to a group (here the group size is $G = 12$). Units, which are in a certain distance of each other (interleaving distance $D = 3$), are packetized together (packet size $P = G/D = 4$). In the event of a loss, the burst loss of P units is traded against P isolated losses of unit size. The additional latency introduced is thus $L = (P-1)D + 1 - P = (P-1)(D-1) = 6$ units. This delay is added permanently to the play-out delay, because units have to be buffered at the sender before being interleaved and finally packetized. Note that the mean bandwidth of the flow is not changed (no redundant data is generated), however the flow exhibits more burstiness: Packet departure times for the non-interleaved case are after the generation of unit 4, 8 and 12 respectively. When interleaving is used the earliest departure of the three packets of the group is after unit 10, 11 and 12 respectively. In summary the applicability to interactive voice is limited to short groups in conjunction with other loss recovery algorithms (see below).

Sample Interleaving / Interpolation A special case of interleaving is where the unit is equivalent to a sample. Jayant ([JC81]) proposed to put consecutive samples of a waveform coder into two different packets (thus $G = 2P$, $D = 2$) and combine this operation with loss concealment: The speech signal is partitioned into sequences of $x(n)$ ($n \in [1, G]$). The sample with even indices $x(2m)$ ($m \in [1, P]$) are packed into one packet. The odd samples $x(2m-1)$ are put into another packet.

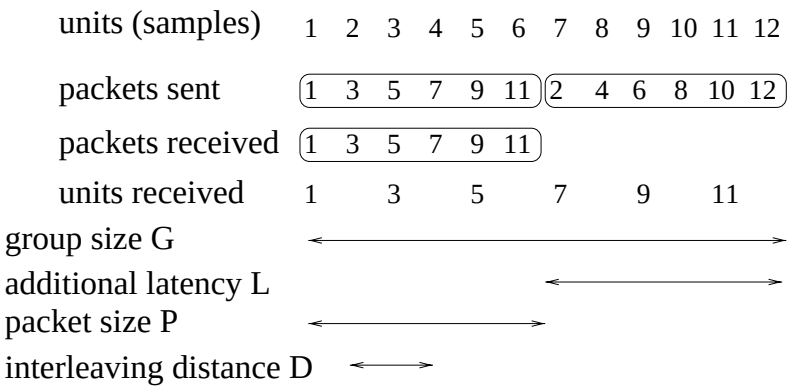


Figure 3.5: Odd-even sample interpolation

If one of those two packets is lost the missing samples can be interpolated using the samples of the respective other packet. Thus it is possible to at least recover the important low frequency parts of the signal. The interpolation problem is easier to be solved than the generation of an entire packet content. The overhead in terms of computation is low. A delay corresponding to one packet length is added at the sender. Figure 3.5 shows the "odd-even" sample interleaving/interpolation ([JC81, Jay93]). Ingle and Vaishampayan ([IV95]) present a similar system using DPCM encoding, where the decoder consists of three sub-decoders with different transfer functions which are used in dependence if only the first, the second or both of the two packets are received.

Multirate representation with LP estimation In [CC97] the sample interpolation scheme is extended to allow also larger values than $D = 2$ ($G = PD$). Additionally, the speech segment of length G samples is represented as a $P \times D$ matrix. Thus one axis is describing the packet number $\in [1, D]$ and the other axis gives the sample position $\in [1, P]$, resulting in a *multirate state-space* representation. When e.g. only one packet out of the group of D packets is received, the problem of $(D-1)P$ missing consecutive samples is shifted to interpolating between P samples with a distance of $D-1$ between them. Now, the linear prediction coefficients (section 2.1.3.2) of the linearly interpolated samples are estimated (or the result for the previous group is used). The estimation is done in the multirate state-space domain by minimizing the mean-square error. This amounts to using a Kalman state estimation technique. Because linear prediction is used, the scheme is also called "model-based recovery". The performance in terms of speech quality is good because the method combines interleaving (with $D = 2, 4$) of short segments (P corresponding to 16 and $8ms$ respectively) with linear prediction. It should be noted that this scheme is sender-based due to the interleaving, but the LP estimation is done entirely at the receiver (as in paragraph 3.1.3.2).

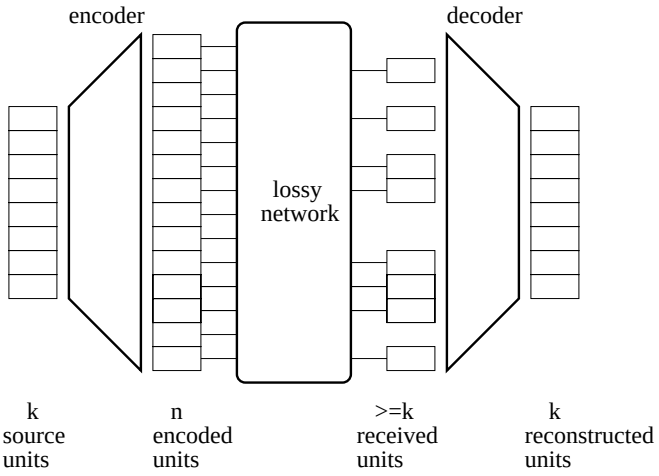


Figure 3.6: Principle of Forward Error Correction

3.1.2.2 Forward Error Correction (FEC)

While interleaving methods just change the way in which the data are transmitted to the receivers, Forward Error Correction adds redundancy for the recovery of lost packets at the receivers. Generally, FEC can be formulated as follows: when some redundancy encoding is applied over k units resulting in $n - k$ redundancy units, the information can be fully recovered if at least k out of the n units are received (Fig. 3.6, [Riz97]).

FEC approaches can be grouped into two orthogonal design dimensions:

- transport: "piggybacking" vs. separate stream of FEC data
- coding: channel vs. source coding

Transport The solution for FEC transport using "piggybacking" ([HSHW95], [Ros97b]) is shown in Fig. 3.7 (for simplicity we assume that for every unit exactly one corresponding redundant unit is generated ($k = 1, n = 2$). Obviously the amount of piggybacked data is highly dependent on the FEC generation process[3] (see the section on FEC coding, p. 40, below).

When a packet is generated, e.g. containing units 9 and 10 (Fig. 3.8), the redundant encodings of earlier data (here: units e and f representing the content of units 5 and 6) are added to the packet payload. In the event of a packet loss (here the packet containing 5 and 6 is lost), the additional payload can be used to recover the loss. Important parameters are the number of piggybacked redundant payloads (redundancy "levels" $n - k$) and their respective distances to the original data (D in Fig. 3.7):

[3] In [FdSeS99] e.g. two independent FEC generation processes are used, resulting in a variable amount of piggybacked data.

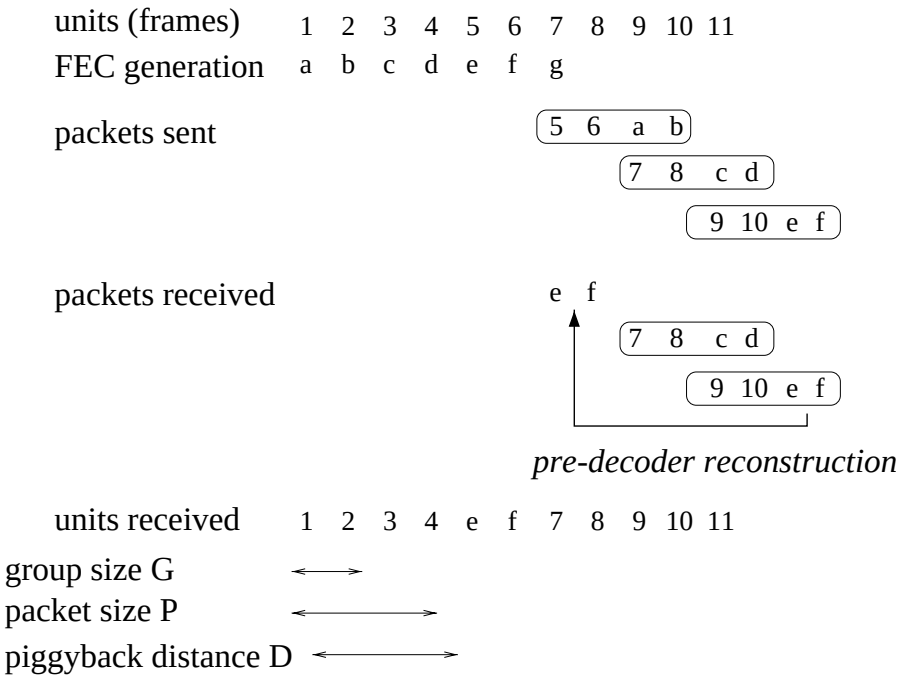


Figure 3.7: Piggybacking of redundant data

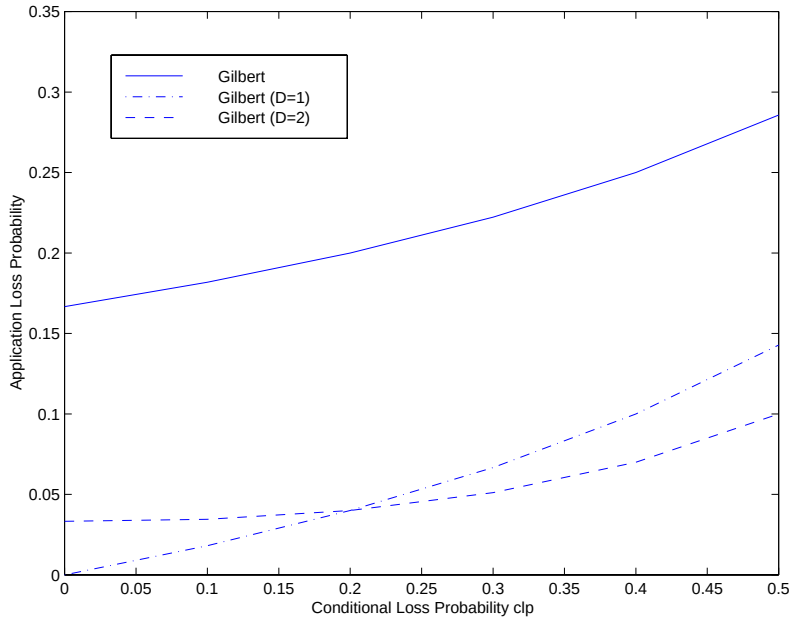


Figure 3.8: Application-level loss probability dependent on the piggybacking distance D ($p_{01} = 0.2$)

The piggyback distance constitutes a tradeoff: on one hand if the distance is increased the play-out delay is increased. On the other hand, if $clp > p_{01}$ (i.e. losses are correlated, section 2.2.1.1, p. 24), by using a higher distance the application-level loss probability is lower. This effect can be seen from Figure 3.8, which shows the application-level packet loss probability when varying the conditional loss probability (clp) and keeping p_{01} constant ($p_{01} = 0.2$). The solid line ("Gilbert") gives the unconditional loss probability ulp, i.e. no FEC is used, using a Gilbert model. The other curves show the application loss probability for the same model with one level of redundancy (i.e. one packet contains redundancy to repair exactly one other packet) for piggyback distances of $D = 1$ and $D = 2$ respectively. The application loss probability can be computed as follows ([BFPT99]):

$$ulp = \frac{p_{01}}{1 - p_{11} + p_{01}} \text{ (No FEC)} \tag{3.1}$$

$$ulp_{D=1} = clp\ ulp = \frac{p_{01} clp}{1 - p_{11} + p_{01}} \text{ (FEC, D=1)} \tag{3.2}$$

$$ulp_{D=2} = ulp\ p_{01}\ (1 - clp) + ulp\ clp^2 = \frac{p_{01}^2 (1 - clp) + p_{01} clp^2}{1 - p_{11} + p_{01}} \text{ (FEC, D=2)} \tag{3.3}$$

Note that for $D = 2$ the two separate terms correspond to two loss patterns 101 and 111, where 1 stands for a lost packet and 0 for a successful packet arrival (see section 4.1.1). The piggyback scheme is relatively simple to implement: only one stream of packets has to be treated for one media flow. This also allows for simple recovery operations (only one sequence number space is needed for a flow; no additional sequence number recovery is necessary). The number of packets sent (corresponding to the induced per-packet processing cost) is the same as for the case without FEC. The per-packet overhead (see RFC 2198 [PKH+97]) is typically less than for sending a separate FEC stream (this amounts to using additional header fields versus an entire packet header with a different RTP payload type). Especially when the payload is relatively small as compared to the header the additional overhead is acceptable: e.g. considering a non-aggregated G.729 data flow with two frames per packet (corresponding to $20ms$ voice) results in a payload of 20 octets (see section 3.1.1.2). If the redundancy unit is of the same size and considering 40 octets of IP/UDP/RTP overhead, this amounts to sending IP datagrams of either $40+20$ or $40+20+20$ octets length. This results in a 33% bandwidth increase (however not considering the additional header needed for payload type and timestamp recovery).

The last two advantages are traded against a lower probability of a successful error recovery as compared to the solution using a separate stream: a loss causes a primary payload *and* a redundancy payload to be lost which poses a problem especially together with using additional source codings as redundancy (see section 3.1.2.2 below).

Considering sending a separate stream, the opposite arguments as just introduced apply ([RS99]). The key advantage of this scheme is the backwards compatibility (some receivers may receive and decode the FEC stream while non-FEC-capable receivers just discard the FEC packets). The non-backwards compatibility for the

piggybacking solution is caused by the needed additional header rather than having a new profile/payload type.

Coding The straightforward way to implement FEC with regard to the coding scheme is to apply well-known methods of the information theory field (parity/ Reed Solomon/ Hamming codes) to blocks of bits corresponding to packets rather than to a stream of bits ([RS99]), often referred to as *channel coding*.

In the following we briefly describe the parity technique ([Ros97b, RS99, ABE+94]) because it is simple to implement (both in the encoder and decoder) and thus has found wide acceptance. The simplest case of parity FEC is considering $k = 2$ units (units x and y). One unit of redundancy is computed ($x \oplus y$), thus $n = 3$. Then if one of the two units as well as the redundancy unit is received, the respective other unit can be recovered:

$$x \oplus (x \oplus y) = y \tag{3.4}$$

Channel coding allows for the exact reconstruction of lost packets independently of specific payload types (voice/video) and specific source coding algorithms. This independence makes it possible to better separate the recovery process from the decoding process (e.g. the delay adaptation algorithm within an audio tool will adapt its delay automatically if packets are delayed due to error recovery [Ros97b]). This "generic" FEC also allows to efficiently protect certain (RTP) header fields (payload type, timestamp; [RS99]). The computational effort is typically small as compared to source coding, however obviously payload-specific properties cannot be exploited.

Early work on using *source coding* as redundancy proposed the transmission of some (redundant) information about some basic speech parameters (short-time energy and zero-crossing measurements, [ECZ93]). This can be seen as a sender-supported loss concealment which suffers from the same problems as concealment (see section 3.1.3: only relatively short gaps in the signal can be recovered).

This work has been extended to using the output of entire source coders as the redundant information. Thus the same signal is basically transmitted several times, encoded with possibly different encodings ([HSHW95, KHHC97, PRM98]). It has been recommended that the secondary encoding (i.e. the redundant encoding) should be encoded with a lower quality source coding. On one hand, this is because the probability that this data has to be used is low (equal to the loss probability ulp), thus the quality impact is not that significant. On the other hand a lot of bandwidth should not be spent on the redundant source coding as $(1 - ulp) \times 100\%$ of the data is wasted (i.e. it is not needed at the receiver). A feature of the scheme is its simplicity: all existing codecs in tools can be used to generate payload-specific redundancy. Generally less overhead (redundancy) than for channel coding is generated as no exact (lossless) reconstruction of the data is desired.

Discussion All combinations of FEC transport (piggybacking or separate stream) and coding (source or channel coding) are applicable: in [RS99] a separate FEC stream together with channel coding is used. Figueiredo et al. ([FdSeS99]) propose

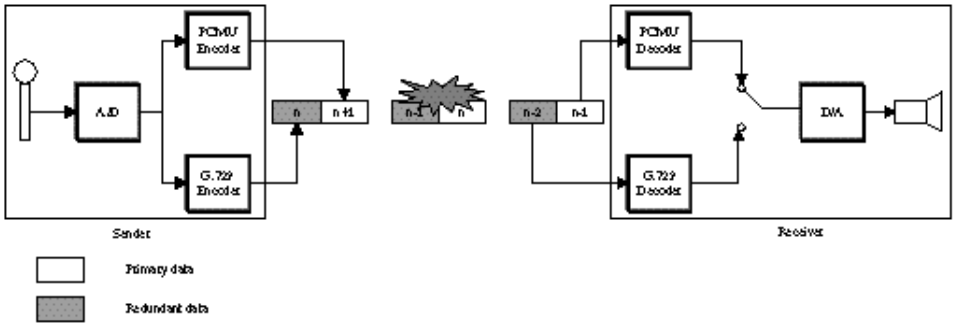


Figure 3.9: Loss of synchronization of the redundancy decoder caused by a packet loss.

piggybacking in connection with channel coding. A "typical" combination however (advocated in several references: [HSHW95, KHHC97, PRM98, PHH98]) is using piggybacking together with source-coded FEC. Yet this specific combination results in the following problem: when a frame is lost, all decoders suffer loss of synchronization and deliver decoded speech signals with bad quality. An example is illustrated in Figure 3.9 where the sender transmits PCM μ-law audio data as primary data and G.729 audio data as redundant data. When a data packet arrives at the receiver, the PCM μ-law audio data is played and the G.729 frame is passed to the G.729 decoder to keep it synchronized with the G.729 encoder at the sender. The output of the G.729 decoder for a frame is discarded if the PCM μ-law data for that frame is also received. If a packet is lost and the following packet is received, the G.729 frame is played to cover the gap in the PCM μ-law audio stream. However, because the G.729 decoder also has just lost a frame ($n-1$ in Figure 3.9), it suffers a loss of synchronization, resulting in a worse quality of the speech signal decoded from the replacement frame (n in Figure 3.9). Thus it is reasonable to run the same coding scheme for the primary and redundant encoding schemes. Another reason for this is to decrease the computational complexity for the decoding process as a whole by running only one decoder.

The overhead of the FEC schemes is significant with respect to the additional data to be transmitted. To accomodate losses, the bit-rate has to be increased first in proportion to the number of consecutive losses to be repaired. Thus FEC mechanisms need to be coupled with control algorithms to avoid harming other flows (see section 3.1.2.3). FEC schemes need also to be couple to the play-out delay adaptation algorithm (section 2.2.1) to avoid a significant increase in the average play-out delay ([RQS00]). Yet, the schemes are useful for reconstructing small bursts of lost packets in a deployment scenario where only few flows use the scheme. They are suitable also for larger packet sizes (when concealment (section 3.1.3) cannot be applied), in the case where packet header overhead is of greatest concern (e.g. using low-speed links). Thus FEC mechanisms constitute a mechanism to promote the

use of multimedia applications in the Internet without first deploying QoS support mechanisms throughout the network.

3.1.2.3 Adaptivity

As an end-to-end intra-flow QoS solution either a rate-adaptive sender (as in [BG96]) or the transmission of the signal encoded in several layers ([Ise96]) with adaptive receivers (as in [TFPB97, MJV96]) can be employed. These approaches must implement mechanisms to assure bandwidth fairness ("TCP friendliness"). Additionally, the network should monitor misbehaving flows, as aggressive applications may monopolize the bandwidth otherwise ([FF97]).

For voice however, it is difficult to realize adaptivity with the currently standardized codecs (see e.g. the description of G.729 in section 2.1.3.2) due to their output in form of a single[4] fixed rate of fixed-size frames. Additionally, considering the low per-flow bandwidth, the per-flow gain using adaptivity is low (as compared to video where the adaptivity may range over one order of magnitude in bandwidth, [SS98a]). When considering large groups with heterogeneous receivers, difficulties in choosing the proper adaptivity strategy (sending rate/layering) to suit all receivers arise. Due to these reasons (which we will elaborate in the following two paragraphs), we do not consider adaptivity in this thesis.

Sender Adaptation Rate adaptation, i.e. varying the coder output bit-rate according to (RTCP, [SCFJ96]) loss reports by receivers, is currently not feasible for speech transmission due to the lack of a codec which offers such flexibility as mentioned above. However such codecs (e.g. wavelet codecs, [Ise96]) are under development, but did not find wide deployment yet. Bolot ([BG96]) proposes to switch between available codecs for non-continuous bit-rate adaptation. We argue that this is problematic due to the non-linear (or even non-continuous) relation between the bandwidth and the subjective quality of the codecs. The MOS (subjective quality, Table 4.5) values for the codecs employed do not differ much (e.g. the ITU codecs G.723.1, G.729, G.728, G.726 and G.711 cover a bit-rate range from 5.3 kbit/s to 64 kbit/s while the subjective quality differs by less than 0.25 on a 1-to-5 MOS scale ([CK96]), which covers the quality range from "bad" to "excellent". Additionally, considering the service model, when switching codecs the choice of the codec/subjective quality is taken away from the user and it could be argued to take always the codec with the best quality / bit-rate relation (assuming the availability of sufficient computing power which will typically be the case).

For rate adaptation, the low per-flow bandwidth has to be considered together with the necessary overhead in terms of *feedback* (RTCP control traffic). Generally, to react properly to either transient or persistent congestion, it is crucial to receive up-to-date feedback information from receivers, which will not easily be feasible for large multicast groups (RTP scales down its feedback interval with the group size to

[4]The G.723.1 ([Uni96c], Table 2.1) codec offers two output bit-rates and the possibility to switch between without a quality impairment, however the bit-rates are not very different ($5.3 kbit/s$ and $6.3 kbit/s$).

ensure that only a fixed amount of session bandwidth is used for RTCP ([SCFJ96]) control traffic). Yet even for unicast it may be difficult to realize this on long paths with congestion in both directions.

An important combination of loss recovery mechanisms is the association of FEC (section 3.1.2.2) schemes and rate adaptivity, as the amount and distribution within the packet stream of FEC data has to be chosen carefully ([Gar96, BFPT99]). Podolsky et al. ([PRM98]) evaluated the performance of FEC schemes, considering the impact of adding FEC for the voice fraction on the network load. They have shown that if an increasing number of flows uses FEC, the amount of FEC has to be carefully controlled, otherwise adding FEC can be detrimental to overall network utilization and thus the resulting speech quality. They used however theoretic rate-distortion curves not backed by either subjective testing or objective speech quality measurements. Using the terminology introduced in chapter 1 (Table 1.1), adaptivity is needed to realize FEC as an intra-flow instead of an inter-flow QoS enhancement scheme (inter-flow QoS means here protecting one best effort flow on an end-to-end basis at the expense of another best effort flow). Note that if the FEC data is a source coding itself (section 3.1.2.2/Coding) the comments from above on adaptation also apply to the redundant data. Also, the up-to-date feedback from the receivers about the loss process is as crucial as for the main payload. Bolot et al. have presented a combined rate and error control algorithm which determines the optimal amount of side information to be transmitted in addition to how the stream is partitioned between the main data and the redundancy. While the scheme is appealing it is mainly useful only with a truly bandwidth-scalable codec. Furthermore, the impact of a large-scale deployment (as in [PRM98]) and the impact of the feedback delay on the adaptation quality in such a scenario need still to be assessed.

Receiver Adaptation Receiver-based adaptation presumes that the signal that is transmitted is decomposed into several "layers" of which at least one is decodable by its own. Furthermore it is necessary that receivers can request the number of layers they want to receive ([MJV96, TFPB97]). The IP Multicast architecture ([SM96]) offers a suitable framework, as the individual layers can be mapped to different multicast addresses. Then, receivers can join these groups to receive the traffic. If they leave a group and nobody else requested the delivery of data belonging to that group, the multicast delivery tree is pruned back and the subnet of the receiver (and possibly upper branches of the tree) will be relieved of the traffic associated with that group decreasing the probability of congestion. For voice, besides the problem of a suitable codec for such schemes, the gain in flexibility might not justify the layer resynchronization overhead (as compared to simulcasting the signal in different qualities). It should be noted that the described loss avoidance mechanism is closely related to approaches which map the layering on prioritization (see section 3.3.2).

3.1.3 Receiver-only mechanisms: loss concealment

A speech signal can be (roughly) partitioned into voiced and unvoiced regions. Voiced signal segments show high periodicity (pitch period, cf. chapter 2). When packetizing, the contents of consecutive packets resemble each other. Concealment algorithms try to exploit this by processing the signal segments around the gap caused by a lost packet and then filling the gap appropriately.

Usual concealment schemes are "receiver-only", i.e. they do not introduce additional implementation, processing and data overhead at the transmitter and are thus well suited for heterogeneous multicast environments. This means that transmitters may use different audio tools than the receivers, and receivers can mitigate packet loss according to their specific quality requirements. Additionally, backwards compatibility and thus simple deployment is assured.

However, the applicability is limited to isolated losses of small to medium-sized packets (the quasi-stationary property of the signal can be assumed with a high probability only for speech segments smaller than about $40ms$). To conceal with a high output speech quality, a high number of successfully received packets around the gap are necessary, resulting in additional play-out delay[5]. As the fixed packetization interval is unrelated to the "importance" of the packet content and to changes in the speech signal, some parts of the signal cannot be concealed properly due to the unrecoverable loss of entire phonemes.

3.1.3.1 Silence substitution

The simplest possibility of loss treatment is to replace the missing speech segment by samples with the value 0 ("silence substitution" or "zero stuffing", [GLWW86, San95]). However even for very low loss probabilities ($ulp > 0.01$ for typical packet lengths, cf. section 3.1.1.1) the speech quality turns out to be unacceptably low (see the discussion in section 2.2.1.1).

3.1.3.2 Waveform substitution

The replacement of a missing signal segment by another segment which is generated from correctly received speech (and possibly processed further) is called "waveform substitution". The procedure can be described as follows ([GWDP88]):

- identification of gaps in the signal as either a missing packet or silence (when silence detection (section 2.2) is enabled) using sequence number and timestamps ([SCFJ96]),

[5]Note that loss concealment algorithms typically add a delay of at least that corresponding to one packet length, because the algorithm is triggered only when a missing packet has been detected. If the packet following the missing packet is needed only for detection and not for the concealment operation itself, the concealment algorithm could be started immediately after the receipt of the previous packet and prepare a replacement packet without any indication if the packet under consideration will really be lost. This behavior constitutes a tradeoff between a higher permanent computational load versus a lower playout-delay (cf. section 5.1.4).

- buffering of recently received signal segments,
- signal processing to replace the missing segment.

Only replacement segments which represent a "short" speech segment (cf. the introduction to this section) will yield in most cases a high speech quality. For larger segments if the speech within the segment has not been stationary, additional distortion is introduced. Thus especially in transition areas of phonemes of different types ("voiced", "unvoiced"), waveform substitution is problematic.

Noise Insertion The next step after using silence substitution with only slightly increased complexity is to use noise as a replacement for the missing speech segment. Noise insertion exploits the effect of "phonemic restoration" ([PHH98]), i.e. that the interpolation ability of the human auditory system is increased if noise rather than silence is perceived instead of the missing speech segment. This has been reported to be true for both intelligibility and quality.

In addition to receiver-based noise generation, it is possible to use information transmitted by the sender for appropriate noise generation. This is proposed in the context of silence detection (section 2.2.2), where during silent periods the sender sends "comfort noise" indication packets[6] (carrying the noise power level) for appropriate noise generation during "silent" periods (in fact the play-out of actual silence instead of ambient noise is perceived as disturbing by listeners). The indication packets may thus also be used in the loss repair process.

Packet Repetition The repetition of the most recently received packet is the simplest method to approximate the missing waveform. It is only necessary to buffer a copy of the last packet. Fig. 3.10 shows the original signal $s(n)$, a signal $\tilde{s}(n)$ with every second packet lost, as well as the resulting signal using packet repetition $\hat{s}(n)$ in the time domain, where n is the sample number. Because the packetization interval L is not related to the speech pitch period p, discontinuities in the signal occur (Fig. 3.10). Together with a typically reverberating sound caused by exactly the same speech material to be played twice, this method results in a only slightly improved speech quality as compared to silence substitution ([LBL92]).

Pattern Matching The Pattern Matching technique ([GLWW86, GWDP88], [San95]) repeats a correctly received signal segment of which maximum similarity with the lost segment is assumed. This is accomplished by matching a sample pattern immediately preceding the gap to a series of samples received earlier. As entire signal segments of at least one packet duration are completely repeated, this may cause (as for Packet Repetition described above) echoing sounds.

[6]For RTP (section 2.2.3.1), the RTP profile defined in [SC00b] defines a generic comfort noise. Additionally, several codecs (G.723.1, G.729, GSM) have codec specific comfort noise data that are triggered by specific bits in the coded data stream.

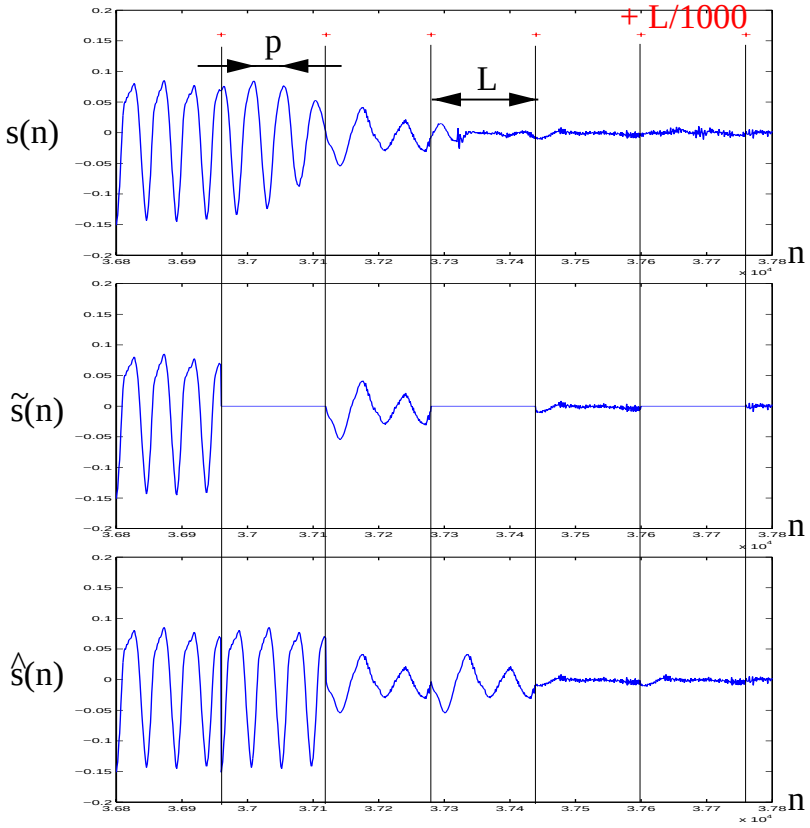


Figure 3.10: Packet repetition loss concealment

Pitch Waveform Replication Echoes can be avoided by Pitch Waveform Replication ([RS78, GLWW86, GWDP88]) where only one pitch period found in the most recently received packet is repeated throughout the missing packet. This is accomplished by measuring the pitch period of the signal content immediately preceding the gap and copying a sequence of samples of the pitch period length until the gap is filled (Fig. 3.11).

An extension to this technique called Phase Matching ([VA89]) provides for synchronization on both edges of the substitute, thus reducing a clicking distortion caused by the discontinuities introduced by the two methods described above. The pitch period is measured before and after the gap. Thus the repetition of the sample sequence is compressed or expanded in time to be in phase with the following signal segment. Thus slight changes in the pitch frequency can be taken into account. Additionally, the amplitude of the repeated segments is adapted according to the difference of the amplitudes before and after the gap.

A technique which can be seen as a combination of PWR and pattern matching is the Reverse Order Replicated Pitch Periods algorithm (RORPP, [Tel99]). While being basically identical to PWR for short signal segments ($\leq 10ms$), for longer missing segments the search algorithm uses earlier segments of pitch period length for

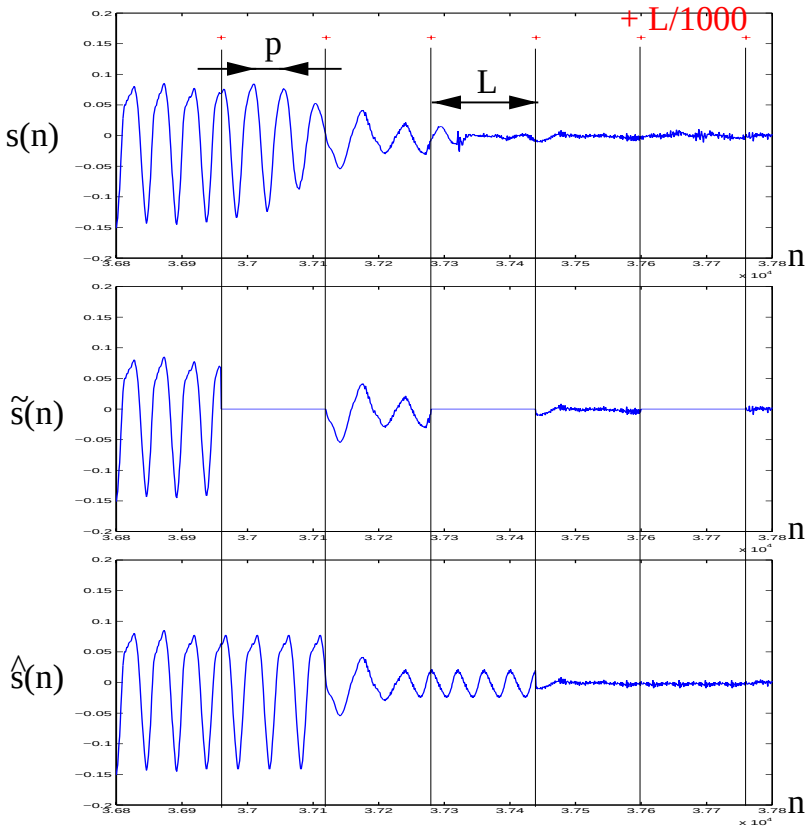


Figure 3.11: Pitch Waveform Replication (PWR) loss concealment

concealment. To avoid discontinuities the algorithm uses extensively the "overlap-add" (OLA) technique which is also called "packet merging" or "blending". In OLA some very short segment ($\approx 2ms$) at the edge of a correctly received speech packet, as well as a segment of the same length of replacement speech material, which should precede or follow that edge, are multiplied with complementary windows. Then both signals in the windowed area are added, thus enabling a smooth transition between the received and the replacement speech.

Time-scale Modification The techniques described in the previous paragraphs have in common that with an increasing length of the lost segment the perceived quality deteriorates severely. That deterioration is in part due to the violated assumption of speech stationarity, however, to a large extent it is due to the specific distortions introduced by the different concealment techniques themselves. This phenomenon is known as the "assymetry effect" ([Bee97]). The time-scale modification (TM) technique introduced in [San95, SSYG96] can overcome this problem by "stretching" a signal segment of a certain length before the gap to cover the segment which is missing (Fig. 3.12). This is done *without changing the pitch period*, i.e. no "new" speech material which might lead to the assymetry effect is introduced. To avoid a discontinuity at the left edge of the gap the replacement speech material is

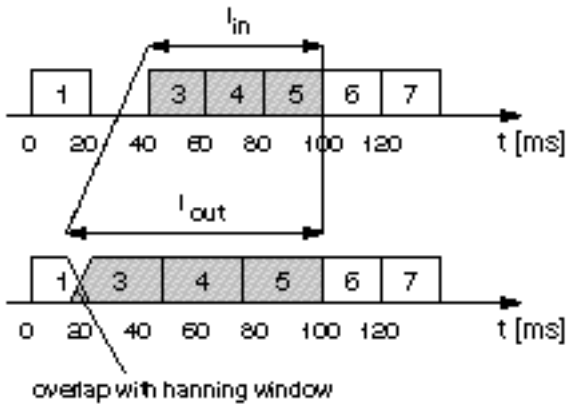


Figure 3.12: Time-scale modification loss concealment

overlap-added to the received speech (packet 1 in Fig. 3.12).

The work is based on the generic time-scale modification WSOLA (Waveform Similarity OverLap-Add, [VR93, Gru94]). In [San95, SSYG96] it is shown by using also a "component judgment" subjective test (section 4.2.2.1) in addition to a MOS test (section 4.2.2) that the techniques discussed previously in this section have a dominant disturbing component ("tinny/metal", "echoing/reverberating", "interrupted/clicking"), while such a component cannot be identified for TM. The MOS results are higher (especially for larger loss gaps). A disadvantage of the method is the relatively high additional delay introduced (it is recommended to use speech material representing a time interval of $60ms$ for the concealment of a $20ms$ segment), because it is not only necessary to buffer the amount of $60ms$ speech, but to withhold it from the play-out buffer (because all buffered samples are modified). However as TM is a receiver-only scheme, when the network conditions are good (i.e. loss and delay are low) and thus loss concealment is not needed at all, this lower bound can be disabled through the delay adaptation algorithm. As for Pitch Waveform Replication described in the previous paragraph, it was proposed to adjust the phase of the replacement signal at both edges ([SRG97]) further improving the quality.

LP-based waveform substitution Most of the described schemes in the previous sections apply well-known methods of speech processing (pitch estimation, etc.) for loss concealment. Thus also the well-known technique of linear prediction (paragraph 2.1.3.2) is an interesting candidate to alleviate the packet loss problem.

Fig. 3.13 shows the approach based on linear prediction proposed by Clüver ([Clu98]). When a packet is correctly received, the PCM signal $x(n)$ (represented by its z transform $X(z)$) is used to compute the LP filter coefficients. The difference signal ($D(z)$) is then fed to the LP synthesis filter ($\hat{D}(z) = D(z)$) which uses the computed filter coefficients, resulting in an output signal ($\hat{X}(z) = X(z)$) which is

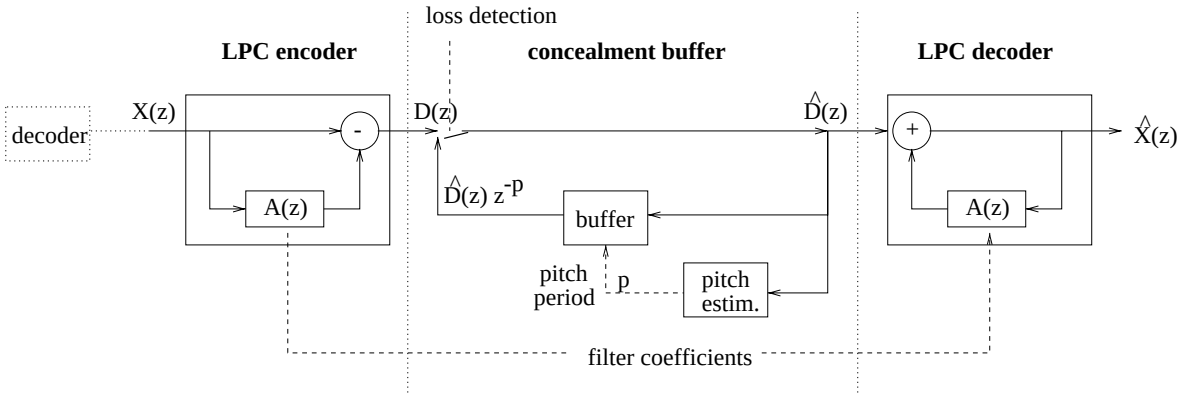


Figure 3.13: LP-based waveform substitution.

identical to the input signal (assuming ideal filters[7]). Additionally to these operations, the pitch period p is estimated and a segment of the LP difference signal corresponding to this period is buffered. When a packet loss is detected, no LP analysis is performed, however the previous difference signal ($\hat{D}(z)z^{-p}$) which has been buffered is used as the replacement excitation to excite the LP synthesis filter using the previous filter parameters.

In [Clu98] this method is extended to comprise voiced/unvoiced detection as well as LP-based waveform substitution in sub-bands. Results show that the sub-band/LP-based method is superior to pitch waveform replication especially for packet lengths larger than $20ms$, however it is not certain if the achievable gain in speech quality justifies the additional implementation complexity. The presented method is similar to the multirate state space method introduced in section 3.1.2.1, however there the LP synthesis is closely tied to the interleaving/sample interpolation process. The described LP-based waveform substitution works (as the other waveform substitution techniques) on the decoded signal (post-decoder concealment, Fig. 3.2). This concept allows for modularity (the concealment component is independent of the decoder) and thus simple deployment. For sample-based codecs, especially simple memory-less companded PCM, this works fine. However when using highly adaptive codecs, the PCM signal used for the concealment LP analysis is already degraded by the coding scheme. Furthermore, for such codecs errors propagate also into signal segments adjacent to the loss gap and thus decrease the achievable quality for subsequent waveform substitution with this scheme. Therefore in the following section we look at concealment mechanisms which are integrated into the decoding process.

[7]Note that due to the limited filter precision, some distortions are introduced during this process. An obvious solution to this is to detect successful packet arrivals and use $X(z)$ directly for the play-out.

3.1.3.3 Codec-specific concealment

For codecs which are based on a linear prediction ([Uni96a, Uni96c]) or transform coding, it is possible that the decoder algorithm is run with repeated or estimated parameters. So the problems described in the previous section with a concealment process being disjoint from the decoder are avoided and no significant additional computations apart from the usual decoding process have to be done.

In section 2.1.3.2, p. 19, we have described the operation of the G.729 decoder. As a typical example how frame losses are concealed, we now describe the internal concealment algorithm of the G.729 decoder:

When a frame is lost or corrupted, the G.729 decoder uses the parameters of the previous frame to interpolate those of the lost frame and performs loss concealment to reduce the degradation of speech quality of the reconstructed speech signal. In particular, the following steps are taken:

- The line spectral pair coefficients of the last good frame are repeated.
- The adaptive- and fixed-codebook gains are taken from the previous frame but they are damped to gradually reduce their impact.
- If the last reconstructed frame was classified as voiced, the fixed-codebook contribution is set to zero. The pitch delay is taken from the previous frame and is repeated for each following frame. If the last reconstructed frame was classified as unvoiced, the adaptive-codebook contribution is set to zero and the fixed-codebook vector is randomly chosen.

When a frame loss occurs, the decoder cannot update its state, resulting in a divergence of encoder and decoder state. Thus, errors are not only introduced in the current frame but also in the following ones. In addition to the impact of the missing codewords, distortion is increased by the missing update of the following internal state parameters:

- The predictor filter memories for the line spectral pairs.
- The linear prediction synthesis filter memories.

Section 5.2.3 will give results on the performance of the described scheme.

3.2 Hop-by-Hop loss control

End-to-end loss recovery is very useful especially with regard to its simplicity of deployment. However the performance of the various techniques is highly dependent on the parameters of the actual loss process within the network. Controlling the loss process (intra-flow QoS) can modify these parameters, if it is not possible to avoid losses altogether for particular flows (inter-flow QoS).

In this section we first discuss purely local approaches which are typically limited to intra-flow QoS enhancement. Then we review inter-flow QoS approaches, discuss their applicability to voice and identify useful mechanisms which can be mapped to the intra-flow QoS case.

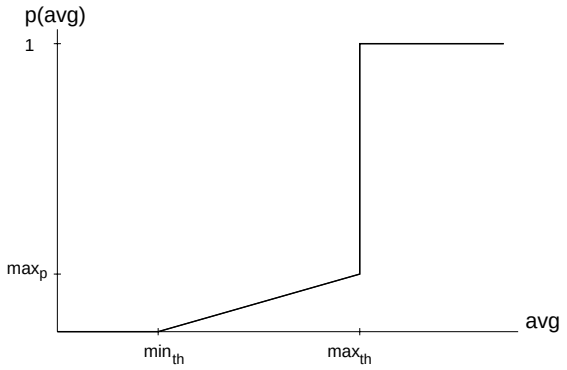


Figure 3.14: RED drop probabilities

3.2.1 Local approach: queue management

Currently, the most widely used mechanism for queue management in the Internet is FIFO (First-In-First-Out) with tail dropping on queue overflow. Queue management methods which are controlled purely local (i.e. only local criteria are used in a dropping decision) try to improve on the simple "drop-on-overflow" discipline in terms of achievable throughput and minimal delay for all flows.

The most widely known representative of this category is RED (Random Early Detection, [FJ93]). RED influences the probability of a packet drop before the queue is full: the measurement of an average queue size triggers random suppression of packets with an increasing probability p as the average queue size (avg, Fig. 3.14) increases. This signals congestion to adaptive flows (TCP), reduces the average delay and allows bursty traffic to be better accommodated, while still maintaining a utilization similar to a drop tail queue. The random dropping takes effect only between a minimum (min_{th}) and a maximum threshold (max_{th}). This is done to avoid on one hand unnecessary packet drops during temporary congestion and on the other hand to drop packets quickly when severe congestion has been detected. Local queue management mechanisms are able to improve the overall performance of best effort networks, however they are obviously limited in their achievable performance goals and still suffer from misbehaving flows. Therefore it is proposed in [FF97] to extend RED by identifying (and discriminating) such flows. Although work on queue management for multimedia flows exists ([PJS99]), only (static) inter-flow QoS is addressed. However we argue that queue management is a good candidate to be extended to also enhance intra-flow QoS requirements of multimedia flows (chapter 6).

3.2.2 Distributed approaches

3.2.2.1 The Internet Integrated Services architecture

A lot of work has been devoted recently to explore service differentiation in the Internet on a per-flow basis, in particular in the context of the IETF Integrated

Services model ([BCS94], Fig. 3.15, cf. Fig. 2.6). These approaches, which we classify as inter-flow QoS (chapter 1, Table 1.1), provide mechanisms to isolate flows from each other, to establish rate and delay guarantees and to provide controlled sharing of excess bandwidth ([GP98]). Flows are described by their traffic envelope using token and leaky buckets. If flows violate their contracted traffic profile, packets are delayed, discarded or treated as best effort.

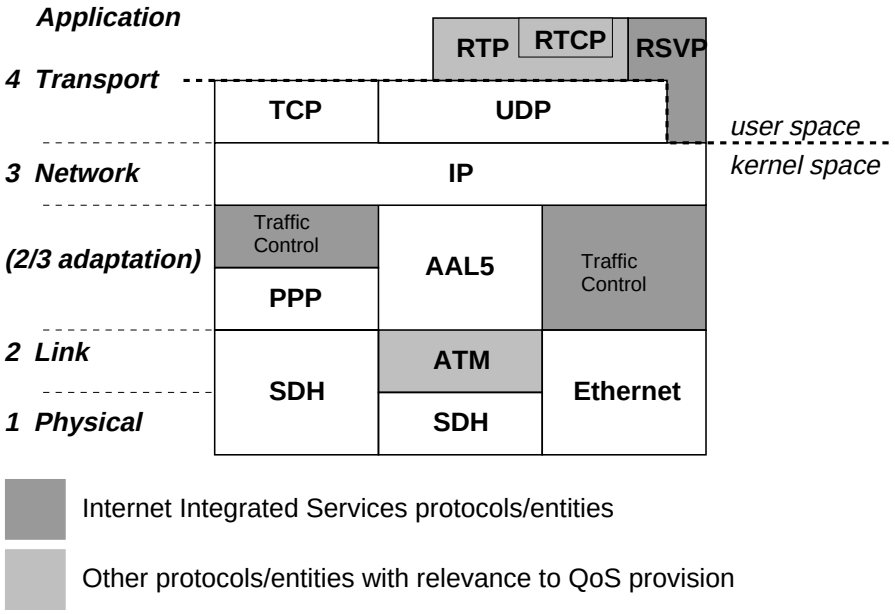


Figure 3.15: Integrated Services protocols and entities

The functional blocks required at every individual hop to establish the Quality of Service guarantees can be described as follows:

1. *Signaling* - registration of senders and receivers.
   The senders advertise the traffic specifications (TSpec) of the flow. Receivers then use this information to request the desired Quality-of-Service using the Flow Specification (Flowspec = Receiver-Tspec plus Rspec), where the Rspec is a parameter describing the desired end-to-end service level (e.g. the end-to-end delay). The signaling in the Integrated Services model is realized by the Resource Reservation setup Protocol (RSVP, [BZB+97]).

2. *Classifier* - association of an IP datagram to a flow.
   The IP layer constitutes a multiplexing layer for packets coming from various network interfaces (at a router) or various local UDP/TCP sockets. After the routing decision has been taken and the packets are de-multiplexed to the correct outgoing interface it is necessary to associate the packets to the respective flows (or the "best effort" class) to be able to schedule the departure of the packets over the network interface correctly. For IPv4 the classification is done by matching a packet's IPv4 source and destination address, the protocol

ID as well as the transport layer ports against the parameters obtained via the QoS signaling protocol.

3. *Packet Scheduler* - schedules the order in which the queues are served to which the packets have been associated by the classifier are served.
   It should be noted that the displayed structure of the traffic control elements (Fig. 3.16) is valid for a QoS-passive medium, i.e. when the link layer does not implement QoS control mechanism (Fig. 3.15: layer 2/3 adaptation). For complex link layers these mechanisms need either be mapped on or replaced by the respective link layer means (e.g. in [SCSW97, eCS+97, SWZS99, AAOS98] this is described for ATM (cell switching) as a link layer using ATM and IP signaling respectively).

4. *Policy Admission Control* - administrative admission of reservation requests.
   Using both information obtained using the QoS signaling protocol (RSVP) as well as dedicated policy protocols, the policy admission control checks if the flow is authorized to receive the desired QoS.

5. *Capacity Admission Control* - admission of reservation requests in terms of available resources.
   Besides global and local policy constraints, the resource usage at a particular network element needs to be taken into account when admitting reservation requests. A Capacity Admission Control algorithm could use either only the maintained state about admitted reservations or can take into account the actual resource usage (Measurement-Based Admission Control). Another design dimension is if only the current reservation requests/usage is monitored or if future states are taken into account (Resource Reservation in Advance: ReRA).

Figure 3.16 shows the described functional blocks and their interaction.
The Integrated Services model comprises two service classes:

- *Guaranteed service* ([SPG97]): this service is intended for non-adaptive flows which need a strict delay bound (e.g. distributed simulation tools and distributed games). By exporting information from every network element and forwarding this information towards the receiver, it is possible to achieve a mathematically provable bound on the delay and 0% packet losses (congestion loss, see section 2.2.1.1, p. 23)

- *Controlled Load service* ([Wro97]): here (as above) the mean loss seen over large time intervals should approximate the link error rate, i.e. virtually no congestion losses/system losses occur. However no commitment about the expected end-to-end delay is made.

Applicability with regard to QoS signalling (RSVP) The major drawback of RSVP, namely, its inability to scale with respect to the number of flows due to

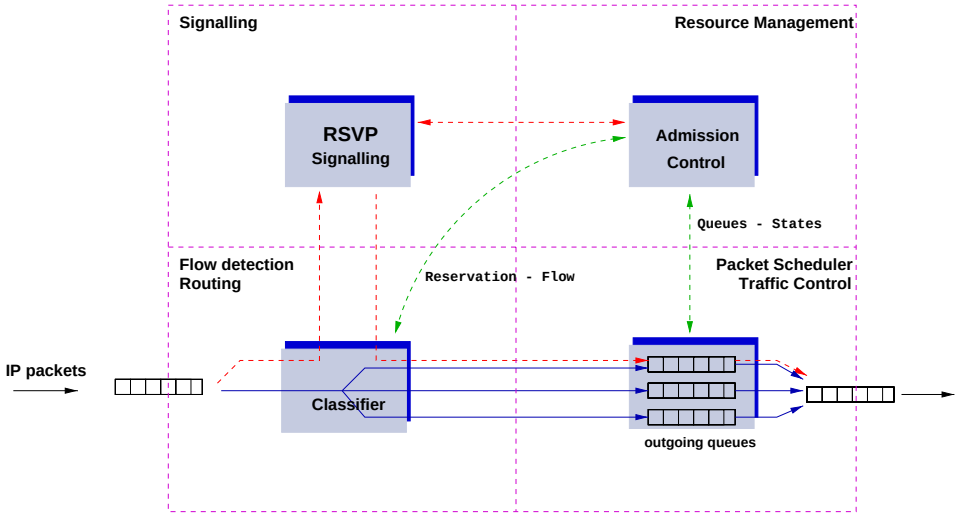


Figure 3.16: Functional blocks of a network element (router) in the Integrated Services model

per-flow state maintenance and processing is relatively well understood and reported ([PS98]). Due to the typically small per-flow bandwidth for voice flows (and thus high state overhead to bandwidth ratio), this property is particularly important for voice.

Additionally, due to the similarity of voice flows, some options of RSVP (which are very useful for other flow types) are rather a burden than a feature. An example is the initial exchange of sender and receiver Traffic Specifications (TSpecs) in RSVP PATH and RESV messages, where the sender advertises its traffic properties (which are most probably well known at the receiver through e.g. the RTP payload type, [SCFJ96, SR98]). The receiver then typically reserves exactly with these parameters (see the paragraph on service model applicability below).

The near-immediate setup of a connection in a circuit-switched network should be approximated in a packet-switched network as far as possible. However the two-way end-to-end reservation setup might take significantly longer (RSVP processing/state update, admission control, traffic control configuration) which advocates pre-configuration in parts of the network. To be able to access pre-configured resources additional mechanisms like the association of RSVP session to groups or packet marking (section 3.2.2.2) are necessary.

A sender-based approach to a reservation protocol like YESSIR ([PS98]) seems thus much better suited to accomodate voice flows. If an adequate basic provisioning for aggregated voice flows is possible, even the operation without any per-flow QoS signalling is possible by using the mechanisms described in this thesis to achieve a graceful degradation under temporary congestion.

Applicability with regard to the services classes In addition, a mismatch between the properties of the currently existing Internet service classes and the

requirements of telephone-quality speech traffic can be observed: the Guaranteed service is intended for non-adaptive flows which need a strict delay bound. However, all typical voice applications can adapt fairly well to changing delay (jitter, section 2.2.1)[8]. The Controlled Load service offers a service which can be expected from a lightly-loaded best effort network, i.e. virtually no congestion losses occur. However, as we will see in chapter 4.2 this service is somewhat too conservative as voice is relatively tolerant to losses as long as the mean loss rate is bounded and the loss correlation is controlled.

Applicability with regard to the service model The Integrated Services model is well suited for fully protected flows, i.e. for flows that obey the traffic contract, and for network services that ensure very low packet loss rates for such flows. However, there are shortcomings if the user wants to pay only for a partial reservation. Integrated Services flowspecs allow to request such partial reservation which might result in temporarily non-negligible loss rates. As all packets within a single flow are treated as if equally important, the current state of the traffic shaper/policer, the scheduler policy and the congestion situation at the network element determine *which* packets are shaped, policed or dropped. Thus a rather conservative in-advance traffic characterization of the flow with regard to the inter-flow QoS control is necessary to avoid an uncontrolled impact on the intra-flow QoS and thus on user perception (see section 4.3). For real-time traffic (audio and video) this means that known properties of user perception or satisfaction in response to packet loss are not taken into account.

3.2.2.2 The Differentiated Services architecture

The Differentiated Services (DiffServ) architecture ([BBC+98, Kil99]) focuses on only qualitative QoS assurance on a per-packet basis which has better scaling properties by only maintaining per-flow state at the edges of a network and enforcing hop-by-hop QoS for aggregated traffic in the network core. The specific treatment of a packet is triggered by the DSCP (DiffServ Code Point) byte it carries in the header. The type of treatment is specified with different Per Hop Behaviors (HB). Currently standardized are the Expedited Forwarding PHB (EF, [JNP99]) which allows preferred treatment of packet in terms of delay. Thus a mechanism implementing EF must consist of at least two queues with a scheduler. The other PHB is Assured Forwarding (AF, [HBWW99]) which aims at the provisioning of bandwidth. Thus to avoid reordering of packets of one flow carrying different DSCPs, AF should be realized using one queue. One particular approach to do this is RIO ('RED with IN and OUT', [CF97, MBJMD99], cf. Fig. 3.14). With RIO, two average queue sizes are computed (Fig. 3.17): one just for the IN packets and another for both IN and OUT packets. Packets marked as OUT are dropped earlier (in terms of the average queue size) than IN packets. Thus the desired differentiated treatment of packets can be realized using a single queue, while maintaining the desirable properties of

[8]The service can however be important for special (911) calls e.g.

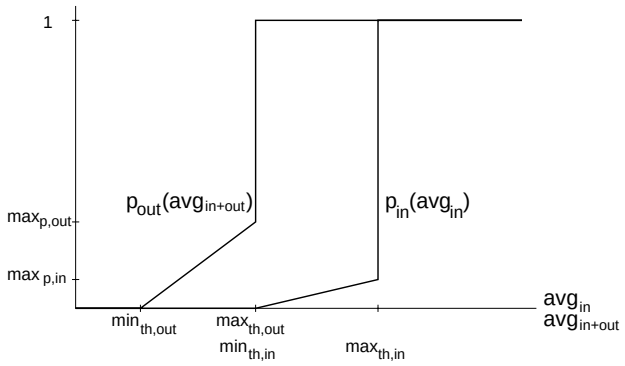


Figure 3.17: RIO drop probabilities

RED (cf. section 3.2.1).

Besides the advantage of aggregation, per-packet QoS offers also the opportunity that an application may control the desired QoS on a per packet (and thus per ADU - Application Data Unit) basis. Thus intra-flow QoS requirements can be mapped on network prioritization. Therefore we consider the Differentiated Services architecture to be a framework for one of our approaches to intra-flow loss control (section 6.3).

3.3 Combined end-to-end and hop-by-hop approaches

While the approaches presented in chapter 3.1 are network-aware in the sense that they "repair" packet loss or even adapt to the current network congestion state (sender adaptation, p. 42), they operate in a best-effort network in which all packets are of the same importance. In this chapter however we will now review approaches which employ combined mechanisms at the end-to-end and hop-by-hop level with both minimal and explicit cooperation (explicit cooperation here means that an interface between the end-to-end method and the network support mechanisms exists: a higher priority for a certain packet is requested explicitly).

3.3.1 Implicit cooperation

There are research efforts on how to use information on the flow structure (e.g. association of packets to frames) to allow a graceful degradation of the flow when losses for that flow cannot be avoided (network adaptation / filtering). We describe these mechanisms with the term intra-flow QoS enhancement (chapter 1, Table 1.1). For video traffic there have been several proposals (e.g. Frame-Induced Packet Discarding ([RRV93]), Transcoding, Transform Coefficient Filters), some of which also include an alignment with inter-flow QoS mechanisms ([WHD94, WZ98]). However, these application-level approaches typically suffer from adding significant (application layer) complexity to nodes interior to the network and contradict with network security constraints. Furthermore they are generally very dependent on the

supported payload types, which are subject to change over time. Due to the low per-flow bandwidth for real-time voice, most of the approaches mentioned above do not easily apply. A voice stream typically cannot be filtered/transcoded further.

In a less application-specific, combined approach to loss recovery and control, the network enforces a certain loss probability as well as certain periodic patterns (e.g. alternating drop). The knowledge about the loss conditions (unconditional/conditional loss probability, loss patterns) can then be exploited by the end-to-end algorithm.

Koodli and Krishna ([KK97]) define an end-to-end "noticeable loss rate" metric (cf. also section 4.1.6), where the application specifies an acceptable task loss of a scheduler over a time window. Then this desired metric is translated to a per-subtask control algorithm at a node. Seal and Singh ([SS96]) present the enforcement of "loss profiles" at the transport layer of the source host or an intermediate node ([BS96]). "Loss profiles" are pre-defined discarding functions ("clustered"/"random" loss) operating over certain time windows on logical data segments designated by the application.

Other examples for implicit cooperation are mechanisms where a flow uses some form of inter-flow QoS (per-flow reservation / per-packet prioritization) however not for the entire necessary flow bandwidth. Then FEC/concealment mechanisms may adapt to and exploit the modified loss conditions on an end-to-end basis. Few related work is available in the area: Vega ([Gar96], chapter 9.3) briefly analyzed using end-to-end bandwidth adaptation[9] and FEC together with either FIFO or FQ (Fair Queuing) and showed improvements in the behavior of their control algorithm with FQ. Shacham and McKenney ([SM90]) presented an approach using generic FEC (section 3.1.2.2, p. 40) and buffer management within the network. They were able to show the performance improvement due to this dual approach. However no details about the amount of cooperation and needed protocol support between FEC and buffer management are given: it is unclear how the ADU (block) association of (redundancy) packets is derived at the node implementing the buffer management algorithm.

In this thesis we will develop end-to-end methods with minimal cooperation which are suitable for stand-alone operation (chapter 5) however particularly benefit by some intra-flow QoS network support (chapter 6).

3.3.2 Explicit cooperation

Besides the possibility of dividing the signal for transmission on a per-sample basis as described in paragraph 3.1.2.1 numerous other possibilities to partition the signal for transmission exist. In contrast to the scheme just mentioned the following approaches generate packets of variable importance for the recovery of the speech signal, thus resulting into an adaptation to the signal, presuming the presence of a network service which enforces explicitly the "importance" given by the application.

[9]Note that the bandwidth adaptation is done only in a discrete fashion by switching between codecs, cf. section 3.1.2.3, p. 42.

Thus, contrary to the methods in the previous section, these schemes are voice-specific, they are also referred to as *embedded coding*. The necessary operations can be done before, within or after the encoder (Fig. 3.1, p.32).

The integration of the encoding process and the network transmission constitutes a tradeoff: on one hand the signal is transmitted with a much higher resilience to packet loss, on the other hand the complexity of implementation and deployment is higher (especially because an interface to the network QoS mechanisms is necessary). Chapter 7 redefines one end-to-end-only approach developed in this thesis (section 5.2.4) to be a combined one with explicit cooperation (section 7.2).

3.3.2.1 Pre-encoder payload analysis

Embedded coding[10] / packetization ([Jay93, GV93, ST89]) means transmitting the bits representing a sample in different packets. Thus packets of different importance are generated corresponding to the bit significances (LSB/MSB: least/most significant bit).

Other known mechanisms are working at the signal (PCM) level in conjunction with other methods. In class-oriented coding/recovery ([DPF89]) the signal content is classified into broad categories ("voiced", "unvoiced", "silence/background noise"). Then every category is encoded separately (e.g. using different quantization resolutions). Also, every class can be protected differently within the network.

3.3.2.2 Encoder-based and post-encoder payload analysis

Here the signal is partitioned in the code domain and thus can exploit useful properties of the employed coding scheme. Then the different parts of the encoded signal are transmitted using packets of different importance. Most of the time these schemes integrate the packetization with the generation and partition of the codewords, which is most efficient however requires modifications to the encoder itself.

Backward-adaptive encodings For *(A)DPCM codecs* a modified DPCM (Fig. 2.2, p. 16) encoding process is designed ([LBL92, ST89]), which is shown in Figure 3.18: a signal with less resolution is used for the prediction. Some LSBs are deleted before being fed into the predictor feedback loop, symbolized in Figure 3.18 by Q_{MSB} (quantization with less resolution). Also, these LSBs are packetized and transmitted separately (with a lower priority than the packets containing the MSBs). When an LSB packet is lost, the quality of the prediction of the ADPCM decoder is not affected. Thus the huge impact of the mis-synchronization of the encoder and decoder which has been described in section 2.2.1.1, p. 23, can be avoided. This advantage in terms of robustness against packet loss is traded against a permanently lower prediction quality due to the deleted bits in the predictor feedback loop.

Yong ([Yon92]) has presented two methods on how to treat *frame-based codecs* in this context, in particular CELP-based ones: In CELP, the LP and pitch param-

[10]The term "coding" though widely used is somewhat misleading here, as only a different packetization of the PCM samples is done.

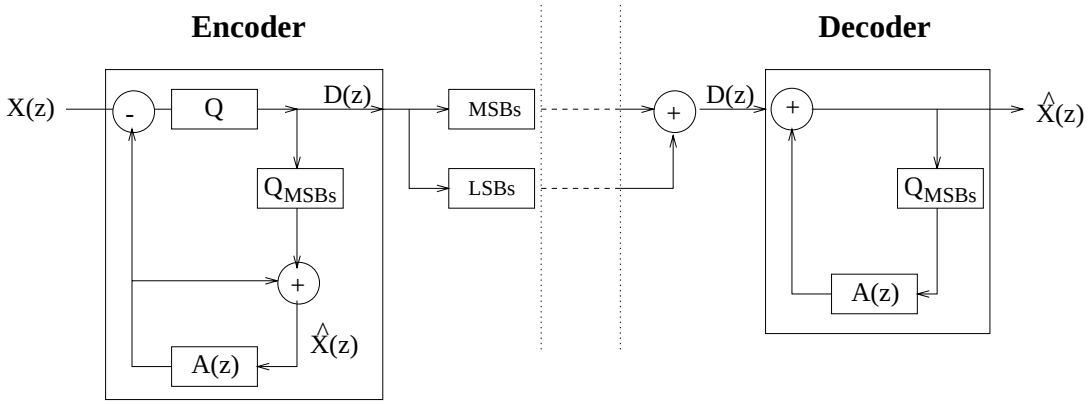


Figure 3.18: Embedded DPCM system

eters are considered critical. Thus they are packetized separately into high priority packets, whereas the excitation vectors are put into packets with low priority ("splitting"). Note that this scheme can be categorized as "post-encoder payload analysis". Another option which actually modifies the encoding scheme itself ("encoder-based payload analysis") is to build a two-stage encoding process where each stage contains its own full CELP encoding, however the second stage encodes only the residual signal of the first stage. The overall bit-rate of $16kbit/s$ is shared between the stages ($8kbit/s$ for each stage, [Yon92]).

The disadvantage of the splitting scheme is that when a lot of low priority packets are lost, nearly no excitation vector is received and the output of the synthesis filter will converge to zero. In the two-stage encoding however at least the signal at a lower bit-rate/quality can be reproduced. The tradeoff here is in the higher computational and implementation complexity of the two-stage scheme.

Multi-resolution encodings The bit splitting solution for PCM explained in section 3.3.2.1 can already be considered to be a multi-resolution encoding. In transformation / sub-band encodings (e.g using wavelets [Ise96]) the coding process generates similar "layers" of coefficients which represent the signal in the frequency domain over a certain time interval. As the low frequency parts are more important to user perception, again packets of different importance can be generated. Then these groups of packets can be transmitted with different redundancy protections (section 3.1.2.2) or use different priorities in the network. Note that this approach is closely related to receiver adaptation (adaptation in terms of the number of layers which can be received: section 3.1.2.3). A major disadvantage of all approaches using layering is the necessary re-synchronization of several packets representing signal content of the same time interval.

Chapter 4

Evaluation Models and Metrics

To assess the impact of packet loss on voice traffic, usually some "mean loss rate" is used. As an example, Fig. 4.1 shows mean loss rates $p_m(s)$ for a voice stream versus its sequence number s averaged over a sliding window of five and 100 packets respectively ($p_5(s)$, $p_{100}(s)$). It can be seen that the distribution of loss rates (and

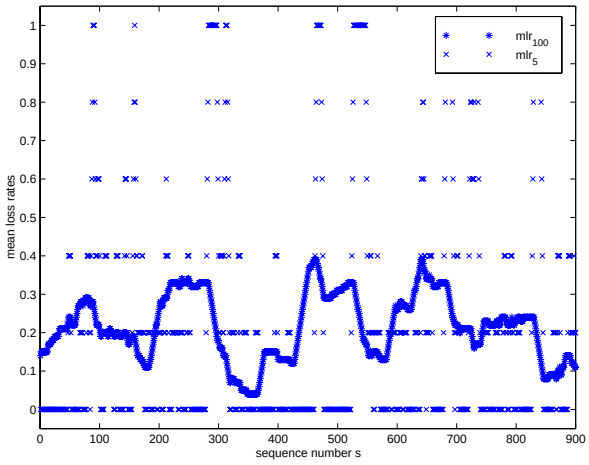


Figure 4.1: Mean loss rates for a voice stream averaged over 5 and 100 packets

thus the perceptual impact) over a small window size ($p_5(s)$) varies strongly. At the same time, the mean loss rate evaluated over the larger window size ($p_{100}(s)$) varies within a much smaller interval. However the length of phonemes (section 2.1.1) as an important unit of speech perception is in the range of the time interval corresponding to the smaller window of five packets. Therefore a mean loss rate which is averaging over packet loss events which are too distant from each other can only coarsely quantify the loss impact (e.g. only distortions perceived as "drop outs" may be detected). Our definition of intra-flow QoS (Table 1.1) thus covers only a "short term" range, i.e. roughly an interval of less than a talk spurt length (cf. section 2.2.2).

In section 4.1 we provide an intra-flow characterization of the packet loss pro-

cess by developing a model based on loss run-lengths. Section 4.2 looks at how application-level QoS and thus user perception can be described. We introduce conventional objective and subjective quality metrics as well as novel perceptual metrics for objective speech quality assessment. Section 4.3 then describes the relationship between the introduced packet-level and speech quality metrics. In section 4.4 we describe the employed traffic model and topology for the simulation of the behavior of individual network nodes.

4.1 Packet-level loss models and metrics

In section 2.2.1.1, p. 24, we have introduced the Gilbert model to describe the packet loss process. There we have identified the need for a more detailed characterization of the loss process, building on existing metrics. As throughout this thesis our main focus of interest is intra-flow QoS, i.e. a description of the loss process for the packets of a flow with regard to each other, our goal is to build a framework model with the following properties:

- expression of unconditional and conditional[1] loss metrics (including e.g. those of the Gilbert model);
- adjustable complexity dependent on specific application/network requirements.
- using only one quantity which is easily traceable as a basic metric from which all model parameters/metrics are computed.
- low number of parameters/states resulting in simple implementation and the opportunity of on-line parameterization.
- applicability to mechanisms which influence loss correlation (chapter 6).

In [SC00a] we have introduced our model heuristically. Jiang and Schulzrinne then have shown in [JS00b] how our model can be derived from a general Markov model with simplifying assumptions. We adopt this approach in section 4.1.1 and present further evidence that the simplification, while reducing the model complexity, does not impair the significance of the derived metrics. Then, we introduce the model with unlimited, limited and only two states respectively, where we show that the two state model is equivalent to the Gilbert model. Section 4.1.5 presents a similar (but separate) model to capture the distribution of no-loss (good-) run-lengths. Sections 4.1.6 and 4.1.7 present metrics which are composed of loss- and no-loss metrics and discuss efficient computation. Finally, in section 4.1.8, we show the applicability of the developed metrics to actual Internet loss traces.

[1] Note the respective correspondence between unconditional / conditional and long-term / short-term metrics.

State	Probability of being in the state	Probability of *l(s)=0*	Probability of *l(s)=1*
000	0.8721	**0.9779**	0.0221
001	0.0208	0.6112	0.3888
010	0.0142	**0.8819**	0.1181
011	0.0102	0.2710	0.7290
100	0.0208	**0.9278**	0.0722
101	0.0036	0.4198	0.5802
110	0.0102	**0.8109**	0.1891
111	0.0481	0.1539	0.8461

Table 4.1: State and transition probabilities computed for an Internet trace using a general Markov model (third order) by Yajnik et. al. [YKT95]

4.1.1 General Markov model

The *loss indicator function* for a certain flow (see the definition in section 2.2.3) at a certain node dependent on the packet sequence number s is:

$$l(s) = \begin{cases} 0: & \text{packet } s \text{ is not lost} \\ 1: & \text{packet } s \text{ is lost} \end{cases} \tag{4.1}$$

Considering the periodic packetization of a voice stream, the loss indicator as a function of the packet sequence fully captures the loss seen over time. However, it should be noted that when variable bitrate sources or silence detection (section 2.2.2) are employed that the loss indicator function only approximates the loss as a function of time (cf. section 1.1).

A general Markov model which describes the loss process using the loss indicator function is defined as follows ([YKT95, JS00b]):

Let $P(\, l(s) \mid l(s-m), \ldots, l(s-2), l(s-1)\,)$ be the state transition probability of a general Markov model of order m. All combinations for the values (0 and 1) of the sequence $l(s-m), \ldots, l(s-2), l(s-1)$ appear in the state space. As an example $P(\, l(s) = 1 \mid l(s-2), l(s-1) = 01\,)$ gives the state transition probability when the current packet s is lost, the previous packet $s-1$ has also been lost and packet $s-2$ has not been lost. The number of states of the model is 2^m. Two state transitions can take place from any of the states ($l(s) = 0$ or $l(s) = 1$ where s is the sequence number of the next packet). Thus the number of parameters which have to be computed is 2^{m+1}. Even for relatively small m this number of parameters is difficult to be evaluated and compared. Also, this approach does not seem feasible for on-line computation (e.g. for network-aware applications which need more information about the current loss process than just one or two parameters).

Table 4.1 shows some values for the state and transition probabilities for a general Markov model of third order measured in the Internet by Yajnik et. al. ([YKT95]). It is interesting to note that for all states with $l(s-1) = 0$ the probability for the next packet not to be lost ($l(s) = 0$) is generally very high (> 0.8, in bold typeface)

whereas when $l(s-1)=1$ the state transition probabilities to that event cover the range of 0.15 to 0.61. That means that past no-loss events do not affect the loss process as much as past loss events. Intuitively this seems to make sense, because a successfully arriving packet can be seen as an indicator for congestion relief. Andren et. al. ([AHV98]) as well as Yajnik et. al. ([YMKT98]) both confirmed this by measuring the cross correlation of the loss- and no-loss-run-lengths. They came to the result that such correlation is very weak. This implies that patterns of short loss bursts interspersed by short periods of successful packet arrivals occur rarely (note in this context that in Table 4.1 the pattern 101 has by far the lowest state probability).

Thus, in the following we design a model which only considers the past *loss* events for the state transition probability. While only few information is lost, the number of states of the model can be reduced from 2^m to $m+1$. This means that we only consider the state transition probability $P(\ l(s) \mid l(s-k),\ldots,l(s-1)\ )$ with $l(s-k+i)=1\ \forall\ i\in[0,k-1]$, however with a variable parameter k $(0<k\leq m)$. Note that with this definition we can relate the model to complete run-lengths ($l(s)=0$ and $l(s-k-1)=0$):

Using equation 4.1, we define a *loss run length* k for a sequence of k consecutively lost packets detected at s_j $(s_j>k>0)$ with $l(s_j-k-1)=0, l(s_j)=0$ and $l(s_j-k+i)=1\ \forall\ i\in[0,k-1]$, j being the j-th "burst loss event". Note that the parameters of the model become independent of the sequence number s and can now rather be described by the occurrence o_k of a loss run length k.

4.1.2 Loss run-length model with unlimited state space

We define the random variable X as follows: $X=0$: "no packet lost", $X=k$: "*exactly*[2] k consecutive packets lost", and $X\geq k$: "*at least*[3] k consecutive packets lost". With this definition, we establish a loss run-length model (Fig. 4.2) with an unlimited (possibly infinite) number of states, which gives loss probabilities dependent on the burst length[4]. In the model, for every additional lost packet which adds to the length of a loss burst a state transition takes place. If a packet is successfully received, the state returns to $X=0$. Thus the state probability of the system for $k>0$ is $P(X\geq k)$. Given the case of a finite number of arrivals for a flow a, that experiences $d=\sum_{k=1}^{\infty} k o_k$ packet drops, we have the *relative frequency* $p_{L,k}=\frac{o_k}{a}$ for the occurrence of a loss burst of length k. Thus we can approximate the state probabilities of the model for $k>0$ by the cumulative loss rate $p_{L,cum}(k)=\sum_{n=k}^{\infty} p_{L,n}$ (Table 4.2). An approximation for the expectation of the random variable X can be computed as $p_L=\sum_{k=1}^{\infty} k p_{L,k}$ and identified with the "mean loss rate".

[2] "Exactly" means that the two packets immediately preceding and following the k lost packets are not lost with probability 1: $l(s)=0$ and $l(s-k-1)=0$.

[3] "At least" means that the packet immediately preceding the k lost packets is not lost with probability 1: $l(s-k-1)=0$.

[4] The basic model structure is similar to the one employed by Varma ([Var93]) and Hsu et al. ([HOK97]).

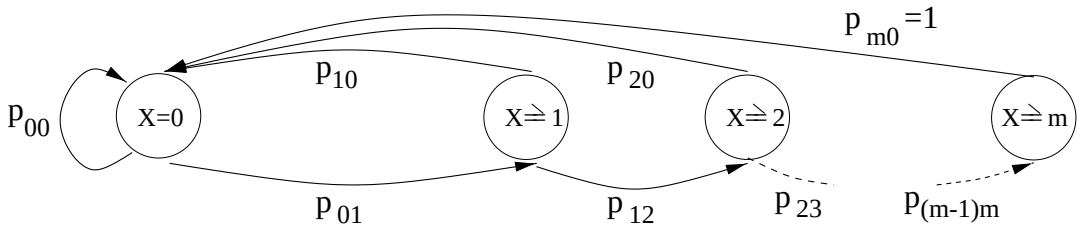


Figure 4.2: Loss run-length model with unlimited state space: $m \to \infty$ states

Loss run-length model (unlimited states)	a arrivals	$a \to \infty$
burst loss ($k > 0$)	$p_{L,k} = \frac{o_k}{a}$	$P(X = k)$
mean loss	$p_L = \sum_{k=1}^{\infty} k p_{L,k}$	$E[X]$
cumulative loss ($k > 0$)	$p_{L,cum}(k) = \sum_{n=k}^{\infty} p_{L,n}$	$P(X \geq k)$ (state prob.)
conditional loss ($k > 0$)	$p_{L,cond}(k-1,k) = \frac{p_{L,cum}(k)}{p_{L,cum}(k-1)} = \frac{\sum_{n=k}^{\infty} o_n}{\sum_{n=k-1}^{\infty} o_n}$	$P(X \geq k \mid X \geq k-1)$ (state transition prob. $p_{(k-1)(k)}$)
burst loss length ($k > 0$)	$g_k = \frac{o_k}{\sum_{n=1}^{\infty} o_n}$	$P(Y = k)$
mean burst loss length	$g = \frac{d}{\sum_{k=1}^{\infty} o_k} = \frac{\sum_{k=1}^{\infty} k o_k}{\sum_{k=1}^{\infty} o_k} = \sum_{k=1}^{\infty} k g_k$	$E[Y]$

Table 4.2: QoS metrics for the loss run-length model with unlimited state space: $m \to \infty$

The matrix of state transition probabilities for this model is given by

$$\begin{bmatrix} p_{00} & p_{01} & 0 & \cdots & 0 \\ p_{10} & 0 & p_{12} & & \vdots \\ p_{20} & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ p_{(n-2)0} & 0 & \cdots & 0 & p_{(n-2)(n-1)} \\ p_{(n-1)0} & 0 & 0 & \cdots & 0 \end{bmatrix}$$

The transition probabilities which can also be described as conditional loss probabilities can be computed easily as:

$$p_{(k-1)(k)} = P(X \geq k | X \geq k-1) = \frac{P(X \geq k \cap X \geq k-1)}{P(X \geq k-1)} = \frac{P(X \geq k)}{P(X \geq k-1)}$$

Again, if the burst loss occurrences o_k constitute a statistically relevant dataset, we can compute approximations for the conditional loss probabilities as given in Table 4.2.

Additionally, we also define a random variable Y which describes the distribution of burst loss lengths with respect to the burst loss events j (and not to packet events like in the definition of X). $E[Y]$ then is the expected mean burst loss length (loss gap). Table 4.2 shows the performance metrics of the loss run-length model for a finite number of arrivals a using the loss run length occurrences o_k, as well as the relation to the transition/state probabilities of the model ($a \to \infty$) with the random variables X and Y. The cumulative loss rate for $k = 0$ is defined as the "no loss" case (corresponding to $P(X = 0)$):

$$p_{L,cum}(k=0) = 1 - \sum_{k=1}^{\infty} p_{L,cum}(k) = 1 - \sum_{k=1}^{\infty} \frac{\sum_{n=k}^{\infty} o_n}{a} = 1 - \sum_{k=1}^{\infty} \frac{k o_k}{a} = 1 - p_L \quad (4.2)$$

The relationship between the metrics conditioned on either packet events or burst loss events can be described as follows:

$$p_{L,k} = g_k \frac{\sum_{n=1}^{\infty} o_n}{a} = g_k \frac{\sum_{n=1}^{\infty} n o_n}{ga} = g_k \frac{p_L}{g}$$

When considering loss probabilities instead of the loss rates based on a finite number of arrivals this can be written as:

$$\frac{P(X=k)}{P(Y=k)} = \frac{E[X]}{E[Y]} \quad (4.3)$$

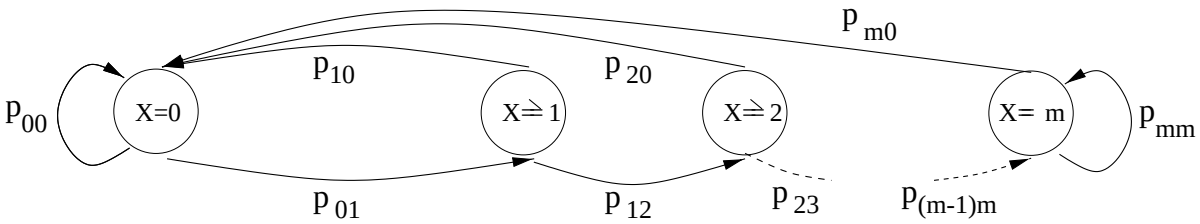


Figure 4.3: Loss run-length model with limited state space: $(m+1)$ states

4.1.3 Loss run-length model with limited state space

To assess the performance of a network with respect to real-time audio and video applications, a model with a limited number of states is sufficient. This is due to the fact that real-time audio and video applications have strict requirements and cannot use a network service with a significant number of "long" loss bursts. For these applications it is desirable to use only few model parameters, and to focus on key aspects of the loss process. In addition, memory and computational capabilities of the system that performs modeling have to be taken into account (see also section 4.1.7).

For these reasons we derive from the basic model a loss run-length model with a finite number of states $(m+1)$. Fig. 4.3 shows the Markov chain for the model. Table 4.3 gives performance metrics similar to those in Table 4.2, however the state probability for the final state m, $P(X=m)$, and the probability for a transition from state m to state m are added[5]. For $0 < k < m$, $X = k$ represents as before "*exactly* k consecutive packets lost". Due to the limited memory of the system, the last state $X = m$ is just defined as "m consecutive packets lost". Thus $P(X = m)$ can be seen as a measure for the "loss over a window of size m" (independently of actually larger loss run lengths).

Figure 4.4 shows the base metrics used to compute the loss run-length based metrics. In Figure 4.4 (a), each point indicates whether there was a loss (1) or not (0), representing the loss indicator function. Figure 4.4 (b) shows the loss run lengths. Figure 4.4 also shows some of the state transitions when a given loss trace is applied to a model of either $m = 2$ or $m \geq 4$. With $m = 2$ for a loss burst of length $k = 4$, the system is three times $(k - m + 1$, see Fig. 4.4 (c)) in state 2, and thus two $(k - m)$ transitions $m \rightarrow m$ occur. This leads to the computation of $p_{L,m}$ and $p_{L,cond}(m)$ (as approximations for $P(X = m)$ and $P(X = m|X = m)$ respectively) as given in Table 4.3.

Interestingly, Miyata et al. ([MFO98]) propose precisely $p_{L,m}$ as a performance measure for FEC-based audio applications. This "sliding window" of m consecutively lost packets allows to reflect specific applications' constraints, e.g. m can be

[5]The burst loss length metrics for $k = m$ are computed similarly and are therefore not shown.

Loss run-length model ($m+1$ states)	a arrivals	$a \to \infty$
burst loss $(0 < k < m)$	$p_{L,k} = \frac{o_k}{a}$	$P(X = k)$
burst loss $(k = m)$ loss over window m	$p_{L,m} = \sum_{n=m}^{\infty} \frac{(n-m+1)o_n}{a}$	$P(X = m)$ (state probability)
mean loss	$p_L = \sum_{k=1}^{\infty} \frac{k o_k}{a}$	$E[X]$
cumulative loss $(0 < k \leq m)$	$p_{L,cum}(k) = \sum_{n=k}^{\infty} \frac{o_n}{a}$	$P(X \geq k)$ (state probability)
conditional loss $(0 < k \leq m)$	$p_{L,cond}(k-1,k) = \frac{p_{L,cum}(k)}{p_{L,cum}(k-1)} = \frac{\sum_{n=k}^{\infty} o_n}{\sum_{n=k-1}^{\infty} o_n}$	$P(X \geq k \mid X \geq k-1)$ (state transition prob. $p_{(k-1)(k)}$)
conditional loss $(k = m)$	$p_{L,cond}(m,m) = \frac{\sum_{n=m}^{\infty}(n-m)o_n}{\sum_{n=m}^{\infty} n o_n}$	$P(X = m \mid X = m)$ (state transition prob. p_{mm})

Table 4.3: QoS metrics for loss run-length model with limited state space: $(m+1)$ states

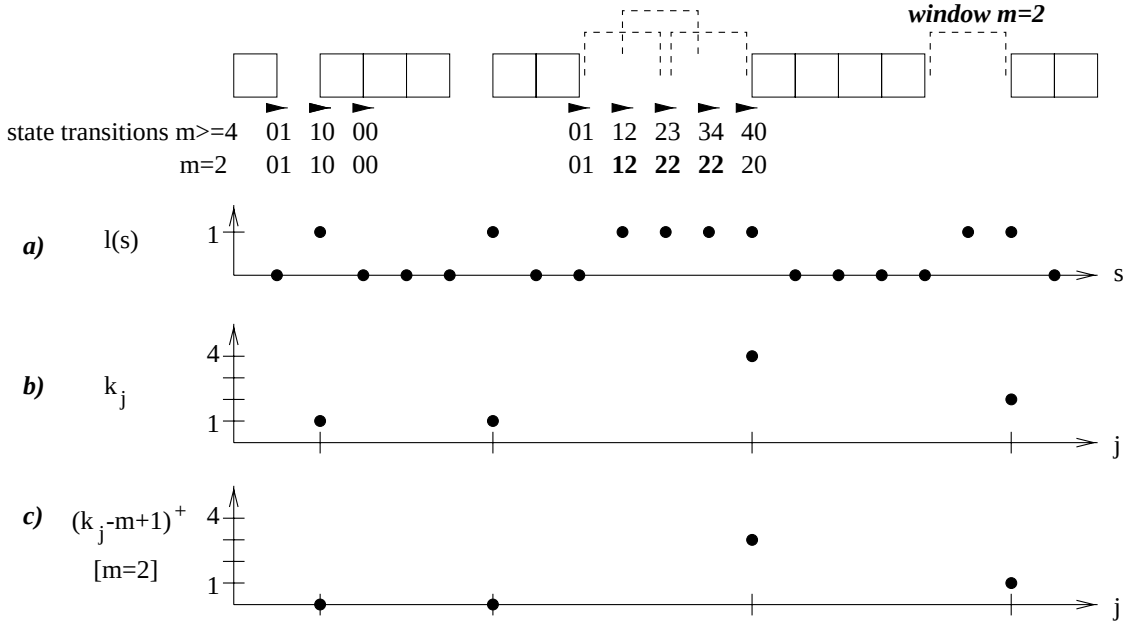


Figure 4.4: Basic loss metrics

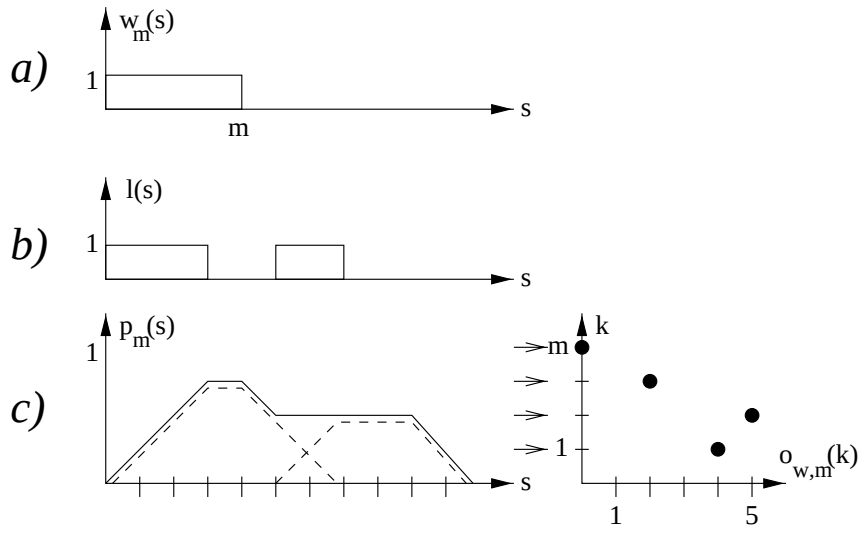


Figure 4.5: $p_m(s)$: mean loss rate over a sliding window of length m

set to the lowest number of consecutively lost packets for which a complete audio "dropout" is perceived by a user. Then, larger loss bursts do not have a higher impact and thus do not need to be taken into account with their exact size. We extend the above approach by looking at the occurrence of a certain number of packets lost within the window of length m. This allows e.g. to assess how effective FEC protection applied to groups of packets would be without keeping track of the actual Application Data Unit (ADU) association of the individual packets. In section 1 we introduced the mean loss rate over a sliding window of length m, $p_m(s)$, which can be formally defined as the convolution of the analysis window with the loss indicator function (Fig. 4.5):

$$p_m(s) = \frac{l(s) * w_m(s)}{m} = \frac{\sum_{\nu=0}^{a} l(\nu) w_m(s-\nu)}{m}$$

The tradeoff following from the above formula can be described as follows: computing the actual histogram of $p_m(s)$ (solid line in Fig. 4.5 c) captures the accurate sequential relation of the loss bursts, however makes the measure still depend on s (this is the approach taken in [BSUG98]). When using only the sum of histograms calculated separately over each individual loss burst (dashed line Fig. 4.5 c) some information is lost, however only the tracking of loss bursts k is needed. To describe the latter approach we use $o_{w,m}(k)$ which describes the occurrence of k *consecutive* packets lost within the window of length m

$$o_{w,m}(k) = \begin{cases} (m-k+1)o_k + 2\sum_{n=k+1}^{\infty} o_n: & 0 < k < m \\ \sum_{n=m}^{\infty} (n-m+1)o_n: & k = m \end{cases}$$

Summing over the weighted $o_{w,m}(k)$ we get in fact the overall mean loss rate:

$$\begin{aligned}
&\sum_{k=1}^{m} \frac{k o_{w,m}(k)}{ma} = \\
= \;& \frac{1}{ma}\left[\sum_{k=1}^{m-1}\left(k(m-k+1)o_k + 2k\sum_{n=k+1}^{\infty} o_n\right) + m\sum_{n=m}^{\infty}(n-m+1)o_n\right] \\
= \;& \frac{1}{ma}\left[\sum_{k=1}^{m-1} k(m-k+1)o_k + 2\left(\sum_{k=1}^{m-1} k\sum_{n=k+1}^{m-1} o_n + \sum_{k=1}^{m-1} k\sum_{n=m}^{\infty} o_n\right)\right. \\
& \left. + m\sum_{n=m}^{\infty}(n-m+1)o_n\right] \\
= \;& \frac{1}{ma}\left[\sum_{k=1}^{m-1} kmo_k - \sum_{k=1}^{m-1} k(k-1)o_k + 2\left(\sum_{k=1}^{m-1}\frac{k(k-1)}{2}o_k + \frac{m(m-1)}{2}\sum_{n=m}^{\infty} o_n\right)\right. \\
& \left. + \sum_{n=m}^{\infty} nmo_n - m(m-1)\sum_{n=m}^{\infty} o_n\right] \\
= \;& \frac{m\sum_{k=1}^{\infty} ko_k}{ma} = p_L
\end{aligned}$$

Similar window-based metrics were proposed also in ([OMF98, KR00, NKT94]).

The run-length-based model with a finite state space implies a geometric distribution for residing in the last state $X = m$. When we consider e.g. an estimation model order of $\hat{m} = 2$ we can easily derive estimates for higher order model representations. As an example we consider the probability of a burst loss length of k packets:

$$\hat{P}(Y = k) = \begin{cases} P(Y = k) = 1 - p_{12}: & 0 < k < \hat{m} \\ p_{12}\, p_{22}^{k-2}\,(1 - p_{22}): & \hat{m} \le k < m \end{cases} \tag{4.4}$$

Note that here Y represents the random variable used in the model of order m. For $k < \hat{m}$ the formula yields the exact value.

4.1.4 Gilbert model

For the special case of a system with a memory of only the previous packet ($m = 1$), we can use the runlength distribution for a simple computation of the parameters of the commonly-used Gilbert model (Fig. 4.6) to characterize the loss process (X being the associated random variable with $X = 0$: "no packet lost", $X = 1$ "*a* packet lost"). Then the "loss over window m" is equal to the mean loss or **unconditional loss probability *ulp***, and only one **conditional loss probability *clp*** (transition $1 \to 1$) is defined.

The matrix of transition probabilities of the Gilbert model is:

$$\begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}$$

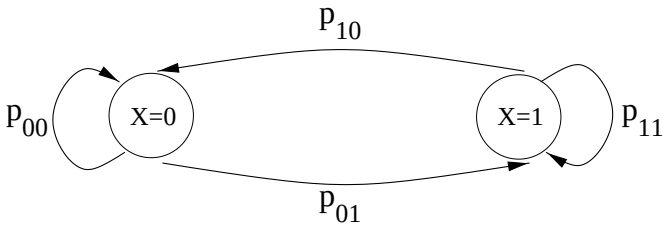


Figure 4.6: Loss run-length model with two states (Gilbert model)

The Gilbert model implies a geometric distribution for residing in state $X = 1$. For the probability of a burst loss length of k packets we thus have (using the ulp/clp notation of table 4.4) the following estimate for a higher order model representation (note that in the following X and Y represent the random variables used in the higher-order models):

$$\hat{P}(Y = k) = clp^{k-1}(1 - clp),\ 0 < k < m. \tag{4.5}$$

Based on equation 4.5, we can compute the mean burst loss length $E[Y]$ as:

$$E[Y] = \sum_{k=0}^{\infty} k clp^{k-1}(1 - clp) = \frac{1}{1 - clp}$$

Note that $E[Y]$ is computed based on the clp only, i.e. that the value of the mean burst loss length is dependent only on the loss behavior of two consecutive packets. The run-length based metrics allow us to easily confirm this result by Ferrandiz and Lazar ([FL90]) using the result of Table 4.2:

$$g = \frac{d}{\sum_{k=1}^{\infty} o_k} = \frac{\sum_{k=1}^{\infty} k o_k}{\sum_{k=1}^{\infty} k o_k - \sum_{k=1}^{\infty}(k-1)o_k} = \frac{1}{1 - p_{L,cond}}$$

Finally, using equations 4.3 and 4.5 we can derive a Gilbert model-based estimate for the probability of "exactly k consecutive packet lost":

$$\hat{P}(X = k) = \hat{P}(Y = k)\frac{ulp}{\frac{1}{1-clp}} = ulp\ clp^{k-1}(1 - clp)^2,\ 0 < k < m. \tag{4.6}$$

The values for $\hat{P}(X = k)$ and $\hat{P}(Y = k)$ can be compared to the actual values for the higher order loss run-length models to see how well the actual loss process is approximated by the simple two state model. Table 4.4 shows a summary of the performance metrics for the Gilbert model. Note that p_{01} can be computed irrespectively of the model order from equations 2.4 and 2.5 as

$$p_{01} = \frac{ulp(1 - clp)}{1 - ulp} \tag{4.7}$$

Gilbert	a arrivals	$a \to \infty$
burst loss ($k = 1$) loss over window 1	$p_L = \sum_{k=1}^{\infty} \frac{k o_k}{a}$ mean loss rate	$P(X = 1)$ **unconditional loss prob. *ulp***
conditional loss ($k = 1$)	$p_{L,cond}(1,1) = = \frac{\sum_{n=1}^{\infty}(n-1)o_n}{d}$	$P(X = 1 \mid X = 1)$ **conditional loss prob. *clp***
mean burst loss length	$g = \frac{d}{\sum_{k=1}^{\infty} o_k} = \frac{1}{1 - p_{L,cond}}$	$E[Y] = \frac{1}{1 - P(X = 1 \mid X = 1)}$

Table 4.4: QoS metrics for the loss run-length model with two states (Gilbert model)

Using Eq. 4.2 this corresponds to

$$p_{L,cond}(0,1) = \frac{p_{L,cum}(1)}{p_{L,cum}(0)} = \frac{\sum_{k=1}^{\infty} \frac{o_k}{a}}{1 - p_L}$$

4.1.5 No-loss run-length model with limited state space

User perception is not only affected by the length of burst losses ($k \in [1, m]$), but also by the length of the intervals between consecutive losses. In section 4.1.1 we have referenced related work on the weak cross correlation of the loss- and no-loss-run-lengths. Therefore it is reasonable to construct a separate but similar model to that which has been introduced in section 4.1.2.

We define a *no-loss run-length* (or good run-length) K detected at s_J ($s_J > K > 0$) with $l(s_J - K - 1) = 1, l(s_J) = 1$ and $l(s_J - K + i) = 0 \; \forall \; i \in [0, K-1]$, J being the J-th "no-loss burst event". As in paragraph 4.1.3, we limit K to an interval $[1, M]$ dependent on the application. For audio, M could e.g. be set to the lowest value for which consecutive loss events are perceived by the user as being separate rather than a single distortion of the signal. The occurrence of a loss distance K is given by o_K.

By defining a random variable X' as: $X' = 0$ if a packet was lost, $X' \geq k$ if at least k packets have *not* been lost, we can derive the same state model as the loss run-length model with finite state space for the "no-loss" case. Once m consecutive packets have been served (meaning not lost), the following packet arrivals (state transition: $m \to m$) are not taken into account in terms of the distance to the previously lost packet. Similarly to Tables 4.2-4.4 we can define model parameters and QoS metrics for the no-loss run-length model. Additionally, we also have a random variable Y' which describes the distribution of no-loss lengths with respect

to the no-loss events J. Of particular interest here is the relative frequency of a no-loss length K: $G_K = \frac{o_K}{\sum_{N=1}^{\infty} o_N}$ ($P(Y' = K)$ for $a \to \infty$).

4.1.6 Composite metrics

Obviously, both no-loss and loss models of any order can be combined to form a single model. Additionally, it is possible to define metrics based on both no-loss and loss events. An example is a measure which already exists in the literature ([KK98, KR00]) called the *noticeable loss rate (NLR)*. NLR defines a *loss distance constraint* (which is the no-loss runlength model order M) above which losses are excluded from the measure (are said to be not "noticeable"). Since the loss distance constraint must be at least one, all the losses in a loss run-length (except the first dependent on the distance to the previous loss) are said to be noticeable. Thus, using the previously introduced variables the NLR can be defined as follows:

$$NLR_M = \frac{d - \sum_{K=1}^{M-1} o_K}{d} = 1 - \frac{\sum_{K=1}^{M-1} o_K}{\sum_{k=1}^{\infty} k o_k}$$

4.1.7 Parameter computation

In this section we have demonstrated how to use loss and no-loss run-lengths to compute models ranging from two states (Gilbert model) over $m + 1$ states to a potentially infinite state space. However, our formulas used the assumption that all (no-)loss burst lengths up to potentially infinite length are tracked. In a real system however, we clearly need to limit the maximum tracked burst length as a tradeoff between needed model complexity to assess the network performance with regard to specific applications, and memory or computational limitations.

Therefore we can limit the tracing of run-lengths up to a length γ, with $\gamma \geq m$. Typically γ will be set according to the highest model order required ($\gamma = m$). This results in $p_L = \sum_{k=1}^{\gamma} \frac{k o_k}{a} + \frac{d_\gamma}{a}$, where d_γ are the packet drops which occur in bursts with higher lengths than γ. $p_{L,cum}(k) = \sum_{n=k}^{\infty} \frac{o_n}{a} = \sum_{n=k}^{\gamma} \frac{o_n}{a} + \frac{e_\gamma}{a}$, where e_γ are burst loss events with bursts larger than length γ. Thus essentially two additional counters are necessary, which keep track of $e_\gamma = \sum_{k=\gamma+1}^{\infty} o_k$ as well as $d_\gamma = \sum_{k=\gamma+1}^{\infty} k o_k$, rather than the individual o_k values.

4.1.8 Application of the metrics

We can identify the following two major applications of the introduced packet-level metrics:

- trace analysis (real/simulation traces): what model order is applicable/sufficient for a certain application in a certain network environment ?

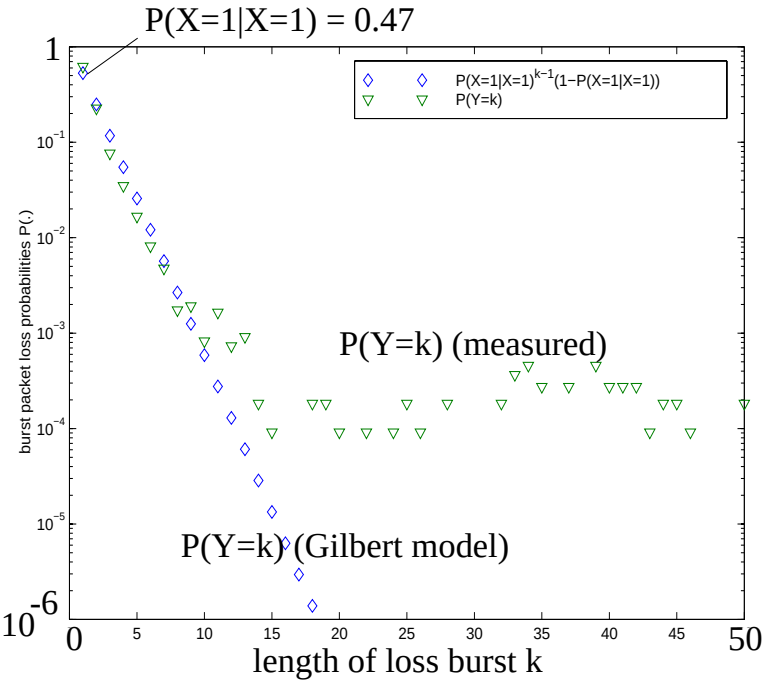


Figure 4.7: Example 1: Gilbert model fit

- trace synthesis: performance assessment of hop-by-hop loss control and end-to-end loss recovery

While we use the latter application extensively in the remaining chapters of this thesis, in this section we present some exemplary results of a measurement study. The traces were collected on three long paths ($\gtrapprox$ 15 hops) in both directions respectively between GMD Fokus, Germany, Nokia Research Center, Boston, Massachusetts and the ICSI in Berkeley, California between November 1999 and February 2000. We used periodic traffic sources comparable to voice sources without silence detection ($20ms$ voice (80 octets) per packet, 32 kbit/s). 100000 packets per trace (ca. 1/2 hour) were sent during various times per day. All examples shown here exhibit persistent network behavior (over several hours or even days). By visual inspection of a sliding window average $p_m(s)$ (as in section 1, however with a window size of 1000) we checked the traces for non-stationarity (abrupt changes in the smoothed loss rate, linear increase or decay seen over the entire trace) before applying our models with limited state spaces.

Figure 4.7 shows an exemplary measurement with $P(X = 1) = 0.0418$ and $P(X = 1|X = 1) = 0.4694$ giving values for the measured values of $P(Y = k)$ and those for the two-state Gilbert model $P(X = 1|X = 1)^{k-1}(1 - P(X = 1|X = 1))$. We see that the probability $P(Y = k)$ to lose exactly k consecutive packets in a burst loss event drops geometrically fast in an interval of approximately $[1, 10]$. In this case, a loss run-length model confirms that the loss process is approximated well by the Gilbert model $P(X = 1|X = 1)^{k-1}(1 - P(X = 1|X = 1))$. For larger bursts

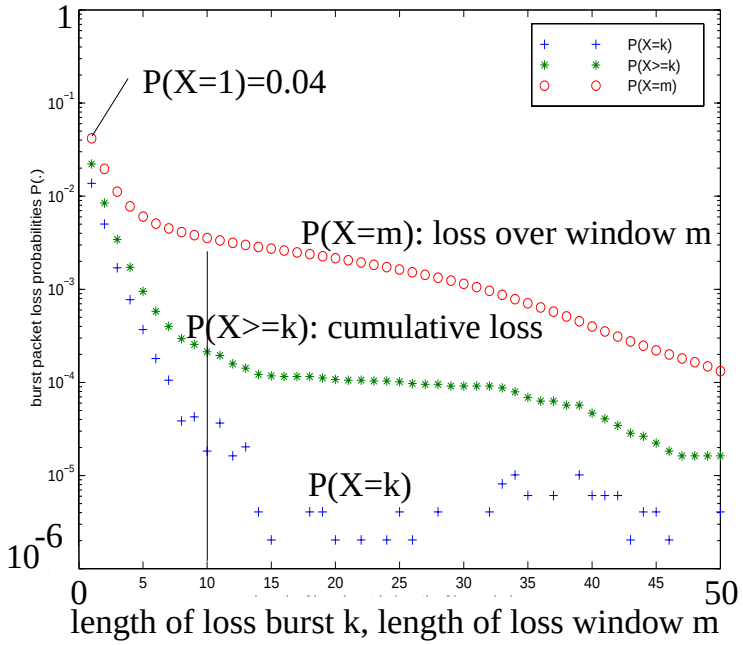


Figure 4.8: Example 1: state probabilities

($k > 10$), the loss burst probabilities are significantly larger than for the Gilbert model. Thus the loss process in that area is underestimated. However, the absolute values of the loss probabilities are several orders of magnitude smaller than for the singleton loss case ($k = 1$) and do not seem to follow a specific distribution (the given numbers are only based on few events). Therefore it is not necessary that this area is considered by a model.

Fig. 4.8 shows raw data $P(X = k)$ as well as the state probabilities for the model with limited ($k < m$) and unlimited state space. Additionally, the state probability $P(X = m)$ for the model with limited states is given. Assuming a model order of $m = 10$ the $P(X \geq k)$ values left of the vertical solid line are the state probabilities and the $P(X = 10)$ on top of this line is the final state probability. From the absolute values and the distance between $P(X \geq 9)$ and $P(X = 10)$ (less than one order of magnitude) we can conclude that some loss events with statistical significance for $k > 10$ exist, however no "outages" occur.

The conditional loss probabilities $P(X \geq k|X \geq k-1)$ (Fig. 4.9) increase with increasing loss burst length k, i.e. every loss increases the probability to loose the next packet as well. However their values are already very close to 1 for $k \gtrapprox 10$ and stay there (in the shown area). This means that (as mentioned above) only few burst loss events larger than 10 packets take place and thus models with a higher number of states do not give much additional information.

For the second example (Fig. 4.10) we see that the simple two-state model cannot adequately capture the loss process. We see three peaks in the distribution of the measured probability for a loss gap of length k (at $k \in [14, 20, 29]$). As the test

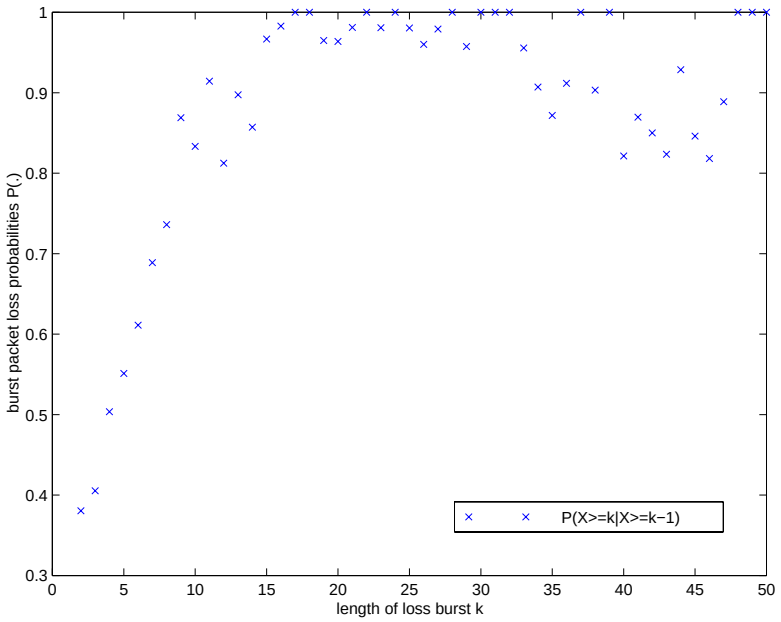


Figure 4.9: Example 1: conditional loss probabilities

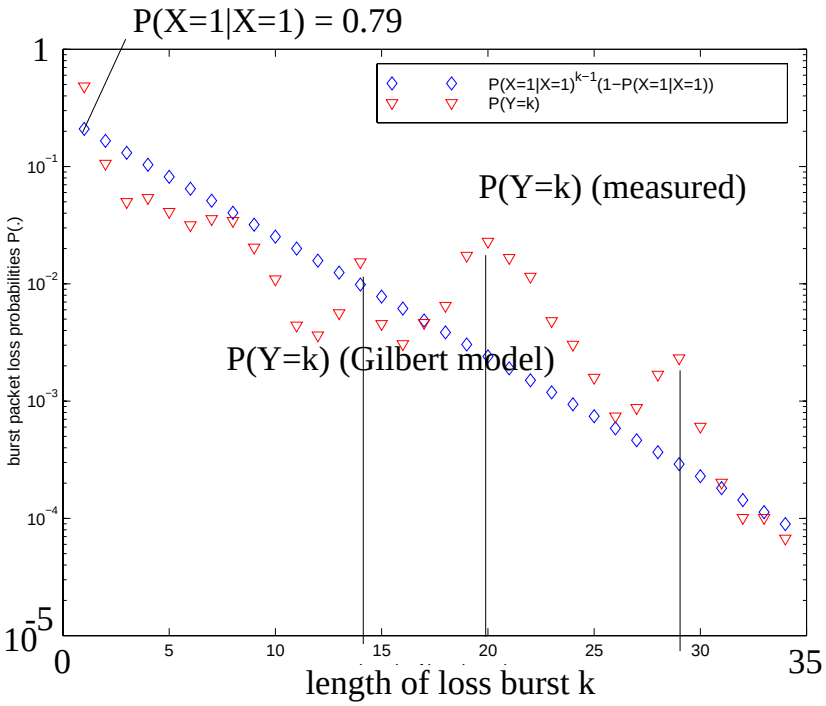


Figure 4.10: Example 2: Gilbert model fit

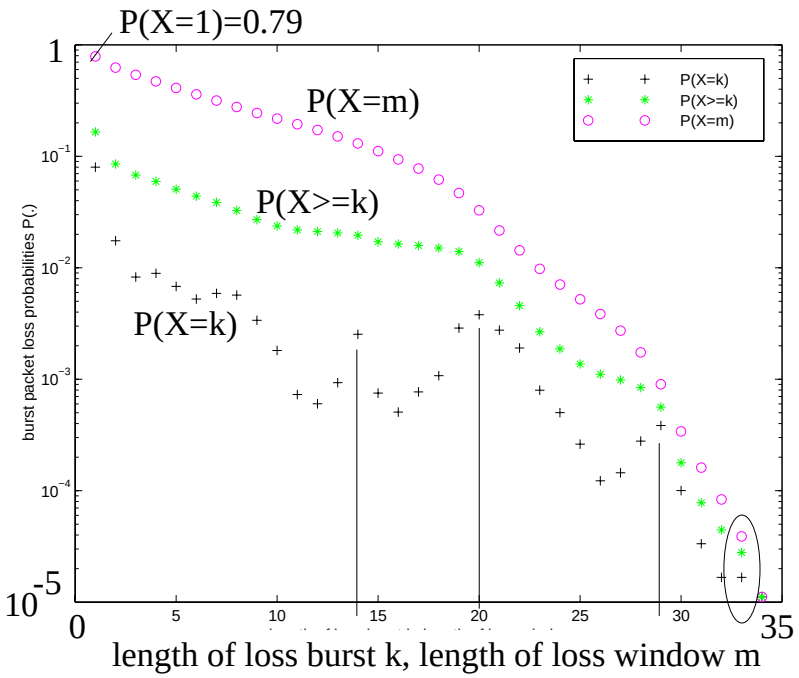


Figure 4.11: Example 2: state probabilities

traffic is periodic this amounts to frequent outages with a duration of 280, 400 and $580ms$ respectively. This could be explained by routers updating their forwarding tables, dropping packets until the update is finished. However a final conclusion would clearly require a closer look at sub-paths and individual network elements.

Fig. 4.11 shows that the overall extreme loss with $P(X = 1) = 0.79$ is not due to additional longer outages, but only to bursts with a length of $k \leq 35$ (note that the curves for $P(X \geq k)$ and $P(X = m)$ are very close to the raw data). This is also reflected in $P(X = 1)$ being equal to $P(X = 1|X = 1)$.

Example 3 (Figures 4.12 and 4.13) exhibits another "network pathology" however with different properties. Figure 4.12 shows the complete failure of the Gilbert model (the estimated conditional loss probability is close to one resulting in a virtually horizontal line for the estimated $P(Y = k)$ values). The probabilities for loss bursts larger than 80 packets (Fig. 4.13) reveal here the reason for this: in an interval of about $k \in [100, 125]$ a significant probability mass is concentrated. The distance between $P(Y = k)$ and $P(X = k)$ shows that only few events (but with a significant overall number of lost packets) contribute to this effect. Note that there is again a clear cutoff for the existing burst lengths (here however at $k = 135$).

The large loss bursts (which are perceived as "drop outs" and thus do not need to be taken into account in detail) completely bias the result of the Gilbert model with regard to the probability for short bursts. This emphasizes the advantage of the run-length-based model which as a general Markov model captures short bursts with the full available accuracy of the trace and aggregates the probabilities for longer bursts into the last state. That means that outages do not seriously affect

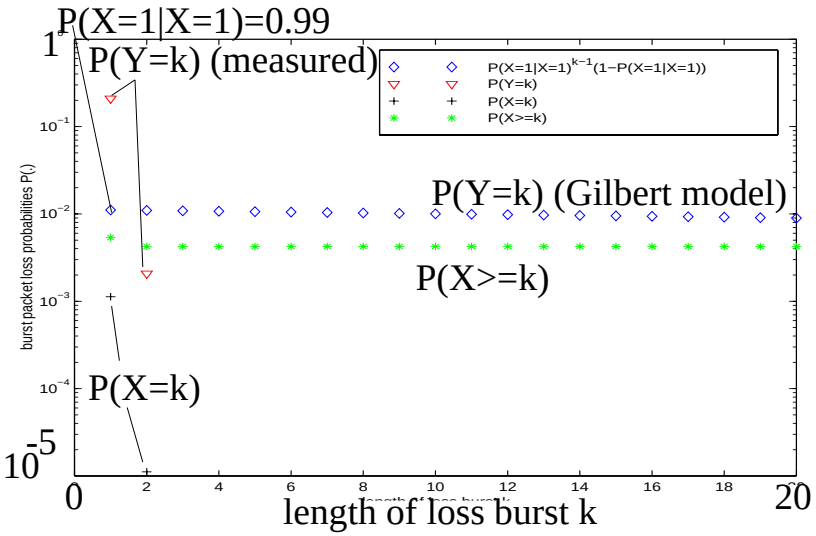


Figure 4.12: Example 3: Gilbert model fit

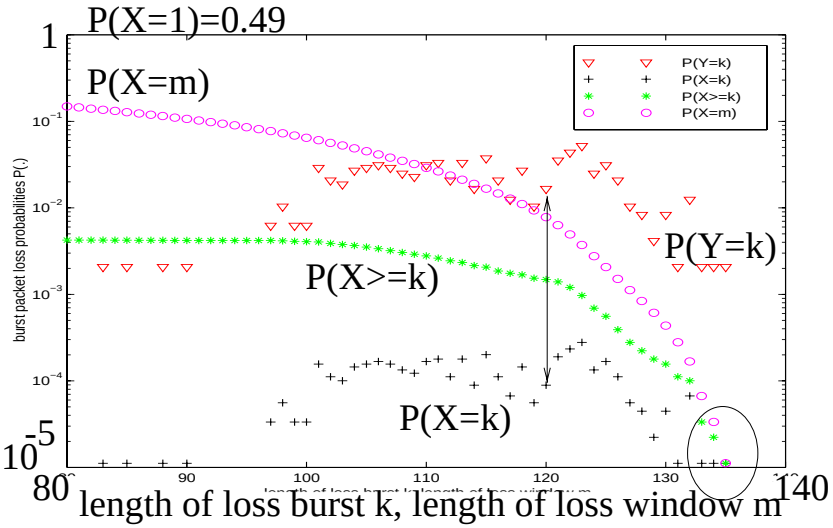


Figure 4.13: Example 3: state probabilities

the measurement result which is important with regard to mapping the result on user perception.

4.2 User-level speech quality metrics

In the previous chapter we have characterized the loss process of information at the packet level (see Fig. 2.6 for the architectural overview). However the packet loss measures must be translated to information loss at the application level (where for voice the sample or frame is the relevant unit). Due to the simple flow structure for voice this often amounts to just taking into account the number of samples/frames per packet[6] (therefore we do not formally define separate application-level metrics; see also section 3.1.1.1). The application-level loss of information must then be mapped to a measure of speech quality as perceived by a human being.

In general, there are two ways to measure the speech quality: subjective and objective measurements. In subjective measurements, listeners subject to a certain test environment listen to a set of speech signals and assess without being told about their origin. While subjective tests should be considered to be a very important tool to evaluate the performance of any speech-related system, they are time-consuming, expensive, error-prone, and only difficult to reproduce.

Objective measures map the measured application-level loss to a quality value comparing the speech signal with and without loss (typically this is done directly at the sample (PCM)-level). We present (and in later chapters apply) objective measures where perceptual models are employed for the mapping. Objective speech quality measurements avoid the disadvantages of subjective testing mentioned above.

4.2.1 Objective quality measurement

In objective speech quality measurement, speech quality is evaluated by measuring the "distortion" of the decoded speech signal compared to a "reference" speech signal, where "distortion" can be defined with mathematical expressions. Note that a reference signal can be reasonably defined in several ways: e.g. the original signal without any quality degradation or the signal distorted by the speech encoding/decoding process but without any packet loss on the transmission path.

4.2.1.1 Signal-to-Noise Ratio

The most common quality measure in the past has been the Signal-to-Noise ratio (SNR) of a sequence of samples of length l: ([Del93, JN84])

[6]It should however be noted that in particular for frame-based codecs, packet loss causes a de-synchronization of the encoder and the decoder. Thus a packet loss has not only an influence on the time interval represented by the lost packet but also on following packets. Furthermore, as we show in section 5.2.3, some groups of frames are more important to the perceptual quality than others. Hence the stream of voice frames exhibits a structure like e.g. a video stream with frames of different importance. However this structure is not fixed in advance and is not periodic like e.g. the group of pictures of an MPEG stream.

$$SNR = 10 \lg \frac{\sum_{n=0}^{l-1} x^2(n)}{\sum_{n=0}^{l-1} [x(n) - y(n)]^2} dB = 10 \lg \frac{\sum_{n=0}^{l-1} x^2(n)}{\sum_{n=0}^{l-1} e(n)^2} dB \tag{4.8}$$

$x(n)$: input signal of the system
$y(n)$: output signal

$e(n)$ is the error signal with n being the sample index relative to the start of the sequence. All errors in the time-domain signal are weighted equally however they might lead to completely different perceived distortion, because they affect the subjective speech attributes differently.

An objective quality measure which takes into account that the speech signal is non-stationary (i.e. that the speech energy may vary significantly between short time intervals) is the segment-based Signal-to-Noise Ratio. Signal-to-Noise Ratio values are calculated over a number of N short signal segments (e.g. with l being the packetization interval). The values SNR_i ($i \in [1, N]$) are then averaged to yield a single value as a quality representation for a longer speech segment:

$$SNR_{avg} = \sum_{i=1}^{N} \frac{SNR_i}{N} = \frac{1}{N} \sum_{i=1}^{N} 10 \lg \frac{\sum_{n=(i-1)l}^{il-1} y(n)^2}{\sum_{n=(i-1)l}^{il-1} e(n)^2} dB \tag{4.9}$$

Typically a lower and upper bound on the individual SNR_i value is set to avoid a bias of the result e.g. caused by input signal segments containing silence or output signal segments which are very close to the signal content (e.g. speech from correctly received packets adjacent to a loss gap which is windowed for loss concealment ("packet merging", [San95]) or speech from correctly received packets which has been distorted by error propagation from preceding lost packets[7]). Obviously the choice of the bounds is difficult and additionally the properties of the speech signal are still not well reflected.

An SNR-measure which has a higher correlation with subjective testing is the frequency-weighted segmental SNR ([Del93], p.595). For every segment, signal energies are computed separately within certain frequency bands which are then weighted according to results on the psycho-acoustical impact of distortions in the respective frequency band. Main disadvantage of this measure is the computational effort needed.

4.2.1.2 Perceptual objective metrics

Unlike the SNR methods, novel objective quality measures attempt to estimate the subjective quality as closely as possible by modeling the human auditory system

[7]Note that therefore the designation of SNR_{avg} as "SNR per missing packet" ([GLWW86, JC81]) only rarely applies.

in terms of hearing (section 2.1.1) and auditory judgment ([Vor99a]). Auditory judgment is done by comparing the reference signal to the test signal (decoded speech signal) with a distance measure, whereby both signals are perceptually transformed.

In our evaluation we use two objective quality measures[8]: the Enhanced Modified Bark Spectral Distortion (EMBSD, [YY99]) and the Measuring Normalizing Blocks (MNB, [Vor97, Vor99a, Vor99b]) described in detail in the Appendix II of the ITU-T Recommendation P.861 ([Uni98]). These two objective quality metrics are reported to have a very high correlation with subjective tests ([Vor99b, YBY98]). With these measures it is possible to establish a relation to the range of subjective test result values (MOS, section 4.2.2) which is close to being linear. Furthermore they are recommended as being suitable for the evaluation of speech degraded by "transmission errors in real network environments such as bit errors and frame erasures" ([Uni98, YY99]).

Measuring Normalizing Blocks (MNB) The MNB method ([Vor97, Vor99a, Vor99b, Uni98]) focuses only on the most important properties of speech (section 2.1.1) with regard to its model of hearing. More emphasis is put on modeling the auditory judgment. Therefore MNB includes only a frequency mapping to Bark (critical frequency bands) as well as a logarithmic transformation from power to approximated perceived loudness as the hearing model. However the auditory judgment is modeled by analyzing the signal at multiple time (TMNB) and frequency (FMNB) scales. The following equations ([Vor97, Vor99a]) describe such an operation in continuous time for a TMNB where $R(t,f)$ is the reference signal, $T(t,f)$ is the test signal (input to MNB), $\tilde{T}(t,f)$ is the test signal where the measured deviation $e(t,f_l)$ in a critical frequency band ranging from f_l to f_u has been removed (output of MNB). Finally, $\{m_{2i-1}, m_{2i}\}, i \in [1,N]$ are the measurement results for this particular MNB:

$$
\begin{aligned}
e(t,f_l) &= \frac{1}{f_u - f_l}\int_{f_l}^{f_u} T(t,f)df - \frac{1}{f_u - f_l}\int_{f_l}^{f_u} R(t,f)df \\
\tilde{T}(t,f) &= T(t,f) - e(t,f_l) \\
m_{2i-1} &= \frac{1}{t - t_{i-1}}\int_{t_{i-1}}^{t_i} \max(e(t,f_l),0)dt \\
m_{2i} &= \frac{-1}{t - t_{i-1}}\int_{t_{i-1}}^{t_i} \min(e(t,f_l),0)dt
\end{aligned}
$$

The individual measurements are grouped hierarchically from larger to smaller scales, i.e. the output signal $\tilde{T}(t,f)$ is the input signal of the next MNB structure (in the used hierarchies ([Uni98]) FMNB and TMNB structures are interspersed).

[8]Other approaches include e.g. using conventional speech recognizers ([Mil99, CLMT99]) for intelligibility assessment. While this approach is appealing due to the potentially widely used and accepted test receptors, it only covers a subset of the desired distance measure, that is the "phonetic distance" ([WSG92]). Additionally this approach is still in its early stages and also needs either standardization or a de-facto standard to become significant.

This should reflect the adaptation and reaction to the signal by a listener. MNBs are by design idempotent, i.e. if in a hierarchical structure two MNBs are identical the measurement result $\{m_{2i-1}, m_{2i}\}$ of the second MNB will be zero (i.e. an MNB removes the deviation of a perceptual component). Finally, the actual perceptual difference, also known as Auditory Distance (AD), between the two signals is a linear combination of the measurements where the weighting factors represent the auditory attributes. The higher AD is, the more the two signals are perceptually different and thus the worse the speech quality of the test signal is.

Enhanced Modified Bark Spectral Distortion (EMBSD) The Bark Spectral Distortion (BSD) measure ([WSG92]) assumes that speech quality is directly related to the speech loudness which is defined as the perceived feeling for a given frequency and sound pressure level ([ZF99, Nov96]). Loudness estimation is done using critical band analysis, equal-loudness preemphasis and the intensity-loudness power law. In discrete time the BSD measure is defined as the averaged squared Euclidean difference between the estimated loudness $L_T^{(j)}(i)$ of the test signal $T(n)$ and the estimated loudness $L_R^{(j)}(i)$ of the reference signal $R(n)$ where i is the index of the critical band i ([ZF99]) and j is the frame index (N being the number of frames and K being the number of critical bands, [YBY98]):

$$BSD = \frac{\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{K}\left[L_R^{(j)}(i) - L_T^{(j)}(i)\right]^2}{\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{K}\left[L_R^{(j)}(i)\right]^2}$$

The Modified BSD measure (MBSD, [YBY98]) defines the perceptual distortion as the estimated loudnesses' average difference and introduces a noise masking threshold below which perceptual distortion is not taken into account. This is expressed for each critical band i with a binary indicator $M(i)$ with $M(i) = 0$ if the distortion is imperceptible and $M(i) = 1$ if it is perceptible:

$$MBSD = \frac{1}{N}\sum_{j=1}^{N}\left[\sum_{i=1}^{K} M(i)|L_R^{(j)}(i) - L_T^{(j)}(i)|\right]$$

The difference between the MBSD and the enhanced MBSD (EMBSD, [YBY98, YY99]) is that a new cognition model based on post-masking effects and 15 loudness components are used, loudness vectors are normalized, and the spreading functions in noise masking threshold calculation are removed in the EMBSD. The result value for EMBSD is called "Perceptual Distortion". As MNB it also constitutes a distance measure, i.e. the larger the value, the worse the speech quality is.

4.2.2 Subjective testing

When comparing the quality of speech coding and transmission systems, methods of subjective quality assessment play a major role. Such speech assessment tests are grouped into two categories (section 2.1.4, [Pap87], S. p. 186):

- Intelligibility tests: recognition of particular words by different test persons: *What* is the speech content ?
- Quality tests: assessment of entire utterances. *How* is the speech perceived ?)

For the examination of the performance of loss recovery and particularly loss concealment algorithms, the test of choice is a *quality* test. This is the case, because the goal of loss recovery is typically to increase the audibility of the distorted speech signal rather than the repair of a heavily distorted signal where the intelligibility is affected. However, there are of course interconnections of intelligibility and quality: good quality implies good intelligibility (the converse is not necessarily true).

Speech quality is a multidimensional variable. Therefore we can summarize the following properties of subjective speech quality criteria:

- The ultimate goal of a speech signal is to be perceived and processed by a human being. Therefore the judgment by humans is of the utmost importance in the assessment of speech quality
- Different methods which introduce different artifacts into the speech signal can be compared. This is not possible with purely mathematical objective quality assessment (SNR), however objective methods like the one used in chapter 5.2.4 are to some extent capable of a comparison.

Disadvantages of subjective tests are:

- There may be a strong dependence on the particular used test signals: speaker (pitch frequency), length of the test signal, words used as speech material, recording quality (noise, clipping), placement of errors within the test signal
- time effort: a lot of different persons have to take the test for it to be meaningful
- largely different perception of the artifacts in the signal by the test subjects (resulting in a large variability)
- different interpretation of the term "speech quality" by different test persons; time-variant interpretation by one test person
- huge dependence on particular test conditions, therefore questionable reproducibility of the results
  a) ambient noise, quality of the audio hardware (A/D converters) and headsets/loudspeakers
  b) order of the test signals
  c) number of the test signals presented in a sequence
  d) (non-)expertise of the test subjects
  e) possibility to listen several times to a particular test signal

Clearly, the previous arguments underline that subjective tests are inevitable, however that test results and conclusions based on these results have to be carefully evaluated. Additionally, it is important not to compare test results of different test events.

4.2.2.1 Methodologies for subjective speech quality assessment

In [IEE69], three basic methods for subjective speech quality measurement have first been identified and recommended. The ITU-T P.800 document ([Uni96d]) standardizes similar subjective quality measurement for speech transmission systems. Additionally to conventional "listening tests", P.800 also describes "conversation-opinion tests", where an actual test conversation is assessed. ITU-T P.830 ([Uni96e]) elaborates the methods introduced in P.800 for telephone-band and wide-band digital codecs.

Preference methods For the *isopreference method*, the test signal (i.e. in our case the distorted signal enhanced by a loss recovery mechanism) is compared directly to reference signals with different amounts of distortion ([DPF89]). The degree of distortion of the reference signal is described by a parameter, typically the SNR. The isopreference value is the parameter value of the reference signal, where 50% of the test persons vote against (and for respectively) the test signal. When applied to loss recovery, this method lacks a suitable parameter which allows comparisons between different algorithms. An SNR is not suitable here for the reasons mentioned in section 4.2.1. It would be suitable e.g. for assessment of codecs where a quantization parameter can be varied. Also, using the unconditional loss probability ulp as a parameter does not seem to be adequate, because then the listener is exposed to very different artifacts (e.g. an interrupted distortion for the non-concealed signal vs. some echoing introduced by a loss concealment algorithm based on simple segment repetition). If we also consider the expected time consumption for the test (every test signal has to be compared to a range of reference signals) the isopreference method does not seem to be suitable for the assessment of loss recovery methods.

Using the *relative preference method* (paired comparisons, Comparison Category Rating: CCR, [Uni96d]), the test signal is compared directly to reference signals with varying amounts of distortion. Therefore for loss recovery methods the same problem as described above, the choice of the test parameter, appears. Here, however, the reference signals are also compared among each other, thus constituting a "quality axis", on which the results of test signal / reference signal comparisons can be measured.

For an assessment of loss recovery algorithms it makes sense to build a quality axis from comparisons of the original, some test signals with various degrees of distortion as well as a "worst case" signal which contains all artifacts. Then, all test signals processed by the different algorithms under test should be tested versus each other ([BS85], p.33) and versus the reference signals in both sequences AB and BA.

The advantage of this technique (in contrast to methods discussed below) is that only one decision between two alternatives (A or B) by the test person is necessary. Thus even minor differences in quality can be examined. However, this comes at a high cost in terms of the number of tests per person: To test a algorithms/QoS enhancement mechanisms the generation and assessment of the following signals is necessary:

$ref = z + 2$ reference signals:

- Original: $x(n)$
- z distorted signals under the network loss condition i: $x_i(n), \quad i \in [1, z]$
- "Worst case signal": $x_{wc}(n)$

$test = a\ z$ test signals $y_{a,z}(n)$
(for all a QoS enhancement mechanisms the z signals have to be treated)

For o originals $x(n)$ we have $o(ref + test)(ref + test - 1)$ necessary comparisons, thus every test subject has to listen to $b = 2o(ref + test)(ref + test - 1)$ speech signals. For $z = 2$,$a = 3$ and $o = 4$ e.g. we have $b = 720$. If reference and test signals are assessed separately and only one of the sequences AB and BA is presented b is computed as follows:

$$b = o(ref(ref - 1) + test(test - 1))$$

For the example given we have $b = 168$.

As the scheme introduced below is still below the latter value for b the relative preference method has not been adopted.

Category judgment This test is based on an assessment of the overall impression of the speech signal quality by the test persons into intuitively clear categories (→ Tab. 4.5) on an absolute scale (Absolute Category Rating: ACR [Uni96d]).

category	*speech quality*	*level of distortion ([Del93], p.578)*
1	unsatisfactory	very annoying and objectionable
2	poor	annoying but not objectionable
3	fair	perceptible and slightly annoying
4	good	just perceptible but not annoying
5	excellent	imperceptible

Table 4.5: Speech quality categories

The test is divided in two phases:

1. "Anchoring"

2. Assessment phase

The goal of the "anchoring" phase is that the test persons can align their concept of perceived quality with the quality scale (1 to 5). One possibility is that the original signal $x(n)$ as well as a "worst case" signal $x_{wc}(n)$ (category 1), which contains all the distortions of the different output signals (with and without QoS enhancement) are presented. The evaluation of the category judgment test is typically done using a *'Mean Opinion Score* (MOS, [IEE69], p.232): the percentage of persons who have chosen category i ($l_{\%,i}$) is weighted with the category i, thus yielding the average of the judgments.

$$MOS = \frac{\sum_i l_i \; i}{l} \tag{4.10}$$

l_i: number of test persons who have chosen category i
l: number of test persons

A major disadvantage of this test procedure is that some test persons may always judge a signal as better or worse respectively than other, thus leading to a high variability of the results. Thus some measure of variability like the standard deviation should be taken into account. Additionally the subjective scale of a person might not be equidistant (therefore it is always problematic to compute quality measures based on averages of judgments by different people).

Main advantage of the category judgment method is the reasonable effort/result tradeoff (see p.85). The number of necessary judgments is $o(ref + test)$. For the parameter choice given earlier we have $b = 40$ test conditions. Thus the time effort per person is low resulting in a low probability of mis-concentration and tiredness of the test subject. Due to these reasons the category judgment test with MOS evaluation is the most widely used subjective speech quality assessment method ([ST89, Yon92, Pap87]).

4.3 Relating speech quality to packet-level metrics

Functions which describe the sensitivity/tolerance of users with regard to characteristic parameters which influence the performance of the application are typically called "utility" or "satisfaction" functions ([She95, RI97, BFPT99]). For Internet multimedia applications, utility functions should relate the available network resources for a particular flow to the perceived quality. Fig. 4.14 shows a conventional utility curve for waveform-coded voice dependent on the unconditional loss probability (ulp). The strong performance degradation starting at low loss rates is due to the missing ability of the source to adapt its rate for delivering a "complete" stream with gracefully degraded quality (cf. section 2.1.1). However this curve is only schematic (it has been based various subjective test results: [GS85, ST89, HSHW95, SSYG96, San98b, San98a]) and thus can give only a crude,

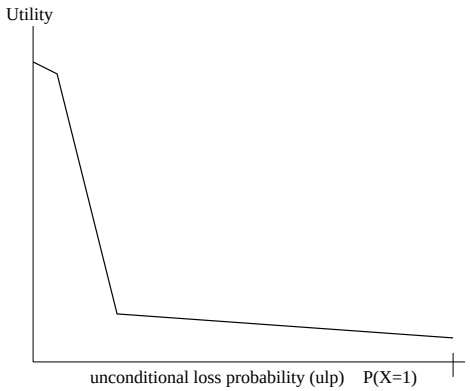


Figure 4.14: Simple utility function for sample-based voice (schematic)

qualitative impression on the variability of the speech quality. Additionally, as we have seen in chapter 4.1, using a long-term packet-level metric like the *ulp* is most probably not adequate as a basis for speech quality metrics as it hides the short-term variability of the transmission path.

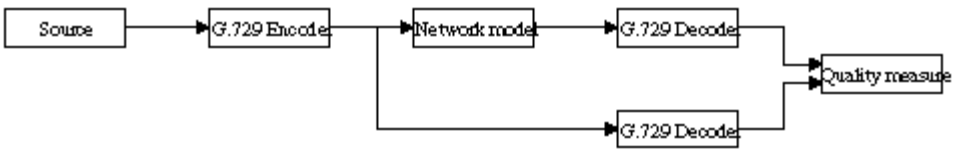


Figure 4.15: Model for generating utility curves for a particular speech codec

Therefore to retrieve a realistic utility function, we use the network model as developed in chapter 4.1 to generate loss patterns. As the model for the sake of simplicity does not imply the notion of the actual packet sequence (sequence number s), we apply it several times to actual samples containing male and female voices using different seeds for the random process to generate different loss patterns. By averaging the result of the objective quality measure for several loss patterns, we have a reliable indication for the performance of the codec operating under a certain network loss condition[9] (for the following examples we used the G.729 (section 2.1.3.2) speech codec). The "network loss condition" is described by the

[9]Note that instead of employing fixed utility functions, which have been measured previously, the objective speech quality measures introduced in the previous paragraph allow to compute a quality value on the fly using the actual speech material which should be transmitted. This can be then be used to trigger pro-active protection methods against loss (cf. e.g. section 3.1.2.2) or in the feedback loop of the excitation search of an analysis-by-synthesis speech encoder (cf. section 2.1.3.2, p. 19).

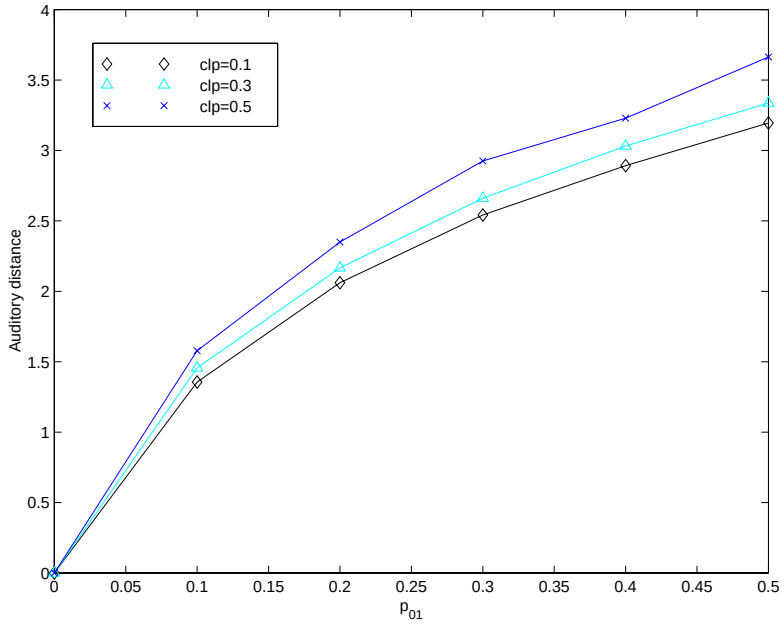


Figure 4.16: Utility curve based on the Auditory Distance (MNB)

parameter pair p_{01} and clp (conditional loss probability) of the loss run-length model with model order $m = 1$ (section 4.1.4). See Figure 4.15 for the building blocks of a model to generate utility curves for a particular speech codec.

The results for MNB (section 4.2.1.2) and EMBSD (4.2.1.2) given in figures 4.16 and 4.17) show that with increasing p_{01} and clp in the Gilbert model (and thus increasing packet loss rate and loss correlation), the auditory distance (in case of MNB) and the perceptual distortion (in case of EMBSD) are increasing, i.e. the speech quality of the decoded speech signals is decreasing. As in the schematic utility curve which is based on subjective tests shown above, there is a quality drop considering the lossless case ($p_{01} = 0$, distance measure$= 0$) and the first measurement point at $p_{01} = 0.1$. Then, the results show a continuous, close to linear, decrease in quality when the probability to enter the loss state p_{01} is increased. It is also demonstrated that an increasing loss correlation (clp parameter) has some impact on the speech quality however the effect is relatively weak pointing to a certain robustness of the G.729 codec with regard to the resilience to consecutive packet losses (in section 5.2.3 we analyze the concealment of the G.729 decoder in more detail). These observations indicate that the two objective quality measures are reasonably related to the network model parameters and can be used for the speech quality evaluation of the loss recovery and control schemes influencing these parameters.

The final step in relating objective speech quality measures to user perception is to map the results of the objective speech quality measures to a finite range of values, which is then closely related to a mean opinion score (equation 4.10).

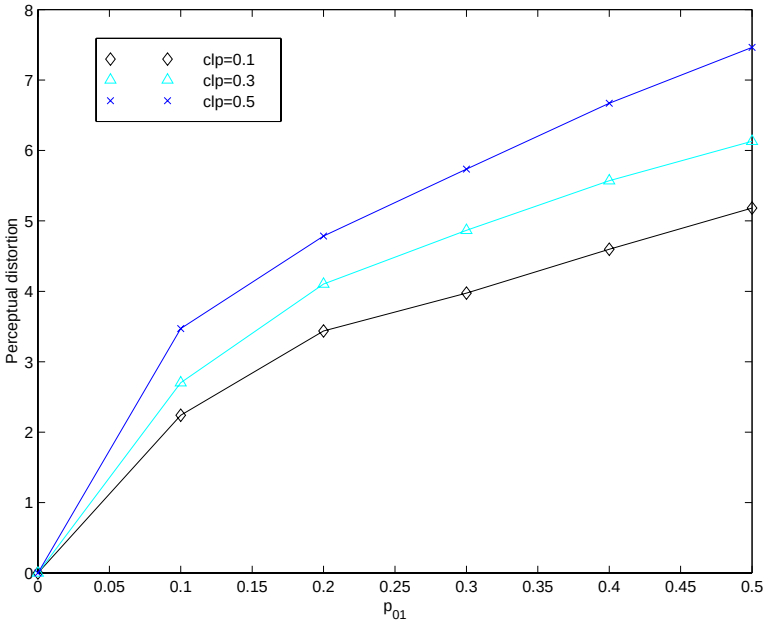


Figure 4.17: Utility curve based on the Perceptual Distortion (EMBSD)

While such a "logistic function" exists and is well motivated e.g. for the PSQM[10] method ([Uni98], chapter 10), we are not aware of such a function for EMBSD at the time of writing. For MNB, Voran ([Vor99a]) proposes to use the function $f(AD) = (1 + \exp aAD + b)^{-1}$, where a and b are constant parameters. While for MNB, values for a and b as used in [Vor99a] could be taken, the parameter choice for EMBSD is less obvious. In a comparison of different speech quality metrics with regard to the correlation with subjective test results ([Vor99b]), these parameters are simply chosen such as to maximize that correlation. Obviously this approach is not applicable here due to the lack of validating subjective test results. Therefore in Table 4.6 (cf. Table 4.5) we give approximate values for the direct mapping of MOS to auditory distance and perceptual distortion respectively. Those values are derived from visual inspection of the results in Figures 4.16 and 4.17 considering the similarity of the test conditions for MNB and EMBSD and the subjective quality range.

Corresponding to the components of the generic structure for audio tools (Figures 3.1 and 3.2), Figure 4.18 shows the components of our generic measurement setup which we will use to design and evaluate our approaches to end-to-end-only as well combined end-to-end and hop-by-hop loss recovery and control. The shaded boxes show the components in the data path where mechanisms of loss recovery can be located. For every approach we will identify which components at which locations are enabled. Together with the parameters of the network model (sec-

[10]The PSQM is recommended ([Vor99b, Uni98]) as being less suitable for the analysis of the impact of "frame erasures" therefore we did not include it as an objective speech quality measure.

category	*speech quality*	*MNB Auditory Distance*	*EMBSD Perceptual Distortion*
1	unsatisfactory	4	8
2	poor	3	6
3	fair	2	4
4	good	1	2
5	excellent	0	0

Table 4.6: Provisional conversion table from MOS values to Auditory Distance (MNB) and Perceptual Distortion (EMBSD)

tion 4.1) and the perceptual model (or the subjective test conditions, section 4.2) we obtain a measurement setup which allows us to map a specific PCM signal input together with network model parameters to a speech quality measure. While using a simple end-to-end loss characterization, we generate a large number of loss patterns by using different seeds for the pseudo-random number generator (for the results presented here we used 300 patterns for each simulated condition for a single speech sample). This procedure takes thus into account that the input signal is not homogeneous (i.e. a loss burst within one segment of that signal can have a largely different perceptual impact than a loss burst of the same size within another segment).

4.4 Packet-level traffic model and topology

The models introduced in section 4.1 allow a comprehensive end-to-end characterization of the loss process and make it possible to easily link perceptual metrics to an end-to-end model (sections 4.2 and 4.3). Thus, an end-to-end performance assessment for end-to-end-only loss recovery algorithms as well as network-supported end-to-end mechanisms is possible. However, for the design and performance evaluation of supporting hop-by-hop loss control schemes it is important to simulate the behavior of individual network elements. To characterize this behavior, i.e. how a certain scheduling/queue management algorithm can deal with arriving traffic causing congestion, it is necessary to simulate individual packet arrivals and departures (discrete event simulation). Then, again, the developed loss metrics can be employed to provide a comprehensive characterization of the behavior of an individual or concatenated network elements.

We employ the NS-2 network simulator ([UCB98]) and implement the proposed hop-by-hop loss control schemes (chapter 6) in addition to the available drop tail and RED (section 3.2.1) queuing disciplines. Furthermore we extend the flow monitoring capabilities to allow for the tracing of the occurrence o_k of burst losses of length k as introduced in section 4.1.1 at every node.

We use a traffic model that reflects results from various recent Internet Access-LAN and Internet backbone measurements (e.g. [CMT98] and [NLM96, IKL97]):

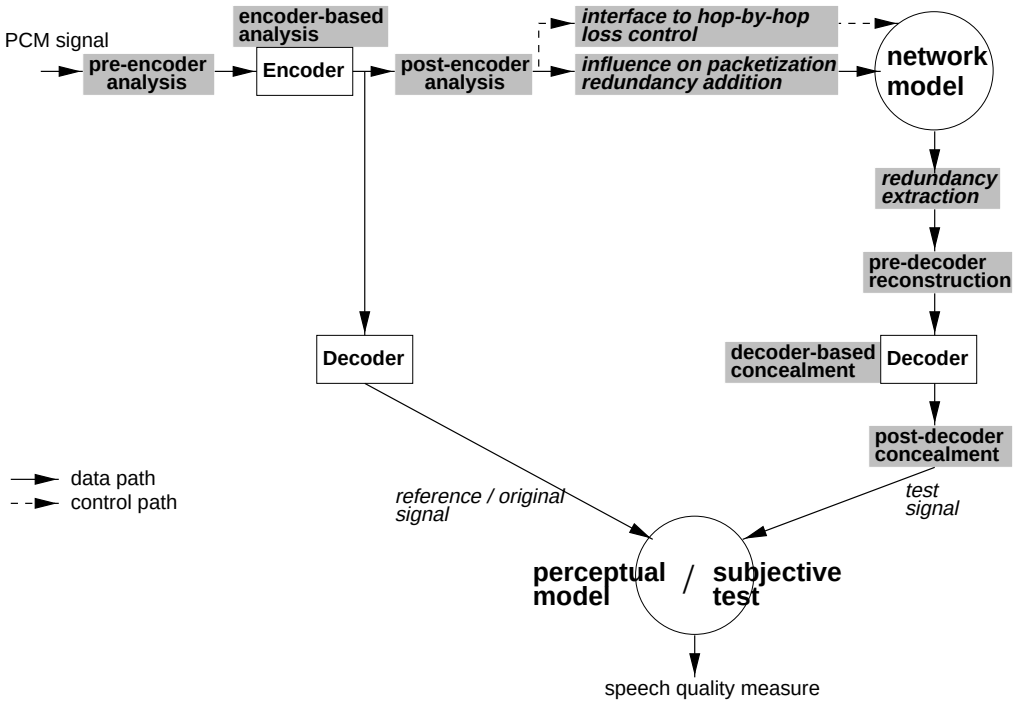


Figure 4.18: Components of the loss recovery/control measurement setup

the majority of traffic (in terms of flows and volume) are http transfers ("H-type" background traffic). The rest are mostly short-lived flows dominated by DNS traffic ("D-type" background traffic), which has a relatively large share of the active flows, yet only a small share of the traffic volume[11]. The values we chose for modeling of individual sources are shown in Table 4.7. To model Web traffic we use a Pareto distribution ([CB97b]) both for the ON and OFF periods of the source. By using a variance-time ($var(X(m)) - m$) plot ([LTWW93]), describing the variance of the process of arrivals X dependent on the scale of averaging m, we determined that the aggregation of the described background traffic sources produces long-range dependent traffic. As the loss control algorithms try to influence the loss burstiness of individual flows, it is crucial to reflect the existing "burstiness on all time scales" of the aggregate arrival process in the model. To model voice sources with silence detection, we employed a model widely used in the literature (see e.g. [NKT94]) where ON (talk-spurt) and OFF periods are exponentially distributed with a speaker activity of 36%.

Table 4.7 also gives "raw" peak bandwidths and packet sizes (i.e. including packet header overhead[12]). The range of $30...34\frac{kBit}{s}$ D-type BT bandwidth and $0.12...0.14s$ for the on-/off-times is due to the changing number of flows and load

[11]The small per-flow bandwidth of the D-type BT allows us to set the background traffic load with a relatively fine granularity.

[12]We assume 8 octets link level overhead and 20, 20, 8, 12 octets IP-, TCP-, UDP-, RTP-packet overhead respectively.

	H-type BT	*D-type BT*	*FT (voice)*
flow share (%) (of background traffic)	75	25	-
peak bandwidth ($\frac{kbit}{s}$)	256	30...34	83.2
packet size (octets)	8+20+20+512	8+20+8+92	8+20+8+12+160
on/off distribution	Pareto	Exponential	Exponential
shape parameter	1.9	–	–
mean burst length (packets)	20	4	18
mean ontime (s)	0.35	0.12...0.14	0.36
mean offtime (s)	0.7	0.12...0.14	0.64

Table 4.7: Source model parameters

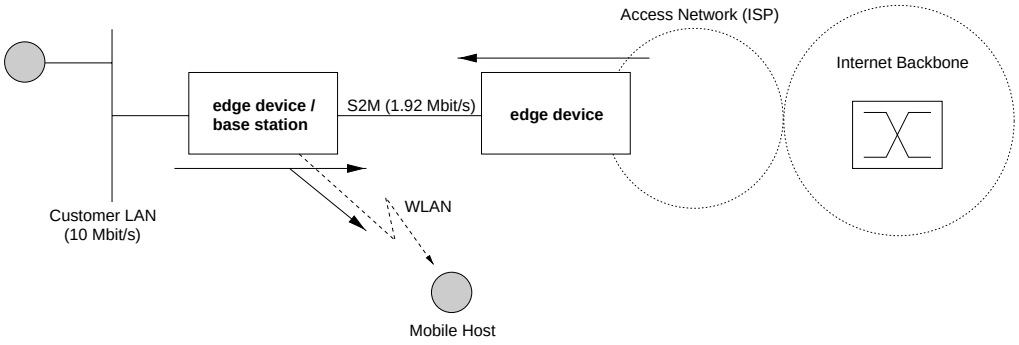


Figure 4.19: Simulation scenario (single-hop topology)

in the experiments presented in chapter 6. Packet inter-departure times within a burst are uniformly distributed in the interval $[0.95I, 1.05I]$ (with I being the packet inter-departure time calculated from the values of Table 4.7) to avoid phase effects caused by the exact timing of packet arrivals in the simulator.

We have found a simulation time of $5 \cdot 10^4$ seconds (13.9 hours[13] with the number of packet arrivals ranging from $16 \cdot 10^6$ to $27 \cdot 10^6$) sufficient for the Pareto sources to "warm up" and thus to guarantee that the traffic shows long-range dependence as well as to result in a statistically relevant number of drop events even for low loss rates as a basis for performance measures ($p_{L,cond}$). We have averaged the results for one flow group (H, D, voice). In figures presenting our results in chapter 6 we also plot error bars giving the standard deviation for the averaged values (this is to verify that every flow of a group has identical behavior seen over the entire simulation time).

We use two simple network topologies: In the first, several background and foreground flows experience a single bottleneck link (e.g. an small bandwidth access link connecting a customer LAN to an ISP or a base station connecting mobile hosts

[13]The initial 10^4 s were discarded from the datasets.

to a LAN, Fig. 4.19). In our simulation the bottleneck link has a link-level bandwidth of $\mu = 1920kbit/s$ (which is a typical bandwidth for an ISDN[14] PRI or an xDSL access). Several flows fed to the gateway over $10Mbit/s$ links are multiplexed to either a Drop-Tail (DT), a PLoP queue, a DiffRED queue or a conventional RED[15] output queue. This topology is used in section 6.2 and 6.3.

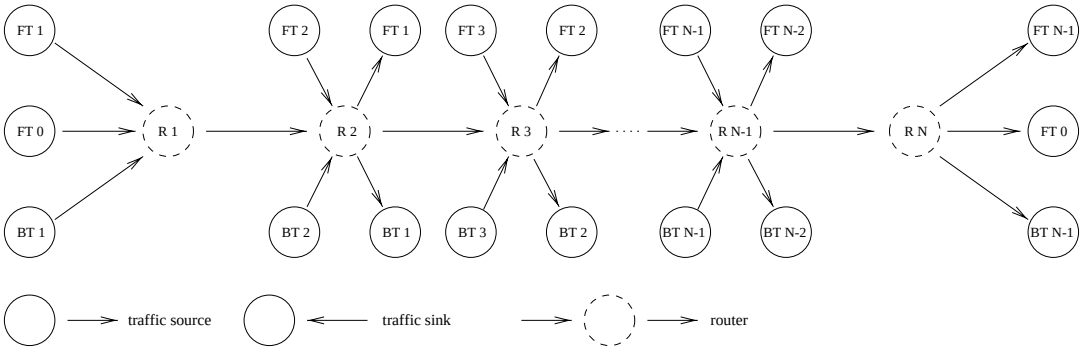


Figure 4.20: Multi-hop network topology for the simulations

The second network topology consists of a concatenation of several instances of the first scenario. As shown in figure 4.20 the foreground traffic consists of flows which pass through the whole path and are our main focus of interest (FT0) and flows which simulate cross traffic. The share of cross FT within the FT is set to 50%. In the figure routers are designated by Rx, FT sources and destinations by FTx and BT sources and destinations by BTx where x is a number for the path from the source x to the destination x. At every node also new background cross traffic is injected. This topology is used in section 6.4.

4.5 Conclusions

In this chapter, we have developed a novel framework model which provides a comprehensive characterization of the loss distribution within a flow. The Markov model is based on identifying a certain model state with the occurrence of a certain loss run-length.

We started by demonstrating the necessity of loss metrics by using an informally defined short-term (window-based) mean loss rate. Then a general Markov model has been introduced which is on one hand able to capture adequately the loss process, however on the other hand is very complex. Using the validated assumption that past loss events affect the next loss event much more than successfully arrived packets, we then derived a run-length-based model from the general Markov model. This model is of significantly reduced complexity, but still allows to capture loss bursts of a length up to the model order with the full precision of the general

[14]We neglect connection setup times and the fragmentation into channels.
[15]We have used the implementation of the NS-2 distribution.

Markov model (the no-loss runs and the position between loss and no-loss runs are not captured however). Several performance metrics (mean loss, conditional loss, mean burst loss length) have been introduced in their run-length-based definition. Returning to our starting point, we also have shown how this model can serve in the approximation of a window-based mean loss rate yet without keeping track of the individual position of the loss bursts within the flow. By reducing the number of states to two we provided a run-length based definition of the well-known Gilbert model. The relationship between Gilbert model parameters and the parameters used in the higher order models has been discussed. We have highlighted the different meaning of the metrics when they are either conditioned on packet events (random variable X) or burst loss events (random variable Y). Finally, by applying the run-length-based models to measurement traces of IP voice flows, we demonstrated the tradeoffs between accurate multi-parameter modeling and employing the simple two-state Gilbert model.

We were able to found a framework in which most of the previously unrelated loss metrics existing in the literature can be defined and used together. Generally the model should be used when with regard to the application level, the loss process cannot be adequately described by just comparing the impact of *isolated* losses versus the impact of *burst* losses. We conclude that if any of the following conditions does *not* apply, a run-length based model is very useful (otherwise a Gilbert model yields sufficient information):

- simple applications (like sample-based voice traffic), i.e. the loss impact on the decoder at the user-level is clearly different for isolated losses versus losses that occur in bursts.
- simple end-to-end loss recovery (when e.g. the FEC employed cannot repair burst losses).
- no "outages" are contained in a trace (i.e. the loss run-length distribution is not "heavy-tailed")
- only "conventional" queue management (Drop Tail, RED) is used throughout the flow's path (see section 6.4.1).

For future work, the autocorrelation of the loss indicator function, as well as the autocorrelation and composite metrics (like the cross-correlation function) of the loss/no-loss run-lengths ([YMKT98]) should be used when further analyzing the areas of applicability of the run-length models.

In section 4.2 we have then introduced ordinary and perception-based objective speech quality metrics and discussed different methods of subjective testing of speech quality. Then in section 4.3 we showed how to derive functions which relate packet-level metrics to objective speech quality measures. We employed a run-length-based model to produce synthesized loss patterns and linked the results with objective speech quality when using a particular codec. We provided a provisional conversion table to allow comparisons to results of subjective tests (MOS values). Thus a much

more precise characterization of speech quality is possible than the one just based e.g. on a long term loss rate linked to a Signal-to-Noise Ratio. Finally, in section 4.4 we have described the employed traffic model and topology for the simulation of individual network nodes.

Chapter 5

End-to-End-Only Loss Recovery

The basic thoughts on the different impact of packet loss on sample- and frame-based codecs expressed in section 2.2.1.1 also point to a separate treatment of sample-based and frame-based codecs in terms of end-to-end loss recovery. The discussion of the related work in that area in the previous chapter, particularly the discussion of LP-based waveform substitution on p. 48 and codec-specific loss concealment in section 3.1.3.3, has also supported this argument. Therefore we present in section 5.1 our approach for sender-supported loss concealment for sample-based codecs. While feasible in principle, simply mapping sample-level loss concealment mechanisms to frame-based codecs does not result in an acceptable cost/quality tradeoff (section 5.2.1). Therefore in section 5.2, an approach using selective addition of redundancy in connection with codec-specific concealment is developed.

5.1 Sample-based codecs

Sample-based codecs are still very important for voice transmission in general as well as for Voice over IP. This is true even when comparing the only limited compression achievable to low-bit-rate codecs, which are typically frame-based. Several aspects support this argument: tandem configurations[1] and transcoding in the network can be supported without an extreme impact on quality. Also, using a sample-based, only lightly compressed voice format gives more flexibility when voice streams are stored for further processing. Finally, sample-based coded voice flows have a higher intrinsic loss resilience due to no (for memoryless coding: μ- or A-law PCM) or few error propagation.

Thus, speech properties which are not exploited in the coding process can be used to enhance the flow's resilience to packet loss. One such key property is the long-term correlation within a speech signal (section 2.1.3). In section 3.1.3 we have observed that in known concealment schemes, because the fixed packetization

[1] Tandeming occurs when devices within the network perform decoding of a voice stream and feed the reconstructed signal to another encoder for further transmission. In mixed circuit-switched and packet-switched networks this will frequently be the case (it is also needed when MCUs (multipoint control units, [MM98]) are employed).

interval is unrelated to the long-term correlation, the relative "importance" of the packet content and changes in the speech signal cannot be taken into account in the concealment scheme. That means that some parts of the signal cannot be concealed properly due to the unrecoverable loss of entire phonemes. Furthermore, experience with the time-scale modification technique (p. 47) has shown that the speech quality deteriorates also because of the specific distortions introduced by the different concealment techniques. This phenomenon is known as the "assymetry effect" ([Bee97]). Therefore we aim to use the long-term correlation to influence the packetization interval of a voice stream at the sender before sending it over a lossy packet-switched network. If a packet is lost, the receiver can conceal the loss of information by using adjacent signal segments of which (due to the pre-processing/packetization at the sender) a certain similarity to the lost segment can be assumed.

5.1.1 Approach

We propose a scheme called *Adaptive Packetization and Concealment* (AP/C, [San98b, San98a]), which maps the the basic speech property of periodicity (section 2.1.1) on the packet size resulting in variable size packets. Previously, in [SB85] and [SF85], variable size packetization has only been proposed for variable compression of PCM voice. The approach of looking at variable size segments of the speech waveform has also been adopted for compression in [KH95], where the encoding parameters describe the evolution of a characteristic waveform segment.

At the sender, auto-correlation of the signal is used for pitch period estimation. Then, two audio "chunks" of estimated pitch period length are packed into one packet. This results in small packets being sent for voiced speech, large packets sent for speech classified as unvoiced (which includes noise and silence).

When loss is detected at the receiver, adjacent speech "chunks" (of the previous and the following) packet are reused. Only a simple sample rate conversion needs to be performed on those chunks to scale them to the needed length and subsequently fill the gap caused by the lost packet.

Figures 5.1 and 5.2 show the basic structures of the sending and receiving entities of an audiotool with added functional blocks for AP/C (cf. Fig. 2.5). Note that in principle, any speech coder able to operate on variable size frames can be used, as the signal is analyzed before the encoder and concealed (when lost) after the decoder.

5.1.2 Adaptive Packetization / Concealment (AP/C)

5.1.2.1 Sender algorithm

The part of the sender algorithm interfacing to the audio device copies PCM samples from the audio device to its input buffer (Fig. 5.3). Pitch period estimation is done by auto-correlation and short-term energy measurement of an input segment of $2T_{max}$ samples (T_{max} being the correlation window size). The auto-correlation is

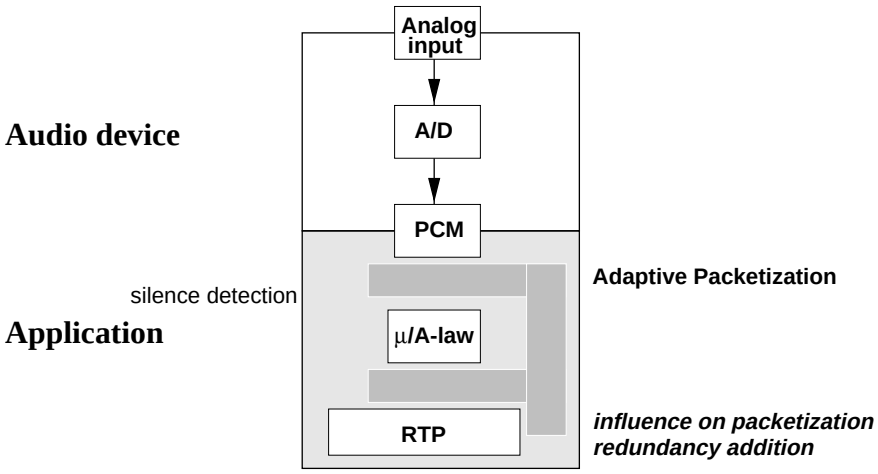


Figure 5.1: Structure of an AP/C enhanced audio tool (sender)

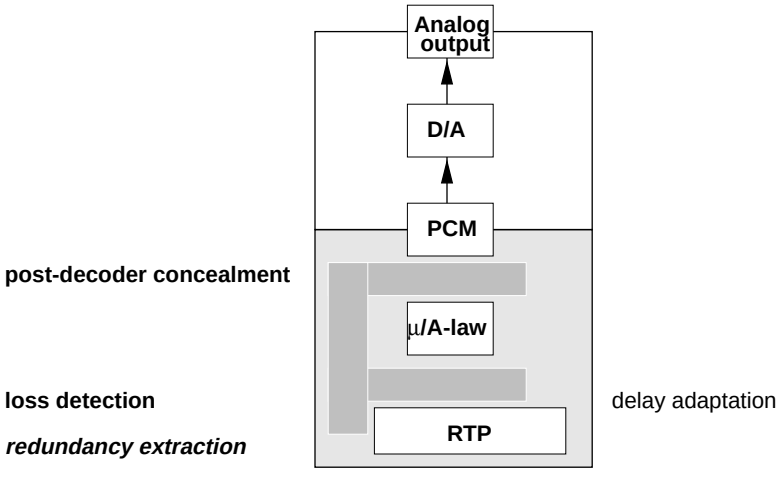


Figure 5.2: Structure of an AP/C enhanced audio tool (receiver)

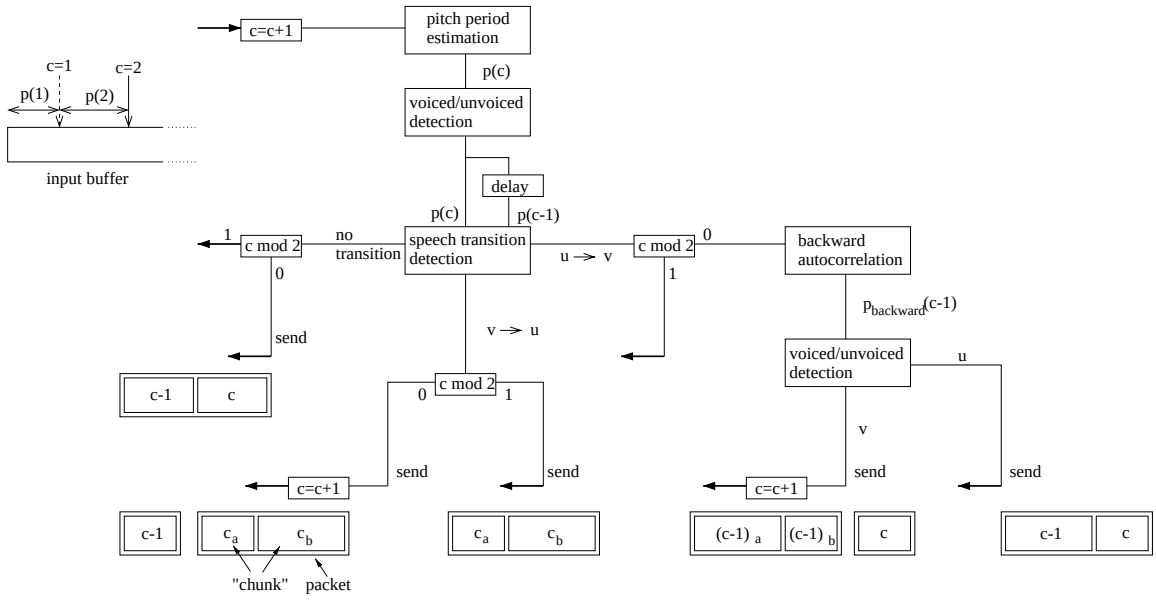


Figure 5.3: AP/C sender algorithm

defined as follows (where $s(n)$ is the signal, cf. Eq. 2.3):

$$r_{ss}(k) = \sum_n s(n)s(n+k) \tag{5.1}$$

The result is the value $p(c)$ (c being the number of the found segment, which we call "chunk") reflecting the periodicity for voiced speech (note that only a reliable detection of periodicity and changes in periodicity is necessary; the exactness of the pitch period value itself is not as crucial as when used for speech coding). For unvoiced speech, the algorithm typically picks a value close to T_{max} (Fig. 5.4/Fig. 5.5). Then the input buffer pointer is moved by $p(c)$ samples (thus constituting a "chunk"), c is incremented and if necessary new audio samples are fetched from the audio device.

Another routine (which may run in parallel and should be integrated with the silence detection (section 2.2.2) function) provides a simple check for speech transitions:

$$\Delta p = |p(c) - p(c-1)| > \Delta T$$
$$\text{and either}$$
$$p(c) < T_u \text{ and } p(c-1) \geq T_u \text{ (unvoiced} \rightarrow \text{voiced: } \mathbf{uv})$$
$$p(c) \geq T_u \text{ and } p(c-1) < T_u \text{ (voiced} \rightarrow \text{unvoiced: } \mathbf{vu})$$

where ΔT and T_u are pre-configured, fixed bounds.

To alleviate the incurred header overhead, which would be prohibitive for IP-based transport if every chunk is sent in one packet, two consecutive chunks are associated to one packet (see Figures 5.4 and 5.5).

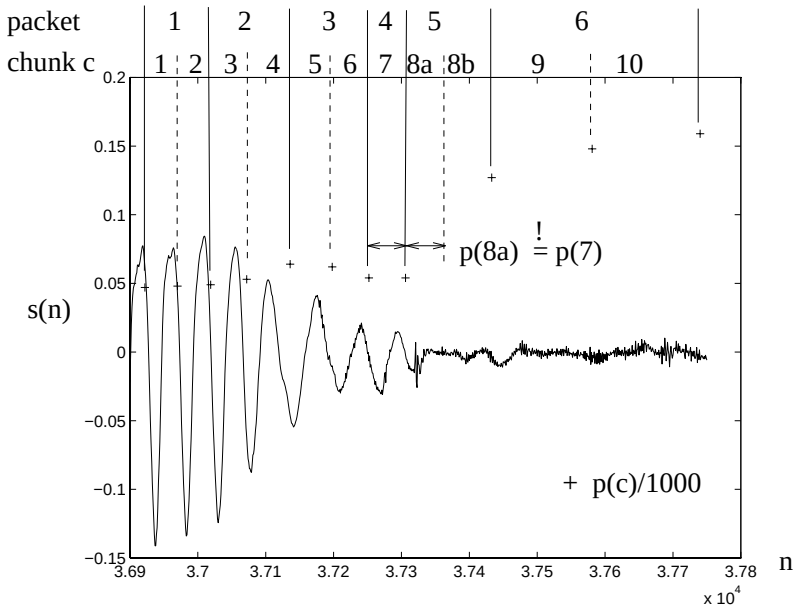


Figure 5.4: AP/C sender operation: transition voiced → unvoiced

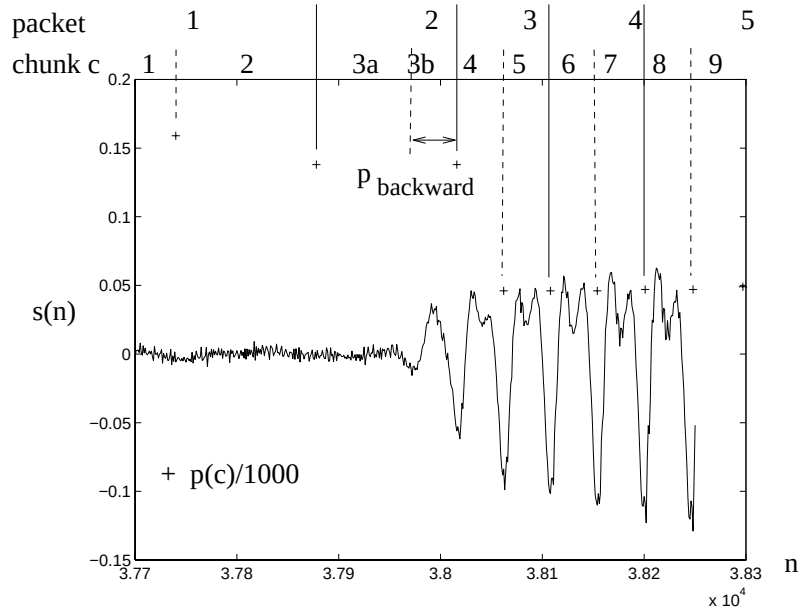


Figure 5.5: AP/C sender operation: transition unvoiced → voiced

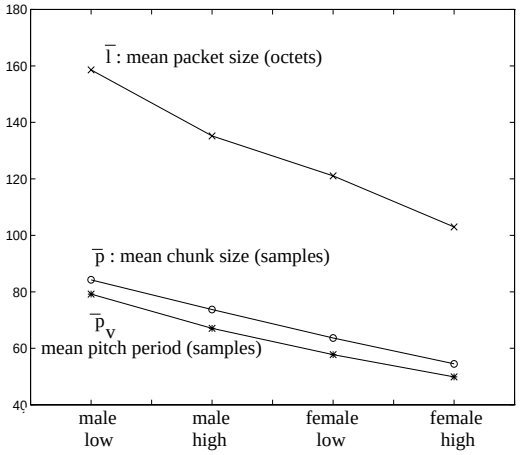


Figure 5.6: Dependency of the mean packet size $\bar{l}$ on the mean chunk size $\bar{p}$ and mean pitch period $\bar{p}_v$

However, if a *vu* transition has been detected, the "transition chunk" is partitioned into two parts (8a/b in Fig. 5.4) with $p(c_a)$ set to $p(c-1)$ and $p(c_b) = p(c) - p(c_a)$ where $p(c)$ is the original chunk size. Note that if $c \ mod \ 2 = 0$, the chunk $c-1$ (no. 7 in Fig. 5.4) is sent as a packet containing just one chunk.

When a *uv* transition has taken place, *backward* correlation of the current chunk with the previous one (no. 3 in Fig. 5.5) is tested as it may already contain voiced data (due to the forward auto-correlation calculation). If true, again the previous chunk is partitioned with

$$p(c_b - 1) = p_{backward}(c-1) \ \text{ and } \ p(c_a - 1) = p(c-1) - p(c_b - 1)$$

where $p_{backward}$ is the result of the backward correlation. Note that the above procedure can only be performed if $c \ mod \ 2 = 0$, otherwise the previous chunk has already been sent in a packet (a solution to this problem would be to retain always two unvoiced chunks and check if the third contains a transition, however the gain in speech quality when concealing would not justify the incurred additional delay).

With the above algorithm "more important" (voiced) speech is sent in smaller packets and thus the resulting loss impact/distortion is slightly less significant than using fixed size packets of the same average length, even without concealment. Note that this carries the assumption that the network's loss probability parameters are independent of the packet size. This in turn depends on the calculation of the queue size in packets or in bytes ([FJ93]).

With our scheme, the packet size is now adaptive to the measured pitch period. Fig. 5.6 shows this dependency for four different speakers. The mean packet size $\bar{l}$ is approximately twice the mean pitch period $\bar{p}_v$, as the most frequent combination is a packet consisting of two voiced chunks.

Distributions of the packet size for test signals of about $10s$ featuring four different speakers in Fig. 5.7 ($n(l)$: number of packets of size l octets, N: overall number

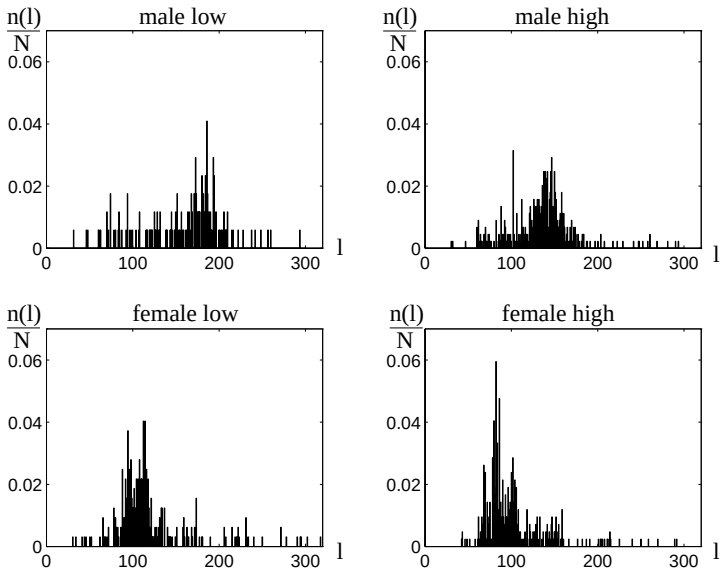


Figure 5.7: Normalized packet size frequency distributions for four different speakers

of packets) show that the parameter settings[2] can accomodate a range of pitches, as their overall shapes are similar to each other. As mentioned above, the most common packets contain two voiced chunks (*vv* packets), as distributions are centered around a value that is twice the mean pitch period (i.e. the mean of voiced chunks).

Fig. 5.8 shows the resulting relative packet header overhead for different speakers. The overhead is comparable to a typical parameter setting in IP networks (160 octets payload [$= 20ms$ $\mu-$law PCM audio] in an IP/UDP/RTP packet [20+8+12 octets header]), yet increases with increasing mean pitch period.

To support a possible concealment operation it is necessary to transmit the intra-packet boundary between two chunks as additional information in the packet itself and the following packet. That amounts to two octets of "redundancy" for every packet, that can either be transmitted by the proposed redundant encoding scheme for RTP ([PKH+97]) or by using the RTP header extension (cf. sections 2.2.3.1 and 5.1.5).

5.1.2.2 Receiver algorithm

At the receiver, packet loss is detected by means of RTP sequence numbers (and timestamps when using silence detection), taking into account the current play-out delay (when late packets have to be assumed as lost). Due to the pre-processing at

[2] $T_{min} = 30$ (start offset point of the auto-correlation); $T_u = 120$; $T_{max} = 160$ samples. Note that the packet size extends from sending a single voiced chunk ($l \geq T_{min}$) to sending two unvoiced chunks ($l \leq 2T_{max}$).

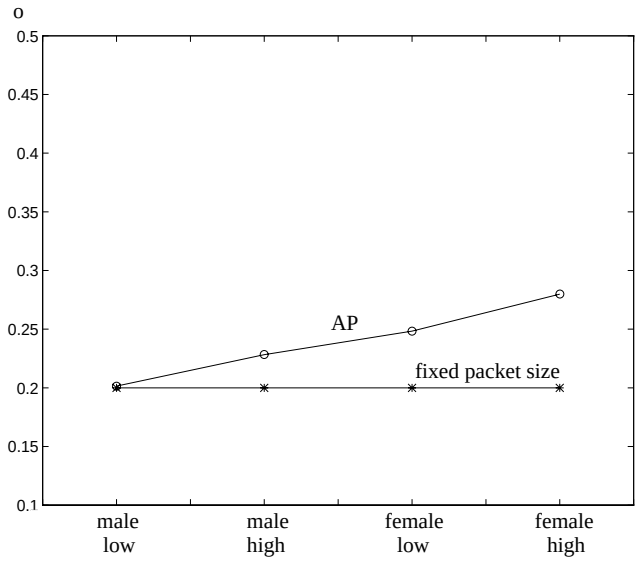


Figure 5.8: Relative cumulated header overhead o for AP and fixed packet size (160 octets) assuming 40 octets per-packet overhead for four different speakers

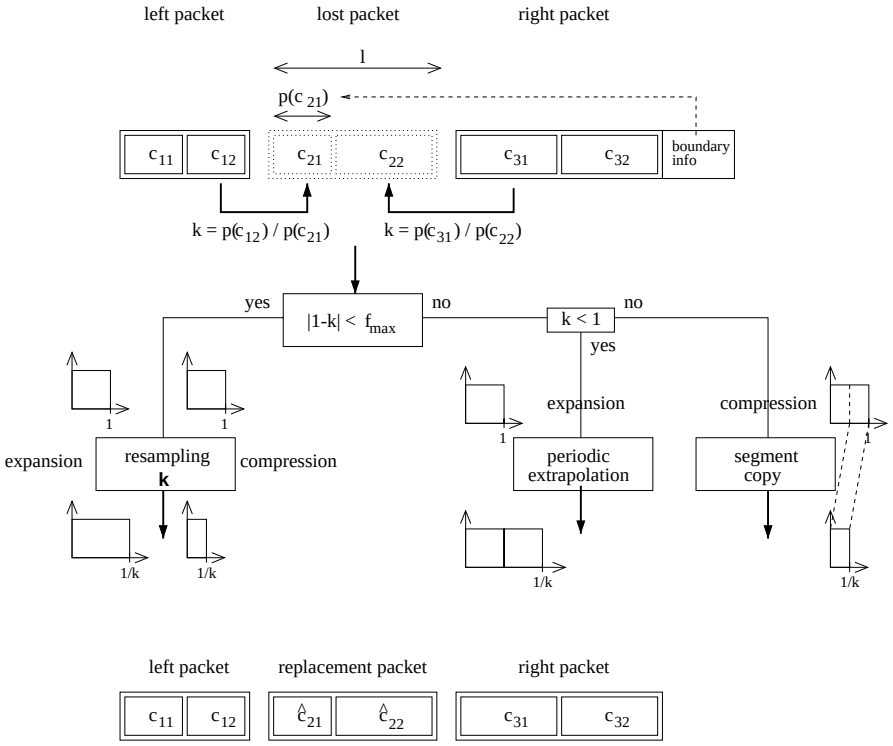


Figure 5.9: AP/C receiver operation

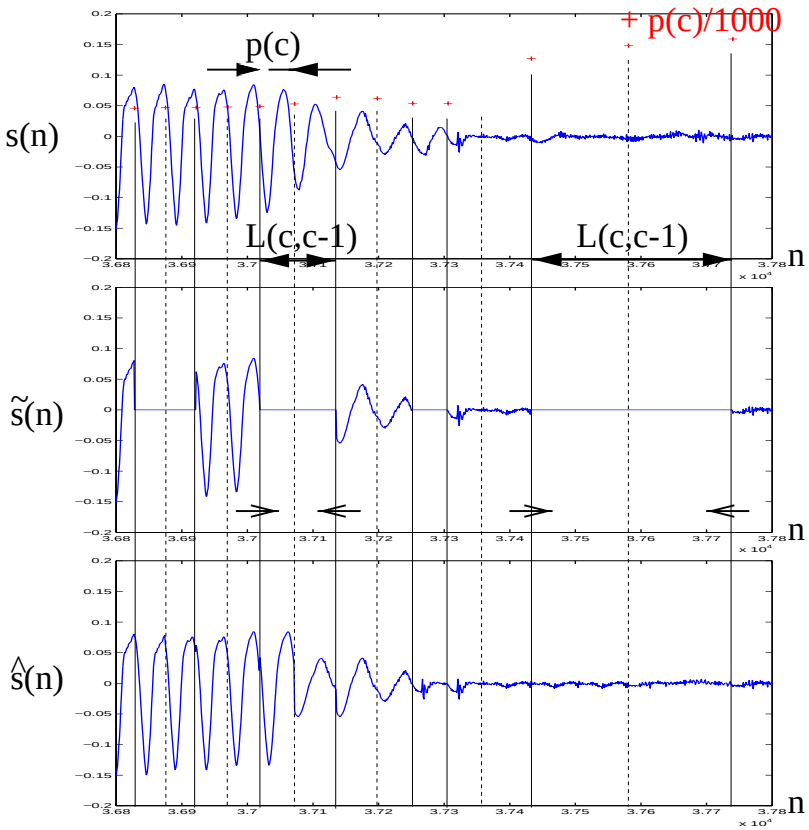


Figure 5.10: Concealment of a distorted signal ($ulp = 0.5$, $clp = 0$)

the sender, the receiver can assume that the chunks of a lost packet resemble the adjacent chunks. The adjacent chunks (c_{12} and c_{31} in Fig. 5.9) are re-sampled in the time domain by a factor of $k = c_{12}/c_{21}$ and $k = c_{31}/c_{22}$ for the left and right adjacent packet respectively. This is done to match the lost chunk sizes, which are given by the packet length and the transmitted intra packet boundary[3]. Re-sampling is done using a linear interpolator (as in [VA89]). The conversion factor k is linearly varied throughout the signal segment. This enables a replacement signal with a correct phase, thus avoiding discontinuities in the concealed signal leading to distortions, while maintaining the original pitch frequency at both edges of the replacement packet.

Then these chunks are copied into the output buffer as a replacement for the lost packet. No time-scale adjustment ([SSYG96]) is necessary as the chunk sizes are small. Because the sizes of the lost and the adjacent chunk most probably only differ slightly for either voiced or unvoiced speech (and thus the respective spectra), no *specific* distortion caused by the operation can be observed. We tested also to adjust the amplitudes of the replacement chunks according to the amplitude of the original ones. An informal subjective evaluation showed however that the additional overhead (computation and transmission of the energy contained in a chunk) does

[3]Further study is needed, how good an estimation of the intra-packet boundaries would perform.

packet			expansion (exp.)
left	lost	right	compression (comp.)
$v\|u_a$	$u_L u$		$u_a \ll u_L \rightarrow$ exp.
	uu_L	$u_a\|v$	$u_a \ll u_L \rightarrow$ exp.
uu_a	$u_L\|v$		$u_a \gg u_L \rightarrow$ comp.
	$v\|u_L$	$u_a u$	$u_a \gg u_L \rightarrow$ comp.
	$u(u\|v)_L$	$v_a v$	$v_a \ll (u\|v)_L \rightarrow$ exp.
$u(u\|v)_a$	$v_L v$		$(u\|v)_a \gg v_L \rightarrow$ comp.

Table 5.1: Concealment of/with packets containing speech transitions leading to high expansion or compression

not justify the gain in speech quality (yet a gain in per-packet SNR was clearly visible). Fig. 5.10 shows the concealment operation in the time domain, where $L(c, c-1)$ designates the length of a packet consisting of two chunks c and $c-1$.

Transitions in the signal might lead to extreme expansion/compression operations. Table 5.1 lists the possible cases. v_a, u_a are voiced/unvoiced *available* chunks, v_L, u_L are voiced/unvoiced *lost* chunks which are relevant for the case. A $u(u|v)$ packet is a packet where the second chunk contains an unvoiced/voiced transition that was not recognized by the sender algorithm (see section 5.1.2.1). To avoid extreme expansion/compression an upper bound f_{max} for the re-sampling has been introduced (Fig. 5.9): $|1-k| \overset{!}{<} f_{max}$. We used $f_{max} = 50\%$. If the bound is exceeded when compressing, adjacent samples of the relevant length are taken and inserted in the gap ("segment copy" in Fig. 5.9). An audible discontinuity which might occur can be avoided by overlap-adding the concealment chunk with the adjacent ones. High expansions are avoided by repeating a chunk until the necessary length is achieved ("periodic extrapolation" in Fig. 5.9) and then again overlap-adding it.

5.1.3 Results

Figure 5.11 shows the measurement setup for the evaluation of AP/C (cf. Figure 4.18). We employ a Gilbert model to simulate losses and compare the impact of losses on a flow with and without concealment at the receiver (note that in both cases we use the adaptive packetization at the sender).

5.1.3.1 Objective quality assessment

Conventional objective measurements (like an SNR) are not appropriate for AP/C, because AP/C does not aim at mathematically exact reconstruction. However, the adaptive packetization and subsequent re-sampling should perform somewhat better than silence substitution concerning mathematical correctness. Measured SNR values for AP/C are in fact always above those for the distorted (silence substitution) signal. This confirms our conjecture, however we employ the EMBSD measure (see section 4.2.1.2) to assess the user-level performance.

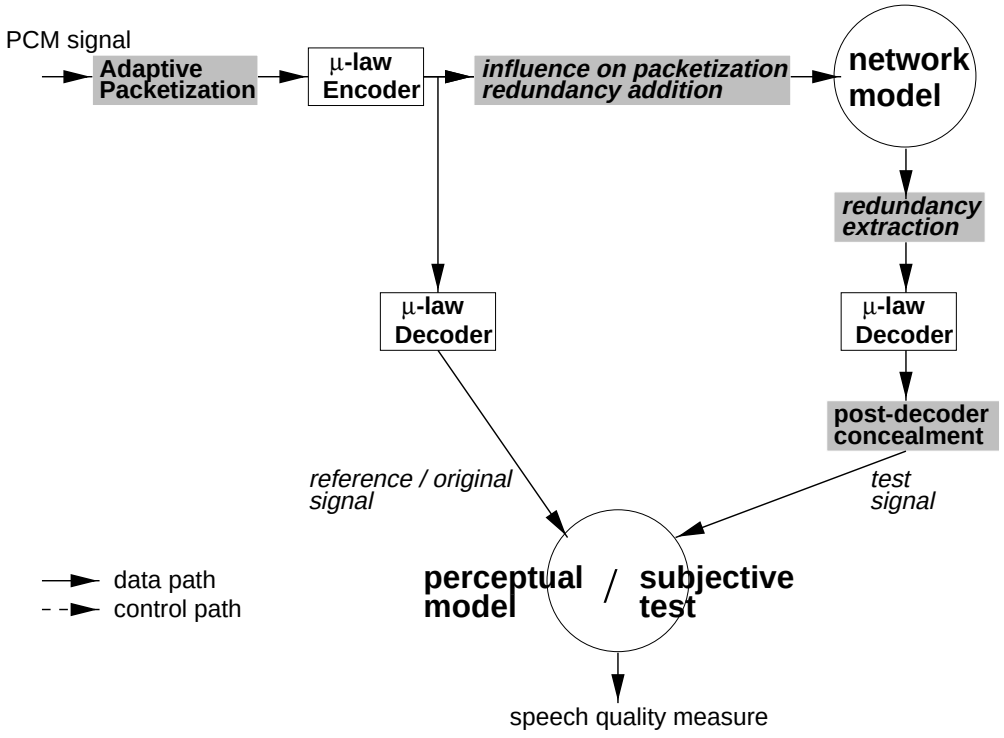


Figure 5.11: Components of the AP/C loss recovery measurement setup.

We vary the parameters p_{01} and clp of the Gilbert model (sections 2.2.1.1 and 4.1.4). For each loss condition (p_{01}/clp pair) the result of the objective quality measures for several loss patterns as well as the resulting values for the ulp for the patterns are averaged. So we have a reliable indication for the performance of AP/C under a certain network loss condition. Note that every measurement point in the figures designated by the different symbols (circle, diamond, etc.) corresponds to a p_{01} value ($p_{01} \in [0.05, 0.15, 0.25, 0.35, 0.45]$, p_{01} is increasing with increasing ulp: Eq. 2.6).

Figure 5.12 shows the case for silence substitution, i.e. an AP flow without loss concealment enabled. The resulting speech quality is insensitive to the loss distribution parameter (clp). The results are even slightly decreasing for an increasing ulp, pointing to a significant variability of the results. In Figure 5.13 the results for AP/C are depicted. When the loss correlation (clp) is low, AP/C provides a significant performance improvement over the silence substitution case. The relative improvement with regard to silence substitution increases with increasing loss (ulp). For higher clp values AP/C approaches the silence substitution case and shows similar performance for $clp \gtrapprox 0.3$. For very high ulp and clp values the performance is worse than for silence substitution. We suspect that this is due to a higher probability for AP/C that very large gaps occur (due to the long packetization time for unvoiced speech). Figures 5.12 and 5.13 respectively also contain a curve showing the performance under the assumption of random losses (Bernoulli model, $ulp = clp$).

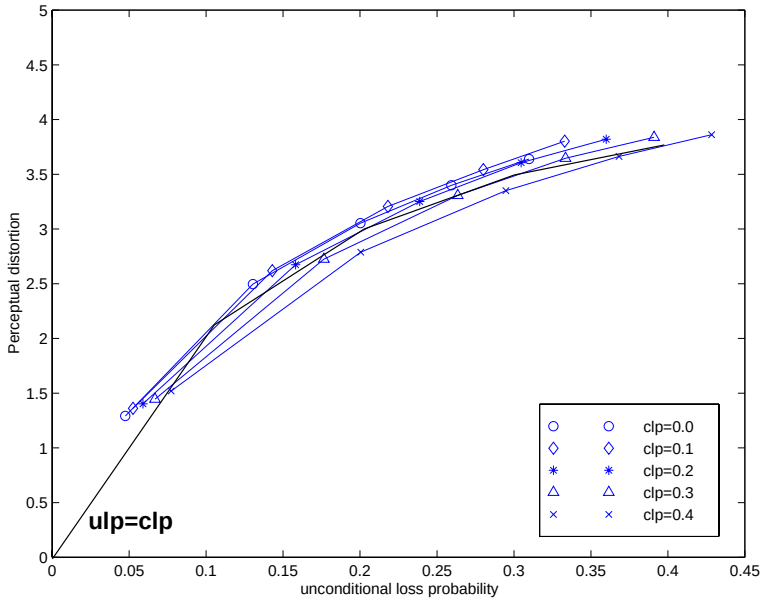


Figure 5.12: Perceptual Distortion (EMBSD) for silence substitution

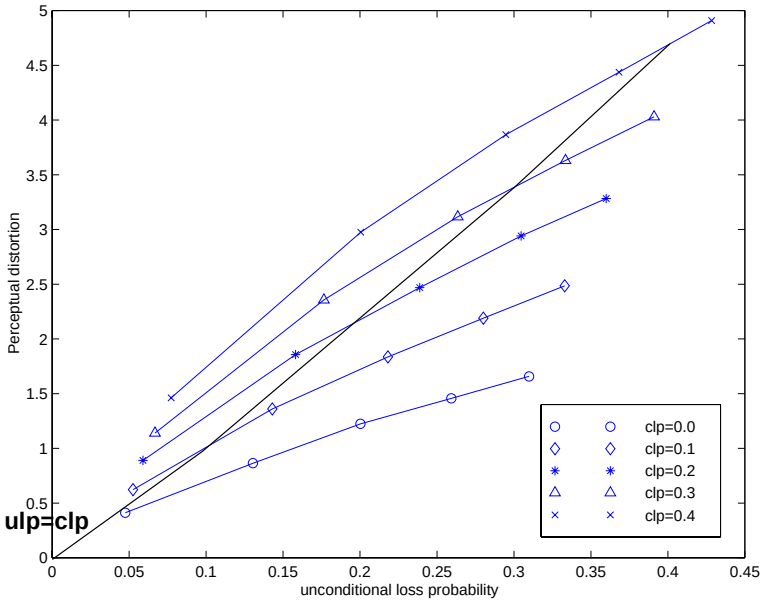


Figure 5.13: Perceptual Distortion (EMBSD) for AP/C

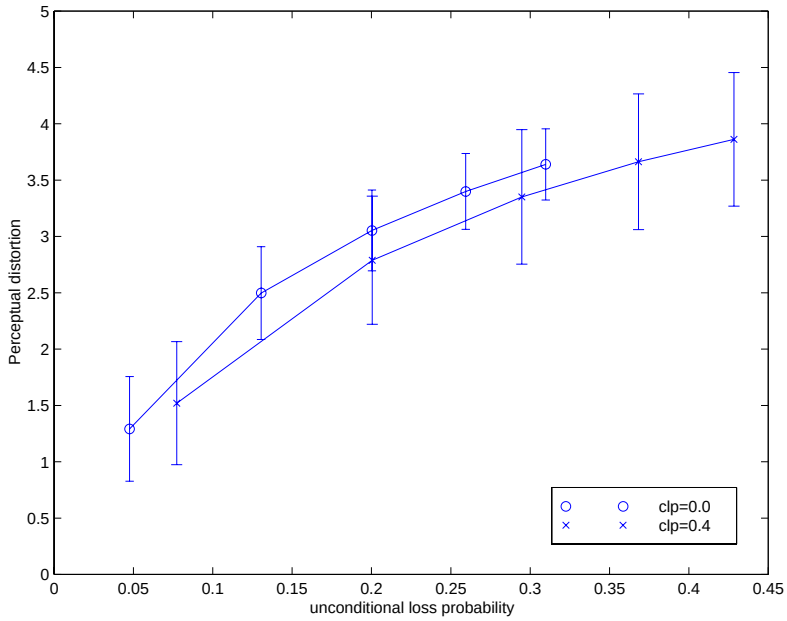


Figure 5.14: Variability of the perceptual distortion (EMBSD) for silence substitution

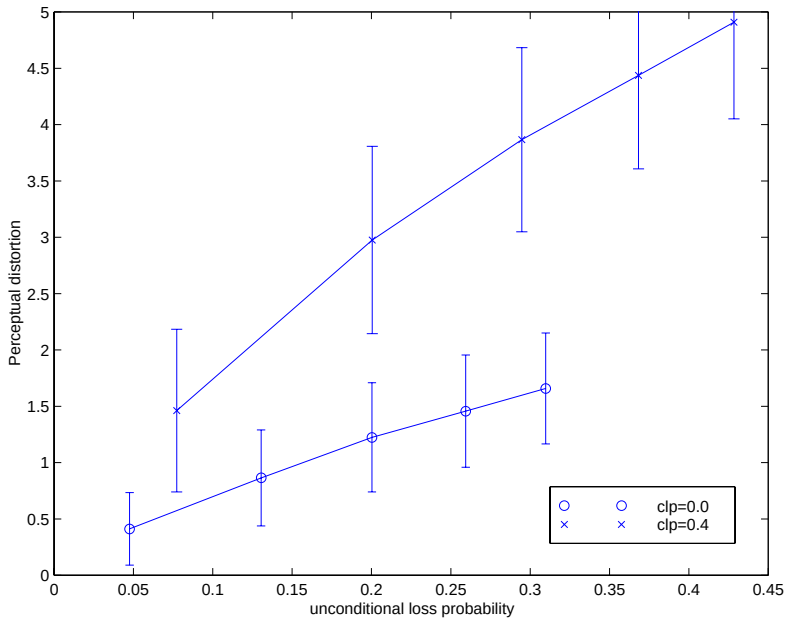


Figure 5.15: Variability of the perceptual distortion (EMBSD) for AP/C

As we found by visual inspection that the distributions of the perceptual distortion values for one loss condition seem to approximately follow a normal distribution we employ mean and standard deviation to describe the statistical variability of the measured values. Figures 5.14 and 5.15 present the perceptual distortion as in the previous figures but also give the standard deviation as error bars for the respective loss condition. While still clearly showing the improvement of AP/C, the figures show the increasing variability of the results with increasing loss correlation (clp), while the variability does not seem to change much with an increasing amount of loss (ulp). Thus, care with regard to the number of loss patterns on which the results are based must be taken when using objective speech quality measurement to assess the impact of loss correlation on user perception.

5.1.3.2 Subjective test

To validate the objective evaluation of AP/C to some extent, a subjective test was carried out. Test signals were the four signals (with different speakers) also used in the objective analysis of section 5.1.2 (PCM 16 bit linear, sampled at 8 kHz). AP/C is compared against silence substitution and also the simple receiver-based concealment algorithm "Pitch Waveform Replication" (PWR, cf. paragraph 3.1.3.2), which is the only one able to operate under very high loss probabilities (considering isolated losses). With PWR, one pitch period found in the packet preceding the missing one is repeated throughout the loss gap.

Primary goal of the test was to assess the performance improvement in the presence of numerous, yet isolated losses, as the objective quality assessment has shown that AP/C can perform well when the clp is low. The parameter set for the subjective test is therefore $clp = 0$ and $ulp \in [0, 0.2, 0.3, 0.5]$. While it would be interesting to validate the objective results also for other clp values, the necessary number of test conditions (section 4.2.2) for such an evaluation are prohibitive. Thirteen non-expert listeners evaluated the overall quality of 40 test conditions (4 speakers $\times$ (3 algorithms $\times$ 3 loss probabilities + original) on a five-category scale (Mean Opinion Score: MOS, see section 4.2.2). Before testing started, an "Anchoring" procedure took place, where the quality range (Original = 5, "Worst Case" (WC) signal[4] = 1) was introduced.

Figures 5.16/5.17 show the MOS values for the four different speakers (male low/high, female low/high). For the loss probability values we give the unconditional loss probability, however based on lost samples rather than packets. This allows a slightly better comparison between the results for speakers with different pitches, as we deal with variable size packets. It can be seen the results for all speakers confirm that AP/C leads to a significant enhancement in speech quality compared to the "silence substitution" case, which is maintained also for a higher loss probability. However for speakers (female) with higher pitch frequencies, the relative performance (distance between "silence substitution" and "AP/C") decreases. A reason for this is the chosen start offset point T_{min} (= 30 samples) of the auto correlation computation,

[4]In this test we used the unconcealed 50% loss signal.

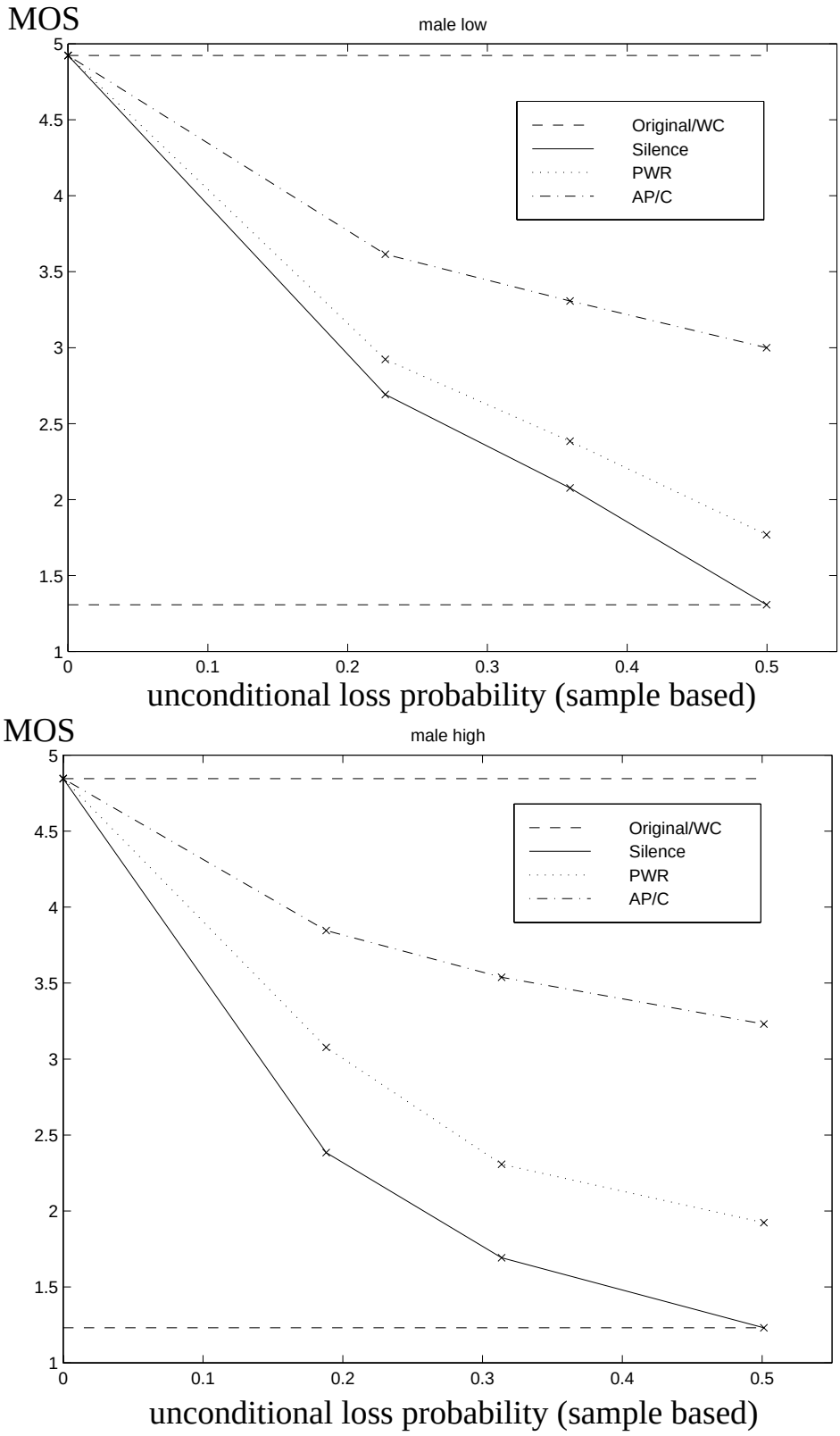


Figure 5.16: MOS as a function of sample loss probability for speakers 'male low' and 'male high'

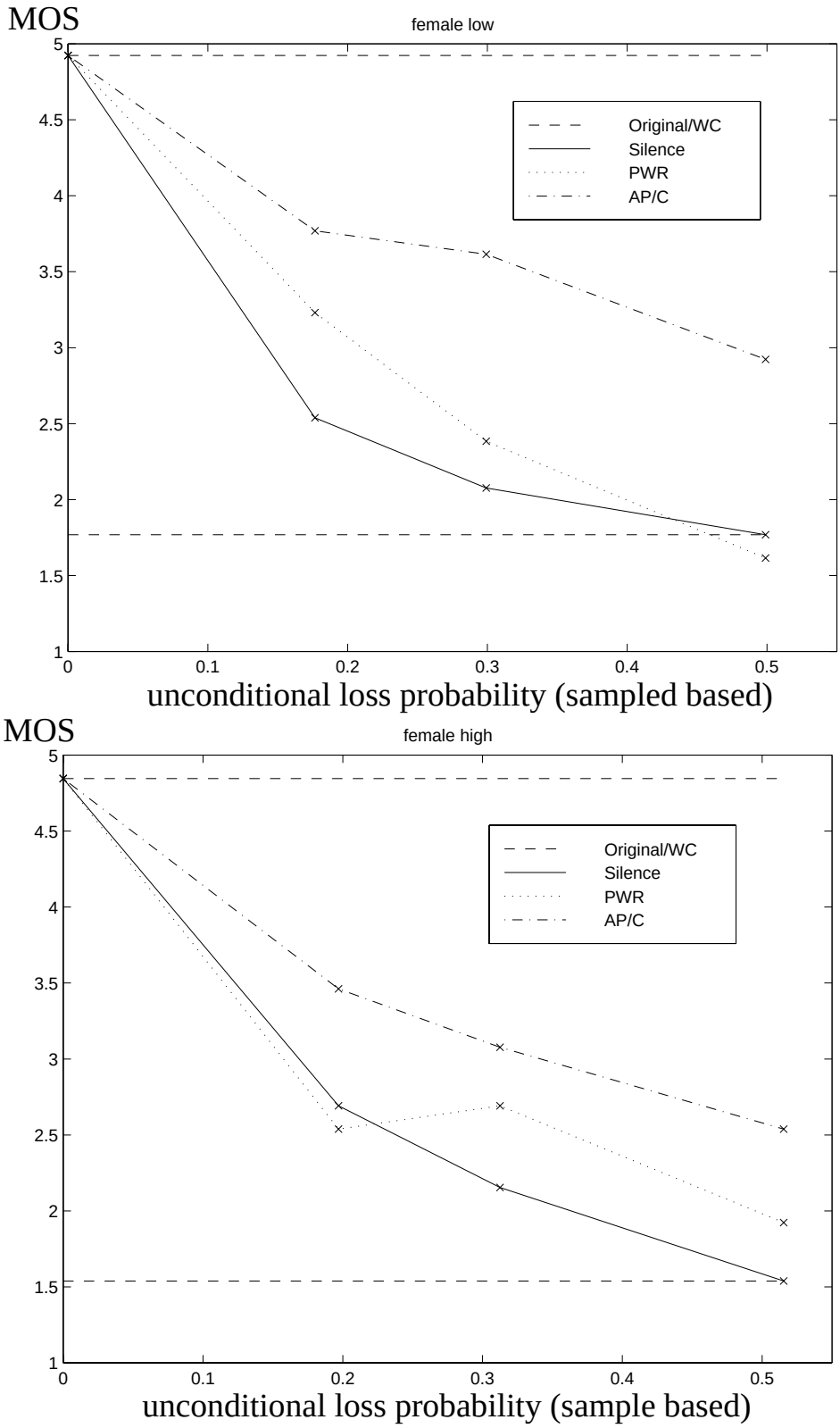


Figure 5.17: MOS as a function of sample loss probability for speakers 'female low' and 'female high'

which constitutes a lower bound on the chunk/packet size to avoid excessive packet header overhead, but also limits the accurateness of the periodicity measurement (note the small distance between the peak of the packet size distribution and the lower bound in Fig. 5.7 for "female high"). Additionally, female speakers receive relatively high MOS values for the worst case signal (> 1.5). This is due to the adaptive packetization: a higher number of shorter gaps is introduced (compared to fixed size packetization with the same loss probability) which are less perceptible. The PWR algorithm performs well for loss probabilities of up to about 20% (cf. [SSYG96]), however, speech quality drops significantly for higher loss probabilities, as the specific distortions introduced by that algorithm become significant. Standard deviations of MOS values for all but two of the forty test conditions are below 1.

5.1.4 Discussion

The additional delay introduced by the AP/C scheme consists of

- time interval corresponding to the length of the buffered speech segment needed for the sender processing (auto-correlation computation) of the second chunk minus the actual size of the second chunk (as this belongs to the "conventional" packetization interval to create a packet) : $T_{max} \leq d_S \leq 2T_{max} - T_{min}$.
- time corresponding to one packet length after a loss was detected at the receiver ($T_{min} \leq d_R \leq 2T_{max}$)
- time needed for computations d_C

The computational complexity is slightly lower at the sender and significantly lower at the receiver when compared to the complexity of a simple voice encoder/decoder (like an LPC-10 codec). This is because only a subset of the operations (auto-correlation, sample rate conversion) have to be performed (thus $d_C \ll d_S + d_R$). This makes the scheme well suited for multicast environments with low-end receivers. As shown in Fig. 5.8, the additional packet header overhead is even for the highest pitch voice below 10%, which is comparable to adding a very low bit-rate additional source coding to reconstruct isolated losses ([HSHW95]). Because of the dependency on the pitch period, AP/C is aimed at speech transmission only.

5.1.5 Implementation of AP/C and FEC into an Internet audio tool

The AP/C scheme , as it has been described in the sections above, has been implemented into the NeVoT (Network Voice Terminal, [Col98]) audio tool. In addition to AP/C the modified version of NeVoT 3.35 comprises the following functions:

- RFC 2198 ("Redundant RTP payload for audio data")-conformant redundancy transmission ([PKH+97])

- a generic receiver loss recovery layer within NeVoT:
  Incoming packets are sorted according to their sequence number; losses are detected and the needed redundant data are extracted from arriving packets and copies are buffered. Just before the play-out, a detection of packets still missing takes place and singular losses are concealed using AP/C.

  The loss recovery layer is configurable in two ways:

  1. The loss recovery performance is directly coupled to the play-out delay, thus enabling control over the loss versus delay tradeoff. If a very low play-out delay is selected, packets carrying a redundant payload might arrive too late, such that the redundant payload cannot be used any more. Also,too few data for a successful loss concealment might be available in the loss recovery buffers. For a discussion of the interaction of FEC and the delay adaptation algorithm see the paper by Rosenberg et al. ([RQS00]).
  2. The loss concealment and the amount of used redundant data for play-out can be adjusted. This allows the receiver to assess the quality differences by switching between different redundancy levels and enabling concealment additionally to the selected amount of redundancy.

The number of redundancy layers which are received is detected automatically. The button list on the right hand side controls only the play-out of the received redundancy. As default the maximum amount of redundancy (2 layers) received is played out.

Any combination of sample-based codecs (PCM, ADPCM), loss concealment and redundancy can be used. The play-out delay also influencing the loss recovery performance as described above is adjusted with the conventional delay sliders contained in the per-session configuration window of the MInT conferencing environment ([SS98b]).

5.1.5.1 Configuration

"Loss Control" window

The Loss Control window can be accessed via the *Settings* pull-down menu of the NeVoT window. A snapshot of the window with its default parameter settings is shown in Figure 5.18.

Sender redundancy The upper part of the window entitled *Sender redundancy* gives control over the sender part of the loss control functions.

It contains a checkbutton to switch *Adaptive Packetization (AP)* on and off. AP uses the header extension to transmit a minimal amount of data to support the receiver concealment operation, therefore it should only be enabled if the receiver tool is able to correctly parse the header extension (see below).

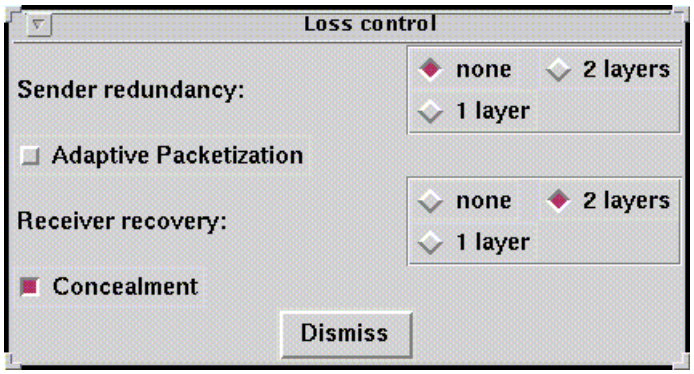


Figure 5.18: Loss Control window

On the right hand side, there is a button list with which the RFC2198-conformant *redundancy* can be enabled for either AP or fixed-size packet flows: either 100% (*1 layer*) or 200% (*2 layers*) redundancy using the same coder are available.

Receiver recovery The lower part of the window entitled *Receiver recovery* contains the receiver control options corresponding to the sender functionality described above.

If an AP data flow is received, the checkbutton *Concealment* enables the concealment of single packets still missing after redundancy (if available) has been extracted.

5.1.5.2 User-level performance

Using an audio tool rather than a dedicated measurement tool to run automated measurements is difficult as it is necessary to open the audio device remotely (which can only be done by the superuser) to be able to send and receive (even when sending audio from a file or receiving data without play-out, the audio device is needed for timing). A better possibility to evaluate the actual implementation is network emulation. This allows to combine the measurements done in section 4.1.8 with the implementation. A disadvantage of this procedure is that the measured traces only record congestion loss. Therefore we cannot assess the impact of loss due to late packets. However the amount of late packets is known to be low typically ([San95]) and that information would mainly needed for the design of play-out delay adaptation algorithms, not for the evaluation of loss recovery algorithms.

Figure 5.19 shows the measurement setup used to test the implementation. It consists of the described NeVoT sender and receiver implementations running on different hosts which are attached to a near loss free real network (such as a local area network). The sender implementation contains a packet dropper below the RTP protocol processing. The dropper is driven by a tracefile which contains the loss indicator function (Eq. 4.1). Note that the dropper can additionally be configured via the user interface to drop packets according to Gilbert model parameters.

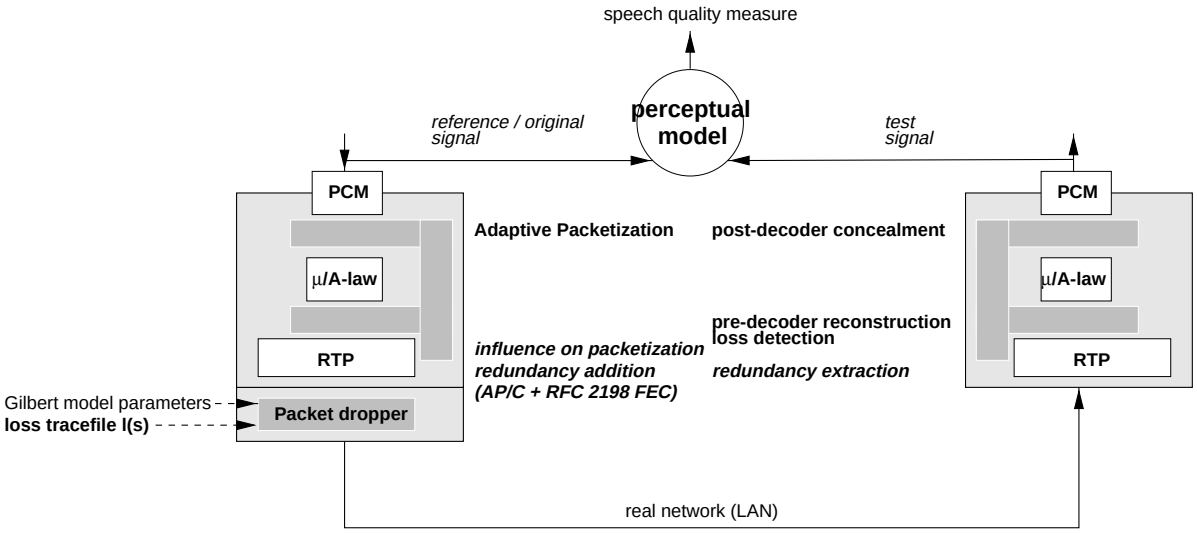


Figure 5.19: Measurement of the AP/C+FEC implementation using a network emulation configuration

FEC	*without concealment*	*with concealment*
none	0.7321	0.6643
1 layer	0.2737	0.1673
2 layers	0.1007	0.0970

Table 5.2: Auditory distance (MNB) results for the network emulation setup

The usual NeVoT input/output capabilities (live voice, voice data file) can now be used over a configuration which emulates the loss behavior recorded in the tracefile. Finally, after cutting the recorded data to the right length (the actual start point has to be found), objective quality measures can be used to evaluate the perceptual quality.

Table 5.2 shows MNB results with and without AP/C loss concealment for a trace where $ulp \approx 0.2$ and $clp \approx 0.3$ (we verified that a Gilbert model characterization is valid for the trace; cf. section 4.1.8). As speech material we have used the 'male high' sample also used in the previous subjective test (section 5.1.3.2) and concatenated several copies of the sample until the trace duration ($1min$) has been reached. We observe that without any FEC, AP/C yields only a slightly better perceptual quality, because there are already a significant number of burst losses which cannot be concealed. When one layer of FEC is added, the perceptual quality for both cases is increased. AP/C is now able to conceal a higher percentage of the losses (which are unrecoverable by the FEC) and thus the relative performance to the case without concealment is increased. If a second layer of FEC is added, the performance for both cases is virtually identical as nearly all losses can be repaired

with the FEC. Note however that the gain in perceptual quality by adding another FEC layer is small, as the ulp is already very low for the one-layer case (Eq. 3.2). So the reconstruction with one layer of FEC in connection with loss concealment constitutes a reasonable tradeoff between the achievable quality and the additionally necessary redundant data.

5.1.5.3 Conclusions

The auto-correlation can be calculated in the time domain according to its definition in equation 5.1. Another way to compute the auto-correlation is to compute the discrete Fourier transform $S(e^{j\omega T})$ of the input segment and then use this result to calculate the discrete Fourier transform of the auto correlation $r_{ss}(k)$ (i.e., $R_{ss}(e^{j\omega T})$). The auto-correlation $r_{ss}(k)$ is the inverse Fourier transform of $R_{ss}(e^{j\omega T})$:

$$r_{ss}(k) = \sum_n s(n)s(n+k) = s(k) * s(-k) \leftrightarrow S(e^{j\omega T})S(e^{-j\omega T})$$

$$\Rightarrow R_{ss}(e^{j\omega T}) = S(e^{j\omega T})S(-e^{j\omega T})$$

This method is found to be faster and consumes less CPU resources than the first one: computation in the time domain of K points of the auto-correlation function for an N point window requires on the order of $K \times N$ multiplications and additions while computation of the auto-correlation function by the second method requires on the order of $Nlog_2K$ multiplications and additions ([RS78]). In our work, we use the C routines of FFTW to compute the Fast Fourier transform. Besides its very good performance, FFTW also supports Fourier transform of any size and has a very good documentation for installation and functional description ([MIT99]).

Backwards compatibility to existing audio tools is ensured, as long as the tools can receive properly variable length PCM packets (and then mix them into their output buffer). Typically this should be the case as also with fixed-size packets the packet size might change during a session while not leading to a perceivable interruption or degradation of the output signal. Additionally, correct treatment of the RTP header extension is certainly needed. Finally specific delay adaptation algorithms might need to be modified, however using a measured mean of past packet sizes of a flow should yield the same performance as for fixed-size packets. We informally tested the tools RAT (Version 3.0.31 [UCL98]), vat (Version 4.0b2, [LBN98]) and FreePhone (Version 3.7b1, [INR00]) on Sun Solaris platforms. We found that only FreePhone was able to decode the AP stream.

Interoperability with other implementations of the RFC 2198 FEC scheme (RAT and FreePhone) has been tested successfully.

5.2 Frame-based codecs

Considering the backward-adaptive coding schemes of the G.723.1 and G.729 source coders (section 2.1.3.2), packet loss results in loss of synchronization between the encoder and the decoder. Thus, degradations of the output speech signal do not

only occur during the time period represented by the lost packet, but also propagate into following segments of the speech signal until the decoder is resynchronized with the encoder. To alleviate this problem, both G.723.1 and G.729 decoders contain an internal (codec-specific) loss concealment algorithm. In this chapter we first discuss if and how the AP/C scheme introduced in chapter 5.1 is applicable. Then we present our approach towards an efficient end-to-end protection scheme for frame-based codecs. Therefore we analyze the loss resilience of a particular frame-based codec (G.729) and design our proposed scheme accordingly.

5.2.1 AP/C for frame-based codecs

Two properties of modern, frame-based speech coders do not allow a straightforward application of AP/C ([San98a]):

- synchronization of encoder and decoder (synchronization is lost during a packet loss gap, thus the decoding is worse after the gap due to previous decoder state loss, especially for backward-adaptive codecs [Clu98])
- operation on (small) *fixed size* speech frames (e.g. $F = 10ms$ for G.729 [Uni96a], $F = 30ms$ for GSM [Deg96] and G.723.1 [Uni96c], where F is the time interval corresponding to a frame)

The first problem can only be alleviated by either trading higher loss-resilience against higher bit-rate (i.e. using a non-adaptive codec like PCM) or, as a compromise, using a hybrid codec (waveform/parametric), where the impact of a packet loss to subsequently decoded speech is less severe (see section 5.2.3 with regard to the G.729 codec).

The second issue should be tackled by a close integration of coding and packetization as well as decoding and concealment (+FEC) functions (section 5.2.2). However, to allow the operation together with existing codecs, we evaluate a simple fragmentation scheme.

Fig. 5.20 shows the packetization, when speech boundaries found by the AP algorithm are used to associate frames of length F to the actual packets sent over the network. As AP packets overlap the frame boundaries, a significant amount of redundant data as well as additional alignment information (s_i) need to be transmitted (yet redundant data can be used in a possible concealment operation e.g. by overlap-adding it to the replacement signal). To allow analysis, we assume a constant AP packet size of $l = kF + n$, k being a positive integer.

The fragmentation data "overhead" associated with packet i can then be written as follows:

$$o_f = \left(in - \left\lfloor \frac{in}{F} \right\rfloor F \right) + \left(\left\lceil \frac{(i+1)n}{F} \right\rceil F - (i+1)n \right)$$

For a sequence of N packets, this results in:

$$O_f = F \sum_{i=0}^{N-1} \left(\left\lceil \frac{(i+1)n}{F} \right\rceil - \left\lfloor \frac{in}{F} \right\rfloor \right) - nN$$

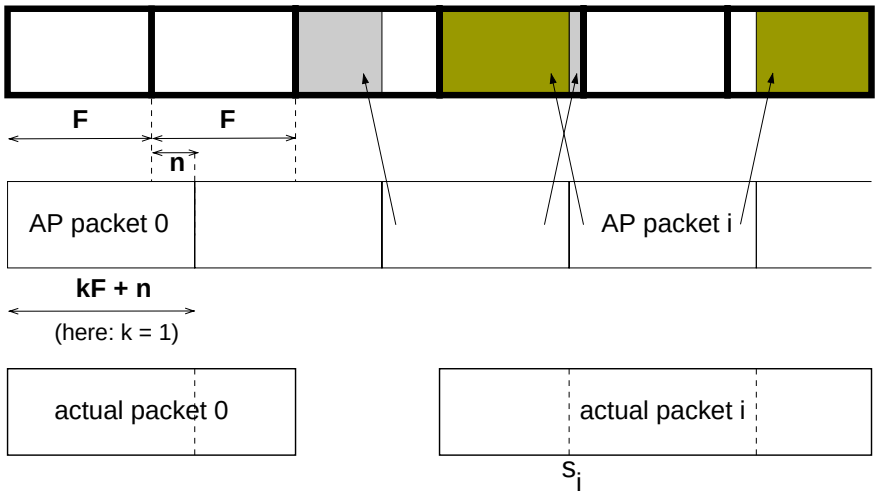


Figure 5.20: Packetization of a framed signal

Speaker	p_v [samples]	O'_f [%]	relative overhead (measured) [%]
male low	79.20	50.50	48.86
male high	67.05	59.65	58.36
female low	57.74	69.27	64.40
female high	49.88	80.20	76.36

Table 5.3: Relative fragmentation overhead for four different speakers (mean pitch period: p_v) for $F = 10ms$

With $F \mathrm{mod} n = 0$, we have $O_f = N(F - n)$. Assuming $n \ll F$, $O'_f = O_f/(2p_vN)$ gives an indication of the relative fragmentation overhead which can be expected for different speakers/ranges of packet sizes (p_v being the mean pitch period). Table 5.3 compares those values to measurement results. The fragmentation scheme results in an increase e.g. for the G.729 codec from $8kbit/s$ to $12 - 14.4kbit/s$ (in terms of payload). Table 5.3 also shows that the mean value p_v of the chunks classified as voiced, can be used as an estimate for an adaptive packetization "equivalent" packet size (cf. Fig. 5.6).

In summary, we can say that the above approach is not satisfying, because the the problem of de-synchronization of encoder and decoder cannot be addressed. Moreover, when frames is lost, the decoder already might apply some concealment algorithm using its internal state information from the last good frames. Due to the lack of this internal state information, a PCM-level concealment over a codec-level concealment will probably not much improve the speech quality (see also the discussion in section 3.1.3.2). Therefore in the following sections we will explore a QoS enhancement scheme which is closely tied to the performance of the internal loss concealment algorithm of a frame-based codec.

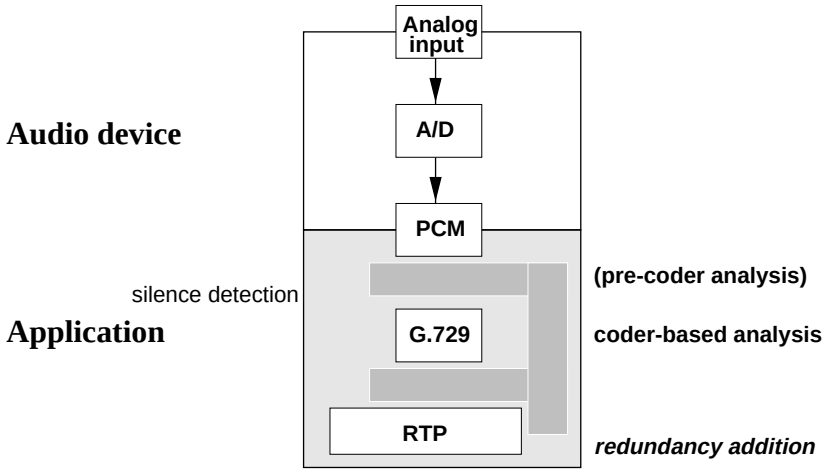


Figure 5.21: Structure of an SPB-FEC enhanced audio tool (sender)

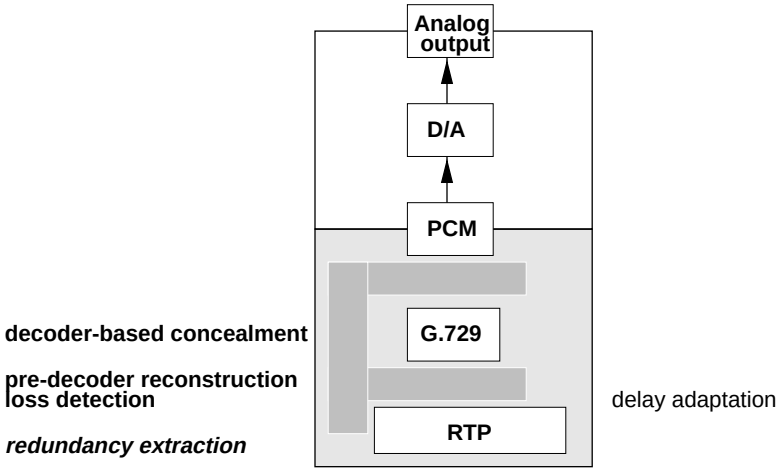


Figure 5.22: Structure of an SPB-FEC enhanced audio tool (receiver)

5.2.2 Approach

The previous section has shown that frame-based codecs cannot be treated similarly to sample-based codecs with regard to loss recovery. As the speech redundancy has been intensively exploited by the coding scheme (see introduction to chapter 5.1), additional redundant data needs to be added to the voice stream for loss recovery. Using the generic structure of an audiotool introduced in section 3.1 (Figures 3.1 and 3.2) our approach can be motivated as follows:

We adopt the approach of using a speech encoder (G.729) as the analysis module (source-coded FEC: section 3.1.2.2, Figure 3.1) with the following properties (components are shown in Figures 5.21 and 5.22):

- Information available at the encoder which can be used for the redundancy / loss recovery is exploited.

- No generic concealment (see section 3.1.3) schemes are employed, as codec-specific concealment is already implemented in the decoder (section 5.2.1).
- Only one source coder for both the main and the redundant payload is used.
- The amount of redundancy can be adjusted by the analysis module taking into account the decoder concealment process.

We only use one source coder to reduce the overall computational complexity. Additionally (if redundant data of a packet is coded with different audio encodings and "piggy-backing" on the following packets is used), when an important frame is lost, all decoders suffer loss of synchronization and deliver decoded speech signals with bad quality (as described in section 3.1.2.2). The key difference to other FEC approaches is that we aim to take the "concealability" of the signal at the receiver into account. Therefore in the next section we analyze the concealment behavior of a particular codec in detail.

5.2.3 G.729 frame loss concealment

In section 2.1.3.2 we have described the operation of the G.729 encoder and decoder. Furthermore the internal loss concealment scheme of the G.729 has been introduced in section 3.1.3.3. Here we now want to explore the impact of frame loss at different positions (voiced/unvoiced areas) within the speech signal.

In [Ros97a], Rosenberg investigated the issues of error resilience and recovery and measured the resynchronization time of the G.729 decoder after a frame loss. He pointed out that the energy of the error signal increases considerably and the Mean Opinion Score (MOS) of subjective tests decreases significantly when the number of consecutive lost frames increases from one to two, and gradually from there. He drew the conclusion that a single lost frame can be concealed well by the G.729 decoder but not more. In this section, we take a further step by attempting to answer the question: how does the speech quality degrade and how does the error propagate when a number of consecutive voiced/unvoiced frames are lost ?

The first experiment is to measure the resynchronization time of the decoder after k consecutive frames are lost. The G.729 decoder is said to have resynchronized with the G.729 encoder when the energy of the error signal falls below one percent of the energy of the decoded signal without frame loss (this is equivalent to a signal-to-noise ratio (SNR, Eq. 4.8, p. 80) threshold of $20dB$). The error signal energy (and thus the SNR) is computed on a per-frame basis (see section 4.2.1.1, eq. 4.9 with L being the frame size and $e(n)$ being the difference signal between the decoded signal with and without frame loss).

Figure 5.23 shows the resynchronization time (expressed in the number of frames needed until the threshold is exceeded) plotted against the position of the loss for different values of k. The speech sample is produced by a male speaker where an unvoiced/voiced (uv) transition occurs in the eighth frame.

The second experiment consists of measuring the energy of the error signal over N frames after k consecutive frames are lost. The position where the frame loss

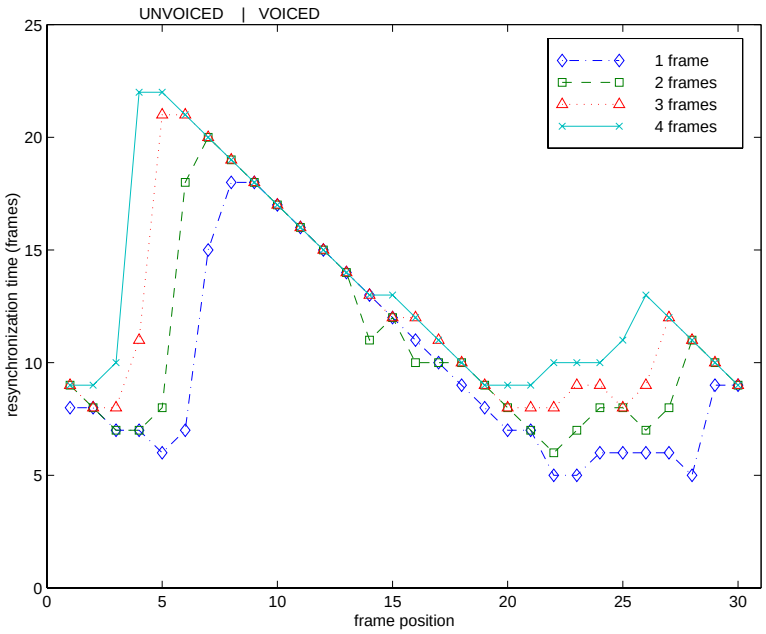


Figure 5.23: Resynchronization time (in frames) of the G.729 decoder after the loss of k consecutive frames ($k \in [1, 4]$) as a function of the frame position.

(burst) occurs is varied and then the average SNR (Eq. 4.9) over the N following frames is computed. In the experiment, we measure the average SNR over $N = 15$ consecutive frames after the frame loss which we consider an appropriate mean value for the resynchronization time. When measuring the average SNR over 10 and 20 consecutive frames after the frame loss similar results were obtained. (The first experiment has shown that the resynchronization time ranges from 5 to 22 frames depending on the position of the frame loss and the burst size. Previous experiments in [Ros97a] came to comparable results). Figure 5.24 shows the average SNR plotted against the frame loss position for the same speech sample.

Figure 5.23 and Figure 5.24 show that the position of a frame loss has a significant influence on the resulting signal degradation[5], while the degradation is not that sensitive to the length of the frame loss burst k. The loss of unvoiced frames seems to have a rather small impact on the signal degradation and the decoder recovers the state information fast thereafter. The loss of voiced frames causes a larger degradation of the speech signal and the decoder needs more time to re-synchronize with the sender. However, the loss of voiced frames at an unvoiced/voiced transition leads to a significant degradation of the signal. We have repeated the experiments for different male and female speakers and obtained similar results. Taking into account the used coding scheme, the above phenomenon could be explained as follows:

[5]While SNR measures often do not correlate well with subjective speech quality, the large differences in the SNR-threshold-based resynchronization time clearly point to a significant impact on subjective speech quality.

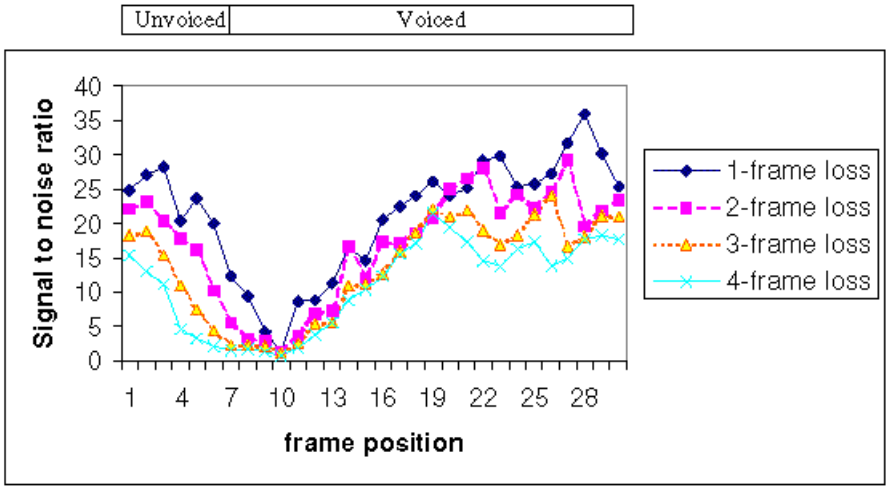


Figure 5.24: Mean SNR (dB) of the G.729-decoded speech signal after the loss of k consecutive frames ($k \in [1, 4]$).

- Because voiced sounds have a higher energy and are also more important to the speech quality than unvoiced sounds, the loss of voiced frames causes a larger degradation of speech quality than the loss of unvoiced frames.
- Due to the periodic property of voiced sounds, the decoder can conceal the loss of voiced frames well once it has obtained sufficient information on them.
- The decoder fails to conceal the loss of voiced frames at an unvoiced/voiced transition because it attempts to conceal the loss of voiced frames using the filter coefficients and the excitation for an unvoiced sound. Moreover, because the G.729 encoder uses a moving average filter to predict the values of the line spectral pairs and only transmits the difference between the real and predicted values, it takes a lot of time for the decoder to re-synchronize with the encoder once it has failed to build the appropriate linear prediction filter.

Figure 5.25 demonstrates the impact of frame loss at different positions on the decoded speech signal (in this case a male voice is used) in the time domain. We can clearly see that a frame loss at the beginning of the voiced signal causes a significant distortion of the decoded speech signal while the loss of other voiced and unvoiced frames is concealed rather well by the G.729 decoder. Using several different male and female speech data files, we obtained similar results.

5.2.4 Speech Property-Based Forward Error Correction (SPB-FEC)

The experiments we have carried out in the previous section have shown that the loss of frames at the beginning of a voiced signal causes a significant speech signal

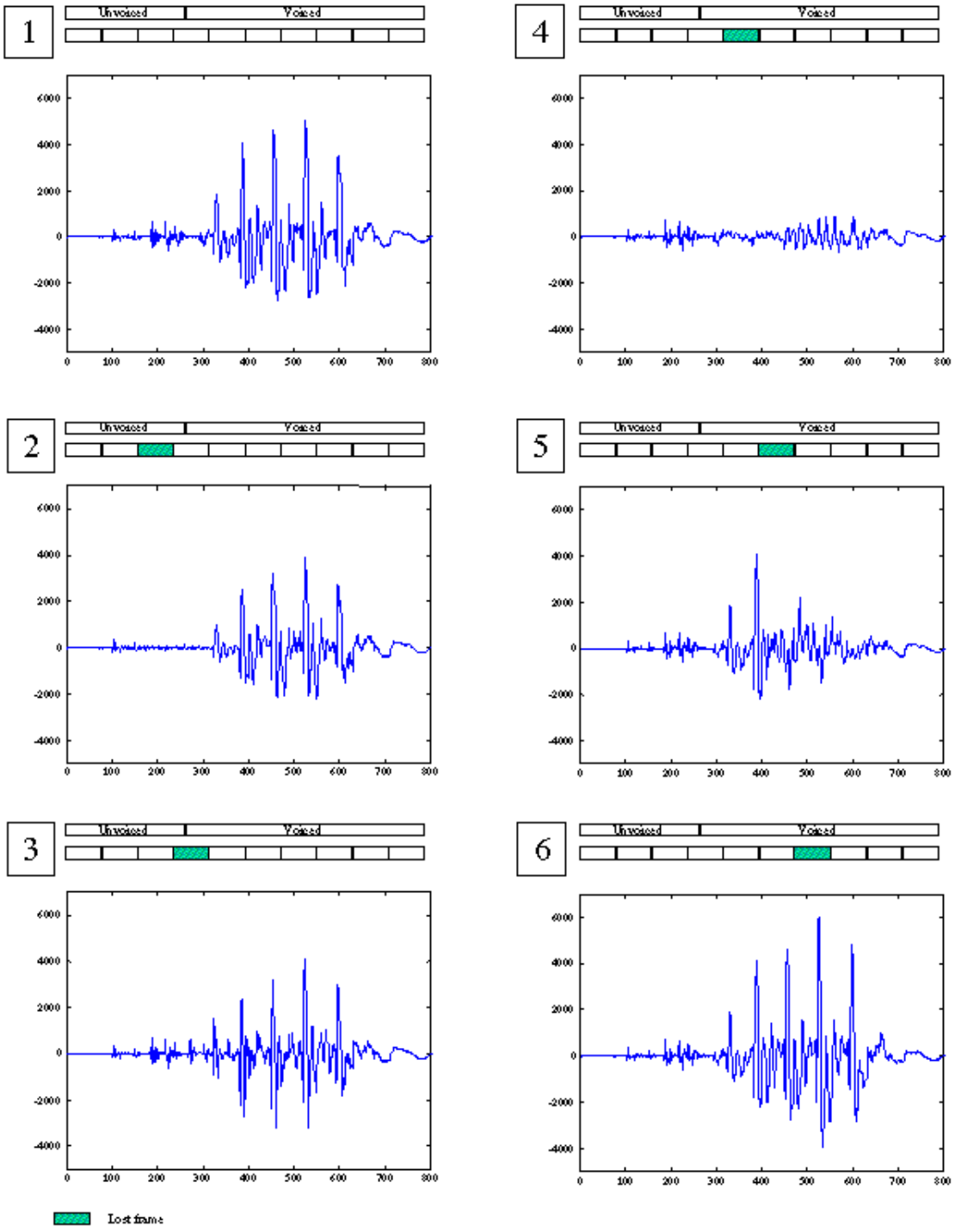


Figure 5.25: Decoded speech signal without and with frame loss at different positions

```
protect = 0
foreach (k frames)
    send(k frames)
    classify = analysis(k frames)
    if (protect > 0)
        if (classify == unvoiced)
            protect = 0
        else
            sendFEC(k frames)
            protect = protect−k
        endif
    else
        if (classify == uv_transition)
            sendFEC(k frames)
            protect = N − k
        endif
    endif
endfor
```

Figure 5.26: SPB-FEC pseudo code

degradation and a frame-based decoder like the G.729 decoder can conceal the loss of other voiced segments well once it has obtained sufficient information on the voiced signal. The loss of unvoiced frames is also concealed well by the decoder. This knowledge is exploited to develop a new FEC scheme called Speech Property-Based FEC (SPB-FEC, [SL00, Le99]). In contrast to other FEC schemes that equally distribute the amount of redundant data on all data packets, the SPB-FEC scheme concentrates the amount of redundant data on the frames essential to the speech quality and relies on the decoder's concealment for other frames.

Senders can either run a parallel algorithm for voiced/unvoiced decision or couple this algorithm with the encoder's operation. The first method is a generic approach (useful when coder-internal state cannot be accessed) and could use the time corresponding to the algorithmic delay of the G.729 encoder. However, generally, this method may duplicate functionality already available in the encoder and thus unnecessarily consume CPU resources. In our experiments we have chosen the second method. The voiced/unvoiced decision in G.729 is made in the decoder only however, so that the sender also has to run a decoder to decode its own frames and detect voiced/unvoiced transitions. This method is very simple however adds the G.729 decoding delay (about $7.5ms$, [Bla00]) at the sender side.

Figure 5.26 shows the simple algorithm written in a pseudo-code that is used to detect a uv transition and protect the voiced frames at the beginning of a voiced signal. In the algorithm, the procedure *analysis()* is used to classify a block of k

frames as voiced, unvoiced, or uv transition[6].

The procedures *send()* and *sendFEC()* are used to send a block of k frames (as a single packet) and redundant data to protect these frames. N is a pre-defined value and defines how many frames at the beginning of a voiced signal are to be protected. Our simulations have shown that the range from 10 to 20 are appropriate values for N (depending on the network loss condition). In the simulation presented in section 5.2.4.2, we choose $k = 2$, a typical value for interactive speech transmissions over the Internet ($20ms$ of audio data per packet). A larger number for k would help to reduce the relative overhead of the protocol header but also increases the buffer delay and makes sender classification and receiver concealment in case of packet loss (due to a large loss gap) more difficult.

5.2.4.1 Reference FEC schemes

In general, there are two methods to send redundant data: in a separate flow or "piggy-backed" on the following packets containing the main payload (section 3.1.2.2). While the first method has the advantage of backwards compatibility, we choose the second method for our simulation because of the lower protocol header and router processing overhead. We use two other FEC schemes as reference to evaluate the SPB-FEC: In the first FEC scheme, the two frames of the packet (n) are piggy-backed on the packet $(n+2)$ (we do not piggy-back the two frames of the packet (n) on the packet $(n+1)$ to mitigate the effect of packet burst loss, Eq. 3.1). This FEC scheme has a redundancy overhead of 100%. In the second FEC scheme, the four frames of the packet (n) and $(n+1)$ are XORed (p. 40) and the result is piggy-backed on the packet $(n+2)$. If the packet $(n+2)$ and one of the packets (n) or $(n+1)$ arrive at the receiver, the lost packet can be recovered. This FEC scheme has a redundancy overhead of 50%.

The speech property-based FEC scheme is similar to the reference FEC scheme 1. However, in the SPB-FEC scheme, only when an unvoiced/voiced transition is detected, the FEC mechanism is turned on to protect the voiced frames at the beginning of a voiced signal, resulting in a redundancy overhead of 41.9% (for the speech material used in the experiments below). Figure 5.27 illustrates the two reference FEC schemes.

5.2.4.2 Simulation description

We first simulate a network where voice data flows using packets containing two frames (i.e. $20ms$ speech segments) without any redundant data are transmitted. The network loss parameters p_{01} and p_{11} are varied in constant steps to obtain an impression on the sensitivity and expected range of the objective quality measurements' result values (Figure 5.28 shows the network loss rate (unconditional loss probability) associated with the pairs of p_{01} and clp (cf. section 4.1.4) in the first simulation step). The voice data flow with frame loss is decoded. The results are

[6]The voiced/unvoiced (vu) transition is unimportant in the algorithm and is classified as unvoiced.

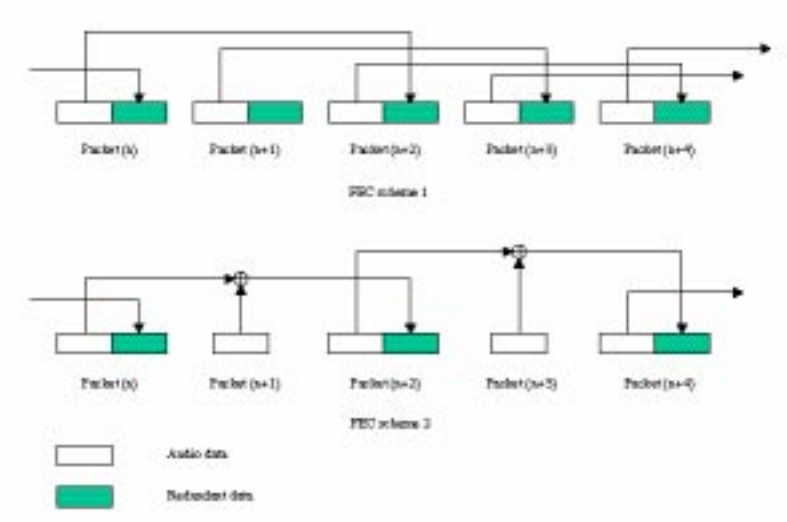


Figure 5.27: Two reference FEC schemes.

Network loss condition 1	Network loss condition 2	Network loss condition 3	Network loss condition 4	Network loss condition 5
$p_{01} = 0.05$ $clp = 0.2$	$p_{01} = 0.1$ $clp = 0.3$	$p_{01} = 0.15$ $clp = 0.4$	$p_{01} = 0.2$ $clp = 0.5$	$p_{01} = 0.25$ $clp = 0.6$
$ulp = 0.07$	$ulp = 0.125$	$ulp = 0.2$	$ulp = 0.29$	$ulp = 0.39$

Table 5.4: Parameter sets for different network loss conditions

then compared with the decoded speech signal without frame loss using the objective quality measures.

In the second step, the simulated network is applied to voice data flows using the SPB-FEC scheme, the two reference FEC schemes described in section 5.2.4.1, and a scheme without redundant data respectively. Every speech data packet contains two frames and possibly some redundant data depending on the respective FEC scheme. We use five (p_{01}, p_{11}) value pairs reflecting real network loss conditions (Table 5.4) measured in the Internet ([Bol93]). The FEC schemes are then used to recover the information contained in the lost packets to the largest extent possible. Figure 5.29 shows the application loss rate of the schemes with and without FEC, i.e. the loss rate seen by the G.729 decoder after FEC decode (if any) has been performed for the five network loss conditions. Obviously, the more redundant data is transmitted, the lower is the application loss rate.

Then, the voice data streams (possibly still with some frame losses) are decoded. These decoded speech signals and the decoded speech signal without frame loss are then evaluated by the objective quality measures to demonstrate the efficiency of the FEC schemes. The two simulation steps for the evaluation of the FEC schemes are illustrated in Figure 5.30.

For each pair of p_{01} and p_{11}, we use the same speech sample containing different

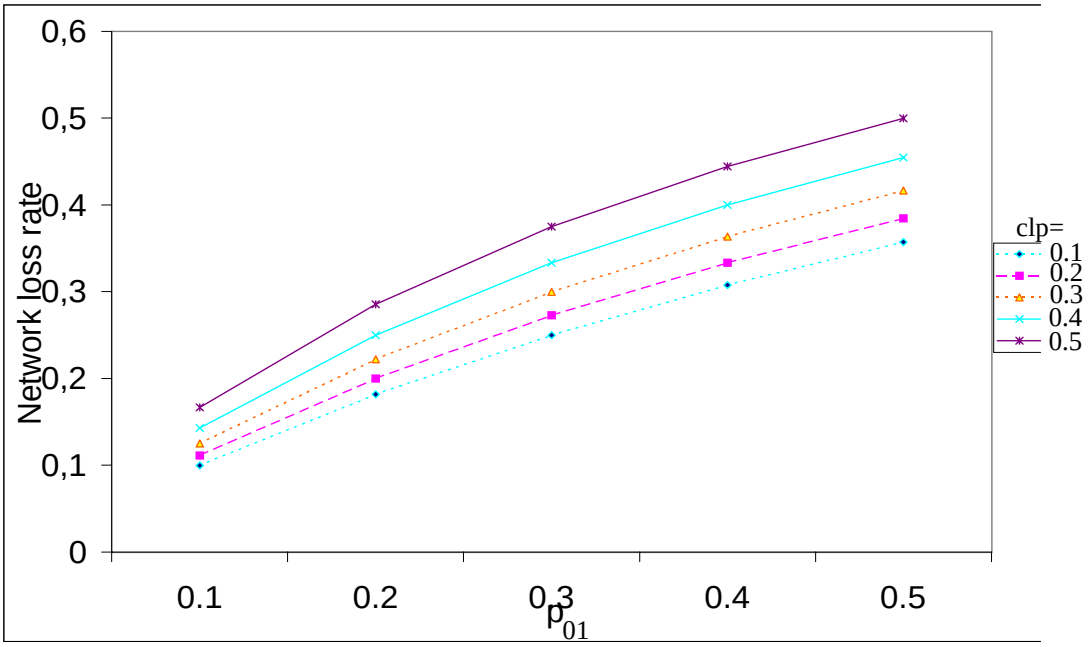


Figure 5.28: Network-level loss rate (unconditional loss probability) in simulation step 1.

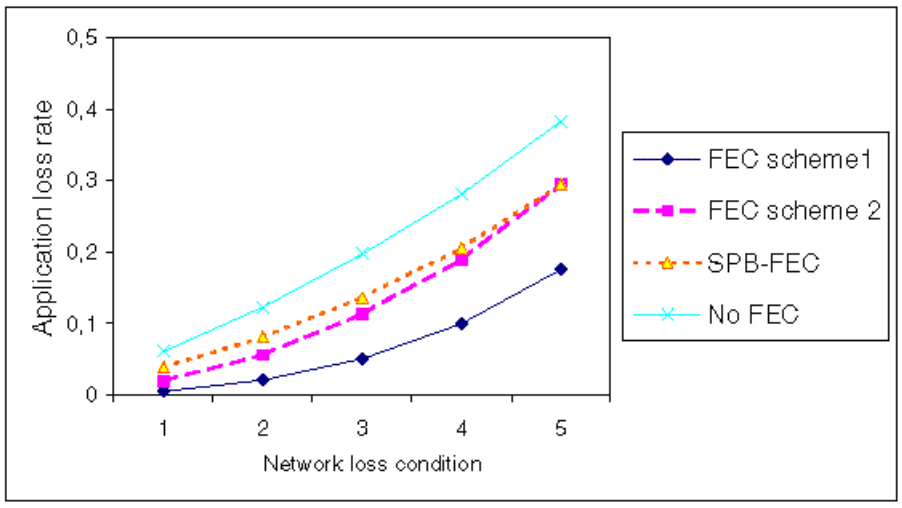


Figure 5.29: Application-level loss rate for different FEC schemes and network loss conditions.

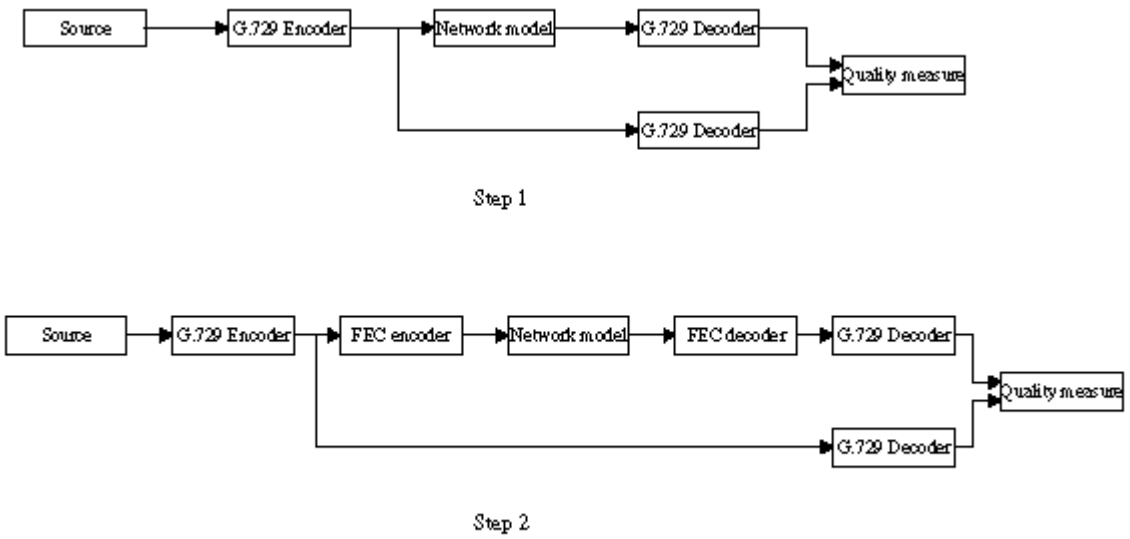


Figure 5.30: Simulation steps for the evaluation of the FEC schemes.

male and female voices as input to the simulation but use different seeds for the pseudo-random number generator to generate different loss patterns. This is important because, as we have seen in section 5.2.3, different loss patterns can have largely different levels of impact on the speech quality, e.g. a loss pattern dropping only voiced frames would result in a worse speech quality than a loss pattern dropping only unvoiced frames. By averaging the result of the objective quality measures for several loss patterns, we have a reliable indication for the performance of the G.729 codec and the FEC schemes under a certain network loss condition.

5.2.5 Results

It has been feasible to employ the frame-based SNR for the experiments in section 5.2.3 because there we have examined only one system (G.729 without any per-packet protection) under different error conditions. Now, however, we will compare several systems (G.729 with permanent and different partial protection modes) under similar error conditions. The system with permanent protection will be able to reconstruct more packets whereas the other systems rely much more on the internal concealment of the G.729 decoder, which is able to maintain a low signal degradation under the conditions described in section 5.2.3. However, the relation of the resulting speech qualities cannot adequately be captured by an SNR (e.g. the gradual dampening of the gain coefficients of the previously received frame during the loss concealment improves the speech quality, but lets the recovered signal largely deviate from the original signal in the mathematical sense).

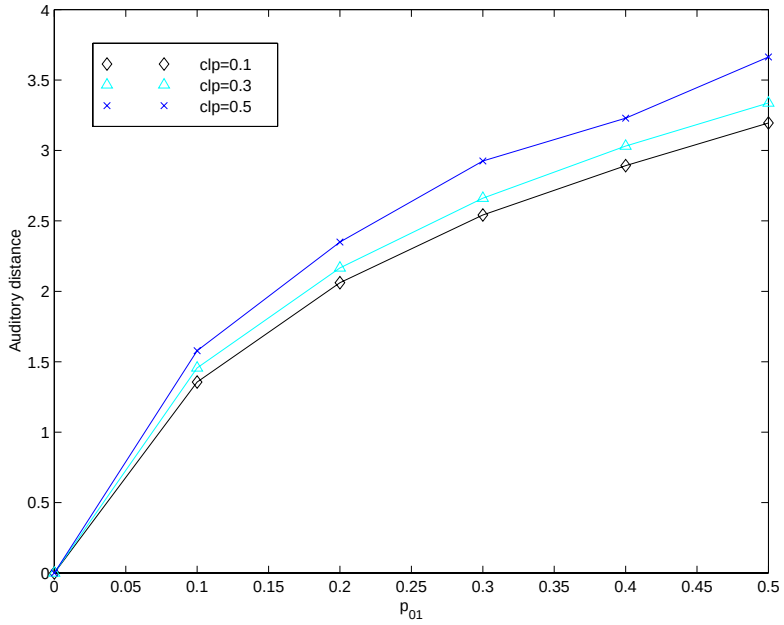


Figure 5.31: Auditory Distance for simulation step 1

In MNB (see section 4.2.1.2), the perceptual difference between the test signal and the reference signal is measured at different time and frequency scales. The perceptual difference, also known as Auditory Distance (AD), between the two signals is a linear combination of the measurements where the weighting factors represent the auditory attributes. The higher AD is, the more the two signals are perceptually different and thus the worse the speech quality of the test signal is (see section 4.2.1.2). Figure 5.31 (Fig. 4.16) and Figure 5.32 show the auditory distance evaluated by MNB resulting from the two simulation steps.

Figure 5.33 (Fig. 4.17) and Figure 5.34 show the perceptual distortions evaluated by EMBSD (see section 4.2.1.2 and 5.1.3.1) resulting from the two simulation steps.

The results of MNB and EMBSD for the second simulation step (Figure 5.33 and Figure 5.34) show the quality of the decoded speech signals for the different FEC schemes. We can see that the decoded speech signal without FEC has the highest auditory distance (in case of MNB) and the highest perceptual distortion (in case of EMBSD) and thus the worst speech quality. This is obvious because the scheme without FEC transmits no redundant data and has the highest application loss rate. However, the auditory distance and the perceptual distortion of the SPB- FEC is significantly lower than those of the reference FEC scheme 2 even though SPB-FEC has a higher application loss rate. The auditory distance and the perceptual distortion of the SPB-FEC method come even very close to those of the reference FEC scheme 1 although the application loss rate of scheme 1 is much lower. These results validate the strategy of our SPB-FEC scheme that does not distribute the

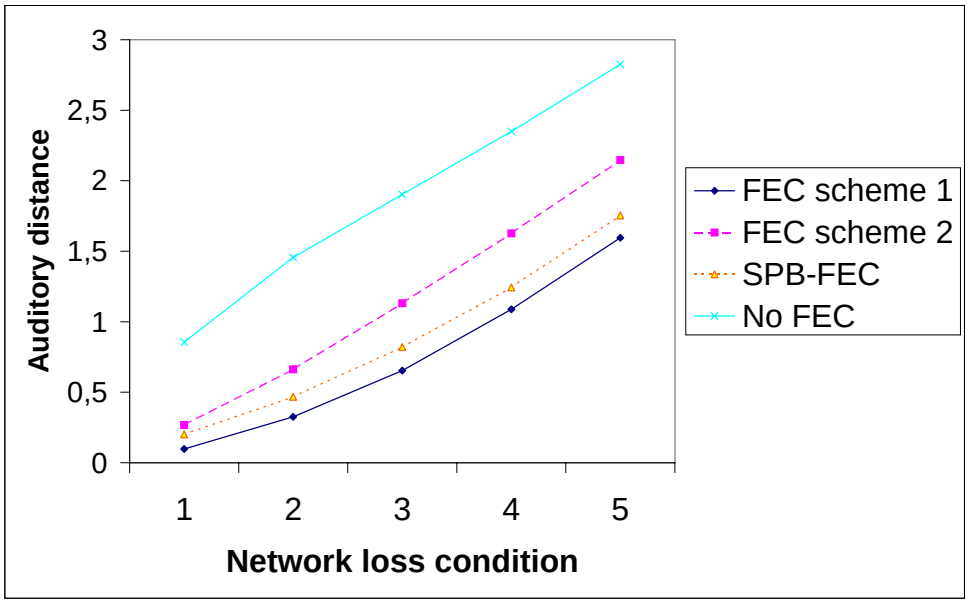


Figure 5.32: Auditory Distance for the FEC schemes

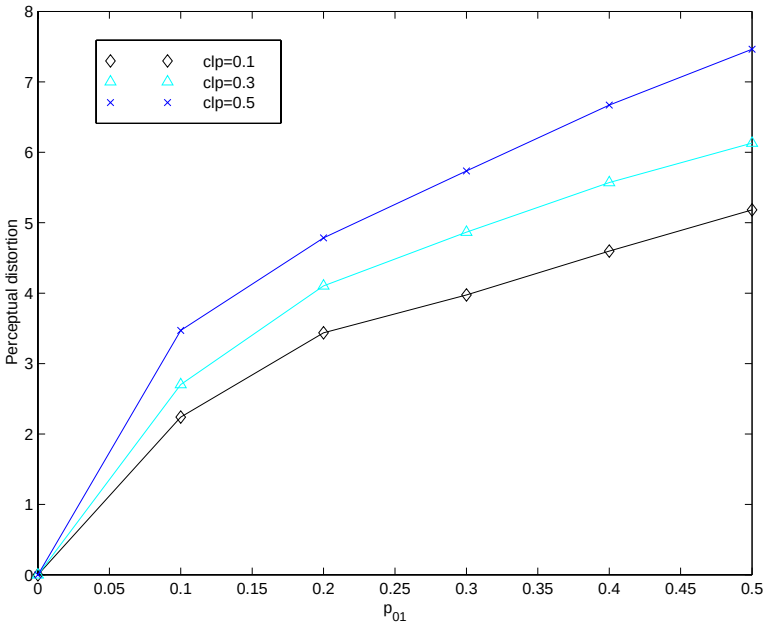


Figure 5.33: Perceptual Distortion for simulation step 1

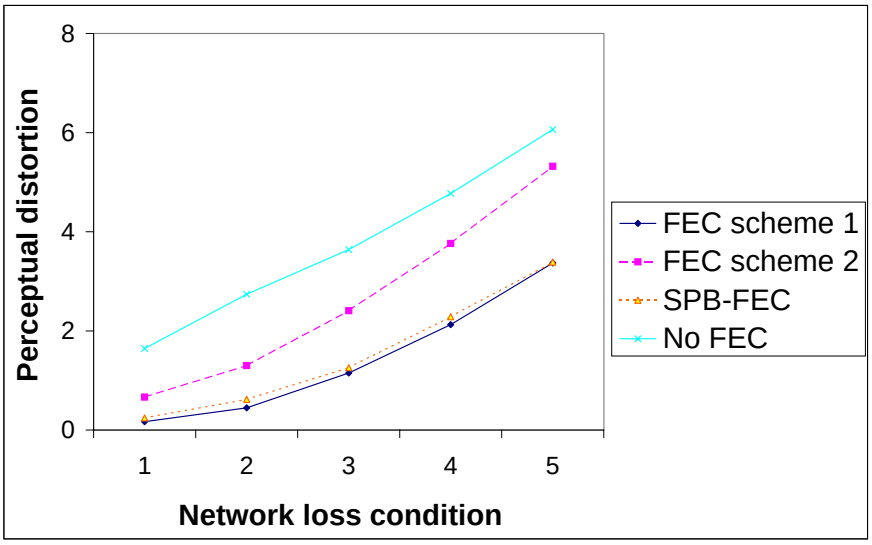


Figure 5.34: Perceptual Distortion for the FEC schemes

amount of redundant data equally on all packets but rather protects a subset of frames which are essential for the speech quality.

5.3 Conclusions

In section 5.1 an end-to-end loss recovery technique for sample-based voice traffic based on sender-supported concealment has been presented (Adaptive Packetization / Concealment: AP/C). The main idea of the scheme is to employ preprocessing of a speech signal at the sender to support possible concealment operations at the receiver. It results in an inherent adaptation of the network to the speech signal, as predefined portions of the signal ("chunks" assembled to packets) are dropped under congestion.

AP/C has been evaluated in comparison to "silence substitution" by using a simple loss model in connection with objective speech quality measurement. When the loss correlation (clp) is low, AP/C provides a significant performance improvement over the silence substitution case. The relative improvement with regard to silence substitution increases with increasing loss (ulp). A subjective test for various loss conditions, where AP/C has been additionally compared to the Pitch Waveform Replication (PWR) concealment algorithm and shown superior performance, has confirmed these conclusions. Though AP/C has some sender support it is still backwards compatible assuming correct legacy receiver implementations in terms of RTP (use of the header extension) and variable size packets (correct determination of the

play-out point).

As the evaluation has shown, AP/C (as all concealment algorithms) is best applicable when the conditional loss probability is low. Therefore some combination with supporting network algorithms controlling the loss distribution is highly desirable (chapter 6). AP/C is difficult to be applied to frame-based codecs because typically the frame size of frame-based codecs cannot be varied from frame to frame. Additionally due to the backwards-adaptive coding scheme employed by these codecs the synchronization of encoder and decoder is lost during a packet loss gap. Thus the decoding is worse after the gap due to previous decoder state loss. Then this low-quality speech is used in the concealment. Furthermore, AP/C exploits the long-term correlation for loss resilience, however when using frame-based codecs, this property has been already exploited to some extent for coding (i.e. the correlation has been removed). Thus protection for frame-based codecs needs to be codec-specific.

Building on these conclusions, in section 5.2 we showed (while the basic functional blocks within an audio tool are retained) that for low-bit-rate frame-based codecs it is important to carefully evaluate the internal coder-specific loss concealment algorithms. For the G.729 codec, we have investigated the impact of frame loss at different positions within a speech signal on the quality and gained the knowledge that the loss of voiced frames at the beginning of a voiced signal segment leads to a significant degradation in speech quality while the loss of other frames are concealed rather well by the decoder's concealment algorithm. We have then exploited this knowledge to develop a speech property-based FEC scheme (SPB-FEC) that protects the voiced frames that are essential to the speech quality while relying on the decoder's concealment in case other frames are lost. Simulations using a simple Gilbert model and subsequent evaluation using objective speech quality measures showed that our FEC scheme performs almost as good as other FEC schemes at a significantly lower redundancy overhead.

The parameter N, describing the number of to-be-protected frames, has been set in our evaluation in a "safe" way, i.e. such that for various loss patterns the described effect of the complete failure of the internal concealment (section 5.2.3) does not occur. Clearly further improvement is possible here. Also, further varying the number of frames per packet (parameter k) could be promising as we have seen that the impact of burst frame loss is not as severe as expected, although an increasing k also increases the buffer delay and makes sender classification more difficult.

We have used the ITU-T G.729 implementation which includes silence detection ([Uni96b]). Silence detection (cf. section 2.2.2) means here that during time periods identified as "inactive" a lower bit-rate stream is emitted, which contains codewords representing the characteristics of the background noise. This information is then used to generate comfort noise (section 3.1.3.2/Noise insertion). In the presented work we did not distinguish between frames of the two categories (regular codec frame, noise frame). While we believe that this does not affect our results in general (the used speech material did not contain significant silent periods), an extended

version of the SPB-FEC algorithm should take the two frame categories into account.

Although we only investigated the inter-operation of the G.729 codec and our speech property-based FEC scheme, we believe that a similar gain in speech quality can be expected when our scheme is applied to support other frame-based codecs (e.g., the G.723.1 codec) that operate in a similar way (in particular, G.723.1 incorporates an algorithm similar to that of G.729 to conceal frame loss using the codewords of the previous frames).

Despite its promising results, SPB-FEC faces the general problem of FEC schemes: transmitting redundant data also adds more load to the network and thus worsens congestion in the Internet (note that due to the missing property of adaptivity (Table 1.1), SPB-FEC is an inter-flow QoS scheme). Besides, SPB-FEC, as any other FEC scheme, only reduces but cannot come close to eliminate the possibility of losing important frames. The presented end-to-end scheme thus could be complemented by enforcing periodic loss patterns at congested routers. This is the subject of the following chapter. An option which avoids the addition of redundancy is to explicitly map the pattern of essential and non-essential packets (which contrary to waveform codecs is not a simple periodic pattern) discovered by the SPB algorithm onto network prioritization. This approach, which enables both intra- and inter-flow loss protection, is discussed in section 7.2.

Chapter 6

Intra-Flow Hop-by-Hop Loss Control

Section 5.1 has shown that for sample-based codecs a periodic loss pattern of only isolated losses can improve the performance of loss concealment. This holds also for the forward error correction scheme employed in section 5.2 (FEC schemes are generally sensitive to particular loss patterns dependent on their generation pattern; section 3.1.2.2). Additionally the analysis of the internal loss concealment in section 5.2 has shown that the concealment performance is highly dependent on the (non-periodic) loss pattern. This underlines the importance of intra-flow QoS as discussed in the introduction. To control intra-flow QoS, typically filtering higher-layer information within the network is proposed, which is both expensive in terms of resources, as well as undesirable with regard to network security (section 3.3.1).

In this chapter, we present queue management algorithms that allow to enhance the intra-flow QoS at the packet level without higher-layer filtering. Thus the algorithms can bridge the gap between employing end-to-end loss recovery mechanisms in a best-effort-only Internet and deploying service differentiation and reservation (including charging and accounting) in every node.

For the Predictive Loss Pattern (PLoP) algorithm presented in section 6.2 we use the heuristic approach of observing the packet sequence directly when dropping a packet. We then use some observations about the Random Early Detection (RED, [FJ93]) algorithm to design appropriate modifications for that algorithm which fulfill our goals (section 6.3). Within each section, we present simulation results for a voice service showing the performance at a congested network element in terms of the performance metrics of chapter 4.1 and processing/state overhead.

For the evaluation of hop-by-hop loss control schemes which should *support* the performance of end-to-end algorithms, some assumption about the requirements of the end-to-end level must be made. Here, considering voice traffic, we assume that it benefits from a simple, periodic loss pattern, i.e. either the encoding is sample-based with loss concealment (section 5.1) or any encoding together with RFC 2198 piggyback FEC (section 3.1.2.2/Transport) with $n - k = 1$ redundant units piggybacked in a distance of $D = 1$ is used. Due to this assumptions we are able to employ simple Gilbert model metrics most of the time (unconditional and

conditional loss probabilities: section 4.1.4; see also the conclusions of section 4.5). Section 6.4 then presents a comparison between the two developed algorithms and extends the results to a multi-hop network scenario.

In the following section we now want to explore how the packet-level metrics introduced in chapter 4.1 can be applied to characterize the goal of hop-by-hop schemes and the performance of the respective algorithms realizing that goal.

6.1 Approach

Short-term QoS (see the introduction to section 4.1) has been mainly mentioned in the context of admission control, i.e. in the access control path of multiplexers ([NKT94, LNT96]). In contrast, we consider a dynamic Internet scenario where real-time flows can start and end at any time without explicit QoS setup, i.e. we have no a-priori knowledge of connections, and thus QoS has to be enforced in the data path.

To characterize the behavior of the network as seen by one flow, we use the metrics introduced in chapter 4.1, in particular the Gilbert model (see sections 2.2.1.1 and 4.1.4). The unconditional loss probability using Gilbert model parameters can be expressed as follows (Eq. 2.6):

$$ulp = \frac{p_{01}}{1 - p_{11} + p_{01}}$$

Fig. 6.1 shows how the (clp, ulp) space is covered by the Gilbert model using p_{01} as a parameter.

Frequently (e.g. for uncontrolled queues with Drop-Tail queue management) the probability of a packet being lost is higher in case the previous packet is also lost than in case the previous packet has not been lost ([SKT92]). This is reflected by $p_{01} \leq clp$, i.e. for a queue with length K, the probability of a transition to state 1 (queue length $= K$) is smaller if the previous state has been 0 (queue length $\in [0, K]$) than if the previous state has already been 1. With $p_{01} \leq clp$ we also have $ulp \leq clp$ (upper half of Fig. 6.1). For $p_{01} = clp$ the Gilbert model is equivalent to a 1-state (Bernoulli) model with $ulp = clp$.

By modifying the queue management algorithm, we cannot change the conditional loss probability clp below a theoretical limit. This limit represents deterministic loss patterns. It defines the *deterministic conditional loss probability* clp_{det} given by the following function (Fig. 6.1):

$$clp_{det} = \begin{cases} 0 \colon 0 \leq ulp < 0.5 \\ 2ulp - 1 \colon 0.5 \leq ulp \leq 1 \end{cases} \tag{6.1}$$

Fig. 6.1 leads us to the conclusion that queue management algorithms can be designed that allow the adjustment of the conditional loss probability for individual flows, while keeping the unconditional loss probability within a controlled bound around the value that is determined by the background traffic intensity, buffer size, and scheduling policy, but not by the queue management algorithm itself. In the

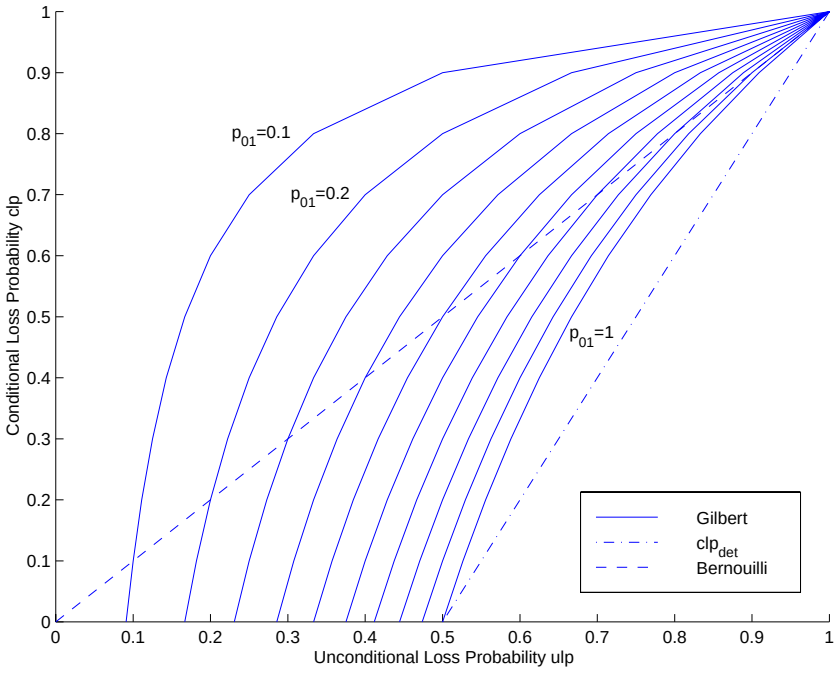
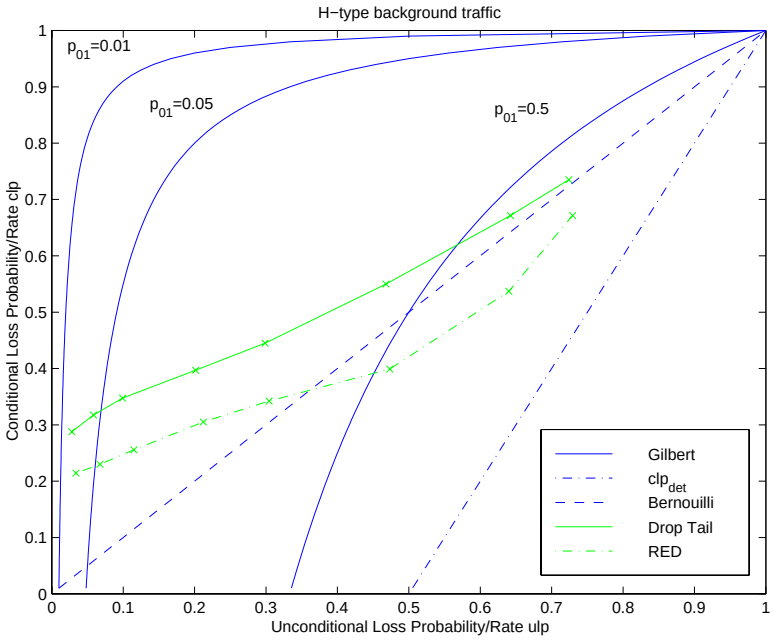


Figure 6.1: Conditional loss probability vs. unconditional loss probability: models and bound

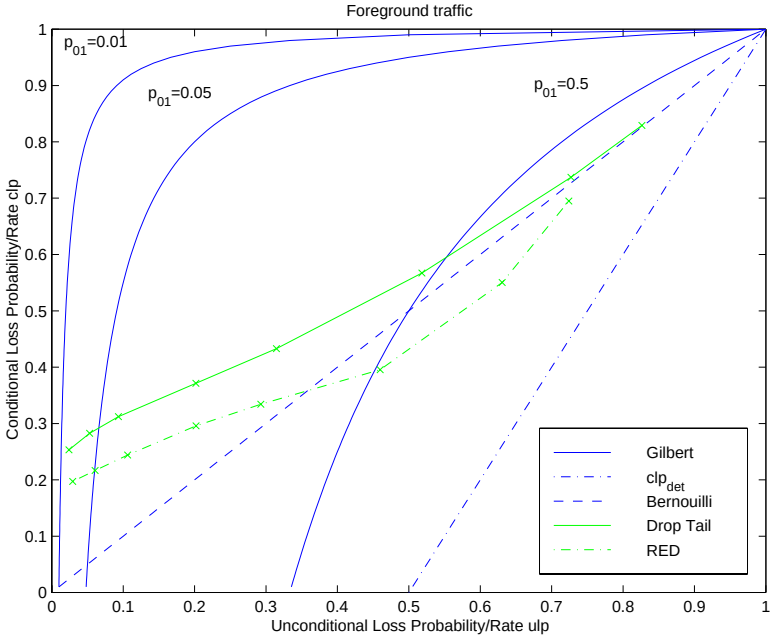
following we investigate the impact of different queue management algorithms, calling flows sharing a queue under the control of such an algorithm *foreground* traffic (FT) and the remaining flows in that queue *background* traffic (BT).

With the RED algorithm (section 3.2.1) there is already an existing queue management algorithm whose modifications to the queue behavior can be described with Gilbert model parameters. To be able to accomodate bursts in the queue, as well as not to over-react during transient congestion, the instantaneous queue size q is low-pass filtered resulting in an *average queue size* (avg) which is used to compute the drop probability (see Fig. 3.14). By employing RED, the parameter p_{01} of the queue is thus increased by gradually increasing the packet drop probability (according to the measured average queue size) before the queue is completely filled.

However, being interested in the clp, we see from Fig. 6.1 that for a given ulp, increasing p_{01} amounts to a reduction in the clp. This effect is also shown in Fig. 6.2 for simulations we conducted with parameters detailed in section 4.4. Fig. 6.2 a) and b) show clp vs. ulp for bursty background traffic and periodic foreground traffic respectively. For all ulp values, the conditional loss probability when using RED is below that for a Drop Tail queue. Only under heavy overload (when the RED algorithm is also just tail dropping most of the time), the RED curve approaches the Drop Tail one. The asymptote for both algorithms for extreme ulp values is the Bernoulli model ($ulp = clp$). It should also be noted that the results shown deviate heavily from the Bernoulli model for low ulp values where the clp is significantly



(a)



(b)

Figure 6.2: Conditional loss probability vs. unconditional loss probability: simulations of Drop-Tail and RED algorithms for "H-type" background traffic (a) and foreground traffic (b)

larger than the ulp[1].

In the remaining part of the chapter we now explore how the ulp/clp relation could be changed on behalf of the application. For described type of voice traffic, a static value of $clp \rightarrow clp_{det}$ is the goal.

6.1.1 Design options

Controlling loss burstiness is dealing with intra-flow QoS and thus does not necessarily need to be coupled with some form of admission control/Services Level Agreements (SLAs) and inter-flow QoS enforcement. Consequently, a flow can signal its participation either by *explicit (per-flow) signaling*, by *explicit (per-packet) marking* or implicitly by *detection* of flows (e.g. by using the RTP payload type) at the routers.

In section 6.2, an approach is presented based on detection of foreground traffic flows. State is kept on those flows which have been discriminated previously (i.e. lost packets). A queue management algorithm is proposed, which enforces preconfigured "drop profiles" on flows using a drop front queue with selective discarding (i.e. an already queued packet is discarded if it is more eligible to be dropped than the packet at the front of the queue). Note that with a detection approach it is thus only possible to enforce periodic patterns, where the length of the drop profile determines the period.

For the explicit marking scheme (as compared to detection of flows) we identify the following key advantages: The gateways are not required to build and maintain per-flow state. Non-periodic patterns can be enforced without keeping additional state. Routers just need to know the (IP level) marker bits and do not need any knowledge about specific flow types. There is no need to lookup fields in the (possibly encrypted) packet payload (like flow type, sequence number, etc.). A simple integration into the Differentiated Services Architecture (section 3.2.2.2) is possible: there per-packet marking is used to enable preferred treatment of flows (inter-flow QoS), but this is also accommodating the enforcement of intra-flow loss patterns.

However, also the following issues have to be accounted for: the operation at senders and routers is permanent even in the absence of congestion. Participating end-systems (or first-hop routers) need to be upgraded to do the packet marking. Usage control mechanisms need to be implemented if the initial marking is not under control of the service provider: in section 6.3 we introduce a mechanism which influences the packet dropping probabilities when the number of packets marked as "not eligible" for a drop differs significantly from the number of packets marked as "eligible" to avoid abuse of the proposed scheme. However, in the absence of per-flow state, this mechanism will degrade the quality for all FT flows in the same

[1] A comparison of Fig. 6.2 a) and b) also shows that RED is biased in favour of the FT (Fig. 6.2 b): the measurement points indicated by the markings on the RED curve shift to relatively lower ulp values with increasing load as compared to the BT case depicted in Fig. 6.2 a). This effect can also be seen in Fig. 6.17.

way[2].

Due to the well-known complexity of explicit signaling approaches (section 3.2.2.1, p. 53) we do not consider such an approach here.

To summarize, the main goal is to approximate the given loss requirements of the foreground traffic (spread inevitable loss over a larger time period) while at the same time avoiding a negative impact on the background traffic, especially on adaptive BT like TCP or rate-adaptive real-time flows. Unnecessary burst losses should be avoided where possible. "Unnecessary" here means that the impact of a dropped packet on a particular FT flow is much higher than on another previously unharmed one also currently active. Additionally, the incurred overhead (control state and additional processing) at the gateway has to be adequate for the only intra-flow QoS assurance given.

Three options for the basic structure of a burst loss control algorithm can be identified:

1. per FT flow queuing (n queues)
2. per FT/BT queuing (2 queues)
3. single queue

We explore only item 3. because it has the desirable feature that (using flow detection) the algorithm needs to be active only during times of congestion as well as simplicity, scalability (no scheduling between queues, only queue management is needed), easier combination with schedulers and thus potentially simpler deployment.

6.2 Implicit cooperation: the Predictive Loss Pattern (PLoP) algorithm

The PLoP algorithm ([SC98]) aims at equally distributing necessary packet drops within a single queue between flows belonging to a certain group of flows with similar properties/ QoS requirements (foreground traffic: FT). This is done to minimize violations of the given advance characterization of the flow's sensitivity to burst losses ("drop profiles").

[2]It should be noted that "proper" marking behavior (i.e. marking an equal number of packets as "not eligible" and "eligible") can be compared to proper behavior in terms of TCP adaptivity. While it is possible to modify the TCP congestion control in such a way to aggressively grab more bandwidth, it is considered to be against the "netiquette" and in fact is pretty uncommon in the Internet.

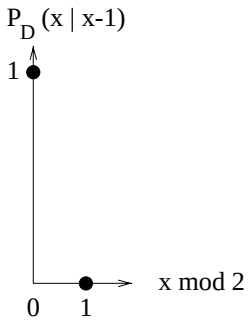


Figure 6.3: Drop profile for sample-based voice

6.2.1 Drop profiles

The task of a "drop profile" is to translate the applications' end-to-end loss pattern requirements (i.e. the minimization of the conditional packet loss probability) to a *per-packet* behavior of a queue management algorithm at a single node.

A comparable approach is taken by Koodli and Krishna ([KK97], cf. section 3.3.1) where the application specifies an acceptable task loss of a scheduler over a time window which is then translated to a per-subtask control algorithm at a node. Seal and Singh ([SS96]) present the enforcement of "loss profiles" at the transport layer of the source host or an intermediate node ([BS96]).

For voice traffic we define a simple profile of the conditional *drop* probability $P_D(x|x-1), x > 0$ as in Fig. 6.3. $P_D(x|x-1)$ gives the probability used in a *drop experiment* (i.e. a random number is generated and compared against $P_D(x|x-1)$). Note that this profile does not designate consecutive packets (sequence number s) of the flow, but packets consecutively subject to a drop experiment (index x). Thus the profile describes rather the worst case, where during times of congestion every packet of a flow is subject to a drop experiment. If this profile is successfully enforced at a node, the resulting conditional loss probability of a particular, previously unharmed flow at this node is 0. This profile does not give information about an actual unconditional loss probability that can be expected, however it clearly establishes an upper bound on the unconditional loss probability $clp_{det} = 0 \Rightarrow max(ulp) = 0.5$ (Eq. 6.1).

The distribution of drop profiles could range from hard-coding within the PLoP algorithm (which we assume to be sufficient for a basic voice service, see section 1) up to "active" (per-flow) setup. Details on the distribution are however beyond the scope of this chapter.

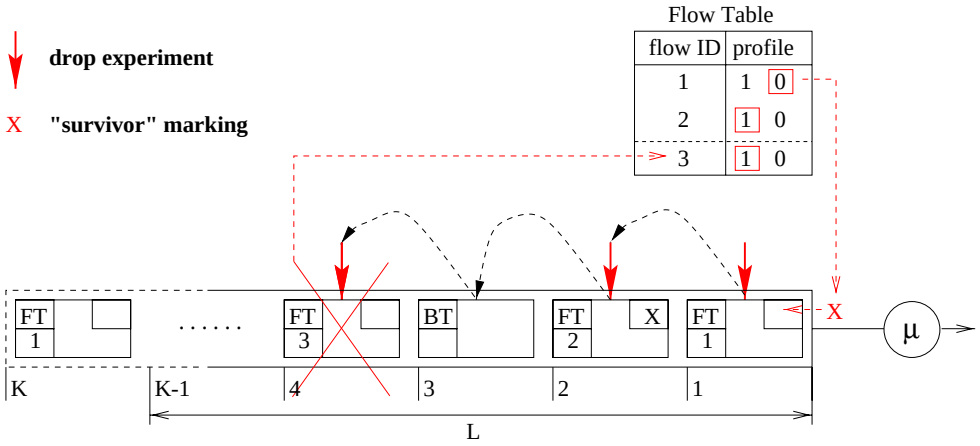


Figure 6.4: PLoP drop experiment

6.2.2 Description of the algorithm

When the queue length exceeds its threshold[3], a packet is selected to be dropped. After the first drop of a packet of a particular FT flow[4], the flow identifier (ID) and the index referring to a corresponding drop probability of the profile for the next drop is recorded in the flow table (Figure 6.4). The flow ID is the [protocol ID, source address/port, destination address/port] tuple for IPv4. With IPv6 the flow label can be used. Note that a flow might also consist of aggregated "micro"-flows ([RS96, RS98]).

When another FT packet must be dropped a drop experiment is performed (Figure 6.4). The table is checked, whether the ID of the selected packet has already been stored. If true, a random number is generated and the packet is dropped with a probability as found in the table record and the index into the profile within the flow table is updated. If this drop experiment does not result in an actual drop, the packet is marked as a "survivor" and the next packet matching the FT requirement is searched for in the queue ("force drop", see Fig. 6.5 for the algorithm pseudo code). This procedure is repeated until an actual drop has taken place. If the end of the queue is reached (i.e. no adequate replacement packet for the original packet was found: "force failure"), either the original packet or a BT packet is dropped.

[3]In our current implementation, the threshold is set to the maximum queue size. However, to better accomodate transient congestion and the additional processing time needed to execute the algorithm it seems promising to combine PLoP e.g. with RED (section 3.2.1) to control the average queue size. Additionally, the drop probabilities could be weighted with the average queue size.

[4]Usually the first profile probability is 1 however this number could be modulated dependent on the congestion state like with RED.

```
PLoP()
 if queue threshold exceeded
        delete timer
        if (packet ∈ FT) // flow type filter
              status = drop_experiment()
              if (status == FAILED) //"force failure"
                    drop // other policy: drop BT packet
        else
              drop
 elseif (not idle)
        if (timer expired)
              delete flow table, go idle
        elseif (timer not running)
              start timer
```

```
drop_experiment()
 if (flow not in flow table) // flow ID filter
        create flow table entry
 generate random number R ∈ [0 , 1]
 if R ≤ P_D(x|x − 1) and (packet not "survivor")
        drop
        return OK
 else // "force drop" of an FT packet
        mark as "survivor"
        if (end_of_queue)
              return FAILED
        else
              lookup next FT packet in queue
              status = drop_experiment()
              return status
```

Figure 6.5: Predictive Loss Pattern algorithm pseudo code

6.2.3 Properties

As enforcing drop profiles also results in establishing an upper bound on the unconditional loss probability (cf. section 6.2.1), the amount of flows concurrently under PLoP protection has to be limited accordingly. For voice traffic, the maximum flow table size is set to $\lfloor \frac{B\hat{p}_L}{r\alpha} \rfloor$ (with B: interface/link bandwidth, $0 < \hat{p}_L < 1$: upper bound on the mean loss rate as determined from the profile, r: rate expected of an individual flow during talk-spurts and $\alpha = 0.6$ (conservative) estimate of the speaker activity).

Flow table management policy Considering a limited flow table size, a flow table management policy is needed which defines when a flow ID is added/dropped from the flow table is needed. Two basic flow table management policies can be identified: preemptive and non-preemptive.

In the preemptive policy the table size is limited and handled in a FIFO way, i.e. if the length of the table is exceeded by adding a new entry, the oldest entry is dropped. The flow table is deleted entirely, when an "uncongested" state[5] persists to avoid keeping old state in the table. Using a non-preemptive policy, all packets belonging to flows not present in the (full) flow table are dropped, because otherwise the minimal guarantee on the loss rate would be violated. Rather than degrading the service given to all other flows below the acceptable minimum level, other "calls" are "blocked". For this policy, additional per flow table entry timers are needed,

[5]The "uncongested" state is determined by monitoring the (non-)access to the flow table over a time interval. Note that after expiration of the timer, PLoP stays idle and does not consume any resources.

otherwise entries of inactive flows could persist during congestion. Due to this additional overhead, the preemptive policy is used in our simulator implementation.

Force failure policy As the policy for the case when no adequate replacement packet was found in the queue ("force failure"), we adopted dropping of the packet that originally would have been protected. One might argue that a "force failure" is mainly due to a flow occupying more than its fair share of the buffer space which therefore should be discriminated. However, without further knowledge (state) of misbehaving flows ([FF97]), it should be avoided to randomly drop any background traffic. Results of section 6.2.4.1 show that (presuming sufficient buffer space) FT flows can be sufficiently protected even under overload conditions. Another reason for a "force failure" can be that only very few FT flows are active at a gateway. Here we argue that the impact of dropping background traffic due to the small overall FT bandwidth is minimal. However, as long as the link-speed equivalent buffer is larger than the FT inter-arrival time, this type of "force failure" virtually does not occur.

Choice of the dropping discipline and search direction The distribution of loss bursts has been shown to be similar for front and tail dropping disciplines ([ZR96, SKT92]). Combined with the possibility of searching for PLoP replacement packets from either end of the queue, four strategies exist which lead to marking packets at different queue locations.

All solutions except *drop from front/search from front* lead to accumulation of "survivor" packets in the queue (packets not dropped due to the PLoP drop logic should be as close as possible to the head of the queue to avoid unsuccessful drop experiments).

6.2.4 Results

To assess the performance of PLoP, we evaluated a scenario where several flows experience a bottleneck link (e.g. a small bandwidth access link connecting a customer LAN to an ISP or a base station connecting mobile hosts to a LAN; see section 4.4 for a detailed description).

We implemented the algorithm into a modified version of the NS-2 simulator ([UCB98]), which allows tracing of the occurrence o_k of burst losses of length k for individual flows (section 4.1.2). Thus for a given number of packet arrivals a (experiencing $d = \sum_{k=1}^{\infty} k o_k$ drops) of a flow we have the mean loss rate (ulp for $a \to \infty$) $p_L = \frac{d}{a}$ (Table 4.4). With $b = \sum_{k=1}^{\infty}(k-1)o_k$ being the occurrence measure of "two consecutive packets lost", we calculate a *conditional loss rate* as $p_{L,cond} = \frac{b}{d}$ (clp for $d \to \infty$). Note that for longer drop profiles (section 6.2.1) additional measures are needed (section 4.1). Additionally, we monitor various PLoP queue parameters.

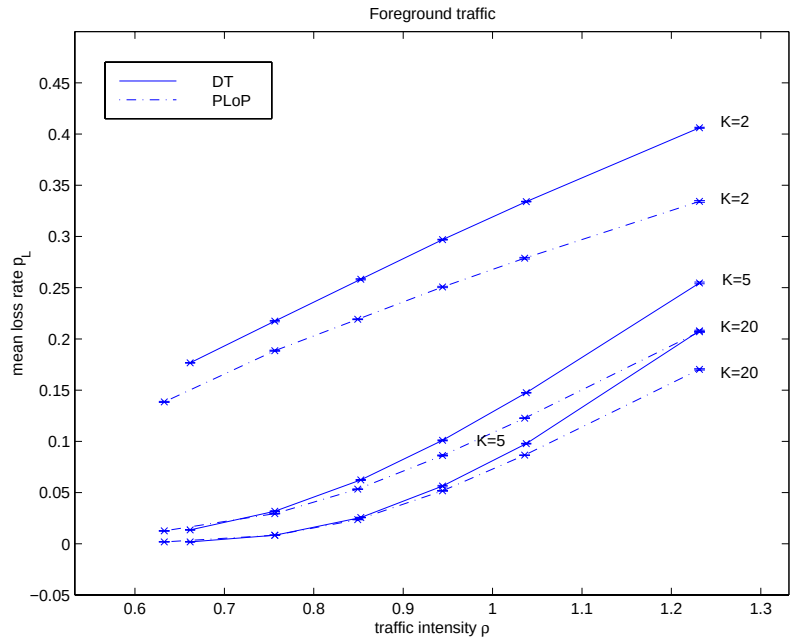


Figure 6.6: Foreground traffic: mean loss rate

6.2.4.1 Variation of the background traffic load

We set the share of voice traffic to 10% of the gateway bandwidth for all experiments, resulting in six active voice flows. The share of BT traffic (at a traffic intensity $\rho = \frac{\lambda}{\mu} = 1$, λ being the offered load) is set to 80% (18 flows) for H- and 10% (6 flows) for D-type BT respectively. For other traffic intensities, the BT share is varied while keeping the ratio of H- and D-type BT approximately equal, resulting in 12 H-type, 4 D-type ($\rho = 0.66$), up to 24 H-type and 8 D-type flows ($\rho = 1.23$) active.

Fig. 6.6 shows the mean loss rate p_L as a function of the traffic intensity ρ. Except for low buffer sizes ($K < 5$), we see that for $\rho < 0.9$, p_L has approximately the same value for Drop Tail (DT) and PLoP and thus seems to be acceptable in terms of fairness towards the BT. For higher loads, curves for the PLoP algorithm start to approach their asymptote (maximum possible loss rate) which is given by $\hat{p}_L = 0.5$ ($\hat{p}_L \to 1$ for DT, section 6.2.1).

Looking at the conditional loss rate $p_{L,cond}$ in Figures 6.8 and 6.9, we see that for DT, increasing the buffer size (except for very low buffer sizes) has virtually no effect on $p_{L,cond}$. For lower loads $\rho < 0.9$, $p_{L,cond} = \frac{b}{a}$ for $K = 20$ is even larger than $p_{L,cond}$ for $K = 5$. This is *not* due to a larger number of burst losses b for the larger buffer size, as can be seen from Fig. 6.7 where $\frac{b}{a}$ is decreasing with larger buffer sizes. The values for b, as well as the difference between values for b for different buffer sizes are small compared to a. Thus for lower loads (where the queue can drain between bursts) the loss process is dominated by burst losses caused by very large arrival bursts (burst size $\gg$ buffer size K) and singleton losses (which appear only in the denominator but not in the enumerator of $p_{L,cond}$). Note that the Pareto distribution

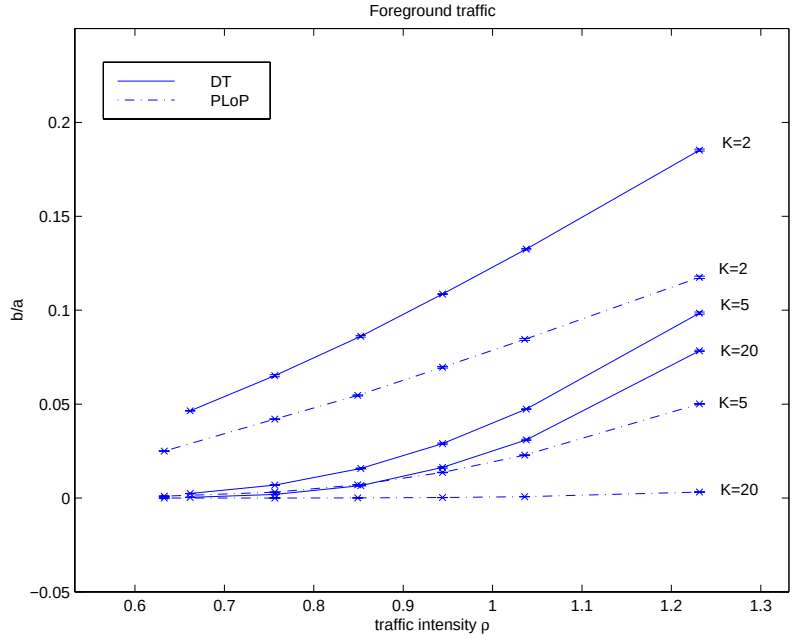


Figure 6.7: Foreground traffic: b/a

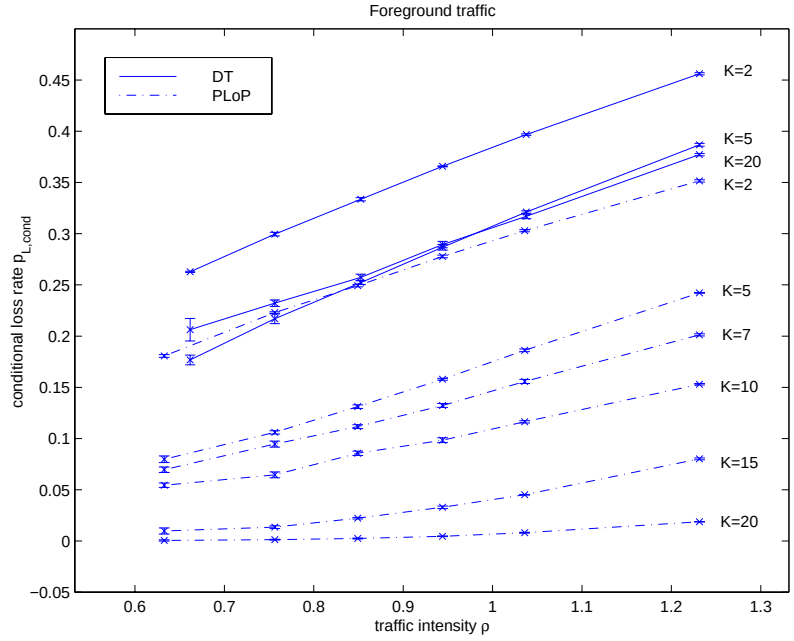


Figure 6.8: Foreground traffic: conditional loss rate as a function of traffic intensity (parameter: buffer size)

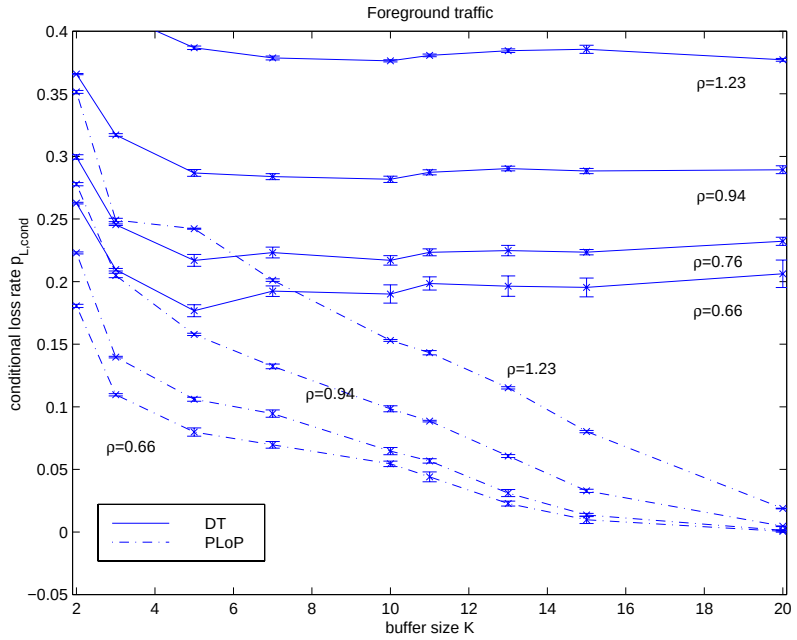


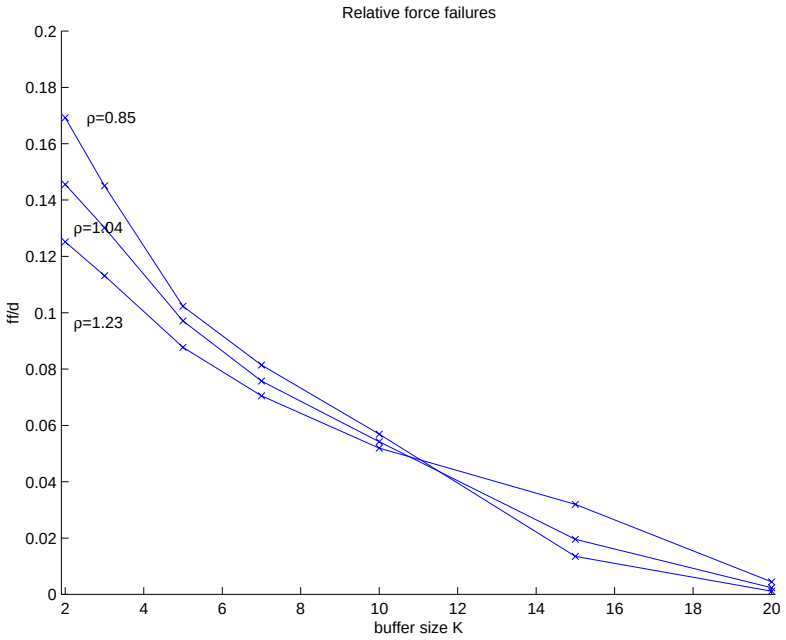
Figure 6.9: Foreground traffic: conditional loss rate as a function of buffer size (parameter: traffic intensity)

which is used for the BT traffic generation is heavy-tailed, i.e. significant parts of the probability mass are concentrated at rare (but large) bursts and frequent bursts of only few packets.

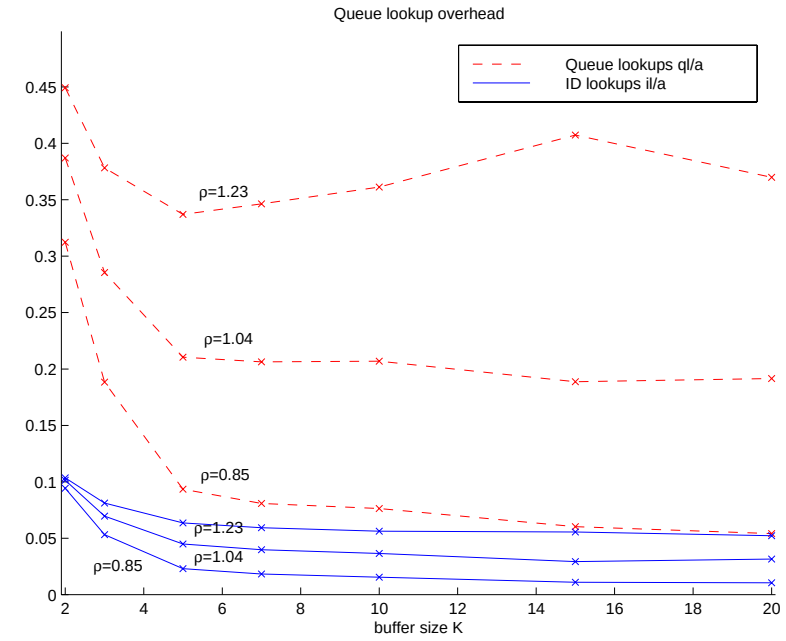
The behavior of $p_{L,cond}$ for the PLoP algorithm shows that PLoP can exploit larger buffer spaces to avoid burst losses within one flow. The achievable gain ranges from an enhancement of about 10% for $K = 2$ (yet still as DT dependent on the offered load) to virtually no burst losses for $K = 20$ (only weakly depending on the load, Fig. 6.8). Fig. 6.9 shows the linear decrease of $p_{L,cond}$ with increasing buffer size starting from $K \approx 5$. In Fig. 6.6 it can be seen that for decreasing buffer size and increasing load, PLoP becomes increasingly unfair (under these conditions the FT share of the number of drops is smaller than the FT bandwidth share of 10%) resulting in relatively less force failures for higher loads (Fig. 6.10 (a)). For larger values of K ($K \geq 11$), fair operating points are reached. This is due to the fact that the link-speed equivalent buffer is larger[6] than the voice packet inter-arrival time of $20ms$. Thus a consecutive packet of the *same* flow (which can be surely dropped) can be found with a higher probability in the queue.

To assess whether PLoP can achieve its limited QoS assurance goals with less processing overhead than the other design options given in section 6.1.1, we also traced the relative number of queue lookups $\frac{ql}{a}$ (i.e. searching in the queue and

[6]Assuming a voice packet at the head of the queue and nine H-type BT packets behind it, the time distance (time the voice packet has already been present in the queue under overload) from the head of the queue to the eleventh buffer is $\frac{(9\times 560+208)\times 8\ bit}{1.92\times 10^6\ bit/s} = 21.87ms$.

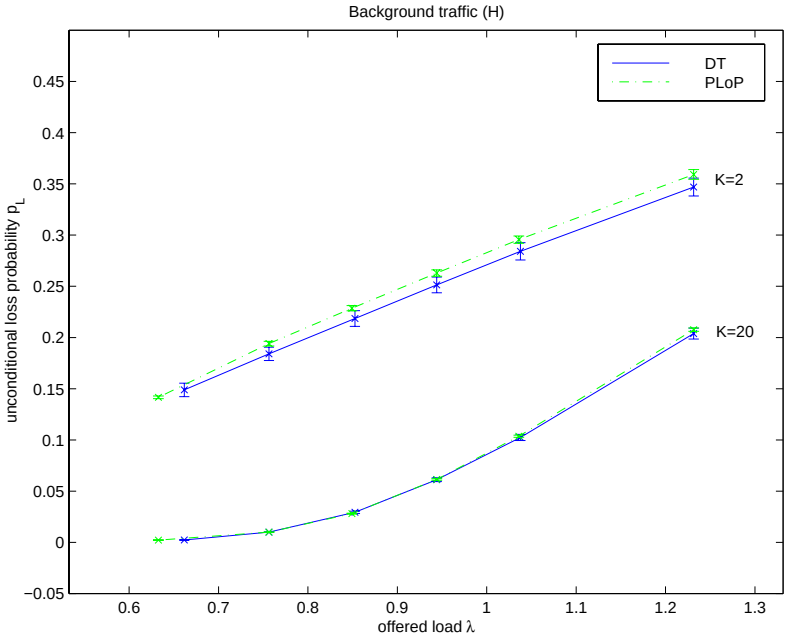


(a)

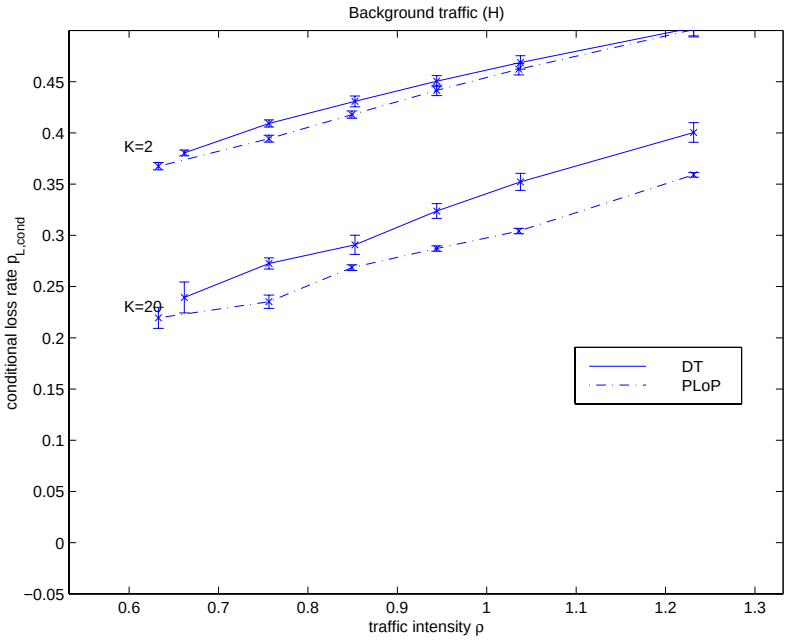


(b)

Figure 6.10: PLoP queue performance parameters



(a)



(b)

Figure 6.11: H-type BT performance measures

filtering on the flow type, Fig. 6.10 (b)). This number can be compared against the value $\frac{ql}{a} = 1$ for design option 2. (section 6.1.1: 2 queues - every packet has to be flow type filtered). It can be seen that except for overload conditions and very low buffer sizes, the overall queue lookup overhead stays roughly below twice the mean FT loss rate (Fig. 6.6) for the used drop profile. Additionally, the relative number of full flow ID lookups $\frac{il}{a}$ (=1 for design option 1.: a separate queue for every FT flow) is shown. Again, except for overload conditions and very low buffer sizes, the ID lookup overhead (which also indicates the relative number of drop experiments necessary, Fig. 6.5) stays clearly below 10% (the FT share of the bandwidth).

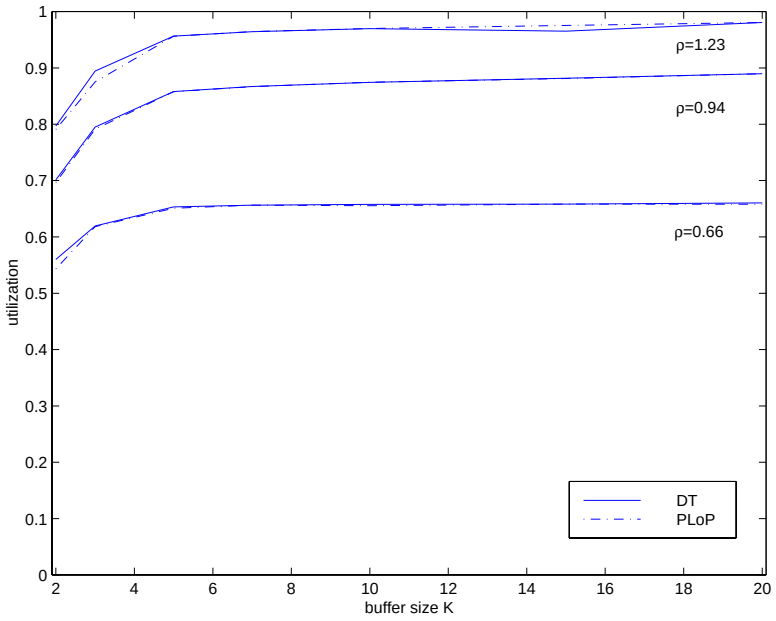


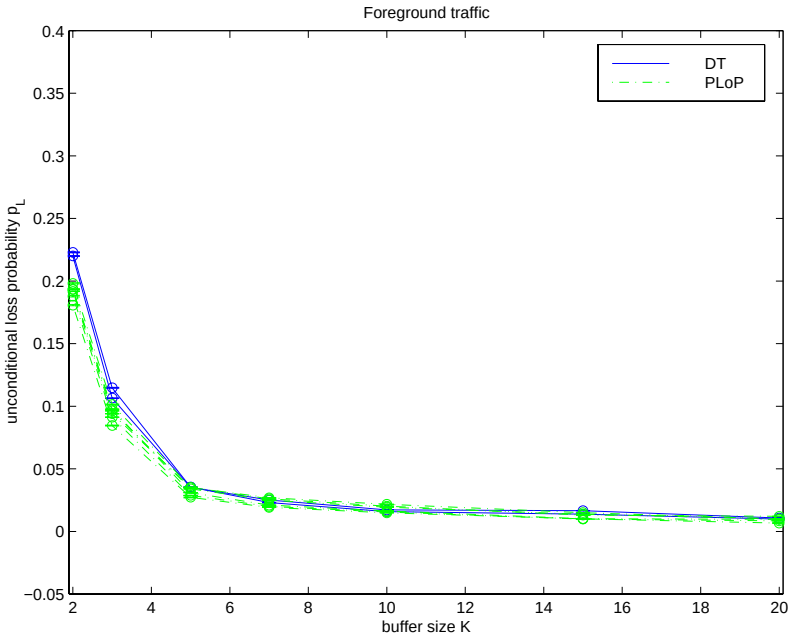
Figure 6.12: Link utilization

Fig. 6.11 (a) shows that background traffic is not negatively affected by PLoP operation in terms of the conditional loss rate. The small increase for higher loads is due to the asymptotic behavior of the FT's p_L described above. The conditional loss rate (Fig. 6.11 (b)) is slightly lower for PLoP than for DT.

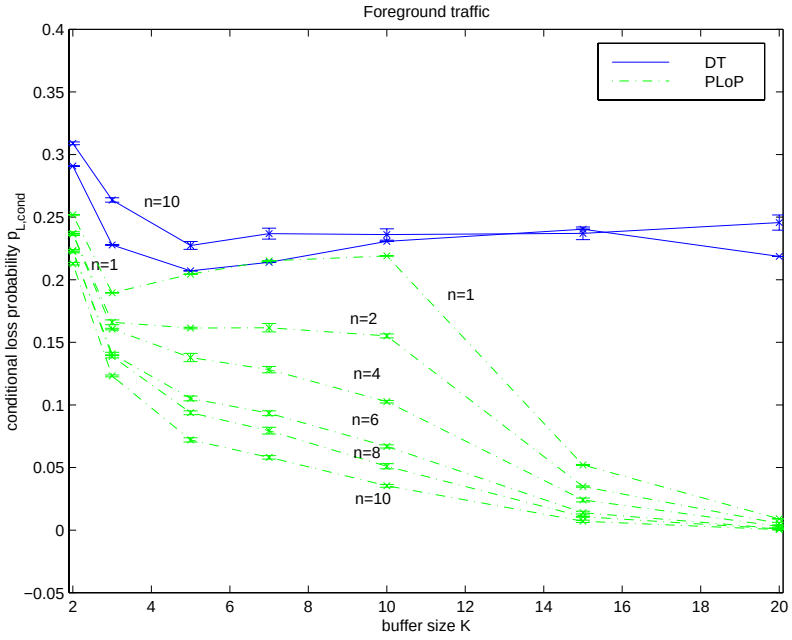
The overall utilization (Fig. 6.12) achieved is equal for either DT and PLoP, because the aggregated loss process (for all flows) has not been changed significantly.

6.2.4.2 Variation of the number of foreground traffic flows

As PLoP relies on shifting losses between flows belonging to a certain group, in this section we will explore the performance if only few flows of the FT group are present. Fig. 6.13 (a) shows that when varying the number of voice flows present from 1 to 10 no impact on the FT mean loss rate can be observed (again except for low buffer sizes $K < 5$). The conditional loss rate for PLoP (Fig. 6.13 (b)) varies significantly as expected, dependent on the number of FT flows. Note that the actual traffic



(a)



(b)

Figure 6.13: Foreground traffic performance measures as a function of buffer size (parameter: number of FT flows)

intensities for the following examples are slightly different ($\rho = 0.75...0.77$), due to the changing partition of flow groups (this explains the difference of the two $p_{L,cond}$ curves ($n = 1$, $n = 10$) for DT). For $n = 1$ and $K < 11$, $p_{L,cond}$ is virtually equal to the DT case.

For larger buffer sizes, when the flow's inter-arrival time is smaller than the link-speed equivalent buffer, $p_{L,cond}$ drops from 20% ($K = 11$) to 6% ($K = 15$) due to the added probability of finding a replacement packet in the queue belonging to the *same* flow as the protected packet. With increasing n, the effect becomes less and less important for the success of the PLoP algorithm.

6.3 Explicit cooperation: the Differential RED (DiffRED) algorithm

Based on the observations in section 6.1, in this section we present another simple network mechanism ([SC99]) which allows loss control on a per-packet rather than on a per-flow basis. In contrast to the previously introduced PLoP algorithm, this one uses explicit (per-packet) marking (section 6.1.1) to differentiate the conditional loss characteristics of a flow within a QoS class.

6.3.1 Description of the algorithm

RIO (see section 3.2.2.2) has been designed to lower the *ulp* seen by particular flows at the expense of other flows. In this work however, we want to keep the *ulp* as given by other parameters[7] while modifying the *clp* parameter for the foreground traffic. Fig. 6.14 shows the conventional RED (section 3.2.1) drop probability curve (p_0 as a function of the average queue size for all arrivals avg), which is applied to all unmarked ("0") traffic (background traffic). Foreground traffic packets marked as non-eligible for a drop ("+1") are dropped with a probability as given by the lower thick line. This lower probability is compensated by the higher drop probability for the foreground traffic packets marked as "-1", i.e. packets eligible for a drop. By this a service differentiation for foreground traffic is possible which does not differ from conventional RED behavior in the long term average (i.e., in *ulp*) .

However, considering a rather small fraction of FT traffic at the gateway and using the average queue size avg ($avg_1 = avg$, Fig. 6.14) for the calculation of the $+1,-1$ drop probabilities p_{+1} and p_{-1} we can identify the following problem: the state of the queue and thus the avg value may have changed significantly between consecutive FT arrivals. Thus a value for the drop probability is computed which does not reflect adequately the evolution of the queue state as seen by the FT fraction and its contribution to it. Ideally $p_0(avg_1(s)) + p_0(avg_1(s+1))$ (where packet s is a $+1$ packet and packet $s+1$ is a -1 packet or vice versa, and $avg_1(s)$ is the value of avg_1 at arrival of packet s) should equal the drop probability computed for the -1

[7]Note that this does not preclude a combination with mechanisms enforcing a certain *ulp*, e.g. with a link sharing scheduler.

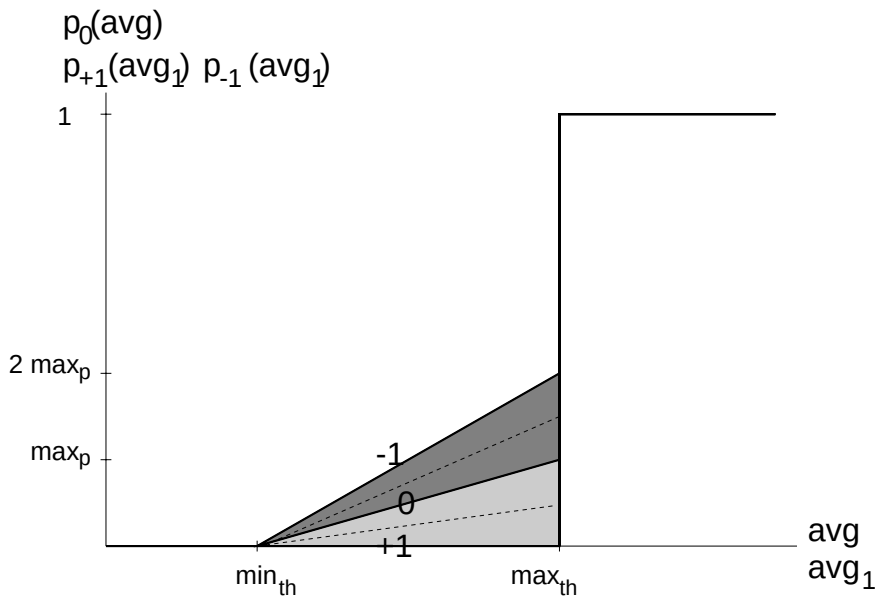


Figure 6.14: DiffRED drop probabilities as a function of average queue sizes

packet (either $p_{-1}(avg_1(s))$ or $p_{-1}(avg_1(s+1))$). If this relation is not approximated by the algorithm, it can lead to an unfair distribution of drops between the FT and the BT fraction.

The described problem can be solved by changing the low pass filter parameter as a function of the ratio of the number of FT arrivals to the overall number of arrivals when sampling the queue size and then computing an additional average queue size for the FT arrivals (avg_1). However, in this case we need to keep additional state about the number of FT arrivals, need to re-calculate the filter parameter and avg_1 at every arrival.

Instead, our approach avoids this complexity by sampling the queue length q only at the FT arrival instants. Now, the avg_1 filter is a sub-sampled version of the avg filter, with a subsampling factor equal to the current ratio of all arrivals to the FT arrivals. Fig. 6.15 shows the magnitude of the filter frequency response (assuming a time-invariant system) when modifying the filter parameter $w_{q,1}$ (solid lines), as well as when keeping $w_{q,1}$ constant and changing the sampling frequency to f_s' (dashed lines).

Now we can compute the drop probabilities for the different priority packets as follows:

$$p_0(avg) = \begin{cases} 0\text{: } avg < min_{th} \\ max_p \frac{avg - min_{th}}{max_{th} - min_{th}}\text{: } min_{th} \leq avg < max_{th} \\ 1\text{: } avg \geq max_{th} \end{cases}$$

$$p_{-1}(avg_1) = \begin{cases} 0\text{: } avg_1 < min_{th} \\ 2max_p \frac{avg_1 - min_{th}}{max_{th} - min_{th}}\text{: } min_{th} \leq avg_1 < max_{th} \\ 1\text{: } avg_1 \geq max_{th} \end{cases}$$

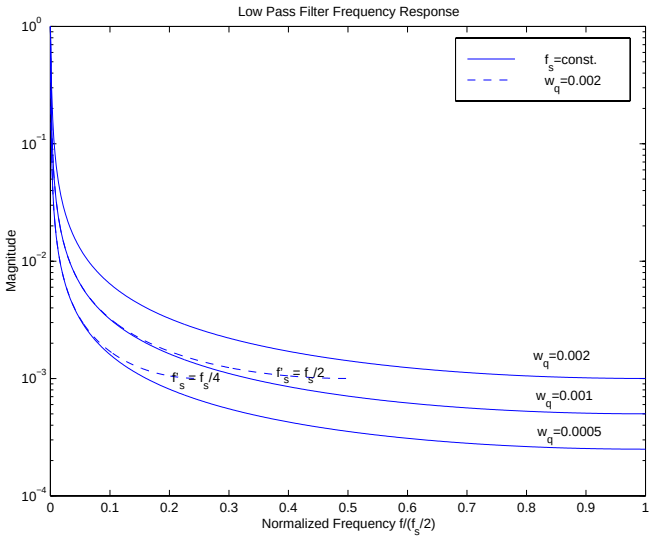


Figure 6.15: Low-pass filter frequency response

$$p_{+1}(avg_1) \;=\; \begin{cases} 0\text{: } avg_1 < max_{th} \\ 0\text{: } min_{th} \le avg_1 < max_{th} \\ 1\text{: } avg_1 \ge max_{th} \end{cases}$$

Fig. 6.16 gives the pseudo code for the Differential RED algorithm (cf. [FJ93], Fig. 2).

Irregular partition of +1/-1 arrivals To discourage abuse by malicious users who could send just +1 packets, we compute low-pass filtered values of the arrival function of +1 packets (arv_{+1}) and −1 packets (arv_{-1}). The arrival function is defined as follows:

$$a_{x,FT} = \begin{cases} 0\text{: } & \text{FT packet type} \neq x \\ 1\text{: } & \text{FT packet type} = x \end{cases}$$

Note that the arrival function describes the FT arrival process[8], and not the sampling of overall arrivals at $+1, -1$ arrival instants. The arrival function for all FT packets $a_{|1|,FT}$ is thus 1 for all samples ($arv_{|1|} \to 1$). The choice of the averaging filter parameter allows to adjust the burst length of $+1, -1$ packets respectively which can be accomodated, while avoiding a persistent mismatch of the partition between +1 and −1 packets.

A correction is added to $p_{-1}(avg_1)$ and $p_{+1}(avg_1)$ to decrease the −1 loss probability and to increase the +1 probability at the same time thus degrading the service

[8]Note that it is necessary here to describe the *arrival* process rather than the distribution of packets which have been accepted into the buffer.

Initialization

$avg \leftarrow 0$

$count \leftarrow -1$

for each packet arrival

if queue has been idle

$m \leftarrow f(time - q_{time})$

$avg \leftarrow (1 - w_q)^m avg$

$avg_1 \leftarrow (1 - w_{q,1})^m avg_1$

$avg \leftarrow (1 - w_q)avg + w_q q$

$priority =$ filter(arriving packet)

if $priority! = 0$

$avg_1 \leftarrow (1 - w_{q,1})avg_1 + w_{q,1} q$

if $min_{th} \leq avg < max_{th}$ and $priority! = 1$

increment $count$

if $priority == -1$

$p \leftarrow \frac{2max_p(avg_1 - min_{th})}{max_{th} - min_{th}}$

else

$p \leftarrow \frac{max_p(avg - min_{th})}{max_{th} - min_{th}}$

$p_a \leftarrow p/(1 - count \cdot p)$

with probability p_a:

drop the arriving packet

$count \leftarrow 0$

else if $max_{th} \leq avg$

drop the arriving packet

$count \leftarrow 0$

else $count \leftarrow -1$

if queue is empty

$q_{time} \leftarrow time$

State

avg: average queue size for all packets

avg_1: average queue size calculated at arrivals of packets with $|priority| = 1$

q_{time}: time when queue goes idle

$count$: packets since last marked packet

Fixed parameters

w_q: low-pass filter parameter for avg computation

$w_{q,1}$: low-pass filter parameter for avg_1 computation

min_{th}: minimum queue threshold

max_{th}: maximum queue threshold

max_p: maximum value for p

Other parameters

p_a: current packet-marking probability

q: current queue size

$time$: current time

$f(t)$: $\frac{link\_bandwidth}{assumed\_mean\_packet\_size \cdot t}$

Figure 6.16: Differential RED algorithm pseudo code

for all users[9]. The correction depends on the mismatch between the $+1$ and -1 arrivals. The shaded areas above and below the $p_0(avg)$ curve (Fig. 6.14) show the operating area when the correction is added. The corrected values for the $+1, -1$ drop probabilities for the interval $min_{th} \leq avg_1 < max_{th}$ are:

$$\begin{aligned} p'_{-1}(avg_1) &= p_{-1}(avg_1) - \frac{|arv_{+1} - arv_{-1}|}{arv_{|1|}} p_0(avg_1) \\ p'_{+1}(avg_1) &= \frac{|arv_{+1} - arv_{-1}|}{arv_{|1|}} p_0(avg_1) \end{aligned}$$

Every congested DiffRED hop will increase the mismatch between the number of $+1$ and -1 packets at the next hop. If this effect becomes significant is a function of the number of congested hops already traversed by the flows present, as well as the congestion situation at a gateway and the relation of the presence of "fresh" flows which enter the network and flows which have already experienced several congested gateways. Note that the higher the individual loss of a flow, the higher is the ratio of $+1$ to -1 packets of that flow. Thus the flow is protected more at subsequent gateways supporting end-to-end fairness.

Packet marking policy It would be possible to realize a variable marking granularity, i.e. that marking across flows and thus also inter-flow differentiation is possible. This means that a flow sent by a host could receive more $+1$ marking on the expense of another one sent concurrently, which would mark more packets as -1. However, the ratio of packets marked as $+1$ to the packets marked as -1 must remain 1 over short time intervals (the length of these time intervals depend on the DiffRED gateway filter parameters). Thus either ingress monitoring and suppression of mis-behaving flows or volume-based charging is needed, as otherwise users could inject just -1 traffic to completely mark another flow as $+1$[10].

6.3.2 Results

We used the same simulation scenario as in section 6.2.4 with the parameters as given in section 4.4. The foreground traffic share of the offered load $\frac{\lambda_{FT}}{\lambda}$ was varied at a fixed traffic intensity level to assess the performance of RED, DiffRED without sub-sampling ($avg_1 = avg$) and DiffRED with sub-sampling. The mean of the traffic intensities for the examples is $\bar{\rho} = 0.9521$ with standard deviation of $\sigma_\rho = 0.0013$ (the differences in the traffic intensity levels are due to the changing distribution of flow types and thus traffic patterns). The distributions of flows range from 20/7/1 H/D/voice flows at $\frac{\lambda_{FT}}{\lambda} = 0.01$ to 9/3/32 H/D/voice flows at $\frac{\lambda_{FT}}{\lambda} = 0.5$, where "H" and "D" flows constitute the BT fraction and "voice" flows are foreground traffic as

[9]Another option, yet with significantly higher overhead, would be to identify and deny access to the misbehaving flows.

[10]It should also be noted that, while we only consider the packet level in this chapter, to be completely fair monitoring of the packet sizes would be necessary.

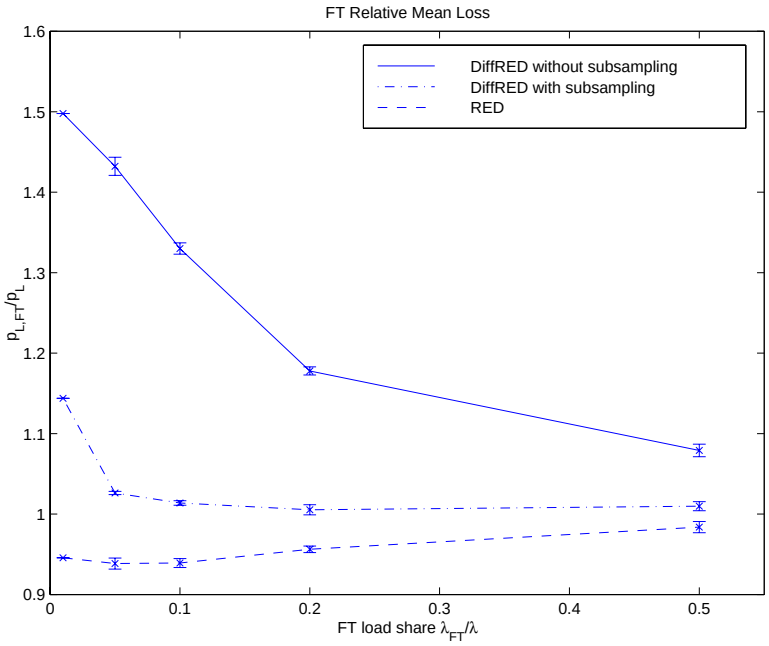


Figure 6.17: Foreground traffic relative mean loss rate

described in section 4.4. The (Diff)RED parameters used for all simulations are as follows: $min_{th} = 5$, $max_{th} = 15$, $max_p = 0.1$, $w_q = w_{q,1} = 0.002$. The queue size is 20 packets.

Fig. 6.17 shows the average of the mean loss rates of the FT flows $p_{L,FT}$ normalized with the mean loss rate calculated over all traffic p_L. It can be seen that for DiffRED without sub-sampling, the algorithm drops significantly more packets of the FT flows, due to the missing correlation of the avg and thus p_{+1} and p_{-1} values between consecutive FT arrivals. With sub-sampling however the FT flows receive a mean loss rate just above p_L except for very low FT shares. For plain RED the figure shows that the algorithm is biased slightly against the non-adaptive bursty H-type BT traffic and thus is in favor of the non-bursty FT traffic ($\frac{p_{L,FT}}{p_L} < 1$), an effect which decreases with increasing FT share (see also Fig. 6.2 and cf. [FJ93], section 9, for an analysis of RED in the presence of bursty adaptive (TCP) traffic). Fig. 6.18 shows the described properties in terms of the H-type BT traffic. We obtained the same utilization with any of the three algorithms. This is expected, because all three algorithms use the same minimum and maximum threshold parameters and the behavior when $min_{th} < avg < max_{th}$ in terms of the *aggregate* traffic seen over time intervals significantly larger than flow burst intervals is identical.

Figs. 6.19 and 6.20 show the conditional loss rates $p_{L,cond,FT}$ and $p_{L,cond,H-BT}$ for the foreground and H-type background traffic respectively. Here we give the absolute values as we cannot reasonably define a $p_{L,cond}$ value for the entire system (across different flow types with different traffic envelopes). In the given scenario we can decrease the conditional loss rate for FT traffic by at least two orders of magnitude

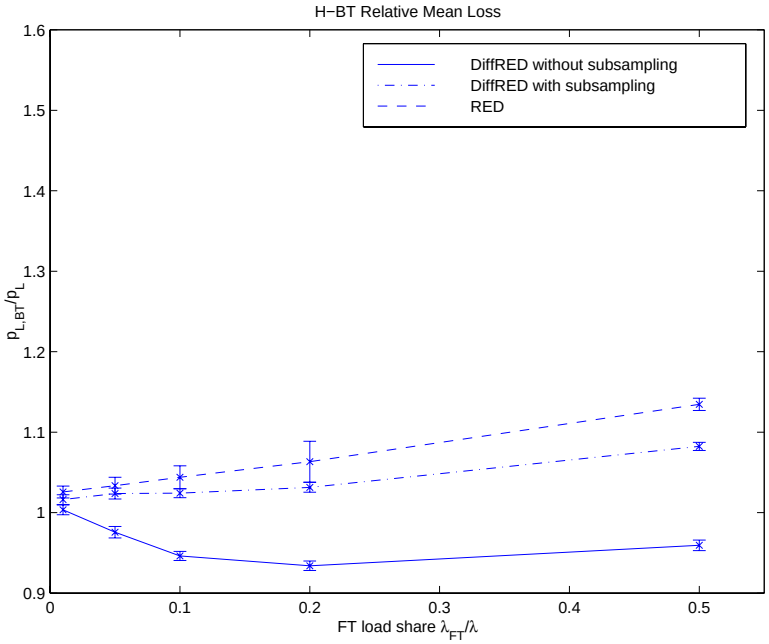


Figure 6.18: Background traffic relative mean loss rate

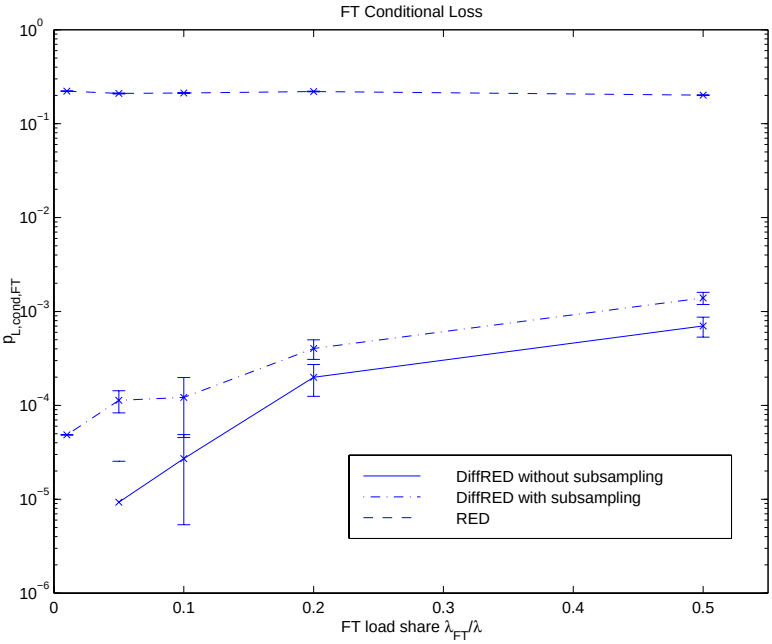


Figure 6.19: Foreground traffic conditional loss rate

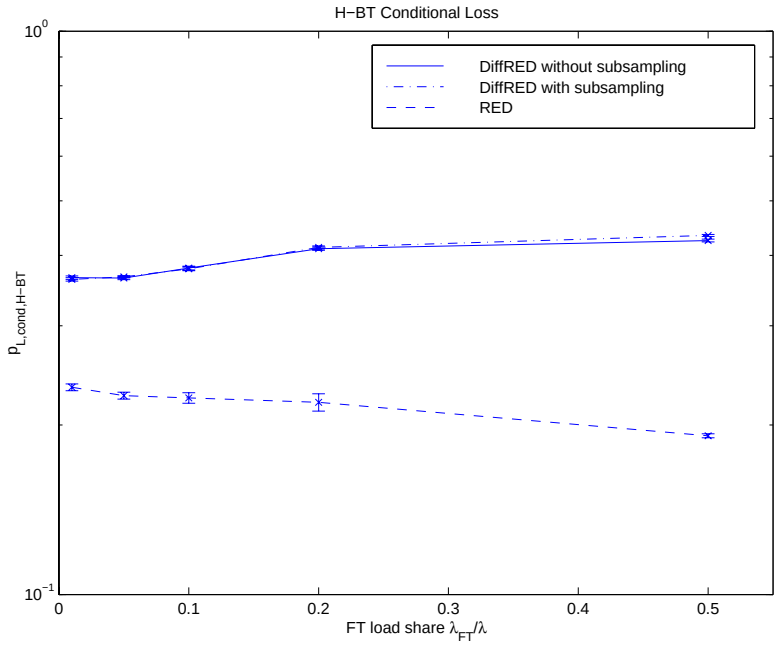


Figure 6.20: Background traffic conditional loss rate

by employing DiffRED instead of RED (Fig. 6.19). $p_{L,cond,FT}$ is increasing with the flow share because for an increasing number of voice flows we have a higher probability that bursts of +1 packets arrive which might drive the avg just over the max_{th} limit (where p_{+1} jumps from 0 to 1)[11].

Apart from the overhead of keeping an additional average queue size (avg_1)[12], the cost of employing DiffRED can be seen in Fig. 6.20 as we now impose a higher conditional loss rate on the (non-adaptive) background traffic. In DiffRED a (burst of) +1 packet(s) has a direct impact on the conditional loss probability of a BT flow. In the detection approach proposed in [SC98], we have directly associated +1/-1 events, i.e. an +1 packet is only protected if an -1 packet which can be dropped at once instead is already present in the queue. Thus the loss processes of the FT and BT packets are less correlated. The disadvantages are, however, a potentially larger buffer requirement, the dropping of already queued traffic (including the overhead of searching in the queue) and higher resulting FT conditional loss rate.

6.4 Comparison between PLoP and DiffRED

We use the same simulation environment as in sections 6.2.4 and 6.3.2 with the parameters as given in section 4.4, using a network of several hops (Fig. 4.20).

[11]Note that the allowed burstiness for +1 packets can be controlled with the $w_{q,1}$ parameter.

[12]Plus keeping the low-pass filtered arrival values, if the correction as described in section 6.3.1 is enabled.

Foreground traffic consists of several flows which have voice data characteristics (to enable DiffRED operation, every voice source marked its packets alternately with $+1$ and -1). The foreground traffic share of the offered load $\frac{\lambda_{FT}}{\lambda}$ was set to 10%. Details of the background traffic are described in section 4.4. The traffic intensity at every hop is fixed at $\rho = 1.0$. In the following, four algorithms are evaluated (the queue length is 20 packets for all algorithms):

- **Drop Tail** (DT) (as a reference),
- **Predictive Loss Pattern** (PLoP, section 6.2),
- **Optimal Predictive Loss Pattern** (OPLP) and
- **Differential RED** (DiffRED, section 6.3).

The OPLP algorithm works exactly as the PLoP algorithm, however it keeps state about the sequence numbers of packets of a flow seen (see section 6.2). OPLP thus gives a good impression where the performance limit of algorithms working purely locally (without inter-hop communication by e.g. packet marking) is, yet this algorithm does not seem viable in real high-speed network environments (due to performance and security constraints).

6.4.1 Results

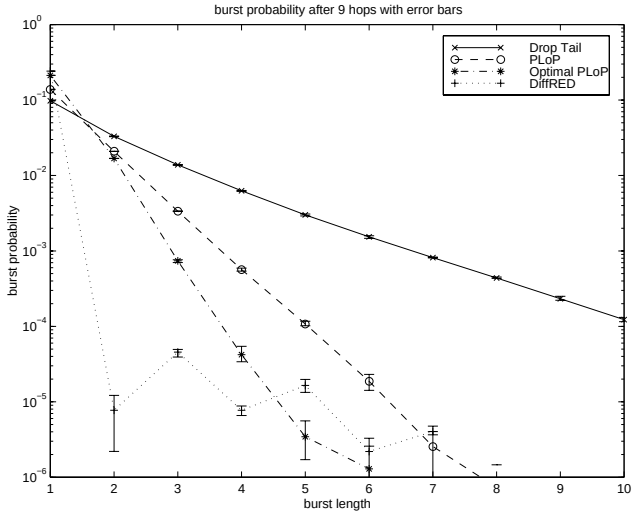


Figure 6.21: Burst loss rate $p_{L,k}$ as a function of burst length k after nine hops

Figure 6.21 shows the burst loss rate $p_{L,k}$ dependent on the burst length k for the nine-hop topology. The featured results are the mean values of all FT flows. We also plot error bars giving the standard deviation for the averaged values (this is to verify that every flow of a group has identical behavior seen over the entire simulation time).

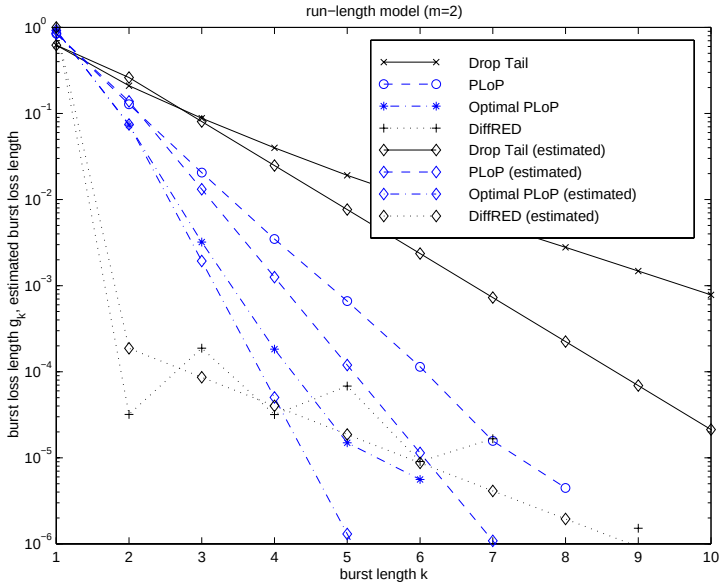


Figure 6.22: Comparison of actual and estimated burst loss length rate as a function of burst length k after 9 hops: three state run-length-based model

We can observe that DiffRED shapes the burst probability curve in the desired way (the ideal behavior would be the occurrence of only isolated losses which can be expressed with $clp = 0$ in terms of Gilbert model parameters; see Equation 6.1). Most of the probability mass is concentrated at isolated losses ($k = 1$) and all burst probabilities for $k > 1$ are at least three order of magnitude smaller. The other three algorithms show (roughly) only a geometrically decreasing burst loss probability with increasing burst length (with different slopes demonstrating the quality versus state tradeoff). Thus, considering voice as the foreground traffic of interest, with DiffRED a large number of short annoying bursts can be traded against a larger number of isolated losses as well as very long loss bursts. Avoiding longer loss bursts (which are perceived as outages) is however better achieved by PLoP and OPLP.

In section 4.5 we have summarized the conditions for which a simple two-state (Gilbert) model is sufficient to describe the loss process. As we now compare novel queue management mechanisms, it is interesting to evaluate the accurateness of run-length-based models of different order. Recall that probability for a certain burst loss length can be estimated using a Gilbert model as $\hat{P}(Y = k) = clp^{k-1}(1 - clp), 0 < k < m$ (Eq. 4.5). For a three-state model the corresponding formula is (Eq. 4.4):

$$\hat{P}(Y = k) = \begin{cases} 1 - p_{12}: & k = 1 \\ p_{12}\ p_{22}^{k-2}\ (1 - p_{22}): & 2 \leq k < m \end{cases}$$

Table 6.1 shows the parameter values for the three-state model computed from the simulation trace. For Drop Tail, PLoP and Optimal PLoP the values for p_{12} are close to those for p_{22}. For DiffRED however, p_{12} is several orders of magnitude

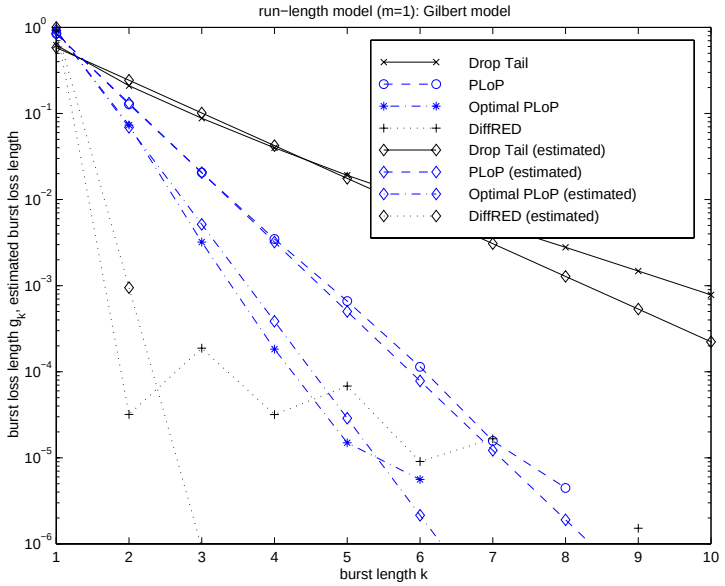


Figure 6.23: Comparison of actual and estimated burst loss length rate as a function of burst length k after 9 hops: two-state run-length-based model (Gilbert)

$\rho = 1.0$	*Drop Tail*	*PLoP*	*Optimal PLoP*	*DiffRED*
p_{01}	0.0222	0.0211	0.0251	*0.0248*
p_{12}	0.3786	0.1528	0.0766	*0.0003*
p_{22}	0.3081	0.0953	0.0259	*0.4651*
ulp	0.0368	0.0243	0.0264	0.0242

Table 6.1: Parameter values for the three state run-length-based model derived from simulation

smaller than p_{22}. This suggests that for DiffRED a three-state model characterization is appropriate whereas for the other algorithms the two-state (Gilbert) model is sufficient (note that for $p_{12} = p_{22}$ the three-state model is equivalent to the two-state one). Therefore in Table 6.2 we also present the computed parameter values for the two-state representation.

Figures 6.22 and 6.23 show the rates for the actual and the estimated burst loss lengths for a three-state ($m = 2$) and a two-state ($m = 1$, Gilbert) model respectively. We can see that the three-state model estimation as expected from the parameter values of Table 6.1 reflects the two areas of the DiffRED operation (the sharp drop of the burst loss length rate for $k = 2$ and the decrease along a geometrically decreasing asymptote for $k > 2$). This effect cannot be captured by the two-state model which thus overestimates the burst loss length rate for $k = 2$ and then hugely underestimates it for $k > 2$.

$\rho = 1.0$	*Drop Tail*	*PLoP*	*Optimal PLoP*	*DiffRED*
p_{01}	*0.0222*	*0.0211*	*0.0251*	0.0248
p_{11} (clp)	*0.4171*	*0.1561*	*0.0747*	0.0009
ulp	0.0368	0.0243	0.0264	0.0242

Table 6.2: Parameter values for the two-state run-length-based model (Gilbert) derived from simulation

Interestingly, for the other queue management methods, especially for Drop Tail, while both models capture the shape of the actual curve, the lower order model is more accurate in the estimation. This can be explained as follows: if the burst loss length probabilities are in fact close to a geometrical distribution, the estimate is more robust if all data is included (note that the run-length based approximation of the conditional loss probability p_{mm} only includes loss run-length occurrences larger or equal to m (Table 4.3): $p_{L,cond}(m) = \frac{\sum_{n=m}^{\infty}(n-m)o_n}{\sum_{n=m}^{\infty} no_n}$).

How the discussed differences between the algorithms develop along the path is shown in Figure 6.24. After the first hop both DiffRED and OPLP have almost the same behavior as expected. But after several hops the curves differ increasingly. On every hop DiffRED can protect "+1" packets by early-dropping "-1" packets thus using avg_1 as "memory" about every individual flow. OPLP even with keeping individual state on the sequence numbers can only choose among the packets currently present in the queue (the "memory" is limited to the queue size) and might not find an adequate victim (force failure). The intersection point of the DiffRED curve with the PLoP and OPLP curve moves towards longer bursts with an increasing number of hops. Note that contrary to PLoP, OPLP and DiffRED are able to observe the actual loss pattern of the flow rather than just the arrival pattern at a particular network element. So in summary DiffRED is able to decide best when to drop which packet.

For a complete discussion of the loss process influenced by the respective algorithms, we also have to look at the unconditional loss probability (obviously the ulp when using different algorithms needs to be approximately equal to allow a fair comparison between algorithms). The conditional loss probability (clp) then allows to describe the performance with regard to burst loss in a more comprehensive way (however with the limitations in terms of the modeling inaccuracy by the Gilbert model just described).

Figure 6.25 shows how the unconditional loss probability and conditional loss probability of the foreground traffic develop through the path. The ulp values for all algorithms differ only insignificantly as desired for a meaningful comparison of the burst loss properties. However the clp results are very different for every method. The DiffRED algorithm achieves the best result and shows that the flow is protected not only at every single hop but gets a path oriented protection through the packet marking. An increasing number of losses increases the ulp, but the clp decreases.

To explain this we again employ the metrics of section 4.1. When using DiffRED

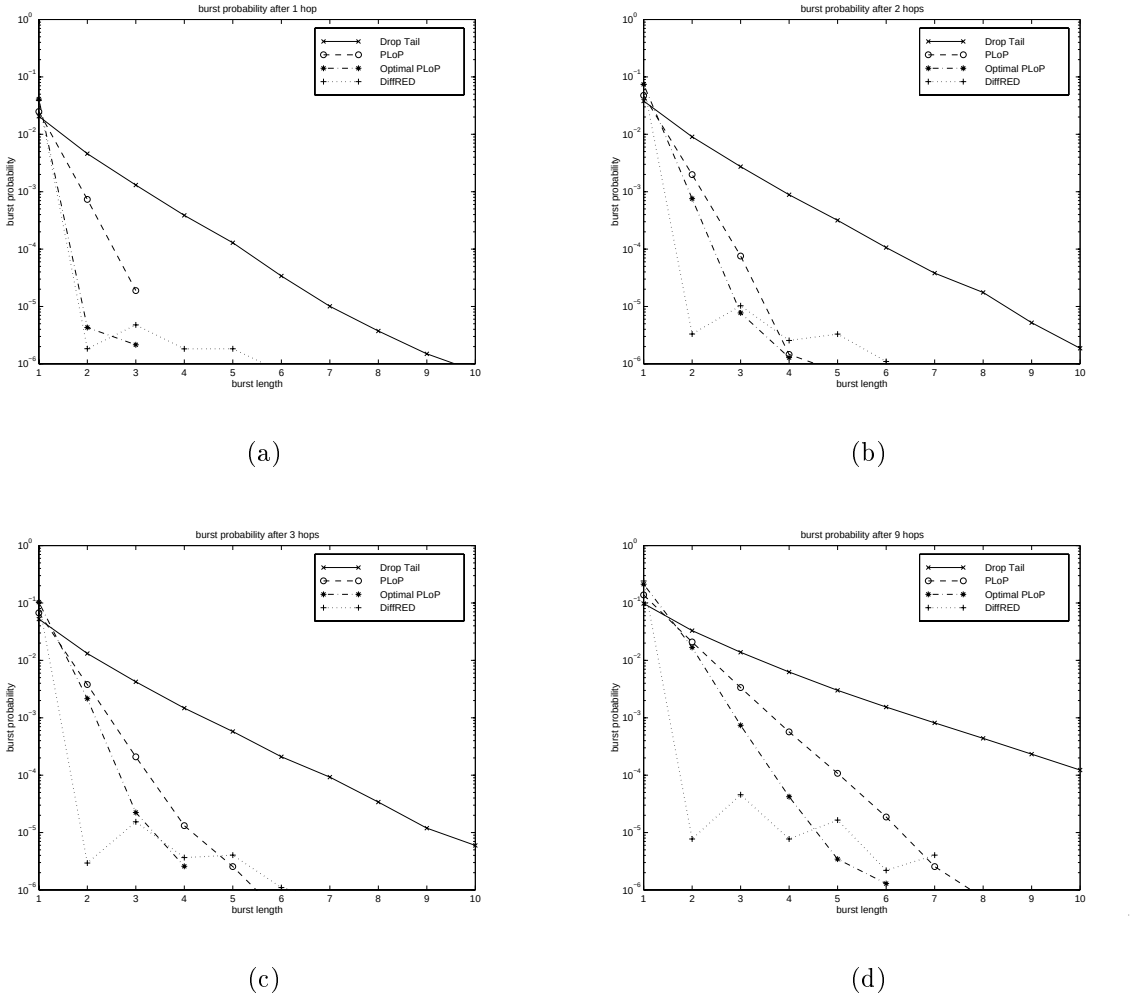


Figure 6.24: Burst loss rate as a function of burst length k after a) 1 hop, b) 2 hops, c) 3 hops, d) 9 hops

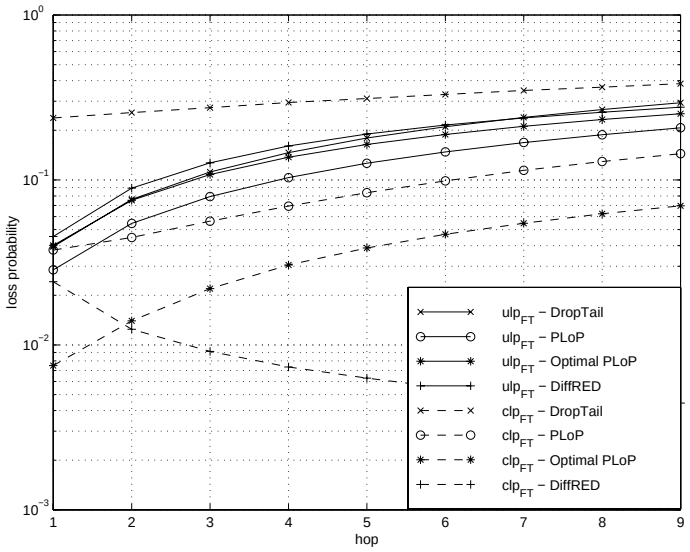


Figure 6.25: Development of FT *ulp* and *clp* on the transmission path

at every hop "+1" packets are dropped with lower probability than "-1" packets. The occurring losses are concentrated on the "-1" packets. Thus it happens that the denominator of $p_{L,cond} = \frac{\sum_{k=1}^{\infty}(k-1)o_k}{\sum_{k=1}^{\infty}ko_k}$ increases faster than the enumerator, resulting in a decreasing *clp* simultaneously to an increasing *ulp*.

Another interesting issue is the behavior of the algorithm relative to the background traffic flows, i.e. the fairness to uncontrolled traffic. Figure 6.26 shows the values for *ulp* and *clp* at every hop (we averaged the results for one flow group (H-type BT)). These values are not cumulative values but computed for only one hop because this cross traffic uses only one hop of the path respectively.

The almost identical *ulp* curve of DT, PLoP and OPLP shows that all three algorithms have only minor influence on the background traffic. The DiffRED algorithm retrieves some of its performance from the BT but at a tolerable level.

6.5 Conclusions

In section 6.1 we characterized the desired behavior of a hop-by-hop loss control algorithm in terms of the simple packet-level metrics introduced in section 4.1 under the assumption that a simple, periodic loss pattern enhances the performance of the end-to-end loss recovery. Then, several design choices and tradeoffs for loss control algorithms were identified (per-flow or per-packet signaling of participation in the scheme, per-flow or per-packet class state, local or distributed operation (section 3.2), etc.).

Section 6.2 discussed the Predictive Loss Pattern (PLoP) algorithm. PLoP reduces the conditional loss probability with limited overhead for a wide range of load conditions. If the link-speed equivalent buffer is larger than the expected maximum

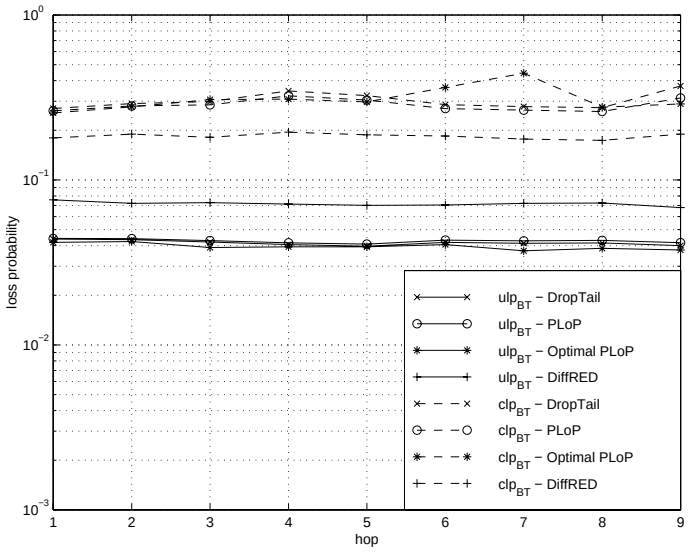


Figure 6.26: BT (cross traffic) *ulp* and *clp* values at the hops 1-9

traffic period, unfairness of the algorithm towards background traffic is avoided. The algorithm operates only during times of congestion and does not require explicit cooperation of the applications.

Then we have shown how intra-flow QoS requirements of applications can be mapped to simple differentiated packet marking which is then enforced within the network by a simple queue management mechanism, the Differential RED (DiffRED) algorithm. By extending the well-known RED algorithm to comprise additional drop probability functions, we are able to control the conditional loss characteristics of individual flows while keeping their unconditional loss probability within a controlled bound around the value expected using a conventional RED algorithm. The differences to RIO (RED with IN and OUT packets) which already employs additional drop probabilities for RED can be summarized as follows:

- "Differential" loss probability curves (a higher loss probability of one packet is compensated by a lower loss probability for another packet)
- Sub-sampling of the queue length value on FT arrival instants to allow for a fair distribution of losses between FT and BT
- Monitoring of the ratio between $+1/-1$ arrivals to adjust their loss probabilities in case a mismatch (ratio $\neq 1$) between them exists

In section 6.4 we then compared PLoP and DiffRED representing the different design choices summarized in Table 6.3. We find that both types of algorithms do not have a significant impact on conventional traffic. It is possible to fulfil the goal of controlling the conditional loss probabilities. For the given scenario algorithms using packet marking are found to be superior because a high probability for short bursts can be traded against a higher probability for isolated losses as well as higher

Predictive Loss Pattern (PLoP)	*Differential RED (DiffRED)*
mapping application requirements to *periodic* drop profiles at network nodes	sender defines acceptable (non-periodic) loss pattern
partial per-flow state (drop experiments)	packet marking (loss history)
shift drop among group of flows concurrently present in the queue	"differential" loss probability curves

Table 6.3: Comparison of PLoP and DiffRED properties

(but acceptable) probability for very long loss bursts. This is mainly due to the "memory" realized with the average queue size (the congestion indication and dropping decision is influenced by a longer term monitoring process). Furthermore, with packet marking non-periodic loss patterns can be realized which seems particularly interesting with regard to the results on the loss impact for frame-based codecs (section 5.2.3). Thus, marking-based algorithms also allow for an explicit cooperation of the end-to-end and the hop-by-hop algorithm.

Chapter 7

Combined End-to-End and Hop-by-Hop Loss Recovery and Control

In the absence of any hop-by-hop loss control support, we have used the loss concealment schemes (AP/C and the G.729 loss concealment) together with Forward Error Correction (sections 5.1.5.2 and 5.2.4 respectively). FEC, however, requires additional data to be sent (thus increasing the network load) and itself is vulnerable to loss and loss correlation (section 3.1.2.2). As such FEC schemes are not adaptive (and cannot be adaptive due to the inflexible codecs) they have to be classified as inter-flow QoS (Table 1.1). Therefore we now aim at linking the developed end-to-end schemes with the intra-flow hop-by-hop loss control support mechanisms. For AP/C this is simple (section 7.1) as AP/C requires only implicit cooperation (the loss pattern is crucial, not which particular packet is lost). With regard to our approach for frame-based codecs, in section 7.2 we present how to explicitly map the pattern of essential and non-essential packets onto network prioritization, thus avoiding the addition of redundancy. While this approach enables both intra- and inter-flow loss protection, we particularly highlight the intra-flow QoS aspect.

Similarly to simulations in earlier chapters, we use the same speech sample containing different male and female voices for each loss condition as input to our simulation. We employ different seeds for the pseudo-random number generator to generate different loss patterns (for the results presented here we used 300 patterns for each simulated condition). This is allows on one hand to employ a simple model characterization rather than a large number of traces of a discrete event simulations and on the other hand takes into account that the input signal is not homogeneous (i.e. a loss burst within one segment of that signal can have a largely different perceptual impact than a loss burst within another segment).

$\rho = 1.0$	*Drop Tail*	*PLoP*	*Optimal PLoP*	*DiffRED*
p_{01}	0.0222	0.0211	0.0251	0.0248
p_{11} (clp)	0.4171	0.1561	0.0747	-
p_{12}	-	-	-	0.0003
p_{22}	-	-	-	0.4651

Table 7.1: Parameter values for the two- and three state run-length-based model derived from simulation ($\rho = 1.0$)

$\rho = 2.0$	*Drop Tail*	*PLoP*	*Optimal PLoP*	*DiffRED*
p_{01}	0.4201	0.2238	0.5093	0.3026
p_{11} (clp)	0.5032	0.1743	0.0123	-
p_{12}	-	-	-	0.0001
p_{22}	-	-	-	0.4540

Table 7.2: Parameter values for the two- and three state run-length-based model derived from simulation ($\rho = 2.0$)

7.1 Implicit cooperation: Hop-by-Hop support for AP/C

In sections 5.1.3.1 and 5.1.3.2 we have evaluated the AP/C scheme. There we have found a significant dependence of the performance of the scheme on the conditional loss probability (clp). This implies that a simple periodic pattern of alternating losses has much less impact on the resulting speech quality than bursty losses when loss concealment is enabled. Then, in chapter 6 we developed intra-flow hop-by-hop loss control algorithms and assessed their ability to control the conditional loss probability. Therefore, through the separation introduced by the end-to-end loss model, it is possible to directly link the separate performance evaluation of the hop-by-hop loss control with the results of perceptual speech of AP/C.

But as we have also seen in section 6.4.1, for the DiffRED algorithm a higher order model characterization is reasonable. Therefore in Table 7.1 we summarize the model parameters computed from the simulation results (Tables 6.1 and 6.2 in section 6.4.1). Additionally we also present the derived model parameters for an overload scenario (traffic intensity $\rho = 2.0$) in Table 7.2. Note that while AP/C does not need explicit cooperation from the network, for DiffRED operation alternating packet marking needs to be enabled. However, contrary to the presented SPB-MARK algorithm in the next section, the entity doing the marking (e.g. a first hop router) does not need to be aware of the packet payload content.

In Figure 7.1 the perceptual distortion as evaluated by EMBSD is shown for the simulation with the parameter sets of section 5.1.3.1, as well as the model parameters of Tables 7.1 and 7.2. For silence substitution, the behavior of all four algorithms

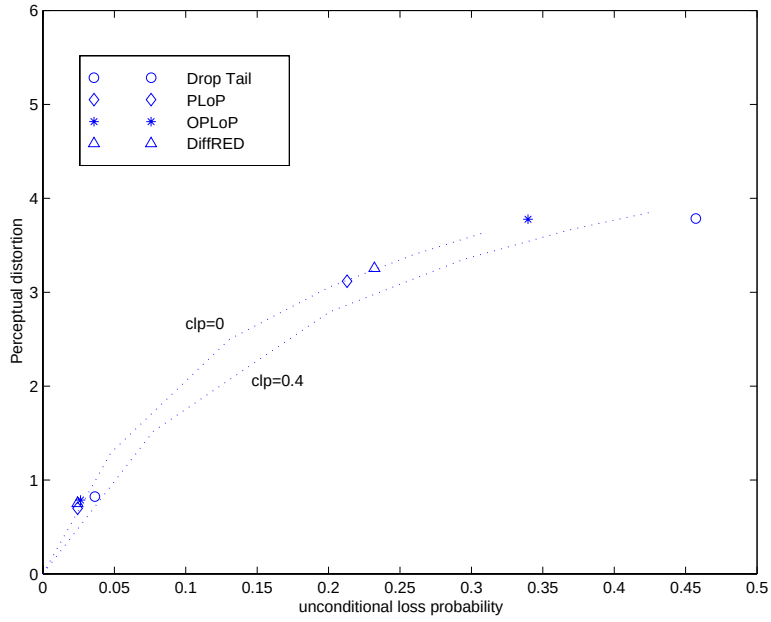


Figure 7.1: Perceptual Distortion (EMBSD) of silence substitution using different loss control algorithms

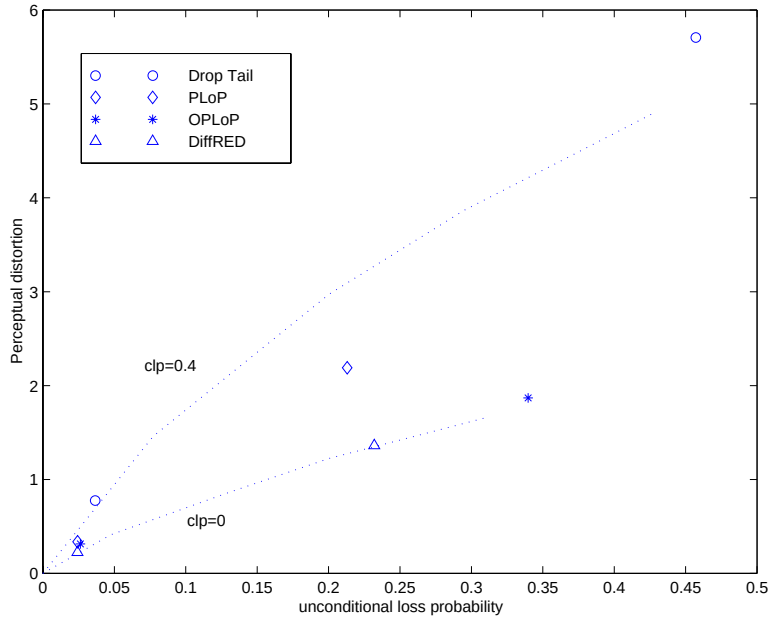


Figure 7.2: Perceptual Distortion (EMBSD) of AP/C using different loss control algorithms

does not have any measurable influence on the speech quality as expected. The measurement points for $\rho = 1.0$ are very close to each other at a $ulp < 0.05$. For $\rho = 2.0$, however, an effect as in Figure 6.6 is visible: the algorithms become increasingly unfair, i.e. foreground traffic packets see a lower loss probability than background traffic.

Figure 7.2 confirms for $\rho = 1.0$ that all three algorithms (PLoP, Optimal PLoP and DiffRED) perform better than Drop Tail. Again, in the overload situation the unfairness when compared to Drop Tail is visible. While both DiffRED and Optimal PLoP are able to maintain a near optimal clp, Optimal PLoP is a lot fairer towards background traffic (i.e. the ulp is closer to the drop tail case) in the overload case.

For PLoP, as the reason for the unfairness the asymptotic behaviour due to the bound on the ulp has already been mentioned (section 6.2.4.1): $max(ulp) = 0.5$ for PLoP versus $max(ulp) = 1$ for DropTail. For RED (on which DiffRED is based) we have already mentioned the bias in favour of the FT against the bursty BT (Figs. 6.2 and 6.17). Furthermore with DiffRED it should be noted that for avg_1 values just around max_{th} a similar difference in the asymptotic behaviour as for PLoP exists:

$$\begin{aligned} \lim_{avg_1 \to max_{th}-0} ulp &= 0.5 \\ \lim_{avg_1 \to max_{th}+0} ulp &= 1 \end{aligned}$$

7.2 Explicit cooperation: Speech Property-Based Packet Marking

In the paragraph on receiver adaptation (p. 43) and in section 3.3.2 we have mentioned that certain codecs (sub-band codecs) or PCM packetization schemes generate packets of different importance with regard to the expected perceptual quality when a packet is lost. In this section we present such an approach of explicitly mapping end-to-end knowledge on the hop-by-hop packet transmission. However we use (as in chapter 5.2) a standardized speech codec which has not been designed to emit a layered data stream. Therefore the importance of packets is deduced using the SPB algorithm introduced in section 5.2.4 ([SLW00, SLC00]). As the underlying queue management we use the Differential RED algorithm as it has shown superior performance and is the only one which can support non-periodic patterns when compared to the other considered algorithms. Therefore we first derive a simple end-to-end loss model for a DiffRED enhanced network. Sections 7.2.2 and 7.2.3 then present the simulation scenario and the results using different end-to-end loss recovery schemes.

7.2.1 A simple End-to-End model for DiffRED

In section 7.1 it has been feasible to directly merge the results of sections 5.1.3.1 and 6.4.1 due to the periodic (alternating) pattern used for both evaluations. Now, however, we consider sources which can emit bursts of packets belonging to a single

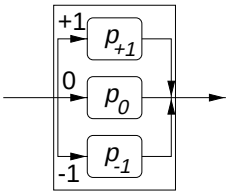


Figure 7.3: "Class-Bernoulli" model for DiffRED.

class. Therefore within an end-to-end model (cf. section 1.3) we need to explicitly associate a drop probability with a single packet.

We use a separate one-state Markov model (Bernoulli model, cf. section 4.1.4) to describe the network behavior as seen by each class of packets. "Best effort" packets (designated by "0" in Fig. 7.3) are dropped with the probability p_0, whereas packets marked with "+1" and "-1" are dropped with probabilities of p_{+1} and p_{-1} respectively. This is a reasonable assumption with regard to the interdependence of the different classes in fact, as sections 6.3 and 6.4 have shown that DiffRED achieves a fair amount of de-correlation of +1 and -1 packet losses.

So if we first consider an alternating initial marking sequence {+1,-1,+1,-1,...} again, the loss of exactly one packet ($p_{L,1}$, corresponding to $P(X = 1)$) can be expressed with p_{+1} and p_{-1}. Here we use the fact that considering the initial marking only two loss patterns for any burst length may occur: $(l_{-1}(s-1), l_{+1}(s) = 10)$ and $(l_{+1}(s-1), l_{-1}(s) = 10)$ (we employ the terminology used in section 4.1.1 again whereby additionally the index of the loss indicator functions designates the class association (-1 / $+1$) of the packet). So for the probability $P(X = 1)$ estimated with p_{+1} and p_{-1} we have:

$$\begin{aligned} \hat{P}(X=1) &= p_{+1}(1-p_{-1}) + p_{-1}(1-p_{+1}) \\ &= p_{+1} + p_{-1} - 2p_{+1}p_{-1} \end{aligned} \tag{7.1}$$

Similar computations of estimates for $p_{L,2}$ and $p_{L,3}$ apply:

$$\begin{aligned} \hat{P}(X=2) &= p_{-1}p_{+1}(1-p_{-1}) + p_{+1}p_{-1}(1-p_{+1}) \\ &= p_{+1}p_{-1}(2-p_{+1}-p_{-1}) \end{aligned} \tag{7.2}$$

$$\begin{aligned} \hat{P}(X=3) &= p_{+1}p_{-1}p_{+1}(1-p_{-1}) + p_{-1}p_{+1}p_{-1}(1-p_{+1}) \\ &= p_{+1}^2p_{-1} + p_{-1}^2p_{+1} - 2p_{-1}^2p_{+1}^2 \end{aligned} \tag{7.3}$$

Within the 0 and -1 classes there is of course loss correlation due the low-pass filtered queue length in connection with non-periodic (bursty) arrivals of SPB-marked packets. However due to the association of packets to classes according to their perceptual importance the impact of loss correlation here is far less significant than for a conventional queuing discipline like Drop Tail seen over the entire flow. Additionally there is loss correlation in all classes under heavy overload when avg and avg_1 are larger than max_{th} (section 6.3.2).

The effect of loss correlation within the classes can be seen for the simulation in Figure 7.4 (we repeat here partially the content of Figure 6.22). In the simulation, the frequency for the loss of exactly three packets $p_{L,3}$ is larger than $p_{L,2}$. However, for the run-length model due to equation 4.4 clearly $\hat{P}(X=3) \leq \hat{P}(X=2)$ is the case. Note that also for our "class-Bernoulli" model, using equations 7.2 and 7.3, it is easy to show by contradiction that $\hat{P}(X=3) \leq \hat{P}(X=2)$ always holds :

$$\begin{aligned} \hat{P}(X=3) &> \hat{P}(X=2) \\ p_{+1}p_{-1}(p_{-1}+p_{+1}-2p_{+1}p_{-1}) &> p_{+1}p_{-1}(2-p_{+1}-p_{-1}) \\ p_{-1}+p_{+1}-p_{+1}p_{-1} &\not> 1 \;\forall\, p_{-1}, p_{+1} \in [0,1] \end{aligned}$$

So the deviation of the models from the simulation results is due also to the loss correlation within the classes which is not captured by both models (for the class-Bernoulli model that means that the actual probability $P(l_{-1}(s-3), l_{+1}(s-2), l_{-1}(s-1), l_{+1}(s) = 1110)$ is larger than the estimated $p_{-1}p_{+1}p_{-1}(1-p_{+1})$ term within equation 7.3).

To complete the simple DiffRED model, the relationship between the "-1" and "+1" drop probabilities can be derived as follows: Let $a = a_0 + a_{+1} + a_{-1}$ be the overall number of emitted packets by that flow and $a_x, x \in [-1, 0, +1]$ be the number of packets belonging to a certain class. Then, with $a_{+1} = a_{-1} = a_{|1|}$ and considering that the resulting service has to be best effort in the long term, we have:

$$a_0 p_0 + a_{+1}p_{+1} + a_{-1}p_{-1} \stackrel{!}{=} a p_0$$

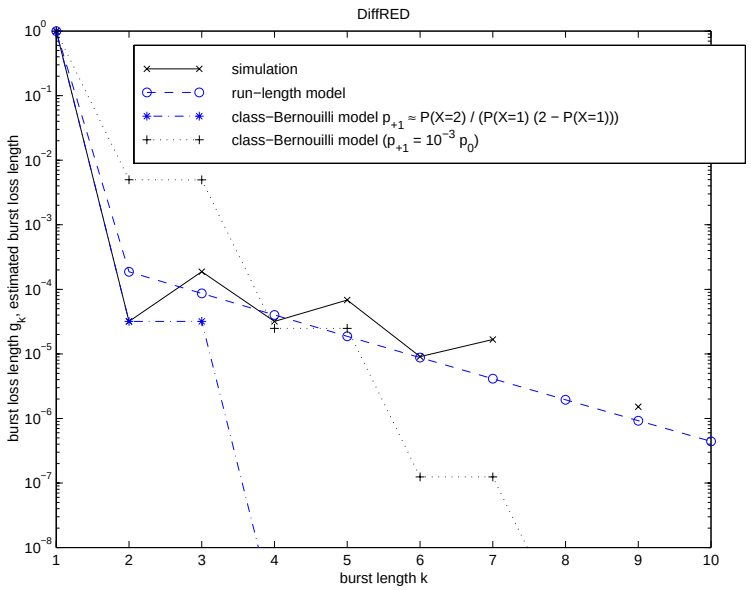


Figure 7.4: Comparison of actual and estimated burst loss length rate of DiffRED as a function of burst length k after 9 hops

$$\begin{aligned} a_{|1|}(p_{+1} + p_{-1}) &= (a - a_0)p_0 \\ a_{|1|}(p_{+1} + p_{-1}) &= 2a_{|1|}p_0 \\ p_{-1} &= 2p_0 - p_{+1} \end{aligned}$$

In our simulations we will use the drop probability p_0 as the variable parameter. Therefore we need to determine a reasonable value for the drop probability within the +1 class (p_{+1}) derived from the simulation results. Using $p_{+1} \ll p_{-1}$ and $2p_{+1} \ll 1$ as approximations we get from equation 7.1:

$$\begin{aligned} \hat{P}(X = 1) &\approx p_{-1}(1 - 2p_{+1}) \\ &\approx p_{-1} \end{aligned} \tag{7.4}$$

For $\hat{P}(X = 2)$ we have (considering $p_{+1} \ll p_{-1}$) from equation 7.2:

$$\hat{P}(X = 2) \approx p_{+1}p_{-1}(2 - p_{-1}) \tag{7.5}$$

Using both equations 7.4 and 7.5 we can compute p_{+1} for this simulation as:

$$p_{+1} \approx \frac{\hat{P}(X = 2)}{\hat{P}(X = 1)(2 - \hat{P}(X = 1))}$$

With the actual values from the simulation $p_{L,1} = 0.2420$ and $p_{L,2} = 7.72\ 10^{-6}$ we get $p_{+1} \approx 2\ 10^{-5}$. Using theses values, Figure 7.4 gives the estimated burst loss length rate as a function of the burst length for the class-Bernoulli model. To accomodate for the inaccuracy of the class-Bernoulli model, to be more pessimistic concerning the +1 bursts and as well to include some correlation between the classes we have set $p_{+1} = 10^{-3}\ p_0$ for the subsequent simulations (Figure 7.4). This should allow a reasonable evaluation of how losses in the +1 class affect the performance of the SPB-algorithms.

7.2.2 Simulation description

We compare two flavors of "differential" marking schemes: in the first one (ALT-DIFFMARK, Figure 7.5) packets are alternatingly marked as being either "-1" or "+1". The second scheme (SPB-DIFFMARK) is driven by an SPB marking algorithm (Figure 7.6). SPB gives only a binary marking decision ("essential" or "normal" packet). Therefore, we employ a simple algorithm to send the necessary "-1" packets for compensation (Figure 7.6, cf. Figure 5.26): after a burst of "+1" packets has been sent, a corresponding number of "-1" packets is sent immediately. State about the necessary number of to-be-sent "-1" packets is kept in the event that the SPB algorithm triggers the next "+1" burst before all "-1" packets necessary for compensation are sent. Thus seen over time intervals which are long compared to the $+1/-1$ burst times, the mean loss for the flow will be equal to the "best effort" case.

We will also evaluate related inter-flow loss protection. The first scheme uses full protection (FULL MARK, all packets are marked as "+1"). Packets of flows using

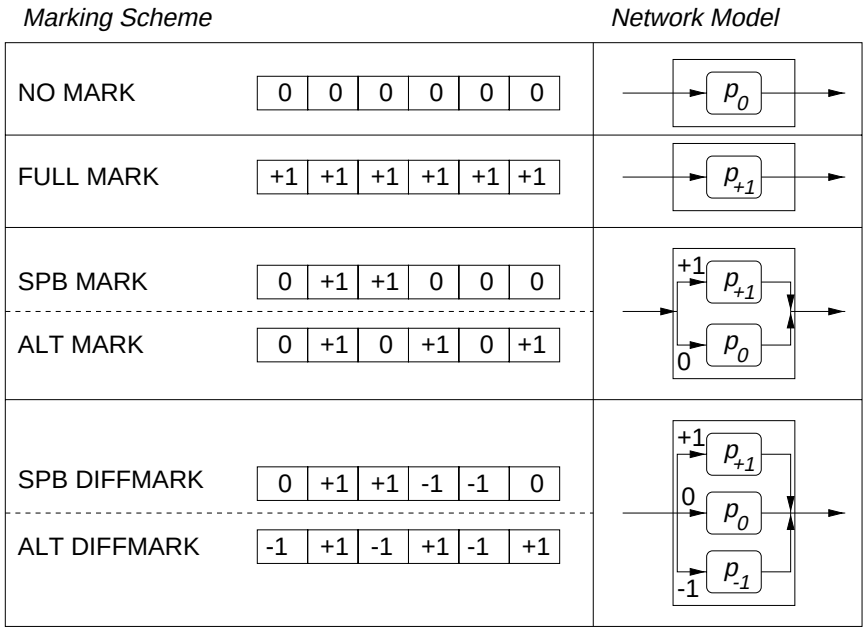


Figure 7.5: Marking schemes and corresponding network models.

the SPB-MARK scheme will either see p_{+1} (Fig. 7.5) or the drop probability p_0 (the algorithm is similar to the one depicted in Fig. 7.6 without the "-1" compensation enabled). For comparison we again use a scheme where packets are alternatingly marked as being either "0" or "+1" (ALT-MARK). Finally, packets of pure "best effort" flows are dropped with the probability p_0 (NO MARK case in Fig. 7.5).

We simulate the transmission of G.729 voice flows using packets containing two frames (i.e. $20ms$ speech segments). The simulated network is then applied to voice data flows using the proposed marking schemes: the drop probability parameter p_0 is varied in constant steps to obtain an impression on the sensitivity and expected range of the objective quality measurements' result values. For the SPB marking schemes the percentage of "+1"-marked packets was 40.4% for the speech material used. We obtained similar marking percentages for other speech samples. The ALT marking schemes mark exactly 50% of their packets as being "+1". The resulting voice data streams are decoded. These decoded speech signals are then evaluated by the objective quality measures.

7.2.3 Results

Figures 7.7 and 7.8 show the auditory distance/perceptual distortion evaluated by the MNB and EMBSD algorithm respectively. The results of MNB and EMBSD for the unprotected flows (Figure 7.7 and Figure 7.8: "NO MARK") show that with increasing p_0 in the network model (and thus increasing packet loss rate and loss correlation), the auditory distance (in case of MNB) and the perceptual distortion (in case of EMBSD) are increasing significantly, i.e. the speech quality of the decoded

```
protect = 0
foreach (k frames)
    classify = analysis(k frames)
    if (protect > 0)
        if (classify == unvoiced)
            protect = 0
            if (compensation > 0)
                compensation = compensation−k
                send(k frames, "−1")
            else
                send(k frames, "0")
            endif
        else
            send(k frames, "+1")
            protect = protect−k
            compensation = compensation+k
        endif
    else
        if (classify == uv_transition)
            send(k frames, "+1")
            protect = N − k
            compensation = compensation+k
        else
            if (compensation > 0)
                compensation = compensation−k
                send(k frames, "−1")
            else
                send(k frames, "0")
            endif
        endif
    endif
endfor
```

Figure 7.6: SPB-DIFFMARK pseudo code

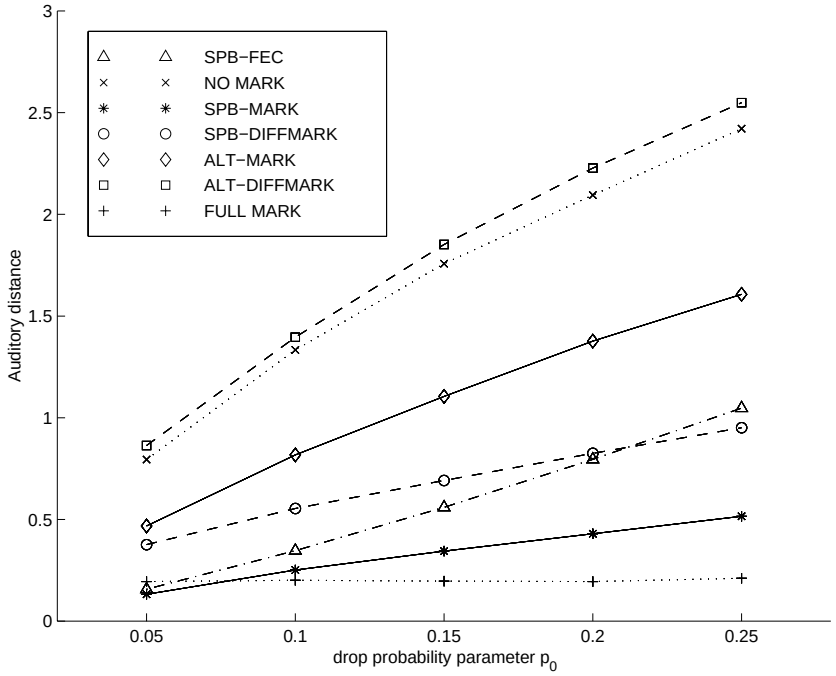


Figure 7.7: Auditory Distance (MNB) for the marking schemes and SPB-FEC

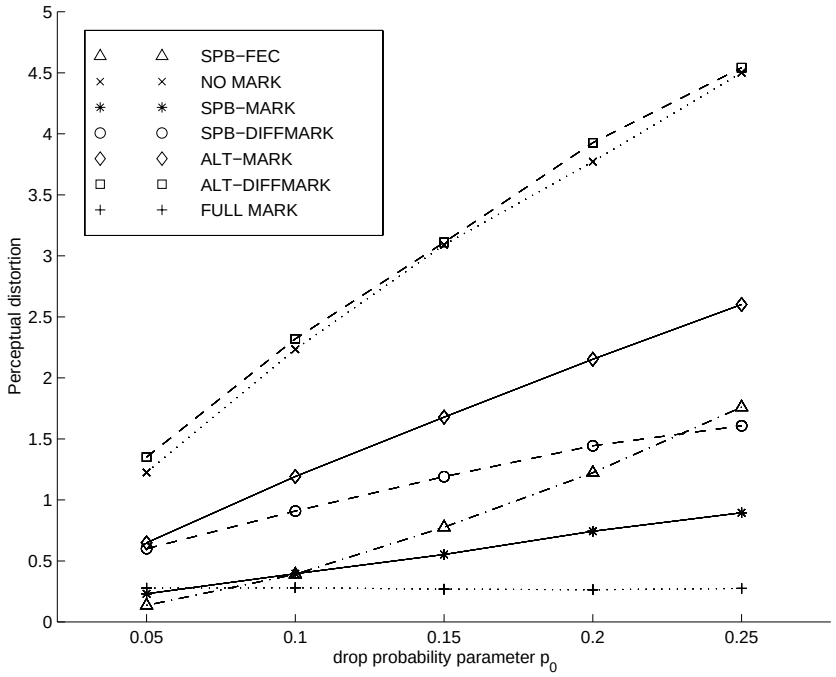


Figure 7.8: Perceptual Distortion (EMBSD) for the marking schemes and SPB-FEC

speech signals is decreasing. When comparing the "NO MARK" results to the curves when marking is enabled, we can see that the decoded speech signal without marking has the highest auditory distance (in case of MNB) and the highest perceptual distortion (in case of EMBSD) and thus the worst speech quality.

The differential marking scheme (SPB-DIFFMARK) offers a significantly better speech quality even when only using a network service which amounts to "best effort" in the long term. Note that the ALT-DIFFMARK marking strategy does not differ from the "best effort" case. SPB-DIFFMARK is also even better than the inter-flow QoS ALT-MARK scheme, especially for higher values of p_0. These results validate the strategy of our SPB marking schemes that do not equally mark all packets with a higher priority but rather protect a subset of frames that are essential to the speech quality.

The SPB-FEC scheme (section 5.2.4), which uses redundancy to protect a subset of the packets, enables a very good output speech quality for low loss rates. However, it should be noted that the amount of data sent over the network is increased by about 40%. Note that the simulation presumes that this additionally consumed bandwidth itself does not contribute significantly to congestion. This assumption is only valid if a small fraction of traffic is voice. Podolsky et al. ([PRM98]) evaluated the performance of FEC schemes, considering the impact of adding FEC for the voice fraction on the network load. They have shown that if an increasing number of flows uses FEC, the amount of FEC has to be carefully controlled, otherwise adding FEC can be detrimental to overall network utilization and thus the resulting speech quality. They used however theoretic rate-distortion curves not backed by either subjective testing or objective speech quality measurements. The SPB-FEC curve is convex with increasing ulp, as due to the increasing loss correlation an increasing number of consecutive packets carrying redundancy is lost leading to unrecoverable losses. The curve for SPB-DIFFMARK is concave however, yielding better performance for $p_0 \gtrapprox 0.2$.

The inter-flow QoS ALT-MARK scheme (50% of the packets are marked) enhances the perceptual quality. However, the auditory distance (in case of MNB) and the perceptual distortion (in case of EMBSD) of the SPB-MARK scheme (with 40.4% of all packets marked) is significantly lower and very close to the quality of the decoded signal when all packets are marked (FULL MARK). This also shows that by protecting the entire flow only a minor improvement in the perceptual quality is obtained. The results for the FULL MARK scheme also show that, while the loss of some of the +1 packets has some measurable impact, the impact on perceptual quality can still be considered to be very low.

Figures 7.9 and 7.10 give the results for the NO MARK, SPB-DIFFMARK and FULL MARK marking schemes with the standard deviation as error bars. Interesting here is the the significantly lower variance of the results based on MNB. The results for the FULL MARK scheme show that while p_{+1} is increasing with p_0 a decrease in speech quality as a consequence is not measurable.

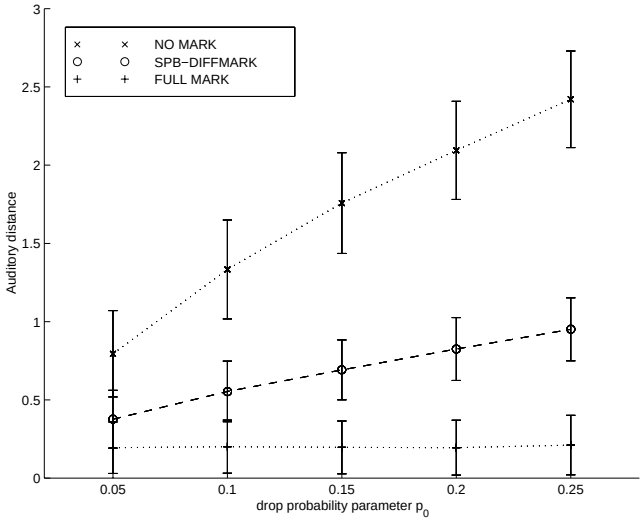


Figure 7.9: Variability of the Auditory Distance (MNB) for the marking schemes

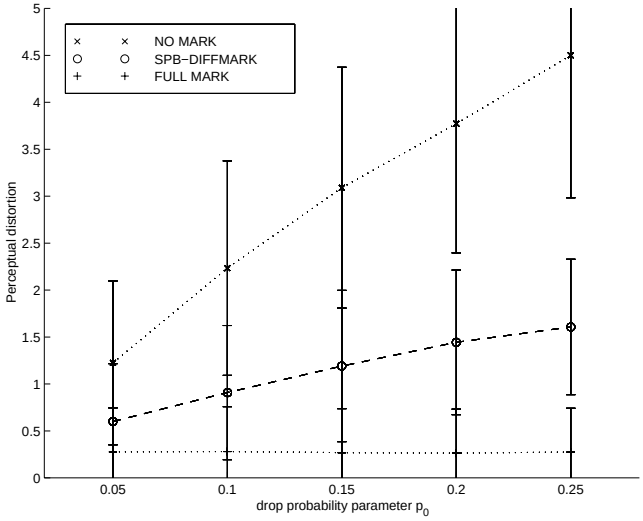


Figure 7.10: Variability of the Perceptual Distortion (EMBSD) for the marking schemes

7.3 Conclusions

For implicit cooperation of end-to-end loss recovery and hop-by-hop loss control the performance evaluation is separable. The end-to-end scheme needs a certain network characteristic which can be described by end-to-end model parameters and the hop-by-hop mechanism provides a service optimizing this metric. In section 7.1 we have compared different model parameter sets derived from the evaluation of the particular hop-by-hop loss control schemes. We have confirmed that the DiffRED algorithm provides the best performance of the evaluated algorithms. However we have also seen that all algorithms show some unfairness under heavy overload conditions.

In section 7.2 we have adopted the DiffRED algorithm for hop-by-hop loss control as it has shown superior performance and is the only one which can support non-periodic patterns when compared to the other considered algorithms. We have then developed speech property-based marking schemes which protect the voiced frames that are essential to the speech quality by marking the packets carrying them with a higher priority while relying on the decoder's concealment in case other lower priority packets are lost. Simulations using a simple network model and subsequent evaluation using objective quality measures show that the "differential" (intra-flow QoS) packet marking scheme SPB-DIFFMARK performs much better than the conventional best effort service, requiring only per-hop control over the loss patterns rather than the loss rates in connection with a simple end-to-end algorithm. The (inter-flow QoS) SPB-MARK scheme performs almost as good as the protection of the entire flow at a significantly lower number of necessary high-priority packets. All proposed marking schemes can be realized within the IETF Differentiated Services architecture.

Thus, combined intra-flow end-to-end and hop-by-hop schemes seem to be well-suited for heavily-loaded networks with a relatively large fraction of voice traffic. This is the case because they do need neither the addition of redundancy nor feedback and thus yield stable voice quality for higher loss rates due to absence of FEC and feedback loss. Such schemes can better accomodate non-adaptive codecs like the G.729, which are difficult to integrate into FEC schemes requiring adaptivity of both the codec and the redundancy generator. Also, it is useful for adaptive codecs running at the lowest possible bit-rate. Avoiding redundancy and feedback is also interesting in multicast conferencing scenarios where the end-to-end loss characteristics of the different paths leading to members of the session are largely different. However, our work has clearly focused on linking simple end-to-end models which can be easily parameterized with the known characteristic of hop-by-hop loss control to user-level metrics. An analysis of a large scale deployment of non-adaptive or adaptive FEC as compared to a deployment of our combined schemes needs clearly further study.

Chapter 8

Conclusions

This dissertation is concluded by summarizing the methodology and major results and pointing out directions for future research. By a combination of theoretical analysis, simulation, implementation and measurement in the Internet we have endeavored to contribute to the efficient protection of voice traffic transmitted over a lossy packet-switched network.

The introduction of the novel concepts of intra- and inter-flow Quality-of-Service together with the joint consideration of end-to-end and hop-by-hop schemes for QoS enhancement have allowed a new view on the field. The intra-flow QoS concept reflects the variable sensitivity of a voice application to packet loss. This variability is due to temporal sensitivity (loss correlation of packets) and sensitivity to payload heterogeneity (packets of variable importance exist). This makes the loss distribution within the flow a crucial parameter and QoS mechanisms should thus enable different levels of protection for packets.

Adaptive packetization, the selective addition of redundancy as well as selective packet marking have been identified as suitable intra-flow QoS enhancement mechanisms at the sender. Corresponding schemes within the network and the receiver are selective discarding and reconstruction/concealment respectively. At the end-to-end level these building blocks are used to concentrate redundancy on essential packets (thus reducing the necessary bandwidth for error protection) and to conceal the loss of less important packets with the information contained in the essential packets. Intra-flow hop-by-hop schemes on the other hand allow trading the loss of one packet, which is considered essential, against another one of the same flow which is of lower importance. As both packets require the same cost in terms of network transmission, a gain in terms of user perception is obtainable. When both end-to-end and hop-by-hop mechanisms are combined, the notion of "importance" refers either directly to the described variable loss sensitivity (then we are considering the case of explicit end-to-end/hop-by-hop cooperation) or to the impact of a loss on the operation of the application enhanced by the end-to-end algorithm (the case with only minimal cooperation).

The concept of intra-flow QoS also implied the need for metrics (chapter 4) describing the loss process of consecutive packets: We have built a framework in which most of the previously unrelated inter- and intra-flow loss metrics existing in the

literature can be defined and used together. By applying these run-length-based models to measurement traces of IP voice flows, we demonstrated the tradeoffs between accurate multi-parameter modeling and employing the simple two-state Gilbert model. We conclude that for applications with simple utility functions, simple end-to-end loss recovery mechanisms and conventional queue management algorithms, the Gilbert model yields sufficient information. If one of these assumptions does not hold however a characterization using higher order models is needed. Here, run-length-based models offer a reasonable accuracy versus simplicity tradeoff. Considering the described payload heterogeneity it is necessary to quantify the level of "importance" of particular packets and to capture the effect of loss at the user level, i.e. the impact on user perception. We have done a first step in this direction by relating the packet-level metrics to objective speech quality measures. There we employed the Gilbert model to produce synthesized loss patterns and linked the results to objective speech quality when using a particular codec.

Our approach to end-to-end QoS enhancement (chapter 5) has lead us to the conclusion that sample- and frame-based codecs should be treated separately: Redundancies within a speech signal can be exploited both for compression and loss resilience. The higher the compression of the signal is, the lower is the intrinsic loss resilience (section 2.2.1.1). For (low-compressing) sample-based codecs without loss concealment we have found that they neither exhibit significant temporal sensitivity nor sensitivity to payload heterogeneity. With loss concealment however, the speech quality is increased but the amount of increase exhibits strong temporal sensitivity. (High-compressing) frame-based codecs amplify on one hand the impact of loss by error propagation, though on the other hand such coding schemes help to perform loss concealment by extrapolation of decoder state. Contrary to sample-based codecs we have shown that the concealment performance may "break" at transitions within the speech signal however, thus showing strong sensitivity to payload heterogeneity.

In chapter 6 we have characterized the desired behavior of a hop-by-hop loss control algorithm for the support of end-to-end loss recovery in terms of the packet-level metrics and identified several design choices and tradeoffs for loss control algorithms (per-flow or per-packet signaling of participation in the scheme, per-flow or only per-packet class state, local or distributed operation). Two queue management algorithms representing orthogonal design choices have been developed, implemented and evaluated. We have found that both types of algorithms do not have a significant impact on conventional traffic. It is possible to control the loss distributions for individual flows while keeping their unconditional loss probability within a controlled bound around the value expected using conventional Drop Tail or RED algorithms. Algorithms using packet marking are found to be superior because a high probability for short bursts can be traded against a higher probability for isolated losses as well as a higher (but acceptable) probability for very long loss bursts. This is mainly due to the "memory" realized with the average queue size (the congestion indication and dropping decision is influenced by a longer term monitoring process). Furthermore, with packet marking complex, non-periodic loss patterns can be realized, i.e. an explicit cooperation of the end-to-end and the hop-by-hop algorithm can take place.

Reusing the results on the loss impact on frame-based codecs, we have developed

such a scheme in chapter 7. It has been shown that trading the loss of one packet which is marked as essential against another one of the same flow which is of lower importance performs much better in terms of speech quality than using the conventional best effort service[1].

In section 2.2.3.2 we have highlighted the qualitative tradeoffs for large-scale deployment of generic loss avoidance, recovery and control mechanisms. Then, in the last chapter, we have demonstrated some of the technical tradeoffs between combined schemes for intra-flow QoS support and pure end-to-end or hop-by-hop schemes. However, for future work, to do this comparison in a large-scale network scenario which takes into account the impact of adding redundancy to the network load seems to be very interesting ([PRM98]). In addition to a technical analysis in the large scale, the economical implications (cost of end-to-end versus hop-by-hop deployments) would need to be assessed to allow for a final judgment which (combination of) mechanisms are most useful for actual deployment.

We also consider a speech-property- *and* rate-adaptive (cf. section 3.1.2.3) Forward Error Correction scheme to be highly desirable. In such a scheme, the sender receives feedback information on the network loss conditions from the receivers and uses this information to determine the optimal amount of redundant and payload data. Note that to fulfill our definition of intra-flow QoS in Table 1.1 any intra-flow QoS FEC scheme must be rate-adaptive. While the theoretical foundations of such a scheme have been outlined by Bolot et. al. ([BFPT99]), the inflexibility of current speech codecs precludes further advances in this direction. Our speech-property based scheme for the identification of essential frames could also be linked to a dropping mechanisms of non-essential frames at the sender. This would enable some, though limited, rate adaptivity for both the main and the redundant payloads. A comprehensive solution, however, should come from interaction with the speech coding community ([KJ00]) to realize a codec which can be truly adaptive over a wide range of bit-rates and resulting speech qualities.

To extend the applicability of speech-property-adaptivity it would make sense to use the objective speech quality measures (which we employed for off-line trace analysis) in the analysis module of the sender to enable the on-line identification of packets which are more important than others independently of the coding scheme. However this "short-term" objective speech quality measurement clearly requires modifications to the measurement algorithms. The reason for this is that the MOS as the target value of the measurement is not well defined for short time intervals (in the range of one or few packets). Thus extensive subjective testing followed by revalidation and/or modification of the measurement model is necessary ([Vor00]).

In this thesis we did not address the joint implementation of intra-flow with inter-flow hop-by-hop QoS mechanisms. However we believe that both proposed algorithms (especially DiffRED) could be combined with such methods. As the

[1]We have also evaluated the mapping of an end-to-end algorithm to inter-flow protection. We have found that the selective marking scheme performs almost as good as the protection of the entire flow at a significantly lower number of necessary high-priority packets.

algorithms are working on a single queue only (on the dropping decision) their implementation is somewhat orthogonal to inter-flow QoS enhancement mechanisms which deal mainly with scheduling multiple queues (service decision). Obviously when bringing them together their parameters are inter-related (the scheduling weight parameter of the queue directly influences the level of packet dropping in the queue). While the choice of parameters in such a configuration is known to be non-trivial, the problem is manageable.

While we distinguished between the end-to-end and hop-by-hop level, future programmable / active networks might blur the strict boundary between them. They would allow the placement of payload-aware processing modules inside the network thus possibly changing the "end-to-end" view to "edge-to-edge" or "domain-to-domain".

In [PKH+97], Perkins et al. justified the deployment of (non-adaptive) FEC mechanisms for packet-audio as follows: "The disruption of speech intelligibility even at low loss rates which is currently experienced may convince a whole generation of users that multimedia conferencing over the Internet is not viable." However, widespread deployment of non-adaptive FEC might worsen congestion finally ([PRM98]). This holds especially for large-scale multicast conferencing scenarios where the individual receivers experience largely different loss characteristics. We therefore recommend to use speech- and network-adaptive end-to-end loss recovery in connection with intra-flow packet marking within the Internet which remains "best effort". No complete charging/accounting is needed and only some control of flow aggregates at domain boundaries in terms of marking fairness is necessary, thus making deployment easier. Intra-flow QoS both at the end-to-end and hop-by-hop level could thus be a starting point on the transition path to the deployment of more complex (inter-flow) QoS support mechanisms.

Appendix A

Acronyms

A/D Analog / Digital conversion

AAL ATM Adaptation Layer

ADPCM Adaptive Differential Puls Code Modulation

ADU application data unit

AD Auditory Distance

ALT-DIFFMARK Alternating Differential Packet Marking

ALT-MARK Alternating Packet Marking

AP/C Adaptive Packetization / Concealment

ATM Asynchronous Transfer Mode

BSD Bark Spectral Distortion

BT background traffic

CELP Code Excited Linear Prediction

CS-ACELP Conjugate Structure Algebraic Code Excited Linear Prediction

D/A Digital / Analog conversion

DNS Domain-Name System

DPCM Differential Puls Code Modulation

DT Drop Tail

DiffRED Differential RED

EMBSD Enhanced Modified Bark Spectral Distortion

FEC forward error correction

FIFO First-In First-Out

FT foreground traffic

GSM Groupe Speciale Mobile

IETF Internet Engineering Task Force

IP Internet Protocol

ISDN Integrated Services Digital Network

ISP Internet Service Provider

ITU International Telecommunications Union

LAN Local Area Network

LPC Linear Predictive Coding

LP Linear Prediction

MBONE Multicast Backbone

MNB Measuring Normalizing Blocks

MOS Mean Opinion Score

MPEG Motion Picture Experts Group

OPLP Optimal Predictive Loss Pattern

OSI Open Systems Interconnection

PAM Puls Amplitude Modulation

PCM Puls Code Modulation

PLoP Predictive Loss Pattern

PPP Point-to-Point Protocol

PSTN Public Switched Telephone Network

PT Payload Type

PWR Pitch Waveform Replication

QoS Quality of Service

RED Random Early Detection

RFC Request For Comments

RIO RED with In and Out

RSVP Resource ReSerVation Protocol

RTCP Real-time Transport Control Protocol

RTP Real-time Transport Protocol

SDH Synchronous Digital Hierarchy

SIP Session Initiation Protocol

SNR Signal-to-Noise Ratio

SPB-DIFFMARK Speech Property-Based Differential Packet Marking

SPB-FEC Speech Property-Based Forward Error Correction

SPB-MARK Speech Property-Based Packet Marking

TCP Transmission Control Protocol

TM Time-scale Modification

ToS Type of Service

UDP User Datagram Protocol

VAD Voice Activity Detection

VoIP Voice over IP

clp conditional loss probability

codec coder/decoder

http hyper-text transfer protocol

ulp unconditional loss probability

Bibliography

[AAOS98] A. Acharya, F. Ansari, M. Ott, and H. Sanneck. "Dynamic QoS support for IP Switching using RSVP over IP-SOFACTO". In *International Symposium on Broadband European Networks (SYBEN '98)*, Zurich, Switzerland, May 1998. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9805:-RSVP_ipsofacto.ps.gz.

[ABE+94] A. Albanese, J. Blöemer, J. Edmonds, M. Luby, and M. Sudhan. "Priority Encoding Transmission". In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, IEEE Computer Science Press, 1994.

[AHV98] J. Andren, M. Hilding, and D. Veitch. "Understanding End-to-End Internet Traffic Dynamics". In *Proceedings IEEE GLOBECOM*, Sydney, Australia, November 1998.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. "An Architecture for Differentiated Services". RFC 2475, IETF, December 1998. ftp://ftp.ietf.org/rfc/rfc2475.txt.

[BCS94] R. Braden, D. Clark, and S. Shenker. "Integrated Services in the Internet Architecture: an Overview". RFC 1633, IETF, 1994. ftp://ftp.ietf.org/rfc/rfc1633.txt.

[Bee97] J. Beerends. "Psycho-acoustic models". Electronic mail, KPN Research, June 1997. http://sound.media.mit.edu/dpwe-bin/mhindex.cgi/AUDITORY/postings/1997/192.

[BFPT99] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. "Adaptive FEC-Based Error Control for Interactive Audio in the Internet". In *Proceedings IEEE INFOCOM*, New York, NY, March 1999.

[BG96] J.-C. Bolot and A.V. Garcia. "Control Mechanisms for Packet Audio in the Internet". In *Proceedings IEEE INFOCOM*, pages 232–239, San Francisco, CA, April 1996.

[Bla00] U. Black. *Voice over IP*. Prentice Hall, 2000.

[BLHHM95] M. Bjorkman, A. Latour-Henner, U. Hansson, and A. Miah. "Controllability and Impact of Cell Loss Process in ATM Networks". In *Proceedings of IEEE GLOBECOM*, pages 916–920, 1995.

[Bol93] J.-C. Bolot. "Characterizing End-to-End Packet Delay and Loss in the Internet". *Journal of High-Speed Networks*, Vol. 2(3):305–323, December 1993.

[BS85] J. Blauert and E. Schaffert. *Automatische Sprachein- und -ausgabe.* Schriftenreihe der Bundesanstalt für Arbeitsschutz, Dortmund 1985. Forschung-Fb Nr. 417, S. 30-42.

[BS96] K. Brown and S. Singh. "Loss Profiles at the Link Layer". In *3rd Intl. Workshop on Mobile Multimedia Communication*, September 1996.

[BSUG98] M.S. Borella, D. Swider, S. Uludag, and G.Brewster. "Internet Packet Loss: Measurement and Implications for End-to-End QoS". In *Proceedings of the International Conference on Parallel Processing*, August 1998.

[BVG97] J. C. Bolot and A. Vega Garcia. "The case for FEC-based error control for packet audio in the Internet". *ACM Multimedia Systems*, 1997.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. "RSVP - Version 1 Functional Specification". RFC 2205, IETF, November 1997. ftp://ftp.ietf.org/rfc/rfc2205.txt.

[Cas92] S. Casner. "First IETF Internet Audiocast". *Computer Communication Review, ACM SIGCOMM*, 22(3):92–97, July 1992.

[CB97a] G. Carle and E. Biersack. "Survey of Error Recovery Techniques for IP-based Audio-Visual Multicast Applications". *IEEE Network Magazine*, 11(6), November/December 1997.

[CB97b] M. Crovella and A. Bestavros. "Self-similarity in world wide web traffic: evidence and possible causes". *IEEE/ACM Transactions on Networking*, Vol. 5(6):835–846, December 1997.

[CC97] Y.L. Chen and B.S. Chen. "Model-based multirate representation of speech signals and its application to recovery of missing speech pakets". *IEEE Transactions on Speech and Audio Processing*, 15(3):220–231, May 1997.

[CF97] D. Clark and W. Fang. "Explicit Allocation of Best Effort Packet Delivery Service". Technical Report, MIT LCS, 1997. http://diffserv.lcs.mit.edu/Papers/exp-alloc-ddc-wf.pdf.

[CK96] R. Cox and P. Kroon. "Low Bit-Rate Speech Coders for Multimedia Communication". *IEEE Communications Magazine*, pages 34–41, December 1996.

[CKS93] I. Cidon, A. Khamisy, and M. Sidi. "Analysis of Packet Loss Processes in High-Speed Networks". *IEEE Transactions on Information Theory*, 39:98–108, January 1993.

[CLMT99] C.M. Chernick, S. Leigh, K. Mills, and R. Toense. "Testing the Ability of Speech Recognizers to Measure the Effectiveness of Encoding Algorithms for Digital Speech Transmission". In *Proceedings of MILCOM*, October 1999.

[Clu98] Kai Cluever. *Rekonstruktion fehlender Signalblöcke bei blockorientierter Sprachübertragung (Reconstruction of missing signal blocks for block-orientated voice transmission)*. PhD thesis, Telecommunications Department, Technical University of Berlin, January 1998. http://www-ft.ee.tu-berlin.de/Publikationen/kcd.pdf.

[CMT98] K. Claffy, G. Miller, and K. Thompson. "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone". In *Proceedings INET*, Geneva, Switzerland, July 1998. http://www.isoc.org/inet98/proceedings/6g/6g_3.htm.

[Coh80] D. Cohen. "On Packet Speech Communications". In *Proceedings of the Fifth International Conference on Computer Communications*, pages 271–274, Atlanta, GA, October 1980.

[Col98] Columbia University, Dept. of Computer Science. *Network Voice Terminal (NeVoT)*, 1998. http://www.cs.columbia.edu/~hgs/nevot/.

[Con97] International Multimedia Teleconferencing Consortium. "Service Interoperability Implementation Agreement 1.0". Technical report, Voice over IP Forum, Technical Committee, December 1997.

[CSS+98] Georg Carle, Henning Sanneck, Dorgham Sisalem, Michael Smirnow, Adam Wolisz, and Tanja Zseby. "Dienstqualitätsunterstützung im Internet (Quality of Service support in the Internet)". *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, 3/98, September 1998.

[CSS00] G. Carle, H. Sanneck, and M. Schramm. "Adaptive Hybrid Error Control for IP-based Continous Media Multicast Services". In *First International Workshop on Quality for future Internet Services (QofIS 2000)*, Berlin, Germany, September 2000. ftp://ftp.fokus.gmd.de/pub/glone/papers/Carl0009:Hybrid.ps.gz.

[CT97] H. W. Chu and D. H. K. Tsang. "Dynamic Bandwidth Allocation for VBR Video Traffic in ATM Networks". In *Proceedings of ICCCN*, pages 306–312, 1997.

[Deg96] J. Degener. "GSM 06.10 lossy speech compression". Documentation, TU Berlin, KBS, October 1996. http://kbs.cs.tu-berlin.de/~jutta/toast.html.

[Del93] J.R. Deller. *Discrete-Time Processing of Speech Signals*. Prentice Hall, Englewood Cliffs 1993.

[DLW96] B. Dempsey, J. Liebeherr, and A.C. Weaver. "On Retransmission-Based Error Control for Continuous Media Traffic in Packet-Switching Networks". *Computer Networks and ISDN Systems*, 28(5):719–736, March 1996.

[DPF89] L.A. DaSilva, D.W. Petr, and V.S. Frost. "A Class-Oriented Replacement Technique for Lost Speech Packets". *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-37(10):1597–1600, October 1989.

[eCS+97] L. Salgarelli (ed.), A. Corghi, M. Smirnow, H. Sanneck, and D. Witaszek. "Supporting IP Multicast Integrated Services in ATM Networks". Internet Draft, IETF Integrated Services over Specific Link Layers (ISSLL) Working Group, November 1997.

[ECZ93] N. Erdöl, C. Castelluccia, and A. Zilouchian. "Recovery of Missing Speech Packets Using the Short-Time Energy and Zero-Crossing Measurements". *IEEE Transactions on Speech and Audio Processing*, 1(3):295–303, July 1993.

[FdSeS99] D. Figueiredo and E. de Souza e Silva. "Efficient Mechanisms for Recovering Voice Packets in the Internet". In *Proceedings IEEE GLOBECOM*, pages 1830–1837, Rio de Janeiro, Brazil, November 1999.

[FF97] S. Floyd and K. Fall. "Router Mechanisms to Support End-to-End Congestion Control". Technical Report, Network Research Group, LBNL, February 1997.

[FJ93] S. Floyd and V. Jacobson. "Random Early Detection Gateways for Congestion Avoidance". *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[FL90] J.M. Ferrandiz and A.A. Lazar. "Consecutive Packet Loss in Real-Time Packet Traffic". In *Proceedings of the Fourth International Conference on Data Communications Systems, IFIP TC6*, pages 306–324, Barcelona, June 1990.

[Gar96] Andrés Vega Garcia. *Mécanismes de Contrôle pour la Transmission de l'Audio sur l'Internet (Control Mechanisms for Audio Transmission over the Internet)*. PhD thesis, Université de Nice, Nice, France, October 1996.

[GLWW86] D.J. Goodman, G.B. Lockhart, O.J. Wasem, and W. Wong. "Waveform Substitution Techniques for Recovering Missing Speech Segments in Packet Voice Communications". *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34(6):1449–1464, December 1986.

[GP98] R. Guérin and V. Peris. "Quality-of-Service in Packet Networks: Basic Mechanisms and Directions". Research Report RC21089, IBM, January 1998.

[Gru94] R. Grudszus. "Untersuchung von Verfahren zur Zeitdehnung und -stauchung von Sprachsignalen (Examination of Methods for the Time-Scale Expansion and Compression of Speech Signals)". Diploma Thesis, Lehrstuhl für Nachrichtentechnik, Erlangen, September 1994.

[GS85] J. Gruber and L. Strawczynski. "Subjective Effects of Variable Delay and Speech Clipping in Dynamically Managed Voice Systems". *IEEE Transactions on Communications*, Vol. COM-33(8), August 1985.

[GV93] M. Garrett and M. Vetterli. "Joint Source/Channel Coding of Statistically Multiplexed Real Time Services on Packet Network". *IEEE/ACM Transactions on Networking*, February 1993.

[GWDP88] D.J. Goodman, O.J. Wasem, C.A. Dvorak, and H.G. Page. "The Effect of Waveform Substitution on the Quality of PCM Packet Communications". *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-36(3):342–348, March 1988.

[HBWW99] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. "Assured Forwarding PHB Group". RFC 2597, IETF Diffserv Working Group, June 1999. ftp://ftp.ietf.org/rfc/rfc2597.txt.

[HCB96] M. Handley, J. Crowcroft, and C. Bormann. "The Internet Multimedia Conferencing Architecture". Internet Draft (expired), IETF MMUSIC Working Group, February 1996.

[HOK97] C. Hsu, A. Ortega, and M. Khansari. "Rate control for robust video transmission over wireless channels". In *Proceedings of Visual Communications and Image Processing (VCIP)*, pages 1200–1211, San Jose, CA, February 1997.

[HSHW95] V. Hardman, M. Sasse, M. Handley, and A. Watson. "Reliable Audio for Use over the Internet". In *Proceedings INET*, http://info.isoc.org/HMP/PAPER/070/abst.html, 1995.

[HSSR99] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. "SIP: Session Initiation Protocol". RFC 2543, IETF, March 1999. ftp://ftp.ietf.org/rfc/rfc2543.txt.

[IEE69] IEEE. "IEEE Recommended Practice for Speech Quality Measurements". *IEEE Transactions on Audio and Electroacoustics*, AU-17:227–245, September 1969.

[IKL97] M. Ilvesmäki, K. Kilkki, and M. Luoma. "Packets or ports - the decisions of IP switching". In *Broadband Networking Technologies, Seyhan Civanlar, Indra Widjaja, Editors, Proceedings SPIE Vol.3233*, pages 53–64, Dallas, TX, November 1997.

[INR00] INRIA. *Freephone*, 2000. http://zenon.inria.fr/rodeo/fphone/.

[Ins98] European Telecommunications Standards Institute. "Telecommunications and Internet Protocol Harmonization over Networks (TIPHON); General aspects of Quality of Service (QoS)". Technical report tr 101 329 v1.2.5 (1998-10), European Telecommunications Standards Institute, 1998.

[Ise96] M. Isenburg. "Transmission of multimedia data over lossy networks". Technical Report TR-96-048, ICSI, 1996. http://www.icsi.berkeley.edu/~isenburg/studyA4.ps.gz.

[IV95] A. Ingle and V. Vaishampayan. "DPCM System Design for Diversity Systems With Applications to Packetized Speech". *IEEE Transactions on Speech and Audio Processing*, 3(1):48–58, January 1995.

[Jay93] N.S. Jayant. "High Quality Networking of Audio-Visual Information". *IEEE Communications Magazine*, pages 84–95, September 1993.

[JC81] N.S. Jayant and S.W. Christensen. "Effects of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample-Interpolation Procedure". *IEEE Transactions on Communications*, Vol. COM-29(2):101–109, February 1981.

[JH98] J.Rosenberg and H.Schulzrinne. "Issues and Options for RTP Multiplexing". Internet Draft, Work in Progress, IETF AVT Working Group, October 1998. ftp://ftp.ietf.org/internet-drafts/draft-ietf-avt-muxissues-00.txt.

[JN84] N.S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice Hall, Englewood Cliffs 1984.

[JNP99] V. Jacobson, K. Nichols, and K. Poduri. "An Expedited Forwarding PHB". RFC 2598, IETF Diffserv Working Group, June 1999. ftp://ftp.ietf.org/rfc/rfc2598.txt.

[JS00a] W. Jiang and H. Schulzrinne. "Analysis of On-Off Patterns in VoIP and Their Effect on Voice Traffic Aggregation". In *Proceedings of the Ninth Conference on Computer Communications and Networks (ICCCN)*, Las Vegas, NV, October 2000.

[JS00b] W. Jiang and H. Schulzrinne. "QoS Measurement of Internet Real-Time Multimedia Services". In *Proceedings NOSSDAV*, Chapel Hill, NC, June 2000.

[KBS+98] T. Kostas, M. Borella, I. Sidhu, G. Schuster, J. Grabiec, and J. Mahler. "Real-Time Voice Over Packet-Switched Networks". *IEEE Network Magazine*, 12(1), January/February 1998.

[KH95] W. B. Kleijn and J. Haagen. "A Speech Coder Based on Decomposition of Characteristic Waveforms". In *Proceedings ICASSP*, pages 508–511, Detroit, MI, 1995.

[KHHC97] I. Kouvelas, O. Hodon, V. Hardman, and J. Crowcroft. "Redundancy Control in Real-Time Internet Audio Conferencing". In *Proceedings of Audio-Visual Services over Packet Networks (AVSPN 97)*, Aberdeen, Scotland, September 1997.

[Kil99] K. Kilkki. *Differentiated Services.* Macmillan Technical Publishing, Indianapolis 1999.

[KJ00] W. B. Kleijn and A. Jefremov. "Packet loss resiliency of Waveform-Interpolation codecs". Personal communication, KTH TMH, May 2000.

[KK97] R. Koodli and C.M. Krishna. "Supporting Multiple-tier QoS in a Video Bridging Application". In *IFIP Fifth International Workshop on Quality of Service (IWQOS '97)*, New York, NY, USA, May 1997.

[KK98] R. Koodli and C.M. Krishna. "Noticeable loss: A Metric for Capturing Loss Pattern for Continous Media Applications". In *Internet Routing and Quality of Service, S. Civanlar, P. Doolan, J. Luciani, R. Onvural, Editors, Proceedings SPIE Vol.3529A*, Boston, MA, November 1998.

[KR97] R. Koodli and R. Ravikanth. "Impact of Loss Characteristics on Real-Time Applications". Presentation to the IPPM Working Group, Proceedings of the 39th IETF, Washington, DC, USA, December 1997. http://ietf.org/proceedings/97dec/slides/ippm-nokia/index.htm.

[KR00] R. Koodli and R. Ravikanth. "One-way Loss Pattern Sample Metrics". Internet Draft, IETF IPPM Working Group, July 2000. ftp://ftp.ietf.org/internet-drafts/draft-ietf-ippm-loss-pattern-03.txt.

[LBL92] M.M. Lara-Barron and G.B. Lockhart. "Speech Encoding and Reconstruction for Packet Networks using Redundancy". *IEE Colloquium on Coding for Packet Video and Speech Transmission*, 199(3):1–4, February 1992.

[LBN98] LBNL Network Research Group. *Visual Audio Tool (VAT)*, 1998. http://www-nrg.ee.lbl.gov/vat.

[LCC+98] B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, and S. Wolff. *A Brief History of the Internet, Version 3.1.* Internet Society, 1998. http://www.isoc.org/internet/history/brief.html.

[Le99] N. Le. "Development of a Loss-Resilient Internet Speech Transmission Method". Diploma Thesis, GMD Fokus / Lehrstuhl für Telekommunikationsnetze, TU Berlin, Berlin, Mai 1999.

[LNT96] Z. Liu, P. Nain, and D. Towsley. "Bounds on Finite Horizon QoS Metrics with Application to Call Admission". In *Proceedings IEEE INFOCOM*, San Francisco, CA, USA, April 1996.

[LSCH00] N. Le, H. Sanneck, G. Carle, and T. Hoshi. "Active Concealment for Internet Speech Transmission". In *Proceedings of the Second International Working Conference on Active Networks*, Tokyo, Japan, October 2000.

[LTWW93] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. "On the Self-Similar Nature of Ethernet Traffic". In *Proceedings ACM SIGCOMM*, San Francisco, CA, September 1993.

[MBJMD99] M. May, J.-C. Bolot, A. Jean-Marie, and C. Diot. "Simple Performance Models of Differentiated Services Schemes for the Internet". In *Proceedings IEEE INFOCOM*, New York, NY, USA, March 1999.

[MFO98] T. Miyata, H. Fukuda, and S. Ono. "New Network QoS measures for FEC-based Audio Applications on the Internet". In *Proceedings IEEE IPCCC 1998*, pages 355–362, Tempe/Phoenix, AZ, USA, February 1998.

[Mil99] K. Mills. "Speech recognition for the quality evaluation of speech transmissions". Personal communication, NIST, August 1999.

[Min79] D. Minoli. "Optimal Packet Length for Packet Voice Communication". *IEEE Transactions on Communications*, COM-27(3):607–611, March 1979.

[MIT99] MIT LCS. *Fastest Fourier Transform in the West*, 1999. http://theory.lcs.mit.edu/~fftw.

[MJV96] S. McCanne, V. Jacobson, and M. Vetterli. "Receiver-driven Layered Multicast". In *Proceedings ACM SIGCOMM*, pages 117–130, Stanford, CA, September 1996.

[MM98] D. Minoli and E. Minoli. *Delivering Voice over IP Networks*. John Wiley and Sons, 1998.

[MS96] M. Meky and T. N. Saadawi. "Degradation effect of Cell Loss on Speech Quality over ATM Networks". In *Broadband Communications*, IFIP, Chapman and Hall, pages 259–271, 1996.

[MYT87] N. Matsuo, M. Yuito, and Y. Tokunaga. "Packet Interleaving for Reducing Speech Quality Degradation in Packet Voice Communications". In *Proceedings GLOBECOM*, pages 1787–1791, 1987.

[NKT94] R. Nagarajan, J. Kurose, and D. Towsley. "Finite-Horizon Statistical Quality-of-Service Measures for High Speed Networks". *J. High Speed Networks*, December 1994.

[NLM96] P. Newman, T. Lyon, and G. Minshall. "Flow Labelled IP: Connectionless ATM under IP". In *Proceedings of Networld + Interop*, Las Vegas, NV, USA, April 1996.

[Nov96] R. J. Novorita. "Improved Mean Opinion Score Objective Prediction of Voice Coded Speech Signals". Master's thesis, Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, 1996.

[OMF98] S. Ono, T. Miyata, and H. Fukuda. "Loss Metrics of Grouped Packets for IPPM". Internet Draft, IETF IPPM Working Group, August 1998. ftp://ftp.ietf.org/internet-drafts/draft-ono-group-loss-00.txt.

[Pap87] P.E. Papamichalis. *Practical Approaches To Speech Coding*, Ch. 7, pages 186–198. Prentice Hall, Englewood Cliffs 1987.

[Par92] C. Partridge. "A Proposed Flow Specification". RFC 1363, IETF, September 1992. ftp://ftp.ietf.org/rfc/rfc1363.txt.

[Per99] C. Perkins. "RTP Payload Format for Interleaved Media". Internet Draft, IETF Audio/Video Transport Working Group, February 1999. ftp://ftp.ietf.org/internet-drafts/draft-ietf-avt-interleaving-01.txt.

[PH98] C. Perkins and O. Hodson. "Options for the Repair of Streaming Media". RFC 2354, IETF, June 1998. ftp://ftp.ietf.org/rfc/rfc2354.txt.

[PHH98] C. Perkins, O. Hodson, and V. Hardman. "A Survey of Packet-Loss Recovery Techniques for Streaming Audio". *IEEE Network Magazine*, Sept./Oct. 1998.

[PJS99] M. Parris, K. Jeffay, and F. Smith. "Lightweight Active Router Queue Management for Multimedia Networking". In *Multimedia Computing and Networking Conference, Proceedings SPIE Vol. 3654*, pages 162–174, San Jose, CA, January 1999.

[PKH+97] C. Perkins, I. Kouvelas, O. Hodson, M. Handley, and J. Bolot. "RTP payload for redundant audio data". RFC 2198, IETF, September 1997. ftp://ftp.ietf.org/rfc/rfc2198.txt.

[PRM98] M. Podolsky, C. Romer, and S. McCanne. "Simulation of FEC-based Error Control for Packet Audio on the Internet". In *Proceedings IEEE INFOCOM*, pages 48–52, San Francisco, CA, March 1998.

[PS98] P. Pan and H. Schulzrinne. "YESSIR: A Simple Reservation Mechanism for the Internet". In *Proceedings NOSSDAV*, Cambridge, UK, July 1998.

[Ram70] J.L. Ramsey. "Realization of Optimum Interleavers". *IEEE Transactions on Information Theory*, IT-16:338–345, May 1970.

[Rhe98] I. Rhee. "Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet". In *Proceedings ACM SIGCOMM*, Vancouver, B.C., September 1998.

[RI97] D. Reininger and R. Izmailov. "Soft Quality of Service with VBR+ Video". In *Proceedings of 8th International Workshop on Packet Video (AVSPN97)*, Aberdeen, Scotland, September 1997.

[Riz97] L. Rizzo. "Effective erasure codes for reliable computer communication protocols". *Computer Communication Review, ACM SIGCOMM*, April 1997.

[RKTS94] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne. "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks". In *Proceedings IEEE INFOCOM*, pages 680–688, 1994.

[Ros97a] J. Rosenberg. "G. 729 Error Recovery for Internet Telephony". Project report, Columbia University, 1997.

[Ros97b] J. Rosenberg. "Reliability Enhancements to NeVoT". Project report, Columbia University, 1997.

[RQS00] J. Rosenberg, L. Qiu, and H. Schulzrinne. "Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet". In *Proceedings IEEE INFOCOM*, Tel Aviv, Israel, March 2000.

[RR95] I. E. G. Richardson and M. J. Riley. "Usage Parameter Control Cell Loss Effects on MPEG Video". In *Proceedings ICC*, pages 970–974, 1995.

[RRV93] S. Ramanathan, P.V. Rangan, and H. Vin. "Frame-Induced Packet Discarding: An Efficient Strategy for Video Networking". In *Proceedings NOSSDAV*, pages 173–184, 1993.

[RS78] L.R. Rabiner and R.W. Schafer. *Digital Processing Of Speech Signals*. Prentice Hall, Englewood Cliffs 1978.

[RS96] J. Rosenberg and H. Schulzrinne. "Issues and Options for an Aggregation Service within RTP". Internet Draft, Work in Progress, IETF AVT Working Group, December 1996. ftp://ftp.ietf.org/internet-drafts/draft-rosenberg-itg-00.txt.

[RS98] J. Rosenberg and H. Schulzrinne. "An RTP Payload Format for User Multiplexing". Internet Draft, Work in Progress, IETF AVT Working Group, November 1998. ftp://ftp.ietf.org/internet-drafts/draft-ietf-avt-aggregation-00.txt.

[RS99] J. Rosenberg and H. Schulzrinne. "An RTP Payload Format for Generic Forward Error Correction". RFC 2733, IETF, December 1999. ftp://ftp.ietf.org/rfc/rfc2733.txt.

[San95] H. Sanneck. "Fehlerverschleierungsverfahren für Sprachübertragung mit Paketverlust (Error Concealment Methods for Speech Transmission with Packet Losses)". Diploma Thesis, Lehrstuhl für Nachrichtentechnik, Erlangen, June 1995.

[San98a] H. Sanneck. "Adaptive Loss Concealment for Internet Telephony Applications". In *Proceedings INET*, Geneva, Switzerland, July 1998. http://www.isoc.org/inet98/proceedings/6e/6e_3.htm.

[San98b] H. Sanneck. "Concealment of Lost Speech Packets Using Adaptive Packetization". In *Proceedings IEEE Multimedia Systems*, pages 140–149, Austin, TX, June 1998. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9806:Adaptive.ps.gz.

[SB85] R. Steele and F. Benjamin. "Variable-Length Packetization of μ-law PCM speech". *AT&T Technical Journal*, 64:1271–1292, July-August 1985.

[SC98] H. Sanneck and G. Carle. "Predictive Loss Pattern Queue Management for Internet Routers". In *Internet Routing and Quality of Service, S. Civanlar, P. Doolan, J. Luciani, R. Onvural, Editors, Proceedings SPIE Vol.3529A*, pages 205–216, Boston, MA, November 1998. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9811:Predictive.ps.gz.

[SC99] H. Sanneck and G. Carle. "A Queue Management Algorithm for Intra-Flow Service Differentiation in the "Best Effort" Internet". In *Proceedings of the Eighth Conference on Computer Communications*

and Networks (ICCCN), pages 419–426, Natick, MA, October 1999. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9910:Intra-Flow.ps.gz.

[SC00a] H. Sanneck and G. Carle. "A Framework Model for Packet Loss Metrics Based on Loss Runlengths". In *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference (MMCN)*, pages 177–187, San Jose, CA, January 2000. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0001:Runlength-Metrics.ps.gz.

[SC00b] H. Schulzrinne and S. Casner. "RTP Profile for Audio and Video Conferences with Minimal Control". Internet Draft, IETF Audio-Video Transport Group, January 2000. ftp://ftp.ietf.org/internet-drafts/draft-ietf-avt-profile-new-08.txt.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. "RTP: a transport protocol for real-time applications". RFC 1889, IETF, January 1996. ftp://ftp.ietf.org/rfc/rfc1889.txt.

[Sch92] H. Schulzrinne. "Voice communication across the Internet: A network voice terminal". Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, MA, July 1992. ftp://gaia.cs.umass.edu/pub/Schu9207:Voice.ps.Z.

[Sch97] H. Schulzrinne. "Re-engineering the telephone system". In *Proc. of IEEE Singapore International Conference on Networks (SICON)*, Singapore, April 1997.

[SCSW97] L. Salgarelli, A. Corghi, H. Sanneck, and D. Witaszek. "Supporting IP Multicast Integrated Services in ATM Networks". In *Broadband Networking Technologies, Seyhan Civanlar, Indra Widjaja, Editors, Proceedings SPIE Vol.3233*, pages 78–88, Dallas, TX, USA, November 1997. ftp://ftp.fokus.gmd.de/pub/step/multicube/wp23/spie.ps.gz.

[SF85] R. Steele and P. Fortune. "An adaptive packetization strategy for A-law PCM speech". In *Proceedings ICC*, pages 941–945 (29.6), Chicago, IL, June 1985.

[She95] S. Shenker. "Fundamental Design Issues for the Future Internet". *IEEE J. Selected Areas in Communications*, September 1995.

[SKT92] H. Schulzrinne, J. Kurose, and D. Towsley. "Loss correlation for queues with bursty input streams". In *Proceedings ICC*, pages 219–224, Chicago, IL, 1992.

[SL00] H. Sanneck and N. Le. "Speech Property-Based FEC for Internet Telephony Applications". In *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking*

Conference (MMCN), pages 38–51, San Jose, CA, January 2000. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0001:Speech-FEC.ps.gz.

[SLC00] H. Sanneck, N. Le, and G. Carle. "Effiziente Dienstqualitätsunterstützung für IP Telefonie durch selektive Paketmarkierung". In *First IP-Telephony Workshop*, pages 139–151, Berlin, Germany, April 2000. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0004:IPTel-Markierung.ps.gz.

[SLW00] H. Sanneck, N. Le, and A. Wolisz. "Efficient QoS Support for Voice-over-IP Applications Using Selective Packet Marking". In *Special Session on Error Control Techniques for Real-time Delivery of Multimedia data, First International Workshop on Intelligent Multimedia Computing (IMMCN)*, pages 553–556, Atlantic City, NJ, February 2000. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann0002:VoIP-marking.ps.gz.

[SM90] N. Shacham and P. McKenney. "Packet Recovery in High-Speed Networks using Coding and Buffer Management". In *Proceedings ACM SIGCOMM*, pages 124–131, San Francisco, CA, June 1990.

[SM96] C. Semeria and T. Maufer. "Introduction to IP Multicast Routing". Internet Draft (expired), IETF, March 1996. draft-rfced-info-semeria-00.txt.

[Soc] Internet Society. *Internet Engineering Task Force.* http://www.ietf.org.

[Spa94] A. Spanias. "Speech Coding: A Tutorial Review". Technical Report, Arizona State University, October 1994. http://www.eas.asu.edu/∼spanias/papers/review.ps.

[SPG97] S. Shenker, C. Partridge, and R. Guérin. "Specification of Guaranteed Quality of Service". RFC 2212, IETF, September 1997. ftp://ftp.ietf.org/rfc/rfc2212.txt.

[SR98] H. Schulzrinne and J. Rosenberg. "Internet Telephony, Architecture and Protocols: an IETF Perspective". *Computer Networks and ISDN Systems*, July 1998.

[SRG97] A. Stenger, R. Rabenstein, and B. Girod. "Fehlerverschleierung für paketierte Sprachübertragung durch Zeitdehnung und Phasenanpassung (Error Concealment for packetized voice transmission using time-scale modification and phase matching)". In *Proc. 9. Aachener Kolloquium "Signaltheorie"*, pages 211–214, Aachen, Germany, March 1997.

[SS96] K. Seal and S. Singh. "Loss Profiles: A Quality of Service Measure in Mobile Computing". *J. Wireless Networks*, Vol. 2(1):45–61, 1996.

[SS98a] D. Sisalem and H. Schulzrinne. "The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme". In *Proceedings NOSSDAV*, Cambridge, UK, July 1998.

[SS98b] D. Sisalem and H. Schulzrinne. "The Multimedia Internet Terminal". *Journal of Telecommunication Systems, Special Issue on Multimedia*, 9(38), 1998.

[SS98c] B. Subbiah and S. Sengodan. "User Multiplexing in RTP payload between IP Telephony Gateways". Internet Draft, Work in Progress, IETF AVT Working Group, August 1998. ftp://ftp.ietf.org/internet-drafts/draft-ietf-avt-mux-rtp-00.txt.

[SSSK99] D. Sisalem, M. Smirnov, H. Sanneck, and J. Kuthan. "Towards the Next Generation Multimedia IP-Telephony". *Next Generation Internet in Europe, ACTS Project InfoWin (AC 113)*, ISBN 3-00-004250-4:16–19, 1999.

[SSYG96] H. Sanneck, A. Stenger, K. Ben Younes, and B. Girod. "A New Technique for Audio Packet Loss Concealment". In *Proceedings IEEE Global Internet (Jon Crowcroft and Henning Schulzrinne, eds.)*, pages 48–52, London, England, November 1996. ftp://ftp.fokus.gmd.de/pub/glone/papers/Sann9611:New.ps.gz.

[ST89] J. Suzuki and M. Taka. "Missing Packet Recovery Techniques for Low-Bit-Rate Coded Speech". *IEEE Journal on Selected Areas in Communications*, Vol. 7(5):707–717, June 1989.

[SWZS99] H. Sanneck, D. Witaszek, T. Zseby, and M. Smirnov. "MULTICUBE - IP Multicast over ATM Research". *Next Generation Internet in Europe, ACTS Project InfoWin (AC 113)*, ISBN 3-00-004250-4:97–103, 1999.

[Tel99] Committee T1 Telecommunications. "American National Standard for Packet Loss Concealment for Use with ITU-T Recommendation G.711". Draft T1 Standard T1A1.7/99-012r4, Alliance for Telecommunications Industry Solutions (ATIS) / American National Standards Institute (ANSI), 1999.

[TFPB97] T. Turletti, S. Fosse-Parisis, and J.-C. Bolot. "Experiments with a Layered Transmission Scheme over the Internet". Research Report 3296, INRIA, November 1997.

[UCB98] UCB/LBNL/VINT. *Network simulator ns-2*, October 1998. http://www-mash.cs.berkeley.edu/ns/ns.html.

[UCL98] UCL, Dept. of Computer Science. *Robust Audio Tool (RAT)*, 1998. http://www-mice.cs.ucl.ac.uk/mice/rat.

[Uni90] International Telecommunications Union. "5-, 4-, 3-, and 2-bits Sample Embedded Adaptive Differential Pulse Code Modulation (ADPCM)". Recommendation G.727, ITU-T, 1990.

[Uni96a] International Telecommunications Union. "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)". Recommendation G.729, ITU-T, March 1996.

[Uni96b] International Telecommunications Union. "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP), Annex B: A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70". Recommendation G.729 - Annex B, ITU-T, March 1996.

[Uni96c] International Telecommunications Union. "Dual Rate Speech Coder for Multimedia Communications transmitting at 5.3 and 6.3 kbit/s". Recommendation G.723.1, ITU-T, March 1996.

[Uni96d] International Telecommunications Union. "Methods for subjective determination of transmission quality". Recommendation P.800, ITU-T, August 1996.

[Uni96e] International Telecommunications Union. "Subjective Performance Assessment of Telephone-Band and Wideband Digital Codecs". Recommendation P.830, ITU-T, February 1996.

[Uni96f] International Telecommunications Union. "Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service". Recommendation H.323, ITU-T, May 1996.

[Uni98] International Telecommunications Union. "Objective quality measurement of telephone-band (300-3400 Hz) speech codecs". Recommendation P.861, ITU-T, February 1998.

[Uni99] International Telecommunications Union. "Proposed Scope of Draft New Recommendation E.VoIPQoS, Appendix C". Study group 2, question: 3/2,5/2, ITU-T, April 1999.

[VA89] R. Valenzuela and C. Animalu. "A New Voice Packet Reconstruction Technique". In *Proceedings ICASSP*, pages 1334–1336, May 1989.

[Var93] V.K. Varma. "Testing Speech Coders for Usage in Wireless Communication System". In *Proceedings of IEEE Speech Coding Workshop*, pages 93–94, Montreal, 1993.

[VNJ99] S. Varadarajan, H. Ngo, and J.Srivastava. "Error Spreading: A Perception-Driven Approach Orthogonal to Error Handling in Continous Media Streaming". *submitted to IEEE Transaction on Networking*, 1999.

[Vor97] S. Voran. "Estimation of perceived speech quality using measuring normalizing blocks". In *Proceedings IEEE Speech Coding Workshop 1997*, pages 83–84, Pocono Manor, 1997.

[Vor99a] S. Voran. "Objective Estimation of Perceived Speech Quality - Part I: Development of the Measuring Normalizing Block Technique". *IEEE Transactions on Speech and Audio Processing*, 7(4):371–382, July 1999.

[Vor99b] S. Voran. "Objective Estimation of Perceived Speech Quality - Part II: Evaluation of the Measuring Normalizing Block Technique". *IEEE Transactions on Speech and Audio Processing*, 7(4):383–390, July 1999.

[Vor00] S. Voran. "'Short-term' objective speech quality assessment". Personal communication, ITS.T, May 2000.

[VR93] W. Verhelst and M. Roelands. "An Overlap-Add Technique based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech". In *Proceedings ICASSP*, pages 554–557, April 1993.

[WHD94] L. Wolf, R. Herrtwich, and L. Delgrossi. "Filtering Multimedia Data in Reservation-Based Internetworks". Technical Report 43.9608, IBM European Networking Center, August 1994.

[Wro97] J. Wroclawski. "Specification of the Controlled-Load Network Element Service". RFC 2211, IETF, September 1997. ftp://ftp.ietf.org/rfc/rfc2211.txt.

[WSG92] S. Wang, A. Sekey, and A. Gersho. "An Objective Measure for Predicting Subjective Quality of Speech Coders". *IEEE Journal on Selected Areas in Communications*, 10(5):819–829, June 1992.

[WZ98] R. Wittmann and M. Zitterbart. "AMnet: Active Multicasting Network". In *Proceedings ICC*, Atlanta, GA, June 1998.

[YBY98] W. Yang, M. Benbouchta, and R. Yantorno. "Performance of the Modified Bark Spectral Distortion as an Objective Speech Quality Measure". In *Proceedings ICASSP*, 1998.

[YKT95] M. Yajnik, J. Kurose, and D. Towsley. "Packet Loss Correlation in the MBone Multicast Network: Experimental Measurements and Markov Chain Models". Technical Report 95-115, Department of Computer Science, University of Massachusetts, Amherst, 1995.

[YMKT98] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. "Measurement and Modelling of the Temporal Dependence in Packet Loss". Technical Report 98-78, Department of Computer Science, University of Massachusetts, Amherst, 1998.

[Yon92] M. Yong. "Study of Voice Packet Reconstruction Methods Applied to CELP Speech Coding". In *Proceedings ICASSP*, pages II/125–128, March 1992.

[YY99] W. Yang and R. Yantorno. "Improvement of MBSD by Scaling Noise Masking Threshold and Correlation Analysis with MOS Difference Instead of MOS". In *Proceedings ICASSP*, pages 673–676, Phoenix, AZ, March 1999.

[ZF96] H. Zhu and V. S. Frost. "In-service Monitoring for Cell Loss Quality of Service Violations in ATM Networks". *IEEE/ACM Transactions on Networking*, 4(2):240–248, 1996.

[ZF99] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. Springer, Berlin 1999. Second Edition.

[ZR96] Z.Liu and R. Righter. "The Impact of Cell Dropping Policies in ATM Networks". Technical Report 3047, INRIA, November 1996.