A Fine-Grain Scalable and Channel-Adaptive Hybrid Speech Coding Scheme for Voice over Wireless IP

Improvements Through Multiple Description Coding

Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultäten
der Georg-August-Universität zu Göttingen

vorgelegt von
Marco Zibull
aus Kiel

Göttingen 2006

Diese Dissertation ist elektronisch veröffentlicht und unter
http://webdoc.sub.gwdg.de/diss/2006/zibull/zibull.pdf archiviert.

This dissertation is published electronically and available via
http://webdoc.sub.gwdg.de/diss/2006/zibull/zibull.pdf.

D7

Referent	Prof. Dr. Dieter Hogrefe
Koreferent	Prof. Dr. Stephan Waack
Tag der mündlichen Prüfung:	30.10.2006

Don't aim for success if you want it; just do what you love and believe in, and it will come naturally.

– *David Frost*

Abstract

Wireless LANs (WLANs) are being more and more widely deployed at present. They are a key element in dynamic business environments where permanent access to network resources is vital. They also provide a perfect solution for the creation of ad-hoc networks in emergency conditions within areas where dense wireless networks are in place.

Voice over IP (VoIP) is a form of voice communication that uses data networks to transmit voice signals. The signal is appropriately encoded at one end of the communication channel, sent as packets through the data network, then decoded at the receiving end and transformed back into a voice signal.

Since both technologies are sufficiently mature at the moment, VoIP over WLAN communication is being developed. However the intrinsic characteristics of each of these two technologies cause specific issues to appear that must be addressed in order to ensure a successful deployment of VoIP over WLANs. This is particularly important when considering the use of WLAN technology in the context of emergency situations.

In order to improve the service quality of voice over wireless networks an innovative MDC-based hybrid speech coding scheme was developed. Two main aspects are aimed while development. Firstly, the scheme must be channel-adaptive to withstand difficult network conditions while in parallel fine-grain scalable concerning bandwidth demands to further support the adaption mechanism, secondly. To prove the concepts, a universal network- and error simulation tool was implemented which allows for simulation of various network conditions. Achieved improvements compared to current state-of-the-art solutions are verified by objective speech quality measurement tools introduced with ITU-T P.862.2.

Acknowledgement

I would like to take this opportunity to acknowledge those who have provided the help and guidance to complete my doctoral degree at the Telematics Group, University of Goettingen.

First and foremost I would like to thank my supervisor Prof. Dr. Hogrefe for his kind support and the possibility to research under excellent conditions. His enormous trust gave me both of the freedom and pressure to quest new world. The experiences that I was allowed to make in the research projects will be a perfect base for my future professional career.

Furthermore I like to thank my colleagues and friends (especially Andre Riedel) at the Telematics Group for making the time in Goettingen unforgettable.

To my parents I owe the reason for my existence. Thank you both for pushing me to reach for the stars and providing a wonderful family. Thank you Frank for being a admirable brother.

Thank you!

Contents

List of Figures

List of Tables

Abbreviations, Acronyms, and Terms

2G	2nd generation of mobile communication technologies
3G	3rd generation of mobile communication technologies
3GPP	3rd Generation Partnership Project
4G	4th generation of mobile communication technologies
AAC	Advanced Audio Coding
ACE	Adaptive Header Compression
ACELP	Algebraic Code Excited Linear Predictor
ACK	Acknowledge
ACL	Access Control List
ACR	Absolute Category Rating
ADC	Analog Digital Converter
ADM	Adaptive Delta Modulation
ADPCM	Adaptive Pulse Code Modulation
ADSL	Asymmetric Digital Subscriber Line
AES	Advanced Encryption Standard
AMR	Adaptive Multi-Rate
ANSI	American National Standard Institute
AP	Access Point
APC	Adaptive Predictive Coding
API	Application Programming Interface
ARP	Address Resolution Protocol
ARPU	Average Revenue Per User
ARQ	Automatic Repeat reQuest
ASIC	Application-Specific Integrated Circuit
ATC	Advanced Transfer Cache
ATH	Absolute Threshold of Hearing
ATM	Asynchronous Transfer Mode
AWGN	Additive White Gaussian Noise
BAN	Body Area Network
BBNGN	Broadband Next Generation Network
BEP	Bit Error Probability
BER	Bit Error Rate
BMGL	Synchronous Key-stream Generator (**B**lum, **M**icali, **G**oldreich and **L**evin)

BNF	Backus Naur Form
BPSK	Binary Phase-Shift Keying
BSAC	Bit-Sliced Arithmetic Coding
BSS	Basic Service Set
BTS	Base Transmission Station
BWA	Broadband Wireless Access
CALEA	Communications Assistance Law Enforcement Act
CBC	Cipher Block Chaining
CCITT	Comité Consultatif International Télégraphique & Téléphonique
CCK	Complementary Code Keying
CCR	Comparison Category Rating
CCSA	China Communications Standard Association
CDMA	Code Division Multiple Access
CELP	Code-Excited Linear Prediction
CID	Context Identifier
CNAME	Canonical Name
CNG	Comfort Noise Generation
COPS	Common Open Policy Service
CPL	Call Processing Language
CPTR	Compressed Real-time Protocol Header
CRC	Cyclic Redundancy Check
CRTP	Compressed RTP
CSCF	Call Session Control Function
CSMA	Carrier Sense Multiple Access
CTCP	Compressed TCP
CWT	Continuous Wavelet Transform
DAC	Digital to Analog Converter
DCF	Distributed Coordination Function
DCR	Degradation Category Rating
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DFS	Direct Frequency Selection
DFT	Discrete Fourier Transformation
DHCP	Dynamic Host Configuration Protocol
DIFS	DCF Interframe Space
DL	Downlink
DLFP	Downlink Frame Prefix
DLSR	Delay Since Last Sender Report
DMOS	Double-diffused Metal Oxide Semiconductor
DNS	Domain Name Service

DPCM	Differential PCM
DPSK	Differential Phase Shift Keying
DSL	Digital Subscriber Line
DSP	Digital Signal Processing
DSSS	Direct Sequence Spread Spectrum
DTMF	Dual Tone Multi Frequency
DTX	Discontinuous Transmission
DWT	Discrete Wavelet Transform
EAP	Extensible Authentication Protocol
ECRTP	Enhanced Compressed RTP
EFR	Enhanced Full Rate
EIRP	Equivalent/Effective Isotropic(ally) Radiated Power
ESP	Encapsulated Security Payload
ETR	ETSI Technical Report
ETS	Emergency Telecommunication Service
ETSI-TISPAN	European Telecommunication Standard Institute - Telecommunications and Internet Converged Services and Protocols for Advanced Networking
EZW	Embedded Wavelet Zero-tree
FCC	Federal Communications Commission
FCH	Frame Control Header
FDD	Frequency-division Duplex
FEC	Forward Error Control/Correction
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum
FIFO	First In First Out
FIPS	Federal Information Processing Standard
FIR	Finite Impulse Response
FRF.12	concerns fragmentation of large frames to smaller units and interleaving of real-time frames. Thereby voice data can be transmitted in conjunction with other data frames without considerable delays - see `http://www.frforum.com`
FSMC	Finite-state Markov Chain
FTP	File Transfer Protocol
FWT	Fast Wavelet Transform
GFSK	Gaussian Frequency-Shift Keying
GGSN	GPRS Gateway Support Node
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HDSL	High bit rate Digital Subscriber Line

HDTV	High-Definition TeleVision
HSDPA	High Speed Downlink Packet Access
HTTP	Hypertext Transport (or Transfer) Protocol
ICMP	Internet Control Message Protocol
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering task force
IIR	Infinite-duration Impulse-Response
IKE	Internet Key Exchange
IMA	Interactive Multimedia Association
IP	Internet Protocol
IPHC	IP Header Compression
IPT	IP Telephony
ISDN	Integrated Services Digital Network
ISM	Industry, Scientific, Medical
ISO	International Standards Organization
ISP	Internet Service Provider
ISUP	Integrated Services User Part (ISDN)
ITU-T	International Telecommunication Union - Telecom.
JPEG	Joint Photographic Experts Group
JSC	Joint Source-Channel
KLT	Karhunen-Loève-Transformation
LAN	Local Area Network
LEAP	Lightweight Extensible Authentication Protocol
LFI	Link Fragmentation and Interleaving
LIC	List of Insignificant Coefficients
LIFO	Last In First Out
LNP	Local Number Portability
LOS	Line of Sight
LPC	Linear Predictive Coding
LSB	Least Significant Bit
LSP	Label Switch Path
LSR	Label Switch Router
LTP	Long-Term Predictor
MAC	Media Access Control
MAN	Metropolitan Area Network
MANET	Mobile Ad hoc Networking
MBWA	Mobile Broadband Wireless Access
MCU	Multipoint Control Unit
MDC	Multiple Description Coding
MDCT	Modified Discrete Cosine Transformation

MGCP	Media Gateway Control Protocol
MIME	Multi-purpose Internet Mail Extension
MIMO	Multiple Input Multiple Output
MIPS	Million Instructions Per Second
MJPEG	Motion Joint Picture Expert Group
MMSE	Minimum Mean Squared Error
MOS	Mean Opinion Score
MP3	MPEG layer 3
MPEG	Motion Pictures Expert Group
MPLS	Multi-Protocol Label Switching
MRD	Marketing Requirements Document
MSB	Most Significant Bit
MSE	Mean Square Error
MTA	Message Transfer Agent
MTU	Maximum Transmission Unit
NAV	Network Allocation Vector
NGN	Next Generation Network
NIC	Network Interface Card
NII	National Information Infrastructure
NLOS	Non Line of Sight
NTP	Network Time Protocol
NTT	Nippon Telegraph and Telephone Corporation
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection
OSPF	Open Shortest Path First - a routing mechanism
OTA	Over-the-Air
P-CSCF	Proxy Call Session Control Function
P2P	Peer-to-Peer
PAM	Pulse Amplitude Modulation
PAN	Personal Area Network
PBCC	Packet Binary Convolutional Coding
PC	Point Coordinator
PCF	Point Coordination Function
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PDF	Probability Density Function
PDP	Policy Decision Points
PDU	Protocol Data Unit
PEP	Policy Enforcement Points
PER	Packet Error Rate
PESQ	Perceptual Evaluation of Speech Quality

PHB	Per-Hop-Behavior
PHY	Physical Layer
PIFS	PCF Interframe Space
PLC	Packet Loss Concealment
PMP	Point-to-Multipoint
PN	Pseudo-random Numerical
POTS	Plain Old Telephone Service
PPP	Point to Point Protocol
PSAP	Public Safety Answering Point
PSK	Phase-Shift Keying
PSQM	Perceptual Speech Quality Measure
PSTN	Public Switched Telephone Network
PZW	Perceptual Zero-tree Wavelet
QAM	Quadrature Amplitude Modulation
QCELP	Qualcomm CELP
QoS	Quality of Service
QPSK	Quadrature Phase-Shift Keying
RAN	Radio Access Network
RAS	Remote Access Services
RC4	Rivest Cypher algorithm version 4
RED	Random Early Dropping
RF	Radio Frequency
RFC	Request For Comments
RGB	Red Green Blue
RMS	Root Mean Square
ROCCO	Robust Checksum-based header Compression
ROHC	Robust Header Compression
RPE	Regular Pulse Excited
RSA	Rivest Shamir Adelman
RSSI	Received Signal Strength Indicator
RST	Reset
RSVP	Resource Reservation Setup Protocol
RTCP	Real-Time Transport Control Protocol
RTI	Real-Time Intolerant
RTP	Real-Time (Transport) Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round Trip Turnaround Delay
SACK	Selective Acknowledgment
SAP	Service Access Point
SBC	Subband Coding
SC	Single Carrier

SDES	Source Description
SDM	Spatial Division Multiplexing
SDP	Session Description Protocol
SDR	Software Defined Radio
SDU	Service Data Unit
SFN	Single Frequency Network
SHF	Super High Frequency
SIFS	Short Interframe Space
SIP	Session Initiation Protocol
SISO	Single Input Single Output
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SNR	Signal-to-Noise Ratio
SPHIT	Set Partitioning In Hierarchical Trees
SPL	Sound Pressure Level
SPOF	Single Point Of Failure
SRP	Selective Repeat Protocol
SRTP	Secure RTP
SS	Subscriber Station
SS7	Signaling System 7
SSH	Secure Shell
SSL	Secure Socket Layer
STFT	Short Time Fourier Transform
STP	Short-term Predictor
TCA	Traffic Conditioning Agreement
TCP	Transport Control Protocol
TDD	Time Division Duplex
TDM	Time-division multiplexing
TLS	Transport Layer Security
ToS	Type of Service field in IP header used to differentiate traffic flows
TTC	Telecommunications Technology Council
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UGS	Unsolicited Grant Service
UHF	Ultra High Frequency
UL	Uplink
UMTS	Universal Mobile Telecommunication System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

VAD	Voice Activity Detection
VCR	Video Cassette Recorder
VDSL	Very high bit rate Digital Subscriber Line
VJHC	Van Jacobsen Header Compression
VLC	Variable Length Coding
VoIP	Voice over IP
VoWLAN	Voice over Wireless LAN
VPN	Virtual Private Network
VSF-OFDM	Variable-Spreading-factor Spread Orthogonal Frequency Division Multiplexing
WAN	Wide Area Network
WCDMA	Wideband Code Division Multiple Access
WECA	Wireless Ethernet Compatibility Alliance
WEP	Wired Equivalent Privacy
WFQ	Weighted Fair Queuing
WIBRO	Wireless Broadband
Wi-Fi	Wireless Fidelity
WiMAX	Interoperability standard of IEEE 802.16 (e)
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPA/WPA2	Wi-Fi Protected Access
WRC	World Radio Conference
WRED	Weighted Random Early Detection
WVCS	Wireless Voice Communication System
WWI	Wireless World Initiative
WWRF	Wireless World Research Forum
YUV	Color Model Y: Luminance, U, Cb: Chroma channel, U axis, blue component) V, Cr: Chroma channel, V axis, red component

1 Introduction

Wireless access works well for cordless and mobile phones. Millions of telephone calls are conducted over cordless and cellular telecommunication systems every day. The transmission of voice over wireless links is highly optimized. The common wireless systems such as DECT, GSM, and UMTS are highly cost effective and efficient. These technologies are based on substantial research results in the field of communication and signal processing theory and perform best when one application (such as telephony) is transmitted over one channel (e.g. a wireless link) using a dedicated circuit switched link. This knowledge is affirmed by theoretical research results considering joint source-channel coding [CSR04, KW69], which jointly optimizes the source (e.g. speech) for a single channel (e.g. the wireless link).

Contrary the Internet allows the joint transport of many different multimedia services such as web, games, video and audio. Multiple applications can be transmitted concurrently. The transmission can take place over multiple links in row and even on multiple routes in parallel. But the Internet, as other packet-switched networks too, cannot be as resource efficient on wireless links because packet-switching comes at the cost of controlling and negotiating the transmission schedule of each packet. Thus, a single IP-based telephone call requires more communication resources than a circuit-switched based call. But due to the statistical multiplexing gain of packet-switched networks and considering the overall system costs, Internet based communication might be cheaper and will be important in future - even for telephony services as extensive analyses by Hwang [Hwa01] showed.

Best-Effort packet-switched networks, like the Internet do not offer reliable transmission of packets to applications with real-time constraints such as voice. Thus, the loss of packets impairs the application-level utility. Therefore improving the transmission performance of the Internet is a worthwhile objective. It is especially important in wireless communication networks, because they often have a low capacity, tight energy constraints, and time varying channel qualities. Wireless access is frequently used but it is a bottleneck in current and will likely remain a bottleneck in future broadband communication systems.

In this thesis, the question on how to increase the robustness and efficiency of IP-based telephony over wireless links is addressed. The goal is to enable the usage of wireless, mobile technologies for Internet based services, especially telephony, with an equal or better level of user satisfaction as DECT, GSM and UMTS can already achieve.

Keywords: VoIP, VoWLAN, Multiple Description Coding, Speech Processing, Wireless, Wavelets

1.1 Technology Overview

The increased demands for mobility and flexibility in our daily life are demands that lead the development from wired LANs to wireless LANs (WLANs). With the advent and ubiquitous of wireless technology, a wide range of advanced services are expected to be supported including appealing services that currently exist in wired systems. Nevertheless, the resource constraints in wireless environment may render difficulty to realizing all the desirable services. Today a wired LAN can offer users high bit rates to meet the requirements of bandwidth consuming services like video conferences, streaming video etc. With this in mind a user of a WLAN will have high demands on the system and will not accept too much degradation in performance to achieve mobility and flexibility.

There are at the moment IP phones which are similar in shape with the regular telephones but instead of being connected to a phone socket they are plugged into a network connection. There exist as well IP phones with built-in wireless support. Hence the act of making a phone call using VoIP can be identical to that of using regular phones. The quality of the communication itself can be different however, and it is the most important aspect of the transition from standard telephone networks to Internet telephony. One reason for such a transition is that VoIP communication is more flexible than standard telephony. By making the appropriate choice for the codec one can control the amount of bandwidth required and one determines the intrinsic associated quality.

However, since the communication channel is not reserved but shared with other applications, voice packets can arrive at the receiver with a different inter-packet gap (jitter) than they had at the sender, out of order, and some of them can even be lost. Assessing the relationship between precisely these factors, as quantified by means of network QoS parameters, and the User-Perceived Quality (UPQ) of VoIP communication is a prerequisite for any performance and dependability analysis of VoIP over WLAN.

Users of wireless networks are involved in several domains: enterprise (managers, IT personnel and other campus mobile workers), education (principals, professors, maintenance staff), health (doctors, nurses, technicians), manufacturing (supervisors, quality control people, experts), retail (managers, inventory clerks, shipping/receiving personnel). Several reasons make WLANs essential for their activity. These users are highly mobile, either because they don't have a desk or because they are away from their desk a significant amount of time. They need to be instantly reachable (currently the primary communication strategy is voice, plus messaging).

They also require instant access to key data. In this context VoIP over WLAN (VoWLAN) appears as the most obvious solution for the voice communication of mobile type that these users need. IP telephony has low-bandwidth requirements (below 64 Kbps), therefore one may assume that VoIP is easy to use on wireless LANs. However combining the two technologies today is difficult. Experiments show that even a small amount of data traffic on the same network can lead to seriously degraded audio quality and dropped calls, even with QoS features enabled [New05].

The main reason is that, when handling voice and data traffic on the same network, contention must be managed in terms of delay and jitter rather than forwarding rates. Most vendors only begin to adjust their products for voice/data convergence, therefore performance of VoIP (and real-time applications in general) over wireless media can be an issue. The difficulty in finding appropriate QoS solutions derives from some of the inherent properties of WLANs.

We'll analyse first the everyday situation when no contention management techniques is used. Under these circumstances systems usually encounter no problem in delivering near-toll-quality audio, even without QoS enforcement, when only a small number of calls are active. Depending on system features, a number of simultaneous calls of six and above may lead to decreased audio quality, and some of the calls may even be dropped [New05]. If background data is added to the scenario then VoIP performance deteriorates seriously. This is the case even when the total amount of traffic doesn't exceed half of the sustainable rate of a network (3 Mbps compared to 6 Mbps). This situation is not unexpected given that the lack of QoS implies that there is no control over the interaction between different application traffic flows. Not managing contention leads to unpredictable results, which can have adverse effects on real-time applications such as VoIP.

Under such circumstances the *miracle solution* in fixed networks is to throw bandwidth at the problem and over-provision the network capacity by a couple of orders of magnitude. It is a known fact that on many existing 1 Gbps and higher-rate networks the average utilization is below 1%. Unfortunately this is not feasible for wireless networks, where theoretical rates of only 54 Mbps are still a luxury. The industry realized that to deploy successfully VoIP on WLANs the networks need to be optimised for voice traffic. QoS enforcement is nowadays recommended by WLAN equipment manufacturers when deploying multiple applications with different requirements on the same WLAN.

Since no QoS over WLAN standard existed until recently, most manufacturers, both for WLAN equipment and WLAN phones, implemented either proprietary QoS mechanisms or preliminary versions of 802.11e (such as a subset of 802.11e, the Wireless Media Enhancements protocol). Hence there is no unified way to manage quality in current day WLANs.

The QoS mechanism most often supplied is related to bandwidth management.

Transport Medium	BER	Interfere Sensitiveness
Air	$10^{-3} - 10^{-5}$	High
Twisted Pair Cable	10^{-5}	High
Baseband Coax	$10^{-7} - 10^{-8}$	Possible through elect.magnetic fields
Broadband Coax	$10^{-8} - 10^{-9}$	"
Fiber Cable	$< 10^{-12}$	Very Low

Table 1.1: BER characteristics for different transport media

Existing QoS implementations in WLAN devices allow the allocation of bandwidth to a given workgroup. Allocating bandwidth to a given workgroup is useful in distinguishing between employees and guests associated with the enterprise network. Some devices, such as Aruba and Cisco products, can also allocate bandwidth on a per-user basis. However in the case of VoIP and other real-time applications it is the timely servicing of high-priority traffic that matters, not the average data rates.

On the other hand further improvement hide in aspects of infrastructure design, protocols - especially adaptation to wireless link needs -, compression of header information as for multimedia transmission a considerable amount of the total traffic is contributed by packet-header overhead. Hence, the general rule *the smaller the packet-size the more efficient the transmission is* could yield just to the opposite by even introducing more packet-header overhead if, to reach the same bit rate, a multiple of packets have to be transported instead.

1.2 Identified Weaknesses and Ideas

Current challenges in the field of VoWLAN are security issues, roaming, missing standards (e.g. QoS) and most important – technology constrained, weak natured and therefore unreliable transmission channel characteristics. Much effort has been made to overcome these problems but they still exist. In this thesis, especially the last major problem is engaged. Table 1.1 illustrates typical bit error characteristics for different transport media [RS98].

Hence, for a typical 802.11g connection with an actual bit rate of 18 Mbps, 180 Bits up to 18.000 Bits are invalid/erroneous yielding to enormous problems while providing IP telephony over wireless links. To overcome this fact and to provide better quality VoWLAN-services, innovative speech coding schemes have to be developed, as current state-of-the-art codecs are primarily developed for wired or generally for circuit-switched networks, hence focussing other possible difficulties.

While thinking of a typical HotSpot situation where people communicating in several ways with different requirements to the underlying network the solution to various service-quality problems might be solved by introducing fine-grain scalable bit rate adaption in order to withstand difficult network situations. With the pos-

sibility to adopt the respective bit rate of a single participant fine-granularly, the degradation of the current used service isn't that large while trying to compensate current network difficulties as by reducing the bit rate in large steps.

In an economical point of view, the customer must be attracted by robust quality services, which might be guaranteed by channel-adaption mechanisms combined with fine-grain scalability. Additionally, customers might be attracted, if superior service quality could be enabled. For telephony based services, this could be possible by introducing wideband speech processing which yields to enriched and high intelligibly voice communication.

To fulfill all these requirements, an innovative hybrid speech coding scheme was developed. The scheme is hybrid in the manner of dealing with two different encoders to enable more robust and therefore increased VoWLAN-quality. One of the encoders is a CELP-based coder while the other is WPT-based encoder (Wavelet Packet Transform) with additional psychoacoustic considerations. The developed scheme is on the one hand channel-adaptive and on the other hand fine-grain scalable.

1.3 Structure of This Thesis

This thesis is organized as follows: after this introduction the thesis continues with chapters considering IP telephony (2) with respect to both wired and wireless issues followed by a detailed discussion of wireless networks, their specific protocols and their relationship to VoWLAN (3). Additionally, the developed network- and error simulation is presented here. Elaborately, speech coding and processing is discussed in chapter (4). In chapter (5) the developed channel-adaption mechanism is presented. The fine-grain scalability features of the introduced speech coding scheme is discussed in chapter (6). In chapter (7) MDC (Multiple Description Coding) issues are presented and how they are applied to to the speech coding scheme developed in this thesis. Finally, chapter (8) presents simulation results for various different error classes.

This thesis ends with a conclusion discussing the results of all chapters and an outlook on future research challenges.

2 IP-based Telephony

2.1 Introduction

Internet Telephony allows to offer voice services across networks using Internet protocols. IP Telephony consists among others of signalling and transmission protocols. The signalling protocols (ITU-T H.323 [SR98] or IETF SIP [DF99]) establish, control and terminate a telephone call. The principle components of a Voice over IP (VoIP) system, which cover the end-to-end transmission of voice, are displayed in figure 2.1. First, at the source the analogue processing, digitalization, encoding, packetization, and protocol processing are performed. Then, the resulting packets are transmitted through the network, comprising of IP networks. At the receiver, protocol entities process the packets and deliver them to the playout scheduler/buffer. In the next step, the speech frames are decoded and played out. Because telephony consists of bidirectional transmission a similar technique is taking place in the opposite direction. In the following, the principal components of VoIP systems will be discussed in detail.

2.2 Voice over IP

Today most of the telephony is still made on the traditional Public Switched Telephone Network (PSTN). This means that a call reserves the connection between the two users and no one else can use this connection. The difference with Internet Telephony, also called Voice-over-IP (VoIP), is that the transport is made on an IP-network. It is possible to send packets between two or more parties without reserving the connection. Voice over IP is an extensive subject, but at the core it comes down to trying to transport speech signals in an acceptable way from sender to destination over an IP network. The definition of *acceptable* depends on the particular situation we are dealing with. If, for example, speech signals are being transported as part of a real-time communication between two persons, it will mean that the real-time aspects of this conversation must be respected: the overall delay between sending and receiving should be low to avoid irritably long gaps of silence. If, however, speech signals are being transmitted as part of a one-way process - e.g. an on-line radio show or a lecture - the delay constraints are less strict since the interactive aspect is no longer present.

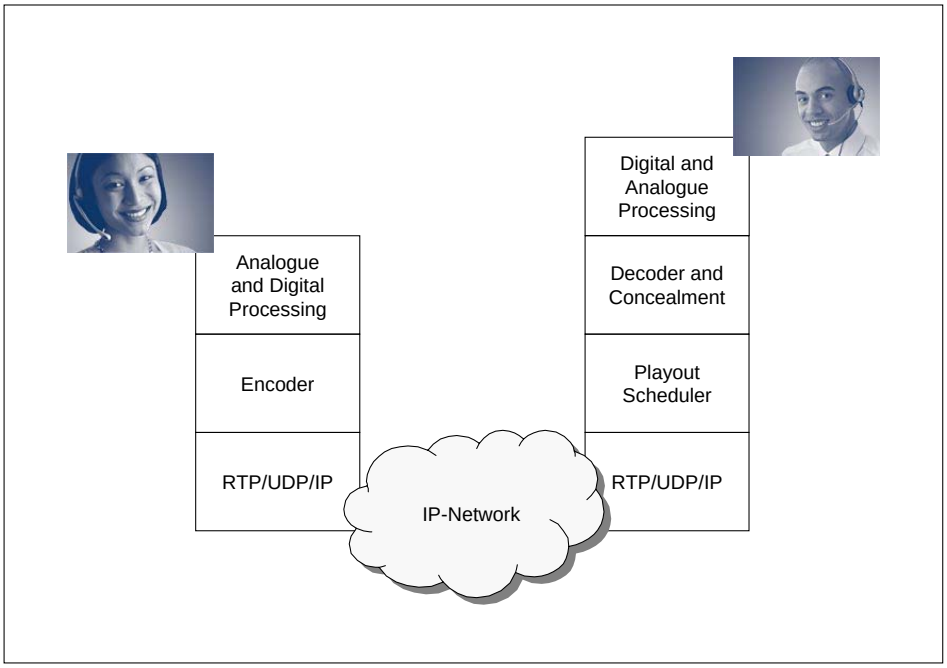


Figure 2.1: A VoIP telephone call

The first kind of use is the telephone alternative. This means that some kind of VoIP system is used to make a voice call to another person. This can be done in several ways. First of all, if a PC that can be connected to some kind of network is available, it can be used to make a call to somebody else who is also connected to that network. This PC would then be equipped with speakers and a microphone and a VoIP application would be used to make the call. The PC could have a direct connection to a computer network but a connection through a dial-up link is also possible. The second case is a slight variation of the first one. In this case, a telephone is connected to the PC and used in a similar way as you would when making a normal call. The PC does all the necessary work to set up the call and to transmit the speech signals. This also means that the PC has to be switched on before the call can be made. This type of configuration might be easier to use for people who do not work with computers often. As with the previous case, the connection to the network can be either direct or through a dial-up link.

Finally, the use of a PC and the requirement of a network could be omitted by the use of a VoIP gateway. This is a special device that connects the PSTN with a computer network and performs the necessary actions and conversations to make the call possible. This configuration would be best for persons who do not have a PC. It is probably also the easiest to use, since most people are familiar with using

a telephone and there does not have to be a PC around.

With VoIP, not only the normal telephone features can be made possible, but also a wide range of new features could be created, especially when using VoIP on a PC. Whiteboarding could be used to make working together easier, a log book with information about incoming and outgoing calls could be kept, conversations could easily be recorded and security could be enhanced by using encryption algorithms. When using VoIP over a Local Area Network (LAN), there is usually plenty of bandwidth available and the delay between sending and receiving is usually very low. Here, VoIP can often be used without problems. But when a Wide Area Network (WAN) is used - the Internet for example - problems can arise. One problem is the delay: while the delay on a LAN is usually very low, on a WAN this is not necessarily true. If the delay gets too large, the conversation will not be very pleasant. Another problem is the quality of the speech signals. When certain routes get too heavily loaded, packets on the WAN will be lost. These lost packets cause interruptions in the speech signal. In turn, these interruptions, when large enough, can also disturb the conversation. To alleviate the load, a lot of VoIP programs use compression techniques. However, compression often causes a certain degradation of the signal. This may or may not be disturbing to the listener, but with heavy compression, telephone quality will rarely be achieved.

VoIP techniques can be used for a wide variety of other applications which require voice or sound in general to be transmitted over a computer network and where timing and synchronization are important issues. The same techniques also work when it is not sound, but video information which has to be transmitted.

To be able to send speech information across a computer network, the speech signal has to be encoded into a digital representation. In general, the signal will be detected by a microphone and transformed into a digital one by a special device, a sound card for example. This process is called grabbing or digitization and it is often also referred to as sampling. To maintain the real-time aspects of the conversation, it is necessary for the receiver to start receiving the signal as soon as possible after the sender has started it. To accomplish this, at regular small intervals blocks of digitized speech information are sent across the network, where they can be processed by the receiver. When a digitized block is received, it has to be transformed back into an audio signal. The output of the process will usually go to speakers, so that the receiver will be able to hear what the sender is saying. Like the digitization step, this process is also done by a special device. In essence, regeneration is the reverse operation of grabbing. Several things have to be considered before transforming the digitized signal. First of all, if multiple persons are allowed to talk at the same time, like in a virtual environment, the speech signals of those persons have to mixed together at the receiver. Second, when sending blocks of data across a network, there will be tiny variations in the time it takes each block to get to the destination. If unlucky, these variations can even be rather large. Suppose we start

playing back the voice signal in a block as soon as we received it. Because of the jitter, it is possible that the next block has not yet arrived when the output of the first one is finished. To overcome this problem some buffering will have to be performed to make sure that when finished with one block, the next will be available. However, this buffering will introduce a certain amount of delay so care must be taken to avoid that the overall delay will be too large.

The digitized information requires a certain amount of the available bandwidth of the connection. Very often compression schemes are used to reduce the required bandwidth for voice communication. Several types of compression exist. Some of them use general compression techniques which are also used on other kinds of data; other types try to exploit the fact that we are dealing with voice information to achieve large compression ratios. Of course, combinations are also possible. Once the compressed blocks with speech data reach the destination, they have to be decompressed. This means that given the compressed signal, the original digitized signal has to be reconstructed as good as possible. The decompression is very closely related to compression as it must be the inverse operation of the compression scheme that was used. Compression is very important when the connection is slow, like with dial-up links for example. Finally, the blocks have to be sent from source to destination, across the network. Some timing information should probably be added to the data, to make it possible for the receiver to reconstruct the exact order of the blocks. This is necessary because blocks may be lost, delayed or duplicated during the transfer.

2.2.1 Protocols

Several VoIP protocols have been suggested to realize VoIP communication. Two major protocols or frameworks should be emphasized:

2.2.1.1 H.323

The ITU-T document about H.323 is a recommendation for multimedia conferencing over packet based networks without QoS support. It is a part of the H.32X series of recommendations which all describe multimedia conferencing but over different types of networks. These recommendations are:

- H.320 Narrowband Integrated Services Digital Network (N-ISDN)
- H.321 Broadband Integrated Services Digital Network (ISDN)
- H.322 Guaranteed bandwidth packet switched network
- H.323 Non-guaranteed bandwidth packet switched network

- H.324 The analogue phone system

End systems conforming to the H.323 recommendation can communicate with each other, either point-to-point or in a multipoint conference. These end systems may have different capabilities, but each must at least support G.711 audio encoding. Video support and other audio coders are optional. H.323 also defines how to do general data transfers, but this feature also is optional. The recommendation allows communication with end systems on a different type of network, conforming to other H.32X standards. This requires special devices which connect to the different networks and do the necessary conversions. Management and accounting support are also provided. This way it is possible to specify for example the maximum amount of bandwidth that may be occupied with H.323 calls. Accounting is provided to support billing of the callers. The H.323 recommendation defines a framework for the development of supplementary services. Currently, two such services are already defined: call transfer and call forwarding. Finally, since packet based networks - like IP networks - are often not very secure, H.323 defines several mechanisms to provide better security [Sta03b].

Four components are specified in recommendation H.323: terminals, gateways, gatekeepers and multipoint control units (MCUs). A terminal is a system where H.323 data and signaling streams originate and terminate. It was already mentioned that such a system must at least be capable of handling G.711 audio. A gateway is a device which allows H.323 capable systems to communicate with other H.32X systems. Gateways connect the different networks together and perform the necessary transformations. For example, it may be necessary to change signaling information or to use another audio encoding. A gateway is optional in a H.323 enabled network. A gatekeeper is an optional component, but is very useful when present. When a gatekeeper is present, all terminals, gateways and MCUs must be registered with it. Two important services are provided by a gatekeeper. The first one is address translation from an alias - an international phone number for example - to a network address - an IP address for example. The second major service of a gatekeeper is bandwidth management. A gatekeeper could be configured to limit the bandwidth used by H.323 calls or to only allow a certain amount of simultaneous calls. An optional feature of a gatekeeper is to route calls. When a call is routed through a gatekeeper, this allows more effective control and more information about the call. This could be used to bill calls or to re-route a call to another system when a user is unavailable at the called endpoint. A MCU is used for conferences between three or more endpoints. It contains a multipoint controller (MC) and possibly a number of multipoint processors (MPs). Participants send their control information to the MC so that endpoint capabilities can be exchanged and communication parameters can be negotiated. A MP is used to process the incoming media, for example to mix several streams together. Three models for multipoint conferencing are defined. In

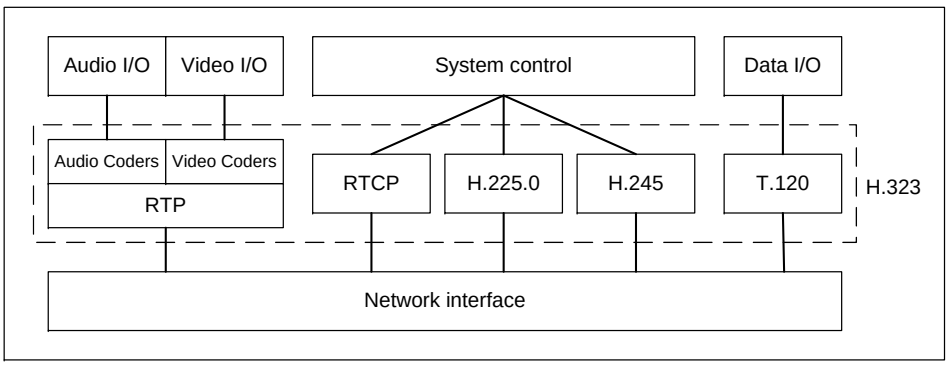


Figure 2.2: H.323 architecture
(following [Kar99])

all models each participant sends its control information is to the MCU, where it can be processed by the MC. In the centralized model, each participant also sends its media to the MCU. In the decentralized model the different media are distributed by multicasting them. In the hybrid model, some participants use multicasting to distribute the media, others send their media directly to the MCU.

The H.323 recommendation is often called an *umbrella specification*. This is because it uses several other ITU-T recommendations to provide its functionality. The structure of the H.323 architecture is illustrated in figure 2.2.

The audio coders are the ITU-T G.-standards. The video coders defined in the recommendation are H.261 and H.263. The H.263 coder was designed for low bit rate transmission but is more complex than H.261. Both audio and video are encapsulated in RTP packets and then transmitted across the network. Additional information about these transmissions is provided by RTCP.

Before two or more parties can communicate with each other, the call first has got to be set up. This is done using mechanisms defined in H.225.0 and H.245. A part of the H.225.0 recommendation specifies how a call should be set up and torn down. When the call has been established, the capabilities of the involved end systems are exchanged so that each end system can select the appropriate coders. This capability exchange is done by H.245, which also defines other functions, for example the opening and closing of logical channels to transport audio and video.

Another part of the H.225.0 recommendation specifies how the interaction with a gatekeeper should be done. This is a done by a protocol called RAS, which stands for Registration, Admission and Status. The RAS functions include gatekeeper discovery and endpoint registration with a gatekeeper. Functions like bandwidth management and admission control are also done by RAS messages.

H.323 end systems can also exchange general data with each other. How this should be done is specified in the T.120 recommendation. Like H.323, this is also an umbrella recommendation, defining how to use other protocols to exchange data.

How security services should be provided is defined in recommendation H.235. Authentication is provided by admission control of endpoints, which is done by a gatekeeper. Data integrity and privacy are implemented using encryption techniques. Finally, non-repudiation is also provided by a gatekeeper. Non-repudiation means that nobody can deny that he participated in a call.

2.2.1.2 SIP

IETF (Internet Engineering Task Force) and other working groups have put lot of efforts to come up with a protocol, which could lay standards for Internet Telephony. These efforts gave birth to Session Initiation Protocol (SIP). The imminent acceptance of the SIP as an official IETF standard marks an important milestone to the IP telephony industry. That milestone is the merging of Internet based distributed technologies with traditional telephony. SIP standardization has moved from MMUSIC (Multiparty Multimedia Session Control) to the SIP Working Group (WG). SIP WG has primary responsibility for the future development of SIP, but SIP-related work occurs in a number of IETF working groups.

SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions (conferences) or Internet telephony calls. SIP can invite participants to unicast and multicast sessions; the initiator does not necessarily have to be a member of the session to which it is inviting. Media and participants can be added to an existing session. SIP transparently supports name mapping and redirection services, allowing the implementation of ISDN and intelligent network telephony subscriber services. These facilities also enable personal mobility, which provides capability to reach a called party at a single, location-independent address. As a traditional text-based Internet protocol, it resembles the hypertext transfer protocol (HTTP) and simple mail transfer protocol (SMTP). Like these protocols, SIP is a textual protocol based on the client-server model, with requests generated by one entity (the client), and sent to a receiving entity (the server) which responds them. A request invokes a method on the server and can be sent either over TCP or UDP. The most important SIP method, of the currently six, is the INVITE method, used to initiate a call between a client and a server. The other SIP methods are ACK, OPTIONS, BYE, CANCEL and REGISTER. A new method INFO has also been proposed as part of SIP-extensions and is detailed in RFC 2976 [Don00]. SIP uses Session Description Protocol (SDP) for media description. SIP supports five aspects of establishing and terminating multimedia communications; which are user location, user capabilities, user availability, call setup and call handling. SIP 2.0 is detailed in RFC 2543 [HSSR99].

There are three components in SIP architecture, namely, user agents, network servers and SIP messages.

User Agents A user agent is an application that acts on behalf of a user. It can act both as a User Agent Client (UAC) and User Agent Server (UAS); as the user probably is wishing to both be able to call and to be called. UAC is used to initiate a SIP request. UAS receives requests and returns responses on behalf of the user. The response accepts, rejects or redirects the request. These user agents contain the full SIP state machine and can be used without intermediate servers.

Network Servers There are three kinds of network servers, namely, proxy servers, redirect servers and registrar servers. SIP servers, on occasion, will need to contact an external location server to determine callee's possible location(s).

A *SIP proxy* server forwards requests to the next server after deciding which it should be. A proxy server interprets, and, if necessary, rewrites a request message before forwarding it. This next server could be any kind of SIP server; the proxy does not know and does not have to know. Before the request has reached the UAS it may have traversed several servers. As a proxy server issues both requests and responses it contains both a client and a server. A proxy server can either be stateful or stateless. When stateful, a proxy remembers the incoming request, which generated outgoing requests, and the out-going requests. A stateless proxy forgets all information once an outgoing request is generated. A proxy server can fork the incoming request to multiple locations if the callee has multiple-location registrations with the server. A forking proxy is always stateful because it needs to remember the states of all the branches to which the incoming SIP request was forked.

Redirect server, does not forward requests to the next server. It accepts a SIP request and maps the address to zero or more new addresses and returns these addresses to the client and then client can contact the server directly. Unlike a proxy server, it does not initiate its own SIP request. Unlike a user agent server, it does not accept calls.

A *registrar* is a server that accepts REGISTER requests and maintains the availability details of various servers and clients. A registrar is typically co-located with a proxy or redirect server and may sometimes offer location services also.

SIP Messages are typically of type requests and responses. Requests flow from client to a server and a response from a server to a client. These, requests and responses, include different headers to describe the details of the communication. SIP being a text-based protocol makes its header largely self-describing and minimizes the cost of entry. SIP maintains a common structure of all messages and their

header fields, allowing a generic parser to be written. Request and response use a generic message format, which consists of a start-line, one or more header-fields (headers), an empty line indicating the end of the header fields, and an optional message-body. SIP was designed for character-set independence, so that any field can contain any ISO 10646 character. Together with the ability to indicate languages of enclosed content and language preferences of the requester, SIP is well suited for international use. To make SIP signaling more secure, encryption and authorization can be used. Encryption can for example be used to prevent packet sniffers and other eavesdroppers from seeing who is calling whom. Authorization is used to prevent an active attacker from modifying and replaying SIP requests and responses.

SIP header fields are similar to HTTP header fields in both syntax and semantics. Messages use header-fields to specify such things as caller, callee, the path of the message, type and length of message body and so on. Some of the header fields are used in all messages, the rest is used when appropriate. A SIP application does not need to understand all these headers, though it is desirable. The entity receiving simply silently ignores headers that it does not understand. The order in which the headers appear is generally of no importance, except for the Via field and that hop-by-hop headers appear before end-to-end headers. There are 44 SIP headers listed in the Internet draft of RFC 2543 [HSSR99], dated November 2000. These headers can be divided into four different groups of headers:

- *General header* fields apply to both request and response messages
- *Entity header* fields define information about the message body or, if no body is present, about the resources identified by the request
- *Request header* fields act as request modifiers and allow the client to pass additional information about the request, and about the client itself, to the server
- *Response header* fields allow the server to pass additional information about the response, which cannot be placed in the Start-Line (in responses it is called Status-Line).

These header fields give information about the server and about further access to the resource identified by the Request-URI. The request is characterized by the Start-Line, called Request-Line and starts with a method token followed by a Request-URI and the protocol version. There are six different kinds of requests in the current version of SIP (version 2.0). They are referred to as methods and are here listed with their functionality. New SIP method INFO is also proposed as part of the SIP-extensions.

- *REGISTER* conveys information about a user's location to a SIP server
- *INVITE* this method indicates that the user or service is being invited to participate in a session. The message body MAY contain a description of the session to which the callee is being invited. For a two-party call, the caller indicates the type of media it is able to receive as well as their parameters such as network destination. A success response indicates in its message body which media the callee wishes to receive
- *ACK* this request confirms that the client has received a final response to an INVITE. ACK is used only with INVITE requests. It may contain a message body with the final session description to be used by the callee. If the message body is empty, the callee uses the session description in the INVITE request
- *OPTIONS* this method queries the capabilities of the server/end system, but does not set up a connection
- *BYE* the user agent client uses BYE to indicate to the server that it wishes to release the call
- *CANCEL* this request cancels a pending request with the same Call-ID, To, From and CSeq (sequence number only) header field values, but does not affect a completed request or existing calls. (A request is considered completed if the server has returned a final response)
- *INFO* an additional SIP method proposed, as part of the SIP-extensions is INFO method. The intent of the INFO method is to allow for the carrying of session related control information that is generated during a session. INFO method is detailed in RFC 2976 [Don00]. Other SIP extension methods are also being proposed.

Following the Request-Line, after the SIP headers, the request may contain a message body, which is separated from the headers with an empty line. The message body is always a session description and if present the type of Internet media in it is indicated by the Content-Type header field.

The recipient, after receiving and interpreting a request message, responds with a SIP response message, indicating the status of the server, success or failure. The responses can be of different kinds and the type of response is identified by a status code, a 3-digit integer. The first digit defines the class of the response. The other two have no categorization role. The six different classes that are allowed in SIP are here listed with their meaning. These classes can be categorized by provisional and final responses. A provisional response is used by the server to indicate progress, but does not terminate a SIP request. A final response terminates a SIP request.

1xx response codes are provisional responses and 2xx onwards responses are final responses.

SIP applications are not required to understand the meaning of all registered response codes, though it is desirable. However applications must be able to recognize the class of the response and treat any unrecognized response as being the x00 response code of the class.

SIP Mobility With a PSTN network, Local Number Portability (LNP) poses an implementation challenge. However it is a trivial application for SIP services if the user has a domain name, and address such as `max@mustermann.com`. With their own domain name, users can actually have service portability by choosing the service provider, for example when on relocation, to host their service. The caller may always use the same address, phone number or URL, but will be redirected transparently to the network, location or device of choice of the called party. Mobility in an IP environment is classified as:

- *Personal mobility* - different terminals, same personal identity (address)
- *Terminal mobility* - the ability to maintain communications when moving a single end system from one subnet to another
- *Service mobility* - keep same services while mobile

SIP has been chosen for call control for the 3rd generation wireless network by the 3GPP (Third Generation Partnership Project) initiative.

In the era of network convergence, a key challenge for the network operators and service providers is how to ensure interoperability between different communication protocols. SIP has been widely accepted by service providers because it can deliver enhanced services over next-generation networks. SIP supports interoperability with H.323 and ISUP (ISDN User Part) – key protocols from both the IP and SS7 environments and hence gives service providers an advantage to offer new SIP services that can go well beyond VoIP. SIP interoperability has been demonstrated in SIP bake-offs. The purpose of the bake-off is to test for interoperability of SIP implementations, determine the source of incompatibilities, and if the specification is at fault, prepare a *fix* for the SIP draft revision. So far at least six bake-offs have taken place and leading SIP-products vendors have participated. The number of companies joining these bake-offs has increased tremendously since the first SIP bake-off of April 1999.

SIP provides a simple but powerful platform to get its services and features extended. These extensions will help SIP to cope-with the changes of the Internet Telephony industry. This level of flexibility is critical to the rapidly moving VoIP

field. SIP enhancements tend to be for specialized services, such as ISUP interworking, QoS (Quality of Service) negotiation, liveness detection, caller preferences or presence/instant messaging. All of these are backward compatible with the basic protocol, with extensions negotiated if both sides support them. Basic calls will succeed without the extensions.

SIP is a powerful tool for call control and signaling that is gaining tremendous support among service providers and vendors. SIP turns out to be an ideal protocol for providing truly converged applications. This is primarily because it borrows so heavily from other Internet protocols, and in particular, HTTP and SMTP. SIP supports features like MIME (Multipurpose Internet Mail Extensions), URL (Universal Resource Locators) and DNS (Domain Name System), which renders SIP ideal for converged services. SIP supports CPL (Call Processing Language) which enables the users to upload their location information through CPL scripts and then SIP server can take decision based on the CPL script. With the features like mobility (personal, terminal and service) and interoperability, SIP promises to bring a revolution in the internal telephony industry and has surely made a big impact in the network convergence.

2.2.1.3 H.323 vs. SIP

Since H.323 and SIP offer similar services, which solution should be used? Comparisons of these protocols are given in [SR98] and [DF99].

When the complexity of the two protocols is compared, it seems that SIP is far less complex than H.323. The specification of H.323 is more extensive than that of SIP and defines a lot more elements. Furthermore, H.323 uses a binary encoding mechanism for call signaling and control, whereas SIP is text based. This textual format is easy to decode and much easier to debug than a binary representation. A part of the complexity of H.323 stems from the interaction between several components which are not cleanly separated. Also, in H.323 there may be several ways to accomplish a single task and some of the functionality is present in several parts of the protocol.

Considering the extensibility of the protocols, the experience with other protocols like SMTP and HTTP has been used to make SIP very extensible: new features can easily be incorporated into the protocol. H.323 also allows some extensions, but only at predefined places within the protocol. SIP is quite modular which allows its components to be changed quite easily. H.323 on the other hand, is less modular. Since various protocol components usually need to work together to accomplish a task, it will be harder to simply replace one component.

H.323 was originally intended for use on a single LAN. Currently, this restriction is no longer present, but H.323 can have some difficulties in detecting looping messages. SIP can be used over wide area networks without any difficulties, easily detecting

loops when they occur. H.323 also has some difficulties when the conference size keeps increasing. The use of a Multipoint Controller (MC) is a bottleneck for the conference. When the conference size keeps growing, eventually another protocol will have to be used: H.332 (H.323 Extended for loosely-coupled conferences). Since SIP does not have something similar to a MC, it does not suffer from such scalability problems.

Like was mentioned before, the services provided by H.323 and SIP are roughly the same. However, when it comes to capability exchange services, it seems that H.323 has a much richer set of functionality than SIP. Also, H.323 has various conference control services for which SIP has to rely on external protocols. On the other hand, the personal mobility services provided by SIP are more extensive than similar support in H.323.

Finally, current developments in the field of commercial VoIP solutions show that SIP succeeded, i.e. SIP is mainly used except for some exceptions like for example Skype, which is based on a proprietary P2P protocol and is therefore incompatible with standard SIP solutions.

2.2.2 Related VoIP Protocols

In order to provide Voice over IP, various other protocols have to be considered which are briefly discussed selectively.

2.2.2.1 Session Announcement Protocol (SAP)

The SAP protocol is used for advertising multicast conferences and multicast sessions. A SAP announcer periodically multicasts announcement packets to a well known multicast address and port (port number: 9875). The SAP listener listens to the well known SAP address and port and learns of the multicast scopes using the Multicast Scope Zone Announcement Protocol. A SAP announcer is unaware of the presence or absence of SAP listeners. A SAP announcement is multicast with the same scope as the session it is announcing, thus ensuring that the recipients of the announcement can also be potential recipients of the session being advertised. If a session uses addresses in multiple administrative scope ranges, it is necessary for the announcer to send identical copies of the announcement to each administrative scope range. It is alright for multiple announcers to announce a single session, thus ensuring robustness of the protocol.

The intervening time period between announcements is decided such that the total bandwidth used by all the announcements in a single SAP group is less than a preconfigured limit. Each announcer is required to listen to all the announcements in its group in order to determine the total number of sessions being announced in the group. One of the protocol's objectives is to announce the existence of long-lived

wide area multicast sessions and involves a large startup delay before a complete set of announcements is heard by a listener.

SAP Proxy caches can also be deployed to reduce the inherent delays in SAP. A SAP proxy is expected to listen to all SAP groups in its scope and maintain an up to date list of all announced sessions along with the last receipt time of each announcement. SAP also contains mechanisms to ensure the integrity of session announcements, announcement encryption and also to authenticate the origin of an announcement.

2.2.2.2 Session Description Protocol (SDP)

SDP is an IETF specified protocol (RFC 2327) [HJ98] that helps in describing multimedia sessions. It is used for session announcements, session invitation, etc. For example, the SDP payload gets included in the SIP INVITE packet to convey information about the sender to the recipient and vice versa, before participating in the session. This allows media information to be similarly shared between the parties. An SDP payload includes the following information:

- Session name and purpose
- Address and port number
- Start and stop times
- Media information
- Bandwidth requirement
- Contact information

The above information is conveyed in text format. In general, SDP must convey sufficient information to enable a party to join a session and also to announce the resources to be used in a multiparty conference. The media information that SDP sends is: type of media (audio or video), transport protocol (RTP, UDP, etc.) and media format (MPEG video, H.263 video, etc.).

2.2.2.3 Media Gateway Control Protocol (MGCP)

MGCP defines the communication between *Call Agents* (call control elements) and gateways. It is an IETF specification. Call Agents are also called Media Gateway Controllers. It is a control protocol that monitors the events on IP phones and gateways and instructs them to send media to specified addresses. MGCP has evolved from two earlier protocols - the Simple Gateway Control Protocol and the Internet Protocol Device Control.

As per recommendations, the call control intelligence is located outside the gateway in the Call Agents. These Call agents are assumed to have synchronized with each other and they issue coherent commands to the gateways under their control. The issued commands are executed by the gateways in a master/slave manner. MGCP defines the concepts of *Endpoints* and *Connections* to describe and establish voice paths between two participants. Similarly, it has defined *Events* and *Signals* to describe set-up or tear down of sessions. MGCP is intended to be a simple protocol for enabling development of reliable and cheap local access systems. Accordingly, the programming complexity is concentrated into the Call Agent.

2.2.2.4 Real-time Streaming Protocol (RTSP)

IETF has defined the RTSP as RFC 2326 [SRL98] as a client/server protocol that provides control over the delivery of real-time media streams. It is akin to a *VCR-style* remote control for audio and video streams. Functions such as pause, fast forward, reverse and absolute positioning are provided to the user. It also allows the user to choose the RTP-based delivery mechanisms and also a delivery channel such as UDP, multicast UDP and TCP over IP. The RTSP functions between the media servers and its clients and establishes and controls the connecting audio and video media streams. The media server provides playback and recording of the media streams to the client, whereas the client can request such services from the media server.

RTSP is an application level protocol similar to HTTP but is meant for audio and video. It requires maintenance of states and allows bi-directional requests between client and server. Further, RTSP requests are used by the client to retrieve media, or invite a server to a conference or add a new media to an existing presentation at the server.

2.2.3 Wired Networks

As the IP network was primarily designed to carry data, it does not provide real-time guarantees but only provides best effort service, which is inadequate for voice communication. Upper layer protocols were designed to provide such guarantees. Further, as there are several vendors in the market implementing these protocols, conformance to standards and interoperability issues have become important. The major issues governing transfer of a voice stream over the Internet or using Internet protocols are listed below.

Bandwidth Requirement In the analog world, the voice transmission frequency spectrum requirement is 0-3.4 kHz in the base band, and is nominally called a 4 kHz voice channel for convenience. For digital telecommunication, the signal is sampled

at twice the rate. The minimum-sampling rate required is thus 8 kHz. If each sample contains 8 bits, the digital bandwidth required works out to be 64 Kbps. Telephone Company quality voice requires sampling at 8 kHz. The bandwidth then depends on the level of quantization. With linear quantization at 8 bps or at 16 bps, the bandwidth is either 64 Kbps or 128 Kbps. Further, the quantization (e.g. PCM) is modified by using an A-law or μ-law companding curve. In order to communicate telco-grade voice (or similarly, other real-time applications such as moving video) two different approaches can be attempted. To transmit information of the highest quality over unrestricted bandwidth or to reduce the bandwidth required for transmitting information (voice) of a given quality. Stated differently, decisions are required regarding what information should be transmitted and how it should be transmitted. Compression and decompression (codec) of digital signals is a means of reducing the required bandwidth or transmission bit rate. Certain source data are highly redundant, particularly digitized images such as video and facsimile. If, for example, a digital signal contains a string of zeros, it will be economical to transmit a code indicating that a string of zero follows along with the length of the string. Many different algorithms for compression and decompression of digital codes have been constructed. Pulse code modulation (PCM) and adaptive differential PCM (ADPCM) are examples of *waveform* codec techniques. Waveform codecs are compression techniques that exploit the redundant characteristics of the waveform itself. In addition to waveform codecs, there are source codecs that compress speech by sending only simplified parametric information about voice transmission; these codecs require less bandwidth. Source codecs include linear predictive coding (LPC), code-excited linear prediction (CELP) and multipulse-multilevel quantization (MP-MLQ). Coding techniques for telephony and voice packet are standardized by the ITU-T in its G-series recommendations - also see 4.2.

Delay A very important design consideration in implementing voice communications networks is minimizing one-way, end-to-end delay. Voice traffic is real-time traffic and if there is too long a delay in voice packet delivery, speech will be unrecognizable. An acceptable delay is less than 200 milliseconds. Delay is inherent in voice networking and is caused by a number of different factors.

There are basically two kinds of delay inherent in today's telephony networks:

- Propagation delay - caused by the characteristics of the speed of light traveling via a fiber-optic-based or copper-based medium of the underlying network
- Handling delay (also called serialization delay) - caused by the devices that handle voice information and have a significant impact on voice quality in a packet network. This delay includes the time it takes to generate a voice packet. DSPs may take 5 ms to 20 ms to generate a frame and usually one or

more frames are placed in a voice packet. Another component of this delay is the time taken to move the packet to the output queue. Some devices expedite this process by determining packet destination and getting the packet to the output queue quickly. The actual delay at the output queue, in terms of time spent in the queue before being serviced, is yet another component of this handling delay and is normally around 10 ms. A codec-induced delay is considered a handling delay

Serialization Delay Serialization delay is the amount of time a router takes to place a packet on a wire for transmission. Fragmentation helps to eliminate serialization delay, but fragmentation, such as FRF.12, does not help without a queuing mechanism in place. For example, if a 1000-byte packet enters a router's queue and is fragmented into ten 100-byte packets, without a queuing mechanism in place, a router will still send all 1000-bytes before it starts to send another packet. Conversely, if there is a queuing mechanism in place, but no fragmentation, voice traffic can still fail. If a router receives a 1000-byte packet in its queue and begins sending this packet in an instant before it receives a voice packet, the voice packet will have to wait until all 1000 bytes are sent across the wire, before entering the queue, because once a router starts sending a packet, it will continue to do so until the full packet is processed. Therefore, it is essential that there is a method for a router to break large data packets into smaller ones, and a queuing strategy in place to help voice packets jump to the front of a queue ahead of data packets for transmission.

End-to-End Delay / Jitter End-to-end delay depends on the end-to-end signal paths/data paths, the codec, and the payload size of the packets. Jitter is variation in the delay of arrivals of voice packets at the receiver. This causes a discontinuity of the voice stream. It is usually compensated for by using a play-out buffer for playing out the voice smoothly. Play-out control can be exercised both in adaptive or nonadaptive play-out delay mode.

Echo Cancellation Echo is hearing your own voice in the telephone receiver while you are talking. When timed properly, echo is reassuring to the speaker. If the echo exceeds approximately 25ms, it can be distracting and cause breaks in the conversation. In a traditional telephony network, echo is normally caused by a mismatch in impedance from the four-wire network switch conversion to the two-wire local loop and is controlled by echo cancellers. In voice over packet-based networks or VoIP, echo cancellers are built into the low bit-rate codecs and are operated on each DSP. Echo cancellers are limited by design by the total amount of time they will wait for the reflected speech to be received, which is known as an echo trail. The echo trail is normally 32 ms.

Reliability Traditional data communication strives to provide reliable end-to-end communication between two peers. They use checksum and sequence numbering for error control and some form of negative acknowledgement with a packet retransmission handshake for error recovery. The negative acknowledgement with subsequent re-transmission handshake adds more than a round trip delay to transmission. For time-critical data, the retransmitted message/packet might therefore be entirely useless. Thus, VoIP networks should leave the proper error control and error recovery scheme to higher communication layers. They can thus provide the level of reliability required, taking into account the impact of the delay characteristics. Therefore, UDP is the transport level protocol of choice for voice and like communications. Reliability is built into higher layers. Audio data is delay-sensitive and requires the transmitted voice packets to reach the destination with minimum delay and minimum delay jitter. Although TCP/IP provides reliable connection, it is at the cost of packet delay or higher network latency. On the other hand, UDP is faster compared to TCP. However, as packet sequencing and some degree of reliability are required over UDP/IP, RTP over UDP/IP is usually used for voice and video communication.

Interoperability In a public network environment, in order for products from different vendors to interoperate with each other, they need to conform to standards. These standards are being devised by the ITU-T and the IETF. H.323 from ITU-T is by far the more popular standard. However, SIP/MGCP standards from IETF are rapidly gaining more acceptance as relatively light weight and easily scalable protocols.

Security On the Internet, since anybody can capture packets meant for someone else, security of voice communication becomes an important issue. Some measure of security can be provided by using encryption and tunneling. Usually, the common tunneling protocol used is Layer 2 tunneling protocol, and the common encryption mechanism used is Secure Sockets Layer (SSL).

Integration with PSTN and ISDN IP Telephony needs to co-exist with traditional PSTN for still some more time. It means that both PSTN and IP telephony networks should appear as a single network to users. This is achieved through the use of gateways between the Internet on the one hand and PSTN or ISDN on the other.

Scalability As succeeding VoIP products strive to provide Telco-grade voice quality over IP as is true for PSTN, but at a progressively lower cost, there is a potential for high growth rates in VoIP systems. In such a scenario, it is essential that these systems be flexible enough to grow into large user markets.

2.2.4 Wireless Networks

Wireless technology keeps changing the communications scenario. Wireless local area networks (WLANs), in particular, are being enthusiastically adopted by users worldwide, shaping a new world where tetherless access will be possible not only in homes and offices, but also in an increasing number of previously unconnected places, like shopping malls, libraries, trains and other means of mass transportation, even private motor vehicles. As soon as seamless integration with wide-area coverage provided by 2.5G/3G cellular wireless infrastructures is reached, wireless access will likely become the most common form of network access for an increasing number of users.

The IEEE 802.11 WLAN standard, based on the definition of the medium access control (MAC) protocol and the physical layer (PHY) specifications, became available in 1999 and since then has emerged as the most successful and most widely deployed WLAN standard. So far, the main usage of Wireless LANs has been limited to Internet based services like Web browsing, e-mail, and file transfers. However, as already happened in the traditional wired LANs, a strong interest is quickly emerging towards multimedia applications over WLANs, and interactive voice communications are appearing more and more as the natural evolution of cordless telephony. Not only such technology would have all the advantages of IP-based communications, including a single infrastructure for both data and voice, but it would also deliver significant cost savings, and possibly better voice quality, with respect to cellular telephony.

Several challenges, however, need to be addressed to make WLAN telephony as successful as cellular and wired telephony. Not only the available bandwidth for WLANs is significantly below that of their wired counterparts, but wireless links are also strongly time-varying and may have high error rates. Other issues are specific of 802.11 WLANs, including the MAC layer effects on performance, the consequences of interfering data traffic, and the best configurations for both Access-Point-based and ad-hoc 802.11 networks.

Previous research evaluated the performance of interactive voice traffic over Wireless LANs [CWKS97, VCM01], mainly by means of statistical analysis of throughput and packet losses to assess the number of supported voice conversations. Advanced Quality of Service techniques are also yet to be fully investigated.

Voice over IP over wireless packet networks is becoming increasingly attractive. In particular, the widespread adoption of WLAN technology is creating the basis for the introduction of cordless packet telephony in offices, homes, hospitals, etc.

Two-way conversational applications, however, are characterized by stringent requirements on the end-to-end delay. The upper limit for one-way delay is set to only 150 ms, according to the guidelines of ITU-T Recommendation G.114. Moreover, packet losses should be kept below 1% to prevent significant perceptual degradation.

The WLAN environment is quite challenging on two counts: the wireless link is inherently noisy, due to fading and interference; the contention-based medium access control (MAC) layer and the retransmission-based-error-control scheme may introduce strong delays. Efficient WLAN-based cordless telephony systems must thus rely on careful design of advanced speech transmission solutions - this is one of the main parts of this thesis.

2.2.4.1 Security

The marriage of voice, IP, and wireless offers many benefits, but there's a dark side of this union. A formidable triple of security threats to users and IPT operators (private and public) arose. The customer just see *IP telephony* - but the attacker sees and thinks, *a new phone service I can exploit as I have POTS or cellular*; *a new set of applications and protocols I can probe for specification and implementation flaws*; and *more systems I can try to exploit using traditional (TCP/IP) protocol exploits.*

Many of the motivations to attack IPT users are the same as telephone service attacks: to benefit financially, via toll fraud, identity and information theft, and to gain notoriety, by disrupting service and inconveniencing users. Such attacks are similar to attacks we have seen on cellular and landline phones for years. Others are all too familiar attacks we see against networked computers. IPT phones and computers running IPT software (IPT softphones) are more computer than phone. They have operating and file systems, use Internet protocols, and run data and management as well as voice applications. They are vulnerable to unauthorized access, privilege escalation and *system* misuse, viruses and worms, and all the *classic* denial of service attacks that exploit network protocols (TCP, IP, ICMP, ARP). Many of these attacks take advantage of the IP telephony protocols that were discussed above.

Current implementations of IP telephony call signaling (SIP), voice message delivery (RTP), and control protocols (RTCP) do not provide adequate call party authentication or end-to-end integrity and confidentiality measures on call signaling and call data (e.g., encoded speech). Until these security features are implemented and put into service, attackers have many exploit opportunities which can be categorized as follows:

- *Eavesdropping* on call signaling packets exchanged between SIP servers and SIP phones may provide attackers with IPT user identities, PINs, and SIP phone numbers. An attacker who gains possession of an IPT user's account information and password can alter any user-settable service profile information: for example, in conjunction with a stolen account, the attacker might want to try to change a user's calling plan, alter a voice mail message, or change a call forwarding number. Attackers can also invade IPT caller pri-

vacy and eavesdrop on an IPT call by capturing packets associated with a voice conversation, then later replaying the conversations to obtain sensitive business or personal information.

- Several *identity theft*, *identity fraud*, and *toll fraud attacks* are possible. An attacker can connect a rogue IPT phone to a network and use a stolen or guessed user account and password (PIN) to place phone calls at the victim's expense, much as he would a stolen cell phone or calling card. Attackers can also hijack IPT conversations using techniques similar to those used to hijack TCP connections. The attacker can send a SIP control packet to an endpoint, directing an in-progress to a different device (typically under the attacker's control). The caller is misled into communicating with someone other than the intended party (e.g., the attacker, masquerading as a party to this call).

- *IP telephony call integrity* can be compromised if message or packet authentication is not used (the common situation). For example, attackers can corrupt conversations by intercepting RTP packets, altering the contents and forwarding the modified packet to the original recipient. Other, similar man-in-the-middle attacks are also possible when message integrity is not provided over wireless LANs. For example, an attacker can inject speech, noise or delay (silent gaps) into active call from a rogue access point.

- Denial of service attacks can be launched against IPT's signaling protocol, SIP. Similar to control packet attacks against TCP (SYN, RST), such attacks flood an IPT device with call invitation and registration requests in an attempt to exhaust its resources, to force disconnects, or to falsely signal a busy condition. Call data flooding attacks can be directed at RTP, the protocol that carries digitized voice. Like UDP flooding, attacks of this kind try to exhaust resources by bombarding an IPT device with large volumes of call data. IPT implementations may also be vulnerable to DoS attacks against TCP, UDP, ICMP, and underlying (e.g., wireless) networks. Operating systems and network protocol (TCP/IP stack) on which IPT applications and hardware devices are built may be susceptible to implementation-specific DoS attacks as well. An attacker can instigate a particularly spiteful service disruption attack by forging and sending a password (PIN) change control packet to an IPT phone. The authorized user, unaware of the change, will be unable to place calls.

IPT operators should anticipate the same kinds of attacks that we have seen on cellular and landline phones. These include toll fraud, identity and information theft, and service disruption. They should also anticipate attacks against computer systems that comprise IPT operations systems and infrastructure. Call managers,

IP telephony switches, routers, and IPT-to-PSTN gateways must be protected from unauthorized access, privilege escalation and system misuse, viruses and worms, and denial of service attacks. IPT operators who offer online payment and service plan management must defend against attackers seeking to compromise accounts and databases using a variety of web attacks. These attacks are not IPT specific but a common problem for all telephony carriers.

IPT operators must contend with concerns that public IPT infrastructures will be no less a target for politically motivated attackers (terrorists) than the PSTN, and related concerns that operating IP telephony *securely* encumbers electronic surveillance by law enforcement agencies (CALEA). When VPN tunnels are used to protect (SIP) call control packets and voice/media streams, law enforcement must have access to encryption and authentication keys or they will be unable to conduct lawful intercept/wiretap. The solutions are technically complex, and include methods for disclosing keys to law enforcement that work irrespective of the type of VPN used to protect IPT: IPsec, SSH, SRTP, SSL or proprietary encrypted tunnels. Moreover, for every possible VPN protocol, intercept equipment must be able to deal with a variety of configuration options: IPsec alone has a tunnel and transport mode, multiple ways to identify and authenticate IKE endpoints, and multiple bulk encryption and integrity methods.

IPT operators must also deal with concerns that it's difficult to support Emergency Telecommunications Services in the face of disaster events. Some IPT operators (e.g., `Calleveryone.com`) expressly state they do not provide E911 and are not a lifeline service [Cal05]. Some recommend that subscribers have cellular service if IPT is their only land line phone service. Vonage, Broadvoice, and others support a *nearly 911* service they expressly state is not the same as E911. The subscriber must activate 911-type service; when used, the 911 call is routed to the Public Safety Answering Point (PSAP) or *local emergency service personnel designated for the address that you listed at the time of activation*. Packet8 claims to be the first IPT carrier to support true E911, where the 911 call is labeled and routed as emergency traffic and E911 caller information accompanies the call request.

ETS (Emergency Telecommunication Service) requirements were investigated, which include support of SS7-like mechanisms to distinguish emergency from normal calls and methods to assure such labels can be conveyed across hybrid (packet and PSTN) networks [CA04]. Measures are also needed to assure sufficient accounting information is available to detect abuse. Until these and similar concerns are addressed, IP telephony may only be suitable for second line service or limited (e.g., intra-organizational) private network deployments.

Many vulnerabilities associated IPT's call signaling (SIP), voice message delivery (RTP), and control protocols (RTCP) can be used against operators as well as their subscribers. Like IPT phones, IPT call servers can be flooded with unauthenticated call control packets. Attackers can launch the full spectrum of denial

of service attacks against IPT operators' underlying networks and Internet protocols. IPT applications such as voice mail and short messaging services can be targets of message flooding attacks. Such attacks may prevent legitimate attempts to leave a subscriber a message. Floods, subscriber impersonation, and rogue IPT phone connections create nightmarish customer care scenarios for operators. Resolving disputes with customers who are victims of subscriber impersonation, call hijacking, and rogue phone connections, or billed for thousands of unsolicited flood messages is a resource and revenue drain.

After briefly getting into details about IPT security problems the next paragraph summarizes currently available security mechanisms which were grasped by the wireless standards.

Initial wireless deployments used WEP (Wired Equivalent Privacy) as their security solution, but that standard proved irreparably broken. Many organizations have used their existing VPN infrastructure for remote access to facilitate internal Layer 3 security on their wireless networks. Unfortunately, wireless handsets have limited processing power and aren't based on popular operating systems or processors.

Unlike many other IEEE wireless standards, 802.11i is finished, and the Wi-Fi Alliance offers both WPA and WPA2 certification, which map to the most relevant elements of 11i. Vocera's badges support Cisco's proprietary LEAP (Lightweight Extensible Authentication Protocol, also shown to be broken if weak passwords are used) and WPA-PSK (WPA with Preshared Keys). Because preshared keys must be entered manually, they don't scale well. Even if the PSK is unique for each user, it remains tied to the device. Another option, WPA-Enterprise, relies on the handsets' still-primitive interface for user-based authentication. Unless certificates are used, as in EAP-TLS (EAP with Transport Layer Security), the device must hold the user credentials, which is fatal, if the phone gets lost.

Even great efforts were made, many security related difficulties are not solved yet.

2.2.5 Scalability and Reliability

2.2.5.1 Scalability in VoWLAN Networks

If VoWLAN is to replace legacy TDM voice networks, it must be able to scale to the same level as the network it replaces. Unlike a legacy wired network, it must take into consideration not just switching capacity, but bandwidth and spectrum allocation as well. This section outlines scalability concerns that network planners should take into consideration when planning a VoWLAN application (see figure 2.3).

The IEEE specification for 802.11b calls for a maximum of 11 Mbps of bandwidth. The G.711 standard for voice traffic calls for 64 Kbps. Very simply put, this would suggest that more than 170 simultaneous G.711 conversations could take place on a

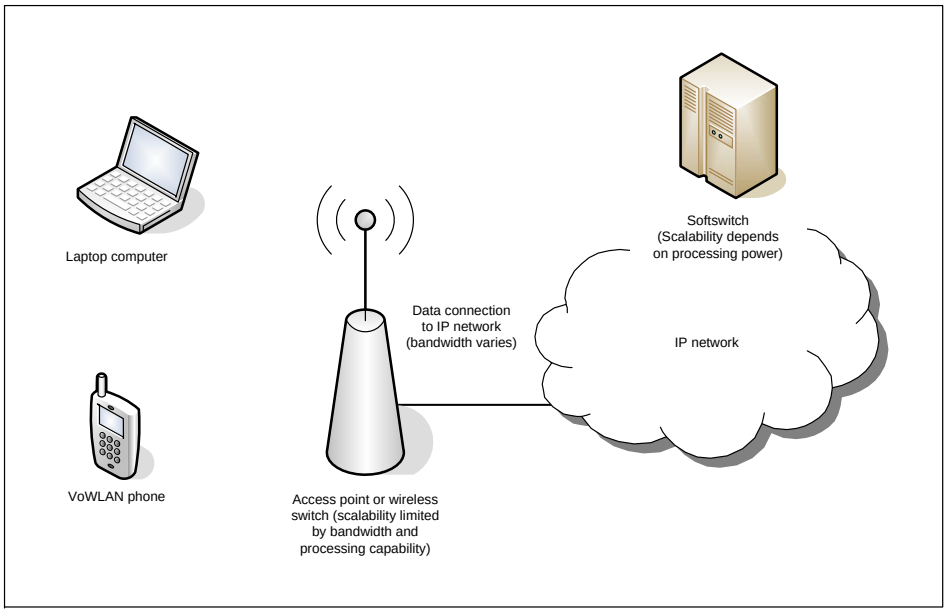


Figure 2.3: Considerations in the scalability of wireless VoIP networks

single access point. However, VoIP adds considerable overhead per conversation, so the actual amount of bandwidth used in an uncompressed VoIP conversation is well in excess of 64 Kbps, depending on what information is to be included in the headers of the voice packets (could exceed 10 Kbps). Also, the use of 802.11b does not guarantee 11 Mbps of bandwidth. Rather, in the case of hotspots, for example, the source of the bandwidth is a T1 (1.54-Mbps) system ordered from the local telephone company. This would suggest more than 20 simultaneous conversations would be possible based on a 64-Kbps stream for the VoIP conversation before figuring in VoIP overhead. The use of other variants of 802.11, namely, 802.11a, which has a maximum bandwidth of 54 Mbps, would suggest, before figuring in VoIP overhead, the possibility of more than 800 simultaneous conversations. Compressing the voice stream to 8 Kbps (G.729) could increase the number of simultaneous conversations per AP.

The bandwidth of an AP, which is one determinant of the maximum number of simultaneous conversations, is not infinite through space. Rather, bandwidth diminishes with distance from the access point. Factors such as trees, buildings, and weather can degrade the penetration capabilities of 802.11 through space. This is called path loss and is described in section 3.4.1. Figure 2.4 illustrates path loss or the degradation of the data link rate, that is, the bandwidth of 802.11 variants through space.

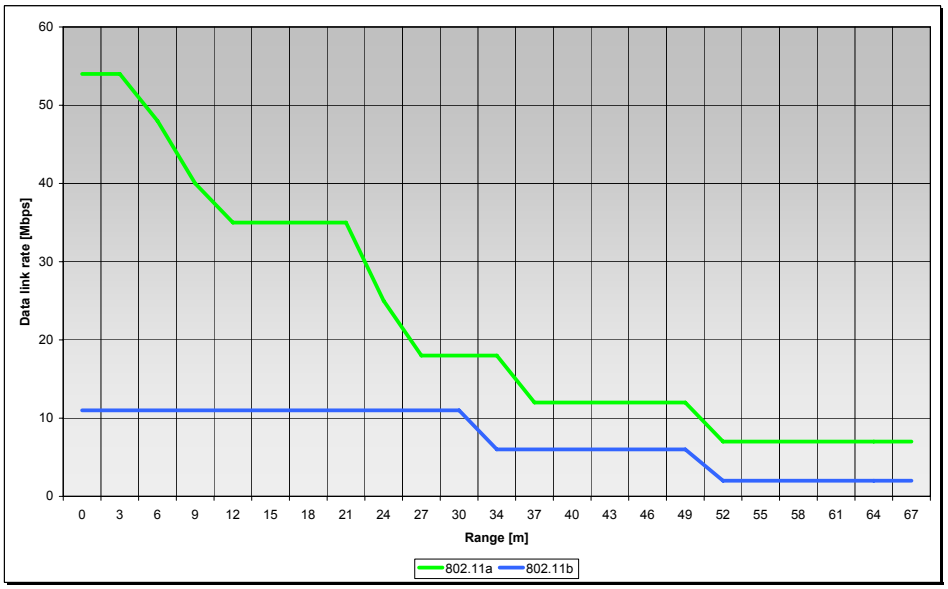


Figure 2.4: Data link rate vs. Indoor range
(the greater the distance from the access point, the greater the degradation in bandwidth, which limits the total number of simultaneous calls per access point)

Why Frequency Bands Are Important The 802.11 technologies can be deployed on four unlicensed frequency bands in two bands called ISM and U-NII. The 2.4-GHz ISM band has an inherently stronger signal with a longer range and can travel through walls better than the 5-GHz U-NII bands. However, the U-NII band allows more users to be on the same channel simultaneously. The 2.4-GHz ISM band has a maximum of three non-overlapping 22-MHz channels, whereas the 5-GHz band has four non-overlapping 20-MHz channels in each of the U-NII bands.

The most difficult part of calculating a link budget is the path loss. Outdoors, the free-space loss (fspl) is well understood. The path loss equation [Jor86] for outdoors can be expressed as follows:

$$\text{fspl} = 20\,log(d) + 20\,log(f) + 36.6dB$$

with distance d [m] and frequency f [MHz]. Therefore at 2.4 GHz, the formula simplifies to

$$\text{fspl} = 20\,log(d) + 40dB$$

This formula holds true as long as one can see along the LOS from the receiver and the transmitter and have a sufficient amount of area around that path called the Fresnel zone. For indoors, this formula is more complicated and depends on factors

such as building materials, furniture, and occupants. At 2.4 GHz, one estimate follows this formula:

$$\text{ipl} = 55db + 0.3dB/d$$

at 5.7 GHz, the formula looks like this:

$$\text{ipl} = 63db + 0.3dB/d$$

with distance d [m] and indoor path loss ipl.

A network planner calculating the capacity of a VoWLAN network should take into account path loss by using the formulas given earlier. Given the direct relationship between bandwidth and the maximum number of simultaneous conversations, a network planner can determine the maximum number of VoWLAN users on a given AP.

Another determinant of scalability for a VoWLAN network is frequency reuse in a given service area. If a service provider or enterprise operator were to use the same frequency over a wide area, there would be inevitable interference, which would limit the maximum number of users of a given VoWLAN network. Determining frequency reuse is an important factor in determining the capacity of a VoWLAN network. Cell phone service providers have used reuse techniques for years.

The 2.4-GHz band has eleven 22-MHz-wide channels defined starting at 2.412 GHz and going for every 5 MHz through 2.462 GHz. Three non-overlapping channels are available – 1, 6, and 11. These non-overlapping channels can be used in a 3-to-1 reuse pattern.

The operating channel center frequencies are defined at every integral multiple of 5 MHz above 5 GHz. The valid operating channel numbers are 36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, and 161. The lower and middle 802.11a subbands accommodate eight channels in a total bandwidth of 200 MHz. The upper 802.11a band accommodates four channels in a 100-MHz bandwidth. The centers of the outermost channels are 30 MHz from the bands' edges for the lower and middle 802.11a bands and 20 MHz for the upper U-NII band. Point-to-point links operate on the other four channels: 149, 153, 157, and 161. This allows four channels to be used in the same area. The 802.11a APs and client adapter cards operate on eight channels: 36, 40, 44, 48, 52, 56, 60, and 64. This allows two 4-to-1 reuse patterns to be used. By using both the low-frequency and mid-frequency ranges together, we can take advantage of a 7-to-1 reuse pattern with a spare. The spare can be added for a fill to extend coverage or to add capacity in areas such as conference rooms where more capacity is needed.

Most of the industry is now focused on enterprise applications where VoWLAN applications allow the employee to roam the premises accessing the wireless LAN with an 802.11 phone, laptop, or PDA. The question then becomes one of how many

wireless VoIP users can access a given access point at a given time? The limitations to scalability are twofold: the bandwidth available and the ability of the access point to process simultaneous sessions. As a result of these two factors, vendors of VoWLAN platforms include APs that are specifically designed to handle voice separate from APs that are part of the data network. Some of these voice-specific APs can process about 10 to 12 simultaneous conversations. Once it appears that there is a demand for more than 12 conversations per AP, then the enterprise can add another AP to cover that area (break room, conference room, and so on). In 2004, high-capacity, phased-array access points were coming on the market that allowed hundreds of simultaneous conversations. However, these products have yet to be rigorously tested in the marketplace.

This section explored the question of scalability of wireless VoIP applications. Potential bottlenecks exist with regard to bandwidth, spectrum allocation and the capacity of the AP or wireless switch to process multiple sessions.

2.2.5.2 VoWLAN Reliability

Availability is often expressed numerically as a percentage of uninterrupted productive time containing from one to five nines. For instance, 99% availability, or "two 9s", equates to a certain amount of availability versus downtime, as does 99.9% (three 9s), and so on. The downtime calculations shown for each of the five 9s (table 2.1) are based on 24-hour, year-round operation.

The terms reliability and availability are often used interchangeably but they are two distinct measures of quality. Reliability refers to component failure rates measured over time, usually a year. Common reliability measures of components are annual failure rate, failures in time, mean time between failure, mean time to repair, and single point of failure (SPOF).

Availability measures reliability and indicates system *uptime* from an operation perspective. System availability is a function of aggregate component reliability, thus availability is likewise measured in terms of time. Availability of a hardware/software module can be obtained by the formula given below for calculating availability:

$$a = \frac{\text{mtbf}}{\text{mtbf} + \text{mttr}}$$

where a is availability, mtbf is *Mean time between failures* and finally mttr is *Mean time to repair*. We can also calculate unavailability:

$$u = \frac{\text{mttr}}{\text{mtbf} + \text{mttr}}$$

where u is unavailability.

Availability	Downtime
90% (one 9)	36.5 days per year
99% (two 9s)	3.65 days per year
99.9% (three 9s)	8.76 hours per year
99.99% (four 9s)	52.55 minutes per year
99.999% (five 9s)	5.25 minutes per year

Table 2.1: Availability and Downtime - the "five 9s"

The annual failure rate afr, using 8.760 hours per year, is calculated as follows:

$$\text{afr} = \frac{8760}{\text{mtbf}}$$

The greatest requirement for availability is for network elements (a circuit switch, for example), which are generally required to provide 99.999% availability or 0.001% unavailability. Such a system is termed highly available (HA).

A common perception is that a wireless network cannot be as reliable as a wired network because the airwaves cannot be as *solid* as a wire. In reality, a wireless form of access in the enterprise or *last mile* offers more forms of backup or redundancy than that found in wired networks [Ort04].

Manufacturers have been designing redundancy into their products for years in the form of redundant power supplies, multiple processors, segmented memory, and redundant disks. A VoWLAN network can incorporate redundancy in the form of multiple channels to back up those channels that fail or become congested. HA is enhanced when each component is replicated in a system. This is called redundancy. If one unit fails, its replicated unit takes over. Redundant configurations are expressed by the notation m:n, where m represents the number of standby unit(s) and n represents the number of active unit(s) supported by the standby unit(s). A typical configuration is 1:1 where there is one active unit for every active unit or 1:6 where there is one standby unit for six active units. Usually, the smaller the n, the greater the protection and cost. Given the highly reliable nature of today's components, a carrier may determine that configurations greater than 1:1 provide sufficient availability. Class 4/5 switches are more likely to use a 1:1 redundancy model because the effect of a failure is more expensive. Moore's law, which states that computing power doubles while computing cost halves every 18 months, has the effect of making redundancy less expensive as time goes by.

A VoWLAN network can deploy redundant access points to cover for access points that fail. Network planners can also plan for overlapping cells of access point coverage. In this way, when one AP becomes inoperable, another AP whose cell covers that of the failed AP can cover those subscribers served by the failed AP. Given the declining price of access points, it is becoming increasingly cheaper to provide high levels of reliability by simply building in redundancy in a VoWLAN network with

redundant access points.

2.3 Summary

VoIP has enjoyed a significant amount of hype in the marketplace. It was initially viewed as a way to get free phone calls over the Internet and has evolved to being viewed as the technology that will replace the legacy PSTN. There have been literally hundreds of companies who have entered the market, the vast majority of which have failed. As with any new technology, there is a certain time required to grow the market and the growth of the VoIP market has been much slower than anticipated. Even so, VoIP is real, it works, and companies that have been able to *hang in there* are starting to reap the reward. Literally hundreds of thousands of end users and a very large number of enterprise customers are now using VoIP as their primary phone service. Also, while many people do not know, a very large percentage of international phone calls going over IP VoIP networks today [Tel04].

As VoIP is established, VoWLAN (Voice over WLAN) - a method of sending voice information in digital form over wireless broadband networks needs further improvement. Major barriers to VoWLAN include inconsistent voice performance and the need for quality of service (QoS); slow and unreliable encryption and authentication; and the proprietary nature of current products. Considered wireless network technologies will be presented in the following chapter.

3 Wireless Networks

3.1 Introduction

Wireless networks utilize radio waves and/or microwaves to maintain communication channels between instances. Wireless networking is a more modern alternative to wired networking that relies on copper and/or fiber optic cabling between network devices. A wireless network offers advantages and disadvantages compared to a wired network. Advantages of wireless include mobility and elimination of unsightly cables. Disadvantages of wireless include the potential for radio interference due to weather, other wireless devices, or obstructions like walls. Wireless networks often extend an existing wired infrastructure. In this chapter indispensable wireless technologies for IP-Telephony and how they have to interplay in order to fulfill specific requirements will be presented.

3.2 Wireless Networks and Future Developments

Wireless Local Area Networks (WLANs) are proliferating throughout the world at an ever increasing pace, evolving from a private means of extending LAN access on company premises to home-based networks, community networks, and now public free and fee-based networks. Due to the growing presence of WLANs, some are calling it a *disruptive* technology that may threaten the status of established players, especially given mobile network operators' moves to establish their own forms of wireless broadband data access [CHSSP03]. The low cost of WLAN hardware and the ease of installation makes it available to individuals and small venue owners that otherwise would not consider implementing network access throughout their homes, offices or shops. Hence, wireless Internet access is promising for various businesses but also for VoIP.

VoIP over wireless packet networks is becoming increasingly attractive. In particular, the widespread adoption of WLAN technology is creating the basis for the the introduction of cordless packet telephony in offices, homes, hospitals, etc. Two-way conversational applications, however, are characterized by stringent requirements on the end-to-end delay. The upper limit for one-way delay is set to only 150 ms, according to the guidelines of ITU-T Recommendation G.114 [Sta03a]. Moreover, packet losses should be kept below 1% to prevent significant perceptual degradation.

The WLAN environment is quite challenging on to counts: the wireless link is inherently noisy, due to fading and interference; the contention-based medium access control (MAC) layer and the retransmission-based error control scheme may introduce strong delays. Efficient WLAN-based cordless telephony systems must thus rely on careful design of advanced speech transmission solutions.

3.2.1 IEEE 802.11 – Wi-Fi

The 802.11 standard provides for two radio-frequency (RF) variations (as opposed to infrared) of the physical layer: direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS). Both of these were designed to comply with FCC regulations (FCC 15.247) for operation in the 2.4-GHz band, which is an unlicensed spectrum. 802.11b uses DSSS.

DSSS systems use technology similar to that of Global Positioning System (GPS) satellites and some types of cell phones. Each information bit is combined with a longer pseudo-random numerical (PN) in the transmission process. The result is a high-speed digital stream, which is then modulated onto a carrier frequency using differential phase-shift keying (DPSK). DSSS works by taking a data stream of zeros and ones and modulating it with a second pattern, the chipping sequence. The sequence is also known as the Barker code, which is an 11-bit sequence (10110111000). The chipping or spreading code is used to generate a redundant bit pattern to be transmitted, and the resulting signal appears as wideband noise to the unintended receiver. One of the advantages of using spreading codes is that even if one or more of the bits in the chip are lost during transmission, statistical techniques embedded in the radio can recover the original data without the need for retransmission. The ratio between the data and width of the spreading code is called processing gain. It is 16 times the width of the spreading code and increases the number of possible patterns to 64k (2^{16}), thus reducing the chances of cracking the transmission. The DSSS signaling technique divides the 2.4-GHz band into fourteen 22-MHz channels, of which 11 adjacent channels overlap partially and the remaining three do not overlap. Data are sent across one of these 22-MHz channels without hopping to other channels, causing noise on the given channel. To reduce the number of retransmissions and noise, chipping is used to convert each bit of user data into a series of redundant bit patterns called chips. The inherent redundancy of each chip, combined with spreading the signal across the 22-MHz channel, provides the error checking and correction functionality to recover the data. Spread spectrum products are often interoperable because many are based on the IEEE 802.11 standard for wireless networks. DSSS is used primarily in interbuilding LANs, because its properties are fast and far reaching.

At the receiver, a matched filter correlator is used to remove the PN sequence and recover the original data stream. At a data rate of 11 Mbps, DSSS receivers use

different PN codes and a bank of correlators to recover the transmitted data stream. The high rate modulation method is called complementary code keying (CCK). The PN sequence spreads the transmitted bandwidth of the resulting signal (hence, the term spread spectrum) and reduces peak power. Total power remains unchanged. On receipt, the signal is correlated with the same PN sequence to reject narrowband interference and recover the original binary data. Regardless of whether the data rate is 1, 2, or 5.5 of 11 Mbps, the channel bandwidth is about 20 MHz for DSSS systems.

One of the basic technologies underlying the IEEE 802.11 series of standards is spread spectrum radio. The fundamental concept of spread spectrum radio is that it uses a wider frequency bandwidth than that needed by the information that is transmitted. Using extra bandwidth would seem to be wasteful, but it actually results in several benefits, including reduced vulnerability to jamming, less susceptibility to interference, and coexistence with narrowband transmissions. There are several spread spectrum techniques including time hopping, frequency modulation, FHSS, DSSS, and hybrids of these. FHSS and DSSS are not modulation techniques, but simply methods of distributing a radio signal across bandwidth. In addition to spreading the signal across a frequency band, spread spectrum systems modulate the signal. Modulation is the variation of a radio signal to convey information. The base signal is called the carrier. The variation may be based on the strength (amplitude modulation), frequency, or phase (frequency offset) of the signal.

The modulation technique directly affects the data rate. Higher data rate modulations are generally more complex and expensive to implement. Modulations resulting in higher data rates pack more information in the same bandwidth. Small disruptions in the signal cause the degradation of more data. This means that the signal must have a higher signal-to-noise ratio (SNR) at the receiver to be effectively processed. Because a radio signal is stronger the closer it is to the source, the SNR decreases with distance. This is why higher speed systems have less range. Examples of modulation techniques used in the IEEE 802.11 series of specifications are binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), Gaussian frequency-shift keying (GFSK), and CCK.

In 1997 the Institute of Electrical and Electronics Engineers (IEEE) adopted IEEE Standard 802.11-1997, the first WLAN standard. This standard defines the media access control (MAC) and physical (PHY) layers for a LAN with wireless connectivity. It addresses local-area networking, in which the connected devices communicate over the air to other devices that are within proximity to each other. The Wireless Ethernet Compatibility Alliance (WECA) industry group certifies its members' equipment as conforming to the 802.11b standard and allows compliant hardware to be certified as Wi-Fi compatible. This is an attempt at a guarantee of intercompatibility between hundreds of vendors and thousands of devices. Table 3.1 lists the variants of 802.11 and provides an overview of the relationship between

802.11 Variant	Description
802.11a	Created a standard for WLAN operations in the 5-GHz band, with data rates of up to 54 Mbps. Published in 1999. Products based on this standard were released in 2003.
802.11b	Created a standard (also known as Wi-Fi) for WLAN operations in the 2.4-GHz band, with data rates of up to 11 Mbps. Published in 1999. Products based on 802.11b include public space Internet kiosks, WLAN services such as Wayport, and wireless home networking products such as the Macintosh AirPort.
802.11d	Publishing definitions and requirements to allow the 802.11 standard to operate in countries not currently served by the standard.
802.11e	Attempting to enhance the 802.11 MAC to increase the quality of service possible. Improvement in capabilities and efficiency are planned to allow applications such as voice, video, or audio transport over 802.11 wireless networks.
802.11f	Developing recommended practices for implementing the 802.11 concepts of access points and distribution systems. The purpose is to increase compatibility between AP devices from different vendors.
802.11g	Developing a higher speed PHY extension to the 802.11b standard, while maintaining backward compatibility with current 802.11b devices. The target data rate for the project is at least 20 Mbps.
802.11h	Enhancing the 802.11 MAC and 802.11a PHY to provide network management and control extensions for spectrum and transmitting power management in the 5-GHz band. This is will allow regulatory acceptance of the standard in some European countries.
802.11i	Enhancing the security and authentication mechanisms of the 802.11 standard.
802.11x	Also aimed at enhancing security of 802.11b.

Table 3.1: IEEE 802.11 variants

802.11b with other 802.11 variants.

3.2.1.1 DSSS

DSSS systems mix high-speed bit patterns with the information being sent to spread the RF carrier. Each bit of information has a redundant bit pattern associated with it, effectively spreading the signal over a wider bandwidth. These bit patterns vary in length and the rate at which they are mixed into the RF carrier. They are called chips or chipping codes and vary in length from as small as 11 bits to extremely long sequences. The speed at which they are transmitted is called the chipping rate. To an observer, these sequences appear to be noise and are also called pseudo-random noise codes (PN-codes). PN-codes are usually introduced into the signal through the use of hardware-based shift registers, and the techniques used to introduce them are divided into several groups including Barker codes, Gold codes, M-sequences, and Kasami codes.

These spreading codes allow the use of statistical recovery methods to repair damaged transmissions. Another side effect of spreading the signal is lower spectral density - that is, the same amount of signal power is distributed over more band-

width. The effect of a less spectrally dense signal is that it is less likely to interfere with spectrally dense narrowband signals. Narrowband signals are also less likely to interfere with a DSSS signal because the narrowband signal is spread as part of the correlation function at the receiver.

The frequency channel in IEEE 802.11 DSSS is 22-MHz wide. This means that it supports three non-overlapping channels for operation. This is why only three IEEE 802.11b DSSS systems can be collocated.

In addition to spreading the signal, modulation techniques are used to encode the data signal through predictable variations of the radio signal. IEEE 802.11 specifies two types of DPSK modulation for DSSS systems. The first is BPSK and the second is QPSK. Phase-shift keying (PSK), as the name implies, detects the phase of the radio signal. BPSK detects 180-degree inversion of the signal, representing a binary 0 or 1. This method has an effective data rate of 1 Mbps. QPSK detects 90-degree phase shifts. This doubles the data rate to 2 Mbps. IEEE 802.11b adds CCK and packet binary convolutional coding (PBCC), which provide data rates up to 11 Mbps.

3.2.1.2 Orthogonal Frequency-Division Multiplexing

IEEE 802.11a (5 GHz) uses orthogonal frequency-division multiplexing (OFDM) as its frequency management technique and adds several versions of quadrature amplitude modulation (QAM) in support of data rates up to 54 Mbps. Bell Laboratories patented OFDM in 1970 and it is based on a mathematical process called the fast Fourier transform (FFT). FFT enables 52 channels to overlap without losing their individuality or orthogonality. Overlapping channels is a more efficient use of the spectrum and enables them to be processed at the receiver more efficiently. IEEE 802.11a OFDM divides the carrier frequency into 52 low-speed subcarriers. Forty-eight of these carriers are used for data and four are used as pilot carriers. The pilot subcarriers allow frequency alignment at the receiver.

One of the biggest advantages of OFDM is its resistance to multipath interference and delay spread. Multipath is caused when radio waves reflect and pass through objects in the environment. Radio waves are attenuated or weakened in a wide range depending on the object's materials. Some materials (such as metal) are opaque to radio transmissions. One can imagine that a cluttered environment would be very different from an open warehouse environment for radio wave transmission and reception. This environmental variability is why it is so hard to estimate the range and data rate of an IEEE 802.11 system. Because of reflections and attenuation, a single transmission can be at different signal strengths and from different directions depending on the types of materials it encounters. This is the multipath aspect of interference. IEEE 802.11a supports data rates from 6 to 54 Mbps. It utilizes BPSK, QPSK, and QAM to achieve the various data rates.

Delay spread is associated with multipath. Because the signal is traveling over different paths to the receiver, the signal arrives at different times. This is delay spread. As the transmission rate increases, the likelihood of interference from previously transmitted signals increases. Multipath and delay spread are not much of an issue at data rates less than 3 or 4 Mbps, but some sort of mechanism is required as rates increase to mitigate the effect of multipath and delay spread. In IEEE 802.11b, it is CCK modulation. In 802.11a, it is OFDM. The IEEE 802.11g specification also uses OFDM as its frequency management mechanism. The adoption and refinement of advanced semiconductor materials and radio transmission technologies for IEEE 802.11 provides a solid basis for the implementation of higher-level functions. The next step up the protocol ladder is the definition of access functionality. Without structured access, the physical medium would be unusable.

OFDM is not a new technique. Most of the fundamental work was done in the late 1960s, and U.S. patent number 3.488.445 was issued in January 1970. Recent Digital Subscriber Line (DSL) work [high bit rate DSL (HDSL), very high bit rate DSL (VDSL), and asymmetric DSL (ADSL)] and wireless data applications have rekindled interest in OFDM, especially now that better signal processing techniques make it more practical. OFDM does, however, differ from other emerging encoding techniques such as code-division multiple access (CDMA) in its approach. CDMA uses complex mathematical transforms to put multiple transmissions onto a single carrier; OFDM encodes a single transmission into multiple subcarriers. The mathematics underlying the code division in CDMA is far more complicated than in OFDM. OFDM devices use one wide-frequency channel by breaking it up into several component subchannels. Each subchannel is used to transmit data. All the low subchannels are then multiplexed into one *vast* combined channel.

3.2.1.3 DCF

The distributed coordination function defines how the medium is shared among members of the wireless network. It provides mechanisms for negotiating access to the wireless medium as well as mechanisms for reliable data delivery. One of the fundamental differences between wired and wireless media is that it is difficult to detect and manage data collisions on wireless media. The primary reason for this is that stations in a radio network are not guaranteed to hear every other station's transmissions. This is typically the case when an AP is used in IEEE 802.11's infrastructure BSS and is called the hidden-node problem.

3.2.1.4 PCF

The point coordination function (PCF) polls associated stations and manages frame transmissions on their behalf. A station performing PCF traffic management is

called a point coordinator (PC). The PCF is an optional capability that provides connection-oriented services for delay-sensitive traffic. The PCF is more complex to implement, but it provides a moderate level of priority frame delivery for time-sensitive transmissions. The PC uses beacon signals to broadcast duration for a contention-free period to all associated stations. This causes them to update their network allocation vector (NAV) and wait for the duration of the contention-free period. In addition, stations must await the PCF interframe space (PIFS) interval to further decrease the possibility of data collisions. The transmission of the additional polling and ACK messages required by the PCF is optimized through piggybacking multiple messages in a single transmission. For example, the PC may append both acknowledgments (Ax) of previous transmissions and polling messages for new traffic to a data frame. This enables the transmission to avoid waiting the interframe interval specified for individual frame transmissions. The basic access method for 802.11 is the DCF, which uses CSMA/CA. This requires each station to listen for other users. If the channel is idle, the station may transmit. If the station is busy, it waits until transmission stops and then enters into a random back-off procedure. This prevents multiple stations from seizing the medium immediately after completion of the preceding transmission. Packet reception in DCF requires acknowledgment. The period between completion of packet transmission and start of the ACK frame is one short interframe space (SIFS). ACK frames have a higher priority than other traffic. Fast acknowledgment is one of the salient features of the 802.11 standard, because it requires ACKs to be handled at the MAC sublayer. Transmissions other than ACKs must wait at least one DCF interframe space (DIFS) before transmitting data. If a transmitter senses a busy medium, it determines a random back-off period by setting an internal timer to an integer number of slot times. On expiration of a DIFS, the timer begins to decrement. If the time reaches zero, the station may begin transmission. If the channel is seized by another station before the timer reaches zero, the timer setting is retained at the decremented value for subsequent transmission. The method described above relies on the physical carrier sense. The underlying assumption is that every station can *hear* all other stations.

3.2.1.5 Mobility

Mobility of wireless stations may be the most important feature of a wireless LAN. The chief motivation of deploying a WLAN is to enable stations to move about freely from location to location either within a specific WLAN or between different WLAN segments. For compatibility purposes, the 802.11 MAC must appear to the upper layers of the network as a standard 802 LAN. The 802.11 MAC layer is forced to handle station mobility in a fashion that is transparent to the upper layers of the 802 LAN stack. This forces functionality into the 802.11 MAC layer that is

typically handled by upper layers in the OSI model. To understand this design restriction, it is important first to appreciate the difference between true mobility and mere portability. Portability certainly results in a net productivity gain because users can access information resources wherever it is convenient to do so. At the core, however, portability removes only the physical barriers to connectivity. It is easy to carry a laptop between several locations, so people do. But portability does not change the ritual of connecting to networks at each new location. It is still necessary to physically connect to the network and reestablish network connections, and network connections cannot be used while the device is being moved.

Mobility removes further barriers, most of which are based on the logical network architecture. Network connections stay active even while the device is in motion. This is critical for tasks requiring persistent, long-lived connections, which may be found in database applications. IEEE 802.11 is implemented at the link layer and provides link-layer mobility. IP does not allow this. The 802.11 hosts can move within the last network freely, but IP, as it is currently deployed, provides no way to move across subnet boundaries. To the IP-based hosts of the outside world, the virtual private network (VPN) access control boxes are the last hop routers. To access an 802.11 wireless station with an IP address on the wireless network, it is possible to simply go through the IP router to the target network regardless of whether a wireless station is connected to the first or third AP. The target network is reachable through the last hop router. As far as the outside world can tell, the wireless station might as well be a workstation connected to an Ethernet. A second requirement for mobility is that the IP address does not change when connecting to any of the access points. New IP addresses interrupt open connections. If a wireless station connects to the first AP, it must keep the same address when it connects to the third AP.

A corollary to the second requirement is that all of the wireless stations must be on the same IP subnet. As long as a station stays on the same IP subnet, it does not need to reinitialize its networking stack and can keep its TCP connections open. If it leaves the subnet, though, it needs to get a new IP address and reestablish any open connections. Multiple subnets are not forbidden, but if you have different IP subnets, seamless mobility between subnets is not possible. The *single IP subnet backbone* restriction is a reflection on the technology deployed within most organizations. Mobile IP was standardized in late 1996 in RFC 2002 [Per96], but it has yet to see widespread deployment. Until Mobile IP can be deployed, network designers must live within the limitations of IP and design networks based on fixed locations for IP addresses. A backbone network may be physically large, but it is fundamentally constrained by the requirement that all access points connect directly to the backbone router (and each other) at the link layer.

3.2.2 IEEE 802.16 – WiMAX

The IEEE 802.16 family of standards and its associated industry consortium promise to deliver high data rates over large areas to a large number of users in the near future. This exciting addition to current broadband options such as DSL, cable, and Wi-Fi promises to rapidly provide broadband access to locations in the world's rural and developing areas where broadband is currently unavailable, as well as competing for urban market share. WiMAX's competitiveness in the marketplace largely depends on the actual data rates and ranges that are achieved, but this has been difficult to judge due to the large number of possible options and competing marketing claims.

IEEE standard 802.16, the first version of which was completed in October 2001, defines the air interface and medium access control (MAC) protocol for a wireless metropolitan area network (WMAN), intended for providing high-bandwidth wireless voice and data for residential and enterprise use. This is the first industry-wide standard that can be used for fixed wireless access with substantially higher bandwidth than most cellular networks. The IEEE 802.16 standard, often referred to as WiMAX, heralds the entry of broadband wireless access as a major new tool in the effort to link homes and businesses to core telecommunications networks worldwide.

In the near future 802.16 will offer a mobile and quickly deployable alternative to cabled access networks, such as fiber optic links, coaxial systems using cable modems, and digital subscriber line (DSL) links. Because wireless systems have the capacity to address broad geographic areas without the costly infrastructure required to deploy cable links to individual sites, the technology may prove less expensive to deploy and should lead to more ubiquitous broadband access. Wireless broadband systems have been in use for several years, but the development of this new standard marks the maturation of the industry and a new level of competitiveness for non-line of sight (NLOS) wireless broadband services.

Historically, 802.16 activities were initiated at an August 1998 meeting called by the National Wireless Electronics Systems Testbed (N-WEST) of the U.S. National Institute of Standards and Technology. The effort was welcomed in IEEE 802, which led to the formation of the 802.16 Working Group, which has held week-long meetings at least bimonthly since July 1999. Development of 802.16 and the included WirelessMAN air interface, along with associated standards and amendments, is the responsibility of IEEE Working Group 802.16 on Broadband Wireless Access (BWA) Standards. The Working Group's initial interest was in the 10-66 GHz range, but more recent interest is behind the 2-11 GHz amendment project that led to IEEE 802.16a and was completed in January 2001. The new 802.16d upgrade to the 802.16a standard was recently approved in June 2004 (now named 802.16-2004), and primarily introduces some performance enhancement features in the uplink. Equipment based on this standard is expected to be dominant in the first

version of products. Currently the standardization of 802.16e is underway, which promises to support mobility up to speeds of 110-130 km/h and an asymmetrical link structure that will enable the subscriber station to have a handheld form factor for PDAs, phones, or laptops.

In order to rapidly converge on a worldwide standard, a staggering number of options are provided in the various 802.16 standards for parameters related to the MAC and physical (PHY) layers. In order to ensure that resulting 802.16-based devices are in fact interoperable, an industry consortium called the WiMAX Forum was created. The WiMAX Forum develops guidelines known as profiles, which specify the frequency band of operation, the PHY to be used, and a number of other parameters. Adherence to a given profile should enable interoperability between vendor products. The WiMAX Forum has identified several frequency bands for the initial 802.16d products, notably in both licensed (2.5-2.69 and 3.4-3.6 GHz) and unlicensed spectrum (5.725-5.850 GHz). Due to all the potential options in the standards, as well as the huge ranges of data rates, ranges, and other performance measures that are being quoted as achievable for 802.16, there is presently a significant amount of confusion about what type of performance can really be expected from WiMAX-compliant systems in the near future, although some larger testbeds for example in Heidelberg, Sankt Augustin, Berlin and Kaiserslautern were built. Data rates up to 3 Mbps are available.

The IEEE 802.l6a/d standard defines three different PHYs that can be used in conjunction with the MAC layer to provide a reliable end-to-end link. These air interface specifications are:

- WirelessMAN-SCa: a single-carrier modulated air interface.
- WirelessMAN-OFDM: a 256-carrier orthogonal-frequency division multiplexing (OFDM) scheme. Multiple access of different subscriber stations (SSs) is time-division
- WirelessMAN-OFDMA: a 2048-carrier OFDM scheme. Multiple access is provided by assigning a subset of the carriers to an individual receiver [1], so this version is often referred to as OFD multiple access (OFDMA).

Of these three air interfaces, the two OFDM-based systems are more suitable for non-LOS operation due to the simplicity of the equalization process for multicarrier signals. Of the two OFDM-based air interfaces, 256-carrier WirelessMAN-OFDM seems to be favored by the vendor community for reasons such as lower peak to average ratio, faster fast Fourier transform (FFT) calculation, and less stringent

[1] In WirelessMAN-OFDMA, multiple access is provided by using a combination of TDMA and OFDMA.

requirements for frequency synchronization compared to 2048-carrier WirelessMAN-OFDMA. All profiles currently defined by the WiMAX Forum specify the 256-carrier OFDM PHY. For this reason, the rest of the work will focus primarily on the 256-carrier OFDM air interface. Of these 256 subcarriers, 192 are used for user data, with 56 nulled for a guard band and eight used as permanent pilot symbols. In order to provide robustness to dispersive multipath channels, 8, 16, 32, or 64 additional samples are prepended as the cyclic prefix, depending on the expected channel delay spread. In order to ensure global implementation, the IEEE 802.16 standard has been defined with a variable channel bandwidth. The channel bandwidth can be an integer multiple of 1.25 MHz, 1.5 MHz, and 1.75 MHz with a maximum of 20 MHz. This large choice of possible bandwidths is being narrowed down to a few possibilities by the WiMAX Forum, whose primary task is to ensure interoperability between implementations of the 802.16d standard by different vendors.

The 802.l6a/d standard defines seven combinations of modulation and coding rate that can be used to achieve various trade-offs of data rate and robustness, depending on channel and interference conditions. These possible combinations, shown in table 3.2, follow a similar pattern to the modulation/coding pairs available in the IEEE 802.11 a/g standard for wireless LANs.

One departure from the 802.11 standard is that 802.16 uses an outer Reed-Solomon (RS) block code concatenated with an inner convolutional code. The RS code is fixed and derived from a systematic $RS(N = 255, K = 239, T = 8)$ code using $GF(2^8)$, and thus adds about 10 percent overhead. The inner convolution code has constraint length 7, and its rate varies between 1/2 and 3/4. Naturally, interleaving is also employed to reduce the effect of burst errors. Turbo coding has been left as an optional feature, which can improve the coverage and/or capacity of the system, at the price of increased decoding latency and complexity. Initial versions of WiMAX-compliant products are not expected to include turbo coding [GWAR05].

The allowed modulation schemes in the downlink (DL) and uplink (UL) are binary phase shift keying (BPSK), quaternary PSK (QPSK), 16-quadrature amplitude modulation (QAM), and 64-QAM. A total of eight pilot subcarriers are inserted into each data burst in order to constitute the OFDM symbol, and they are modulated according to their carrier locations within the OFDM symbol. Additionally, known preambles are used in 802.16d to aid the receiver with synchronization and channel estimation. In the DL a *long preamble* of two OFDM symbols is sent at the beginning of each frame. In the UL a *short preamble* of one OFDM symbol is sent by the SS at the beginning of every frame.

The 802.16 standard provides optional features and a signaling structure that enables the usage of intelligent antenna systems. A separate point-to-multipoint (PMP) frame structure is defined that enables the transmission of DL and UL bursts using directed beams, each intended for one or more SSs. Additional signaling

between the base stations (BSs) and SSs has been defined that allows the SS to provide channel quality feedback to the BS. The real and imaginary components of the channel response for each of the directed beams and specific subcarriers are provided to the BS. The BS can specify the resolution in the frequency domain of this feedback. The standard allows the SS to provide channel response for every 4th, 8th, 16th, 32nd, or 64th subcarrier. Some initial WiMAX-compliant products will implement adaptive antennas to improve the spectral efficiency of the system [GWAR05].

The MAC Layer of IEEE 802.16 was designed for PMP broadband wireless access applications [2]. It is designed to meet the requirements of very-high-data-rate applications with a variety of quality of service (QoS) requirements. The signaling and bandwidth allocation algorithms have been designed to accommodate hundreds of terminals per channel. The standard allows each terminal to be shared by multiple end users. The services required by the end users can be varied in their bandwidth and latency requirements, which demands that the MAC layer protocol be flexible and efficient over a vast range of different data traffic models. The system has been designed to include legacy time-division multiplex (TDM) voice and data, Internet Protocol (IP) connectivity, and Voice over IP (VoIP).

The MAC layer of IEEE 802.16 is divided into convergence-specific and common part sublayers. Convergence-specific sublayers are used to map the transport-layer-specific traffic to a MAC that is flexible enough to efficiently carry any traffic type. The common part sublayer, as its name suggests, is independent of the transport mechanism, and responsible for fragmentation and segmentation of MAC service data units (SDUx) into MAC protocol data units (PDUs), QoS control, and scheduling and retransmission of MAC PDUs.

The bandwidth request and grant mechanism has been designed to be scalable, efficient, and self-correcting. The 802.16 access system does not lose efficiency when presented with multiple connections per terminal, multiple QoS levels per terminal, and a large number of statistically multiplexed users. It takes advantage of a wide variety of request mechanisms, balancing the stability of contention-less access with the efficiency of contention-oriented access. While extensive bandwidth allocation and QoS mechanisms are provided in the standard, the details of scheduling and reservation management are left undefined such that product differentiations may be achieved through different vendor implementations.

The IEEE 802.16 standard has been designed to support frequency-division duplex (FDD) and time-division duplex (TDD). In FDD mode there is additional support for unframed FDD operation, where the transmission does not contain a frame structure and is asynchronous. The MAC at the BS creates a DL frame (subframe for TDD), starting with a preamble that is used for synchronization and

[2] Later amendments to 802.16a and 802.16d also allow for mesh network architecture.

Rate ID	Modulation	Coding	Information [bits/symbol]	Information [bits/OFDM symbol]	Peak data rate in 5 MHz [Mbps]
0	BPSK	1/2	0.5	88	1.89
1	QPSK	1/2	1	184	3.95
2	QPSK	3/4	1.5	280	6.00
3	16QAM	1/2	2	376	8.06
4	16QAM	3/4	3	568	12.18
5	64QAM	2/3	4	760	16.30
6	64QAM	3/4	4.5	856	18.36

Table 3.2: Modulation and coding schemes for 802.16d

channel estimation. A frame control header (FCH) transmitted after the preamble specifies the burst profile for the rest of the frame. This is required since the bursts are transmitted with different modulation and coding schemes. The FCH is followed by one or multiple downlink bursts, each transmitted according to the burst profile and consisting of an integer number of OFDM symbols. The location and profile of the first downlink burst is specified in the downlink frame prefix (DLFP), part of the FCH.

The initial channel estimates obtained from the preamble can be used in adaptive tracking of the channel using the embedded pilot in each OFDM symbol. Since the duration of each frame is short (1-2 ms), it is possible to omit adaptive channel tracking for most fixed wireless applications since the channel is unlikely to change significantly during the frame. Data bursts are transmitted in order of decreasing robustness to allow the SSs to receive reliable data before risking a burst error that could cause loss of synchronization. In the DL, a TDM portion immediately follows the FCH and is used for unsolicited grant service (UGS), useful for constant bit rate applications with strict delay restrictions such as VoIP.

For security, the 802.16d standard specifies the Data Encryption Standard (DES) as the mandatory encryption mechanism for data and Triple DES for key encryption. The allowed cryptographic suites are:

- CBC-Mode 56-bit DES, no data authentication and 3-DES, 128
- CBC-Mode 56-bit DES, no data authentication and RSA, 1024
- CCM-mode AES, no data authentication and AES, 128

The WiMAX Forum is currently evaluating these long-standing choices in light of recent advances in encryption technology. Specifically, the Forum is considering whether to specify the Advanced Encryption System (AES) from the US National Institute of Standards as an alternate encryption method; AES could become the preferred choice for service providers.

3.2.2.1 Uses for WiMAX

WiMAX is a wireless metropolitan area network (MAN) technology that can connect IEEE 802.11(Wi-Fi) hotspots with each other and to other parts of the Internet and provide a wireless alternative to cable and DSL for *last mile* (last km) broadband access. IEEE 802.16 provides up to 50 km (31 miles) of linear service area range and allows connectivity between users without a direct line of sight. Note that this should not be taken to mean that users 50 km (31 miles) away without line of sight will have connectivity. Practical limits from real world tests seem to be around 3 to 5 miles (5 to 8 kilometers) [ROea06, ROea05]. The technology has been claimed to provide shared data rates up to 70 Mbps, which, according to WiMAX proponents, is enough bandwidth to simultaneously support more than 60 businesses with T1-type connectivity and well over a thousand homes at 1 Mbps DSL-level connectivity. Real world tests, however, show practical maximum data rates between 500 Kbps and 2 Mbps, depending on conditions at a given site.

It is also anticipated that WiMAX will allow interpenetration for broadband service provision of VoIP, video, and Internet access - simultaneously. Most cable and traditional telephone companies are closely examining or actively trial-testing the potential of WiMAX for *last mile* connectivity. This should result in better price points for both home and business customers as competition results from the elimination of the *captive* customer bases both telephone and cable networks traditionally enjoyed. Even in areas without preexisting physical cable or telephone networks, WiMAX could allow access between anyone within range of each other. Home units the size of a paperback book that provide both phone and network connection points are already available and easy to install.

There is also interesting potential for interoperability of WiMAX with legacy cellular networks. WiMAX antennas can share a cell tower without compromising the function of cellular arrays already in place. Companies that already lease cell sites in widespread service areas have a unique opportunity to diversify, and often already have the necessary spectrum available to them (i.e. they own the licenses for radio frequencies important to increased speed and/or range of a WiMAX connection). WiMAX antennas may be even connected to an Internet backbone via either a light fiber optics cable or a directional microwave link. Some cellular companies are evaluating WiMAX as a means of increasing bandwidth for a variety of data-intensive applications. In line with these possible applications is the technology's ability to serve as a very high bandwidth *backhaul* for Internet or cellular phone traffic from remote areas back to a backbone. Although the cost-effectiveness of WiMAX in a remote application will be higher, it is definitely not limited to such applications, and may in fact be an answer to expensive urban deployments of T1 backhauls as well. Given developing countries' (such as in Africa) limited wired infrastructure, the costs to install a WiMAX station in conjunction with an existing cellular tower

or even as a solitary hub will be diminutive in comparison to developing a wired solution. The wide, flat expanses and low population density of such an area lends itself well to WiMAX and its current diametrical range of 30 miles. For countries that have skipped wired infrastructure as a result of inhibitive costs and unsympathetic geography, WiMAX can enhance wireless infrastructure in an inexpensive, decentralized, deployment-friendly and effective manner.

Another application under consideration is gaming. Sony and Microsoft are closely considering the addition of WiMAX as a feature in their next generation game console. This will allow gamers to create ad hoc networks with other players. This may prove to be one of the *killer applications* driving WiMAX adoption: Wi-Fi-like functionality with vastly improved range and greatly reduced network latency and the capability to create ad hoc mesh networks [Sha05].

3.2.2.2 QoS

To support Quality of Service (QoS) IEEE 802.16 maintains the concept of a service flow. The Upstream Service Flow Types defined in IEEE 802.16 are: Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Non-Real-Time Polling Service (nrtPS) and Best Effort (BE). Those different types of service flows should be treated differently by the MAC protocol scheduling process.

Unsolicited Grant Service (UGS) Flows UGS is designed to support real-time services flows that generate fixed size data packets on a periodic basis, such as Voice over IP. The service offers fixed size unsolicited data grants (transmission opportunities) on a periodic basis. This eliminates the overhead and latency of requiring the modem to send requests for transmission opportunities. Piggyback requests are prohibited in UGS. The key service parameters for UGS service flows are Unsolicited Grant Size, Grants Per Interval, Nominal Grant Interval and Tolerated Grant Jitter.

Real-Time Polling Service (rtPS) Flows rtPs is designed to support real-time service flows that generate variable size data packets on a periodic basis, such as MPEG video. The service offers periodic unicast request opportunities, which meet the flow's real-time needs and allow the modem to specify the size of the desired grant. The modem is prohibited from using any connection or piggyback requests. The key service parameters here are: Nominal Polling Interval, Tolerated Poll Jitter and Minimum Reserved Traffic Rate.

Non-Real-Time Polling Service (nrtPS) Flows The nrtPs is designed to support non-real-time service flows that require variable size data grants on a regular basis, such as high bandwidth FTP. The service offers unicast polls on a periodic basis, but

using more spaced intervals than rtPS. This ensures that the flow receives request opportunities even during network congestion. In addition, the modem is allowed to use contention and piggyback request opportunities. The key service parameters here are: Nominal Polling Interval, Minimum Reserved Traffic Rate and Traffic Priority.

Best Effort (BE) Flows In BE service the modem is allowed to use contention and piggyback request opportunities, but neither periodic polls nor periodic data grants will be sent, unless they are needed to satisfy the minimum reserved bandwidth for that service. The key service parameters for BE service flows are Minimum Reserved Traffic Rate and Traffic Priority.

3.2.3 Future Wireless Networks

Since July 1999, the IEEE 802.16 Working Group on Broadband Wireless Access (`http://grouper.ieee.org/groups/802/16/`) has been openly developing voluntary consensus standards for Wireless Metropolitan Area Networks with global applicability. Addressing the demand for broadband access to buildings, IEEE 802.16 provides solutions that are more economical than wired-line alternatives. The standards set the stage for a revolution in reliable, high-speed network access in the *last mile* of Internet by homes and enterprises. On December 11th, 2002, the IEEE Standards Board approved the establishment of IEEE 802.20 Mobile Broadband Wireless Access (MBWA) Working Group (`http://grouper.ieee.org/groups/802/20/`). It described the scope of IEEE 802.20 as:

> *Specification of physical and medium access control layers of an air interface for interoperable mobile broadband wireless access systems, operating in licensed bands below 3.5 GHz, optimized for IP-data transport, with peak data rates per user in excess of 1 Mbps. It supports various vehicular mobility classes up to 250 km/h in a MAN environment and targets spectral efficiencies, sustained user data rates and numbers of active users that are all significantly higher than achieved by existing mobile systems.*

According to the above scope, the basic features of IEEE 802.20 include compatibility, coexistence, distinct identity, technology feasibility, and economic feasibility. According to the MBWA announcement, IEEE 802.20 is aimed at mobile communication, and its data rate can reach more than 2 Mbps in high speed mobile application. IEEE 802.20 is the first real broadband wireless network standard that dedicatedly supports the mobility of network.

3.2.4 4th Generation of Mobile Communication

New data services, interactive TV and evolving Internet behavior will influence mobile data usage. Long sessions in always-on mode will force a re-think of radio access technology to achieve the required - but not easy to attain - capacity at low cost. Coverage will be based on large umbrella cells (3G, WiMAX) and numerous pico cells interconnected to provide the user with seamless high data rate (several Mbps) sessions. Scalable and progressive deployments are possible while protecting the operator's long-term investment. The 4G infrastructure operator will mix several technologies, each of which has its optimal usage. The connection to one of them will result in a real-time trade-off which will offer the user the best possible service.

Voice was the driver for second-generation mobile and has been a considerable success. Today, video and TV services are driving forward third generation (3G) deployment. And in the future, low cost, high speed data will drive forward the fourth generation (4G) as short-range communication emerges. Service and application ubiquity, with a high degree of personalization and synchronization between various user appliances, will be another driver. At the same time, it is probable that the radio access network will evolve from a centralized architecture to a distributed one. The evolution from 3G to 4G will be driven by services that offer better quality (e.g. video and sound) thanks to greater bandwidth, more sophistication in the association of a large quantity of information, and improved personalization. Convergence with other network (enterprise, fixed) services will come about through the high session data rate. It will require an always-on connection and a revenue model based on a fixed monthly fee. The impact on network capacity is expected to be significant. Machine-to-machine transmission will involve two basic equipment types: sensors (which measure parameters) and tags (which are generally read/write equipment). It is expected that users will require high data rates, similar to those on fixed networks, for data and streaming applications. Mobile terminal usage (laptops, Personal digital assistants, handhelds) is expected to grow rapidly as they become more user friendly. Fluid high quality video and network reactivity are important user requirements. Key infrastructure design requirements include: fast response, high session rate, high capacity, low user charges, rapid return on investment for operators, investment that is in line with the growth in demand, and simple autonomous terminals. The infrastructure will be much more distributed than in current deployments, facilitating the introduction of a new source of local traffic: machine-to-machine.

A simple calculation illustrates the order of magnitude. The design target in terms of radio performance is to achieve a scalable capacity from 50 to 500 bps/Hz/km^2 (including capacity for indoor use). As a comparison, the expected best performance of 3G is around 10 bps/Hz/km^2 using High Speed Downlink Packet Access (HSDPA), Multiple-Input Multiple-Output (MIMO), etc. No current technology is

capable of such performance.

Based on various traffic analyses, the Wireless World Initiative (WWI) has issued target air interface performance figures. A consensus has been reached around peak rates of 100 Mbps in mobile situations and 1 Gbps in nomadic and pedestrian situations, at least as targets. So far, in a 10 MHz spectrum, a carrier rate of 20 Mbps has been achieved when the user is moving at high speed, and 40 Mbps in nomadic use. These values will double when MIMO is introduced. Clearly, the bit rate should be associated with an amount of spectrum. For mobile use, a good target is a network performance of 5 bps/Hz, rising to 8 bps/Hz in nomadic use.

Many technologies are competing on the road to 4G. Three paths are possible, even if they are more or less specialized. The first is the 3G-centric path, in which Code Division Multiple Access (CDMA) will be progressively pushed to the point at which terminal manufacturers will give up. When this point is reached, another technology will be needed to realize the required increases in capacity and data rates. The second path is the radio LAN one. Widespread deployment of Wi-Fi was expected to start in 2005 for PCs, laptops and PDAs. In enterprises, voice may start to be carried by Voice over Wireless LAN (VoWLAN). However, it is not clear what the next successful technology will be. Reaching a consensus on a 200 Mbps (and more) technology will be a lengthy task, with too many proprietary solutions on offer.

A third path is IEEE 802.16e and 802.20, which are simpler than 3G for the equivalent performance. A core network evolution towards a broadband Next Generation Network (NGN) will facilitate the introduction of new access network technologies through standard access gateways, based on ETSI-TISPAN, ITU-T, 3GPP, China Communication Standards Association (CCSA) and other standards.

How can an operator provide a large number of users with high session data rates using its existing infrastructure? At least two technologies are needed. The first (called *parent coverage*) is dedicated to large coverage and real-time services. Legacy technologies, such as 2G/3G and their evolutions will be complemented by Wi-Fi and WiMAX. A second set of technologies is needed to increase capacity, and can be designed without any constraints on coverage continuity. This is known as *pico cell coverage*. Only the use of both technologies can achieve both targets. Handover between parent coverage and pico cell coverage is different from a classical roaming process, but similar to classical handover. Parent coverage can also be used as a back-up when service delivery in the pico cell becomes too difficult.

Some of the key technologies required for 4G are briefly described below:

OFDMA Orthogonal Frequency Division Multiplexing (OFDM) not only provides clear advantages for physical layer performance, but also a framework for improving layer 2 performance by proposing an additional degree of freedom. Using OFDM,

it is possible to exploit the time domain, the space domain, the frequency domain and even the code domain to optimize radio channel usage. It ensures very robust transmission in multi-path environments with reduced receiver complexity. The signal is split into orthogonal subcarriers, on each of which the signal is *narrowband* (a few kHz) and therefore immune to multi-path effects, provided a guard interval is inserted between each OFDM symbol. OFDM also provides a frequency diversity gain, improving the physical layer performance. It is also compatible with other enhancement technologies, such as smart antennas and MIMO. OFDM modulation can also be employed as a multiple access technology (Orthogonal Frequency Division Multiple Access; OFDMA). In this case, each OFDM symbol can transmit information to/from several users using a different set of subcarriers (subchannels). This not only provides additional flexibility for resource allocation (increasing the capacity), but also enables cross-layer optimization of radio link usage.

SDR Software Defined Radio (SDR) benefits from today's high processing power to develop multi-band, multi-standard base stations and terminals. Although in future the terminals will adapt the air interface to the available radio access technology, at present this is done by the infrastructure. Several infrastructure gains are expected from SDR. For example, to increase network capacity at a specific time (e.g. during a sports event), an operator will reconfigure its network adding several modems at a given Base Transceiver Station (BTS). SDR makes this reconfiguration easy. In the context of 4G systems, SDR will become an enabler for the aggregation of multi-standard pico/micro cells. For a manufacturer, this can be a powerful aid to providing multi-standard, multi-band equipment with reduced development effort and costs through simultaneous multi-channel processing.

MIMO uses signal multiplexing between multiple transmitting antennas (space multiplex) and time or frequency. It is well suited to OFDM, as it is possible to process independent time symbols as soon as the OFDM waveform is correctly designed for the channel. This aspect of OFDM greatly simplifies processing. The signal transmitted by m antennas is received by n antennas. Processing of the received signals may deliver several performance improvements: range, quality of received signal and spectrum efficiency. In principle, MIMO is more efficient when many multiple path signals are received. The performance in cellular deployments is still subject to research and simulations. However, it is generally admitted that the gain in spectrum efficiency is directly related to the minimum number of antennas in the link.

Handover and Mobility Handover technologies based on mobile IP technology have been considered for data and voice. Mobile IP techniques are slow but can

be accelerated with classical methods (hierarchical, fast mobile IP). These methods are applicable to data and probably also voice. In single-frequency networks, it is necessary to reconsider the handover methods. Several techniques can be used when the carrier to interference ratio is negative (e.g. VSF-OFDM, bit repetition), but the drawback of these techniques is capacity. In OFDM, the same alternative exists as in CDMA, which is to use macro-diversity. In the case of OFDM, MIMO allows macro-diversity processing with performance gains. However, the implementation of macro-diversity implies that MIMO processing is centralized and transmissions are synchronous. This is not as complex as in CDMA, but such a technique should only be used in situations where spectrum is very scarce.

Multimedia Service Delivery, Service Adaptation and Robust Transmission Audio and video coding are scalable. For instance, a video flow can be split into three flows which can be transported independently: one base layer (30 Kbps), which is a robust flow but of limited quality (e.g. 5 images/s), and two enhancement flows (50 Kbps and 200 Kbps). The first flow provides availability, the other two quality and definition. In a streaming situation, the terminal will have three caches. In pico cellular coverage, the parent coverage establishes the service dialog and service start-up (with the base layer). As soon as the terminal enters pico cell coverage, the terminal caches are filled, starting with the base cache. Video (and audio) transmissions are currently transmitted without error and without packet loss. However, it is possible to allow error rates of about $10^{-5}/10^{-6}$ and a packet loss around $10^{-2}/10^{-3}$. Coded images still contain enough redundancy for error correction. It is possible to gain about 10 dB in transmission with a reasonable increase in complexity. Using the described technologies, multimedia transmission can provide a good quality user experience.

Coverage Coverage is achieved by adding new technologies (possibly in overlay mode) and progressively enhancing density. A WiMAX deployment as an example: first the parent coverage is deployed; it is then made denser by adding discontinuous pico cells, after which the pico cell is made denser but still discontinuously. Finally the pico cell coverage is made continuous either by using MIMO or by deploying another pico cell coverage in a different frequency band.

3.3 Wireless Network Requirements

When we describe media as real-time, we mean simply that the receiver is playing out the media stream as it is received, rather than simply storing the complete stream in a file for later playback. In the ideal case, playout at the receiver is

immediate and synchronous, although in practice some unavoidable transmission delay is imposed by the network.

The primary requirement that real-time media places on the transport protocol is for predictable variation in network transit time. Consider, for example, an IP telephony system transporting encoded voice in 20-millisecond frames: The source will transmit one packet every 20 milliseconds, and ideally we would like those to arrive with the same spacing so that the speech they contain can be played out immediately. Some variation in transit time can be accommodated by the insertion of additional buffering delay at the receiver, but this is possible only if that variation can be characterized and the receiver can adapt to match the variation.

A lesser requirement is reliable delivery of all packets by the network. Clearly, reliable delivery is desirable, but many audio and video applications can tolerate some loss: In the IP telephony example, loss of a single packet will result in a dropout of one-fiftieth of a second, which, with suitable error concealment, is barely noticeable. Because of the time-varying nature of media streams, some loss is usually acceptable because its effects are quickly corrected by the arrival of new data. The amount of loss that is acceptable depends on the application, the encoding method used, and the pattern of loss.

These requirements drive the choice of transport protocol. It should be clear that TCP/IP is not appropriate because it favors reliability over timeliness, and our applications require timely delivery. A UDP/IP-based transport should be suitable, provided that the variation in transit time of the network can be characterized and loss rates are acceptable. The standard Real-time Transport Protocol (RTP) builds on UDP/IP, and provides timing recovery and loss detection, to enable the development of robust systems.

Despite TCP's limitations for real-time applications, some audio/video applications use it for their transport. Such applications attempt to estimate the average throughput of the TCP connection and adapt their send rate to match. This approach can be made to work when tight end-to-end delay bounds are not required and an application has several seconds worth of buffering to cope with the variation in delivery time caused by TCP retransmission and congestion control. It does not work reliably for interactive applications, which need short end-to-end delay, because the variation in transit time caused by TCP is too great.

The primary rationale for the use of TCP/IP transport is that many firewalls pass TCP connections but block UDP. This situation is changing rapidly, as RTP-based systems become more prevalent and firewalls smarter. RTP provides for higher quality by enabling applications to adapt in a way that is appropriate for real-time media, and by promoting interoperability (because it is an open standard).

3.3.1 Real-time Media Transport

The key standard for audio/video transport in IP networks is the Real-time Transport Protocol (RTP), along with its associated profiles and payload formats. RTP aims to provide services useful for the transport of real-time media, such as audio and video, over IP networks. These services include timing recovery, loss detection and correction, payload and source identification, reception quality feedback, media synchronization, and membership management. RTP was originally designed for use in multicast conferences, using the lightweight sessions model. Since that time, it has proven useful for a range of other applications: in H.323 video conferencing, webcasting, and TV distribution; and in both wired and cellular telephony. The protocol has been demonstrated to scale from point-to-point use to multicast sessions with thousands of users, and from low-bandwidth cellular telephony applications to the delivery of uncompressed High-Definition Television (HDTV) signals at gigabit rates [Per03].

RTP was developed by the Audio/Video Transport working group of the IETF and has since been adopted by the ITU as part of its H.323 series of recommendations, and by various other standards organizations. The first version of RTP was completed in January 1996 [SCFJ96]. RTP needs to be profiled for particular uses before it is complete; an initial profile was defined along with the RTP specification, [Sch96] and several more profiles are under development. Profiles are accompanied by several payload format specifications, describing the transport of a particular media format.

RTP typically sits on top of UDP/IP transport, enhancing that transport with loss detection and reception quality reporting, provision for timing recovery and synchronization, payload and source identification, and marking of significant events within the media stream. Most implementations of RTP are part of an application or library that is layered above the UDP/IP sockets interface provided by the operating system. This is not the only possible design, though, and nothing in the RTP protocol requires UDP or IP. For example, some implementations layer RTP above TCP/IP, and others use RTP on non-IP networks, such as Asynchronous Transfer Mode (ATM) networks.

There are two parts to RTP: the data transfer protocol and an associated control protocol. The RTP data transfer protocol manages delivery of real-time data, such as audio and video, between end systems. It defines an additional level of framing for the media payload, incorporating a sequence number for loss detection, timestamp to enable timing recovery, payload type and source identifiers, and a marker for significant events within the media stream. Also specified are rules for timestamp and sequence number usage, although these rules are somewhat dependent on the profile and payload format in use, and for multiplexing multiple streams within a session. The RTP control protocol (RTCP) provides reception quality feedback,

participant identification, and synchronization between media streams. RTCP runs alongside RTP and provides periodic reporting of this information. Although data packets are typically sent every few milliseconds, the control protocol operates on the scale of seconds. The information sent in RTCP is necessary for synchronization between media streams - for example, for lip synchronization between audio and video - and can be useful for adapting the transmission according to reception quality feedback, and for identifying the participants.

RTP supports the notion of mixers and translators, middle boxes that can operate on the media as it flows between endpoints. These may be used to translate an RTP session between different lower-layer protocols - for example, bridging between participants on IPv4 and IPv6 networks, or bringing a unicast-only participant into a multicast group. They can also adapt a media stream in some way - for example, transcoding the data format to reduce the bandwidth, or mixing multiple streams together.

The final piece of the RTP framework is the payload formats, defining how particular media types are transported within RTP. Payload formats are referenced by RTP profiles, and they may also define certain properties of the RTP data transfer protocol. The relation between an RTP payload format and profile is primarily one of namespace, although the profile may also specify some general behavior for payload formats. The namespace relates the payload type identifier in the RTP packets to the payload format specifications, allowing an application to relate the data to a particular media codec. In some cases the mapping between payload type and payload format is static; in others the mapping is dynamic via an out-of-band control protocol. For example, the RTP profile for audio and video conferences with minimal control defines a set of static payload type assignments, and a mechanism for mapping between a MIME type identifying a payload format, and a payload type identifier using the Session Description Protocol (SDP).

The relation between a payload format and the RTP data transfer protocol is twofold: A payload format will specify the use of certain RTP header fields, and it may define an additional payload header. The output produced by a media codec is translated into a series of RTP data packets - some parts mapping onto the RTP header, some into a payload header, and most into the payload data. The complexity of this mapping process depends on the design of the codec and on the degree of error resilience required. In some cases the mapping is simple; in others it is more complex.

At its simplest, a payload format defines only the mapping between media clock and RTP timestamp, and mandates that each frame of codec output is placed directly into an RTP packet for transport. An example of this is the payload format for G.722.1 audio [Lut01]. Unfortunately, this is not sufficient in many cases because many codecs were developed without reference to the needs of a packet delivery system and need to be adapted to this environment. Others were designed for packet

networks but require additional header information. In these cases the payload format specification defines an additional payload header, to be placed after the main RTP header, and rules for generation of that header.

Many payload formats have been defined, matching the diversity of codecs that are in use today, and many more are under development. At the time of this writing, the following audio payload formats are in common use, although this is by no means an exhaustive list: G.711, G.723.1, G.726, G.728, G.729, GSM, QCELP, MP3, and DTMF.30,34,38,49 The commonly used video payload formats include H.261, H.263, and MPEG.

There are also payload formats that specify error correction schemes. For example, RFC 2198 [ea97] defines an audio redundancy encoding scheme, and RFC 2733 [RS99] defines a generic forward error correction scheme based on parity coding. In these payload formats there is an additional layer of indirection, the codec output is mapped onto RTP packets, and those packets themselves are mapped to produce an error-resilient transport.

3.3.1.1 RTP

A RTP session consists of a group of participants who are communicating using RTP. A participant may be active in multiple RTP sessions - for instance, one session for exchanging audio data and another session for exchanging video data. For each participant, the session is identified by a network address and port pair to which data should be sent, and a port pair on which data is received. The send and receive ports may be the same. Each port pair comprises two adjacent ports: an even-numbered port for RTP data packets, and the next higher (odd-numbered) port for RTCP control packets. The default port pair is 5004 and 5005 for UDP/IP, but many applications dynamically allocate ports during session setup and ignore the default. RTP sessions are designed to transport a single type of media; in a multimedia communication, each media type should be carried in a separate RTP session.

RTP Data Transfer Packet The format of an RTP data transfer packet is illustrated in figure 3.1.

The mandatory RTP data packet header is typically 12 octets in length, although it may contain a contributing source list, which can expand the length by 4 to 60 additional octets. The fields in the mandatory header are the payload type, sequence number, time-stamp, and synchronization source identifier. In addition, there is a count of contributing sources, a marker for interesting events, support for padding and a header extension, and a version number.

Because RTP sessions typically use a dynamically negotiated port pair, it is especially important to validate that packets received really are RTP, and not misdi-

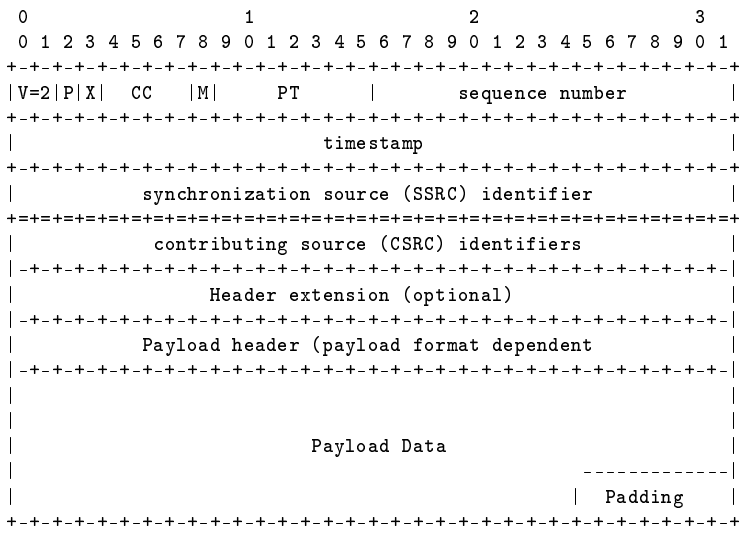


Figure 3.1: RTP data transfer packet

rected other data. At first glance, confirming this fact is nontrivial because RTP packets do not contain an explicit protocol identifier; however, by observing the progression of header fields over several packets, we can quickly obtain strong confidence in the validity of an RTP stream. Possible validity checks that can be performed on a stream of RTP packets are outlined in Appendix A of the RTP specification [SCFJ03]. There are two types of tests:

1. **Per-packet checking**, based on fixed known values of the header fields. For example, packets in which the version number is not equal to 2 are invalid, as are those with an unexpected payload type.

2. **Per-flow checking**, based on patterns in the header fields. For example, if the SSRC is constant, and the sequence number increments by one with each packet received, and the timestamp intervals are appropriate for the payload type, this is almost certainly an RTP flow and not a misdirected stream.

The per-flow checks are more likely to detect invalid packets, but they require additional state to be kept in the receiver. This state is required for a valid source, but care must be taken because holding too much state to detect invalid sources can lead to a denial-of-service attack, in which a malicious source floods a receiver with a stream of bogus packets designed to use up resources.

A robust implementation will employ strong per-packet validity checks to weed out as many invalid packets as possible before committing resources to the per-

flow checks to catch the others. It should also be prepared to aggressively discard state for sources that appear to be bogus, to mitigate the effects of denial-of-service attacks.

It is also possible to validate the contents of an RTP data stream against the corresponding RTCP control packets. To do this, the application discards RTP packets until an RTCP source description packet with the same SSRC is received. This is a very strong validity check, but it can result in significant validation delay, particularly in large sessions (because the RTCP reporting interval can be many seconds).

In addition to normal end systems, RTP supports middle boxes that can operate on a media stream within a session. Two classes of middle boxes are defined: translators and mixers.

A translator is an intermediate system that operates on RTP data while maintaining the synchronization source and timeline of a stream. Examples include systems that convert between media-encoding formats without mixing, that bridge between different transport protocols, that add or remove encryption, or that filter media streams. A translator is invisible to the RTP end systems unless those systems have prior knowledge of the untranslated media.

The defining characteristic of a translator is that each input stream produces a single output stream, with the same SSRC. The translator itself is not a participant in the RTP session - it does not have an SSRC and does not generate RTCP itself - and is invisible to the other participants.

A mixer is an intermediate system that receives RTP packets from a group of sources and combines them into a single output, possibly changing the encoding, before forwarding the result. Examples include the networked equivalent of an audio mixing deck, or a video picture-in-picture device. Because the timing of the input streams generally will not be synchronized, the mixer will have to make its own adjustments to synchronize the media before combining them, and hence it becomes the synchronization source of the output media stream. A mixer may use playout buffers for each arriving media stream to help maintain the timing relationships between streams. A mixer has its own SSRC, which is inserted into the data packets it generates. The SSRC identifiers from the input data packets are copied into the CSRC list of the output packet.

3.3.1.2 RTCP

An RTCP implementation has three parts: the packet formats, the timing rules, and the participant database. There are several types of RTCP packets. The compound packets are sent periodically. The interval between packets is known as the reporting interval. All RTCP activity happens in multiples of the reporting interval. In addition to being the time between packets, it is the time over which

reception quality statistics are calculated, and the time between updates of source description and lip synchronization information. The interval varies according to the media format in use and the size of the session; typically it is on the order of 5 seconds for small sessions, but it can increase to several minutes for very large groups. Senders are given special consideration in the calculation of the reporting interval, so their source description and lip synchronization information is sent frequently; receivers report less often. Each implementation is expected to maintain a participant database, based on the information collected from the RTCP packets it receives. This database is used to fill out the reception report packets that have to be sent periodically, but also for lip synchronization between received audio and video streams and to maintain source description information.

Each RTP session is identified by a network address and a pair of ports: one for RTP data and one for RTCP data. The RTP data port should be even, and the RTCP port should be one above the RTP port. For example, if media data is being sent on UDP port 5004, the control channel will be sent to the same address on UDP port 5005. All participants in a session should send compound RTCP packets and, in turn, will receive the compound RTCP packets sent by all other participants. Note that feedback is sent to all participants in a multiparty session: either unicast to a translator, which then redistributes the data, or directly via multicast. The peer-to-peer nature of RTCP gives each participant in a session knowledge of all other participants: their presence, reception quality, and optionally personal details such as name, e-mail address, location, and phone number.

Five types of RTCP packets are defined in the RTP specification: receiver report (RR), sender report (SR), source description (SDES), membership management (BYE), and application-defined (APP).

One of the primary uses of RTCP is reception quality reporting, which is accomplished through RTCP receiver report (RR) packets, which are sent by all participants who receive data. A receiver report packet is identified by a packet type of 201 and has the format illustrated in figure 3.2.

A receiver report packet contains the SSRC (synchronization source) of the participant who is sending the report (the reporter SSRC) followed by zero or more report blocks, denoted by the RC field.

Each report block describes the reception quality of a single synchronization source from which the reporter has received RTP packets during the current reporting interval. A total of 31 report blocks can be in each RTCP RR packet. If there are more than 31 active senders, the receiver should send multiple RR packets in a compound packet. Each report block has seven fields, for a total of 24 octets.

The reportee SSRC identifies the participant to whom this report block pertains. The statistics in the report block denote the quality of reception for the reportee synchronization source, as received at the participant generating the RR packet.

The cumulative number of packets lost is a 24-bit signed integer denoting the

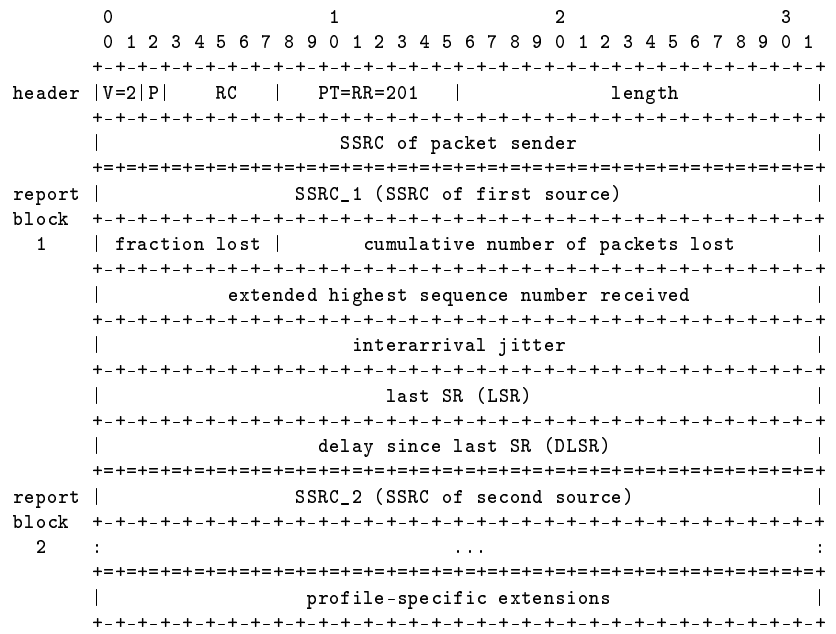


Figure 3.2: RTCP Receiver Report

number of packets expected, less the number of packets actually received. The number of packets expected is defined to be the extended last sequence number received, less the initial sequence number received. The number of packets received includes any that are late or duplicated, and hence may be greater than the number expected, so the cumulative number of packets lost may be negative. The cumulative number of packets lost is calculated for the entire duration of the session, not per interval. This field saturates at the maximum positive value of 0x7FFFFF if more packets than that are lost during the session.

The loss fraction is defined as the number of packets lost in this reporting interval, divided by the number expected. The loss fraction is expressed as a fixed-point number with the binary point at the left edge of the field, which is equivalent to the integer part after multiplying the loss fraction by 256 (that is, if 1/4 of the packets were lost, the loss fraction would be 1/4 x 256 = 64). If the number of packets received is greater than the number expected, because of the presence of duplicates, making the number of packets lost negative, then the loss fraction is set to zero.

The interarrival jitter is an estimate of the statistical variance in network transit time for the data packets sent by the reportee synchronization source. Interarrival jitter is measured in timestamp units, so it is expressed as a 32-bit unsigned integer,

like the RTP timestamp.

To calculate the variance in network transit time, it is necessary to measure the transit time. Because sender and receiver typically do not have synchronized clocks, however, it is not possible to measure the absolute transit time. Instead the relative transit time is calculated as the difference between a packet's RTP timestamp and the receiver's RTP clock at the time of arrival, measured in the same units. This calculation requires the receiver to maintain a clock for each source, running at the same nominal rate as the media clock for that source, from which to derive these relative timestamps. (This clock may be the receiver's local playout clock, if that runs at the same rate as the source clocks.) Because of the lack of synchronization between the clocks of sender and receiver, the relative transit time includes an unknown constant offset. This is not a problem, because we are interested only in the variation in transit time: the difference in spacing between two packets at the receiver versus the spacing when they left the sender. In the following computation the constant offset due to unsynchronized clocks is accounted for by the subtraction.

If S_i is the RTP timestamp from packet i, and R_i is the time of arrival in RTP timestamp units for packet i, then the relative transit time is $(R_i - S_i)$, and for two packets, i and j, the difference in relative transit time may be expressed as:

$$D(i,j) = (R_j - S_j) - (R_i - S_i).$$

The interarrival jitter is calculated as each data packet is received, using the difference in relative transit times $D(i,j)$ for that packet and the previous packet received (which is not necessarily the previous packet in sequence number order). The jitter is maintained as a moving average, according to the following formula:

$$J_i = J_{i-1} + \frac{(|D(i-1,i)| - J_{i-1})}{16}.$$

Whenever a reception report is generated, the current value of J_i for the reportee SSRC is included as the interarrival jitter. The last sender report (LSR) timestamp is the middle 32 bits out of the 64-bit NTP (Network Time Protocol) format timestamp included in the most recent RTCP SR packet received from the reportee SSRC. If no SR has been received yet, the field is set to zero. The delay since last sender report (DLSR) is the delay, expressed in units of 1/65.536 seconds, between receiving the last SR packet from the reportee SSRC and sending this reception report block. If no SR packet has been received from the reportee SSRC, the DLSR field is set to zero.

The reception quality feedback in RR packets is useful not only for the sender, but also for other participants and third-party monitoring tools. The feedback provided in RR packets can allow the sender to adapt its transmissions according to the feedback. In addition, other participants can determine whether problems

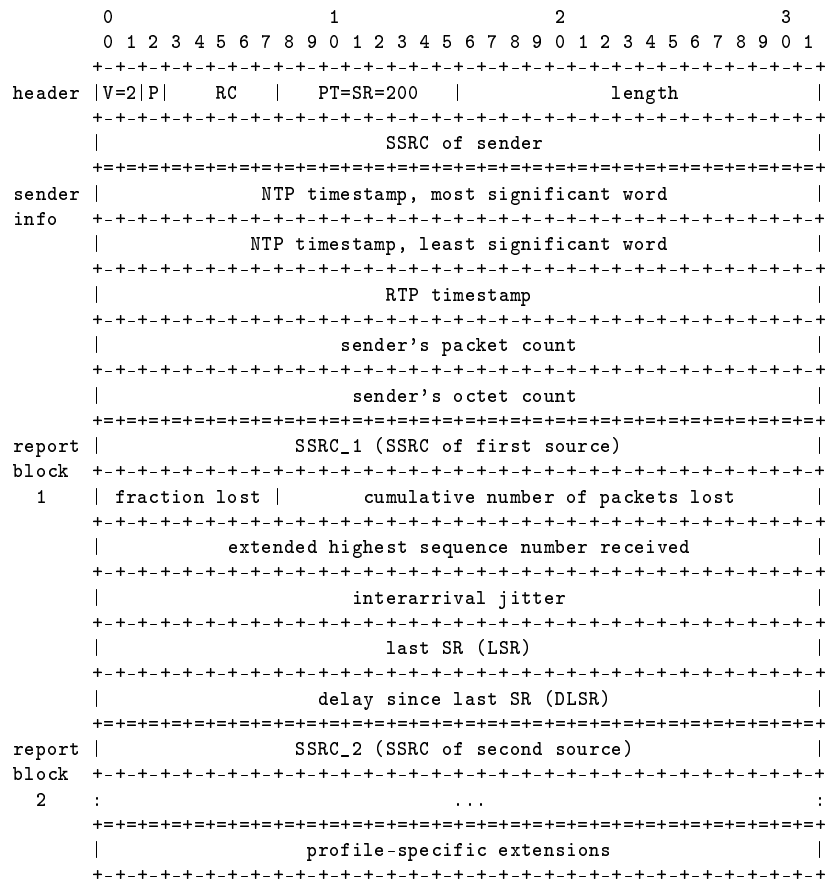


Figure 3.3: RTCP Sender Report

are local or common to several receivers, and network managers may use monitors that receive only the RTCP packets to evaluate the performance of their networks.

In addition to reception quality reports from receivers, RTCP conveys sender report (SR) packets sent by participants that have recently sent data. These provide information on the media being sent, primarily so that receivers can synchronize multiple media streams (for example, to lip-sync audio and video). A sender report packet is identified by a packet type of 200 and has the format illustrated in figure 3.3.

The payload contains a 24-octet sender information block followed by zero or more receiver report blocks, denoted by the RC field, exactly as if this were a receiver report packet. Receiver report blocks are present when the sender is also a receiver. The NTP timestamp is a 64-bit unsigned value that indicates the time

at which this RTCP SR packet was sent. It is in the format of an NTP timestamp, counting seconds since January 1, 1900, in the upper 32 bits, with the lower 32 bits representing fractions of a second (that is, a 64-bit fixed-point value, with the binary point after 32 bits). To convert a UNIX timestamp (seconds since 1970) to NTP time, 2.208.988.800 seconds have to be added.

From the SR information, an application can calculate the average payload data rate and the average packet rate over an interval without receiving the data. The ratio of the two is the average payload size. If it can be assumed that packet loss is independent of packet size, the number of packets received by a particular receiver, multiplied by the average payload size (or the corresponding packet size), gives the apparent throughput available to that receiver. The timestamps are used to generate a correspondence between media clocks and a known external reference (the NTP format clock). This makes e.g. lip synchronization possible.

As noted earlier, RTCP packets are never sent individually, but rather are packed into a compound packet for transmission. Various rules govern the structure of compound packets, as detailed next.

If the participant generating the compound RTCP packet is an active data sender, the compound must start with an RTCP SR packet. Otherwise it must start with an RTCP RR packet. This is true even if no data has been sent or received, in which case the SR/RR packet contains no receiver report blocks (the RC header field is zero). On the other hand, if data is received from many sources and there are too many reports to fit into a single SR/RR packet, the compound should begin with an SR/RR packet followed by several RR packets.

Following the SR/RR packet is an SDES packet. This packet must include a CNAME item, and it may include other items. The frequency of inclusion of the other (non-CNAME) SDES items is determined by the RTP profile in use. For example, the audio/video profile [Sch96] specifies that other items may be included with every third compound RTCP packet sent, with a NAME item being sent seven out of eight times within that slot and the remaining SDES cyclically taking up the eighth slot. Other profiles may specify different choices.

BYE packets, when ready for transmission, must be placed as the last packet in a compound. Other RTCP packets to be sent may be included in any order. These strict ordering rules are intended to make packet validation easier because it is highly unlikely that a misdirected packet will meet these constraints.

A potentially difficult issue in the generation of compound RTCP packets is how to handle sessions with larger numbers of active senders. If there are more than 31 active senders, it is necessary to include additional RR packets within the compound. This may be repeated as often as is required, up to the maximum transmission unit (MTU) of the network. If there are so many senders that the receiver reports cannot all fit within the MTU, the receiver reports for some senders must be omitted. In that case, reports that are omitted should be included in the next compound packet

generated (requiring a receiver to keep track of the sources reported on in each interval).

A similar issue arises when the SDES items to be included within the packet exceed the maximum packet size. The trade-off between including additional receiver reports and including source description information is left to the implementation. There is no single correct solution.

The rate at which each participant sends RTCP packets is not fixed but varies according to the size of the session and the format of the media stream. The aim is to restrict the total amount of RTCP traffic to a fixed fraction - usually 5% - of the session bandwidth. This goal is achieved by a reduction in the rate at which each participant sends RTCP packets as the size of the session increases. In a two-party telephone call using RTP, each participant will send an RTCP report every few seconds; in a session with thousands of participants - for example, an Internet radio station - the interval between RTCP reports from each listener may be many minutes.

Compound RTCP packets are sent periodically, according to a randomized timer. The average time each participant waits between sending RTCP packets is known as the reporting interval. It is calculated on the basis of several factors:

- **The bandwidth allocated to RTCP**. This is a fixed fraction - usually 5% - of the session bandwidth. The session bandwidth is the expected data rate for the session; typically this is the bit rate of a single stream of audio or video data, multiplied by the typical number of simultaneous senders. The session bandwidth is fixed for the duration of a session, and supplied as a configuration parameter to the RTP application when it starts.

- **The average size of RTCP packets sent and received**. The average size includes not just the RTCP data, but also the UDP and IP header sizes (that is, add 28 octets per packet for a typical IPv4 implementation).

- **The total number of participants and the fraction of those participants who are senders**. This requires an implementation to maintain a database of all participants, noting whether they are senders (that is, if RTP data packets or RTCP SR packets have been received from them) or receivers (if only RTCP RR, SDES, or APP packets have been received).

If the number of senders is greater than zero but less than one-quarter of the total number of participants, the reporting interval depends on whether we are sending. If we are sending, the reporting interval is set to the number of senders multiplied by the average size of RTCP packets, divided by 25% of the desired RTCP bandwidth. If we are not sending, the reporting interval is set to the number

of receivers multiplied by the average size of RTCP packets, divided by 75% of the desired RTCP bandwidth:

```
if ((senders > 0) and (senders < (25% of total number of participants)) {
   if (we are sending) {
     Interval = average RTCP size * senders / (25% of RTCP bandwidth)
    } else {
      Interval = average RTCP size * receivers / (75% of RTCP bandwidth)
    }
  }
```

If there are no senders, or if more than one-quarter of the members are senders, the reporting interval is calculated as the average size of the RTCP packets multiplied by the total number of members, divided by the desired RTCP bandwidth:

```
if ((senders = 0) or (senders > (25% of total number of participants)) {
   Interval = average RTCP size * total number of members / RTCP bandwidth
}
```

These rules ensure that senders have a significant fraction of the RTCP bandwidth, sharing at least one-quarter of the total RTCP bandwidth. The RTCP packets required for lip synchronization and identification of senders can therefore be sent comparatively quickly, while still allowing reports from receivers. The resulting interval is always compared to an absolute minimum value, which by default is chosen to be 5 seconds. If the interval is less than the minimum interval, it is set to the minimum. In some cases it is desirable to send RTCP more often than the default minimum interval. For example, if the data rate is high and the application demands more timely reception quality statistics, a short default interval will be required. The latest revision of the RTP specification allows for a reduced minimum interval in these cases

```
Minimum interval = 360 / (session bandwidth in Kbps).
```

This reduced minimum is smaller than 5 seconds for session bandwidths greater than 72 Kbps. When the reduced minimum is being used, it is important to remember that some participants may still be using the default value of 5 seconds, and to take this into account when determining whether to time out a participant because of inactivity. The resulting interval is the average time between RTCP packets. The transmission rules described next are then used to convert this value into the actual send time for each packet. The reporting interval should be recalculated whenever the number of participants in a session changes, or when the fraction of senders changes.

When an application starts, the first RTCP packet is scheduled for transmission on the basis of an initial estimate of the reporting interval. When the first packet is sent, the second packet is scheduled, and so on. The actual time between packets

is randomized, between one-half and one and a half times the reporting interval, to avoid synchronization of the participants' reports, which could cause them to arrive all at once, every time. Finally, if this is the first RTCP packet sent, the interval is halved to provide faster feedback that a new member has joined, thereby allowing the next send time to be calculated as shown here:

```
I = (Interval * random[0.5, 1.5])

if (this is the first RTCP packet we are sending) {
   I *= 0.5
}

next_rtcp_send_time = current_time + I
```

The routine `random[0.5, 1.5]` generates a random number in the interval 0.5 to 1.5.

The proposed standard version of RTP [SCFJ96] uses only these basic transmission rules. Although these are sufficient for many applications, they have some limitations that cause problems in sessions with rapid changes in membership. The concept of reconsideration was introduced to avoid these problems [Per03].

3.3.2 QoS

The main goal of quality of service (QoS) is to help reduce or eliminate delay of voice packets including packet loss that travels across a network. It can be defined as the capability of a network to provide better service to selected network traffic over various underlying technologies such as Frame Relay, ATM, and IP. This network feature helps in differentiating different classes of traffic and treats them differently. The various formal measurements of QoS are:

- *Service Availability* - The availability of the users' network connection and depends on the connected network device
- *Throughput* - the packet delivery rate at the endpoints
- *Delay* - the end-to-end packet delay, while traversing the network
- *Delay Jitter* - the delay variation among similar packets traversing the same path in the network
- *Packet loss rate* - the rate of packet loss, because of packet dropping and corruption

QoS of networks should attempt to maximize service availability and throughput, while at the same time minimizing the remaining measurements.

In the current network scenario, different types of traffic (such as real-time and data) need to share the same network link. These different kinds of traffic require different treatment from the network. This is similar to a first class passenger of an airplane demanding preferential treatment over other passengers. Just as, apart from providing special treatment, a separate airplane cannot be made available to the first class passenger, similarly a separate link or a network connection cannot be given to different customers, though the treatment given can be different. The entire bandwidth has to be shared between priority traffic and regular traffic, and only at places where the traffic flows through active network elements like routers, can these flows be differentiated and treated differently.

The different types of traffic can be grouped as:

- *Non Real-time or Data*: these applications only care about reliable packet delivery like that guaranteed in TCP. They are immune to delays or bandwidth requirements. Examples are web browsing, email, distributed computing, etc.
- *Real-time*: these applications require timely delivery along with reliability. Some of these applications can tolerate an upper bound in delay, whereas others are totally intolerant. This range of expectations is used to further classify real-time applications by QoS models

There are three major architectures or service models for implementing QoS in packet networks. These are:

- *Best Effort Service* - in this model there are no QoS guarantees given to the application, except that a best effort to deliver will be made
- *Integrated Service* - it is also called IntServ. This is an earlier model (1995) devised for integrated service networks such as ISDN or ATM. It believed to have end-to-end QoS strategies implemented at the network elements for all classes of traffic flows
- *Differentiated Service* - also called DiffServ. This model was developed later (around 1998), wherein unlike the IntServ, it implements QoS strategies on per hop basis as Per Hop Behavior (PHB), and avoids signaling mechanisms. It tries to differentiate individual packets and responds with a different behavior to the same

Best Effort Service This model allows the application to send any amount of data at will and without any authorization. The network elements in turn will try their best to deliver the packets to their destinations without any constraints of maximum delay, latency or jitter, etc. They also give up trying to deliver after attempting

for a number of times, in case acknowledgments from the receiver are not received. The network will also not inform the sender that the attempt to deliver has been abandoned. Thus, the end points have to take care and incorporate reliability features within themselves. The IP network is an example of a Best Effort service model.

Integrated Service The Integrated service model is defined by a set of standards laid down by IETF. This model assures different QoS-profile treatment, dictated by the network elements, to the multiple classes of traffic. In this model, the applications are aware of the traffic characteristics that they would put on the network and accordingly signal the network elements to reserve required resources before sending their data. The network elements in turn acknowledge the signal positively if they are able to reserve the resources or else send a negative acknowledgement.

This model categorizes applications, in terms of their network traffic requirements, into three classes. Real-time Tolerant (RTT), where some delays can be tolerated within a small range as in video or audio playback applications. Real-time Intolerant (RTI), which requires minimal or absolutely no delays, as in video conferencing. And finally, Elastic applications, where as long as the packets are delivered reliably no delay constraints are imposed, as in web browsing or email. Accordingly, the model provides for the following classes of service:

- *Guaranteed service*: this service guarantees bandwidth and provides a deterministic upper bound on delay. It is used for RTI applications
- *Controlled Load service*: this service guarantees an average delay, but the specific end-to-end delay by some arbitrary packet cannot be specifically determined

The Integrated service network implements the following mechanisms to guarantee QoS levels:

- *Admission Control* - this allows the network (network elements) to refuse a new traffic flow request from an application, depending upon resource availability. This is usually a policy-based decision taken by the router. Policy Enforcement Points (PEP) and Policy Decision Points (PDP) are components of this mechanism. PEPs and PDPs use a simple request response protocol called Common Open Policy Service (COPS) to communicate between themselves
- *Traffic Shaping and Policing* - traffic shaping ensures that the traffic entering the network conforms to the agreed flow characteristic. An example of traffic shaping is the metered expressways in the US, where each vehicle is made to

stop and wait for a green light at the ramp before entering the expressway. The frequency of entry is governed by the congestion on the expressway. Traffic policing ensures that applications are sending data into the network as per the agreed traffic QoS profile

- *Congestion Management* - this is implemented in core routers in which different queuing techniques are used to create and manage different queues for different traffic; algorithms to help classify packets belonging to the various queues are enforced and finally, queued packets are scheduled for transmission. Some of the major types of queuing techniques used by routers are the First In First Out (FIFO) queue and Weighted Fair Queuing (WFQ)

- *Congestion Avoidance* - unlike congestion management, which deals with the post-congestion situation, congestion avoidance implements strategies to anticipate and avoid congestion. Some strategies, popularly used for congestion avoidance are Tail Drop, Random Early Dropping (RED) and Weighted Random Early Dropping (WRED). All these strategies determine when to drop packets at an active node (router) in anticipation of congestion in the network

- *Link Efficiency* - this mechanism improves link efficiency by helping to deal with situations wherein jumbograms (large packets from FTP-like applications) with lower priority congest and prevent smaller, but higher priority packets, to go through. It is implemented by router vendors as Link Fragmentation and Interleaving (LFI) strategies. An example of LFI usage is the Multilink Point to Point protocol (MLP - RFC 1717 [SLMC94]), wherein datagrams are split, sequenced and recombined over multiple links. MLP uses LFI to break jumbograms into smaller packets and interleave them with smaller packets of higher priority. LFI adds multilink headers to the datagrams to ensure correct transmission and reassembly. Compressed Real-time Protocol Header (CPTR) is another link efficiency improvement mechanism for real-time traffic. CPTR compresses the header of an RTP packet from 40 bytes to 2-5 bytes before transmission

- *Flow-wise Soft state management* - the QoS-capable active network elements need to maintain the state information of all traffic flows. This information is, however, maintained for a limited time (soft) or until explicitly unreserved, unless it is refreshed by the communicating endpoints. RSVP requests using PATH and RESV have to be periodically sent by the endpoints to reserve the allocated resources

Resource Reservation Protocol (RSVP) RSVP protocol specified by IETF in RFC 2205 [BZB+97] helps in providing quality of service in networks. It can pri-

oritize traffic and helps in giving latency guarantees to specific IP traffic streams. Using RSVP, a packet-switched network can be made to give a more deterministic quality of service as in a circuit-switched network. RSVP provides mechanisms or requests to reserve resources at each node along the data path. RSVP makes a distinction between the sender and receiver meaning that requests can be sent/applied in only one direction. An application process would normally have both a sender and receiver component. However, because of the above distinction, it is always the receiver component that makes the QoS request to all the nodes in the reverse path. Although, RSVP is not a routing protocol, it is designed to work with both unicast and multicast routing protocols. As the receiver is responsible for making the RSVP request, it allows larger groups with heterogeneous receiver capabilities/requirements to be accommodated. Dynamic group membership is also possible because of this characteristic.

RSVP has the following essential attributes:

- Maintains soft state in routers and host, enabling graceful support for dynamic membership changes
- Is Receiver-oriented
- Provides unicast and multicast support
- Provides Transparent operation through non-RSVP routers

Differentiated Service This model classifies and conditions network traffic at the entry to a network into different behavior aggregates. Each such behavior aggregate is assigned a DS code point. This differentiated service code point is defined by markups using DF bits (packet fragmentation allowed/not allowed). Packets at the core routers are given differential treatment while forwarding, based on Per Hop Behaviors (PHB) that are in turn based on the above DS code points. DiffServ breaks the whole network into DiffServ domains. A domain is a continuous set of nodes, which supports a common resource provisioning and PHB policy. A DS domain is usually controlled by a single entity, such as an ISP or an organization's Intranet. DiffServ is extended across domains by Service Level Agreements (SLAs) between domains. SLAs specify rules, such as those for traffic, and also remarks, such as actions, for out of profile traffic. Traffic Conditioning Agreements (TCAs) are derived from these SLAs. The domain has a well-defined boundary with two types of nodes, as mentioned below:

- *Boundary nodes*: boundary nodes are located at the boundary of the DiffServ cloud with other domains. These nodes classify and appropriately mark incoming ingress traffic, so that the packets can be forwarded as per the PHB

groups supported within the domain. They also enforce TCAs between their own domains and other domains

- *Interior nodes*: interior nodes are connected to other interior nodes, or to boundary nodes, but they remain within the boundary

A DS node can be the ingress or egress node, depending upon the traffic flow. Traffic enters the DS cloud through an ingress node, and leaves through the egress node. The ingress node is responsible for enforcing the TCA with the sender's domain and the egress node shapes the outgoing traffic in compliance with the TCA of the receiver's domain.

The DiffServ minimizes the signaling requirement by using aggregation and PHB. Flows are classified by predetermined rules, so that they fit into a limited set of class flows. This helps in easing congestion in the backbone.

Edge routers use the 8 bit ToS field of the packet header (also called the DS field in DiffServ domain) to mark the packet for preferential treatment by the core routers. The last 6 bits of the ToS field are used for the DS code and the remaining 2 are reserved for future use. Only the edge routers are required to maintain the per-flow states and perform the shaping and policing. Traffic shaping and policing are computation intensive and as the edge routers are usually placed next to slower access links, they are best suited to perform the same. Whereas inside the core network, packets need to be routed very fast and hence, minimum computation is desirable at the core routers and switches. Per Hop Behaviors is the description of externally observable forwarding behavior demonstrated by a DS node (routers). PHBs can be defined in terms of their resources (buffer and bandwidth) or in terms of their properties (delay and packet loss), or in terms of their priorities relative to other PHBs. PHBs are implemented using buffer and scheduling mechanisms. Multiple PHBs are aggregated together as PHB groups. Given below is the description of two popular PHBs in use.

- *Expedited Forwarding(EF PHB)* - EF PHB is used to provide premium service to the customer. It is a low-delay, low jitter service providing near constant bit rate. Its SLA specifies a peak bit rate which customer applications will receive. It is the customer's responsibility not to exceed the rate. Packets are dropped on exceeding this rate. EF PHB is defined as a forwarding treatment for a particular DiffServ aggregate, where the departure rate of the aggregate's packets from any DiffServ node must equal or exceed a configurable rate

- *Assured Forwarding (AF PHB)* - AF PHB is used to provide assured service to a customer, meaning that he/she will get reliable service even in times of network congestion. The customer will be provided with a fixed bandwidth at all times, as per the SLA

This identifies the subset of network traffic which may receive a differentiated service by being conditioned and/or mapped to one or more behavior aggregates. Packet Classifiers use information in the packet header to select appropriate classes. The two types of classifiers are as follows:

- *Behavioral Aggregate Classifiers* - which select packets on the basis of DS codepoints
- *Multi Field Classifiers* - which select packets based on values of multiple header fields

Traffic conditioning ensures that the traffic entering a DS domain at any point complies with the TCA, between the sender's and receiver's domains and the domain's service provisioning policy. It involves traffic shaping, metering, policing and/or remarking.

The conditioner receives the packets from the classifier and uses a *meter* to measure the temporal properties of the stream against the appropriate traffic profile from the TCA. This information is passed along with the packet. Further processing is done by markers, shapers and policers based on whether the packet is in or out of profile. The *marker* marks or remarks a packet with a DS value corresponding to a correct PHB codepoint. The marker may be configured according to policies. The *shaper* buffers the traffic stream and increases the delay of a stream to make it compliant with a particular traffic profile. Packets may be discarded if the buffer is full. *Droppers* drop packets of a stream to make it compliant with a particular traffic profile. Droppers can be considered as special cases of shapers with buffer size set to zero.

Multi Protocol Label Switching (MPLS) MPLS is a hybrid technology model, which enables very fast forwarding at the core and slower conventional routing at the edges of a network. It combines the best of ATM's circuit-switching and IP's packet-routing. Packets are assigned a label at the entry to a MPLS domain and are switched inside the domain by a simple look-up table. The MPLS domain is usually the core backbone of a provider's network. These labels determine the quality of service rendered to the packet. At the egress router at the domain's edge, the packets are stripped off their labels and are routed in a conventional manner to their destination.

In MPLS, packets are mapped to Forwarding Equivalence Classes (FECs) only once at the ingress router, and the FEC's corresponding *label* is assigned to the packet and is sent along with it. This label is of fixed length and is only locally significant. At later hops at routers within the MPLS domain,this label is used as an index into the routing tables to determine the next hop, as well as the new value

for the label. Unlike conventional IP routing, there is no complex processing of the packet header involved in MPLS. Compare this to conventional IP routing, where the next hop is determined on the basis of the packet's header, and also by running a network layer routing algorithm. The conventional routing is done using two main functions, partitioning the complete set of possible packets into Forwarding Equivalence classes and mapping each FEC to the next hop. This mapping of packets to FECs is done at every router in conventional IP routing and is a costly operation.

MPLS is multi-protocol because this label assignment and label based switching can be used over any underlying network protocol. An MPLS based router is also called Label-Switched Router (LSR), and the path taken by a packet, after being switched by LSRs through an MPLS domain, is called Label-Switched Path (LSP).

A labeled packet usually carries multiple labels organized as a Last In First Out (LIFO) label stack. These labels are present as encapsulation or as markup inside the packet header. At a router, forwarding decisions are always based in the stack's topmost label, independent of the underlying labels. This label stack is useful to implement tunneling and hierarchy.

The advantages of MPLS over conventional routing are evident in some situations, as given here:

- MPLS forwarding can be done by ASIC-based switches, because only a simple table lookup and label replacement is involved. A computation-intensive large prefix search, as in conventional routing, is gainfully avoided
- Incoming packets from different ports, or typified by any other header independent criteria, can be distinguished by the ingress router by assigning them to different FECs. This scores over conventional IP routing, where only the header information can be used for making routing decisions
- MPLS allows packets to be differentiated on the basis of the ingress router used for entering the MPLS domain. This is difficult in IP routing as the intermediate router addresses are not included in the packet's header
- Since the mapping of labels to FECs at the ingress LSRs is a one time activity, these mapping algorithms can be made as complex as desired
- Unlike conventional IP-based source routing, where an extra set of address information is carried, the MPLS needs to only carry a label to specify fixed paths

Constraint-based Routing This is a type of QoS-based routing, in which the viability of a route with respect to meeting specific QoS requirements and also meeting

other network constraints, like policy, is determined. This type of routing has two major objectives:

- Select routes that meet QoS requirements
- Increase network utilization and load distribution. A longer and less congested path may be better for QoS-demanding traffic as compared to the shortest, highly congested path.

In this model, routers are required to exchange various kinds of link state information and dynamically compute routes based on this information. To distribute link state information, most implementers of this model extend the link state information contained in the advertisements of OSPF. However, this increases congestion, because of the need for sending frequent advertisements. This is overcome to an extent by transmitting, only when a significant change to network parameters has occurred, e.g. a sharp fall in network bandwidth, etc. The algorithm to calculate routing tables is based on hop count and bandwidth. In constraint-based routing, the routing tables have to be computed more frequently than in dynamic routing, as the computations can be easily triggered by a number of factors such as bandwidth changes, congestion, etc.

Although constraint-based routing does a better job of meeting QoS requirements and provides better network utilization, its major disadvantage is the high computation overhead and large sized routing tables. Also, the selected long paths may consume more resources than a shorter path. In addition, the routes may be unstable and are transient most of the time, as the routing tables are being updated too frequently, which may also lead to race conditions.

3.3.3 TCP over Wireless

Reliable transport protocols such as TCP are tuned to perform well in traditional networks where packet losses occur mostly because of congestion. However, networks with wireless and other lossy links also suffer from significant losses due to bit errors and handoffs. TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in wireless and lossy systems.

The TCP sender uses the cumulative acknowledgments it receives to determine which packets have reached the receiver, and provides reliability by retransmitting lost packets. For this purpose, it maintains a running average of the estimated round-trip delay and the mean linear deviation from it. The sender identifies the loss of a packet either by the arrival of several duplicate cumulative acknowledgments or the absence of an acknowledgment for the packet within a timeout interval

equal to the sum of the smoothed round-trip delay and four times its mean deviation. TCP reacts to packet losses by dropping its transmission (congestion) window size before retransmitting packets, initiating congestion control or avoidance mechanisms (e.g., slow start [Jac88]) and backing off its retransmission timer (Karn's Algorithm [KC91]). These measures result in a reduction in the load on the intermediate links, thereby controlling the congestion in the network. Unfortunately, when packets are lost in networks for reasons other than congestion, these measures result in an unnecessary reduction in end-to-end throughput and hence, sub-optimal performance. Communication over wireless links is often characterized by sporadic high bit-error rates, and intermittent connectivity due to handoffs. TCP performance in such networks suffers from significant throughput degradation and very high interactive delays [CI95].

Several schemes have been proposed to the alleviate the effects of non-congestion-related losses on TCP performance over networks that have wireless or similar high-loss links [BB95, BSK95, YB94]. These schemes choose from a variety of mechanisms, such as local retransmissions, split-TCP connections, and forward error correction, to improve end-to-end throughput.

There are two different approaches to improve TCP performance in such lossy systems. The first approach hides any non-congestion-related losses from the TCP sender and therefore requires no changes to existing sender implementations. The intuition behind this approach is that since the problem is local, it should be solved locally, and that the transport layer need not be aware of the characteristics of the individual links. Protocols that adopt this approach attempt to make the lossy link appear as a higher quality link with a reduced effective bandwidth. As a result, most of the losses seen by the TCP sender are caused by congestion. Examples of this approach include wireless links with reliable link-layer protocols such as AIRMAIL [APL+95], split connection approaches such as Indirect-TCP [BB95], and TCP-aware link-layer schemes such as the snoop protocol [BSK95]. The second class of techniques attempts to make the sender aware of the existence of wireless hops and realize that some packet losses are not due to congestion. The sender can then avoid invoking congestion control algorithms when non-congestion-related losses occur. Finally, it is possible for a wireless-aware transport protocol to coexist with link-layer schemes to achieve good performance.

Link-layer Protocols There have been several proposals for reliable link-layer protocols. The two main classes of techniques employed by these protocols are: error correction, using techniques such as forward error correction (FEC), and retransmission of lost packets in response to automatic repeat request (ARQ) messages. The link-layer protocols for the digital cellular systems in the U.S. - both CDMA and TDMA - primarily use ARQ techniques. While the TDMA protocol guarantees

reliable, in-order delivery of link-layer frames, the CDMA protocol only makes a limited attempt and leaves eventual error recovery to the (reliable) transport layer. Other protocols like the AIRMAIL protocol [APL+95] employ a combination of FEC and ARQ techniques for loss recovery. The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols. The link-layer protocol operates independently of higher-layer protocols and does not maintain any per-connection state. The main concern about link-layer protocols is the possibility of adverse effect on certain transport-layer protocols such as TCP.

Split Connection Protocols Split connection protocols split each TCP connection between a sender and receiver into two separate connections at the base station - one TCP connection between the sender and the base station, and the other between the base station and the receiver. Over the wireless hop, a specialized protocol tuned to the wireless environment may be used. In [YB94], the authors propose two protocols - one in which the wireless hop uses TCP, and another in which the wireless hop uses a selective repeat protocol (SRP) on top of UDP. They study the impact of handoffs on performance and conclude that they obtain no significant advantage by using SRP instead of TCP over the wireless connection in their experiments. Indirect-TCP [BB95] is a split-connection solution that uses standard TCP for its connection over the wireless link - see figure 3.4. Like other split-connection proposals, it attempts to separate loss recovery over the wireless link from that across the wireline network, thereby shielding the original TCP sender from the wireless link. Since TCP is not well-tuned for the lossy link, the TCP sender of the wireless connection often times out, causing the original sender to stall. In addition, every packet incurs the overhead of going through TCP protocol processing twice at the base station (as compared to zero times for a non-split-connection approach), although extra copies are avoided by an efficient kernel implementation. Another disadvantage of split connections is that the end-to-end semantics of TCP acknowledgments is violated, since acknowledgments to packets can now reach the source even before the packets actually reach the mobile host. Also, since split-connection protocols maintain a significant amount of state at the base station per TCP connection, handoff procedures tend to be complicated and slow.

The Snoop Protocol The snoop protocol introduces a module, called the snoop agent, at the base station - see figure 3.5. The agent monitors every packet that passes through the TCP connection in both directions and maintains a cache of TCP segments sent across the link that have not yet been acknowledged by the receiver. A packet loss is detected by the arrival of a small number of duplicate acknowledgments from the receiver or by a local timeout. The snoop agent retransmits the

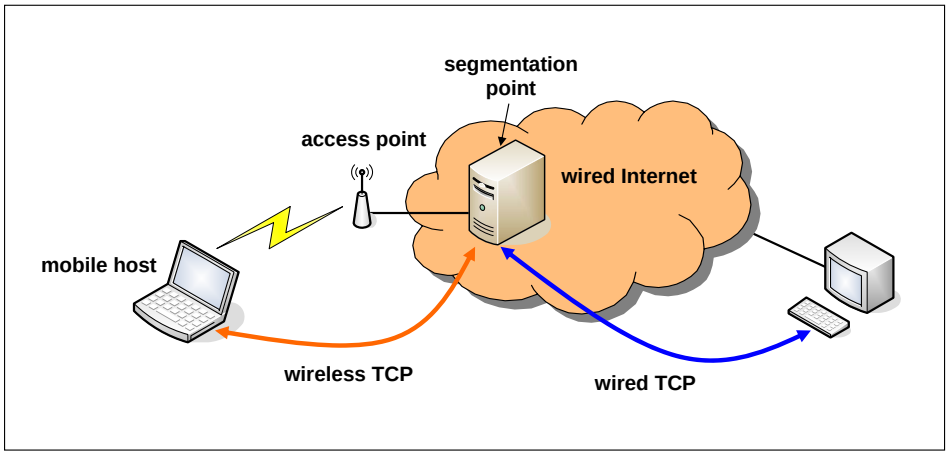


Figure 3.4: Indirect-TCP

lost packet if it has it cached and suppresses the duplicate acknowledgments. In our classification of the protocols, the snoop protocol is a link-layer protocol that takes advantage of the knowledge of the higher-layer transport protocol (TCP). The main advantage of this approach is that it suppresses duplicate acknowledgments for TCP segments lost and retransmitted locally, thereby avoiding unnecessary fast retransmissions and congestion control invocations by the sender. The per-connection state maintained by the snoop agent at the base station is soft, and is not essential for correctness. Like other link-layer solutions, the snoop approach could also suffer from not being able to completely shield the sender from wireless losses.

Selective Acknowledgments Since standard TCP uses a cumulative acknowledgment scheme, it often does not provide the sender with sufficient information to recover quickly from multiple packet losses within a single transmission window. Several studies [FF96] have shown that TCP enhanced with selective acknowledgments performs better than standard TCP in such situations. SACKs were added as an option to TCP by RFC 1072 [JB88]. However, disagreements over the use of SACKs prevented the specification from being adopted, and the SACK option was removed from later TCP RFCs. Recently, there has been renewed interest in adding SACKs to TCP. Two relevant proposals are the recent RFC on TCP SACKs [MMFR96] and the SMART scheme [KM97]. The SACK RFC proposes that each acknowledgment contain information about up to three non-contiguous blocks of data that have been received successfully by the receiver. Each block of data is described by its starting and ending sequence number. Due to the limited number of blocks, it is best to inform the sender about the most recent blocks received. The RFC does not specify the sender behavior, except to require that standard TCP

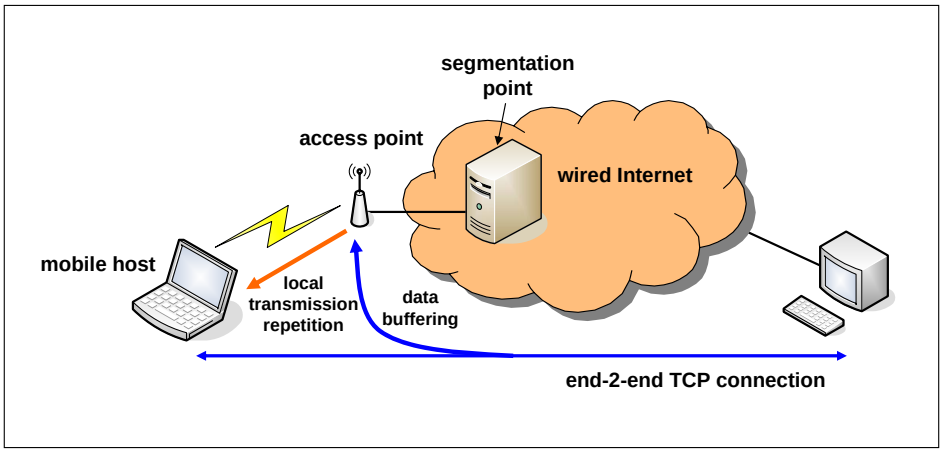


Figure 3.5: Snooping TCP

congestion control actions be performed when losses occur. An alternate proposal, SMART, uses acknowledgments that contain the cumulative acknowledgment and the sequence number of the packet that caused the receiver to generate the acknowledgment (this information is a subset of the three-blocks scheme proposed in the RFC). The sender uses this information to create a bit mask of packets that have been delivered successfully to the receiver. When the sender detects a gap in the bit mask, it immediately assumes that the missing packets have been lost without considering the possibility that they simply may have been reordered. Thus this scheme trades off some resilience to reordering and lost acknowledgments in exchange for a reduction in overhead to generate and transmit acknowledgments.

3.3.4 Protocol Header Compression

The wireless market is widely believed to be the most important future market for data services. Despite the efforts that were made to increase usability and abilities of all different wireless devices, the wireless protocol domain is still challenging. As new services and protocols emerge for wired networks, the need to incorporate those services and protocols in wireless communication systems arises. Existing wireless networks of the second generation (also known as 2G) are mostly circuit-switched and have been developed and optimized for voice transmission. Wireless networks of the 3rd generation (3G) have to support a broad range of application scenarios. 3G networks and terminals such as smart-phones, PDAs, and Laptops are marketed with new services, most importantly multimedia.

Multimedia applications comprise several different application scenarios, including audio, video, and gaming. The bandwidth needs of these applications are higher

Codec	Mean Bitrate [Kbps]	IPv4 $\bar{S}$ [%]	IPv6 $\bar{S}$ [%]
LPC	5.6	74.07	81.08
GSM	13.2	54.79	64.52
ITU-T G.711	64.0	20.00	27.27

Table 3.3: Theoretical upper bound savings (in terms of bandwidth) for voice traffic
(see formula 3.1 - page 101)

than offered in 2G systems. In addition, multimedia applications require more stringent network quality of service (QoS). One of the key issues for the QoS is the suitability for real-time applications. Therefore, delay and jitter are key considerations within the QoS domain. Multimedia applications often use RTP, UDP, and IP as protocols. Each of the protocol layers adds header overhead. Thus, the bandwidth requirements derive from the application (i.e., the payload) and the protocol overhead.

Significant compression of the packet traffic can be achieved by reducing the amount of overhead information. For LPC coded voice, for instance, the IP overhead is 81%, as detailed shortly in table 3.3. In general, for multimedia services, header compression achieves a dramatic saving in bandwidth. Given the high license fees of 3G bands and the migration of IP based services into the wireless format, it is necessary to reduce the header overhead of IP based traffic.

IP is the underlying network layer protocol used for most multimedia application scenarios. Focusing on this protocol domain thus promises the highest gains. IP header compression mechanisms have always been an important part of saving bandwidth over bandwidth limited links. Many header compression schemes exist already, but most of them are not suitable for the wireless environment. Wireless links have typically a very high and variable bit error probability (BEP) due to shadow- and multi-path fading and mobility. With a reduction of the required bandwidth, the latency and Packet Error Probability (PEP) can be improved. This is because the probability that a given packet is affected by link errors is reduced for smaller packets. For multimedia services in wireless environments ROHC was introduced. ROHC was standardized by the Internet Engineering Task Force in RFC 3095 [BBea01]. This compression scheme was designed to operate in error-prone environments by providing error detection and correction mechanisms in combination with robustness for IP based data streams. A connection oriented approach removing packet inter- and intra-dependencies yields a significant reduction of the IP header and other headers.

The motivation for IP header compression is based on the facts that (i) the multimedia payload is typically compressed at the application layer, (ii) the headers occupy a large portion of the packet for some services, and (iii) the headers have significant redundancy.

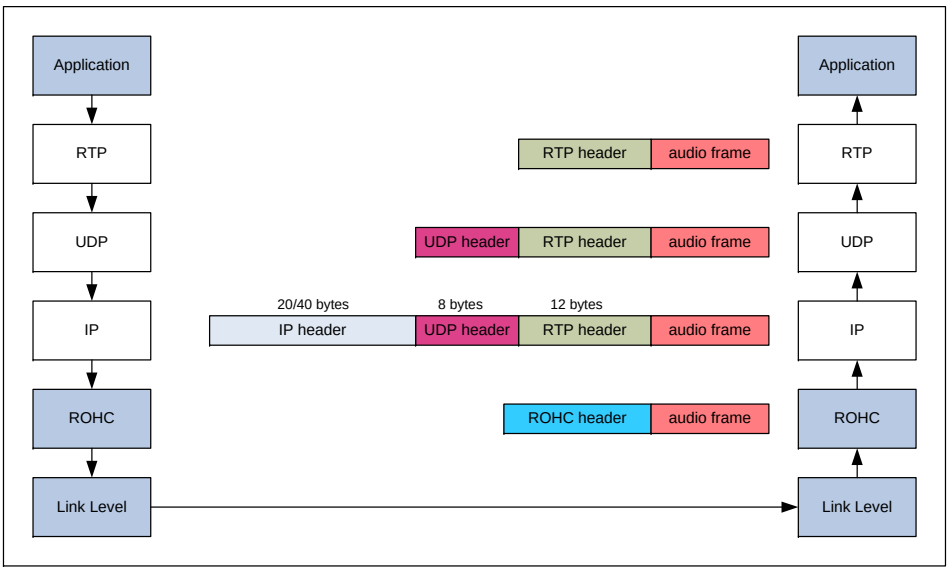


Figure 3.6: Header structure and protocol stack with relevant layers

In figure 3.6 the combined header for a real-time multimedia stream with IPv4, resulting in 40 bytes protocol overhead is illustrated. The protocol headers include the 20 byte IPv4 header, the 8 byte UDP header and the 12 byte RTP header. With IPv6, there is a total of 60 bytes of overhead. In case of GSM coded audio transmission the payload is only 33 bytes (13.2 Kbps * 20 ms) long – the header of IPv4 accounts for 55% of the packet. For IPv6 this ratio is even larger. There are some redundancies among the different headers (IP, UDP, and RTP) of a given packet, but typically there are even larger redundancies between contiguous packets of a given IP flow.

Thus, there are two types of header redundancies:

- *Intra-packet* the headers for the different protocols within a single packet carry identical or deducible information
- *Inter-packet* the headers between consecutive packets have only marginal (incremental) differences

A classification of the header fields into groups results in *non changing* and *changing*. The non changing group consists of *static*, *static-known*, and *inferred* header fields. A large portion of the header fields are static or static-known (20.5 bytes in IPv4 and 44.5 bytes in IPv6). The compression of these fields can be achieved with low to moderate complexity. In fact, these fields could be completely omitted after the first successful transmission. The Segment Length (2 bytes) and Packet

Length (2 bytes) fields (as well as the Header Checksum in IPv4 with 2 bytes) are referred to as inferred. These entries can be inferred from other header fields and are also relatively easily compressed. The changing group consists of *not-classified-a-priori*, *rarely-changing*, *static* or *semi-static* changing, and *alternating* changing header fields. These header fields are more difficult to compress and it depends on the applied header compression scheme how the compression is achieved.

To get a basic idea of the possible savings of header compression, we compute an upper bound for voice communication and for audio streaming. For this upper bound calculation, we assume that with header compression the overhead due to IP, UDP, and RTP is zero. The upper bound on the savings, denoted by S_i, for packet i is then

$$S_i = 1 - \frac{Packet(i)}{Header + Packet(i)} = \frac{Header}{Header + Packet(i)} \tag{3.1}$$

where $Packet(i)$ denotes the size of the payload data and $Header$ denotes the size of the (uncompressed) IP, UDP, and RTP headers. Clearly, the potential savings depend only on the mean packet length. The packet length depends on the service type used. The mean saving

$$\bar{S} = \frac{1}{N} \sum_{i=1}^{N} S_i$$

gives the portion of bandwidth that a wireless network provider can potentially save for a session with N packets.

We consider the Linear Predictive Coding (LPC) with 5.6 Kbps, the GSM codec with 13.2 Kbps, and a codec following the ITU-T standard G.711 with 64.0 Kbps. For the calculation it is assumed that a packet is generated every 20 ms. Therefore, S_i is the same for all IP packets (and thus equal to S). The potential savings for IPv6 are larger than for IPv4 because of the IP header length (IPv4 with 40 byte and IPv6 with 60 byte). A second indication is that smaller IP packets correspond with larger savings S. This is due to the larger ratio of the header compared to the smaller packets. Overall, there is a large potential for bandwidth savings. The savings are most significant for low-bit rate streams, which tend to be common in wireless networks.

The Internet Protocol (IP) is the choice of transport protocol both on wired and wireless networks and this choice is leading to the convergence of telecommunication and data networks. These converged networks will be the building blocks of the All-IP vision. As the networks evolve to provide more bandwidth, the applications, services and the consumers of those applications all compete for that bandwidth. For the network operators it is important to offer a high quality of service (QoS) in order to attract more customers and encourage them to use their network as much

as possible, thus providing higher average revenue per user (ARPU).

As for wireless networks with their high bit error rates (highly prone to interference) and high latency (long round trip times), it is difficult to attain those high bandwidths required. When all these factors are taken into account it means that the available resources must be used as efficiently as possible.

In many services and applications e.g., Voice over IP, interactive games, messaging etc., the payload of the IP packet is almost of the same size or even smaller than the header. Over the end-to-end connection, comprised of multiple hops, these protocol headers are extremely important but over just one link (hop-to-hop) these headers serve no useful purpose. It is possible to compress those headers, providing in many cases more than 90% savings, and thus save the bandwidth and use the expensive resources efficiently. IP header compression also provides other important benefits, such as reduction in packet loss and improved interactive response time. In short, IP header compression is the process of compressing excess protocol headers before transmitting them on a link and uncompressing them to their original state on reception at the other end of the link. It is possible to compress the protocol headers due to the redundancy in header fields of the same packet as well as consecutive packets of the same packet stream.

The IP version 4 header is 20 bytes and when carrying UDP (8 bytes) and RTP (12 bytes, at least), the packet header becomes 40 bytes. A header compression scheme usually compresses such headers to 2 - 4 bytes. On an average, considering a few uncompressed packets and a few relatively large packets, more than 80% savings can be observed. When compared with the payload being carried, in such cases as voice where payload size is usually static and in the range of 20 - 60 bytes, the header size represents a huge overhead. Using header compression in such cases results in major bandwidth savings. The IP version 6 with a header size of 40 bytes is gaining wide acceptance and has been included in Release 5 and onwards versions of 3G wireless networks. In this case, header compression will result in even more savings.

On low bandwidth networks, using header compression results in better response times due to smaller packet sizes. A small packet also reduces the probability of packet loss due to bit errors on wireless links resulting in better utilization of the radio spectrum. It has been observed that in applications such as video transmission on wireless links, when using header compression the quality does not change in spite of lower bandwidth usage. For voice transmission, the quality increases while utilizing lower bandwidth. In short, header compression improves network transmission efficiency, quality and speed with:

- Decrease in packet header overhead (bandwidth savings)
- Reduction in packet loss

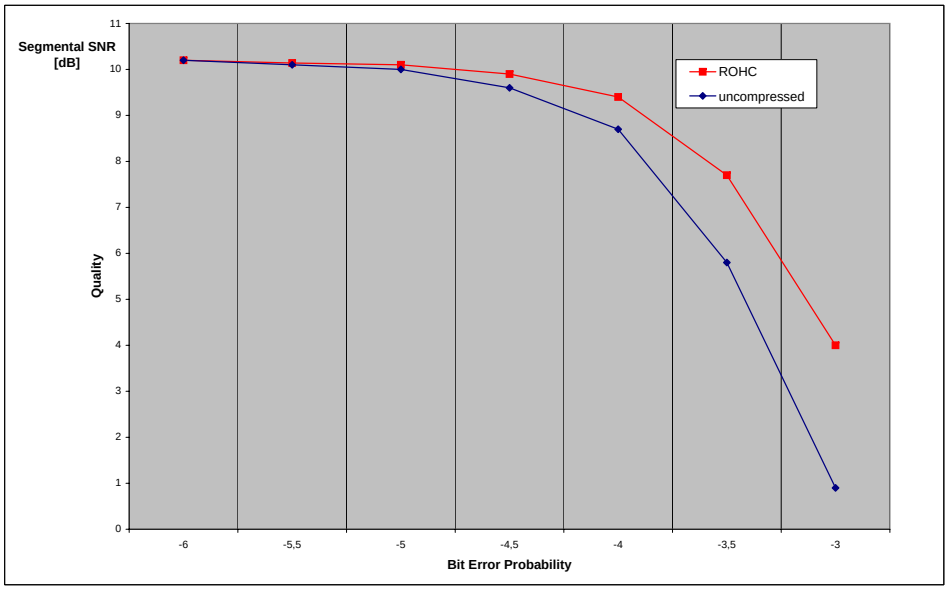


Figure 3.7: Objective voice quality on a wireless link for transmission with ROHC and without header compression

- Better interactive response time
- Decrease in infrastructure cost, more users per channel bandwidth means less infrastructure deployment costs

These benefits lead to improved QoS in the network and the possibility for operators to improve their ARPU. The operators will be able to retain and attract customers with better QoS on the network and more services and content on the links - see figure 3.7.

Objective voice quality (segmental SNR) as a function of bit error probability on a wireless link for transmission with ROHC and without header compression: ROHC improves the voice quality for moderate to large bit error probabilities, while using roughly 46% less bandwidth than transmission without header compression [FHSR03].

The IP protocol together with transport protocols like TCP or UDP and optional application protocols like RTP are described as a packet header. The information carried in the header helps the applications to communicate over large distances connected by multiple links or hops in the network. The information comprises of source and destination addresses, ports, protocol identifiers, sequence numbers, error checks etc. As long as the applications are communicating most of this information carried in packet headers remains the same or changes in specific patterns.

Protocol Header Size *(bytes)*	Total Header Size *(bytes)*	Min. Compr. Header Size *(bytes)*	Compr. Gain *(%)*
IP4/TCP	40	4	90.0
IP4/UDP	28	1	96.4
IP4/UDP/RTP	40	1	97.5
IP6/TCP	60	4	93.3
IP6/UDP	48	3	93.75
IP6/UDP/RTP	60	3	95.0

Table 3.4: Header Compression Gains

By observing the fields that remain constant or change in specific patterns it is possible either not to send them in each packet or to represent them in a smaller number of bits than would have been required originally. This process is described as compression.

The process of header compression uses the concept of flow context, which is a collection of information about field values and change patterns of field values in the packet header. This context is formed on the compressor and the decompressor side for each packet flow. The first few packets of a newly identified flow are used to build the context on both sides. These packets are sent without compression. The number of these first few packets, which are initially sent uncompressed, is closely related to link characteristics like bit error rate (BER) and round trip time (RTT). Once the context is established on both sides, the compressor compresses the packets as much as possible. By taking into account the link conditions and feedback from the decompressor, the compressed packet sizes vary. At certain intervals and in the case of error recovery, uncompressed packets are sent to reconstruct the context and revert back to normal operational mode, which is sending compressed packets.

The header compression module is a part of the protocol stack on the devices. It is a feature, which must be negotiated before it can be used on a link. Both end points must agree if they support header compression and on the related parameters to be negotiated.

In keeping with the principle of end-to-end connectivity over IP, the header compression does not introduce any changes in the fields when it compresses and decompresses the header (reconstructing the header as it was before compression). The header compression is a hop-to-hop process and not applied end-to-end. At each hop in the IP network, it becomes necessary to decompress the packet to be able to perform the operations like routing, QoS etc. Header compression is best suited for specific links in the network characterized by relatively low bandwidth, high bit error rates and long round trip times. Realizing that the chain is as strong as its weakest link, header compression is the solution to improve the efficiency (strength) of this link and provide a better utilization of the network and improve

user experience.

The Internet Engineering Task Force (IETF) was formed in 1986 to foster collaboration on the development and evolution of the Internet and related networking technologies. The IETF develops and standardizes header compression schemes. The header compression standards are evolving and the following standards represent the steps in that evolution process:

- RFC 1144 [Jac90], CTCP
  - IPv4 and TCP,
  - differential coding, no feedback
  - 2 bytes
- RFC 2507 [DNP99], IPHC
  - IPv4 (including options and fragments), IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunnelled IP headers, TCP (including options), UDP, ESP
  - differential coding, feedback
  - 2 bytes
- RFC 2508 [CJ99], CRTP
  - IPv4, IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunnelled IP headers, UDP, RTP
  - differential coding, feedback
  - 2 bytes
- RFC 3095 [BBea01]
  - IPv4 (including options and fragments), IPv6 (including extension headers), AH, Minimal Encapsulation headers, GRE, Tunnelled IP headers, UDP, RTP, ESP
  - Window-based Least Significant Bit, feedback
  - 1 byte

The convergence of networks is making ubiquitous computing a reality. To meet the demands of users, operators and manufacturers, the header compression schemes are also evolving. The ROHC working group within IETF has added the Signaling Compression (RFC 3320 [PBea03]) scheme to compress protocols like SIP (Session Initiation Protocol). This scheme is flexible enough to be able to compress similar signaling protocols in the future. The ROHC working group is working on standardizing the IP-Only profile, the UDP-Lite profile and the TCP profiles to be added to the ROHC framework. The Compressed RTP (CRTP) standard has been enhanced

to support links with high delays, packet loss and reordering. This new standard is ECRTP (RFC 3545 [KCea03]). Work is in progress to standardize tunneling of CRTP flows to get maximum benefit for applications such as VoIP supporting a large number of users.

3.3.4.1 Compressed Transport Control Protocol (CTCP)

The first proposed IP header compression scheme Compressed Transport Control Protocol (CTCP or VJHC) for the Internet was introduced by Van Jacobson in 1990 as RFC 1144 and focuses on the TCP protocol. VJHC processes TCP and IP headers together and compresses the 40 byte TCP+IP header to a 4 byte compressed header. A second benefit from the combined processing is the reduced complexity of the employed algorithms. VJHC is based on *delta coding*. The differences between two packet headers are referred to as the *delta*. Instead of transmitting the entire header, VJHC transmits only the delta. This approach achieves high compression. On the downside, it introduces vulnerability. If only one delta coded header is corrupted, all the following packets are erroneous. To recover from these errors and re-establish the current base header, VJHC sends all TCP re-transmissions uncompressed. Thus, VJHC does not require any signaling between compressor and decompressor. The disadvantage is the sensitivity to error-prone links.

3.3.4.2 IP Header Compression

IP Header Compression (IPHC) [DNP99] provides a number of extensions to VJHC. The most important extensions are support for UDP, IPv6, and additional TCP features. With the explicit support of UDP come additional features, such as multicast. Nevertheless, support for RTP is still not given which makes the scheme unsuitable for many multimedia applications. Similar to VJHC, IPHC relies on the change of header fields as well as on the derivation of header field contents. The encoding also employs the delta-scheme, transmitting only the changes in the header fields. The error correction schemes of VJHC are used for TCP packets. For non-TCP packets, no differential encoding between sender and receiver is used. Thus, the compression for UDP-based streams is worse than for TCP-based streams, but the context is not affected by packet losses. Generally, context update packets are sent periodically to maintain the state at both ends.

3.3.4.3 Compressed Real Time Protocol (CRTP)

The Compressed Real Time Protocol (CRTP) scheme presented in RFC 2508 [CJ99] compresses the 40 bytes header of IP to 4 bytes if the UDP checksum is enabled, or to 2 bytes if it is not. This is possible by compressing the RTP/UDP/IP headers together, similar to the VJHC approach. With the characteristics of the RTP

protocol, the changes for the RTP header fields become partially predictable. In addition, changes in some fields are constant over long periods of time. Thus, the expected change in these fields can be implied without even transmitting the differences. These implied fields are also referred to as first order changes. They are stored with the general context for each specific connection. The differences within fields that have to be compressed are referred to as second order differences. An example for these are video frame skips. Video frames are generally transmitted every 40 ms. In case a frame cannot be encoded (e.g., due to lack of processing power or because of a slower play-out ratio), the implied time no longer is accurate. Therefore, the new first order is set to the second order and the connection context is updated. CRTP cannot use a repair mechanism as VJHC does because UDP/RTP are unidirectional protocols without retransmissions. CRTP uses a signaling message from decompressor to compressor to impart that the context is out of synchronization. For lossy links and long round trip delays, CRTP does not perform well. After a single lost packet several sequential packets are lost within the round trip time. Thus, CRTP is not suitable for cellular links (where the header compression currently is envisioned to be implemented in the wireless terminal and the radio network controller, resulting in significant round trip times.

Robust Checksum-based Compression (ROCCO) [SHJD01] is a refinement of CRTP. ROCCO includes a checksum over the original (uncompressed) header in the compressed header. The checksum facilitates local recovery of the synchronization. In addition, ROCCO incorporates compression profiles (tailored for specific applications, e.g., audio or video streaming) and has a code with hints on the change of header fields in the compressed header. These mechanisms improve the header compression performance, especially for highly error-prone links and long round trip times [CFW+01]. Similarly, Enhanced Compressed RTP (ECRTP) [CCHC01] is a refinement of CRTP. ECRTP uses local retransmissions to more efficiently recover from wireless link errors.

3.3.4.4 Drawback of Existing Schemes on Wireless Links

The majority of the header compression schemes were designed for wired links. Wired links are characterized by low error rates. Wireless links, on the other hand, are characterized by higher and bursty transmission error rates. In addition, wireless services – such as those for 3G – often require real-time protocol support. Thus, schemes designed with different goals have several shortcomings outlined in the following:

- *VJHC and its refinements* these schemes were designed for usage with TCP. They are therefore unsuitable for multimedia applications running on UDP. The used delta coding scheme makes these protocols vulnerable to link-errors.

As noted above, the packets from the instant an error occurs to the end of the TCP timeout window are retransmitted uncompressed. This considerably reduces the achieved performance in wireless environments.

- *IPHC*, *CRTP* they both do not offer the efficiency and robustness needed for wireless links [WCH+02]. The local recovery mechanisms of ROCCO are very helpful in ensuring efficiency and robustness; these basic ideas are incorporated in the design of ROHC.

3.3.4.5 Robust Header Compression (ROHC)

ROHC is envisioned as an extensible framework for robust and efficient header compression over highly error-prone links with long round-trip times. This design is motivated by the large bit error rates (typically on the order of $10^{-4} - 10^{-2}$) and long round trip times (typically 100-200 ms) of cellular networks. The design of ROHC is based on the experiences from the header compression schemes reviewed above. In particular, ROHC incorporates elements from ROCCO and Adaptive Header Compression (ACE) [KCZH00] which may be viewed as a preliminary form of ROHC.

ROHC in its original specification as in RFC 3095 [BBea01] is a header compression scheme with profiles for three protocol suites: RTP/UDP/IP, UDP/IP, and ESP/IP. The IP protocol header can be version 4 or version 6. In case any other protocol suite is used, ROHC does not perform compression by using the uncompressed profile. ROHC is located in the standard protocol stack between the IP-based network layer and the link layer. The need for saving bandwidth is limited to the wireless link. So the compression should work only between two wireless nodes, whereas for the rest of the Internet this operation remains transparent. In the simplest configuration, the wireless sender contains the compressor and the decompressor is located in the wireless receiver. ROHC controls the interaction between these two nodes in order to achieve two main goals:

1. The network providers desire a significant bandwidth saving obtainable by reducing the headers to a shorter ROHC header

2. Despite the compression it is necessary to ensure a QoS acceptable for the customers of the network providers

The compressor can sacrifice the bandwidth saving in order to keep the decompressor synchronized even if errors occur on the link. ROHC can thus sacrifice compression efficiency for error correction capability and does therefore not always work at the peak of its compression abilities. Different levels of compression, called states, are used within ROHC. This state-based approach is a new robust solution against the perils of the wireless link.

Context and States In general, data packets that are transferred over a wireless link are not independent from each other but share certain common parameters such as equal source and destination addresses. Moreover, they usually can be grouped together logically, e.g., data packets that constitute an audio stream and data packets that make up the accompanying video stream. Thus, it makes sense to use a stream-oriented approach in ROHC to compress packet headers. Each stream or flow is identified by its parameters that are common to all packets belonging to this stream. The compressor and decompressor maintain a context for each stream which is identified by the same context identifier (CID) on both sides. A context – being a set of state data – contains, for example, the static and dynamic header fields that define a stream.

The ROHC compressor and decompressor can each be regarded as a state machine with three states. Compressor and decompressor start at the lowest state which is defined as *no context established*. In this state, compressor and decompressor have no agreement on compressing or decompressing a certain stream. Thus, the compressor needs to send a ROHC packet containing all the stream and packet information (static and dynamic) to establish the context. This packet is the largest ROHC header that the compressor sends. In the second state, the static part of the context is regarded as established between compressor and decompressor while the dynamic part is not. In this state, the compressor sends ROHC headers containing information on the dynamic part of a context. These headers are smaller than those sent in the first state but slightly larger than the headers used in the third state. In the third and final state, the static as well as the dynamic part of a context are established and the compressor needs to send only minimal information to advance the regular sequence of compressed header fields. Fall-backs to lower states occur when the compressor detects a change or irregularity in the static or dynamic part (i.e., pattern) of a stream, or when the decompressor detects an error in the static or dynamic part of a context.

The compressor strives to operate as long as possible in the third state under the constraint of being confident that the decompressor has enough and up-to-date information to decompress the headers correctly. Otherwise it must transit to a lower state to prevent context inconsistency and to avoid context error propagation.

Compression of Header Fields The compression of the static part of headers for a stream is similar to the other header compression schemes. Header fields that do not change need only to be transmitted at context establishment and remain constant subsequently. More sophisticated algorithms are needed for compression of the dynamic part. With ROHC, the values of the dynamic header fields are derived as linear functions from the sequence number of each packet which in general increases by one for each new packet. However, these functions for the dynamic

header fields are expected to change and the compressor must therefore be able to effectively communicate these changes to the decompressor. For compressing and decompressing dynamic header fields – either directly or by the use of a function – ROHC employs two basic algorithms:

- *Self-describing variable length values* This algorithm reduces the number of bits needed to transmit an absolute value. Small values are described with fewer bits than large values.

- *Windowed Least Significant Bits (W-LSB) encoding* Dynamically changing values usually have their characteristic dynamic behavior, e.g., always incrementing by one for sequence numbers. Using this knowledge of the dynamic behavior, a window is constructed around a reference value which is a previous, correctly transmitted value. Depending on the distance of the new value from the reference value and the relative position of the window with regard to the reference value, the number of bits to transmit the compressed new value is determined. These bits thus describe the advancement of a value from a reference value and their number is small for header field values that do not randomly change but continuously increase or decrease by usually small differences. The advantage of this algorithm is the consideration of the dynamic behavior of the value when defining the position of the window resulting in a minimization of the average number of bits needed to be transmitted in order to describe the sequence of values for a header field in a stream.

The W-LSB compression algorithm in combination with an elaborate protection scheme for sensible data in ROHC-compressed headers contribute to the robustness of ROHC.

Compressor States The three compressor states are: Initialization and Refresh state (IR), First Order state (FO), and Second Order state (SO). In the IR state there is no context for compression available. Thus, the compressor and decompressor have to transit to a higher state as soon as possible for effective compression. When confident of its success to establish a context, the compressor can change to the SO state immediately. In the SO state, only the transmission of a sequence number is necessary and the value of all other header fields are derived from it. These SO state ROHC headers are the smallest ones with in general one byte size. If an irregularity in a stream occurs, the compressor falls back to the FO state. Depending on the irregularity, different ROHC headers with sizes of two, three or more bytes are used in this state. If the stream returns again to a regular behavior (pattern), the compressor transits up to the SO state.

Decompressor States The three decompressor state names refer to the grade of context completeness. In the No Context (NC) state, the decompressor lacks the static and dynamic part of a context. Consequently, it can only decompress IR packets, i.e. packets sent with uncompressed header fields in the IR compressor state. In the Static Context (SC state), the decompressor lacks only the dynamic part (fully or partially) and therefore needs packets that contain information on dynamic header fields in order to complete the context again. The decompressor usually works in the Full Context state which is reached after the entire context has been established. In case of repeated failures in decompression attempts, the decompressor always transits to the SC state first. Then it often is sufficient to correctly decompress an FO packet to recover to the FC state. Otherwise, receiving further errors leads to the transition to the NC state.

Modes and State Transitions To offer the ability to run over different types of links, ROHC operates in one of three modes: Unidirectional, Bidirectional Optimistic, and Bidirectional Reliable mode. Similarly to the states, ROHC starts at the most basic mode (unidirectional) but can then transit to the other modes if the link is bidirectional. Contrary to the states, modes are not directly related to the level of compression. The modes differ from each other in the amount of coupling between compressor and decompressor by the use of feedback packets sent by the decompressor to the compressor. For example, the Unidirectional mode does not make use of feedback packets at all, while the Bidirectional Reliable mode tightly couples compressor and decompressor by requiring a feedback packet for each update of the context. Mode transitions can be initiated by the decompressor at any time for an established context. To do so, the decompressor inserts a mode transition request in a feedback packet, indicating the desired mode.

Unidirectional Mode (U-mode) This mode is designed for links without a return channel. There is no way for the compressor to be certain whether the decompressor has received the correct context information and is thus decompressing correctly. It can only optimistically assume that the decompressor has received the context data correctly by repeatedly sending the same information. However, the decompressor must have a chance to update and correct its context in case of context errors. The compressor therefore periodically times out and falls back to the FO and IR states. Typically, the timeout period for fallbacks to the IR state is larger than the timeout period for fallbacks to the FO state. The decompressor uses the periodically sent FO and IR packets to verify and possibly correct its context. The compressor also falls back to the FO state whenever the pattern of header field evolution changes. Whenever the compressor is in the IR or FO state it sends multiple packets with the same lower level of compression until it is confident that the decompressor has

established the flow context. The compressor then optimistically transits upward to the higher compression FO or SO state. This adds robustness against single packet errors. The U-Mode is the least robust and least efficient mode among the three ROHC modes.

Bidirectional Optimistic Mode (O-mode) As an extension to the Unidirectional Mode, the Bidirectional Optimistic mode uses feedback packets that are sent from the decompressor to the compressor in order to accelerate state transitions at the compressor and to avoid the periodic fallbacks to the FO and IR states. Context update acknowledgements (ACKs) are used to notify the compressor that the decompressor has successfully received context information. (These ACKs from the decompressor are optional, thus the compressor may still need to use the optimistic upward transitions.) In case of a context error, a context recovery request (NACK) is sent to the compressor, causing a retransmission of context information to update and repair the context at the decompressor. With a Static-NACK the decompressor forces the compressor back to the IR state, thus re-establishing the static context. With these context updates on request, the compressor can achieve a higher compression efficiency compared to the unidirectional mode. Due to the mostly weak protection (3-bit CRCs) of context updating data sent in the Bidirectional Optimistic mode, there is still a not to be neglected probability of context damage that can result in a sequence of incorrectly transmitted packets. For applications that prefer a more reliable transmission with a lower probability of incorrect packet transmission, the Bidirectional Reliable mode was conceived.

Bidirectional Reliable Mode (R-mode) To achieve a lower probability of incorrect packets, a more powerful error correction (7-bit CRCs) is used for context updating information in the Bidirectional Reliable mode. In addition, the compressor transits to the FO or SO state only after receiving an ACK from the decompressor. The compressor transits downward to update the context or upon decompressor request (NACK, Static-NACK). With this mode, the behavior of the compressor and decompressor are thus even closer coupled than with the optimistic mode. The rare NACKs provide a quick context recovery in case of errors. Therefore, the compressor always knows in which state the decompressor is and when to make a state transition.

Timer-based Compression of RTP Timestamps A special, more efficient compression method for the RTP timestamp header field can be applied under certain conditions. If the application hands over RTP packets in real-time to the ROHC compressor, the time difference between the handover of two RTP packets is proportional to the difference of the RTP timestamp header field values in these two

packets. Provided that the transmission channel has a low delay jitter, the decompressor can then use the time difference between the reception of two compressed packets to estimate the new RTP timestamp value based on the previous value. With timer based compression, the compressor needs to send even fewer bits for compression of the RTP timestamp than with the standard W-LSB encoding based method. These bits are used to refine the estimation of the decompressor. The number of bits needed for refinement depends on the amount of delay jitter on the transmission channel which has to be determined.

ROHC over Wireless Ethernet Media The specification for ROHC does not describe the inter-operation with the underlying link-layer in detail. Only a few requirements that have to be met by the link-layer are mentioned (e.g., no packet reordering or duplication on the channel between compressor and decompressor). Additional drafts and RFCs specify how ROHC operates on top of certain link-layer protocols (e.g., PPP [Sim94, Sim97]). A large group of link-layer technologies are formed by the Ethernet-based network technologies. Among them, the wireless local area variants, such as IEEE 802.11 (Wireless LAN) or Bluetooth, exhibit similar bit error characteristics. Consequently, there are efforts under way to standardize the operation of ROHC on top of wireless Ethernet-like media. An exemplary application that can benefit from the employment of ROHC is Voice-over-IP telephony in a wireless LAN environment.

Robust Header Compression is an efficient compression protocol that is especially suitable for transmission of real-time multimedia data over wireless links with high error probabilities. Sophisticated compression and encoding methods and elaborated schemes that provide robustness against transmission errors make Robust Header Compression superior to the previously described header compression protocols. However, a highly increased complexity of the compression algorithms in ROHC results in an increased demand for computing power. However, the advances in microelectronics during the last few years make it possible to cost-effectively provide this computing power in small, mobile devices. Ongoing research efforts primarily focus on the tuning of the parameters of ROHC, see e.g., [WSC+02], the performance evaluation of ROHC, and its adaptation to specific wireless systems, e.g., cellular networks or wireless LANs.

3.4 Error Simulation for Wireless Networks

Wireless network performance simulation depends on knowledge of the statistical distribution of bit errors for each wireless link represented in the network. The

distribution is a function of all the link variables, including the channel, noise, interference, modem, coding, equalization, etc. The bit errors encountered on a communication link can be obtained by a waveform level simulation of the entire link. However, this kind of simulation can be computationally prohibitive, particularly for simulations of networks comprising many links.

A more efficient form of simulation is discrete event simulation, whereby one generates a bit error stream directly. Waveform simulation typically uses many samples per bit and requires simulating the entire communication link for each sample. By contrast, discrete event simulation of bit errors requires only one sample per bit, and, as will be seen, only requires the generation of one or two random numbers per sample.

A wireless link bit error model has been developed, that enables discrete event simulation of the bit errors encountered on wireless links. The model development has been based on error streams derived from real experiments of link performance under various conditions [Wie05]. Values of the model parameters have been determined by analyzing the distributions of the lengths of error bursts and error gaps (error-free intervals). Lemmon [Lem02] showed that the distributions generated by the waveform simulations and by this type of models are quite similar; however, the calculations with the statistical model typically run tens of thousands of times faster than the waveform simulations (the precise increase in speed depends on the type of link being simulated).

Bit error models generate a sequence of noise bits (where zeros represent good bits and ones represent bit errors) that is modulo 2 added to input bits to produce output bits. Models can be grouped into two broad classes: memoryless models and those with memory. In memoryless models the noise bits are produced by a sequence of independent trials. Each trial has the same probability $P(0)$ of producing a correct bit and probability $P(1) = 1 - P(0)$ of producing a bit error.

Measured data from actual communication links indicate that many links have memory, that is, the errors occur in isolated bursts. This is because many link impairments, such as impulsive noise, switching transients, and multipath fades, are bursty in nature. A commonly used technique to endow a model with memory is to make the bit error probability depend on the states of a Markov chain.

The use of Markov chains in bit error models was initiated by Gilbert [Gil60]. The Gilbert model is based on a Markov chain with two states: G (for good) and B (for bad or for burst). In state G, transmission is error-free. In state B, the link has probability h of transmitting a bit correctly. A transition diagram and bit error probabilities for the Markov chain are shown in figure 3.8. For suitably small values of the transition probabilities $p = prob(B - G)$ and $P = prob(G - B)$, the states B and G tend to persist and the model simulates bursts of errors.

This simple model has three independent parameters (p, P, and h) and was originally used to describe performance measurements over telephone circuits. Whether

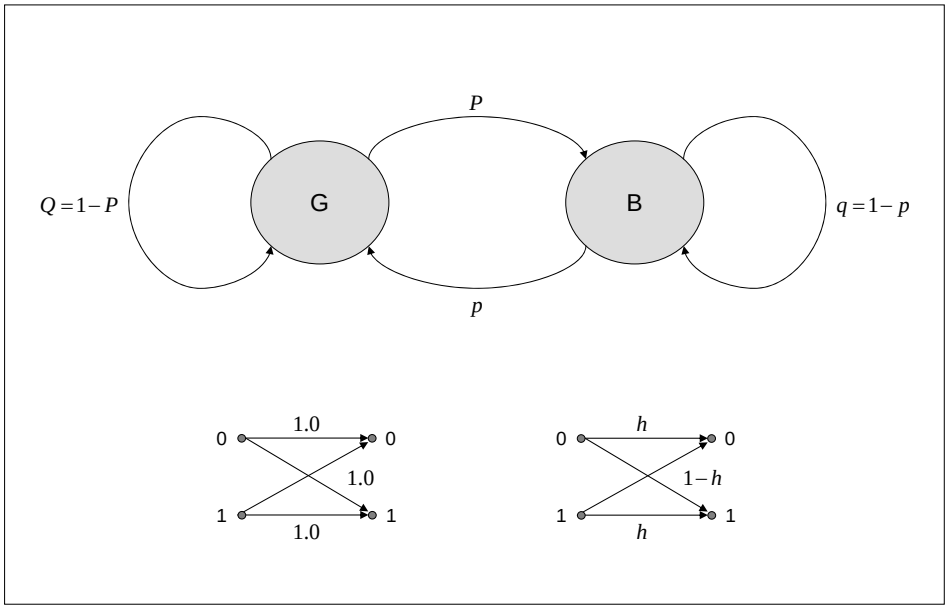


Figure 3.8: Transition diagram and bit error probabilities for the Gilbert model

the model is adequate to describe the error performance of wireless links has been investigated in the present work. One can envisage more complicated models with more parameters (e.g., more than two states in the Markov chain); however, determination of the model parameters from measured data becomes more difficult as the number of parameters increases.

The parameters p, P, and h are not directly observable and must therefore be determined from statistical measurements of the error process. Runs of G alternate with runs of B. The run lengths have geometric distributions, with mean $1/P$ for the G-runs and $1/p$ for the B-runs. The fraction of time spent in state B is therefore $P(B) = P/(P + p)$. Since bit errors occur only in state B, and with probability $1 - h$, the probability of error is

$$P(1) = (1 - h)P(B) = (1 - h)\frac{P}{p + P}.$$

The sequence of states cannot be reconstructed from the sequence of bits in the error process, because both zeros and ones (good bits and bit errors) are produced in the B state. Thus, the distributions of run lengths of the states cannot be used to determine the model parameters from measurements of the error process. However, the bits of the error process itself, (i.e., runs of zeros and ones) are observable, and the distributions of run lengths of zeros (error gaps) and ones (error bursts) can be

used to determine the model parameters.

3.4.1 Classes of Errors

As mentioned earlier, error tracks could be described by sequences of zeros and ones. Hence the Backus-Naur-Form (BNF) is given by:

```
error_trace = observation [ error_trace ]
observation = "0" | "1"
```

For the semantic interpretation of the two symbols of an error trace the following convention is legal:

- 0 indicates an error-free state, hence a successful bit/packet transmission
- 1 indicates an error state, and therefore an unsuccessful bit/packet transmission

To partition an error trace into multiple states a fixed window size has to be determined in order to evaluate the observed characteristics. The window size describes the amount of observations in the direct environment. An experimentally documented error trace could be considered as a hidden Markov-process. Thereby the assumptions of the single states are the result of a hidden random process. Now it can be recognized that every symbol of the alphabet $\{0, 1\}$ is generated by an additional random process. This inner random process is described by a state-specific error probability in the form of packet error rate (PER) or bit error rate (BER).

Possible error sources in the field of wireless transmissions:

- *Attenuation* when electromagnetic energy encounters matter, some of it is lost in the form of heat
- *Front end overload* if a very powerful transmitter of one frequency band is near a receiver of another band, the transmitter may overwhelm filters in the receiver and inject substantial noise
- *Narrowband interference* this is due to an unfriendly transmitter occupying a small frequency band overlapping (perhaps totally) with the band we wish to use
- *Spread-spectrum interference* this is due to an unfriendly transmitter either switching between narrowband frequencies or spreading its energy simultaneously across a wide frequency band

- *Natural background noise* for example, infrared wireless networks may perform poorly if they are near sources of direct sunlight
- *Multipath interference* when electromagnetic radiation reflects off objects or diffracts around objects, it takes multiple paths between the transmitter and the receiver. Since these paths are typically of different lengths, there will be destructive interference, which can greatly reduce signal strength
- *Path loss (dispersion)* the intensity of electromagnetic energy reaching a receiver is decreased by distance even in free space
- *Motion* if two communicating objects are moving with respect to each other, the frequency of the electromagnetic energy changes according to the Doppler effect. While this effect may be significant in some radio environments [HS93], the Doppler shift due to moving a W-LAN unit at the speed of sound would be substantially less than the inaccuracy of the clock crystals employed by W-LAN [Tuc93]
- *Data dependent effects* some modulation schemes can lose clock synchronization in the face of certain long bit patterns
- *Collision and loss of efficiency by CSMA/CA* hence, the possibility of collisions could just be decreased but not totally avoided

Summarized, wireless error sources are in the perspective of the ISO/OSI reference model mainly concentrated in the data-link layer (layer 2). Another important class of errors is *fading*.

In an indoor/outdoor environment, the received signal is made up of numerous attenuated, reflected, diffracted and transmitted versions of the original signal. Such multipath propagation results in a received signal whose amplitude significantly changes with location. This phenomenon is known as *multipath fading*. The latter is decomposed into two categories:

- *Slow fading* describes the slow variations in received signal power when the receiver moves behind obstacles (mountains, houses, etc.)
- *Fast fading* consists of the phasor addition of the various multipath signals since each signal presents a specific amplitude and phase. This signal can combine constructively, i.e. a peak, or destructively, i.e. a fade or minimum.

3.4.2 Error Models

Error models try to map the error behaviour of a wireless communication channel to stochastic models so that observations from the real world can be compared nearly

equally to results received by the models. Hence, by these models generated error tracks can be compared with real error tracks concerning statistical spread metrics as for example the standard derivation or e.g. measures of central tendency for instance the arithmetic mean. The arithmetical mean of an observation is

$$\tilde{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_j, \quad x_j := j^{th} observation$$

and the standard derivation is

$$s = \sqrt{s^2} = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_j - \tilde{x})^2.$$

Special interest is deserved to the characteristics of error bursts, as seen later. These are described to begin and to end with an error [Wie05].

Error models which describe the error behaviour of wireless networks could be differentiated into two observation schemes: errors in the bit layer, and observation of errors per packet or frame layer. Additionally, specific model parameters have to be defined so that a model could represent a typical error behaviour. As of the derivation process two basic classes of models could be differentiated: analytical and empirical error models.

3.4.2.1 Analytical Error Models

In case of analytical error models assumptions are made regarding the reality which should be reflected. Based on these assumptions describing the environmental surroundings spreading-models were defined. Details are covered in [RMV97]. In these spreading-models technical specifications, like for example the effective isotropically radiated power (EIRP) or antenna gain, are considered. The received signal strength with consideration to path loss could be calculated by[3]

$$P_R = P_T \cdot g_T \cdot g_R \cdot \left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{d^\gamma}$$

where P_R is the signal strength on receivers side, P_T is the signal strength on senders side. The wave length is designated with λ, the distance between sender and receiver with d, g_R and g_T are the receivers and senders antenna gain respectively. The distance exponent γ is chosen differently - depending on the environmental surroundings. The following table 3.5 shows concrete values for γ in different scenarios:

[3] assumed an isotropic omni-directional antenna

Environment	γ
free-space	2
urban area cellular radio	2.7 – 4
shadowed urban cellular radio	5 – 6
in-building line of sight	1.6 – 1.8
obstructed in-building	4 – 6
obstructed in factories	2 – 3

Table 3.5: γ values for different environments
(following [RMV97])

However, with the mentioned proceeding the fluctuating behaviour of radio signals is not accommodated. In case of multipath propagation, the signals arrive at indefinite different times with various different amplitudes and phasing for the same signal. The received signal is therefore a composition of these single components, hence correlated, whereby this composition could be constructive but also destructive. These variations of the received signal energy around a certain mean are mapped with stochastic probability density functions (PDF) as a result of random processes. At least two of them will be introduced in the next two paragraphs.

Rayleigh Distribution The most common characterization of small scale fading is by means of the Rayleigh distribution,

$$f_r(r) = \frac{r}{\sigma^2} e^{\frac{r^2}{2\sigma^2}},\ r \geq 0.$$

This is the amplitude distribution of the sum r of a large number of uncorrelated rotating vectors with amplitudes of the same order of magnitude and uniformly distributed phase. The resulting phase is also uniformly distributed in the interval $(0,\ 2\pi)$. The parameter σ represents the RMS value of the received signal. Indoor measurements at 5.3 GHz have shown Rayleigh fading amplitudes in both LoS and NLoS situations [KV98], [KV99]. However, the investigations in [KE97] have shown that in certain situations the Rice distribution can give a better fit with the measured data.

Rice Distribution The Rice distribution describes the amplitude distribution of the sum r of one dominant vector and a large number of uncorrelated rotating vectors with amplitudes of the same order of magnitude and uniformly distributed phase. The Rice distribution is given by

$$f_r(r) = \frac{r}{\sigma^2} I_0\left(\frac{r \cdot \rho}{\sigma^2}\right) e^{-\frac{r^2+\rho^2}{2\sigma^2}},\ r \geq 0$$

where I_0 is the modified Bessel function of the first kind and zero order. The parameter ρ denotes amplitude of the dominant component. The Rice distribution contains the Rayleigh distribution as a special case; elimination of the dominant component turns the distribution of the amplitude from Rice to Rayleigh.

The Rayleigh distribution is also a special case of the Nakagami distribution (with $m = 1$) [Nak60]. Furthermore, for higher values of the parameter m the Nakagami distribution is approximately similar to the Rice distribution. The probability density function is given by

$$f(r) = \frac{2m^m r^{2m-1}}{\Gamma(m)\Omega^m} e^{-(m/\Omega)r^2}$$

where $\Gamma(m)$ is the Gamma function, $\Omega = E\left\{r^2\right\}$ and m is the fading figure defined as

$$m = \frac{\Omega^2}{E\left[(r^2 - \Omega)^2\right]},\ m \geq \frac{1}{2}.$$

Physically speaking the Nakagami m-distribution is obtained when we sum many component vectors, which are not only random in phase but also random in length. It is therefore a more general model than Rayleigh. The Nakagami distribution offers an advantage when analyzing combinations of multiple signals (e.g. diversity, Rake receivers, Smart-antennas) in that it is easier to use than the Rice distribution.

Additive White Gaussian Noise (AWGN) Distribution The AWGN distribution is mainly used to simulate thermal noise within the technical equipment, atmospheric or random interferences. Therefore white noise[4] is added to the signal as depicted exemplary in figure 3.9.

3.4.2.2 Empirical Error Models

In statistical and empirical channel modeling, a number of channel characteristics are represented either directly or statistically from measurements of the mobile radio channel. So the first step of error modeling is to observe and collect attributes of interest. Thereby the attribute is the success of a packet- or frame- respectively bit transmission. These observations construct the error track. In a second step, stochastic parameters (probabilities) have to be defined, which are characterizing the observed error distribution. An example for an extended empirical error model is the MTA algorithm [KZJL03] by A. Konrad et al. in 2003. MTA is the abbreviation for Markov-based Trace Analysis. The work of Lo and Ngai [LN04] describes an approach for collecting parameters for packet- as well as bit errors. They used a

[4]Noise having a frequency spectrum that is continuous and uniform over a specified frequency band. White noise has equal power per Hertz over the specified frequency band

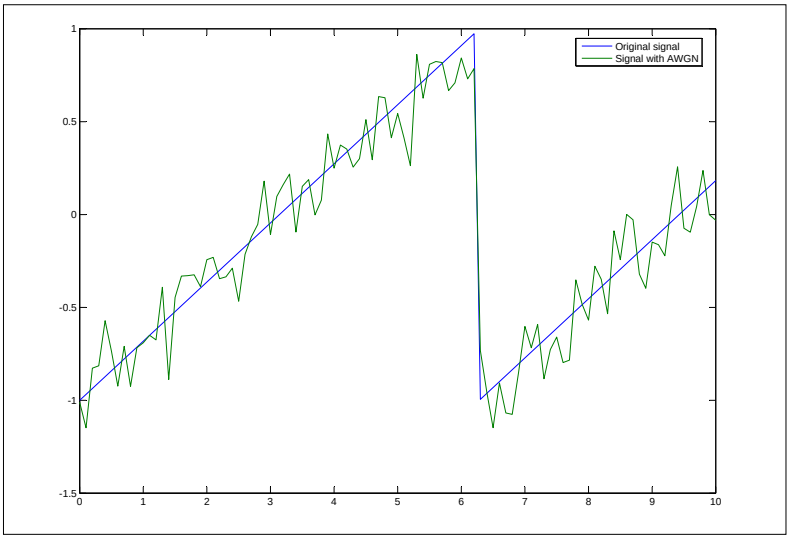


Figure 3.9: Sawtooth with Additive White Gaussian Noise
(the power of the sawtooth signal is assumed to be 0 dBW, with AWGN the SNR is then 10db)

modified Linux device driver for the Prism II chipset (802.11). The speciality about this chipset is, that the whole MAC-functionality is done with software, hence done by the device driver.

In this thesis case, the decision has to be made between a *simplified Markov-Model* and a *Markov-Model* with probabilistic error distribution. The decision about the amount of states of the model remains subjective. The work of Willig and Ebert [EW99] proposes some heuristic rules for the amount of states. Finally, from the observed data, model parameters are derived. These model parameters could be manually determined or with programs, like for example Jahmm [Fra05] or GHMM [Sea05]. In [LN04], a detailed overview and usage of JAHMM is given.

3.4.2.3 Evaluation of Different Error Models

To evaluate the different possibilities in case of wireless error modeling, the characteristics of the considered target have to be focussed. Center of attraction is the radio channel of a wireless LAN. The preceding explanations showed the complexity and variability of propagation and hence the involved error characteristic of electromagnetic waves in UHF and SHF. Multiple time- and location specific variables restrict the area of validity spatiotemporaly regarding an error model for radio channels.

In case of analytical error modeling, comprehensive knowledge about possible influencing factors regarding signal propagation is required. Influencing factors like

the composition of surrounding objects and their influence to propagation of electromagnetic waves, power- and attenuation parameters regarding active and passive transmitter- and receiver technique (antennas, modulators, demodulators etc.). The latter parameters could be determined from the manuals and/or spec sheets of the used technical devices but environmental parameters are quite difficult to receive. Just approximations with a definitely cut back. Also collisions in the data-link layer aren't considered – they are mainly quantified by simulations [EW99].

In contrary to analytical error modeling, empirical error modeling doesn't require detailed knowledge about the environmental surroundings and their influence to propagation. Therewith, this problem is avoided but empirical error modeling depends on the used technical equipment and resources for recording and measurement. In this way, all identified problem sources, collisions included, are covered by the error modeling process.

Thus it appears that both error modeling approaches become more reliable and more skilled, if the amount of system states is increasing [AK03]. In this thesis, an empirical error modeling approach is considered, hence the next paragraph deals with how to determine the parameters for the empirical error model which is used later.

3.4.3 Error Simulation and Implementation

The developed error generator additionally consists of two tools for generating an error track - *sendUDPTrace* and *receiveUDPTrace*. For calculating the model parameters out of an error track, *calcProbs2* was developed. Parameters for an simplified Gilbert-Elliott-Model and for a two-state FSMC[5] (therefore the window-size has to be specified) could be determined. UDP was chosen, because RTP is based on UDP and RTP is used for the voice transmission.

Error Track for a Typical Residential Building Scenario For the following exemplary error modeling, the two just described programs were used. The observed object is an 2 Mbps IEEE 802.11 DSSS radio channel between two access points in ad-hoc mode. The environmental surrounding could be characterized as follows:

- NLOS (non-line-of-sight)
- both communication partners in the same building (residential building, distance appr. 12m)
- multipath propagation → *Fast Fading*

The determined experimental error track is depicted in figure 3.10

[5] Finite-state Markov Chain

```
0000000000 0000000100 0111001000 1001110110 0000000000 0001110100
0000000000 0001000000 0000100000 0000000011 1101000101 1011100000
0011100000 1111100000
```

Figure 3.10: Experimental error track

Based on the error track above, a FSMC- and a simplified Gilbert-Elliott Model is developed:

Two-state FSMC Model At first, the decision about the window size has to be made. This is important for the interpretation of the error track and the choice regarding the amount of model states to be used. The window size conditions the relation between the observations (symbols of the error track) and the model states. Every model state has its assigned symbol, hence the error probability (mean packet error rate, transition probabilities for every state) could be calculated. The error rate in state j is then

$$P_{ej} = \frac{\#\{x_j | x_j = 1\}}{\#\{x_j\}},\ j \in \mathbb{N}, x_j \in \{0, 1\}$$

The error probability is assumed to be uniformly distributed, so that the total amount of erroneous observations ($x_j = 1$), which are ascribed to state j, are related to the total amount of all observations respectively. The calculation of the transition probabilities ensues analogue to the calculation of the error probabilities. With a window size of 5 observations and 2 states, the error track from above leads to the following stochastic parameters depicted in table 3.6.

Attribute	Frequency
good state	88
bad state	52
sporadic errors in good state (P_{e1})	2
sporadic errors in bad state (P_{e2})	34
transitions from good to bad ($P_{12} \Rightarrow P_{11}$)	5
transitions form bad to good ($P_{21} \Rightarrow P_{22}$)	5

Table 3.6: FSMC-Model parameters

Therefore $P_{e1} = \frac{2}{88} = 0.02272$, $P_{e2} = \frac{34}{52} = 0.65385$, $P_{12} = \frac{5}{87} = 0.05747 \Rightarrow P_{11} = 1 - P_{12} = 0.94253$ and $P_{21} = \frac{5}{52} = 0.09615 \Rightarrow P_{22} = 1 - P_{21} = 0.90385$.

And hence the following summarizes the parameter of the error model:

- number of states: 2
- packet error rate in state 1: 0.02273
- packet error rate in state 2: 0.65385
- matrix of transition probabilities
$$\begin{pmatrix} 0.94253 & 0.05747 \\ 0.09615 & 0.90385 \end{pmatrix}$$
- transition limit: 1 packet

Simplified Gilbert-Elliott Model Now the error track above is used to build a simplified Gilbert-Elliott Model with one bad and one good state. The error probabilities in the good state are determined by the impossible respectively sure result in bad state. Therefore the following model parameters result:

- number of states: 2
- packet error rate in state 1: 0.0
- packet error rate in state 2: 1.0
- matrix of transition probabilities
$$\begin{pmatrix} 0.83495 & 0.16505 \\ 0.47222 & 0.52778 \end{pmatrix}$$
- transition limit: 1 packet

The quality of the created models related to mapping of realistic error bundles could be determined by stochastic variables of the simulated/factitious error track, like for example arithmetic mean and standard derivation of the bundle size. Further experiments showed, that the simplified Gilbert-Elliott Model is less useful, to map realistic error bundles that happen in IEEE 802.11 wireless networks. Aráuz and Krishnamurthydie [AK03] came to the same conclusion. The quality could be enhanced while considering more model states $n > 2$. Hence, a very important fact is the window size, regarding limitation and partitioning of system states.

As a result of increased demand for wireless network technologies and their usage, the research field of error modeling for radio channels became important and a quite evolving area. Hence, in scientific literature different approaches were entirely de-

scribed. The majority is based on Markov-models of various occurrences, which are partly the next topic. The reconsidered approaches doesn't claim for completeness but are compatible with the developed error simulation system.

Lo, Ngai The authors used simplified Markov-models with one initial good and multiple bad states, hence, they partitioned the fading state of the channel. While in good state no errors occur (impossible event), errors are the sure event in the bad states.

Intervisibility (LoS), 275 m distance between sender and receiver

- number of states: 3
- bit error probability good state: 0.0
- bit error probability bad states: 1.0
- matrix of transition probabilities

$$\begin{pmatrix} 0.999798 & 0.000202 & 0 \\ 0.005 & 0 & 0.995 \\ 1 & 0 & 0 \end{pmatrix}$$

- transition limit: 1 bit

Multipath propagation, car park (edge)

- number of states: 8
- bit error probability good state: 0.0
- bit error probability bad states: 1.0
- matrix of transition probabilities

$$\begin{pmatrix} 0.999993 & 0.000007 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0.98 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- transition limit: 1 bit

Willig, Ebert The authors used developed an analytical bit error model based on a Markov-model with two states. The observed environmental surrounding reveals fast fading. The attenuation is considered by a Rayleigh-distribution. The mean bit error rate of the two system states is determined by receiver's side SNR with arithmetic mean 20.5 dB and a transition threshold (good ⇒ bad) with 20 dB. For the attenuation, a Doppler effect from the communication of the two partners is assumed.

2 Mbps, BPSK DSSS wireless LAN

- scenario: fast fading (Doppler effect due to mobility)
- number of states: 2
- bit error rate in good state: 10^{-5}
- bit error rate in bad state: 10^{-2}
- matrix of transition probabilities
  $\begin{pmatrix} 0.999918 & 0.000012 \\ 0.000001 & 0.999790 \end{pmatrix}$
- transition limit: 1 bit

3.4.4 Error Concealment and Error Resilience Techniques

In the last section error simulation for wireless networks was discussed. More important than errors and dealing with them is to avoid erroneous wireless audio transmissions. Different mechanisms and techniques will be detailed in the this section.

When sending information over a network errors can occur. Depending on the infrastructure and the underlying technology these errors can range from bit error to losses of complete network packets. Bit errors can range from bit deletions over bit insertions to bit inversions and are due to the imperfectness of physical connections and components. Packets can be lost because an intermediate network element is congested and has to discard new incoming data. When a lot of bits are lost (i.e. because a complete network packet does not reach the receiver) one speaks of a burst error. As mentioned above in nearly all multimedia codecs prediction and variable length coding (VLC) are used in order to save space. Therefore the loss of bits will mostly result in the situation, that not only the lost data itself is unusable but also a special amount of consecutive data, which depends on it. Therefore also bit errors will lead into burst errors due to error propagation.

The techniques described can be divided into three sections:

- sender-based
- receiver-based
- sender/receiver-based

These three sections are not mutual exclusive and should be used in combination with each other. Most of the sender-based techniques add redundancy overhead to the data and are called *error resilience* techniques, the receiver-based methods assume that there have been errors and try to correct or at least conceal them, they are called *error concealment* schemes, and the sender and receiver-based techniques presume a feedback channel from the decoder to the encoder and are based on the interaction of sender and the receiver, therefore called *interactive techniques*. When comparing the different techniques one has to take the given application, available hardware performance and the network characteristics into account, because all of the techniques imply different latency, computing complexity, signal robustness and signal quality and they can handle different error rates with satisfying results.

Sender-based Techniques As mentioned before one main task of multimedia compression is to eliminate redundancy. Sender-based techniques normally work in the opposite way by adding redundancy (creating overhead). This is because it is easier and sometimes even only possible to reconstruct a lost signal in the presence of some redundancy. As these methods try to make the data robust to errors in advance they are called error resilient or forward error correction techniques. Prominent examples are layered encoding and multiple description coding, which are described later.

Receiver-based Techniques The receiver-based methods do not increase the used bandwidth and should be used in combination with the sender-based. All of the methods try to hide the effects of lost data by inserting harmless errors or by trying to reconstruct the lost data.

The easiest method is by inserting some harmless data into the gaps, hoping that the human ear will not notice it or that it will be less annoying. The simplest way possible is a zero length insertion, which is called slicing. The parts before and after the gap are just connected together. Of course this normally will be noticed, especially the bigger the gap gets, but this method is implemented very easily. Better results will be achieved by inserting silence or white or comfort noise into the gap. Some codecs allow the transmission of dynamic noise patterns, wherein the background volume level of the source signal can be advised. These patterns

will be used instead of inserting general white noise. A last variant of this simple insertion approach is just repeating the last sample and optionally fading out.

Interpolation takes the correctly received patterns before (and sometimes after) the gap into account and tries to estimate the missing part by best fitting criteria. These techniques can only be achieved with high computational complexity and memory-supported decoders. The first method is pattern marching, based on samples before and/or after the gap, where a suitable signal is searched for. This can be done for each sub band separately or together based then on the waveform in the time domain. Another method is the use of pitch waveform replication, where a pitch-detecting algorithm is used to find a suitable insertion sample. As a last method of this class timescale modification should be mentioned. There the sample before the gap is taken and played with a slower speed than usual, in order to completely fill the gap. It is like insertion, but one inserts usable data. Instead if using static harmless patterns, an algorithm that finds overlapping pitch vectors on both sides of the gap can be used to control the playing speed in order to avoid abrupt changes.

Sender/Receiver-based Techniques The sender-receiver-based techniques assume that there is a feedback channel between them and therefore the sender can adapt its coding scheme and parameters to the actual channel error characteristics. In many applications such a feedback channel is not available nor effective. This can be because of technical or time-critical reasons: while broadcasting or in multi-point transmission the receivers are possibly not known to the sender. In interactive or real-time communication a feedback channel could not be used because of the delay introduced by it.

In the Internet normally the real time-transport-protocol (RTP) is used for transmitting multimedia data. Then the real time-control-protocol (RTCP) is available, which allows monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. Also when using proprietary or other protocols there often is a possibility to send feedback information to the sender. With interaction between the source and the channel coder the sender can balance the amount of the bandwidth which is used for FEC and which is used for source coding. When there are frequent errors in a connection, it would be wise to lower the video/audio quality and use more bandwidth for FEC. When the connection is mostly error free, one could decrease the amount of bits wasted for FEC and therefore increase the audio quality.

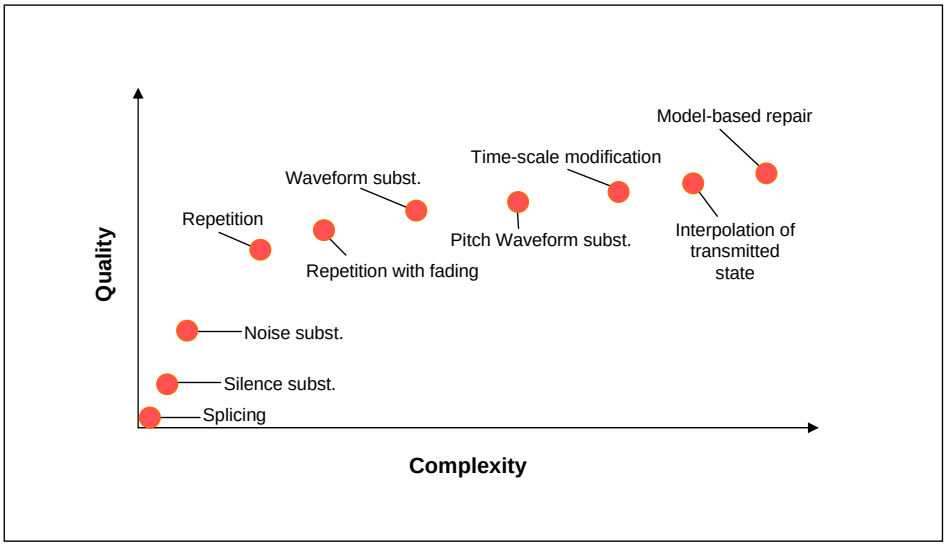


Figure 3.11: Trade-off for error concealment in speech signals
(Rough description by [PHH98])

3.5 Summary

Summarized multiple wireless network technologies could be emphasized to enable VoWLAN-services. At a first glance in this thesis wireless networks built upon IEEE 802.11 are aimed primarily. Nevertheless the developed concepts might be introduced to other wireless technologies as well. As transmission requirements, solutions and problems are intensively discussed the following chapter presents the next step to an improved VoWLAN system. Various speech processing techniques and particularly speech coding aspects are detailed as they are partly used by own developments later in this thesis.

4 Speech Processing

4.1 Introduction

Speech codecs compress the voice in such a way as to affect its quality and in turn the quality of service (QoS) delivered by IP telephony systems. The speech encoder converts the digitized speech signal (after A/D conversion) to a bit-stream, which is packetized and sent over the IP network. The speech decoder then reconstructs the speech signal from the packets received. The reconstructed speech signal is, therefore, an approximation of the original signal. Speech codecs are deployed at end points and, so, determine the achievable end-to-end quality. A speech codec has several important features, including speech quality, bit or compression rate, robustness, delay, sampling frequency, and complexity. Additionally modern approaches consider the Human Auditory System in order to improve intelligibility while increasing the compression ratio. In this chapter elemental speech coding techniques will be presented as well as other mandatory techniques in order to setup up the toolbox for own developments.

4.2 Speech Coding

The telephone network is the most common example of an electronic speech communication system. Due to its scale, a significant amount of research has been directed towards finding efficient methods of transferring the spoken word. Developments in this area have resulted in the gradual replacement of analog components of the network by digital circuits. Digital communication systems outperform analogue systems in the presence of noise and also provide the possibility of easily encrypting the data for security purposes. However, they do have a major disadvantage in that the required bandwidth is increased, and so the cost of the network is also increased. Speech coding attempts to reduce the required channel bandwidth, while at the same time maintain the high quality of the input speech.

Speech signals are inherently analog by nature, therefore a transformation from the continuous domain into the discrete domain must be carried out before any digital coding is performed. This transformation utilizes time discretization or sampling followed by amplitude discretization or quantization. Sampling is information preserving if it is carried out at a sufficiently high rate. Quantization, on the other

hand, always introduces some distortion into the signal. So the operation of the quantizer is critical to the overall performance of the system.

The goal in speech coding is to design a low complexity coder that produces high quality speech with the lowest data transmission rate possible. The properties of low complexity, high quality output and a low bit rate tend to be mutually exclusive and a trade off exists in practical coders.

4.2.1 Speech Production and Perception

The human vocal and auditory organs form one of the most useful and complex communication systems in the animal kingdom. This section briefly describes this system and its properties, with a view to exploiting them when transmitting speech.

4.2.1.1 Speech Production

All speech sounds are formed by blowing air from the lungs through the vocal tract. In the adult male the vocal tract is approximately 17 cm long with a cross-sectional area that varies from zero to about 20 cm^2 [ISP05].

The sole purpose of the lungs, in speech production, is to act as an air supply. The vocal folds are two membranes situated in the larynx. These membranes allow the area of the trachea (the glottis) to be varied. For breathing the vocal folds remain open but during speech production they open and close. Speech can be broken into two broad classes, namely voiced or unvoiced. Although in reality this distinction is not well defined and speech tends to be a mixture of the two classes.

During voiced speech the vocal folds are normally closed. As the lungs force air out pressure builds up behind them. Eventually the pressure is so great that the folds are forced open. This flow of air causes them to vibrate in a relaxation oscillation. The frequency of this vibration is determined by the length of the folds and their tension. This vibration frequency is known as the pitch frequency and for normal speakers it can be anywhere in the range of 50 to 400 Hz. Women and children tend to produce a higher average pitch frequency than men since their vocal folds are shorter. The effect of this opening and closing of the glottis is that the air passing through the rest of the vocal tract appears as a quasi-periodic pulse train. During unvoiced speech the vocal folds are normally open so air can pass freely into the rest of vocal tract.

At this point the speech signal consists of a series of pulses or random noise depending on whether the speech is to be voiced or unvoiced. The spectrum of this signal is essentially flat, although there will be a *fine spectral structure* in the spectrum of the voiced signal due to the pitch frequency and its harmonics. This flat spectrum is then passed through the rest of the vocal tract that can be viewed as a spectral shaping filter. The effect of this is that the vocal tract forces its own

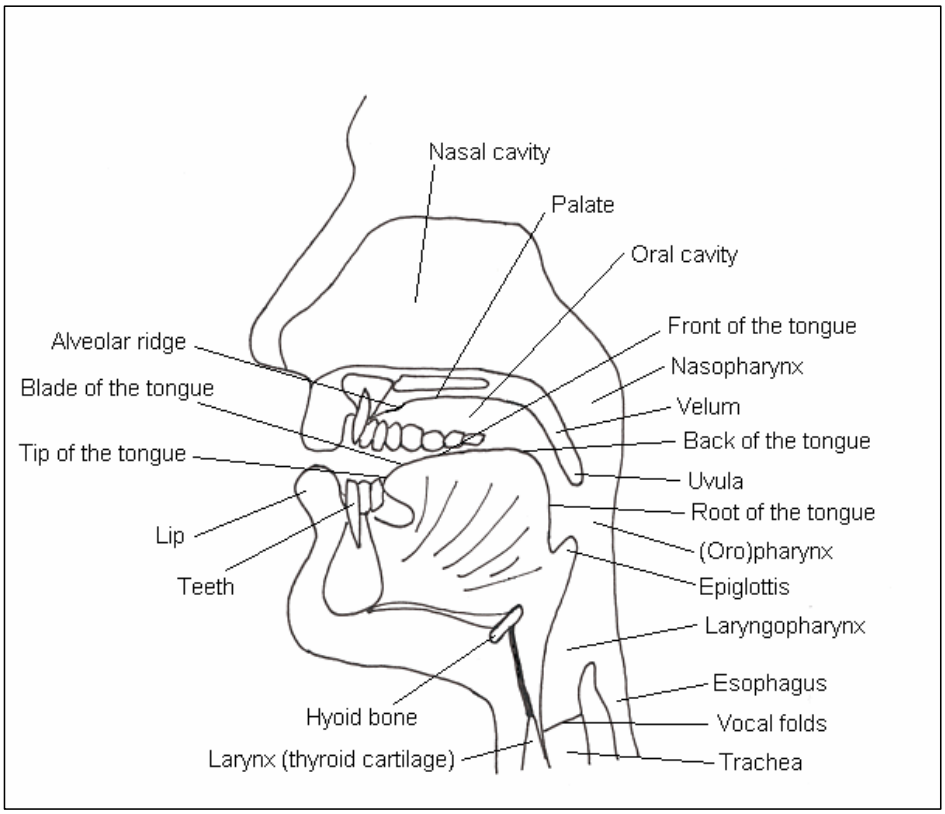


Figure 4.1: The vocal tract
(`http://www.clas.ufl.edu/users/sshear/`)

frequency response on the incoming signal. This frequency response is governed by the size and shape of the vocal tract.

The nasal cavity is an auxiliary path for sound. It begins at the velum and is about 12 cm in length [ISP05]. When the velum is lowered the nasal tract is acoustically coupled with the rest of the vocal tract, dramatically changing the nature of the sound. It is the voluntary variations in the shape of the vocal tract (due to moving the tongue and the mouth) along with the varying state of the vocal folds that produces speech.

4.2.1.2 Properties of Speech

Different speech sounds are distinguished by the human ear on the basis of their short time spectra and how these spectra evolve with time. The effective bandwidth of speech is approximately 7 kHz. For voiced sounds such as vowels, the vocal tract

acts as a resonant cavity. For most people the resonance frequencies are centered on 500 Hz and its odd harmonics. This resonance produces large peaks in the resulting speech spectrum. These peaks are known as formants. The formants of the speech contain almost all the information contained in the signal. This fact means that the vocal tract can be effectively modeled using an all-pole linear system. Voiced speech will also exhibit the effects of the vibrating vocal cords. The effect of this vibration is to introduce a quasi-periodicity into the speech.

The periodic nature of the speech is clearly visible. The first four formants are labeled in the spectrum. The fine spectral information, due to the vibrating vocal folds, is also visible in the spectrum. It should be noted that the formant structure appears to break down above 4 kHz as noise introduced by the turbulent flow of air through the vocal tract begins to dominate. The spectrum also shows the enormous dynamic range and lowpass nature of voiced speech.

Nasals are produced when the oral cavity is blocked and the velum is lowered to couple the nasal cavity with the vocal tract. They are also voiced in nature but the coupling of the oral and nasal cavities introduces anti-resonances or nulls instead of resonances so the formants disappear.

Unvoiced hiss-like sounds like s, f, sh are generated by constricting the vocal tract close to the lips. Both the time and frequency views demonstrate the noise like nature of the signal. Unvoiced speech tends to have a nearly flat or a highpass spectrum. The formants that are so obvious in voiced speech are gone and so is the fine pitch structure. The energy in the signal is also much lower than that in voiced speech.

4.2.1.3 Speech Perception

The sense of hearing is the least understood component of the human speech communication system. Little is known about how the brain decodes the acoustic information it receives. However, quite a lot is known about the receiver it uses to detect these signals, the ear.

The human ear (depicted in figure 4.2) consists of three main sections, the outer, the middle and the inner ears. The outer ear consists of the ear lobe (pinna) and the external auditory canal. The function of the ear lobe is to channel sounds into the ear and aid in the localization of sounds. The external auditory canal channels the sound into the middle ear. The canal is approximately 2.7 cm in length and is closed at one end by the eardrum. Hence it can be viewed as an acoustic tube that resonates at 3055 Hz [ISP05].

The eardrum is a hard membrane, approximately 0.1 mm thick, which is flexible at the edge. When a sound wave strikes this membrane it vibrates. This vibration is then transferred to the three-bone structure in the middle ear and from there to the inner ear. These bones act as a transformer and match the acoustic impedance

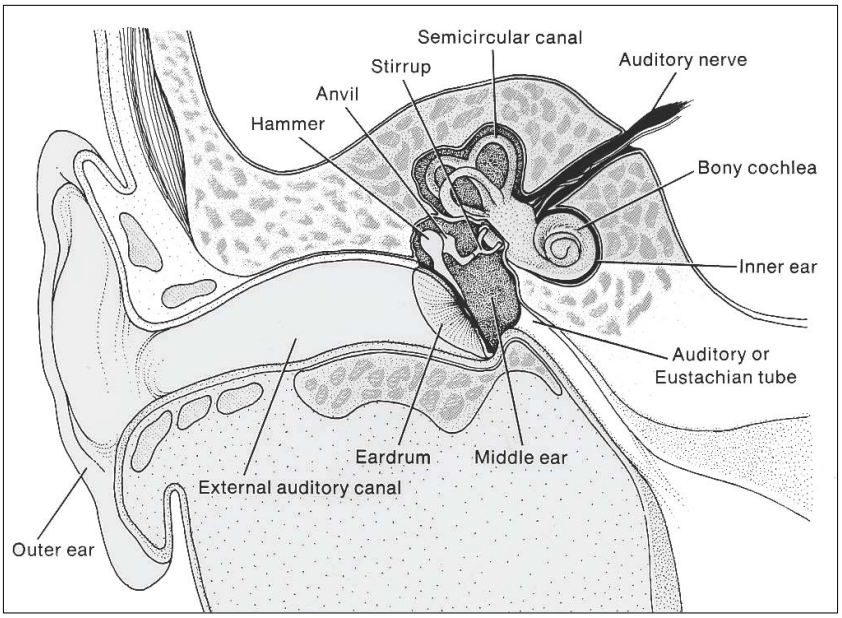


Figure 4.2: The human ear
(http://www.owlnet.rice.edu/~psyc351/)

of the inner ear with that of air. Muscles attached to these bones suppress the vibration if it is too violent and so protect the inner ear. This protection only works for sounds below 2 kHz and it does not work for impulsive sounds. The Eustachian tube connects the middle ear to the vocal tract and removes any static pressure difference between the middle ear and the outer ear. If a significant pressure difference is detected then the Eustachian tube opens and the difference is removed.

The inner ear is composed of the Semicircular canals, the Cochlea, and auditory nerve terminations. The function of the Semicircular canals is to control balance. The Cochlea is fluid filled and helical in shape (it resembles the shell of a snail). Inside the Cochlea there is a hair-lined membrane called the Basilar membrane. This membrane converts the mechanical signal into a neural signal. Different frequencies excite different portions of this membrane allowing a frequency analysis of the signal to be carried out. So the ear is essentially a spectrum analyzer that responds to the magnitude of the signal. The frequency resolution is greatest at low frequencies.

Like any receiver, there is a limit to the sensitivity of the ear. If sounds are too weak, they will not be detected. This is known as the threshold of audibility. This threshold varies with frequency and it can be increased at any given frequency by the presence of a large signal at a nearby lower frequency. This phenomenon is called masking and it is widely used in speech coding. If the quantization noise

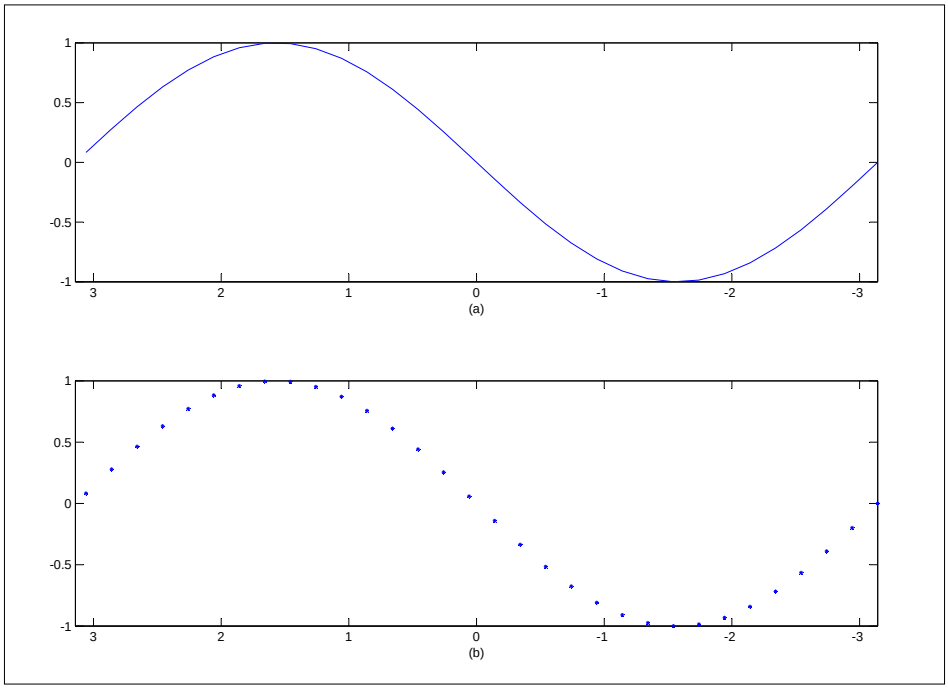


Figure 4.3: The affect of sampling. (a) the continuous signal and (b) the sampled signal

can be concentrated around the formants, then it will be rendered inaudible to the listener.

4.2.2 Sampling

A speech signal is continuous in time. Before it can be processed by digital hardware it must be converted to a signal that is discrete in time. Sampling is a process that converts a continuous time signal into a discrete time signal by measuring the signal at periodic instants in time.

Figure 4.3 shows the effect of sampling on a sinusoidal signal. It is clear that as the number of samples per second (the sampling rate) increases, the sampled signal approximates the continuous signal more closely. In fact, if the sampling rate is high enough the sampled signal will contain all the information that is present in the continuous signal.

The Nyquist sampling theorem states that a signal may be perfectly reconstructed if it is sampled at a rate greater than or equal to twice the frequency of the highest frequency component of the signal.

4.2.2.1 The Nyquist Sampling Theorem

It is necessary to determine under what conditions a continuous signal $g(t)$ may be unambiguously represented by a series of samples taken from the signal at uniform intervals of T . It is convenient to represent the sampling process as that of multiplying the continuous signal $g(t)$ by a sampling signal $s(t)$ which is an infinite train of impulse functions $\delta(nT)$. The sampled signal $g_s(t)$ is:

$$g_s(t) = g(t)s(t)$$

where

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Now the impulse train $s(t)$ is a periodic function and may be represented by a Fourier series:

$$s(t) = \sum_{p=-\infty}^{\infty} c_p e^{jp\omega_0 t}$$

where

$$c_p = \frac{1}{T} \int_{-\frac{T}{2}}^{-\frac{T}{2}} s(t) e^{-jp\omega_0 t} dt = \frac{1}{T}$$

and

$$\omega_0 = \frac{2\pi}{T}$$

$$g_s(t) = g(t) \frac{1}{T} \sum_{p=-\infty}^{\infty} e^{jp\omega_0 t}$$

The spectrum $G_s(\omega)$ of the sampled signal may be determined by taking the Fourier transform of $g_s(t)$. This is most readily achieved by making use of the frequency shift theorem which states that:

$$\text{If } g(t) \rightleftharpoons G(\omega) \text{ then } g(t)e^{j\omega_0 t} \rightleftharpoons G(\omega - \omega_0)$$

Application of this theorem gives:

$$G_s(\omega) = \frac{1}{T} \sum_{p=-\infty}^{\infty} G(\omega - p\omega_0) \tag{4.1}$$

The above equation shows that the spectrum of the sampled signal is simply the sum of the spectra of the continuous signal repeated periodically at intervals of $\omega_0 = \frac{2\pi}{T}$. Thus the continuous signal $g(t)$ may be perfectly recovered from the sampled signal $g_s(t)$ provided that the sampling interval T is chosen so that:

$$\frac{2\pi}{T} > 2\omega_B$$

where ω_B is the bandwidth of the continuous signal $g(t)$. If this condition holds then there is no overlap of the individual spectra in the summation of expression 4.1 and the original signal can be perfectly reconstructed using a lowpass filter $H(\omega)$ with bandwidth ω_B:

$$H(\omega) = \begin{cases} 1, & |\omega| < \omega_B \\ 0, & otherwise \end{cases}$$

This theory underpins the whole of digital audio and means that a digital audio system can in principle be designed which loses none of the information contained in the continuous domain signal $g(t)$. Of course the practicalities are rather different as a band-limited input signal is assumed and the reconstruction filter $H(\omega)$, the A/D and D/A converters must be ideal.

4.2.3 Main Classes of Speech Coding

The central problem in speech coding is to represent the speech signal using as little bits as possible so that quality and intelligibility get damage as little as possible. Speech coding is important in digital mobile phone systems and that's why speech coding methods have advanced considerably in the last 10 years. Thinking commercially, speech coding is the most important application of speech processing field. The requirements for a good speech codec can be:

- quality of speech suffers as little as possible
- the speech is compressed in a small amount of bits
- coding-decoding yields only small delay
- codec is not sensitive to errors in transmission of bits
- coding/decoding is computationally fast
- the codec should perform well with noisy speech (and if possible with other musical signals etc.)
- several consequent encodings should not impair the quality too much

There are no perfect codecs satisfying all the requirements because part of the requirements are contradictory. However by making different compromises, a large number of coding standards for different applications have been developed. For instance in the speech codec of a mobile phone all the requirements above are essential, whereas in recording of speech in databases delay, computational load and error resiliency are inessential - only the quality and good compression ratio counts.

There are plenty of coding methods but they can be divided in roughly three main classes:

- Waveform Coding
- Source Coding
- Hybrid Coding

In waveform coding an effort is made to retain the waveform of the original signal and the coding is based on quantization and removal of redundancies in the waveform. In source coding the parameters of speech (the type of excitation, model of vocal tract, formant frequencies etc.) are coded enabling reconstruction in the decoder. The border line between these classes is volatile, especially in modern synthesis-analysis-codecs in which one tries to reconstruct the waveform of the original speech with good selections of parameters.

In the following will be assumed that the bandwidth of speech is the same as in mobile phone network which is 300-3400 Hz and the sampling frequency is 8 kHz. This is so called narrowband speech. In some application the narrowband network is not used and a higher sampling frequency may be used, for instance in video conferences. In these application the bandwidth is usually 50-7000 Hz and the sampling frequency is 16 kHz. This is so called wideband speech.

4.2.3.1 Waveform Coding

In general, waveform codecs are designed to be signal independent. They are designed to map the input waveform of the encoder into a facsimile-like replica of it at the output of the decoder. Because of this advantage, they can also encode a secondary type of information such as signaling tones, voice band data, or even music. Because of this signal transparency, their coding efficiency is usually quite modest. The coding efficiency can be improved by exploiting some statistical signal properties, if the codec parameters are optimized for the most likely categories of input signals, while still maintaining good quality for other types of signals as well. The waveform codecs can be further subdivided into time-domain waveform codecs and frequency-domain waveform codecs.

Time-Domain Waveform Coding The most well-known representative of signal-independent time-domain waveform coding is the A-law companded pulse code modulation (PCM) scheme. This coding has been standardized by the CCITT at 64 Kbps using non-linear companding characteristics to result in near-constant signal-to-noise ratio (SNR) over the total input dynamic range. More explicitly, the non-linear companding compresses large-input samples and expands small ones. Upon quantizing this companded signal, large-input samples will tolerate higher quantization noise than small samples. Also well-known is the 32 Kbps adaptive differential PCM (ADPCM) scheme standardized in the ITU Recommendation G.721 and the adaptive delta modulation (ADM) arrangement, where usually the most recent signal sample or a linear combination of the last few samples is used to form an estimate of the current one. Then their difference signal, the prediction residual, is computed and encoded with a reduced number of bits, since it has a lower variance than the incoming signal. This estimation process is actually linear prediction with fixed coefficients. However, owing to the non-stationary statistics of speech, a fixed predictor cannot consistently characterize the changing spectral envelope of speech signals. Adaptive predictive coding (APC) schemes utilize two different time-varying predictors to describe speech signals more accurately: a short-term predictor (STP) and a long-term predictor (LTP). The STP is utilized to model the speech spectral envelope, while the LTP is employed to model the line-spectrum-like fine structure representing the voicing information due to quasi-periodic voiced speech. All in all, time-domain waveform codecs treat the speech signal to be encoded as a full-band signal and attempt to map it into as close a replica of the input as possible. The difference among various coding schemes is in their degree and way of using prediction to reduce the variance of the signal to be encoded, so as to reduce the number of bits necessary to represent it.

Frequency Domain Waveform Coding In frequency-domain waveform codecs, the input signal undergoes a more or less accurate short-time spectral analysis. The signal is split into a number of subbands, and the individual subband signals are then encoded by using different numbers of bits in order to obey rate-distortion theory on the basis of their prominence. The various methods differ in their accuracies of spectral resolution and in the bit-allocation principle (fixed, adaptive, semi-adaptive). Two well-known representatives of this class are subband coding (SBC) and adaptive transform coding (ATC).

Waveform coding tries to encode the waveform itself in an efficient way. The signal is stored in such a way that upon decoding, the resulting signal will have the same general shape as the original. Waveform coding techniques apply to audio signals in general and not just to speech as they try to encode every aspect of the signal.

4.2.3.2 Source Coding

The philosophy of vocoders is based on a priori knowledge of the way the speech signal to be encoded was generated at the signal source by a speaker. The air compressed by the lungs excites the vocal cords in two typical modes. When generating voiced sounds, they vibrate and generate a quasi-periodic speech waveform, while in the case of lower energy unvoiced sounds they do not participate in the voice production and the source behaves similarly to a noise generator. The excitation signal denoted by $E(z)$ in z-domain is then filtered through the vocal apparatus, which behaves like a spectral shaping filter with a transfer function of $H(z) = \frac{1}{A(z)}$ that is constituted by the spectral shaping action of the glottis, vocal tract, lip radiation characteristics, and so on. Accordingly, instead of attempting to produce a close replica of the input signal at the output of the decoder, the appropriate set of source parameters is found in order to characterize the input signal sufficiently closely for a given duration of time. First, a decision must be made as to whether the current speech segment to be encoded is voiced or unvoiced. Then the corresponding source parameters must be specified. In the case of voiced sounds, the source parameter is the time between periodic vocal tract excitation pulses, which is often referred to as the pitch p. In the case of unvoiced sounds, the variance or power of the noise-like excitation must be determined. These parameters are quantized and transmitted to the decoder in order to synthesize a replica of the original signal.

The encoder is a simple speech analyzer, determining the current source parameters. After initial speech segmentation, it computes the linear predictive filter coefficients $a_i = 1..p$, which characterize the spectral shaping transfer function $H(z)$. A voiced/unvoiced decision is carried out, and the corresponding pitch frequency and noise energy parameters are determined. These are then quantized, multiplexed, and transmitted to the speech decoder, which is a speech synthesizer.

The associated speech quality of this type of systems may be predetermined by the adequacy of the source model, rather than by the accuracy of the quantization of these parameters. This means that the speech quality of source codecs cannot simply be enhanced by increasing the accuracy of the quantization, that is, the bit rate. Their speech quality is fundamentally limited by the fidelity of the model used. The main advantage of the above vocoding techniques is their low bit rate, with the penalty of relatively low, synthetic speech quality. A well-known representative of this class of vocoders is the 2400 bps American Military Standard LPC-10 codec.

In linear predictive coding (LPC), often more complex excitation models are used to describe the voice-generating source. Once the vocal apparatus has been described by the help of its spectral domain transfer function $H(z)$, the central problem of coding is to decide how to find the simplest adequate excitation for high-quality parametric speech representation. Strictly speaking, this separable model represents a gross simplification of the vocal apparatus, but it provides the only

practical approach to the problem. Vocoding techniques can also be categorized into frequency-domain and time-domain subclasses. However, frequency-domain vocoders are usually more effective than their time-domain counterparts. With source coding (vocoding) typically data rates down to 400 bps are generally realizable.

4.2.3.3 Hybrid Coding

Hybrid coding methods are an attractive tradeoff between waveform coding and source coding, both in terms of speech quality and transmission bit rate, although usually at the price of higher complexity. Every speech coding method, combining waveform and source coding methods in order to improve the speech quality and reduce the bit rate, falls into this broad category. However, adaptive predictive time domain techniques used to describe the human spectral shaping tract, combined with an accurate model of the excitation signal, play the most prominent role in this category. The most important family of hybrid codecs are often referred to as *analysis-by-synthesis* (AbS) codecs.

Waveform coding but also source coding show advantages. With waveform coding the speech quality and thereby intelligibility is very high. However the resulting data rates are so high, that problems can arise while transportation. On the other hand with source coding, which just determines the functional parameters of a speech model and hence data rates are comparably low, the speech loses naturalness and intelligibility which is generally very low. Hybrid coding procedures therefore try to combine the advantages of the both coding alternatives mentioned previously to achieve intelligible speech signals in combination with possibly least data rates. Nowadays relevant schemes came up with data rates between 4 to 16 Kbps generally. Most of these systems make use of linear prediction and differentiate in how the remaining error signal is encoded. The synthesis of the error signal is done with waveform coding procedures, whereby the different hybrid encoders distinguish in accuracy.

Today, the high complexity of hybrid encoders don't depict a big problem in fact of highly integrated semiconductors. Systems exist, which combine a hybrid encoder, a channel coder for robustness, and an encryption module on a single chip.

4.2.4 Transform Coding

Actually transform coding is an advanced form of waveform coding, mainly found for audio (not speech!), image, and video encoding. Three different main transformations excelled and were used in a multitude of research fields. These are the fast-Fourier transform, the discrete-cosine transform and finally the wavelet transform. The main purpose is to change the representation of the original signal form

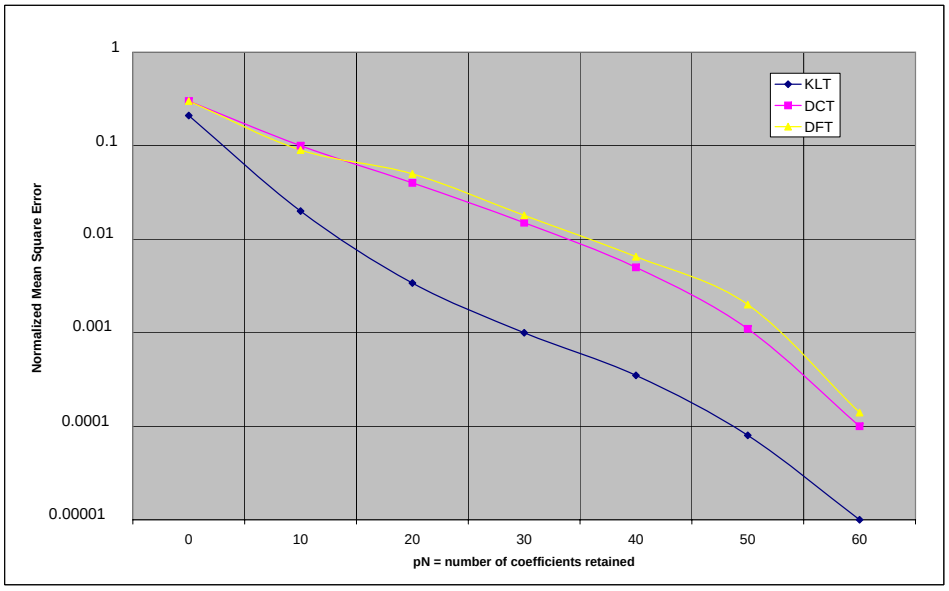


Figure 4.4: Normalized mean square error for different transformations
(Normalized mean square truncation error when percentage p of the $N = 64$ coefficients are retained using transforms on stripes)

time-domain to another domain which is chiefly the frequency-domain to further process the data.

One major difference, which arise while the transformation, between these transforms is the capability to reorganize the energy of the signal into specific, few coefficients. If the whole energy of the signal can be concentrated in just a little amount of coefficients, ideally ordered so that compression can be easily and effectively be done, the resulting data rate could be held low, while maintaining relatively high quality, hence the reconstructed signal is nearly equal to the uncompressed original. This circumstance is illustrated in figure 4.4.

4.2.4.1 Fast-Fourier Transform (FFT)

Linear filtering and Fourier transforms are among the most fundamental operations in digital signal processing. However, their wide use makes their computational requirements a heavy burden in most applications. Direct computation of both convolution and discrete Fourier transform (DFT) requires on the order of N^2 operations where N is the filter length or the transform size. The breakthrough of the Cooley-Tukey FFT comes from the fact that it brings the complexity down to an order of $Nlog_2N$ operations. Because of the convolution property of the DFT,

this result applies to the convolution as well. Therefore, fast Fourier transform algorithms have played a key role in the widespread use of digital signal processing in a variety of applications such as telecommunications, medical electronics, seismic processing, radar or radio astronomy to name but a few.

4.2.4.2 Discrete-Cosine Transform (DCT)

The discrete cosine transform (DCT) is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. It is equivalent to a DFT of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample.

The DCT, and in particular the DCT type II, is often used in signal and image processing, especially for lossy data compression, because it has a strong energy compaction property: most of the signal information tends to be concentrated in a few low-frequency components of the DCT, approaching the Karhunen-Loève transform (which is optimal in the decorrelation sense) for signals based on certain limits of Markov processes.

For example, DCT is used in JPEG image compression, MJPEG video compression, and MPEG video compression. There, the two-dimensional DCT-II of $n \times n$ blocks is computed and the results are quantized and entropy coded. In this case, n is typically 8 and the DCT type II formula is applied to each row and column of the block. The result is an 8×8 transform coefficient array in which the (0,0) element is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial frequencies.

A related transform, the modified discrete cosine transform, or MDCT, is used in AAC, Vorbis, and MP3 audio compression.

4.2.4.3 Karhunen-Loève Transform (KLT)

The KL transform was originally introduced as a series expansion for continuous random processes by Karhunen and Loève. For random sequences Hotelling first studied what was called a method of principal components, which is the discrete equivalent of the KL series expansion. Consequently, the KL transform is also called the Hotelling transform or the method of principal components.

The Karhunen-Loève transform is a mathematical way of determining that linear transformation of a sample of points in L-dimensional space which exhibits the properties of the sample most clearly along the coordinate axes. Along the new axes, the sample variances are extremes (maxima and minima), and uncorrelated. The name comes from the principal axes of an ellipsoid (e.g. the ellipsoid of inertia), which are just the coordinate axes in question.

By their definition, the principal axes will include those along which the point sample has little or no spread (minima of variance). Hence, an analysis in terms of principal components can show (linear) interdependence in data. A point sample of L dimensions for whose L coordinates M linear relations hold, will show only (L-M) axes along which the spread is non-zero. Using a cutoff on the spread along each axis, a sample may thus be reduced in its dimensionality - see [Bis95].

Principal component analysis has in practice been used to reduce the dimensionality of problems, and to transform interdependent coordinates into significant and independent ones. An example used in several particle physics experiments is that of reducing redundant observations of a particle track in a detector to a low-dimensional subspace whose axes correspond to parameters describing the track. In practice, non-linearities of detectors, frequent changes in detector layout and calibration, and the problem of transforming the coordinates along the principal axes into physically meaningful parameters, set limits to the applicability of the method.

The KLT is provably optimal for at least two criteria [Loe60]:

- the KLT decorrelates the data optimally and therefore
- it bundles the information optimally in the lower-frequency region

In terms of energy packing capability, the KLT is optimal in the sense that it distributes the largest amount of signal energy into the direction of the eigenvector of the largest eigenvalue (the direction of largest sample variance), and the second largest amount of signal energy into the second largest eigenvector direction, and so on. Therefore, if one is to choose only k coefficients to best approximate the original signal in L_2 metric, then the optimal choice will be the k coefficients corresponding to the eigenvectors of the k largest eigenvalues.

The two criteria mentioned above are very important for signal compression purposes - but - the KLT is signal dependent and extremely computationally complex comparing to DCT and WT and therefore just conditionally useful for real-time speech encoding respectively speech compression.

4.2.4.4 Wavelet Transform (WT)

Many evolving multimedia applications require transmission of high quality video and audio over the network. One obvious way to accommodate this demand is to increase the bandwidth available to all users. Of course, this *solution* is not without technological and economical difficulties. Another way is to reduce the volume of the data that must be transmitted. There has been a tremendous amount of progress in the field of video/audio compression during the past 10 years. In order to make further progress in video/audio coding, many research groups have begun to use wavelet transforms.

Most of the signals in practice are time-domain signals. When we plot time-domain signals, we obtain a time-amplitude representation of the signal. In many cases, the most important information is hidden in the frequency content of the signal. The frequency spectrum of a signal shows what frequencies exist in the signal.

Fourier transform (FT) and inverse Fourier transform are defined by the following 2 equations:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-2j\pi ft}dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{-2j\pi ft}df.$$

The Fourier transform gives the frequency information of the signal, but it does not tell us when in time these frequency components exist. The information provided by the integral corresponds to all time instances because the integration is done for all time interval. It means that no matter where in time the frequency f appears, it will affect the result of the integration equally. This is why Fourier transform is not suitable for non-stationary signals.

To overcome this weakness, short time Fourier transform (STFT) was developed. In STFT defined below, the signal is divided into small segments which can be assumed to be stationary. In STFT of the signal multiplied by a window function within the Fourier integral. If the window length is infinite, it becomes the FT. In order to obtain the stationarity, the window length must be short enough. The narrower windows affords better time resolution and better stationarity, but at the cost of poorer frequency resolution. One problem with STFT is that one cannot know what spectral components exist at what points of time. One can only know the time intervals in which certain band of frequencies exist.

$$STFT_X^{(\omega)}(t,f) = \int_t [x(t)\omega^*(t-t')]e^{-2j\pi ft}dt$$

The continuous wavelet transform evolved as an alternative approach to STFT to overcome the resolution problem. The wavelet transform is similar to the STFT in that the signal is multiplied by a function similar to windows function in STFT, but the transform is done separately for different segments of the signal. The main differences between the STFT and the CWT is that in CWT, the width of the window is changed as the transform is computed for every single spectral component.

The CWT and the inverse CWT are defined by the following:

$$CWT_x^{\psi}(\tau,s) = \Psi_x^{\psi}(\tau,s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^* \left(\frac{t-\tau}{s}\right) dt$$

$$x(t) = \frac{1}{c_\psi^2} \int_s \int_\tau \Psi_x^\psi(\tau, s) \frac{1}{s^2} \psi \left(\frac{t-\tau}{s} \right) d\tau ds.$$

The transformed signal is a function of two variables τ and s, the translation and the scale parameter, respectively. $\Psi(t)$ is the mother wavelet. The term translation refers to the location of the window. As the window is shifted through the signal, time information in the transform domain is obtained. s is the scale parameter which is defined as $s = \frac{1}{frequency}$. High scales give a global information of a signal (that usually spans the entire signal), whereas low scales give a detailed information of a hidden pattern in the signal (that usually lasts a relatively short time). In practical applications, low scales (high frequencies) do not last for long, but they usually appear from time to time as short bursts. High scales (low frequencies) usually last for the entire duration of the signal.

The CWT of a signal, $x(t)$, is computed in the following way. All the windows that are used are the dilated (or compressed) and shifted versions of the mother wavelet. A mother wavelet is a function that satisfies certain orthonormality and compact support requirements, and there exists an infinite number of such functions. Once the mother wavelet is chosen, the computation starts with $s = 1$ and the CWT is computed for all values of s. For convenience, the first value of s will correspond to the most compressed wavelet. As the value of s is increased, the wavelet will dilate.

The wavelet is placed at the beginning of the signal. The wavelet function at scale 1 is multiplied by the signal and then integrated over all times. The result of the integration is then normalized. The energy normalization is used so that the transformed signal will have the same energy at every scale. The resulting value is the value of the transformation. The wavelet at scale $s = 1$ is then shifted to $t = \tau$. This procedure is repeated for all such that all parts of the signal are covered by the mother wavelet. Now s is increased by a small value then the process repeats.

If the signal has a spectral component that corresponds to the current value of s, the product of the wavelet with the signal gives a relatively large value. If the spectral component that corresponds to the current value of s is not in the signal, the value will be relatively small. For example, the signal in figure 4.5 has spectral components corresponding to s=1 around $t = 125ms$. The CWT of the signal in figure 4.5 will give large values for low scales around $t = 125ms$ and small values elsewhere. For high scales, it will give large values for almost the entire duration of the signal because low frequencies exist at all times. As the window width increases, the transform starts picking up the lower frequency components.

The relationship between time and frequency can be seen in figure 4.6. Each box corresponds to a value of the wavelet transform in the time-frequency plane. All the points in the time-frequency plane that is in a box is one value of the WT. The area is constant although the widths and heights of the boxes change. Each

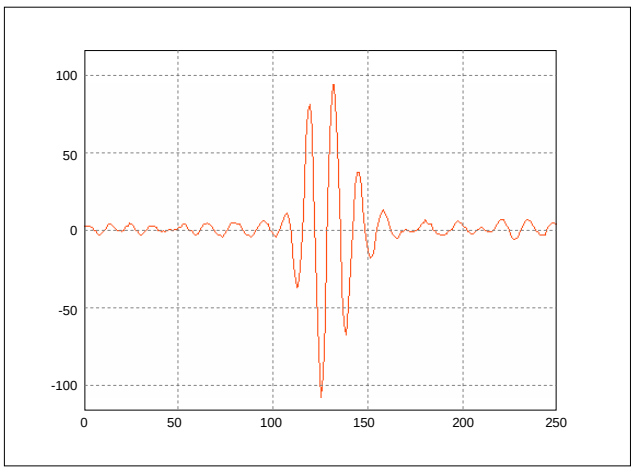


Figure 4.5: Continous Wavelet Transform: $s = 1$ and $t = 125ms$

box represents an equal portion of the time-frequency plane but represents different proportions of time and frequency. At low frequencies, the heights of the boxes are shorter (corresponding to good frequency resolution), but their widths are wider (which means poor time resolution, because of the more ambiguity regarding the value of the exact time), and vice versa. The Heisenberg's uncertainty principle states that we cannot decrease the area of the box beyond some minimum value. For a given mother wavelet, the dimensions of the boxes can be changed, while keeping the area the same, and this is where the power of the wavelets lie.

Although the discretized continuous wavelet transform enables the computation of the continuous wavelet transform by computers, it is not a true discrete transform. As a matter of fact, the wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal is concerned. This redundancy, on the other hand, requires a significant amount of computation time and resources. The discrete wavelet transform (DWT), on the other hand, provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time. The DWT is considerably easier to implement when compared to the CWT.

The main idea with the DWT is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. Recall that the CWT is a correlation between a wavelet at different scales and the signal with the scale (or the frequency) being used as a measure of similarity. The continuous wavelet transform was computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used

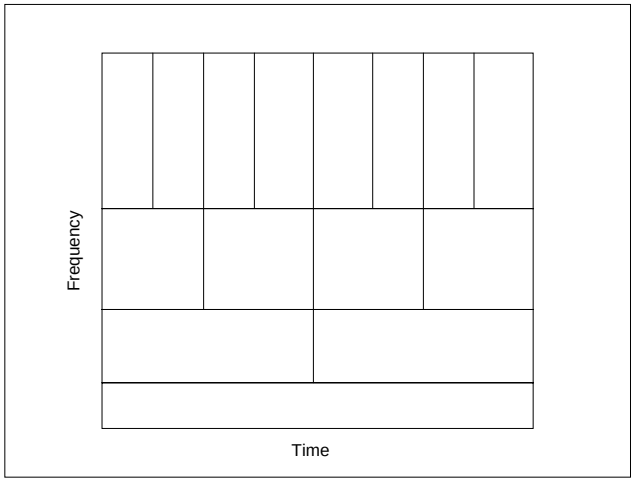


Figure 4.6: Wavelet Transform: relationship between time and frequency

to analyze the signal at different scales. The signal is passed through a series of highpass filters to analyze the high frequencies, and it is passed through a series of lowpass filters to analyze the low frequencies.

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by upsampling and downsampling (subsampling) operations. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor n reduces the number of samples in the signal n times. Upsampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal. For example, upsampling by two refers to adding a new sample, usually a zero or an interpolated value, between every two samples of the signal. Upsampling a signal by a factor of n increases the number of samples in the signal by a factor of n.

Although it is not the only possible choice, DWT coefficients are usually sampled from the CWT on a dyadic grid, i.e., $s_0 = 2$ and $\tau_0 = 1$, yielding $s = 2^j$ and $\tau = k2^j$. Since the signal is a discrete time function, the terms function and sequence will be used interchangeably in the following discussion. This sequence will be denoted by $x[n]$, where n is an integer.

The procedure starts with passing this signal (sequence) through a half band digital lowpass filter with impulse response $h[n]$. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k].$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 1000 Hz component, then half band lowpass filtering removes all the frequencies above 500 Hz.

The unit of frequency is of particular importance at this time. In discrete signals, frequency is expressed in terms of radians. Accordingly, the sampling frequency of the signal is equal to 2π radians in terms of radial frequency. Therefore, the highest frequency component that exists in a signal will be π radians, if the signal is sampled at Nyquist's rate (which is twice the maximum frequency that exists in the signal); that is, the Nyquist's rate corresponds to π rad/s in the discrete frequency domain. Therefore using Hz is not appropriate for discrete signals. However, Hz is used whenever it is needed to clarify a discussion, since it is very common to think of frequency in terms of Hz. It should always be remembered that the unit of frequency for discrete time signals is radians.

After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of $\pi/2$ radians instead of π radians. Simply discarding every other sample will subsample the signal by two, and the signal will then have half the number of points. The scale of the signal is now doubled. Note that the lowpass filtering removes the high frequency information, but leaves the scale unchanged. Only the subsampling process changes the scale. Resolution, on the other hand, is related to the amount of information in the signal, and therefore, it is affected by the filtering operations. Half band lowpass filtering removes half of the frequencies, which can be interpreted as losing half of the information. Therefore, the resolution is halved after the filtering operation. However, the subsampling operation after filtering does not affect the resolution, since removing half of the spectral components from the signal makes half the number of samples redundant anyway. Half the samples can be discarded without any loss of information. In summary, the lowpass filtering halves the resolution, but leaves the scale unchanged. The signal is then subsampled by 2 since half of the number of samples are redundant. This doubles the scale. This procedure can mathematically be expressed as:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n-k].$$

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information.

DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with lowpass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal $x[n]$ is first passed through a halfband highpass filter $g[n]$ and a lowpass filter $h[n]$. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of $\pi/2$ radians instead of π . The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_n x[n] \cdot g[2k - n]$$

$$y_{low}[k] = \sum_n x[n] \cdot h[2k - n]$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure, which is also known as the subband coding, can be repeated for further decomposition. At every level, the filtering and subsampling will result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence double the frequency resolution).

Figure 4.7 illustrates this procedure, where $x[n]$ is the original signal to be decomposed, and $h[n]$ and $g[n]$ are lowpass and highpass filters, respectively. The bandwidth of the signal at every level is marked on the figure as f.

Supposed that the original signal $x[n]$ has 512 sample points, spanning a frequency band of zero to π rad/s. At the first decomposition level, the signal is passed through the highpass and lowpass filters, followed by subsampling by 2. The output of the highpass filter has 256 points (hence half the time resolution), but it only spans the frequencies $\pi/2$ to π rad/s (hence double the frequency resolution). These 256 samples constitute the first level of DWT coefficients. The output of the lowpass filter also has 256 samples, but it spans the other half of the frequency band, frequencies from 0 to $\pi/2$ rad/s. This signal is then passed through the same lowpass and highpass filters for further decomposition. The output of the second lowpass filter followed by subsampling has 128 samples spanning a frequency band of 0 to $\pi/4$ rad/s, and the output of the second highpass filter followed by subsampling has 128 samples spanning a frequency band of $\pi/4$ to $\pi/2$ rad/s. The second

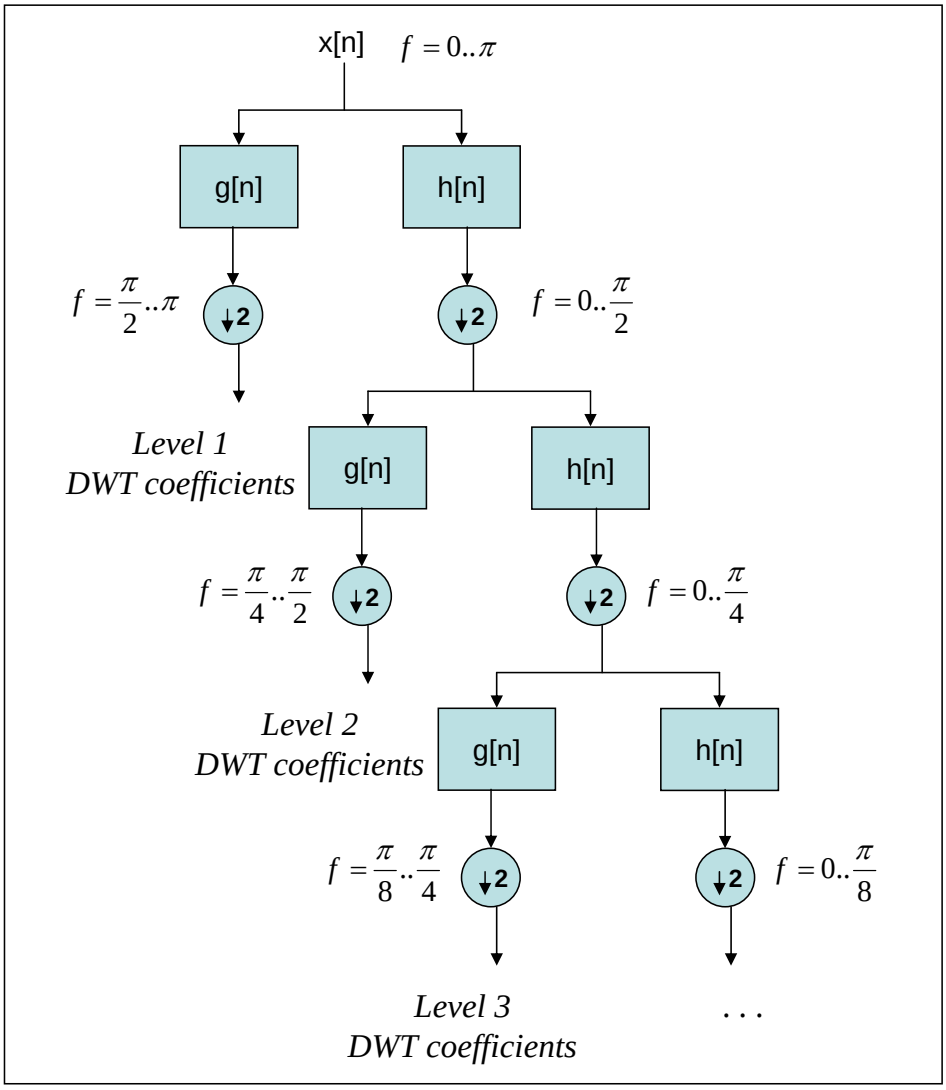


Figure 4.7: Wavelet Transform: n-level decomposition

highpass filtered signal constitutes the second level of DWT coefficients. This signal has half the time resolution, but twice the frequency resolution of the first level signal. In other words, time resolution has decreased by a factor of 4, and frequency resolution has increased by a factor of 4 compared to the original signal. The lowpass filter output is then filtered once again for further decomposition. This process continues until two samples are left. For this specific example there would be 8 levels of decomposition, each having half the number of samples of the previous level. The DWT of the original signal is then obtained by concatenating all coefficients starting from the last level of decomposition (remaining two samples, in this case). The DWT will then have the same number of coefficients as the original signal.

The frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies. The difference of this transform from the Fourier transform is that the time localization of these frequencies will not be lost. However, the time localization will have a resolution that depends on which level they appear. If the main information of the signal lies in the high frequencies, the time localization of these frequencies will be more precise, since they are characterized by more number of samples. If the main information lies only at very low frequencies, the time localization will not be very precise, since few samples are used to express signal at these frequencies. This procedure in effect offers a good time resolution at high frequencies, and good frequency resolution at low frequencies.

The frequency bands that are not very prominent in the original signal will have very low amplitudes, and that part of the DWT signal can be discarded without any major loss of information, allowing data reduction. Figure 4.8a shows a typical 512-sample signal that is normalized to unit amplitude. The horizontal axis is the number of samples, whereas the vertical axis is the normalized amplitude. Figure 4.8b shows the 8 level DWT of the signal in figure 4.8a. The last 256 samples in this signal correspond to the highest frequency band in the signal, the previous 128 samples correspond to the second highest frequency band and so on. It should be noted that only the first 128 samples, which correspond to lower frequencies of the analysis, carry relevant information and the rest of this signal has virtually no information. Therefore, all but the first 128 samples can be discarded without any loss of information. This is how DWT provides a very effective data reduction scheme.

Interpreting the DWT coefficients can sometimes be rather difficult because the way DWT coefficients are presented is rather peculiar. To make a real long story real short, DWT coefficients of each level are concatenated, starting with the last level. An example is in order to make this concept clear:

Supposed we have a 256-sample long signal sampled at 10 MHz and we wish to obtain its DWT coefficients. Since the signal is sampled at 10 MHz, the highest frequency component that exists in the signal is 5 MHz. At the first level, the signal

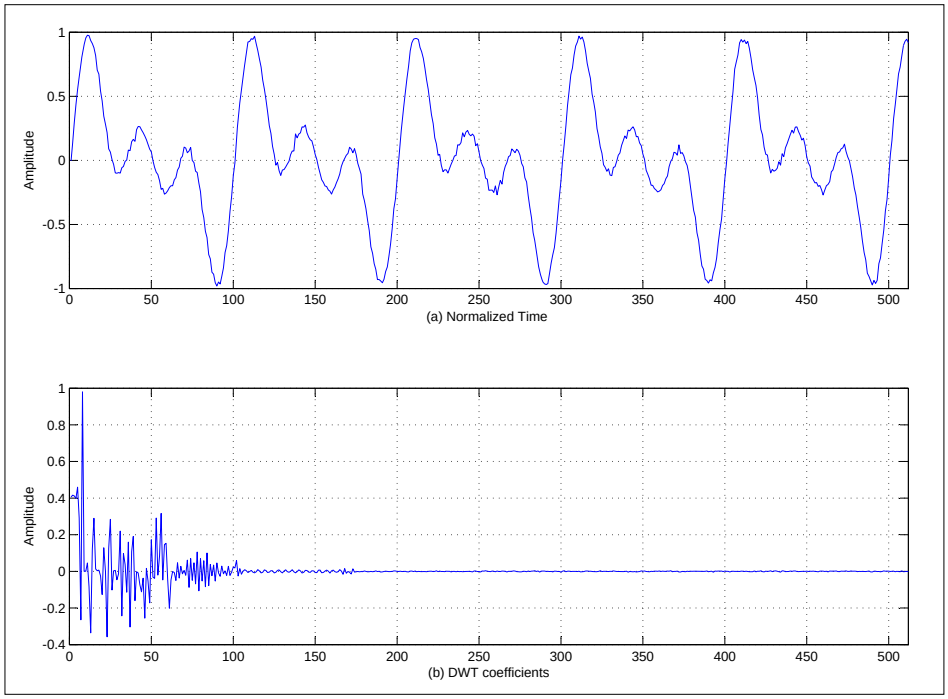


Figure 4.8: 512-sample signal

is passed through the lowpass filter $h[n]$, and the highpass filter $g[n]$, the outputs of which are subsampled by two. The highpass filter output is the first level DWT coefficients. There are 128 of them, and they represent the signal in the [2.5 5] MHz range. These 128 samples are the last 128 samples plotted. The lowpass filter output, which also has 128 samples, but spanning the frequency band of [0 2.5] MHz, are further decomposed by passing them through the same $h[n]$ and $g[n]$. The output of the second highpass filter is the level 2 DWT coefficients and these 64 samples precede the 128 level 1 coefficients in the plot. The output of the second lowpass filter is further decomposed, once again by passing it through the filters $h[n]$ and $g[n]$. The output of the third highpass filter is the level 3 DWT coefficients. These 32 samples precede the level 2 DWT coefficients in the plot.

The procedure continues until only 1 DWT coefficient can be computed at level 9. This one coefficient is the first to be plotted in the DWT plot. This is followed by 2 level 8 coefficients, 4 level 7 coefficients, 8 level 6 coefficients, 16 level 5 coefficients, 32 level 4 coefficients, 64 level 3 coefficients, 128 level 2 coefficients and finally 256 level 1 coefficients. Note that less and less number of samples is used at lower frequencies, therefore, the time resolution decreases as frequency decreases, but since the frequency interval also decreases at low frequencies, the frequency

<table>
<tr><td colspan="4">x</td></tr>
<tr><td colspan="2">Hx</td><td colspan="2">Gx</td></tr>
<tr><td>H^2x</td><td>GHx</td><td>HGx</td><td>G^2x</td></tr>
</table>

Figure 4.9: Second level wavelet packet decomposition

resolution increases. Obviously, the first few coefficients would not carry a whole lot of information, simply due to greatly reduced time resolution.

4.2.4.5 Wavelet Packet Transform (WPT)

For many signals of physical origin - especially those that are discontinous or transient - the WT processes time-frequency localization that is superior to the STFT due to its dyadic tiling. Its time-frequency tiling is, nonetheless, still fixed. The wavelet packet transform (WPT) permit an adaptive time-frequency tiling. This decomposition produce an overcomplete set of subspaces, making it possible to select one of many coordinate systems with which the signals may be *viewed*. Using a cost function designed for a specific signal processing goal (for example, compression or classification) the best basis can be chosen to optimize the coordinate system with respect to frequency via wavelet packet bases.

The wavelet packet transform [Wic00] is a generalized version of the wavelet transform: it retains not only the low but also the high frequency subband, performing a decomposition upon both at each stage. As a result, the tiling of the time-frequency plane is configurable: the partitioning of the frequency axis may take many forms to suit the needs of the application.

Starting with a signal x of length N samples, the first level of decomposition generates the lowpass and highpass subbands Hx and Gx respectively - as with the wavelet transform, each of the half length of x. The second level decomposition generates four subsequences: H^2x, GHx, HGx, G^2x halved in length again; the decomposition to this level is shown in figure 4.9.

This procedure is repeated J times, where $J \leq log_2 N$, resulting in JN coefficients. The computational cost of this decomposition is on the order of $O(JN) \leq O(Nlog_2 N)$. This iterative process generates a *binary wavelet packet tree* structure where the nodes of the tree represent subspaces with different frequency localization characteristics. Considering unbalanced wavelet packet trees frequency localization could be optimized regarding the mapping to bark scaled frequency localization in order to support psychoacoustic modeling as can be seen later in this work - see chapter 8.

The result of the wavelet packet transform is not ordered by increasing frequency. If each node in the level basis is numbered with a sequential binary grey code value $(0000, 0001, 0011, 0010, 0110, ...)$ the nodes can be ordered by frequency by sorting

them into decimal order (0001, 0010, 0011, ...) [Kap02]. In the standard wavelet packet transform the result of the scaling function (the lowpass filter) is placed in the lower half of the array and the result of the wavelet function (the highpass filter) is placed in the upper half of the array. The wavelet packet algorithm recursively applies the wavelet transform to the high and lowpass result at each level, generating two new filter results which have half the number of elements.

If we think of the lower and upper halves of the array that results from the wavelet transform as two children in the wavelet packet tree, a frequency ordered wavelet packet result can be calculated inverting the location of the filter results in the left hand child. The standard wavelet transform is applied to the left child (e.g., the result of the lowpass filter H is placed in the lower half of the result array and the result of the highpass filter G is placed in the upper half of the array. A modified wavelet transform is applied to the right child. Here the result of the highpass filter G is placed in the lower half of the array and the result of the lowpass filter H is placed in the upper half of the array. Each recursive step generates two more children, where the standard transform is applied to the left child and the modified transform is applied to the right child. The best time-frequency resolution is obtained by taking a level basis through the tree which results in a square matrix.

The modified version of the wavelet packet transform does indeed produce a frequency ordered result [JlCH01]. The fact that this is the case is not immediately obvious. Assumed that the wavelet transform is a perfect filter and that we have a signal that ranges from 0 to 64 Hz. The lowpass scaling function H results in exactly the lower half of the frequency spectrum 0 to 32 Hz and the highpass wavelet function results in exactly the upper half of the spectrum 32 to 64Hz. After the first transform step, the lowpass result is on the left branch of the tree and the highpass result is on the right branch. The usual filter pair (H, G) is applied to the left branch and the inverse filter pair (G, H) is applied to the right branch. As the modified wavelet packet tree is recursively built this way, it is not obvious that the result is frequency ordered.

4.2.5 Procedures and Limitations

To effectively transport speech information in communication networks, various speech coding schemes and procedures have been invented. Fundamental approaches will be briefly discussed in the following sections.

4.2.5.1 PCM

PCM (Pulse Code Modulation) codes continuous time speech signal $s_a(t)$ with uniform sampling frequency into samples $s(n)$, where also quantization is uniform. If R bits are reserved for each sample, 2^R different amplitude levels can be coded.

One can think of these amplitude levels being in a codebook and every sample is represented as a binary number index in the codebook entry which best matches the original sample. The original non-quantized signal $s_a(n)$ can be written as

$$s_a(n) = s(n) + q(n),$$

where $s(n)$ is the quantized signal and $q(n)$ is the quantization error. Assumed, that the dynamics of the signal is limited in $[-1, 1]$, the distribution of the quantization error can be modeled with a rectangular distribution $f_q(\xi) = \frac{1}{\delta}, -\delta/2 \leq \xi \leq \delta/2$, where $\delta = 2^{1-R}$ is the quantization step. The average power of the error in the uniform quantizer is

$$\begin{aligned} E\left\{q(n)^2\right\} &= \int_{\delta/2}^{\delta/2} \frac{1}{\delta} x^2 dx \\ &= \frac{1}{\delta}\frac{1}{3}\left[(\delta/2)^3 - (-\delta/2)^3\right] \\ &= \frac{\delta^2}{24} \\ &= \frac{2^{-2R}}{6}. \end{aligned}$$

What is the power of the original signal? It depends on the signal but in the best case all the samples have amplitude ± 1 giving average power 1. The quality of the quantized signal is described by (signal/noise-ratio, SNR), which is defined

$$SNR = 10\, log_{10}\, \frac{S}{E}(dB),$$

where S is the power of the original signal, E is the power of the noise, and unit is decibel (dB). In this case

$$\begin{aligned} SNR &= 10\, log_{10}\, \frac{1}{\frac{2^{-2R}}{6}} \\ &= 10\, R\, log_{10}(4) + 10\, log_{10}(6) \\ &= 6.02\, R + 7.78. \end{aligned}$$

In other words by adding one bit in PCM improves the SNR by 6 dB. In speech quantization a non-uniform quantizer is more profitable, partly because there are mainly lower amplitude in speech, partly because the human ear is more sensitive to the ratios of amplitudes rather than their differences. If in general the amplitude

distribution is known, then an optimal quantizer so called *Lloyd-Max quantizer* to a given error function can be obtained.

The non-uniform quantizer is realized in practice with a non-linear transformation preceding a uniform quantizer and by applying an inverse transformation after the quantization. This method is called *companding* (compression, expanding). In Europe in line telephony phone networks *A-law*-compression is used transforming a sample x as follows:

$$y = \begin{cases} \frac{Ax}{1+log_{10}\,A} & ,0 \leq |x| \leq \frac{1}{A} \\ sgn(x)\frac{1+\,log_{10}(A|x|)}{1+log_{10}(A)} & \frac{1}{A} \leq |x| \leq 1 \end{cases},$$

where $A = 87.56$. In northern America μ-law-compression is used, which is based on similar logarithmic compression:

$$y = sgn(x)\frac{log(1+\mu\,|x|)}{log(1+\mu)},$$

where $\mu = 255$. The inverse transforms (expanding) can be obtained by solving y in terms of x in the equations above. By using non-uniform quantization with 8 bits per sample corresponds roughly uniform quantization with 12 bit per sample in subjective speech quality.

4.2.5.2 Adaptive PCM

The quantization step can be adapted according to the signal; this method is called adaptive quantization. The key idea is to shrink the step when signal level is small when the quantization error is also small. Correspondingly for large amplitude the step is enlarged. Adaptive quantization is particularly useful in low bit rate (< 8) quantization of non-stationary signals. A general method to adapt the quantization step $\Delta(n)$ is:

$$\Delta(n+1) = \Delta(n)M(x(n)),$$

where $M()$ is a fixed function and $x(n)$ is the quantized value of speech on time instant n. The function $M(n)$ can be determined with test material. The good thing in this method is that the quantization levels can be restored without explicit knowledge of changes of quantization steps. At the step n the quantization step is $\Delta(n)$ and the quantized sample(result) is $x(n)$ which is sent to the receiver. The next quantization step $\Delta(n+1)$ depends only on the previous quantization step $\Delta(n)$ (known in the decoder already), the sample $x(n)$ (also known), thus it can be calculated also in the decoder. This is an important synchronization feature between the coder and the decoder also present in much more complicated codecs - the state of the decoder should be the same as in the coder. If possible the errors

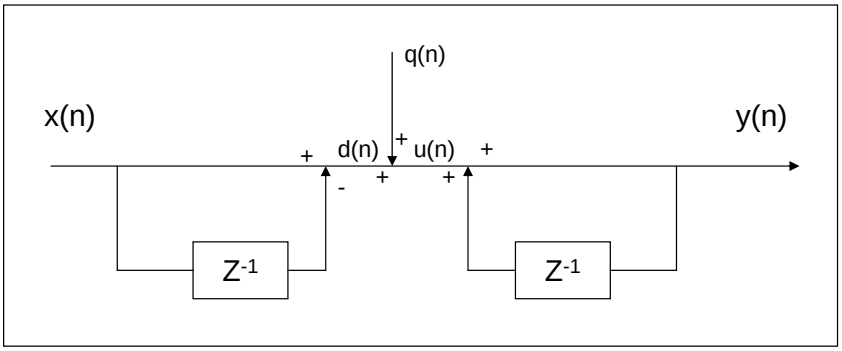


Figure 4.10: Open DPCM-quantizer
(where the prediction is $\hat{s}(n) = s(n-1)$. Quantization noise has been modeled as an additive noise)

in the transmission should cause only temporary divergence in states synchronizing quickly into same state.

4.2.5.3 DPCM

Instead of sending the samples, only differences between the samples are transmitted. This should work if the differences are smaller on average that the samples, implying smaller dynamic range, implying that the quantization steps with the same amount of bits are smaller which naturally attenuates the quantization noise. In fact the power σ_e^2 of the signal $s(n) - s(n-1)$ is:

$$\sigma_e^2 = E\left\{[s(n) - s(n-1)]^2\right\} = 2\sigma_s^2\left[1 - \frac{r_s(1)}{r_s(0)}\right],$$

where σ_s^2 is the power of speech and $r(k)$ is the autocorrelation of speech at lag k. Thus σ_e^2 is smaller than σ_s^2 if $\frac{r_s(1)}{r_s(0)} > 0.5$. For speech in general $\frac{r_s(1)}{r_s(0)} > 0.85$

DPCM-quantization where the difference of the input is computed and sent is illustrated in figure 4.10. Quantization noise is modeled by addition of quantization noise. In the receiver the differences are correspondingly summed.

This method has a serious disadvantage. The quantization noise will be summed in the receiver. One can obtain from the figure:

$$\begin{aligned} D(z) &= X(z)(1 - z^{-1}), \\ U(z) &= D(z) + Q(z), \\ Y(z) &= U(z) + z^{-1}Y(z). \end{aligned}$$

By substituting further

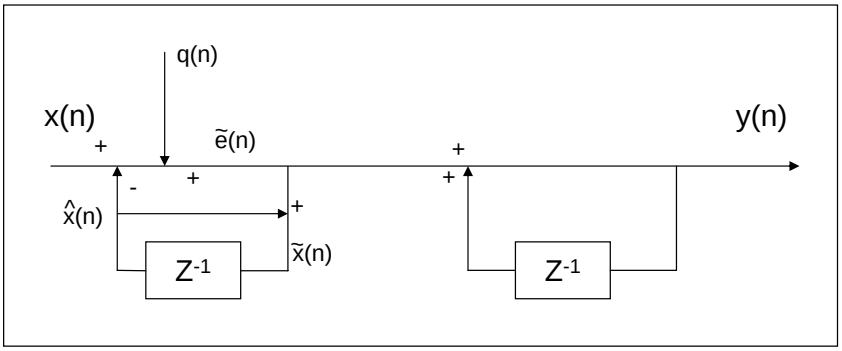


Figure 4.11: Closed DPCM-quantizer
(where predictor is z^{-1}. Quantization noise will not add up in the receiver)

$$\begin{aligned} Y(z) &= \frac{U(z)}{1-z^{-1}} \\ &= \frac{D(z)+Q(z)}{1-z^{-1}} \\ &= \frac{X(z)(1-z^{-1})+Q(z)}{1-z^{-1}} \\ &= X(z)+\frac{Q(z)}{1-z^{-1}}. \end{aligned}$$

The impulse response of the filter $\frac{1}{1-z^{-1}}$ in unit-step function thus the quantization noise sums and its influence in the signal can in principle grow without limits (except saturation will stop this). A system like this where quantization is based in the original signal is called *open-loop* system. The alternative is a closed-loop system, in which the quantization is carried out by calculating the difference between quantized signal and the next sample. In the *closed-loop* systems the quantization noise will not be summed in the receiver because in the coder the quantization value is selected to minimize the actual error in the decoder. This is fundamental principle in synthesis-analysis-codecs to be described later. The closed-loop version of DPCM is in the figure 4.11.

To be calculated is the transfer function of the closed quantizer. From the figure

$$\begin{aligned} \tilde{E}(z) &= X(z)-\hat{X}(z)+Q(z), \\ \tilde{X}(z) &= z^{-1}\tilde{X}(z)+\tilde{E}(z) \end{aligned}$$

which means

$$\begin{aligned}\tilde{X}(z) &= \frac{\tilde{E}(z)}{1-z^{-1}},\\ \hat{X}(z) &= z^{-1}\tilde{X}(z),\\ Y(z) &= \frac{\tilde{E}(z)}{1-z^{-1}},\end{aligned}$$

with $\tilde{E}(z)$ the quantized residual error, $\tilde{X}(z)$ the reconstructed signal and $\hat{X}(z)$ the prediction according to reconstruction. From the formulae above one can see that $Y(z) = \tilde{X}(z)$ which means that the decoder is a part of the encoder. So $\tilde{X}(z)$ is:

$$\begin{aligned}\tilde{X}(z) &= \frac{\tilde{E}(z)}{1-z^{-1}}\\ &= \frac{X(z)-\hat{X}(z)+Q(z)}{1-z^{-1}}\\ &= \frac{X(z)-z^{-1}\tilde{X}(z)+Q(z)}{1-z^{-1}}\end{aligned}$$

which implies

$$\begin{aligned}\tilde{X}(z) &= \frac{1}{1+\frac{z^{-1}}{1-z^{-1}}}\frac{X(z)+Q(z)}{1-z^{-1}}\\ &= (1-z^{-1})\frac{X(z)+Q(z)}{1-z^{-1}}\\ &= X(z)+Q(z),\end{aligned}$$

as it was supposed to.

The next improvement to DPCM-codec is to use more sophisticated predictor instead is simple difference. The filter z^{-1} can be interpreted as predictor $\hat{x} = x(n-1)$, where the the next sample is predicted to be same as the previous. In the closed case $\hat{x} = \tilde{x}(n-1)$. By using a better predictor the prediction error will be smaller and the quantization can be realized with fewer bits. According to normal equations the problem is to determine the autocorrelation of speech, which can be estimated from a speech database. Even a second degree predictor reduces noise by approximately 6 dB, thus giving the same SNR as by sending 1 bit per sample less data (which is 8000 bits per second).

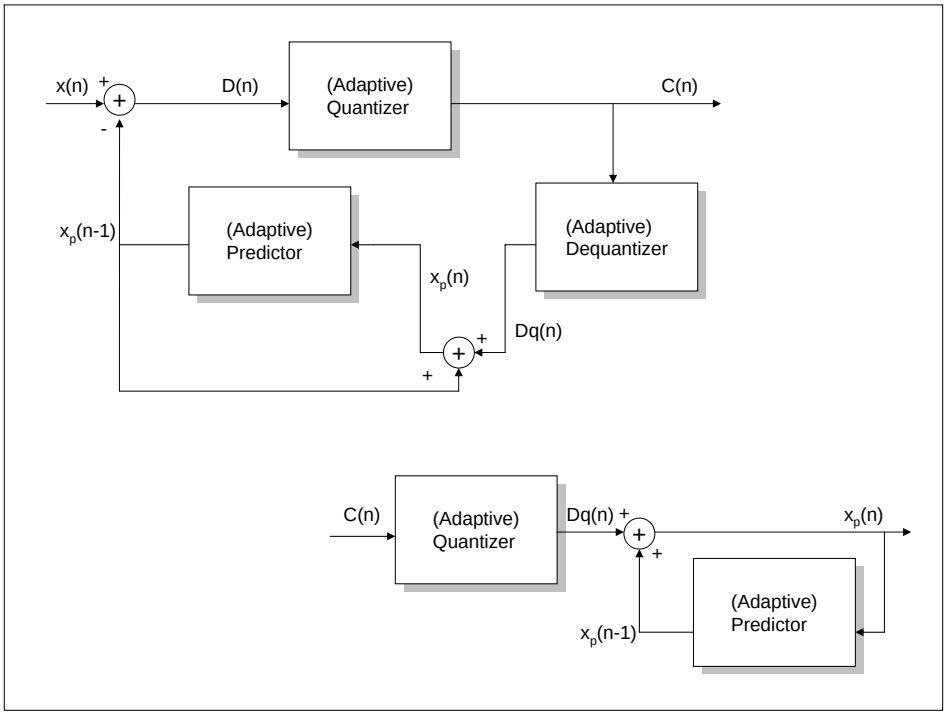


Figure 4.12: ADPCM compression and decompression

4.2.5.4 ADPCM

Figure 4.12 shows a simplified block diagram of an adaptive differential pulse code modulation (ADPCM) coder [RS78]. To be clear, the figure omits details such as bit-stream formatting, the possible use of side information, and the adaptation blocks. The ADPCM coder takes advantage of the fact that neighboring audio samples are generally similar to each other. Instead of representing each audio sample independently as in PCM, an ADPCM encoder computes the difference between each audio sample and its predicted value and outputs the PCM value of the differential. The ADPCM encoder uses most of the components of the ADPCM decoder to compute the predicted values.

The quantizer output is generally only a (signed) representation of the number of quantizer levels. The re-quantizer reconstructs the value of the quantized sample by multiplying the number of quantizer levels by the quantizer step size and possibly adding an offset of half a step size. Depending on the quantizer implementation, this offset may be necessary to center the re-quantized value between the quantization thresholds.

The ADPCM coder can adapt to the characteristics of the audio signal by chang-

ing the step size of either the quantizer or the predictor, or by changing both. The method of computing the predicted value and the way the predictor and the quantizer adapt to the audio signal vary among different ADPCM coding systems.

Some ADPCM systems require the encoder to provide side information with the differential PCM values. This side information can serve two purposes. First, in some ADPCM schemes the decoder needs the additional information to determine either the predictor or the quantizer step size, or both. Second, the data can provide redundant contextual information to the decoder to enable recovery from errors in the bit stream or to allow random access entry into the coded bit stream.

The following section describes the ADPCM algorithm proposed by the Interactive Multimedia Association (IMA). This algorithm offers a compression factor of (number of bits per source sample)/4 to 1. Other ADPCM audio compression schemes include the CCITT Recommendation G.721 (32 kilobits per second compressed data rate) and Recommendation G.723 (24 kilobits per second compressed data rate) standards and the compact disc interactive audio compression algorithm [NAS86, TYYA89].

The IMA is a consortium of computer hardware and software vendors cooperating to develop a de facto standard for computer multimedia data. The IMA's goal for its audio compression proposal was to select a public-domain audio compression algorithm able to provide good compressed audio quality with good data compression performance. In addition, the algorithm had to be simple enough to enable software-only, real-time decompression of stereo, 44.1-kHz-sampled, audio signals on a 20-MHz 386-class computer. The selected ADPCM algorithm not only meets these goals, but is also simple enough to enable software-only, real-time encoding on the same computer.

The simplicity of the IMA ADPCM proposal lies in the crudity of its predictor. The predicted value of the audio sample is simply the decoded value of the immediately previous audio sample. Thus the predictor block in figure 4.12 is merely a time-delay element whose output is the input delayed by one audio sample interval. Since this predictor is not adaptive, side information is not necessary for the reconstruction of the predictor.

Figure 4.13 shows a block diagram of the quantization process used by the IMA algorithm. The quantizer outputs four bits representing the signed magnitude of the number of quantizer levels for each input sample.

Adaptation to the audio signal takes place only in the quantizer block. The quantizer adapts the step size based on the current step size and the quantizer output of the immediately previous input. This adaptation can be done as a sequence of two table lookups. The three bits representing the number of quantizer levels serve as an index into the first table lookup whose output is an index adjustment for the second table lookup. This adjustment is added to a stored index value, and the range-limited result is used as the index to the second table lookup. The summed index

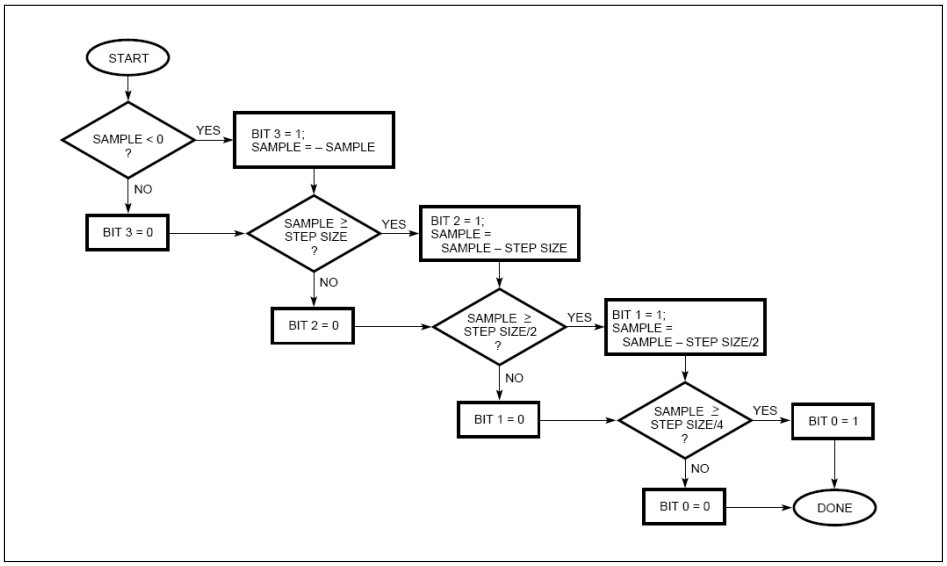


Figure 4.13: IMA ADPCM quantization [Pan93]

value is stored for use in the next iteration of the step-size adaptation. The output of the second table lookup is the new quantizer step size. Note that given a starting value for the index into the second table lookup, the data used for adaptation is completely deducible from the quantizer outputs; side information is not required for the quantizer adaptation. Figure 4.14 illustrates a block diagram of the step-size adaptation process - see [Pan93] for the lookup tables contents.

A fortunate side effect of the design of this ADPCM scheme is that decoder errors caused by isolated code word errors or edits, splices, or random access of the compressed bit stream generally do not have a disastrous impact on decoder output. This is usually not true for compression schemes that use prediction. Since prediction relies on the correct decoding of previous audio samples, errors in the decoder tend to propagate. The next section explains why the error propagation is generally limited and not disastrous for the IMA algorithm.

The decoder reconstructs the audio sample, $x_p(n)$, by adding the previously decoded audio sample, $x_p(n-1)$ to the result of a signed magnitude product of the code word, $C(n)$, and the quantizer step size plus an offset of one-half step size:

$$x_p(n) = x_p(n-1) + \text{stepsize}(n) \times C'(n)$$

where $C'(n)$ equals one-half plus a suitable numeric conversion of $C(n)$.

An analysis of the second step-size table lookup reveals that each successive entry is about 1.1 times the previous entry. As long as range limiting of the second table

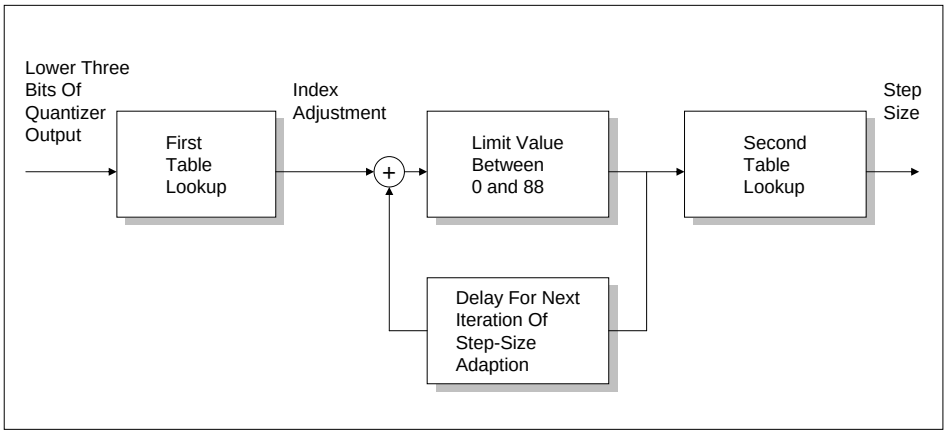


Figure 4.14: IMA ADPCM step-size adaptation

index does not take place, the value for stepsize(n) is approximately the product of the previous value, stepsize$(n-1)$, and a function of the code word, $f(C(n-1))$:

$$\text{stepsize}(n) = \text{stepsize}(n-1) \times f(C(n-1))$$

The above two equations can be manipulated to express the decoded audio sample, $x_p(n)$, as a function of the step size and the decoded sample value at time, m, and the set of code words between time, m, and n

$$x_p(n) = x_p(m) + \text{stepsize}(m) \times \sum_{i=m+1}^{n} \left\{ \prod_{j=m+1}^{i} f(C(j)) \right\} \times C'(i)$$

The terms in the summation are only a function of the code words from time $m+1$ onward. An error in the code word, $C(q)$, or a random access entry into the bit stream at time q can result in an error in the decoded output, $x_p(q)$, and the quantizer step size, stepsize$(q+1)$. The above equation shows that an error in $x_p(m)$ amounts to a constant offset to future values of $x_p(n)$. This offset is inaudible unless the decoded output exceeds its permissible range and is clipped. Clipping results in a momentary audible distortion but also serves to correct partially or fully the offset term. Furthermore, digital highpass filtering of the decoder output can remove this constant offset term. The above equation also shows that an error in stepsize$(m+1)$ amounts to an unwanted gain or attenuation of future values of the decoded output $x_p(n)$. The shape of the output waveform is unchanged unless the index to the second step-size table lookup is range limited. Range limiting results in a partial or full correction to the value of the step size.

The nature of the step-size adaptation limits the impact of an error in the step

size. Note that an error in stepsize$(m+1)$ caused by an error in a single code word can be at most a change of 1.1^9, or 7.45 dB in the value of the step size. Also any sequence of 88 code words that all have magnitude 3 or less completely corrects the step size to its minimum value. Even at the lowest audio sampling rate typically used, 8 kHz, 88 samples correspond to 11 ms of audio. Thus random access entry or edit points exist whenever 11 ms of low-level signal occur in the audio stream.

4.2.5.5 LPC

Historically, digital speech signals are sampled at a rate of 8000 samples/sec. Typically, each sample is represented by 8 bits (using μ-law or a-law respectively). This corresponds to an uncompressed rate of 64 Kbps. With current compression techniques (all of which are lossy), it is possible to reduce the rate to 8 Kbps with almost no perceptible loss in quality. Further compression is possible at a cost of lower quality. All of the current low-rate speech coders are based on the principle of linear predictive coding (LPC). The method of linear predictive analysis is one of the most powerful speech analysis techniques. It has become the predominant technique for estimating the basic speech parameters, e.g. pitch, formants, spectra, vocal tract area functions, and for representing speech for low bit rate transmission or storage. The importance of this method lies both in its ability to provide extremely accurate estimates of the speech parameters, and its relative speed of computation.

The basic idea behind linear predictive analysis is that a speech sample can be approximated as a linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients can be determined. The predictor coefficients are the weighting coefficients used in the linear combination. The philosophy of linear prediction is related to the basic speech synthesis model in which speech can be modeled as the output of a linear, time-varying system excited by either quasi-periodic pulses (during voiced speech), or random noise (during unvoiced speech) as shown in figure 4.15. The linear prediction method provides a robust, reliable, and accurate method for estimating the parameters that characterize the linear, time-varying system.

The following affects speech or happens when we speak respectively:

- speech can be produced, if air is pushed from the lung through the vocal tract and out of the mouth

- for certain voiced sound, the vocal cords vibrate (open and close) - the rate at which the vocal cords vibrate determines the pitch of your voice. Women and young children tend to have high pitch (fast vibration) while adult males tend to have low pitch (slow vibration)

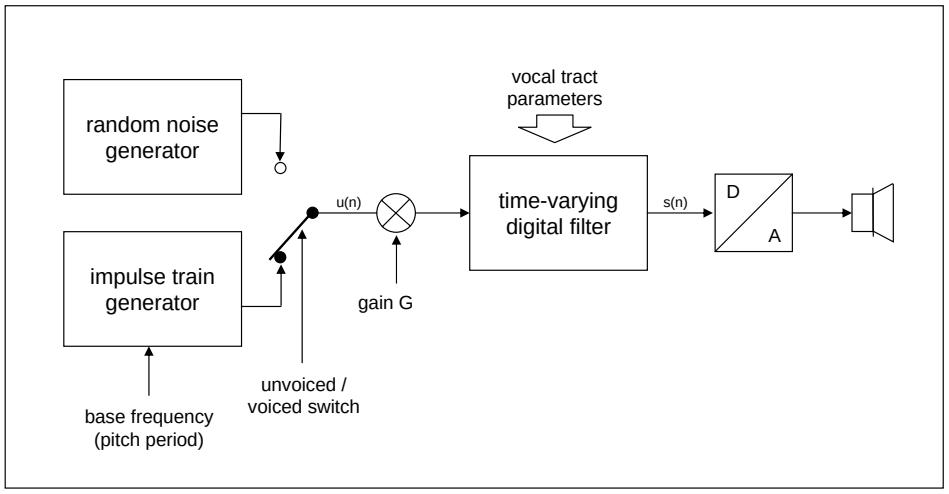


Figure 4.15: LPC model

- for certain fricatives and plosive (or unvoiced) sound, the vocal cords do not vibrate but remain constantly opened
- the shape of the vocal tract determines the sound
- while speaking, the vocal tract changes its shape producing different sound
- the shape of the vocal tract changes relatively slowly (on the scale of 10 ms to 100 ms)
- the amount of air coming from the lung determines the loudness of the voice

The composite spectrum effects of radiation, vocal tract, and glottal excitation are represented by a time-varying digital filter whose steady-state system function is of the form

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{4.2}$$

The system is excited by an impulse train for voiced speech or random noise sequence for unvoiced speech. Thus, the parameters of this model are: voiced or unvoiced classification, pitch period for voiced speech, gain parameter G, and the coefficients (a_k) of the digital filter. These parameters all vary slowly with time.

The pitch period and voice or unvoiced classification can be estimated using one of the many methods discussed in [Kon99] or by methods on linear predictive analysis.

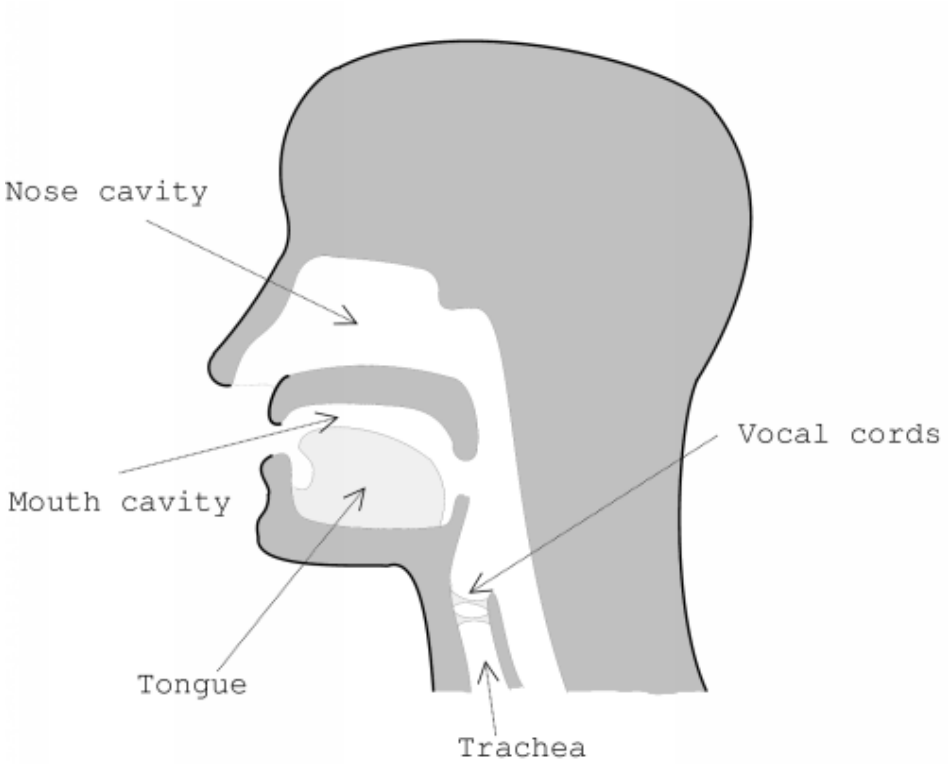


Figure 4.16: Human vocal system

For the system of figure 4.15, the speech samples $s(n)$ are related to the excitation $u(n)$ by the difference equation

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G\,u(n) \tag{4.3}$$

A linear predictor with prediction coefficients, α_k is defined as a system whose output is

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k) \tag{4.4}$$

Such systems were used to reduce the variance of the difference signal in differential quantization schemes. The system function of a p^{th} order linear predictor is the polynomial

$$P(z) = \sum_{k=1}^{p} \alpha_k z^{-k} \tag{4.5}$$

The prediction error, $e(n)$, is defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \tag{4.6}$$

From equation (4.6) it can be seen that the prediction error sequence is the output of a system whose transfer function is

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{4.7}$$

It can be seen by comparing equations (4.3)-(4.6) that if the speech signal obeys the model of equation 4.3 exactly, and if $\alpha_k = a_k$, then $e(n) = G\,u(n)$. Thus, the prediction error filter, $A(z)$, will be an inverse filter of the system, H(z), of equation (4.2), i.e.,

$$H(z) = \frac{G}{A(z)} \tag{4.8}$$

The basic problem of linear prediction analysis is to determine a set of predictor coefficients $\{\alpha_k\}$ directly from the speech signal in such a manner as to obtain a good estimate of the spectral properties of the signal through the use of (4.8). Because of the time-varying nature of the speech signal the predictor coefficients must be estimated from short segments of the speech signal. The basic approach is to find a set of predictor coefficients that will minimize the mean-squared prediction error over a short segment of the speech waveform. The resulting parameters are then assumed to be the parameters of the system function, $H(z)$, in the model for speech production.

For further details see e.g. [RS78].

4.2.5.6 CELP

The CELP[1] technique is based on three ideas:

1. the use of a linear prediction (LP) model to model the vocal tract
2. the use of (adaptive and fixed) codebook entries as input (excitation) of the LP model

[1] described by the US Federal Standard FIPS 1016

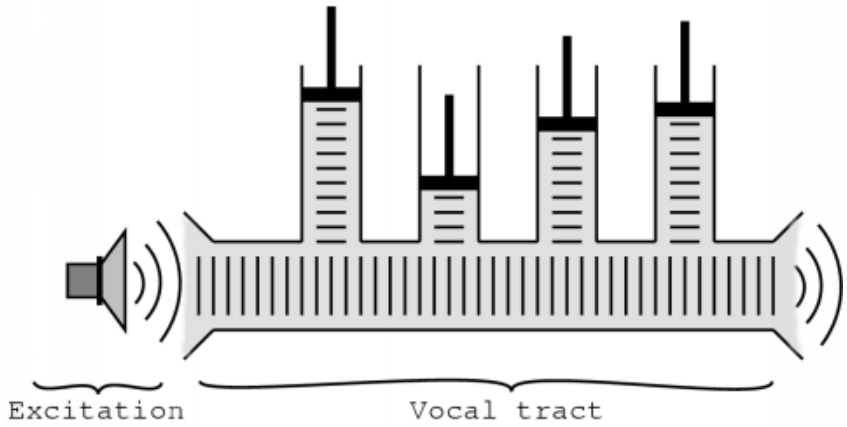


Figure 4.17: Human vocal tract

3. the search performed in closed-loop in a *perceptually weighted domain*

As described above the main problem with vocoders is the simplistic model of the excitation used. One method of circumventing this problem is the multi-pulse coder[McE03], Codebook Excited Linear Prediction (CELP) is another. Most of the more recent speech coding standards are based on CELP.

In the CELP coder the speech is passed through the cascade of the vocal tract predictor and the pitch predictor. The output of this predictor is a good approximation to Gaussian noise. This noise sequence has to be quantized and transmitted to the receiver. Multipulse coders quantize it using a series of weighted impulses. CELP coders use vector quantization. The index of the codeword that produces the best quality speech is transmitted along with a gain term for it.

The codebook search is carried out using an analysis-by-synthesis technique. The speech is synthesized for every entry in the codebook. The codeword that produces the lowest error is chosen as the excitation. The error measure used is perceptually weighted so the chosen codeword produces the speech that sounds the best.

The codebook search is very computationally intensive (originally it required 333 MFLOPS) but fast algorithms have been developed so that a CELP coder can now be implemented in real-time using modern digital signal processing microprocessors. This technique is currently one of the most effective methods of obtaining high quality speech at very low bit rates.

4.2.5.7 GSM Enhanced Full Rate

GSM-EFR is a hybrid encoder, because it uses both the model based system in encoding formants and pitch, and the waveform model for matching with the input

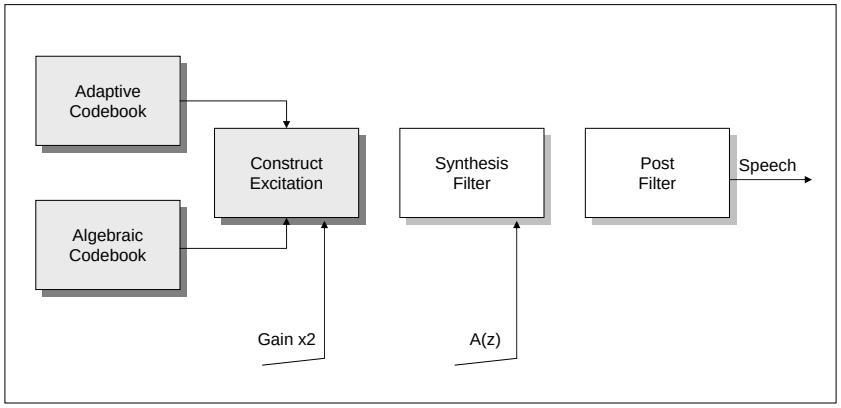


Figure 4.18: GSM-Enhanced Full Rate speech decoder model

signal [Spa94]. This is done with AbS (Analysis by Synthesis) Predictive encoding [KD88]. GSM-EFR has a speech quality that is equal to or better than 32 Kbps ADPCM according to [JV97].

GSM-EFR is standardized by the ETSI in [Ins96]. The excitation in a code excited linear prediction (CELP) codec is selected from a codebook to minimize the perceptual weighted error. The weighting is done so that the quantization noise is placed in the energy formants. This masks the quantization noise and permits the use of fewer bits in encoding.

GSM-EFR encoder uses 20 ms frames which are divided into four subframes of 5 ms each. Initially the input signal is highpass filtered with the cutoff frequency of 80 Hz and scaled. The LPC is done for two different asymmetrical weighted windows of 240 (30 ms) samples with no lookahead. Lookahead is the use of data from a future frame than the one encoded. The LP coefficients are calculated with the Levinson-Durbin algorithm [Bun85]. Each LP coefficient is then converted to a Linear Spectral Pair (LSP) for quantization and interpolation. LSP maps the filter coefficient on to the unit circle in the range of π to $-\pi$. The LP coefficients in LSP domain also remain stable in case of bit errors and therefore interpolation is possible. The coefficients are then converted back to LP filter coefficients to be used in synthesis and weighting filters.

Open-loop pitch analysis is also executed for each half of the frame to get an estimate of the pitch. Closed-loop search is then based on the estimate and performed in every subframe. A target signal and the impulse response of the weighted synthesis filter are used to find the best pitch for the adaptive codebook. The target signal is the weighted speech signal minus zero response from the weighted synthesis filter. The pitch resolution is 1/6 sample in the range 18 to 94 and one sample in the range 95 to 143. In subframe one and three nine bits are used and in subframe

two and four they are coded relative to the previous subframe using six bits. The adaptive codebook gain is also computed and non-uniformly quantized with four bits for each subframe.

The algebraic codebook is an interleaved single-pulse permutation design and encodes 40 positions with ten pulses with values -1 or $+1$. The 40 positions are divided into five different tracks with two pulses on each track. Each track uses seven bits to encode signs and position. The excitation vector is found by minimizing the MSE (mean square error) between the weighted input speech and the synthesized speech. The gain of the algebraic codebook is computed and the correction factor for a Moving Average (MA)-predictor is quantized with a five bit codebook.

Decoding is done in the following way. The adaptive codebook parameters pitch and gain is decoded. An excitation vector is then created from the excitation history using interpolation for short and fractional lags. This excitation is also scaled by the gain factor. The algebraic codebook part is recreated and scaled by the gain. These two excitations are added before filtering with LP-coefficients $A(z)$ in the synthesis filter as seen in figure 4.18. The synthetic speech is then post-filtered with an adaptive filter consisting of a formant part and a tilt compensation part.

4.2.5.8 AMR

In recent mobile communication systems, the algebraic code-excited linear predictive (ACELP) coding algorithm has been adopted for wideband speech codecs such as the Adaptive Multi-Rate Wideband (AMR-WB) standard as well as for many narrow band standard speech codecs such as G.729, the Global System for Mobile Communications (GSM) Enhanced Full Rate, Enhanced Variable Rate Coding, and the AMR codec.

Up to the present, most speech codecs used in mobile communication systems have operated on a narrow bandwidth limited to 200 to 3400 Hz. However, as AMR-WB extends the audio bandwidth from 50 to 7000 Hz, it becomes possible to achieve high quality speech signals both in intelligibility and naturalness. The AMR-WB codec is the speech codec most recently standardized by the 3GPP (3rd Generation Partnership Project) for GSM and WCDMA 3G systems [3GP01] and has been selected as the new ITU-T G.722.2 standard [ITU02]. The AMR-WB codec is a multi-rate codec with nine different bit rates between 6.6 and 23.85 Kbps. To reduce average bit rate, this codec supports the discontinuous transmission (DTX), using Voice Activity Detection (VAD) and Comfort Noise Generation (CNG) algorithms. The coder works on a frame of 320 speech samples (20 ms), and a look ahead of 5 ms is required. So the algorithmic delay for the coder is 25 ms.

Even though it operates on nine different bit rates, the bit allocation for each codec is very similar since each codec is based on the same algorithm, i.e., the ACELP algorithm [BSL+02].

Beside the WB implementation a narrowband (AMR-NB) version exists, which has eight basic bit rates between 4.75 and 12.2 Kbps that also works on the principle of ACELP for all bit rates. To reduce the average bit rate, this codec supports DTX, using VAD and CNG algorithms. The coder works on a frame of 160 speech samples (20 ms), and no look ahead is required. So the algorithmic delay for the coder is 20 ms.

4.2.5.9 iLBC

iLBC is a speech codec developed for robust voice communication over IP. It is designed for narrow band speech, with a sampling rate of 8 kHz. The iLBC codec supports two basic frame lengths, giving a bit-rate of 13.3 Kbps with an encoding frame length of 30 ms and 15.2 Kbps with an encoding frame length of 20 ms.

The essence of the codec is LPC and block-based coding of the LPC residual signal. For each 160/240 (20 ms/30 ms) sample block, the following major steps are performed: A set of LPC filters are computed, and the speech signal is filtered through them to produce the residual signal. The codec uses scalar quantization of the dominant part, in terms of energy, of the residual signal for the block. The dominant state is of length 57/58 (20 ms/30 ms) samples and forms a start state for dynamic codebooks constructed from the already coded parts of the residual signal. These dynamic codebooks are used to code the remaining parts of the residual signal. By this method, coding independence between blocks is achieved, resulting in elimination of propagation of perceptual degradations due to packet loss. The method facilitates high-quality packet loss concealment (PLC) [ADA+04].

4.2.5.10 Speex

The Speex[2] project has been started because there was a need for a speech codec that was open-source and free from software patents. These are essential conditions for being used by any open-source software. There is already Vorbis that does general audio, but it is not really suitable for speech. Also, unlike many other speech codecs, Speex is not targeted at cell phones but rather at Voice over IP (VoIP) and file-based compression. The codec is mainly designed for 3 different sampling rates: 8 kHz, 16 kHz, and 32 kHz. These are respectively referred to as narrowband, wideband and ultra-wideband.

With Speex, it is possible to vary the complexity allowed for the encoder. This is done by controlling how the search is performed with an integer ranging from 1 to 10 in a way that's similar to the -1 to -9 options to gzip[3] and bzip2[4] compression

[2] `http://www.speex.org/`
[3] `http://www.gzip.org/`
[4] `http://www.bzip.org/`

Codec	Scheme	Sampling [kHz]	Bit rate [Kbps]	Delay [ms]	MOS
G.711	PCM ({a,μ}-Law)	8	64	0.125	4.4
G.722	SB-ADPCM	8	64	0.125	4.5
G.723	MP-MLQ	8	5.3/6.3	30	3.8/3.9
G.726	ADPCM	8	24/32	0.125	-/3.85
G.728	LD-CELP	8	16	0.625	3.61
G.729	CS-ACELP	8	8	10	3.92
AMR	ACELP based	8	4.75 – 12.2	?	?
AMR-WB	ACELP based	16	6.6 – 23.85	?	?
GSM	RPE-LTP	8	4 – 21	20	3.8
iLBC	CELP based	8	13.3/15.2	?	4
iSAC	CELP based	16	10 – 32	33 – 63	>4
Speex	CELP based	8/16/32	2.15 – 24.6/ 4 – 44.2	30 – 34	>4

Table 4.1: Overview: current important speech codecs

utilities. For normal use, the noise level at complexity 1 is between 1 and 2 dB higher than at complexity 10, but the CPU requirements for complexity 10 is about 5 times higher than for complexity 1. In practice, the best trade-off is between complexity 2 and 4, though higher settings are often useful when encoding non-speech sounds like DTMF tones [Val03].

Table 4.1 gives an overview of current important speech codecs:

4.3 Measurement of Speech Quality

Voice over IP systems can suffer from significant call quality and performance management problems. Network managers and others need to understand basic call quality measurement techniques, so that they can successfully monitor, manage and diagnose these problems. IP call quality can be affected by noise, distortion, too high or low signal volume, echo, gaps in speech and a variety of other problems. When measuring call quality, there are three basic categories that are studied:

- **Listening Quality** - Refers to how users rate what they *hear* during a call.
- **Conversational Quality** - Refers to how users rate the overall quality of a call based on listening quality and their ability to converse during a call. This includes any echo or delay related difficulties that may affect the conversation.
- **Transmission Quality** - Refers to the quality of the network connection used to carry the voice signal. This is a measure of network service quality as opposed to the specific call quality.

The objective of call quality measurement is to obtain a reliable estimate of one or more of the above categories using either subjective or objective testing methods

Score	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 4.2: MOS Impairment Scale

i.e., using human test subjects or computer based measurement tools.

4.3.1 Subjective Measurements

Subjective testing is the *time honored* method of measuring voice quality, but it is a costly and time consuming process. One of the better known subjective test methodologies is the Absolute Category Rating (ACR) Test.

In an ACR Test, a pool of listeners rate a series of audio files using a five-grade impairment scale ranging from 1 to 5:

After obtaining individual scores, the average or Mean Opinion Score (MOS) for each audio file is calculated. In order to achieve a reliable result for an ACR Test, a large pool of test subjects should be used (16 or more), and the test should be conducted under controlled conditions using a quiet environment. Generally, scores become more stable as the number of listeners increases. In order to reduce the variability in scores and to help with scaling of results, tests commonly include reference files that have *industry accepted* MOS scores.

The chart below (figure 4.19) shows the raw votes from an actual ACR Test with 16 listener votes that resulted in a MOS score of 2.4. The high number of votes for opinion scores 2 and 3 are consistent with the MOS score of 2.4; however, a significant number of listeners did vote scores of 1 and 4.

When conducting a subjective test, it is important to understand that the test is truly subjective, and that the test results can vary considerably. Within the telephony industry, manufacturers often quote MOS scores associated with codecs; in reality, these scores are a value selected from a given subjective test.

Test labs typically use high quality audio recordings of phonetically balanced source text, such as the Harvard Sentences, for input to the VoIP system being tested. The Harvard Sentences are a set of English phrases chosen, so that the spoken text will contain the range of sounds typically found in speech. Recordings are obtained in quiet conditions using high-resolution (16 bit) digital recording systems and are adjusted to standardized signal levels and spectral characteristics. The International Telecommunications Union (ITU) and the Open Speech Repository are sources of phonetically balanced speech material.

In addition to ACR, other types of subjective tests include the Degradation Cat-

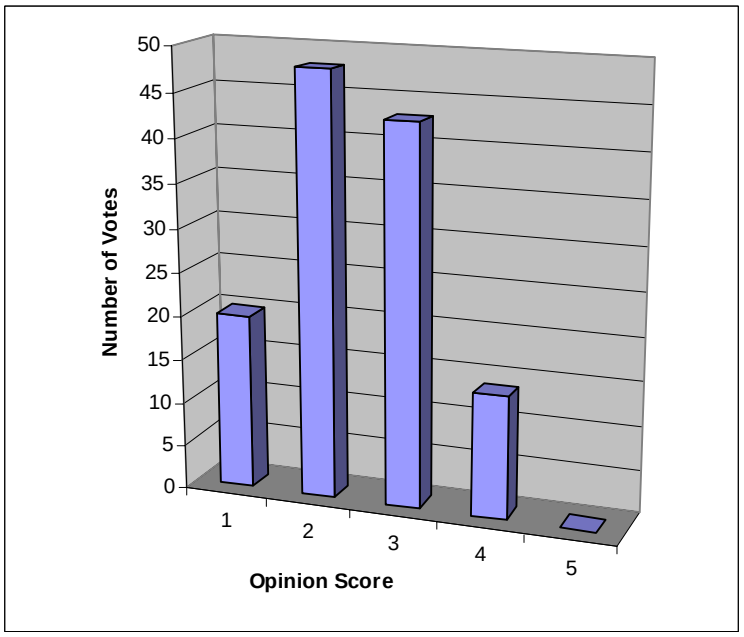


Figure 4.19: ACR Test - Opinion Score

egory Rating (DCR) and Comparison Category Rating (CCR). DCR methodology looks at the level of degradation for the impaired files and produces a D-MOS score. The Comparison Category Rating (CCR) Test compares pairs of files and produces a C-MOS score. In order to differentiate between listening and conversational scores, the International Telecommunication Union (ITU) introduced the terms MOS-Listening Quality (MOS-LQ) and MOS-Conversational Quality (MOS-CQ) with the additional suffixes (S)ubjective, (O)bjective and (E)stimated. Hence, a listening quality score from an ACR test is a MOS-LQS.

Conversational quality testing is more complex, and hence, used much less frequently. In a conversational test, a pool of listeners are typically placed into interactive communication scenarios and asked to complete a task over a telephone or VoIP system. Testers introduce effects such as delay and echo, and the test subjects are asked for their opinion on the quality of the connection. The effect of delay on conversational quality is very task dependant. For non-interactive tasks, one-way delays of several hundred milliseconds can be tolerated; for highly interactive tasks, even short delays can introduce conversational difficulty. The task dependency of delay introduces some question over the interpretation of conversational call quality metrics. For example, two identical VoIP system connections have 300 milliseconds of one-way delay; however, one supports a highly interactive business negotiation, while the other supports an informal chat between friends. In the first example,

users may say that call quality was bad: in the second case, the users probably would not even notice the delay.

4.3.2 Objective Measurements

In an effort to supplement subjective listening quality testing with lower cost objective methods, the ITU developed P.861 (PSQM) and the newer P.862 (PESQ). These measurement techniques determine the distortion introduced by a transmission system or codec by comparing an original reference file sent into the system with the impaired signal that came out. Although these techniques were developed for lab testing of codecs, they are widely used for VoIP network testing.

The P.861 and P.862 algorithms divide the reference and impaired signals into short overlapping blocks of samples, calculate Fourier Transform coefficients for each block and compare the sets of coefficients. P.862 produces a PESQ score that has a similar range to MOS; however, it is not an exact mapping. The new PESQ-LQ score is more closely aligned with listening quality MOS. These algorithms both require access to both the source file and the output file in order to measure the relative distortion.

A widely accepted mapping function (PESQ score to an average ITU-T P.800 MOS scale) is given by

$$y = \begin{cases} 1.0, x \leq 1.7 \\ -0.157268x^3 + 1.386609x^2 - 2.504699x + 2.0233454, x > 1.7 \end{cases}$$

where x is the PESQ score and y is the corresponding mapping LQ MOS.

A quite similar mapping function (figure 4.20, which is originally mentioned in [Sta03c] is given by

$$y = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.4945x + 4.6607}}$$

where it's explicitly stated, that users may approximate this curve using other means (for example a lookup table) with the following constraints:

- The mapped MOS-LQO score should be within ±0.01 absolute of the curve defined in [Sta03c]
- This constraint should be held for all points, no more than 0.01 apart, over the whole raw P.862 range −0.5 to 4.5.

The basic P.862 model provides raw scores in the range −0.5 to 4.5. The wideband extension to Recommendation P.862 includes a mapping function that allows linear comparisons with MOS values produced from subjective experiments that include wideband speech conditions with an audio bandwidth of 50-7.000 Hz. This means

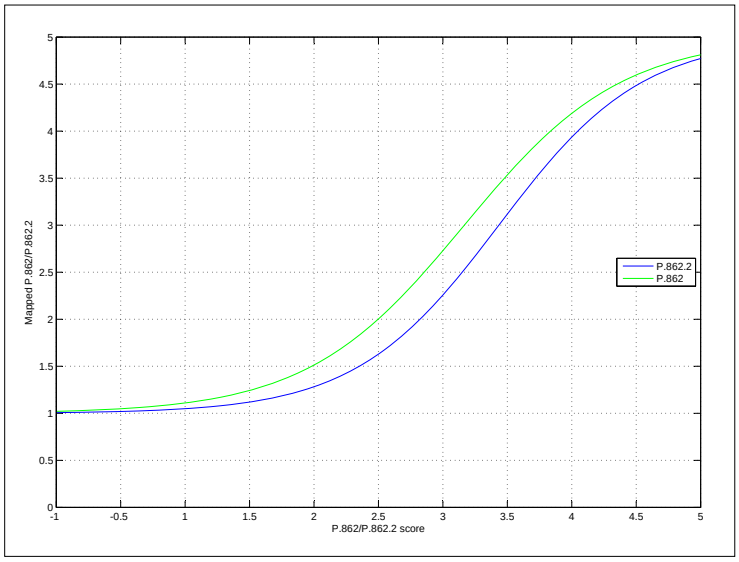


Figure 4.20: P.862 / P.862.2 algorithm's mapping function

that direct comparisons between scores produced by the wideband extension and scores produced by baseline Recommendation P.862 or Recommendation P.862.1 are not possible, due to the different experimental context. The output mapping function used in the wideband extension is defined as follows:

$$y = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.3669x + 3.8224}}$$

where x is the raw model output. The mapping function was derived from data from a number of subjective experiments; some of these experiments contained only wideband speech conditions, others contained a mixture of narrowband, wideband, and intermediate bandwidth speech [Sta05].

In 2004, the ITU standardized P.563, a single-ended objective measurement algorithm that is able to operate on the received audio stream only. The MOS scores produced by P.563 are more widely spread that those produced by P.862, and it is necessary to average the results of multiple tests in order to achieve a stable quality metric. This approach is not suited for measuring individual calls but can produce reliable results when used over many calls to measure service quality.

As this type of algorithm requires significant computation for every sample, i.e., processing for each of 8.000 samples per second for narrowband voice and 16.000 samples per second for wideband voice, the processing load (of the order of 100 MIPS per call stream) and memory requirements are quite significant. For many applications this is impractical; in which case, packet-based approaches should be

used.

The recent meeting of ITU-T's study group 12 on end-to-end transmission performance and quality of service issues ratified in November 2005 two new extensions to PESQ (ITU-T P.862), the industry standard for perceptual voice quality testing. Advanced speech codecs to be used in the context of VoIP networks allow for extended audible bandwidth up to 7 kHz when compared to narrowband 3.7 kHz telephony bandwidth. Consequently, proper assessment calls for an optional wideband mode to be switched on in PESQ instead of the IRS filter frequency response applied in standard P.862 mode. The wide band extension has been added to the PESQ standard documentation under the denotation P.862.2. A joint task group of experts further finalized a PESQ application guide featuring common issues in the context of voice quality measurements that is now available as recommendation ITU-T P.862.3.

4.3.2.1 VQmon and the E Model

VQmon is an efficient VoIP call quality – monitoring technology based on the E Model – it's able to obtain call quality scores using typically less than one thousandth of the processing power needed by the P.861/862/563 approaches. The E Model was originally developed within the European Telecommunications Standardization Institute (ETSI) as a transmission planning tool for telecommunication networks; however, it is widely used for VoIP service quality measurement.

Based on several earlier opinion models, the E Model (described in ETSI technical report ETR 250) has a lengthy history. The E Model was standardized by the ITU as Recommendation G.107 in 1998 and is being updated and revised annually. Some extensions to the E Model that enable it's use in VoIP service quality monitoring were developed by Telchemy, Inc., and have been standardized in ETSI TS 101 329-5 Annex E.

The objective of the E Model is to determine a transmission quality rating, i.e., the R factor that incorporates the *mouth to ear* characteristics of a speech path. The range of the R factor is nominally 0-120. The typical range for R factors is 50-94 for narrowband telephony and 50-110 for wideband telephony. The R Factor can be converted to estimated conversational and listening quality MOS scores (MOS-CQ and MOS-LQ).

The E Model is based on the premise that the effects of impairments are additive. The basic E Model equation is:

$$R = R_o - I_s - I_d - I_e + A.$$

R_o is a base factor determined from noise levels, loudness, etc. I_s represents signal impairments occurring simultaneously with speech, including: loudness, quantiza-

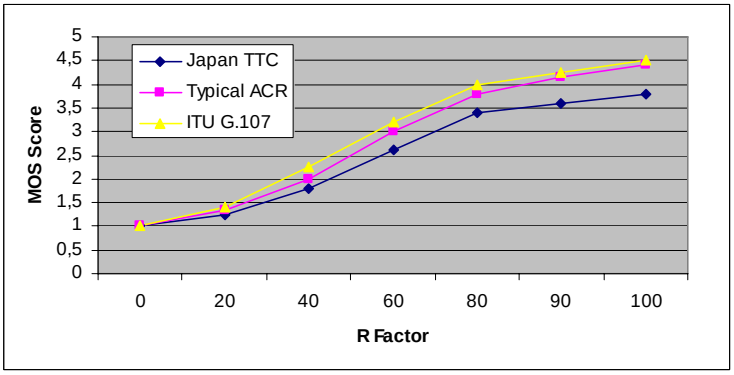


Figure 4.21: MOS Score - R Factor

tion (codec) distortion and non-optimum sidetone level. I_d represents impairments that are delayed with respect to speech, including echo and conversational difficulty due to delay, I_e is the *equipment impairment factor* and represents the effects of VoIP systems on transmission signals. A is the *advantage factor* and represents the user's expectation of quality when making a phone call. For example, a mobile phone is convenient to use; hence, people are more forgiving on quality-related problems. VQmon is an extended version of the E Model that incorporates the effects of time varying IP network impairments and provides a more accurate estimate of user opinion. VQmon also incorporates extensions to support wideband codecs.

The chart (figure 4.21) shows the relationship between the R factor generated by the E Model and MOS. The *official* mapping function provided in ITU G.107 gives a MOS score of 4.4 for an R factor of 93 (corresponding to a typical unimpaired G.711 connection, i.e., the equivalent of a regular telephone connection).

In Japan, the TTC committee developed an R factor to MOS mapping methodology that provides a closer match based on the results of subjective tests conducted in Japan. The TTC scores are traditionally lower than those in the US and Europe due in some part to cultural perceptions of quality and voice transmission. Therefore, the chart above shows three potential mappings from R to MOS.

Another complication is introduced when wideband codecs are used. An ACR test is on a fixed 1-5 scale, and is really a test that is relative to some reference conditions. In a wideband test the same scale is used, hence a wideband codec may have a MOS score of 3.9 even though it sounds much better than a narrowband codec with a MOS of 4.1. This is not the case for R factors, which have a scale that encompasses both narrowband and wideband. Therefore a wideband codec may result in an R factor of 105 whereas a typical narrowband codec may result in an R factor of 93.

The table 4.3 shows a typical representation of call quality levels. Generally, an

User Opinion	R Factor	MOS (ITU)	MOS (ACR)
Maximum Obtainable for G.711	93	4.4	4.1
Very Satisfied	90 – 100	4.3 – 5.0	4.1 – 5.0
Satisfied	80 – 90	4.0 – 4.3	3.7 – 4.1
Some Users Satisfied	70 – 80	3.6 – 4.0	3.4 – 3.7
Many Users Dissatisfied	60 – 70	3.1 – 3.6	2.9 – 3.4
Nearly All Users Dissatisfied	50 – 60	2.6 – 3.1	2.4 – 2.9
Not Recommended	0 – 50	1.0 – 2.6	1.0 – 2.4

Table 4.3: R Factor - MOS Score

R Factor of 80 or above represents a good objective however there are some key things to note:

Since R Factors are conversational metrics, the statement that R Factors should be 80 or more implies both a good listening quality and low delay. Stating that (ITU scaled) MOS should be 4.0 or better is not the same as assuming that this is MOS-LQ and does not incorporate delay. Saying that R should be 80 or more and MOS should be 4.0 or more is not consistent. Telchemy introduced the notation R-LQ and R-CQ to deal with this; hence, an R-LQ of 80 would be comparable with a MOS of 4.0. The typically manufacturer-quoted MOS for G.729A is 3.9 implying that G.729A could not meet the ITU scaled MOS for *Satisfied*. However G.729A is widely used and appears to be quite acceptable - this problem is due to the scaling of MOS and not the codec. Typical ACR scores for codecs should be compared to an ACR scaled range. For example, *Satisfied*, would range 3.7 to 4.1 and hence the G.729A MOS of 3.9 would be within the *Satisfied* range.

When specifying call quality objectives, it is important to be clear about terminology – either specify R Factor (R-CQ) or MOS-CQ or the combination of MOS-LQ and delay. If you using wideband and narrowband codecs then be aware that you need to interpret MOS scores as *narrowband MOS* or *wideband MOS* in order to avoid confusion.

4.4 Human Auditory System

In many applications of acoustics and audio signal processing it is necessary to know what humans actually hear. Sound, which consists of air pressure waves, can be accurately measured with sophisticated equipment. However, understanding how these waves are received and mapped into thoughts in the brain is not trivial. Sound is a continuous analog signal which (assuming infinitely small air molecules) can theoretically contain an infinite amount of information (there being an infinite number of frequencies, each containing both magnitude and phase information.)

Recognizing features important to perception enables scientists and engineers to concentrate on audible features and ignore less important features of the involved

system. It is important to note that the question of what humans hear is not only a physiological question of features of the ear but very much also a psychological issue.

Auditory perception is based on critical band analysis in the inner ear where a frequency to place transformation occurs along the basilar membrane [Nol93]. The power spectra are not represented on a linear frequency scale but on a limited frequency bands called critical bands. The auditory system can be described as a bandpass filterbank, consisting of strongly overlapping bandpass filters with bandwidths in the order of 100 Hz for signals below 500 Hz and up to 5000 Hz for signals at high frequencies. Up to 24.000 Hz 26 critical bands have to be taken in to account.

The human ear can usually hear sounds in the range 20 Hz to 20.5 kHz. With age, the range decreases, especially at the upper limit. Lower frequencies cannot be heard but loud sounds can be felt on the skin; see figure 4.22 and 4.23.

Frequency resolution of the ear is, in the middle range, about 2 Hz. That is, changes in pitch larger than 2 Hz can be perceived. However, even smaller pitch differences can be perceived through other means. For example, the interference of two pitches can often be heard as a (low-)frequency difference pitch. This effect of phase variance upon the resultant sound is known as *beating*.

However, the effect of frequency on the human ear has a logarithmic basis. In other words, the perceived pitch of a sound is related to the frequency as an exponential function. The 12-tone musical scale is an example of this; it evolved due to the way tones are perceived. When the fundamental frequency of a note (or tone) is multiplied by $2^{(1/12)}$, the result is the frequency of the next higher semitone. Going 12 notes higher (an octave) is the same as multiplying the frequency by $2^{(12/12)}$, which is the same as doubling the frequency.

The impact of this is that the raw frequency resolution of the ear is best judged in terms of semitones, or in *cents* which is 1/100 of a semitone.

The *intensity* range of audible sounds is enormous. Our ear drums are sensitive only to the sound pressure variation. The lower limit of audibility is defined to 0 dB, but the upper limit is not as clearly defined. The upper limit is more a question of the limit where the ear will be physically harmed. This limit depends also on the time exposed to the sound. The ear can be exposed to short periods in excess of 120 dB without permanent harm, but long term exposure to sound levels over 80 dB can cause permanent hearing loss.

A more rigorous exploration of the lower limits of audibility determines that the minimum threshold at which a sound can be heard is frequency dependent. By measuring this minimum intensity for testing tones of various frequencies, a frequency dependent Absolute Threshold of Hearing (ATH) curve may be derived. Typically, the ear shows a peak of sensitivity (i.e., its lowest ATH) between 1 kHz and 5 kHz, though the threshold changes with age, with older ears showing decreased

sensitivity above 2 kHz.

The ATH is the lowest of the equal-loudness contours. Equal-loudness contours indicate the sound pressure level (dB), over the range of audible frequencies, which are perceived as being of equal loudness. Equal-loudness contours were first measured by Fletcher and Munson at Bell Labs in 1933 using pure tones reproduced via headphones, and the data they collected are called Fletcher-Munson curves. Because subjective loudness was difficult to measure, the Fletcher-Munson curves were averaged over many subjects.

Robinson and Dadson refined the process in 1956 to obtain a new set of equal-loudness curves for a frontal sound source measured in an anechoic chamber. The Robinson-Dadson curves were standardized as ISO 226 in 1986. In 2003, ISO 226 was revised using data collected from 12 international studies.

Human hearing is basically like a spectral analyzer, that is, the ear resolves the spectral content of the pressure wave without respect to the phase of the signal. In practice, though, some phase information can be perceived. Inter-aural (i.e. between ears) phase difference is a notable exception by providing a significant part of the directional sensation of sound. The filtering effects of head-related transfer functions provide another important directional cue.

In some situations an otherwise clearly audible sound can be masked by another sound. For example, conversation at a bus stop can be completely impossible if a loud bus is driving past. This phenomenon is called masking. A weaker sound is masked if it is made inaudible in the presence of a louder sound.

If two sounds occur simultaneously and one is masked by the other, this is referred to as simultaneous masking. A sound close in frequency to the louder sound is more easily masked than if it is far apart in frequency. For this reason, simultaneous masking is also sometimes called frequency masking. The tonality of a sound partially determines its ability to mask other sounds. A sinusoidal masker, for example, requires a higher intensity to mask a noise-like maskee than a loud noise-like masker does to mask a sinusoid. Computer models which calculate the masking caused by sounds must therefore classify their individual spectral peaks according to their tonality.

Similarly, a weak sound emitted soon after the end of a louder sound is masked by the louder sound. In fact, even a weak sound just before a louder sound can be masked by the louder sound. These two effects are called forward and backward temporal masking, respectively.

The psychoacoustic model provides for high quality lossy signal compression by describing which parts of a given digital audio signal can be removed (or aggressively compressed) safely – that is, without significant losses in the quality of the sound. It explains, for example, how a sharp clap of the hands might seem painfully loud in a quiet library, but hardly noticeable after a car backfires on a busy, urban street. It might seem as if this would provide little benefit to the overall compression ratio,

but psychoacoustic analysis routinely leads to compressed music files that are 10 to 12 times smaller than high quality original masters with very little discernible loss in quality. Such compression is a feature of nearly all modern audio compression formats. Some of these formats include MP3, Ogg Vorbis, Musicam (used for digital audio broadcasting in several countries), and the compression used in MiniDisc.

Psychoacoustics is based heavily on human anatomy, especially the ear's limitations in perceiving sound as outlined previously.
To summarize, these limitations are: (also see 4.4.1)

- High frequency limit
- Absolute threshold of hearing
- Absolute threshold of pain
- Temporal masking
- Simultaneous masking

Given that the ear will not be at peak perceptive capacity when dealing with these limitations, a compression algorithm can assign a lower priority to sounds outside the range of human hearing. By carefully shifting bits away from the unimportant components and toward the important ones, the algorithm ensures that the sounds a listener can hear most clearly are of the highest quality.

4.4.1 Psychoacoustic Effects and Enhancements

High Frequency Limit The high frequency limit of hearing is the upper extent to which a particular animal can perceive sound. Perhaps the most commonly known aspect of the psychoacoustic model is that humans cannot hear frequencies above and below certain thresholds; in fact, most humans can only hear frequencies between 20 Hz and 20.5 kHz. So-called *silent* dog whistles exploit this phenomenon by producing sounds at frequencies higher than those audible to humans but well within the range of a dog's hearing. Likewise, when compressing a digital signal, an acoustic engineer can safely assume that any frequency beyond approximately 20.5 kHz will not have any effect on the perceived sound of the finished product, and thus use a bandpass filter to cut everything outside this range. The sound can then be sampled at the standard CD sample rate of 44.1 kHz, set somewhat higher than the calculated Nyquist-Shannon rate of 41 kHz to allow for the cut-off slope of a reasonable bandpass filter.

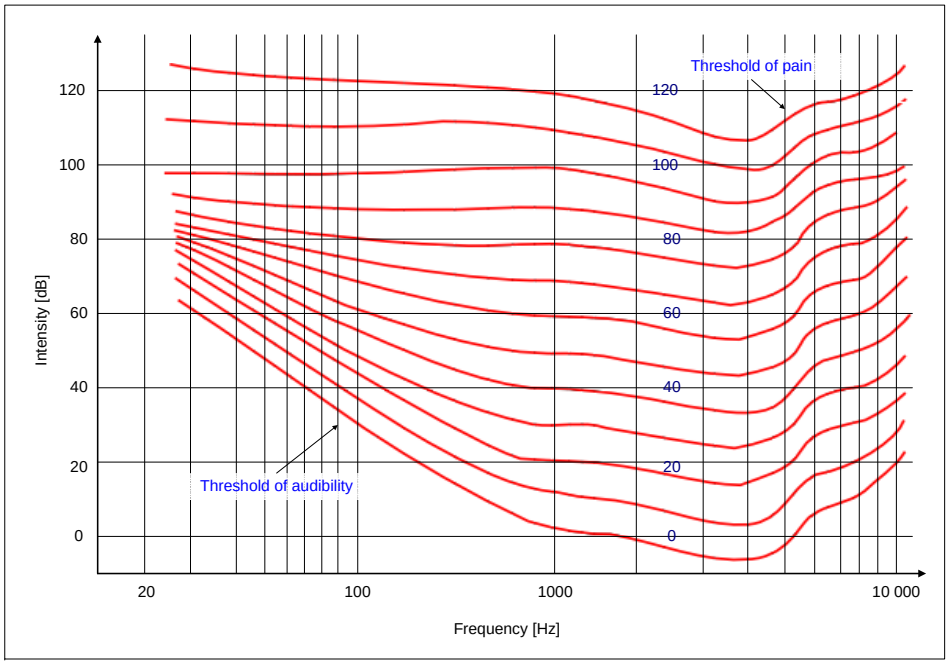


Figure 4.22: The Fletcher-Munson equal-loudness contours
(The lowest of the curves is the ATH)

Absolute Threshold of Hearing The threshold of hearing is the sound pressure level SPL of 20 μPa (micropascals) = $2 \cdot 10^{-5}$ pascal (Pa). This low threshold of amplitude (strength or sound pressure level) is frequency dependent. Also see figure 4.23. The absolute threshold of hearing (ATH) is the minimum amplitude (level or strength) of a pure tone that the average ear with normal hearing can hear in a noiseless environment. The threshold of pain is the SPL beyond which sound becomes unbearable for a human listener. This threshold varies only slightly with frequency. Prolonged exposure to sound pressure levels in excess of the threshold of pain can cause physical damage, potentially leading to hearing impairment.

The Threshold of hearing is frequency dependent, and typically shows a minimum (indicating the ear's maximum sensitivity) at frequencies between 1 kHz and 5 kHz. A typical ATH curve is pictured in figure 4.22. The absolute threshold of hearing represents the lowest curve amongst the set of equal-loudness contours, with the highest curve representing the threshold of pain.

In psychoacoustic audio compression, the ATH is used, often in combination with masking curves, to calculate which spectral components are inaudible and may thus be ignored in the coding process; any part of an audio spectrum which has an amplitude (level or strength) below the ATH may be removed from an audio signal

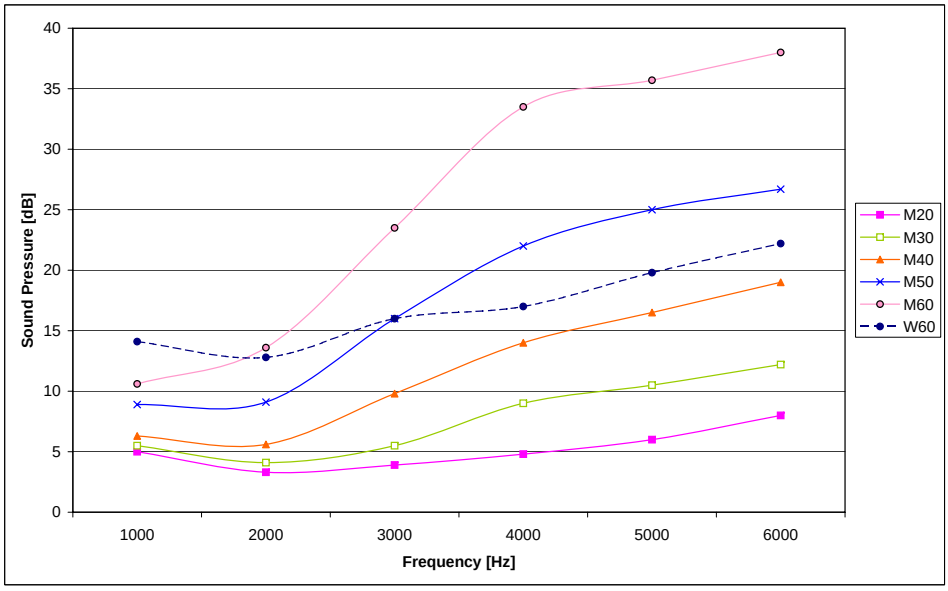


Figure 4.23: Thresholds of hearing for male (M) and female (W) humans

without any audible change to the signal.

The ATH curve rises with age as the human ear becomes more insensitive to sound, with the greatest changes occurring at frequencies higher than 2 kHz. Curves for subjects of various age groups are illustrated in figure 4.23. The data is from the United States Occupational Health and Environment Control, Standard Number:1910.95 App F.

Temporal Masking The effect of temporal masking occurs when a sudden stimulus sound makes inaudible other sounds which are present immediately preceding or following the stimulus. Masking that obscures a sound immediately preceding the masker is called backwards masking or pre-masking and masking that obscures a sound immediately following the masker is called forwards masking or post-masking. Temporal masking's effectiveness attenuates exponentially from the onset and offset of the masker, with the onset attenuation lasting approximately 10 ms and the offset attenuation lasting approximately 50 ms.

Similar to simultaneous masking, temporal masking reveals the frequency analysis performed by the auditory system; forwards masking thresholds for complex harmonic tones (e.g., a sawtooth probe with a fundamental frequency of 500 Hz) exhibit threshold peaks (i.e., high masking levels) for frequency bands centered on the first several harmonics. In fact, auditory bandwidths measured from forwarding masking thresholds are narrower and more accurate than those measured using

simultaneous masking.

Temporal masking should not be confused with the ear's acoustic reflex, an involuntary response in the middle ear that is activated to protect the ear's delicate structures from loud sounds. One example of temporal masking is the illusory continuity of tones, an auditory illusion caused when a tone is interrupted for a short time (approximately 50 ms or less), during which a narrow band of noise is played. Whether the tone is of constant, rising or decreasing pitch, the ear perceives the tone as continuous if the 50 ms (or less) discontinuity is masked by noise. Because the human ear is very sensitive to sudden changes, however, it is necessary for the success of the illusion that the amplitude of the tone in the region of the discontinuity not decrease or increase too abruptly [WWP88].

Simultaneous Masking Simultaneous masking might exist between two concurrent sounds. Sometimes it is also called frequency masking since it is often observed when the sounds share a frequency band e.g. two sine tones at 440 Hz and 450 Hz can be perceived clearly when played separated from each other. But if they were played simultaneously it is nearly impossible to perceive clearly that there is a difference - hence that more than one sine tone is played audible.

4.5 General Compression Techniques

Compression techniques fit into different categories. Entropy, source and hybrid encoding can be distinguished. Entropy encoding is a lossless process, while source encoding is a lossy process. Most multimedia systems use hybrid techniques, which are a combination of the two.

Entropy coding is used regardless of the media's specific characteristics. The data stream to be compressed is considered to be a simple digital sequence and the semantics of the data is ignored. Entropy encoding is an example of lossless encoding as the decompression process regenerates the data completely. Run-length coding is an example of entropy encoding that is used for data compression in file systems.

Source coding takes into account the semantics of the data. The degree of compression that can be reached by source encoding depends on the data contents. In the case of lossy compression techniques, a one-way relation between the original data stream and the encoded data stream exists; the data streams are similar but not identical. Different source encoding techniques make extensive use of the characteristics of the specific medium. An example is the sound source coding, where sound is transformed form time-dependent to frequency-dependent sound concatenations, followed by the encoding of the formants. This transformation substantially reduces the amount of data. Formants are defined as being the maxima of the voice spectrum. In most cases, three to five formants are sufficient to reconstruct the

original signal in the time domain. The major problem is the correct reproduction of the transitions between individual voice units in the time domain.

Hybrid compression techniques are a combination of well-known algorithms and transformation techniques that can be applied to multimedia systems. The simplest compression techniques are based on interpolation and subsampling. Here, it is possible to make use of specific physiological characteristics of the human eye or ear. The human eye is more sensitive to changes in brightness than to color changes. Therefore, it is reasonable to divide the image into YUV components instead of RGB components.

Sampled images, audio or video data streams often contain sequences of the same bytes. By replacing these repeated byte sequences with the number of occurrences, a substantial reduction of data can be achieved. This is called run-length coding, which is indicated by a special flag that does not occur as part of the 255 different bytes in the compressed data stream. To illustrate such a byte-stuffing, we define the exclamation mark "!" as a special flag. A single occurrence of this exclamation flag is interpreted as a special flag during decompression. Two consecutive exclamation flags are interpreted as an exclamation mark occurring within the data. The overall run-length coding procedure can be described as follows: if a byte occurs at least four consecutive times, the number of occurrences is counted. The compressed data contains this byte followed by the special flag and the number of its occurrences. This allows the compression of between 4 and 259 bytes into three bytes only. Remembering that we are compressing at least 4 consecutive bytes, and the number of occurrences can start with an offset of -4. Depending on the algorithm, one or more bytes can be used to indicate the length. In the following example, the character "C" occurs 8 consecutive times and is compressed to 3 characters "C!8".

Run-length encoding is a generalization of zero suppression, which assumes that just one symbol appears particularly often in sequences. The blank (null character - space) in text is such a symbol; single blanks or pairs of blanks are ignored. Starting with a sequence of three blanks, they are replaced by an M-byte (M-byte has the same function as the exclamation mark before) and a byte that specifies the number of blanks of this sequence. Sequences of three, to a maximum of 258 bytes, can be reduced to two bytes. The number of occurrences can be indicated with an offset of -3 (because three blanks are being suppressed). Further variations are tabulators used to substitute a specific number of zeros (or blanks). The substitution depends on the relative position within a line and the definition of different M-bytes to specify a different number of zeros bytes (or blanks). The flag M4-byte could replace 8 zero bytes, and another M5-byte could substitute a sequence of 16 zero bytes. An M5-byte followed by an M4-byte would represent 24 zero bytes.

In the case of vector quantization a data stream is divided into blocks of n bytes each ($n > 1$). A predefined table contains a set of patterns. For each block, a table entry with the most similar pattern is identified. Each pattern in the table is

associated with an index. Such a table can be multi-dimensional; in this case, the index will be a vector. A decoder uses the same table to generate an approximation of the original data stream. For further details and refinements see, e.g., [Gra84].

A technique that can be used for text compression substitutes single bytes for patterns that occur frequently. This pattern substitution replaces, for instance, the terminal symbols of high-level languages ("Else", "Procedure", "Implements"). Using an escape-byte, a larger number of patterns can be considered; this escape-byte indicates that an encoded pattern will follow. The next byte is an index used to reference to one of 256 words. The same technique can be applied to still images, video and audio. In these media, it is not easy to identify small sets of frequently used occurring patterns. It is often better to perform an approximation that looks for the most similar pattern instead of searching for the same pattern, in either case leading to the above described vector quantization.

Diatomic encoding is a variation of run-length encoding based on a combination of two data bytes. This technique determines the most frequently occurring pairs of bytes. According to an analysis of the English language, the most frequently occurring pairs are the following (note, there are blanks included in the pairs "e ", "t ", "a " and "s "): "E ", "T ", "TH", " A", "S ", "RE" and "HE". Replacement of these pairs by special single bytes that do not occur anywhere else in the text leads to a data reduction of more than 10%.

Different characters do not have to be coded with a fixed number of bits. The Morse Alphabet is based on this idea. Frequently-occurring characters are coded with shorter strings and seldom-occurring characters are coded with longer strings. Such statistical encoding depends on the occurrence frequency of single characters or sequences of data bytes. There are different techniques that are based on these statistical methods, the most prominent of which are Huffman and Arithmetic encoding.

Given the characters that must be encoded, together with the probability of their occurrences, the Huffman coding algorithm determines the optimal code using the minimum number of bits [Huf52]. Hence, the length (number of bits) of the coded characters will differ. In text, the shortest code is assigned to those characters that occur most frequently. To determine a Huffmann code, it is useful to construct a binary tree. The leaves (node) of this tree represent the characters that are to be encoded. Every node contains the occurrence probability of one of the characters belonging to this subtree. 0 and 1 are assigned to the branches (edges) of the tree. If the information of an image or audio/video stream can be transformed into a bit stream, such table can be used to compress the data without any loss. The same Huffman table must be available for both encoding and decoding.

If we consider run-length coding and all other methods described so far, which produced the same consecutive symbols (bytes) quite often, it is certainly a major objective to transform images, audio and video into a bit stream.

From the information theory point of view, arithmetic encoding, like Huffman encoding, is optimal [Lan84, PMLA88]. Therefore, the length of the encoded data stream is also minimal. Unlike Huffman coding, arithmetic coding does not encode each symbol separately; each symbol is instead coded by considering the prior data. Therefore, an encoded data stream must always be read from the beginning. Consequently, random access is not possible. In practice, the average compression rates achieved by arithmetic and Huffman coding are similar [Sto88].

4.5.1 Recursive Zero Runlength Encoding

With transformation based compression techniques there are often runs of zeros and it is tempting to try to improve compression by using some form of run length coding. Although run length coding is a well established technique, experience here is that it can be very difficult to do it efficiently. There are two basic approaches to handling runs of frequent symbols, with the simplification here that we need to consider only runs of zeros.

- Length-count encoding
- Wheeler's length-count encoding

In his latest report, Wheeler [Fen96] uses a novel representation for the length. Runs are always of the value 0, and the values 0 and 1 are used to encode the run length, but using digit weights of 1 and 2 instead of the more usual 0 and 1. A sequence of bits $x_0x_1x_2$ then represents the value

$$\sum_{i=0}^{n-1}(1+x_i)2^i = \sum_{i=0}^{n-1}2^i + \sum_{i=0}^{n-1}x_i2^i = (2^n-1) + \sum_{i=0}^{n-1}x_i2^i$$

For most values the most significant bit is implied and need not be encoded; the value is represented in one fewer bits than might be expected. A value of 0 cannot be represented, but that does not matter here. The coding may be generated in two ways (least significant bit first) –

- Increment the value by one and encode that modified value as an ordinary binary number, but ignoring the most significant 1 bit.
- Encode much as usual for a binary number, emitting the low-order bit and then shifting right to eliminate that bit. However, before emitting each bit, decrement the current value by 1.

Ranks greater than 0 are incremented by 1 to give an $(N+1)$ symbol alphabet. There is no special code to introduce a run and the run is terminated by any *non-run* code. When combined with the shorter coding for the length itself, the Wheeler length code is an especially efficient method.

Wheeler's method is nowadays also known as RLE-0 coding and used in the best compressors known. To illustrate the coding the following example is considered. The input sequence is the vector $[4, 3, 0, 1, 8, 0, 0, 0, 0, 0, 0, 0, 9, 7, 6]$ - compressed with RLE-0 the output is $[5, 4, 0, 2, 9, 0, 0, 0, 10, 8, 7]$. But notice the encoded zeros string after 8 in the input sequence as $[0, 0, 0]$ - further improvement could be achieved by a recursive RLE-0 approach proposed first in this thesis.

The recursive-RLE-0 (rRLE-0) version is quite simple. The solution is just to repeat the RLE-0 step that often until the compressed sequence could not be compressed any further, hence the sequence length in step $i+1$ is equal to the sequence length in step i. In order to successfully decompress the compressed sequence a recursive-inverse RLE-0 decoding scheme has to be applied. To work, the recursion depth must be known, hence it must be encoded into the output stream as well. The efficiency of the recursive approach is demonstrated in the following example. Considered an input sequence of 63 zeros in a row. the non-recursive version yields to the output sequence $[0, 0, 0, 0, 0, 0]$ while the recursive version yields to $[2, 2]$ with depth 2. Further simulations showed that the recursion depth for input vectors with length of 320 (number of samples for 20 ms sound at 16 kHz sample rate) is around 3, hence the recursion depth could be efficiently encoded with only 3 bits already considered enough headroom for *special* cases. In figure 4.24 the efficiency gain is depicted for 500 random vectors with zeros spread out over the whole vector respectively.

4.5.2 Burrows-Wheeler-Transformation and Move-to-Front Coding

Michael Burrows and David Wheeler released a research report in 1994 [BW94] discussing work they had been doing at the Digital Systems Research Center in Palo Alto, California. Their paper presented a data compression algorithm based on a previously unpublished transformation discovered by Wheeler in 1983. While the paper discusses a complete set of algorithms for compression and decompression, the real heart of the paper consists of the disclosure of the BWT algorithm.

The BWT is an algorithm that takes a block of data and rearranges it using a sorting algorithm. The resulting output block contains exactly the same data elements that it started with, differing only in their ordering. The transformation is reversible, meaning the original ordering of the data elements can be restored with no loss of fidelity.

The characteristics of the transformation process make the output from the sort ideal for certain kinds of further manipulation. The extreme local fluctuations in the

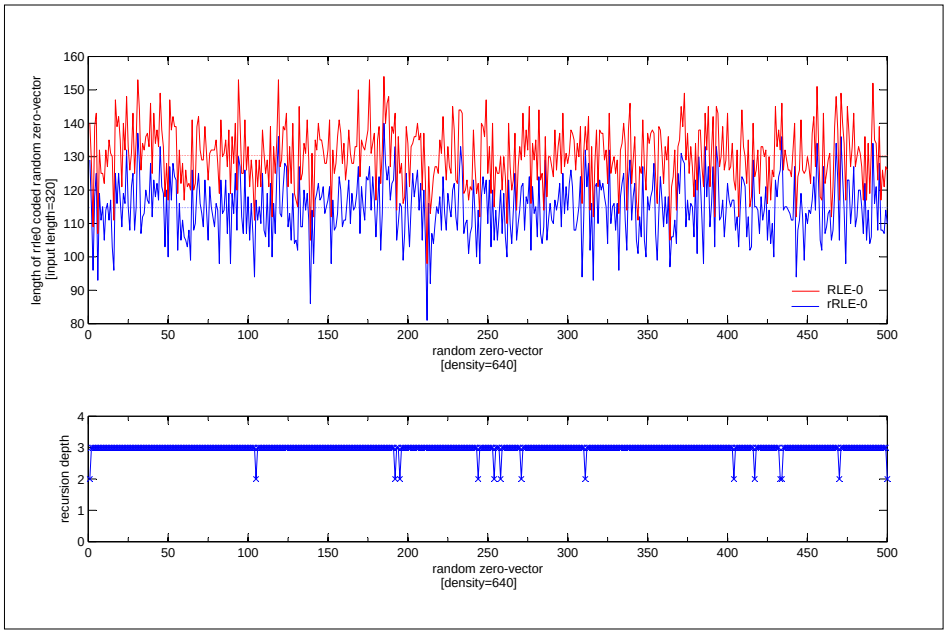


Figure 4.24: Compression efficiency with recursive RLE-0

first order statistics of the output string lead one to use a transformation that boosts and flattens the local fluttering of the statistics. The best example (and, of course, the one given in the original paper) is move-to-front coding. This coder codes a symbol as the number of distinct symbols seen since the symbol's last occurrence. Basically this means that the coder outputs the index of an input symbol in a dynamic LIFO stack and then updates the stack by moving the symbol to the top. This is easy and efficient to implement and results in fast local adaptation. As just a few common symbols will (locally) govern the input to the coder, these symbols will be kept on the top of the stack and thus the output will mainly consist of low numbers. This makes it highly susceptible to first order statistical compression methods which are, in case, easy and efficient to implement.

4.5.3 Adaptive Huffman/Arithmetic Coding

There are a few shortcomings to the straight Huffman compression. First of all, the Huffman tree has to be coded as well, hence producing some overhead which should better be avoided. Also, Huffman compression looks at the statistics of the whole file, so that if a part of the code uses a character more heavily, it will not adjust during that section. Not to mention the fact that sometimes the whole file is not available to get the counts from (such as in live information - VoIP).

The basic concept behind an adaptive compression algorithm is quite simple:

```
1: Initialize the model
2: for all character do
3:    Encode character
4:    Update the model
5: end for
```

Decompression works the same way. As long as both sides have the same initialize and update model algorithms, they will have the same information. The problem is how to update the model. To make Huffman compression adaptive, we could just re-make the Huffman tree every time a character is sent, but that would cause an extremely slow algorithm. The trick is to only update the part of the tree that is affected.

The Huffman tree is initialized with a single node, known as the Not Yet Transmitted (NYT) or escape code. This code will be sent every time that a new character, which is not in the tree, is encountered, followed by the ASCII encoding of the character. This allows for the decompressor to distinguish between a code and a new character. Also, the procedure creates a new node for the character and a new NYT from the old NYT node.

Whenever a character that is already in the tree is encountered, the code is sent and the weight is increased.

In order to for this algorithm to work, we need to add some additional information to the Huffman tree. In addition to each node having a weight, it will now also be assigned a unique node number. Also, all the nodes that have the same weight are said to be in the same block.

These node numbers will be assigned in such a way that:

1. A node with a higher weight will have a higher node number
2. A parent node will always have a higher node number than its children

This is known as the sibling property, and the update algorithm simply swaps nodes to make sure that this property is upheld. Obviously, the root node will have the highest node number because it has the highest weight. Figure 4.25 depicts a possible update procedure.

After a count is increased, the update procedure moves up the tree and inspects the ancestors of the node one at a time. It checks to make sure that the node have the highest node in its block, and if not, swaps it with the highest node number. It then increases the node weight and goes to the parent. It continues until it reaches the root node. This assures that the nodes with the highest weight are closer to the top and have shorter codes.

Arithmetic Coding is now one of the most popular methods of source coding. The basic idea of arithmetic coding was formulated by Elias in the early 1960s [Jel68].

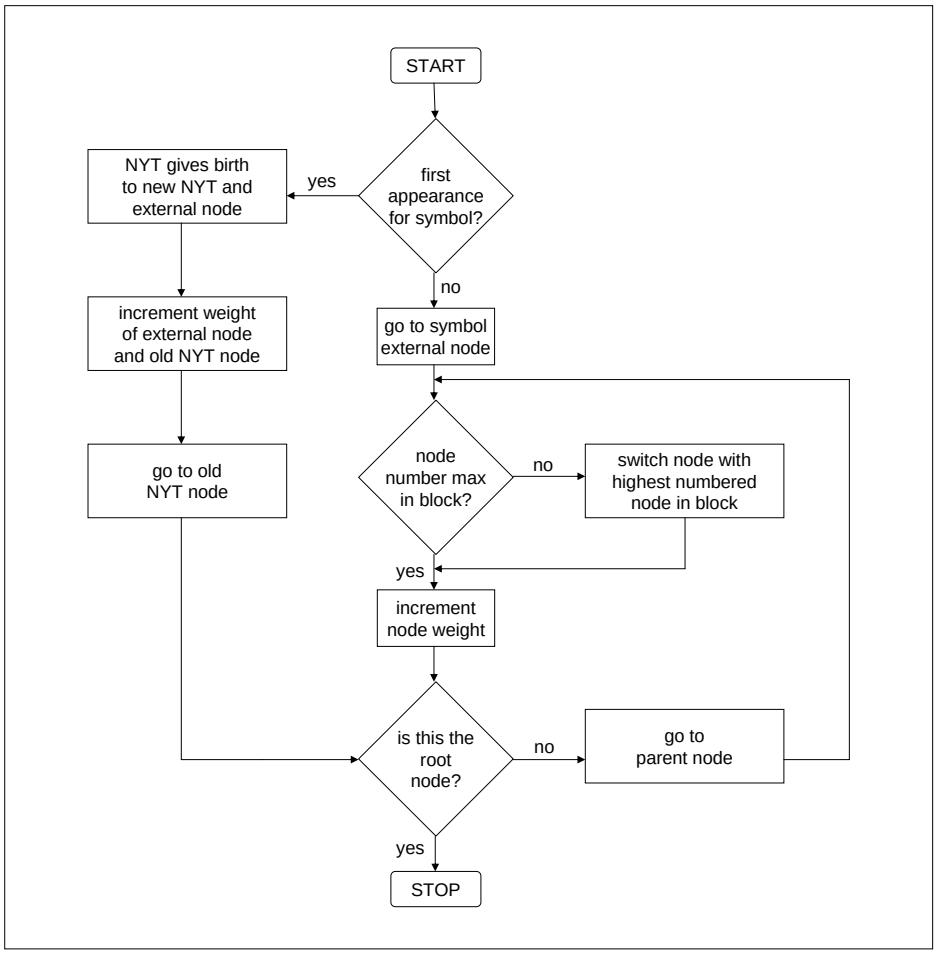


Figure 4.25: Adaptive Huffman Coding - update procedure

The first step toward practical implementation of this idea was made by Rissanen [Ris76] and Pasco [Pas76]. The advantage of arithmetic coding over other coding techniques is that it allows to attain arbitrarily small coding redundancy per one source symbol at less computational effort than any other method. Redundancy is defined as the difference between the mean codeword length and the source entropy. Furthermore, arithmetic coding may easily be applied to sources with unknown statistics, when being combined with adaptive models in an adaptive coding scheme - see e.g. [CW84, Nel96].

The encoding algorithm has two parts: the probability model and the arithmetic encoder. The model reads the next symbol from the input stream and invokes the encoder, sending it the symbol and the two required cumulative frequencies.

The model then increments the count of the symbol and updates the cumulative frequencies. The point is that the symbol's probability is determined by the model from its old count, and the count is incremented only after the symbol has been encoded. This makes it possible for the decoder to mirror the encoder's operations. The encoder knows what the symbol is even before it is encoded, but the decoder has to decode the symbol in order to find out what it is. The decoder can therefore use only the old counts when decoding a symbol. Once the symbol has been decoded, the decoder increments its count and updates the cumulative frequencies in exactly the same way as the encoder.

Adaptive data compression has a slight disadvantage in that it starts compressing with less than optimal statistics. By subtracting the cost of transmitting the statistics with the compressed data, however, an adaptive algorithm will usually perform better than a fixed statistical model.

4.6 Summary

Effective wireless voice communication systems (WVCS) depend on various technologies. As seen, the effective interplay between single components have to be managed and adjusted in order to fulfill specific requirements needed in order to provide real-time services like VoWLAN at a business-serviceable level. A first adjustment needed, is the capability to adapt the needed bandwidth with the actually given channel-capacities. In the following chapter such a method will be presented.

5 Channel Adaptivity

5.1 Introduction

Wireless channels condition and wireless communication in general is known to be error prone and permanently changing. In order to provide high quality communication regarding continuous transmissions, channel adaption mechanisms must be integrated. In this thesis a two-stage channel adaption scheme was developed. In both stages different channel feedback information were evaluated in order to provide control information for the fine-grain scalability unit of the speech codec.

5.2 Channel Feedback

Channel feedback in this thesis is summarized by two information. The first information is the reception quality feedback provided by the transmission control protocol (RTCP). These information will be presented as report packets giving status about the transmission itself and about possibly occurred errors while transmitting as for example packet loss, delay and jitter. The second information is signal strength, which is evaluated for channel adaption purposes and is provided by the wireless NIC. With knowledge of both information a channel adaption scheme is modeled which provides input for the fine-grain bit rate scalable speech codec which will be discussed later.

5.2.1 Evaluation of RTCP-Receiver and Sender Reports

RTP receivers provide reception quality feedback using RTCP report packets which may take one of two forms depending upon whether or not the receiver is also a sender. The only difference between the sender report (SR) and receiver report (RR) forms, besides the packet type code, is that the sender report includes a 20-byte sender information section for use by active senders. The SR is issued if a site has sent any data packets during the interval since issuing the last report or the previous one, otherwise the RR is issued.

Both the SR and RR forms include zero or more reception report blocks, one for each of the synchronization sources from which this receiver has received RTP data packets since the last report. Reports are not issued for contributing sources listed in the CSRC list. Each reception report block provides statistics about the data

received from the particular source indicated in that block. Since a maximum of 31 reception report blocks will fit in an SR or RR packet, additional RR packets may be stacked after the initial SR or RR packet as needed to contain the reception reports for all sources heard during the interval since the last report.

It is expected that reception quality feedback will be useful not only for the sender but also for other receivers and third-party monitors. The sender may modify its transmissions based on the feedback; receivers can determine whether problems are local, regional or global; network managers may use profile-independent monitors that receive only the RTCP packets and not the corresponding RTP data packets to evaluate the performance of their networks for multicast distribution.

Cumulative counts are used in both the sender information and receiver report blocks so that differences may be calculated between any two reports to make measurements over both short and long time periods, and to provide resilience against the loss of a report. The difference between the last two reports received can be used to estimate the recent quality of the distribution. The NTP timestamp is included so that rates may be calculated from these differences over the interval between two reports. Since that timestamp is independent of the clock rate for the data encoding, it is possible to implement encoding- and profile-independent quality monitors.

An example calculation is the packet loss rate over the interval between two reception reports. The difference in the cumulative number of packets lost gives the number lost during that interval. The difference in the extended last sequence numbers received gives the number of packets expected during the interval. The ratio of these two is the packet loss fraction over the interval. This ratio should equal the fraction lost field if the two reports are consecutive, but otherwise not. The loss rate per second can be obtained by dividing the loss fraction by the difference in NTP timestamps, expressed in seconds. The number of packets received is the number of packets expected minus the number lost. The number of packets expected may also be used to judge the statistical validity of any loss estimates. For example, 1 out of 5 packets lost has a lower significance than 200 out of 1000.

From the sender information, a third-party monitor can calculate the average payload data rate and the average packet rate over an interval without receiving the data. Taking the ratio of the two gives the average payload size. If it can be assumed that packet loss is independent of packet size, then the number of packets received by a particular receiver times the average payload size (or the corresponding packet size) gives the apparent throughput available to that receiver.

In addition to the cumulative counts which allow long-term packet loss measurements using differences between reports, the fraction lost field provides a short-term measurement from a single report. This becomes more important as the size of a session scales up enough that reception state information might not be kept for all receivers or the interval between reports becomes long enough that only one report

might have been received from a particular receiver.

The interarrival jitter field provides a second short-term measure of network congestion. Packet loss tracks persistent congestion while the jitter measure tracks transient congestion. The jitter measure may indicate congestion before it leads to packet loss. Since the interarrival jitter field is only a snapshot of the jitter at the time of a report, it may be necessary to analyze a number of reports from one receiver over time or from multiple receivers, e.g., within a single network.

5.2.2 Evaluation of Other Feedback Mechanisms

The IEEE 802.11 standard defines a mechanism by which RF energy is to be measured by the circuitry on a wireless NIC. This numeric value is an integer with an allowable range of 0-255 (a 1-byte value) called the Receive Signal Strength Indicator (RSSI). No vendors have chosen to actually measure 256 different signal levels, and so each vendor's 802.11 NIC will have a specific maximum RSSI value (*RSSI-Max*). For example, Cisco chooses to measure 101 separate values for RF energy, and their RSSI-Max is 100. Symbol uses an RSSI-Max value of 31. The Atheros chipset uses an RSSI-Max value of 60. Therefore, it can be seen that the RF energy level reported by a particular vendor's NIC will range between 0 and RSSI-Max. Therefore, RSSI is an arbitrary integer value, defined in the 802.11 standard and intended for use, internally, by the microcode on the adapter and by the device driver. For example, when an adapter wants to transmit a packet, it must be able to detect whether or not the channel is clear (i.e., nobody else is transmitting). If the RSSI value is below some very low value, then the chipset knows that the channel is clear. This is the *Clear Channel Threshold* and some particular RSSI value is associated with it. When an 802.11 client is associated to an access point and is roaming, there comes a point when the signal level received from the access point drops to a somewhat low value (because the client is moving away from the access point). This level is called the *Roaming Threshold* and some intermediate (but low) RSSI value is associated with it. Different vendors use different signal levels for the Clear Channel Threshold and the Roaming Threshold and, moreover, the RSSI value that represents these thresholds differs from vendor-to-vendor because different RSSI-Max values are implemented.

There is no specified accuracy to the RSSI reading. That is, there is nothing in the 802.11 standard that stipulates a relationship between RSSI value and any particular energy level as would be measured in mW or dBm. Individual vendors have chosen to provide their own levels of accuracy, granularity, and range for the actual power (measured as mW or dBm) and their range of RSSI values (from 0 to RSSI-Max).

The concept of *granularity* is important to consider here, too. Since the RSSI value is an integer it must increase or decrease in integer steps. For example, Symbol

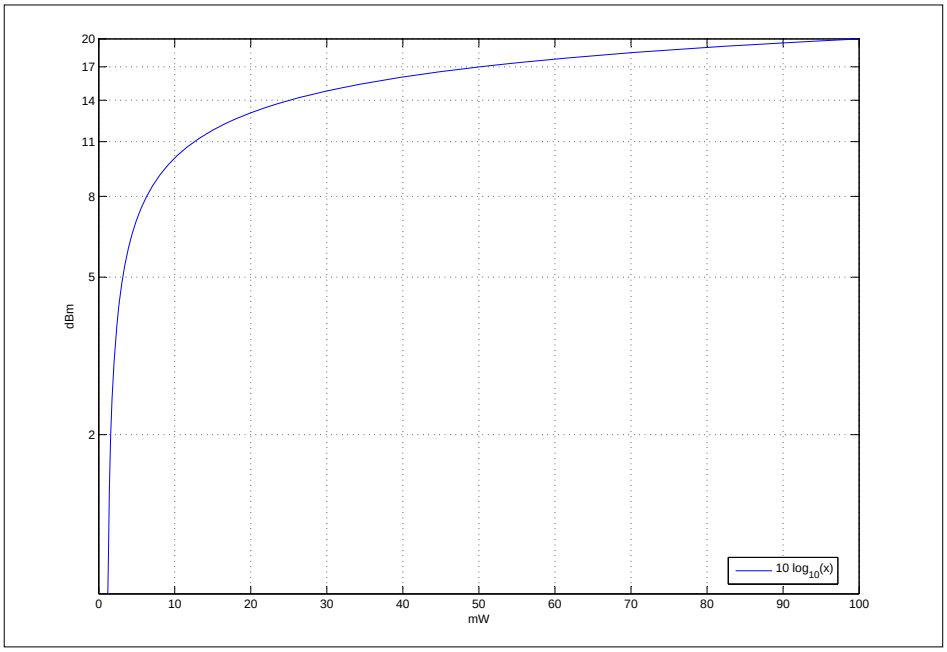


Figure 5.1: Relation between dBm and mW

provides 32 separate steps, Cisco provides 101 (i.e., from 0 to RSSI-Max for any given manufacturer). Whatever range of actual energy is being measured, it must be divided into the number of integer steps provided by the RSSI range. Therefore, if RSSI changes by 1, it means that the power level changed by some proportion in the measured power range. There are, therefore, two important considerations in understanding RSSI. First, it is necessary to consider what range of energy (the mW or dBm range) that's actually being measured. Secondly, it must be recognized that all possible energy levels (mW or dBm values) cannot be represented by the integer set of RSSI values.

As was seen in the dBm-to-mW graph above (figure 5.1), there is not much change in dBm values above roughly 5 mW. Wireless NIC manufacturers do not measure signal strength in that range. RF energy is almost always measured using dBm values, because the measured range would otherwise have mW values with too many zeros to the right of the decimal point. The graph also shows that the slope of change for dBm below 5 mW is very roughly linear, but not exactly. The logarithmic nature of the dBm measurement, coupled with the fact that the RSSI range used for measurement contains dBm *gaps* (due to the integer nature of the RSSI value), has led many vendors to map RSSI to dBm using a table. These mapping tables allow for adjustments to accommodate the logarithmic nature of the curve. The range

of energy that is typically measured begins at or below -10 dBm (and, compared to the +20 dBm of potential output power at a 100 mW access point, that's a relatively weak signal). In addition to the fact that the graph *flattens out* at higher power levels, the -10 dBm upper limit on the energy level measurement range is also consistent with the purpose for RSSI measurements in the first place. RSSI is intended for use in Clear Channel assessment and determination of the Roaming Threshold. It makes sense that the circuitry is designed to provide reasonable accuracy in this range.

To circumvent the complexities (and potential inaccuracies) of using RSSI as a basis for reporting dBm signal strength, it is common to see signal strength represented as a percentage. The percentage represents the RSSI for a particular packet divided by the RSSI-Max value (multiplied by 100 to derive a percentage). Hence, a 50% signal strength with a Symbol card would convert to an RSSI of 16 (because their RSSI-Max=31). Atheros, with RSSI-Max=60, would have RSSI=30 at 50% signal strength.

It can be seen that use of a percentage for signal strength provides a reasonable metric for use in network analysis and site survey work. If signal strength is 100%, that's great! When signal strength falls to roughly 20%, the Roaming Threshold is near. Ultimately, when signal strength is down somewhere below 10% (and probably closer to 1%), the channel is going to be assumed to be clear. This conceptualization obviates the need to consider dBm, the RSSI-Max, or the *knee* in the logarithmic curve of mW to dBm conversion. It allows a reasonable comparison between environments even though different vendor's NICs were used to make the measurements. Ultimately, the generalized nature of a percentage measurement allows the integer nature of the RSSI to be overlooked.

The range of possible measurements is below -10 dBm, which precludes using an off-the-shelf NIC for measurement of high-gain antennae, or anywhere close to an access point (where the signal level is above -10 dBm). In general, the use of a percentage value for signal strength allows for a relatively simple, consistent, reproducible metric that can be used as part of a site survey [Bar02].

In practice, it can be observed that above some particular signal strength (%) traffic moves at 11 Mbps (in 802.11b) and, as the percentage decreases, there's a point where the data begins moving at 5.5 Mbps. Still later the speed drops, ultimately, to 1 Mbps, and finally there are increased numbers of CRC errors with an ultimate loss of reception (see figure 2.4 on page 47).

Beside the signal strength information three other signal quality information must be considered. In Windows systems, the information about wireless cards can be easily accessed by using the *ndisprot*[1] protocol driver. In Linux, the simplest way

[1] `http://ramp.ucsd.edu/pawn/wrapi`

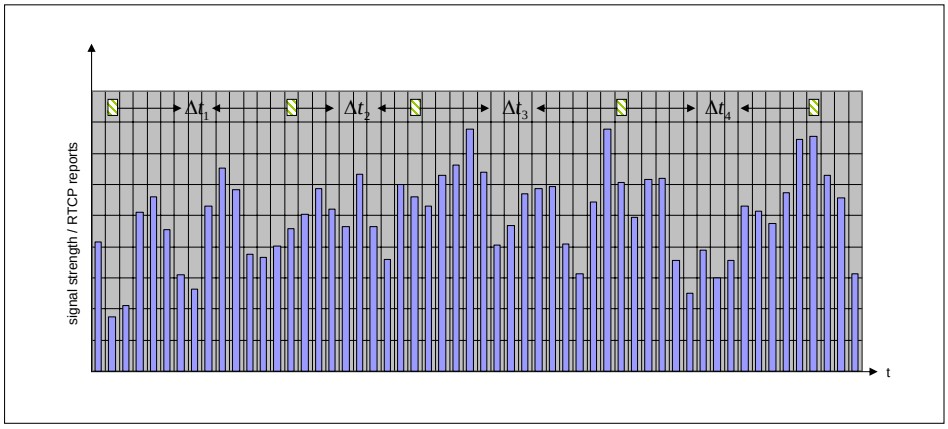


Figure 5.2: Frequency of signal quality / RTCP reports

to do this is using the pseudo-file `/proc/net/wireless`[2]. In this work Linux is used and, for this thesis purposes, the most relevant information in the pseudo-file `/proc/net/wireless` are:

- *link* - **quality** of the link (how good the received signal is)
- *level* - received **signal** strength (how strong the received signal is)
- *noise* - background **noise** strength when no packet is transmitted

In the following section these signal quality information will be included into a mathematical model for optimal parameterization of the proposed speech coding scheme, which is described later.

5.3 Codec Parameterization

As depicted in figure 5.2 the frequency Δt_i^{-1} of the RTCP reports may vary and furthermore depends on the amount of users which take part in the current RTP-session. Contrarily, the signal quality information are nearly constantly available - the frequency and accuracy mainly depends on the NIC of the wireless device. Hence to further improve the adaptivity both information (RTCP reports and signal quality information) are necessary and therefore must be considered in the mathematical model.

[2] the content of the pseudo-file `/proc/net/wireless` may vary a lot in function of the drivers used. For further information see [Bar02] and `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html`

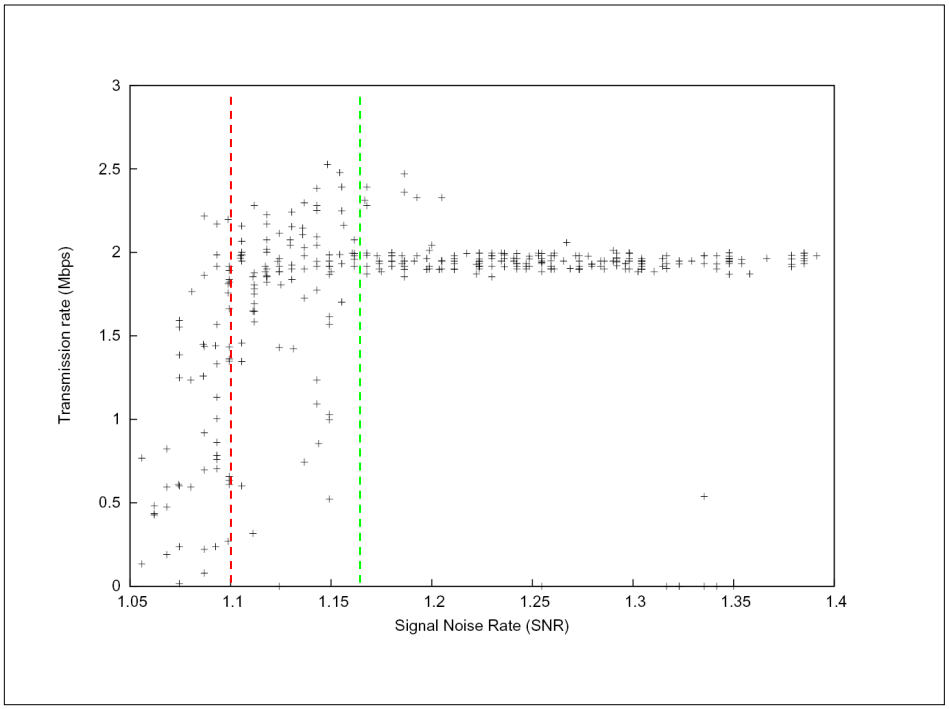


Figure 5.3: SNR and transmission rate correlation

In [dCK05] it was shown that their adaptive streaming algorithm reduces the waste of network capacity and, consequently, increases the effective throughput about 15% based on evaluation of signal quality information. Therefore two threshold values α and β were identified, where α is the smallest SNR value such that the transmission is considered good and, similarly, β is the smallest SNR value such that the transmission is acceptable. For example, in figure 5.3, α and β could be 1.16 and 1.1, respectively.

In order to optimize network utilization, considering a client/server streaming application, the algorithm adapts the streaming in function of signal quality information. It uses SNR, α and β, as follows:

- *Level 0*: if $SNR \geq \alpha$, then the server transmits normally
- *Level 1*: if $\alpha > SNR \geq \beta$, then the server acts in a conservative way, for example reducing the packet size
- *Level 2*: if $\beta > SNR \geq 1$, then the server reduces the transmission rate
- *Level 3*: if $SNR < 1$, then the transmission is interrupted until $SNR > 1$

The adaptive strategy is organized in levels such that clients can automatically find the α and β thresholds in function of its reception quality and application requirements. The algorithm could also keep the thresholds dynamically updated; experiments, however, showed similar performance between static and dynamic usage of α and β. Hence, it is not necessary to frequently update the thresholds but they must be recalculated in case of network configuration changes as, for example, when handoffs occur.

Comprehensive simulations with channel adaptivity regarding the evaluation of RTCP reports were made in [Gök05]. As mentioned earlier, channel adaptivity just based on these obtained results is not sufficient. The gaps between successive RTCP reports might be too large to compensate possible changes in the actual network condition. Therefore the following combined theoretical model has been developed. Both, signal quality information and RTCP evaluation results were considered in order to optimally handle changing network characteristics.

As seen, in order to adapt the α and β thresholds, information about SNR and the related transmission bit rate is required. The latter could be calculated by evaluating the RTCP reports. Besides that, the following information are considered:

- *(SR)* sender's packet count (32 bit) ← sufficient for approximately 23860.93 hours of conversation
- *(RR)* receiver's fraction lost (8 bit) ← sufficient for approximately 5.12 seconds (this is just the average delay between successive reports assumed a 2-person conversation - the latest revision of RTP tolerates even smaller delays, hence 8 bit are enough, even if all packets were lost, since the last transmission of a receiver report)
- *(RR)* receiver's total lost (24 bit) ← sufficient for approximately 93.2068 hours of conversation
- *(RR)* interarrival jitter (32 bit)
- *(RR)* delay since last report (DLSR) (32 bit)
- *(RR)* extended highest sequence number received (32 bit)

From here the following calculations respectively declarations could be made:

- the amount of lost packages between two successive reports, provides the exact amount of not received packages, these can be calculated from the last sequence numbers
- this number divided by the lag of corresponding NTP-timestamps results in the ratio of lost packets per second

- from the amount of delivered and not-delivered packages in relation to the passed time (known by the timestamps), the transmission rate could be calculated

- by the difference between the amount of expected minus the amount of lost packets, the exact number of received packages could be estimated

- the mean deviation of time, between sending and receiving a package could be an indication for imminent packet loss

Finally, the following model (depicted in figure 5.4) was developed to adapt the current transmission bit rate considering actual channel characteristics. Additionally the α and β thresholds are to be adapted every 30 seconds (approx. every 6 RTCP reports). If the channel characteristics changes fast, e.g. in case of moving/mobile users, these update interval could be decreased to further support the adaption process.

Inspired by the work in [dCK05] the developed channel adaption algorithm is based on both channel quality information provided by the RTCP reports and the NIC of the corresponding wireless device. The thresholds α, β, μ and η must be initialized with proper values. These have to be estimated in a multitude of experiments in the real world. Also the different adaption coefficients ($\pm\text{sqi}_{1,2}$ and $\pm\text{rtcp}_{1,2}$) have to be investigated in various tests. Until then, in the theoretical point of view, this adaption algorithm delivers two different adjustment parameters $\text{qsi}_{percent}$ and $\text{rtcp}_{percent}$. These parameters finally are weighted with two additional impact coefficients. These will be used to further improve the accuracy of the algorithm by balancing the importance of the qsi and the rtcp adjustment parameters. The final adjustment of the current output bit rate yields to

$$\text{output}^{(i+1)}_{percent} = \text{output}^{(i)}_{percent} + (\omega_{qsi} \cdot \text{qsi}_{percent} + \omega_{rtcp} \cdot \text{rtcp}_{percent})$$

Additionally the interarrival jitter information from the RTCP reports is used to adjust the jitter buffer. It is assumed, that this approach will be fast enough to respond to changing channel characteristics. Rationally, the fast update frequency for the qsi-information supported and enhanced by the infrequently rtcp-information, both together, should give reliable hints for the adaption of the output bit rate considering changing channel characteristics.

$\text{output}^{(i+1)}_{percent}$ is finally used to parameterize/configure the speech coder output bit rate. This value will be between 0 and 100 %, hence the extremes are *no output* respectively *full quality* with all enhancement levels.

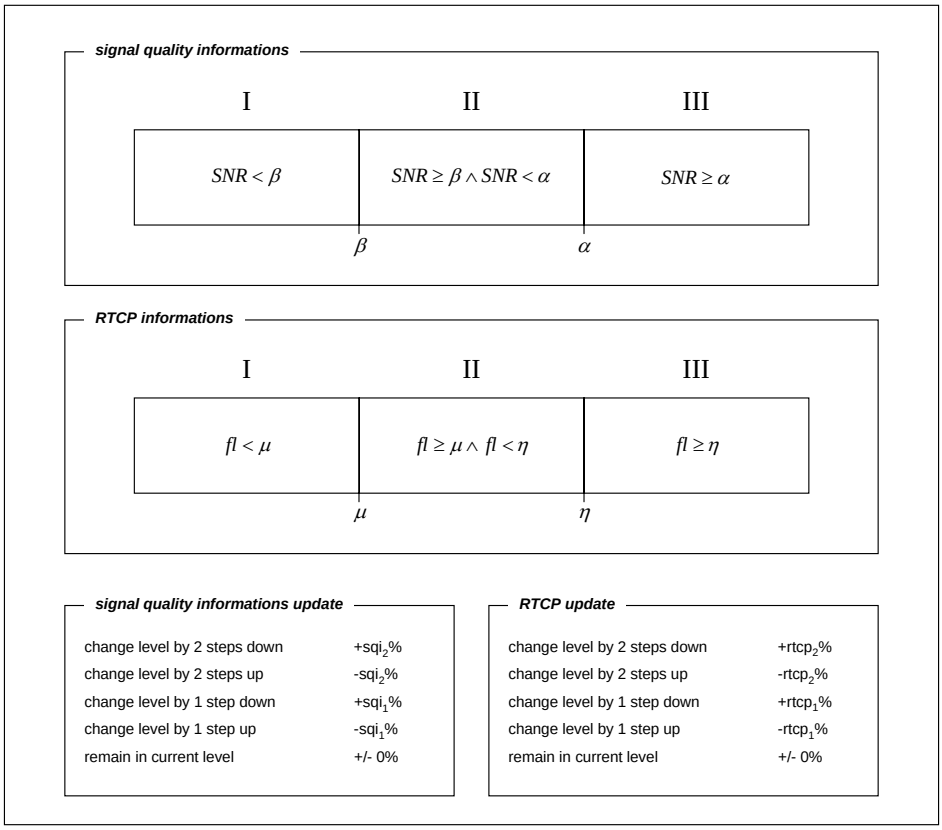


Figure 5.4: Channel adaption based on specific thresholds

5.4 Optimal Distribution of Data Streams

As seen later, the developed speech coding scheme is a multiple description coding scheme. Therefore, at least two *separated* signals from one source signal (the voice signal) will be produced, which are then transmitted over preferably two or more transmission channels to enhance reliability and robustness and therefore the overall speech quality. The theoretical approach seems to be clear but where do we find multiple transmission channels in the wireless world of signal transmission. The following sections present two concepts, in which at least two or more virtual/physical transmission channels exist. Open questions, in which way these channels are provided and how they can be optimally used for the purpose of speech transmission coded the multiple descriptive way arise.

The first concept is MIMO (Multiple Input - Multiple Output) which is proposed as the concept and key factor of 4G mobile communication [Gro05]. This technology

will be and is used to enhance the signal quality and transmission rates and finally to support more participants.

5.4.1 MIMO

Increasing demand for high performance 4G broadband wireless mobile calls for use of multiple antennas at both base station and subscriber ends. Multiple antenna technologies enable high capacities suited for Internet and multimedia services and also dramatically increase range and reliability. This design is motivated by the growing demand for broadband wireless Internet access. The challenge for wireless broadband access lies in providing a comparable quality of service for similar cost as competing wireline technologies. The target frequency band for this system is 2 to 5 GHz due to favorable propagation characteristics and low radio-frequency (RF) equipment cost. The broadband channel is typically non-LOS channel and includes impairments such as time-selective fading and frequency-selective fading. Multiple antennas at the transmitter and receiver provide diversity in a fading environment. By employing multiple antennas, multiple spatial channels are created and it is unlikely all the channels will fade simultaneously.

OFDM is chosen over a single carrier solution due to lower complexity of equalizers for high delay spread channels or high data rates. A broadband signal is broken down into multiple narrowband carriers (tones), where each carrier is more robust to multipath. In order to maintain orthogonality amongst tones, a cyclic prefix is added which has length greater than the expected delay spread. With proper coding and interleaving across frequencies, multipath turns into an OFDM system advantage by yielding frequency diversity. OFDM can be implemented efficiently by using FFT's at the transmitter and receiver. At the receiver, FFT reduces the channel response into a multiplicative constant on a tone-by-tone basis. With MIMO, the channel response becomes a matrix. Since each tone can be equalized independently, the complexity of space-time equalizers is avoided. Multipath remains an advantage for a MIMO-OFDM system since frequency selectivity caused by multipath improves the rank distribution of the channel matrices across frequency tones, thereby increasing capacity.

MIMO is a promising technology to achieve gain in channel capacity at multi-antenna systems. Thereby this gain depends from the signal-to-noise ratio (SNR) at the receiver. So a capacity gain can be either used to increase the distance between sender and receiver entities or to increase the bit-rate of the system. In MIMO systems there are different strategies to achieve the capacity gain.

Array Gain describes the increase in signal-to-noise ratio at the receiver that results from the coherent superposition effect of the signal. The receiver knows about

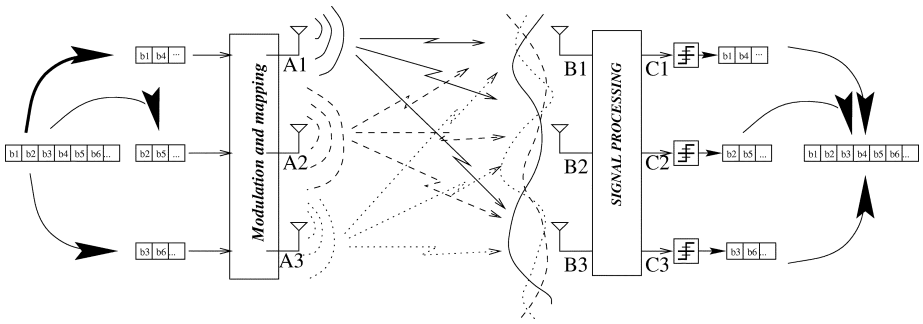


Figure 5.5: Basic spatial multiplexing scheme with three TX and three RX antennas (following [GSS+03])

the characteristics of the wireless channel and modifies the incoming signal so that they coherently add up to a stronger signal.

Diversity Gain The wireless channel has the specific problem of multipath fading or fast fading. Sending the data with diversity is a common strategy to combat the fast fading effect. Diversity means, that the sender sends replicas of the transmitted signal to the receiver to achieve redundancy. In conventional SISO (Single Input - Single Output) systems diversity can be achieved by sending out replicas over time or frequency. In addition to that MIMO-systems enables us to send out replicas over space. The last diversity scheme is the antenna diversity that can effectively defeat multipath-fading. To understand this, we can take an intuitive approach: in conventional multipath-fading wireless channels only very small changes in the position of the receivers antenna can turn the channel from a good fading state in a bad state - and vice versa when assuming the Gilbert-Elliot model for the behaviour of the channel. Taken the same scenario for a MIMO-System equipped with multiple sender and receiver antennas. Assumed that all the antennas have a sufficient distance from each other it is probably that at least one antenna is in a good state.

Multiplexing Gain Both approaches mentioned above are based on the idea to send the same signal over multiple antennas. In contrast to this, at the approach of spatial multiplexing the transmitter multiplexes the bit stream to be transmitted in multiple streams and transmits each of them simultaneously over an own antenna. Figure 5.5 shows the basic scheme of spatial multiplexing. The receiver has knowledge of the characteristics of the wireless channel. With this information the receiver is able to reconstruct the transmitted bit streams and to combine them to the original bit-stream.

Therefore, MIMO is suitable for the purpose of MDC speech coding. As seen later, the serial speech signal is converted (described in multiple ways) in order to transmit the resulting single streams over multiple MIMO channels enhancing robustness and so the overall speech quality.

The second concept of multiple transmission channels is the concept of mobile ad-hoc networks (MANETs), which is described in the following section.

5.4.2 MANETs

Mobile Ad-hoc Networks (MANETs) are formed by mobile wireless hosts without the need of an existing infrastructure, unlike wireless cellular systems which require a centralized control and support system at the base station. Most of the wireless systems deployed today are also centralized systems, wherein the nodes connected to the network communicate through an access point. Interactive voice communication over a wireless mobile ad-hoc network is a challenging problem because of the error prone wireless channel, the changing topology of the network, delays involved in establishing a new link or finding a new route, and the current MAC protocols which were not developed for real-time multimedia communication. One important example is ad-hoc networks based on the IEEE 802.11 standard.

The IEEE 802.11 standard was designed primarily for non-real-time transfer of data and these protocols may not be suitable for real time interactive multimedia. The IEEE 802.11 MAC protocols are designed to minimize collisions and depend on retransmissions to ensure successful transmission of a packet irrespective of the delay incurred by the packet. For good quality conversational voice communication, end to end delay of packets must be under 150 ms for the delay to be imperceptible to the listener. Most of the prior work in this area has been on changes that can be made at the MAC layer to minimize delays due to retransmissions and reduce packet losses due to bit errors.

The 802.11 MAC layer retransmits a packet until the packet is acknowledged by the receiver. Retransmissions increase delay and also cause congestion in the network. Many schemes to reduce retransmission of speech packets have been proposed [DCBR$^+$04, SM04, PSM04, SM03].

Another method to improve reliability of transmission over a MANET is to use path diversity, i.e. send data simultaneously through multiple paths. The probability of all the paths breaking down simultaneously is low and hence the probability of packet loss is reduced, but, sending multiple copies of the same packet is inefficient usage of bandwidth. To improve bandwidth efficiency, a source coding diversity scheme like multiple description coding (MDC) can be used with path diversity.

Using a multiple description coder for voice communication over MANETs was first suggested in [DG$^+$04], where the authors proposed a new MD codec based on the AMR-WB codec and showed that at high error rates in the channel, the MD

codec performs better than a single description (SD) codec sent over a single path.

5.5 Summary

In order to provide continuous wireless transmissions, hence uninterrupted wireless voice communication, additional efforts have to be made. In this chapter an adaption model was developed which is based on two information which are received by the transmission control protocol and on the other hand by the wireless NIC itself. Both information combined provide a parameter for the fine-grain bit rate scalable unit which adapts the data rate appropriately.

Future live tests have to show which configuration of the proposed model yields to the best results - if a best configuration exist at all. This pessimistic attitude results from the fact, that wireless live tests are extremely difficult to handle. No test sequence could be like the other possibly introducing high fluctuation, hence, comparable results are rare if existent at all.

But nevertheless, the ability to scale the data rate at a fine-grain level is shown in the next chapter.

6 Scalability

6.1 Introduction

Audio coding algorithms with bit rate scalability allow an encoder to transmit data at a high bit rate and decoders to successfully decode a lower-rate bitstream contained within the high-rate code. For example, an encoder might transmit 64 Kbps while a decoder would decode at 32, 16 or 8 Kbps according to channel bandwidth, decoder complexity and quality requirements. Scalability is becoming an important aspect of low bit rate audio/speech coding, particularly for multimedia and voice applications where a range of coding bit rates may be required, or where bit rate fluctuates.

Fine-grain scalability, where useful increases in coding quality can be achieved with small increments in bit rate, is particularly desirable.

The growth of the Internet has created a large demand for high-quality streamed audio content. Audio coding with fine-grain bit rate scalability allows uninterrupted service in the presence of channel congestion, achieves real-time streaming with low buffer delay, and yields the most efficient use of available channel bandwidth [Dun01].

While fine-grain bit rate scalability can be extremely useful, it is important that it is achieved without significant coding efficiency penalty relative to fixed bit rate systems, and with low computational complexity.

6.2 Layered Coding

Layered coding is a family of signal representation techniques in which the source information is partitioned into a sets called layers. The layers are organized so that the lowest, or base layer, contains the minimum information for intelligibility. The other layers, called complementary layers, contain *add-on* information which improves the overall quality of the signal. Usually, layered encoding schemes are organized so that some layers (in particular the base layer) are mandatory to reconstruct a coherent signal. Using such schemes then requires either that the net be able to discriminate between packets and provide packets that carry important information with a guaranteed performance (in particular guaranteed maximal loss rate) service, or that these packets be *protected* against loss. This can be done using

a variety of so-called unequal error protecting schemes, which have been the subject of much research effort (e.g. [GV93]).

It is assumed that there exists some sort of prioritized transmission channels. This can be realized by physically disjoined channels with different signal energy or by adding a priority flag to data packets with intermediate routers handling these packets. One can also send multiple copies of the prioritized data and only single instances of non-prioritized one. The channels have not to be physical distinct, one also can use virtual different channels. Of course the data of the base layer is transmitted via the channel with the highest priority and the enhancement layers via the less reliable ones.

In speech coding similar partitioning can be used: with LPC for example the coefficients and pitch parameters could be contained in the base layer and excitation information in the enhanced layers. One can also put the most significant bits into the base and the less significant ones into the enhancement layer(s). Alternative partitioning criteria could be signal-strength, difference to previous packets or if packets can be well predicted from previous ones. It has to be mentioned that transmission errors in the base layer can lead to disastrous results, because it will normally not be possible to create a useful signal only from the enhancement layers.

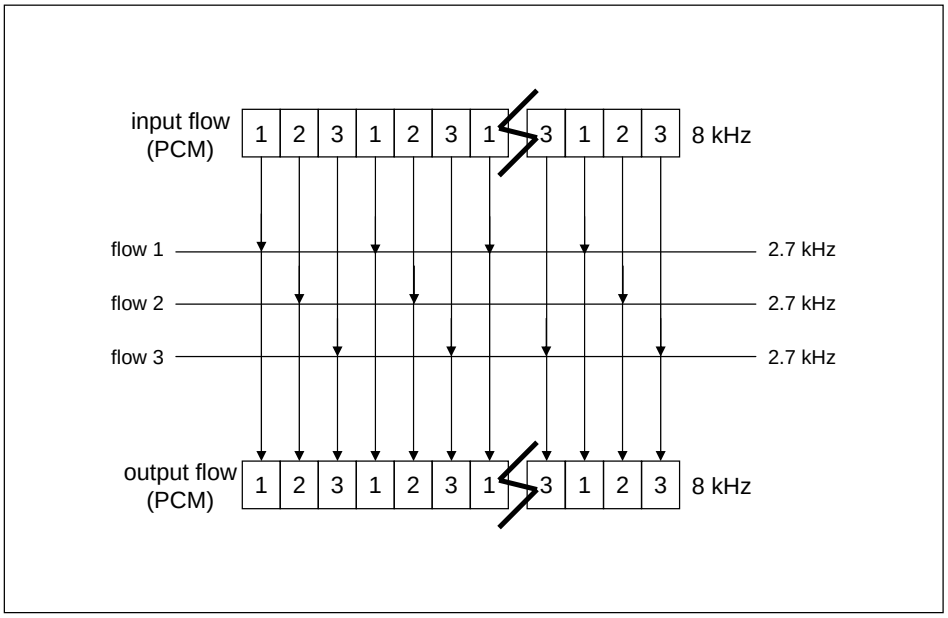
In addition to the intended error robustness, layered encoding can also be used to support multiple receivers which differ in the decoders complexity or network speed: less performant units can just use the base layer, while more performant units can use one or more enhancement layers for decoding.

6.3 Multiband-/Subband Coding

The simplest balanced scheme one can think of is based on straight subsampling. The coding algorithm is based on a temporal subband decomposition. As an example, the case of a PCM (Pulse Coded Modulation) encoded 8 kHz audio signal decomposed into three PCM 2.7 kHz flows is considered, as shown in figure 6.1.

The temporal decomposition algorithm is carried out at the source done for each audio chunk (which typically includes 20 ms or 30 ms of audio). At a destination, a receiver which receives all 3 flows can retrieve the original input signal. If one or two flows are missing, the destination uses the samples received to reconstruct an approximation of the original signal. An upsampling one-to-three flows is shown in figure 6.2. Therefore, the larger the number of received flows is, the better the quality of the reconstructed signal can be.

Temporal decomposition handles signals sampled with different sampling rates. For example, a 48 kHz audio signal can be decomposed into three 16 kHz subflows, each of which can be decomposed into two 8 kHz subflows, which finally yields eighteen 2.7 kHz audio layers.



(following [TPB97])

Figure 6.1: Hierarchical coding scheme

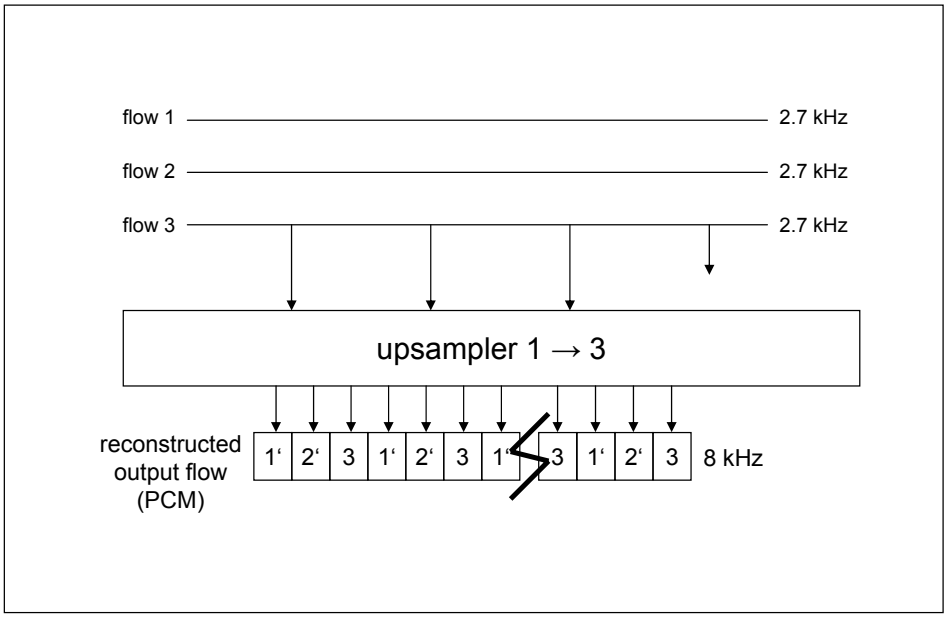


Figure 6.2: Upsampling 1→3 (reconstruction)
(following [TPB97])

The coding scheme described above has the interesting balanced property mentioned above: all the layers have the same importance in the network. However, in case of severe congestion, the congestion control algorithm may select to subscribe to only one flow and in this case, the coding has no mechanism at all to reconstruct lost samples of audio. To achieve robustness against packet loss, the receiver could decide to subscribe to at least 2 flows. One can decide to add to the base flow of each sample, the flow L corresponding to the previous sample (L is the number of layers of the coding). So, in such a scheme, the first layer has twice the bandwidth of other layers and the transmission control scheme handles $(L-1)$ flows instead of L. In contrary to other redundancy schemes, there is no bandwidth wasted when packet loss is null since the *redundancy* information appended to the first layer is always used to improve the reconstruction of the sample. The redundancy added to flow #1 does not imply that this flow has a higher priority since flow #1 is not required to receive it correctly to decode the other flows. Whereas there is no flow more important than another one, all receivers have to adopt the same order in the subscription algorithm (in order to perform efficient pruning to limit the bandwidth sent).

6.3.1 Splitting

The developed speech codec is based on a two stage scheme, hence two subprocesses will be created. Each subprocess handles a different frequency range. While one process handles the range 0-4 kHz the other copes with the range from 4 to 8 kHz. One can think of different strategies to split the frequency range into two or multiple subranges. As future developments of the codec should provide a wider frequency range multiple subranges have to be created aimed at supporting multiple-layered coding. Therefore, instead of using just a simple high- or lowpass a bandpass is used to split the signal into two subranges as detailed earlier.

6.3.1.1 Digital Filters

In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. There are two main kinds of filters, analog and digital. They are quite different in their physical makeup and in how they work.

An analog filter uses analog electronic circuits made up from components such as resistors, capacitors and op amps to produce the required filtering effect. Such filter circuits are widely used in such applications as noise reduction, video signal enhancement, graphic equalizers in Hi-Fi systems, and many other areas. There are well-established standard techniques for designing an analog filter circuit for a

given requirement. At all stages, the signal being filtered is an electrical voltage or current which is the direct analogue of the physical quantity (e.g. a sound or video signal or transducer output) involved.

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialized DSP (Digital Signal Processor) chip. The analog input signal must first be sampled and digitized using an ADC (analog to digital converter). The resulting binary numbers, representing successive sampled values of the input signal, are transferred to the processor, which carries out numerical calculations on them. These calculations typically involve multiplying the input values by constants and adding the products together. If necessary, the results of these calculations, which now represent sampled values of the filtered signal, are output through a DAC (digital to analog converter) to convert the signal back to analog form.

Fast DSP processors can handle complex combinations of filters in parallel or cascade (series), making the hardware requirements relatively simple and compact in comparison with the equivalent analog circuitry.

The following list gives some of the main advantages of digital over analog filters:

- a digital filter is programmable, i.e. its operation is determined by a program stored in the processor's memory. This means the digital filter can easily be changed without affecting the circuitry (hardware). An analog filter can only be changed by redesigning the filter circuit.
- digital filters are easily designed, tested and implemented on a general-purpose computer or workstation.
- the characteristics of analog filter circuits (particularly those containing active components) are subject to drift and are dependent on temperature. Digital filters do not suffer from these problems, and so are extremely stable with respect to both - time and temperature.
- unlike their analog counterparts, digital filters can handle low frequency signals accurately. As the speed of DSP technology continues to increase, digital filters are being applied to high frequency signals in the RF (radio frequency) domain, which in the past was the exclusive preserve of analog technology.
- digital filters are very much more versatile in their ability to process signals in a variety of ways; this includes the ability of some types of digital filter to adapt to changes in the characteristics of the signal.

A digital filter is simply a discrete-time, discrete-amplitude convolver. Basic Fourier transform theory states that the linear convolution of two sequences in the

time domain is the same as multiplication of two corresponding spectral sequences in the frequency domain. Filtering is in essence the multiplication of the signal spectrum by the frequency domain impulse response of the filter.

There are two basic types of digital filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The general form of the digital filter difference equation is:

$$y(n) = \sum_{i=0}^{N} a_i x(n-i) - \sum_{i=1}^{N} b_i y(n-i)$$

where $y(n)$ is the current filter output, the $y(n-i)$'s are previous filter outputs, the $x(n-i)$'s are current or previous filter inputs, the a_i's are the filter's feed forward coefficients corresponding to the zeros of the filter, the b_i's are the filter's feedback coefficients corresponding to the poles of the filter, and N is the filter's order. IIR filters have one or more non-zero feedback coefficients. That is, as a result of the feedback term, if the filter has one or more poles, once the filter has been excited with an impulse there is always an output. FIR filters have no non-zero feedback coefficient. That is, the filter has only zeros, and once it has been excited with an impulse, the output is present for only a finite (N) number of computational cycles.

Because an IIR filter uses both a feed-forward polynomial (zeros as the roots) and a feedback polynomial (poles as the roots), it has a much sharper transition characteristic for a given filter order. Like analog filters with poles, an IIR filter usually has nonlinear phase characteristics. Also, the feedback loop makes IIR filters difficult to use in adaptive filter applications.

Due to its all zero structure, the FIR filter has a linear phase response when the filter's coefficients are symmetric, as is the case in most standard filtering applications. A FIR's implementation noise characteristics are easy to model, especially if no intermediate truncation is used. An IIR filter's poles may be close to or outside the unit circle in the Z plane. This means an IIR filter may have stability problems, especially after quantization is applied. An FIR filter is always stable. FIR filters also allow development of computationally efficient architectures in decimating or interpolating applications.

6.3.1.2 Ideal and Real Digital Filters

Ideal filters, regarding their edge steepness, have vertical slopes, hence very precise filtering or splitting is possible. But real filters couldn't be as exact as ideal filters. In order to reach the quality of an ideal filter high order filters have to be designed. Figure 6.3 and figure 6.4 illustrate ideal and windowed filters of order 32 and 256 respectively.

Figure 6.5 depicts the magnitude response for different digital filters - a bandpass

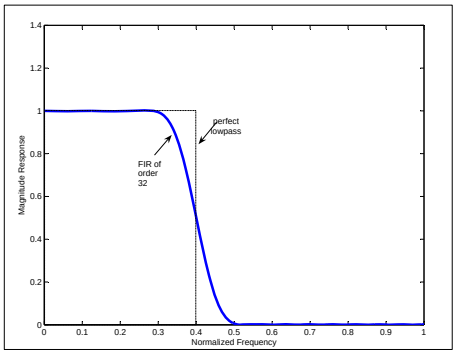


Figure 6.3: FIR order 32

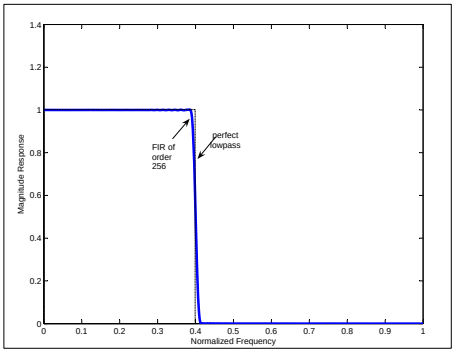


Figure 6.4: FIR order 256

with normalized frequency range limited from 0.45 to 0.55 with different orders ranging from 16 to 512.

The higher the order of the digital filter the more accurate it is, hence, the splitting into multiple subbands is much more precise but the computational complexity raises too as the number of operations increases within the convolution. The results with filters of order≥200 are sufficient. If the splitting is not precise enough, compression is less effective, because already considered frequency parts would be integrated into the whole compression process multiple times. Furthermore, if imprecise filters are used, the magnitude responses aren't flat, hence specific frequency parts would be amplified as others would be damped, resulting in a degraded signal.

Figure 6.6 depicts self-designed filters (bandpass and highpass) of order 200. A Bartlett-Hanning window was applied on both filters. This is defined by:

$$w[k+1] = 0.62 - 0.48 \left| \left(\frac{k}{n-1} - 0.5 \right) \right| + 0.38\, cos \left(2\pi \left(\frac{k}{n-1} - 0.5 \right) \right)$$

The advantage of using a Bartlett-Hanning window is that the magnitude response is then free of heavy ripples, hence a very clean response signal can be applied in order to filter the speech signal (compare figure 6.5 with figure 6.6. The grayed area is handled by a NB-speech codec, while the top frequencies (above 4 kHz) are split into two subbands (4-6 kHz and 6-8 kHz) by a bandpass and a highpass respectively. These two subbands are the two enhancement layers of the developed speech codec scheme, as described later.

6.3.1.3 Fast Fourier and Wavelet Transformation

Two other possible solutions regarding the splitting process are the FFT and the (F)WT. The FFT transforms the signal representation from the time-domain into

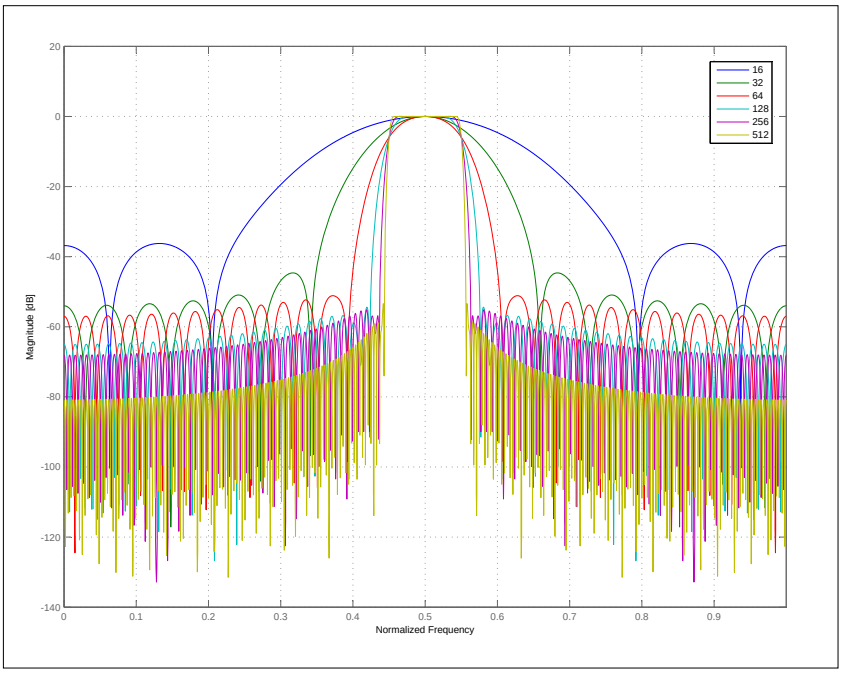


Figure 6.5: Magnitude response for different bandpass filters

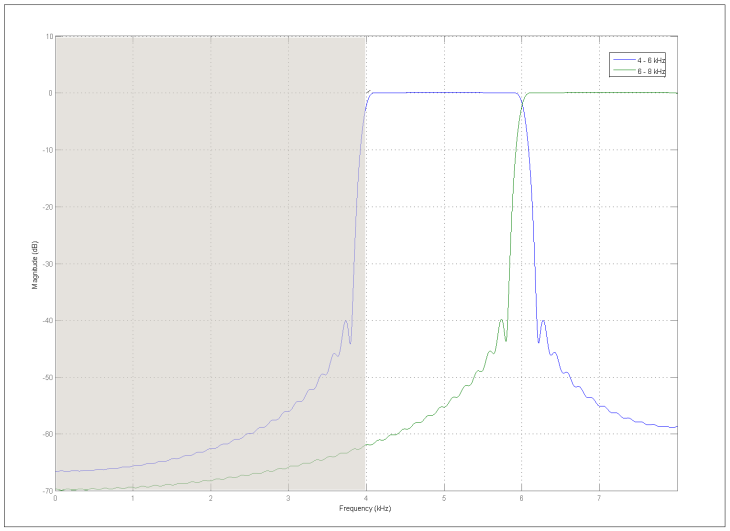


Figure 6.6: Magnitude response for a self-designed band-/highpass

the frequency-domain. By zeroing particular Fourier coefficients, which are responsible for a specific frequency, different filter operations like e.g. low-/high-/bandpass are feasible. Besides the FFT, the (fast) wavelet transform (FWT) could be used to split the source signal into multiple subbands. However, a specific characteristic of this proceeding is that the resulting single subbands comprise information from the other subbands, hence a perfect clear splitting isn't possible. Further studies have to show, that an approach based on wavelet-subband splitting yields to useful signal information which could be further processed properly. It might be even advantageous regarding the multiple description of the source signal, hence, the frequency information which are spread out over all subbands. If some frequency information get lost, it might be, that these lost information could be reconstructed from the given data in the other subbands. It is like to utilize so called *zero-trees*, which are described later, but not for compression but for reconstruction purposes.

6.4 Wavelet Approach

Among the current popular speech compression methods, for example, G.721, G.722, G.728, etc., the speech coding algorithms used are based on the assumption that the speech signal is stationary. But speech/audio signals are non-stationary in nature. Otherwise, these analyses are generally performed in time-domain or frequency-domain. Each of them only uses partial characteristics of an audio signal. Therefore, these methods have some limitation.

The wavelet transform is a time-frequency localization analysis method for non-stationary signals and has been identified as an effective tool for data compression [WV97, ST93]. Why wavelet theory can break through Fourier analysis, is that it changes the uniform band structure to non-uniform band structure, which is more suitable to non-stationary signals. The window width of wavelet analysis is adjustable, which is shorter window at higher frequencies and larger window at lower frequencies. The time-frequency characteristic of a wavelet filterbank is a natural match to some of the properties of wideband speech and audio signals.

To achieve high quality at low bit rate an speech coding scheme must take advantage of the masking effect in human hearing. Auditory masking is a psychoacoustic phenomenon that renders low level signals concentrated in a given frequency region inaudible in the presence of higher level signals at neighboring frequencies. Utilizing this nature, psychoacoustic model shapes the spectrum of the quantization noise properly so that the noise is masked by the audio signal existing simultaneously. Hence, the high quality audio signal is achieved.

It is known that wideband speech and audio coding based on the wavelet transform and psychoacoustic models can achieve perceptually transparent quality at low bit rate. In [ST93], Sinha and Tewfik detail an optimal wavelet structure selection

and dynamic dictionary coding scheme to achieve almost transparent coding of CD-quality signals at bit rates of 48-66 Kbps. This work offers unique insights into the use of adaptive wavelet transforms to audio/speech compression. The major drawback of the method is the large computational effort associated with wavelet packet decomposition and dictionary search.

Wavelet Representation of Audio/Speech Signals In multi resolution analysis, wavelet function $\{\Psi_{j,n}\}$ forms an orthonormal basis for the space of square integrable functions, which is denoted by $L^2(R)$. The following relationship between wavelet function and scaling function $\{\Phi_{j,n}\}$ is satisfied.

$$\langle \Phi_{j,n}, \Phi_{j+1,k} \rangle = h_{n-2k}$$
$$\langle \Phi_{j,n}, \Psi_{j+1,k} \rangle = g_{n-2k}$$

Where $\{h_k\}$ and $\{g_k\}$ is a pair of quadrature mirror filters which satisfy the following condition. The former is a lowpass filter and the later is a bandpass filter

$$\sum_n h_{n-2k} h_{n-2l} = \delta_{kl}, \sum_n h_n = \sqrt{2}$$
$$g_n = (-1)^n h_{-n+1}.$$

A digital speech/audio signal $f(n) \in L^2(R)$ is represented as wavelet decomposition

$$f(n) = \sum_{j=1}^{J} \sum_k \Psi_{j,k} d_{j,k} + \sum_k \Phi_{J,k} c_{J,k}$$

where c_J is the approximation of the original signal at resolution j and d_j denotes the details between c_j and c_{j-1}

$$c_{j,k} = \sum_n h_{n-2k} c_{j-1,n},\ 1 \le j \le J$$
$$d_{j,k} = \sum_n g_{n-2k} c_{j-1,n},\ 1 \le j \le J$$

It is an effective path to speech and audio coding using the human auditory characteristic. Most progress in audio compression in recent years can be attributed to successful application of psychoacoustics to signal compression - see e.g. MP3.

6.4.1 Schemes for Coefficient Selection and Reduction

Considering image coding systems, the superior compression performance of wavelet based coding systems over their DCT based counterpart might suggest that the improved performance was primarily made by replacing DCT with wavelet transform, and hence the choice of transform would matter the most to coding efficiency. However, in strict technical terms, all existing transforms used in signal compression by themselves do not lead to any data reduction. Both DCT and dyadic wavelet transforms, the two most widely used types of transforms in compression, generate as many coefficients as the number of samples. Furthermore, while the original sample values of digital signals are integers, the transform coefficients are non-integers. Therefore, without efficient coding of transform coefficients, a transform not only cannot compress but can even expand the data. The main benefit of transform to data compression is from its property of energy packing. A suitable transform can transfer the majority of signal energy into a few transform coefficients, resulting in a large number of zero and near-zero coefficients. In other words, the probability distribution of transform coefficients is much more biased than that of original samples. The more biased the distribution, the easier it is to compress signals by entropy coding.

Despite the well accepted folklore that lossy signal compression is better done via transform coding, it is the process of entropy coding that actually achieves data reduction. Informally, entropy coding refers to a family of coding techniques that uses shorter codewords for more probable symbols (smaller transform coefficients), and longer codewords for less probable symbols (larger transform coefficients). An optimal variable length code can achieve an average code length that approaches the information theoretic lower bound called entropy, hence the term entropy coding. Entropy coding is also referred to as noiseless or lossless coding since the coding process is perfectly reversible.

Hence, compression efficiency and the resulting speech quality depends on the proper selection of the wavelet coefficients, which for example could be left out or which must be handled very carefully, hence a modification of such coefficients results in a huge unwanted and maybe disturbing change in the final reconstructed signal.

In the following paragraphs, some of the most important schemes or algorithms for coefficient selection and encoding will be discussed.

Near-Zero Thresholding As stated earlier, signal transformations achieved by for example the DCT, KLT or in this case the wavelet transform concentrate the signal energy to a very few transform coefficients while the remaining are zero or near by zero. These coefficients don't contribute much, therefore they could be omitted or at least efficiently be coded to gain a compression effect.

This yields to a very simple approach for wavelet-based compression schemes. By thresholding the near by zero or exactly zero coefficients signal compression ratios of about 8:1 are quite achievable. Hence, about 80% to 95% of the coefficients fall into the category of near by zero or exactly zero - but in a few cases such high ratios aren't achievable, hence the efficiency depends on the input signal exceedingly.

But this approach is to simple, as various experiments showed. Crackling, plopping sounds are disturbing the reconstructed signal after compression. Therefore more advanced techniques have to be considered.

Sparsity-Norm Balance Thresholding With the near-zero thresholding method, it is hard to determine exactly where the optimal threshold is. If possible, we would like to construct an algorithm that guesses an appropriate threshold for a signal. One such method is the sparsity-norm balancing method. Under this heuristic, we test successive thresholds and calculate the norm of the resulting coefficients after each threshold. The p-norm of a vector x is given by ([Lax97]):

$$|x|_p = \left(\sum_i |x_i|^p \right)^{1/p}.$$

The goal in calculating the norm of the coefficients is to balance thresholding with loss in signal quality. The metric used to measure loss of signal quality is the norm, which actually calculates how much total *energy* we lose after compression. Now, if we were to use the vector p-norm, we would use the 2-norm also written l^2-norm given by:

$$|x| = \sqrt{\sum_{k=1}^{n} |x_k|^2}.$$

Since we just want to calculate how much total energy is lost in the signal after compression, this simple heuristic works well and is easy to compute to find the right balance. The norm of the wavelet coefficients of the signal for uniformly distributed threshold values between 0 and 1 is calculated. For each threshold, the ratio

$$100 \cdot \left(\frac{|x_{thr}|}{|x|} \right)^2,$$

where x_{thr} denotes the thresholded coefficients and x the untouched coefficients, is determined. When the threshold is 0, the percent number of zeros should be close to 0%, while the percent norm is 100%. However, when the threshold is at 1, we would expect the percent zeros to be 100%, while the percent norm is 0%. Thus, the curves of the two quantities intersect at some point, and we set the global threshold to be

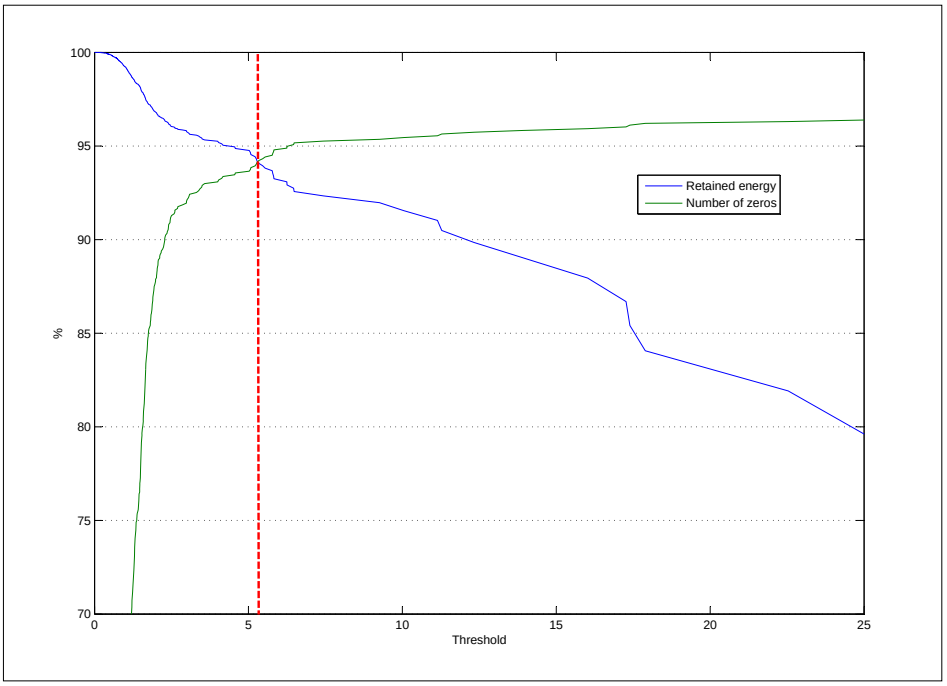


Figure 6.7: Balance sparsity-norm

at this intersection. This method, borrowed from MATLAB's Wavelet Toolbox, is called balanced sparsity-norm and examplaryly is depicted in figure 6.7.

Zero-trees Common approaches for 2-D image compression like EZW (embedded wavelet zero tree) and SPHIT (set partitioning in hierarchical trees) use a fixed significance-tree that captures well the inter- and intraband correlations of wavelet coefficients. For 1-D audio signals, such rigid coefficient correlations are not present.

Significance-tree coding algorithms like EZW [Sha93] or SPHIT [SP96] exploit the fact that it can be beneficial to describe significant coefficients of a bitplane via their position and value information instead of transmitting all values one by one. These spatial orientation trees can be mathematically represented using parent-children coefficient coordinate relationships.

Figure 6.8 shows the case of image compression, were the offspring $O(i,j)$ of the wavelet parent coefficients at position (i,j), except for the highest and lowest pyramid level, have been defined as

$$O(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\}\,.$$

Due to the fact that the 2-dimensional wavelet transformation has a typical coef-

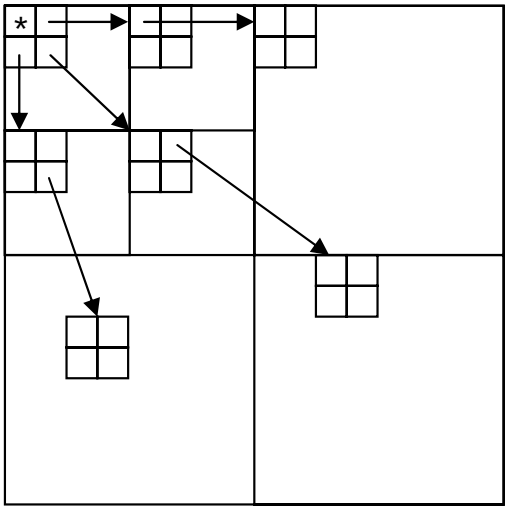


Figure 6.8: 2-D zerotree

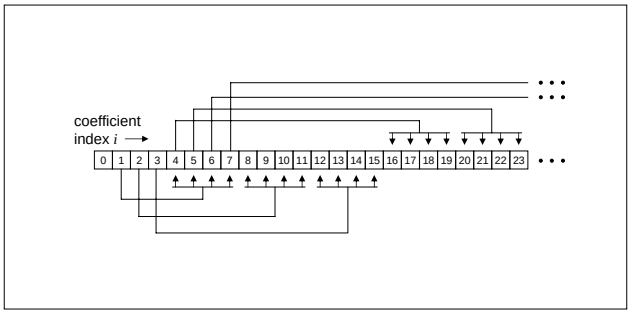


Figure 6.9: 1-D SPHIT

ficient inter- and intraband correlation [LK02], this rigid tree structure can capture the correlation with a reasonable computational complexity, giving an efficient compression scheme.

For 1-dimensional audio signals, the problem of selecting the optimal tree structures remains unsolved despite considerable efforts [SZM05]. Most existing algorithms use a single type of tree as shown in figure 6.9 with the fixed parent-children relationship $O(i) = iN + \{0, 1, \cdots, N-1\}$ for different positive integers N. For the MDCT transform, $N = 4$ was adopted in [Dun01] and the wavelet packet transform was encoded using $N = 2$ in [LP98].

Short Review of Other Schemes Enormous efforts have been made in the study of efficient wavelet-based compression for the 2-D and lately also in the 1-D field of audio/speech signals. A short review of the most prominent approaches for the 1-D case is given in this paragraph.

Recent improvements in the field of audio/speech coding schemes where based on the utilization of perceptual information. In [ACRG99] the authors presented a novel algorithm for scalable coding of wideband audio. The algorithm incorporates perceptual considerations into an adaptive rate-distortion (RD) framework for zero-tree coding of a wavelet-decomposed signal. The transform and coding steps are both (frame) adaptive, where the adaptation involves RD optimization. The resulting coder is hence called the *perceptual zero-tree wavelet* (PZW) coder.

An alternative approach to achieving scalability is ordered *bitplane coding* of transform coefficients, where in each frame coefficient bitplanes are coded in order of significance, beginning with the most significant bits (MSBs) and progressing to the least significant bits (LSBs). This results in fully-embedded coding where the bitstream at a certain rate contains all lower-rate codes, and exhibits fine-grain scalability in contrast to the coarse granularity offered by error-feedforward systems [Dun01]. Bitplane coding can also yield a significant increase in encoding speed since quantization typically requires a single scan through the transform coefficients for each frame, as opposed to the recursive bit allocation search executed in fixed rate coding. Ordered bitplane coding is used in the Bit-Sliced Arithmetic Coding (BSAC) system described by Park et al. [PKS97]. While BSAC coder is reported to perform well in terms of coding efficiency, it requires the use of arithmetic coding which can increase computational complexity, and bit rate granularity is limited to 1 Kbps enhancement steps.

The *EZK* algorithm described in [LDS99] is refinement of SPHIT for use with uniform transform decompositions. A marker is used to record the progress of a scan through the coefficients from low to high frequency, and if new significant coefficients are found within the current bitplane, then the position of the next significant coefficient relative to the marker is recorded by run-length coding. Insignificant coefficients between the marker and the next significant coefficient are moved to the list of insignificant coefficients (LIC), the marker updated and remaining coefficients scanned for further significant descendants. The process is repeated until all coefficients have been scanned.

EZK was previously reported in [LDS99] to achieve improved coding efficiency compared to a *set partitioning in hierarchical trees* (SPHIT) implementation with a tree hierarchy where each parent has only a single offspring. However, such a SPHIT arrangement is unlikely to be effective because it cannot efficiently predict the location of insignificant coefficients. More significance map coding is achieved when each parent in the hierarchy has many children. The problem is how to map th one-to-many hierarchy to a 1-D transform array suitable for audio coding.

Figure 6.9 shows one possible tree hierarchy where each parent coefficient has 4 child coefficients that are clustered together in frequency. Coding efficiency results in [Dun01] indicate that this SPHIT arrangement achieves a significant performance improvement over EZK.

6.5 Summary

Different strategies with their own disadvantages and advantages were presented and discussed for splitting the input signal into two or multiple subbands. Fine-grain scalability depends on differentiated sub frequency ranges which are encoded independently to provide a construction set of base and enhancement layers which in combination yield to basic or high quality signals. Motivated by the aim of a future development of the codec supporting a wider range as of just the range between 0-8 kHz a bandpass filter is used to split the signal information into two sub frequency ranges with high slopes reminiscent of ideal digital filters.

Furthermore current strategies for wavelet based compression and respective coefficient selection were presented. These approaches served for the development of the new strategy which additionally integrates a psychoacoustic model which is presented in the final chapter. But before, transmission robustness and schemes for enhancement will be discussed in the next chapter.

7 Robustness and Optimization

7.1 Introduction

Most existing speech and audio coders were developed to meet a single purpose of delivering the best quality possible under fixed constraints in bit-rate, computational complexity, and algorithmic delay. Recent development in network communications demands the additional capability to cope with the packet-lossy nature associated with these networks. The problem lies within the research area of multiple description coding (MDC). It can efficiently combat packet loss without any retransmission thus satisfying the demand of real time services and relieving the network congestion [BZZ05].

7.2 Multiple Description Coding

Communication networks, such as data, voice, and wireless, can be viewed as time-varying channels in which bandwidth, delay, and complexity constraints fluctuate for applications at the terminals. Speech coding and transmission, perhaps the most popular among such network applications, face challenges different from the more traditional setting of single-channel, homogeneous environment, and fixed-constraints. First, variations in bandwidth and complexity dictate the system be able to operate under different capacities, and provide different service qualities accordingly. More importantly, many networks employ the packet-transmission scheme, and traffic congestion might delay some packets beyond the time slot allowed by protocols that they are considered lost. Therefore, distortion caused by packet loss dominates that caused by individual bit errors, and cannot be corrected from the conventional error-control coding. Requesting and waiting for retransmission of lost information has serious drawbacks: it increases bandwidth and end-to-end delay without guaranteed improvement, degrades efficiency in multi-cast applications if only a few group members suffer packet loss, and in some protocols such as UDP, a reverse link to the transmitter simply does not exist. On the other hand, receiver-based packet recovery methods, such as frame substitution and interpolation [Goo86, ST89], produce acceptable quality only if the duration of loss is very short.

A better solution to cope with packet loss lies within the realm of multiple de-

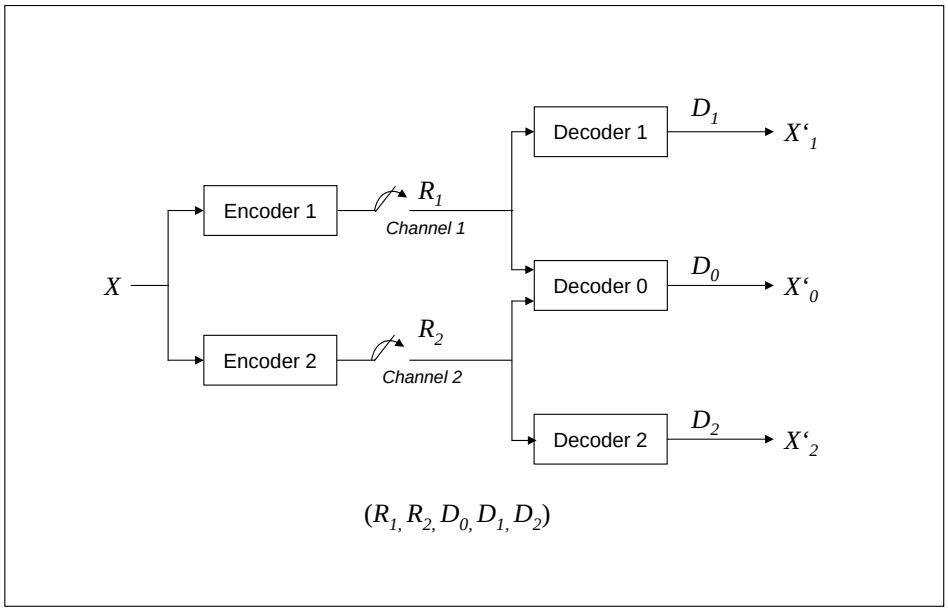


Figure 7.1: A two-channel multiple description coder

scription coding (MDC) [ZRH05]. Its general setup is as follows: a signal source X is given sufficient resources to be coded and transmitted separately over N different channels, each with bit rate R_n. All channels have the same destination, but one or more might be broken, resulting in a complete loss of data they carry. Status of each channel is independent from each other, and is known to receiver/decoders by reading off header information, but unknown to transmitter/encoders. The receiver reconstructs X from the arrived data packets (descriptions), with a corresponding distortion measure D. If all N channels fail simultaneously, a rather unlikely event in network communications, the system has to resort to other less desirable packet-concealment techniques as described in section 3.4.4.

Otherwise, a non-empty subset of information about X arrive at the receiver, and is used for signal reconstruction. There are a total of $2^N - 1$ such subsets, each delivered by a different channel-bank status. Accordingly $2^N - 1$ decoders are built at the receiver side, each designed to reconstruct X based on available information. A best-case scenario with all data packets received is processed by a central decoder, with a central distortion D_0, and each of the rest by its corresponding side decoder, with side distortion $D_n, n = 1 \ldots 2^N - 2$. Thus, at any given time, the system operates at one of the $2^N - 1$ possible points in its rate-distortion region. Figure 7.1 illustrates a MDC system for $N = 2$.

Existing MDC schemes can roughly be divided into three categories:

- quantizer based
- transform based
- erasure codes based

Considerable work is done in the field of quantization based schemes which include scalar quantization in [Vai93, VD94], trellis coded quantization in [VBC98, WO01] and vector quantization in [FE99, GKK02]. Transform based approaches which include correlating transforms in [OWVR97, GK01] and overcomplete expansions in [BDV00, GKK01]. Recently, schemes based on erasure codes have been introduced in [PPR04, PPR01].

Since channels' conditions are unknown to the transmitter, they are of equal reliability, and all encoders are of equal importance. Therefore, each encoder-channel module is given an equal bit rate R_n, and tries to describe X with equal distortion. Layered coding, also known as multi-resolution coding or embedded coding, is an *unbalanced* version of MDC in that the N modules are of different significance, and can be ranked and layered by the importance of information each conveys. A top layer carries a basic information set to serve as the core. Each additional layer, of little value by itself, enhances the output quality successively when it is incorporated back into the previous layers. This structure is more suitable to applications where network's capacity fluctuates, and demands varying output quality in return. However, as will be detailed later, the underlying fundamentals of MDC and layered coding are the same, and as such they encounter similar limitations and tradeoffs. This is particularly true when the signal in interest is speech.

There are currently two distinct camps in MDC study. One concentrates in its information theory aspect, and attempts to derive a system's rate-distortion function under various conditions, or at least its upper/lower bounds. The other leans more toward practicality, and use combinations or slight modifications of existing coding algorithms in each encoder-channel module to provide protection against channel failure.

7.2.1 Information Theory Aspect of MDC

Theoretical work for two-channel MDC originated from a series of papers in the early 1980s [Oza80, WWZ80, GC82]. In information theory, the problem is to determine the achievable $(R_1, R_2, D_0, D_1, D_2)$-tuples for various input signals and distortion measures. A complete characterization of this relationship was derived by Ozarow [Oza80] for the special case of independently and identically distributed Gaussian source with square error as the distortion. Two years later El Gamal and Cover constructed region bounds for the more general case of memoryless input [GC82].

Although an exact mathematical expression of MDC's rate-distortion function is very difficult to obtain, its basic properties can be drawn from intuitive analysis. Supposed a single description (SD) coding of input X yields a minimum (or acceptable) distortion D^* at rate R^*. In designing a multiple description (MD) coder, one naive approach would be to install the identical SD system for each channel. Under this arrangement, output distortion reaches the minimum as long as one channel delivers its packet successfully, i.e. $D_n = D^* \, \forall n$. Information available from other functioning channels increases the rate but does not reduce the distortion. At the other extreme, SD bit-stream is partitioned into non-overlapping subsets, and each is sent over a channel as a packet. Such design optimizes $D_0 = D^*$, but renders all other D_n's unacceptably high because at least a subset of vital information is withheld from each side decoder. A MDC system is of practical purpose only if the inferior side decoders are indeed put into use at some point during communication, therefore designs which sacrifice one completely for the other should be avoided.

A tradeoff between those two extremes is achieved by introducing controlled redundancy among the packets. This procedure enables a side decoder to estimate some information not directly received, and lower its output distortion. Considered the two-channel case which fits most MDC studies. Inter-channel redundancy increases $R_{1,2}$ from $R^*/2$ toward R^*, pushing $D_{1,2}$ arbitrarily close to D^*. With its distortion remaining at $D_0 = D^*$, the central decoder now operates at a higher rate $R^* + \rho$, with the redundant rate $\rho > 0$. Such deviation from the achievable rate-distortion function by the central decoder is inevitable for reducing side distortions to an acceptable level. Thus, for a given ρ, MDC distributes this redundant rate within the system and attempts to jointly minimize the two competing distortion measures, $D_{1,2}(\rho)$ and $D_0(\rho)$. This is the essence of multiple description coding.

It should be noted that the introduced redundancy are among packets/channels. Each packet itself still assembles data as orthogonal as possible to convey the maximum amount of information, similar to traditional source coding. Analogous to traditional channel coding, each basic unit contains knowledge about its neighbors to provide mutual protection, except this basic unit vulnerable to channel error is now a self-contained information packet instead of a binary bit. For this nature, multiple description coding is also known as joint source-channel (JSC) codes.

7.2.2 Practical MDC Systems

Parallel to researches in the theoretical field, many practical MDC systems have been developed. In MD scalar quantizers and vector quantizers [Vai93, SVS99], each channel's information about indices enables its side decoder to reconstruct X to within a quantization cell. When both channels deliver their data, output by the central decoder is refined to an intersection of two quantization cells, therefore the distortion is much smaller. Such systems are optimized for D_0. MD-VQ in [KGK00]

provides more flexible tradeoffs between D_0 and $D_{1,2}$. MDC also appeared in the more complicated transform domain [Goy98, YOVR01]. For an input signal vector X, its MDC transform imposes correlations among coefficients in different channels, while still tries to maintain orthogonality among those sharing the same path. Each side decoder receives half of all transform coefficients, and tries to estimate the other half. X is then reconstructed by performing the inverse transform. Under some specific assumptions, [Goy98] derives the optimal transform for a two-dimensional X, as well as its side distortions. Transform for signal vectors of dimension greater than three have not been obtained. As a compromise, [YOVR01] partitions the input into pairs, and applies a two-variable transform to each. Coefficients of each pair are then transmitted via separated channels. In such arrangement, the system needs to setup the optimal pairing in addition to finding the optimal transform. Applications in sub-band [YR98] and wavelet-based [SRVN98] coders also emerged.

Vast majority of today's MD speech coders can be categorized into two groups: after first decomposing the input into two non-overlapping portions, A and B, they either exploit inherent correlation naturally existed within speech, or artificially create and attach information about each other. The first group includes the earliest MD speech coder in [JC81], where two channels contain alternating PCM waveform samples, and two side decoders construct each missing sample by interpolating its surviving neighbors. Since there are already much dependency between adjacent speech samples, this scheme requires no increase in bit rate, but it may not be suitable for rapidly changing signals. [IV95] extends the idea to DPCM, with all odd-indexed prediction residuals being sent in one channel, and all even-indexed residuals in the other. Prediction coefficients within are modified to create non-white residuals, thus allowing side decoders to estimate unavailable information with a certain degree of confidence.

In the second group, each information packet in a channel contains two distinct components about the input X: a high-resolution description about portion A, X_A, and a coarse description about portion B, $\hat{X}_B$. The other packet carries $\hat{X}_A$ and X_B. The central decoder then reconstructs X using the information set $[X_A, X_B]$. Methods of description include waveform quantizer [SO99], polyphase transform [JO00], and parametric representation [AMV00]. Many other schemes [BVG96], which lean more toward computer science with little signal processing involved, simply employ a higher-rate coder to produce $X_{A,B}$, and a lower-rate coder for $\hat{X}_{A,B}$. This second group is by far the most popular structure being used. All these systems introduce inter-channel redundancy to reduce side distortions at the expense, hopefully modest, of overall source coding efficiency.

IEEE WLAN Standard	OTA Estimates	MAC SAP Estimates
802.11b	11 Mbps	5 Mbps
802.11g	54 Mbps	25 Mbps (if .11b is not present)
802.11a	54 Mbps	25 Mbps
802.11n	200+ Mbps	100 Mbps

Table 7.1: Comparison of different 802.11 transfer rates

7.2.3 Analysis

Almost all research in information theory of MDC assumes memoryless Gaussian input, with square error as distortion measure. Such model does not fit the nature of speech signal at all. It also ignores the well known phenomenon that speech quality, thus distortion, is faithfully measured only by an unique set of subjective criterion, which is still not fully understood. On the other hand, a more realistic system uses an existing algorithm, and conveniently inherits all speech-specific features possessed by the chosen coder (analysis, coding, perceptual weighting, etc). These coders, though near optimal in the single description sense after decades of thorough research effort, do not offer the necessary redundancy for channel failure protection. A sensible balance in between is needed in designing better MDC systems for speech.

7.2.4 IEEE 802.11n

In response to growing market demand for higher-performance wireless local area networks (WLANs), the Institute of Electrical and Electronics Engineers - Standards Association (IEEE-SA) approved the creation of the IEEE 802.11 Task Group N (802.11 TGn) during the second half of 2003. The scope of TGn's objective is to define modifications to the Physical Layer and Medium Access Control Layer (PHY/MAC) that deliver a minimum of 100 Mbps throughput at the MAC SAP (see table 7.1).

The Wi-Fi Alliance has also shown interest in TGn's work toward 802.11n. Industry representatives have come together under the Wi-Fi Alliance - High Throughput Marketing Task Group to define and publish a Marketing Requirements Document (MRD). The Wi-Fi Alliance MRD specifies performance expectations that will enhance the end-user experience in regard to increased throughput, increased range, more robustness to interference and a more reliable user experience throughout the Basic Service Set (BSS).

There are three key areas that need to be considered when addressing increases in wireless LAN performance. First, improvements in radio technology will be needed to increase the physical transfer rate. Second, new mechanisms implementing the effective management of enhanced PHY performance modes must be developed. Third, improvements in data transfer efficiency are needed to reduce the perfor-

mance impacts of PHY headers and radio turnaround delays that would otherwise reduce the improvements achieved with increases in physical transfer rate. At the same time, while developing new approaches to achieve performance, coexistence with existing 802.11a/b/g legacy devices is required. All of these areas must be addressed when considering practical and effective implementations for cost-sensitive market segments.

One approach to increasing the physical transfer rate of wireless systems employs multiple antenna systems for both the transmitter and the receiver. This technology is referred to as multiple-input multiple-output (MIMO), or smart antenna systems. MIMO exploits the use of multiple signals transmitted into the wireless medium and multiple signals received from the wireless medium to improve wireless performance. MIMO can provide many benefits, all derived from the ability to process spatially different signals simultaneously. Two important benefits explored here are antenna diversity and spatial multiplexing. Using multiple antennas, MIMO technology offers the ability to coherently resolve information from multiple signal paths using spatially separated receive antennas. Multipath signals are the reflected signals arriving at the receiver some time after the original or line of sight (LOS) signal has been received. Multipath is typically perceived as interference degrading a receiver's ability to recover the intelligent information. MIMO enables the opportunity to spatially resolve multipath signals, providing diversity gain that contributes to a receiver's ability to recover the intelligent information.

Another valuable opportunity MIMO technology may provide is Spatial Division Multiplexing (SDM). SDM spatially multiplexes multiple independent data streams, transferred simultaneously within one spectral channel of bandwidth. MIMO SDM can significantly increase data throughput as the number of resolved spatial data streams is increased. Each spatial stream requires its own TX/RX antenna pair at each end of the transmission. It is important to understand that MIMO technology requires a separate radio frequency (RF) chain and analog-to-digital converter (ADC) for each MIMO antenna. This increasing complexity ultimately translates to higher implementation costs as higher-performance systems are required.

It is expected, that MIMO technology will play an important role in achieving the IEEE TGn goals. MIMO technology should be used in IEEE 802.11n to evolve the existing OFDM physical interface presently implemented with legacy 802.11a/g. However, practical solutions will likely require additional technological approaches. Implementations requiring more than two RF antenna chains will need to be carefully architected to keep costs down while maintaining performance expectations.

Intensi-fi is Broadcom's implementation of the 802.11n draft specification and the industry's first draft-802.11n solution. These first chipsets comprise the following solutions:

- *BCM4321* - a draft-802.11n media access controller (MAC) and baseband

processor, providing data rates of over 300 Mbps and interfacing to PCI, Cardbus and PCI-Express hosts

- *BCM2055* - fifth-generation 802.11 radio, which integrates multiple 2.4 and 5 GHz radios to support simultaneous spatial streams for draft 802.11n products with 2x2, 3x3, or 4x4 antenna configurations.

Therefore, IEEE 802.11n is positioning MDC-based encoding schemes, hence more robust speech communication in case of wireless networks.

7.3 Error Handling and Reconstruction

If speech information is lost, it is impractical to try to retrieve it. The best option is to camouflage the missing information with something that is similar to the original signal. One approach to concealing lost speech packets is to replay the last packet in place of the lost one. Replaying the previous packet is a simple solution that is acceptable for rare packet loss, however, a more sophisticated method of concealment is needed in situations of frequent packet loss.

One technique for handling frequent packet loss is to estimate the information that would have been in the packet. This form of packet loss concealment generates synthetic speech to cover the missing data. Ideally, the concealment should have spectral characteristics similar to the speaker. For CELP based codecs such as G.729, this is relatively easy since the talker's speech is modeled during encoding. Concealment is more difficult with waveform codecs such as G.711 since the amplitude of the waveform is coded rather than assumptions about how the sound was produced. Packet loss concealment in waveform coding adds computational complexity, memory requirements, and delay. However, waveform codecs like G.711 can recover from packet loss more rapidly since the first speech sample in the first good packet restores the speech to the original, whereas CELP based codecs require a few frames to catch up.

During the normal operation of a speech codec, a receiver decodes packets and immediately sends the output to the audio port. When there is packet loss concealment, a copy of the output is saved in a circular history buffer that calculates the current pitch and waveform characteristics. With the first bad packet, the contents of circular history buffer are used to generate a synthetic replacement signal for the duration of the concealment. When two consecutive frames are lost, repeating a single pitch can result in harmonic artifacts or beeps that are noticeable when the erasure lands on unvoiced speech sounds such as $/s/$, $/f/$, and $/sh/$ or rapid transitions such as the stops $/p/$, $/k/$, and $/d/$. For this reason, concealment algorithms often increase the number of pitch periods used to synthesize the replacement signal

when multiple packets have been lost. This increases the variation in the signal, which is more consistent with real speech and improves the quality dramatically.

Although concealment that is derived from speech codecs uses the spectral characteristics of the speaker, it is important to ensure there is a smooth transition between the synthesized signal and the real speech signal. The first good packet after an erasure needs to be merged smoothly into the synthesized signal. To do this, synthesized speech from the pitch buffer is continued beyond the end of the erasure and mixed with the real signal. This allows a smoother transition between synthesized and real speech.

An error concealment procedure has been incorporated in G.729 to reduce the degradation in the reconstructed speech because of frame erasures in the bitstream. This error concealment process is functional when the frame of coder parameters (corresponding to a 10 ms frame) has been identified as being erased. The mechanism for detecting frame erasures is not defined in the recommendation [Sta96], and will depend on the application.

The concealment strategy has to reconstruct the current frame, based on previously received information. The method replaces the missing excitation signal with one of similar characteristics, while gradually decaying its energy. This is done by using a voicing classifier based on the long-term prediction gain, which is computed as part of the long-term postfilter analysis. The long-term postfilter (see section 4.2.1 of [Sta96]) finds the long-term predictor for which the prediction gain is more than 3 dB. For the error concealment process, a 10 ms frame is declared periodic if at least one 5 ms subframe has a long-term prediction gain of more than 3 dB. Otherwise the frame is declared non-periodic. An erased frame inherits its class from the preceding (reconstructed) speech frame. The voicing classification is continuously updated based on this reconstructed speech signal. The specific steps taken for an erased frame are:

- repetition of the synthesis filter parameters
- attenuation of adaptive and fixed-codebook gains
- attenuation of the memory of the gain predictor
- generation of the replacement excitation

These steps are detailed in [Sta96].

Concealment in iLBC is done in this way. If packet loss occurs, the decoder receives a signal saying that information regarding a block is lost. For such blocks it is recommended to use a packet loss concealment (PLC) unit to create a decoded signal that masks the effect of that packet loss. In the following an example of a PLC unit that can be used with the iLBC codec will be described. The PLC

described operates on the LP filters and the excitation signals and is based on the following principles:

- *Block Received Correctly and Previous Block Also Received* if the block is received correctly, the PLC only records state information of the current block that can be used in case the next block is lost. The LP filter coefficients for each sub-block and the entire decoded excitation signal are all saved in the decoder state structure. All of this information will be needed if the following block is lost

- *Block Not Received* if the block is not received, the block substitution is based on a pitch-synchronous repetition of the excitation signal, which is filtered by the last LP filter of the previous block. The previous block's information is stored in the decoder state structure. A correlation analysis is performed on the previous block's excitation signal in order to detect the amount of pitch periodicity and a pitch value. The correlation measure is also used to decide on the voicing level (the degree to which the previous block's excitation was a voiced or roughly periodic signal). The excitation in the previous block is used to create an excitation for the block to be substituted, such that the pitch of the previous block is maintained. Therefore, the new excitation is constructed in a pitch-synchronous manner. In order to avoid a buzzy-sounding substituted block, a random excitation is mixed with the new pitch periodic excitation, and the relative use of the two components is computed from the correlation measure (voicing level). For the block to be substituted, the newly constructed excitation signal is then passed through the LP filter to produce the speech that will be substituted for the lost block. For several consecutive lost blocks, the packet loss concealment continues in a similar manner. The correlation measure of the last block received is still used along with the same pitch value. The LP filters of the last block received are also used again. The energy of the substituted excitation for consecutive lost blocks is decreased, leading to a dampened excitation, and therefore to dampened speech.

- *Block Received Correctly When Previous Block Not Received* for the case in which a block is received correctly when the previous block was not, the correctly received block's directly decoded speech (based solely on the received block) is not used as the actual output. The reason for this is that the directly decoded speech does not necessarily smoothly merge into the synthetic speech generated for the previous lost block. If the two signals are not smoothly merged, an audible discontinuity is accidentally produced. Therefore, a correlation analysis between the two blocks of excitation signal (the excitation of

the previous concealed block and that of the current received block) is performed to find the best phase match. Then a simple overlap-add procedure is performed to merge the previous excitation smoothly into the current block's excitation.

7.3.1 Digital Audio Restoration

The application of digital signal processing (DSP) to problems in audio has been an area of growing importance since the pioneering DSP work of the 1960s and 70s. In the 1980s, DSP micro-chips became sufficiently powerful to handle the complex processing operations required for sound restoration in real-time, or close to real-time.

The major application for digital audio restoration can be found as vast amounts of important audio material, ranging from historic recordings of the last century to relatively recent recordings on analogue or even digital tape media, were noise-reduced and re-released on CD for the increasingly quality-conscious music enthusiast. Indeed, the first restorations were a revelation in that clicks, crackles and hiss could for the first time be almost completely eliminated from recordings which might otherwise be unreleasable in CD format.

Until recently, however, digital audio processing has required high-powered computational engines which were only available to large institutions who could afford to use the sophisticated digital remastering technology. With the advent of compact disc and other digital audio formats, followed by the increased accessibility of home computing, digital audio processing is now available to anyone who owns a PC with sound card, and will be of increasing importance, in association with digital video, as the multimedia revolution continued into the current millennium. Digital audio restoration will thus find increasing application to sound recordings from the Internet, home recordings and speech, and high-quality noise-reducers will become a standard part of any computer system and Hi-Fi system, alongside speech recognizers and image processors [GR98].

Analogue restoration techniques have been available for at least as long as magnetic tape, in the form of manual cut-and-splice editing for clicks and frequency domain equalization for background noise (early mechanical disc playback equipment will also have this effect by virtue of its poor response at high frequencies). More sophisticated electronic click reducers were based upon highpass filtering for detection of clicks, and lowpass filtering to mask their effect [JG73]. None of these methods was sophisticated enough to perform a significant degree of noise reduction without interfering with the underlying signal quality. For analogue tape recordings the pre-emphasis techniques of Dolby [Dol67] have been very successful in reducing the levels of background noise in analogue tape, but of course the pre-emphasis has to be encoded into the signal at the recording stages.

Digital methods allow for a much greater degree of flexibility in processing, and hence greater potential for noise removal, although indiscriminate application of inappropriate digital methods can be more disastrous than analogue processing. Research groups from many places, including Cambridge, Le Mans, Paris and the US, have worked in the area, developing sophisticated techniques for treatment of degraded audio. For a good overall text on the field of digital audio, including restoration [GRC98], see [BK98]. Another text which covers many enhancement techniques related to those presented here is by Vaseghi [Vas96].

There are several distinct types of degradation common in audio sources. These can be broadly classified into two groups: localized degradations and global degradations. Localized degradations are discontinuities in the waveform which affect only certain samples, including clicks, crackles, breakages and clipping. Global degradations affect all samples of the waveform and include background noise (*hiss*), and certain types of non-linear distortion.

7.3.1.1 Click Removal

At the heart of most click removal methods is an interpolation scheme which replaces missing or corrupted samples with estimates of their true value. It is usually appropriate to assume that clicks have in no way interfered with the timing of the material, so the task is then to fill in the *gap* with appropriate material of identical duration to the click. This amounts to an interpolation problem which makes use of the good data values surrounding the corruption and possibly takes account of signal information which is buried in the corrupted sections of data. An effective technique will have the ability to interpolate gap lengths from one sample up to at least 100 samples at a sampling rate of 44.1 kHz [GR98].

The interpolation problem may be formulated as follows. Considered N samples of audio data, forming a vector x. The corresponding click-degraded data vector is y, and the vector of detection values i_t is i (assumed known for the time being). The audio data x may be partitioned into two subvectors, one containing elements whose value is known (i.e. $i_t = 0$), denoted by $x_{-(i)}$, and the second containing unknown elements which are corrupted by noise ($i_t = 1$), denoted by $x_{(i)}$. Considered, for example, the case where a section of data of length l samples starting at sample number m is known to be missing or irrevocably corrupted by noise. The data is first partitioned into three sections: the unknown section $x_{(i)} = [x_m, x_{m+1}, \ldots, x_{m+l-1}]^T$, the m samples to the left of the gap $x_{-(i)a} = [x_1, x_2, \ldots, x_{m-1}]^T$ and the remaining known samples to the right $x_{-(i)b} = [x_{m+l}, \ldots, x_N]^T$:

$$x = \left[{x_{-(i)a}}^T {x_{(i)}}^T {x_{-(i)b}}^T\right]^T$$

Formed then a single vector of known samples $x_{-(i)} = \left[{x_{-(i)a}}^T {x_{-(i)b}}^T\right]^T$. For

more complicated patterns of missing data a similar but more elaborate procedure is applied to obtain vectors of known and unknown samples. Vectors y and i are partitioned in a similar fashion. The replacement or interpolation problem requires the estimation of the unknown data $x(i)$, given the observed (corrupted) data y. Interpolation will be a statistical estimation procedure for audio signals, which are stochastic in nature, and estimation methods might be chosen to satisfy criteria such as minimum mean-square error (MMSE), maximum likelihood (ML), maximum a posteriori (MAP) or some perceptually based criterion. Numerous methods have been developed for the interpolation of corrupted or missing samples in speech and audio signals [Tuk71, PV90].

7.3.1.2 Detection of Clicks

In the last section methods for interpolation of corrupted samples were presented. All of these methods assumed complete knowledge of the position of click-type corruption in the audio waveform. In practice of course this information is completely unknown a priori and some kind of detection procedure must be applied in order to extract the timing of the degradation. There are any number of ways by which click detection can be performed, ranging from entirely ad hoc filtering methods through to model based probabilistic detection algorithms.

Click detection for audio signals involves the identification of samples which are not drawn from the underlying clean audio signal; in other words they are drawn from some spurious *outlier* distribution. Various criteria for detection are possible, including minimum probability of error and related concepts, but strictly speaking the aim of any audio restoration scheme is to remove only those artifacts which are audible to the listener. Any further processing is not only unnecessary but will increase the chance of distorting the perceived signal quality. Hence a truly optimal system should take into account the tradeoff between the audibility of artifacts and perceived distortion as a result of processing, and will involve consideration of complex psycho-acoustical effects in the human ear. Such an approach, however, is difficult both to formulate and to realize [GR98].

The simplest click detection methods involve a highpass filtering operation on the signal, the assumption being that most audio signals contain little information at high frequencies, while clicks, like impulses, have spectral content at all frequencies. Clicks are thus enhanced relative to the signal by the highpass filtering operation and can easily be detected by thresholding the filtered output. The method has the advantage of being simple to implement and having no unknown system parameters (except for a detection threshold). Of course, the method will fail if the audio signal itself has strong high frequency content or the clicks are band-limited. Along similar lines, wavelets and multi-resolution methods in general have useful localization properties for singularities in signals, and a Wavelet filter at a fine resolution can

be used for the detection of clicks. Other methods attempt to incorporate prior information about signal and noise into a model-based detection procedure.

7.3.1.3 Hiss Reduction

Random additive background noise is a form of degradation common to all VoIP systems. In the case of audio/speech signals the noise, which is generally perceived as *hiss* by the listener, will be composed of electrical circuit noise, irregularities in the storage medium and ambient noise from the recording environment. The combined effect of these sources will generally be treated as one single noise process. Random noise generally has significant components at all audio frequencies, and thus simple filtering and equalization procedures are inadequate for restoration purposes.

Noise reduction has been of great importance for many years in engineering disciplines. The classic least-squares work of Norbert Wiener [Wie49] placed noise reduction on a firm analytic footing, and still forms the basis of many noise reduction methods. In the field of speech processing a large number of techniques has been developed for noise reduction, and many of these are more generally applicable to noisy audio signals [LO79, Lim83].

In addition to the basic noise suppression rules (Wiener, power/spectral subtraction), various alternative rules have been proposed, based upon criteria such as maximum likelihood and minimum mean-squared error. A potentially important development in spectral domain noise reduction is the incorporation of the psycho-acoustical properties of human hearing. It is clearly sub-optimal to design algorithms based upon mathematical criteria which do not account for the properties of the listener. In [Can91] a method is derived which attempts to mimic the pre-filtering of the auditory system for hearing-impaired listeners, while in [LH97] simultaneous masking results of the human auditory system are employed to predict which parts of the spectrum need not be processed, hence leading to improved fidelity in the restored output as perceived by the listener. These methods are in their infancy and do not yet incorporate other aspects of the human auditory system such as non-simultaneous masking, but it might be expected that noise reducers and restoration systems of the future will take fuller account of these features to their benefit.

7.4 Mobility

For wireless telephony, clients such as mobile phones must be able to rapidly disassociate from one access point and connect to another. The delay that occurs during handoff cannot exceed about 50 ms, the interval that is detectable by the human ear. However, current roaming delays in 802.11 networks average in the hundreds of milliseconds [MT05]. This can lead to transmission *hiccups*, loss of connectivity and

degradation of voice quality. Faster handoffs are essential for 802.11-based voice to become widely deployed.

The 802.11r working group of the IEEE is drafting a protocol that will facilitate the deployment of IP-based telephony over 802.11-enabled phones. The 802.11r standard is designed to speed handoffs between access points or cells in a wireless LAN. The working group is drafting the final protocol, which should be approved toward the beginning of 2007[Gab06].

Another problem with current 802.11 wireless gear is that a mobile device cannot know if necessary QoS resources are available at a new access point until after a transition. Thus, it is not possible to know whether a transition will lead to satisfactory application performance. 802.11r refines the transition process of a mobile client as it moves between access points. The protocol allows a wireless client to establish a security and QoS state at a new access point before making a transition, which leads to minimal connectivity loss and application disruption. The overall changes to the protocol do not introduce any new security vulnerabilities. This preserves the behavior of current stations and access points.

When approved, 802.11r will govern the way roaming mobile clients communicate with candidate access points, establish security associations and reserve QoS resources. Under 802.11r, clients can use the current access point as a conduit to other access points, allowing clients to minimize disruptions caused by changing channels. There are tradeoffs among the speed of a handoff, the certainty of communication with an access point and disruption of current communications. A client can stay on its current channel and use its current access point to communicate with other candidate access points. This minimizes disruption to the client's data stream but does not allow the client to determine anything about its ability to communicate with other access points over the air. The client also can change to the channel of another access point. This allows the client to be certain of the quality of its communication with the other access point over the air but causes some disruption to communication with its current access point.

As usage of wireless networks increases, the density of access points may increase to satisfy increased capacity, which will lead to more frequent handoffs. Such environments would benefit from the capabilities provided by 802.11r.

7.5 Security

Due to inexpensive wireless establishment of WLANs, users but especially small to middle-sized businesses could be swayed to build up separated VoWLAN-networks. Therefore, first VoWLAN and also hybrid telephones, which support VoIP as well as UMTS or GSM are currently available. Consequently, VoWLAN and its specific security threats will be of utmost importance.

In order to classify these threats reasonably, an evaluation of security targets in VoIP-Systems is beneficial. Although these targets are orthogonally to each other, interplays regarding their accomplishment exist. Hence, the theft of an access password (loss of confidentiality) yields to loss of integrity of a system, if the attacker uses the password to get unauthorized access in order to change the configuration. Analogue, the loss of integrity regarding configuration information could lead to loss of confidentiality e.g. if an encryption module is deactivated or a more weak natured encryption scheme is activated.

Primary Security Targets In general communication systems three classical security targets are differentiated: confidentiality, integrity, and availability. Thereby confidentiality designates the protection of unauthorized abandonment of information, integrity the protection of unauthorized alteration of information, while availability the protection of unauthorized withholding of information.

Secondary Security Targets From the general primary security targets additional aims could be pointed out. For example, authenticity could be defined as integrity of message-content and message-origin while accountability could be defined as availability and integrity of identities (subjects) and their executed actions, if primary security targets are applied to specific meta information or system services as for example the sender's identity or general accounting information.

7.5.1 Security Threats

In general wireless networks are frail to any kind of attack - equally if passive or active, because the aggressor has uncomplicated access to the network compared to wired networks even exteriorly of physical boundaries of the provider, hence from a safe distance anonymous operation is comparatively simple. Therefore, today's WLAN components implement several security mechanisms, which prevents attacks largely. Unfortunately, many of these security mechanisms are insufficient and conceal extensive risks as far as one relies on them exclusively not considering security schemes in higher levels.

MAC ACL Access Points implement just a rudimentary access protection to the wireless network, as they solely handle clients whose MAC address is listed in so called MAC ACL (access control list). The security is based on the fact, that every terminal has its own unambiguous unalterable MAC address. But this assumption is wrong, as several available WLAN cards permit to change the MAC address with help by a manipulated device driver (MAC spoofing) [Won05]. Hence, the attacker is able to get into the network despite the fact of a MAC ACL. Consequently, by using a softphone on a laptop, (s)he could use a MAC-address of a VoIP-telephone

registered in the WLAN and for instance could telephone by debiting the victim's account. Furthermore, network services like for example the DHCP or ARP-answers could be manipulated in order to tamper the configuration of VoWLAN telephone [ea05].

WEP WEP is part of the IEEE 802.11 standard ratified in September 1999. WEP uses the stream cipher RC4 (Rivest Cipher) for confidentiality and the CRC-32 checksum for integrity. WEP relies on a secret key that is shared between a mobile station (e.g. a laptop with a wireless Ethernet card) and an access point (i.e. a base station). The secret key is used to encrypt packets before they are transmitted, and an integrity check is used to ensure that packets are not modified in transit. The standard does not discuss how the shared key is established. In practice, most installations use a single key that is shared between all mobile stations and access points.

A stream cipher operates by expanding a short key into an infinite pseudo-random key stream. The sender XORs the key stream with the plaintext to produce ciphertext. The receiver has a copy of the same key, and uses it to generate identical key stream. XORing the key stream with the ciphertext yields the original plaintext. This mode of operation makes stream ciphers vulnerable to several attacks. If an attacker flips a bit in the ciphertext, then upon decryption, the corresponding bit in the plaintext will be flipped. Also, if an eavesdropper intercepts two ciphertexts encrypted with the same key stream, it is possible to obtain the XOR of the two plaintexts. Knowledge of this XOR can enable statistical attacks to recover the plaintexts. The statistical attacks become increasingly practical as more ciphertexts that use the same key stream are known. Once one of the plaintexts becomes known, it is trivial to recover all of the others.

WEP has defenses against both of these attacks. To ensure that a packet has not been modified in transit, it uses an Integrity Check (IC) field in the packet. To avoid encrypting two ciphertexts with the same key stream, an Initialization Vector (IV) is used to augment the shared secret key and produce a different RC4 key for each packet. The IV is also included in the packet. However, both of these measures are implemented incorrectly, resulting in poor security.

The integrity check field is implemented as a CRC-32 checksum, which is part of the encrypted payload of the packet. However, CRC-32 is linear, which means that it is possible to compute the bit difference of two CRCs based on the bit difference of the messages over which they are taken. In other words, flipping bit n in the message results in a deterministic set of bits in the CRC that must be flipped to produce a correct checksum on the modified message. Because flipping bits carries through after an RC4 decryption, this allows the attacker to flip arbitrary bits in an encrypted message and correctly adjust the checksum so that the resulting message

appears valid.

The initialization vector in WEP is a 24-bit field, which is sent in the cleartext part of a message. Such a small space of initialization vectors guarantees the reuse of the same key stream. A busy access point, which constantly sends 1500 byte packets at 11 Mbps, will exhaust the space of IVs after $\frac{1500 \cdot 8 \cdot 2^{24}}{11 \cdot 10^6} \approx 18302.42$ seconds, or 5 hours. (The amount of time may be even smaller, since many packets are smaller than 1500 bytes.) This allows an attacker to collect two ciphertexts that are encrypted with the same key stream and perform statistical attacks to recover the plaintext. Worse, when the same key is used by all mobile stations, there are even more chances of IV collision. For example, a common wireless card from Lucent resets the IV to 0 each time a card is initialized, and increments the IV by 1 with each packet. This means that two cards inserted at roughly the same time will provide an abundance of IV collisions for an attacker. Worse still, the 802.11 standard specifies that changing the IV with each packet is optional!

Denial-of-Service By the following DoS attacks the availability of the whole wireless LAN or single stations could be jammed. In case of VoIP, attackers could prevent signaling of incoming and or outgoing calls. Likewise, running calls could be interrupted and canceled. Mostly, these attacks are also successful against WPA-protected WLANs [GN04]. Additionally, in case of 802.1X EAP-specific DoS-attacks, e.g. fake EAPoL-failure messages or flooding with EAPoL-start messages, the availability of WLANs could be massively affected [ea05].

The obvious DoS-attack is the usage of a jammer (type of transmitter used to jam radio waves) in the corresponding frequency range of 2.4 GHz in case of 802.11b/g and 5 GHz with 802.11a/h.

By far more cunning are DoS-attacks, which exploit protocol weaknesses. They are similarly effective as a jammer, but considerably more to the point applicable and furthermore more difficult to detect. Moreover, they could be performed with standard WLAN-hardware and freely available software, because manipulated device drivers could get direct access to protocols of the network access layer.

Objectives are for example the central WLAN access point or single terminals or connections which could be comprised to the point. For more detailed description of DoS-attacks see e.g. [RHA04]

Eavesdropping Network security attacks are typically divided into passive and active attacks. These two broad classes are then subdivided into other types of attacks as depicted in figure 7.2.

A passive attack is defined as an attack in which an unauthorized party gains access to an asset and does not modify its content [KO02]. Passive attacks can be either eavesdropping or traffic analysis (sometimes called traffic flow analysis).

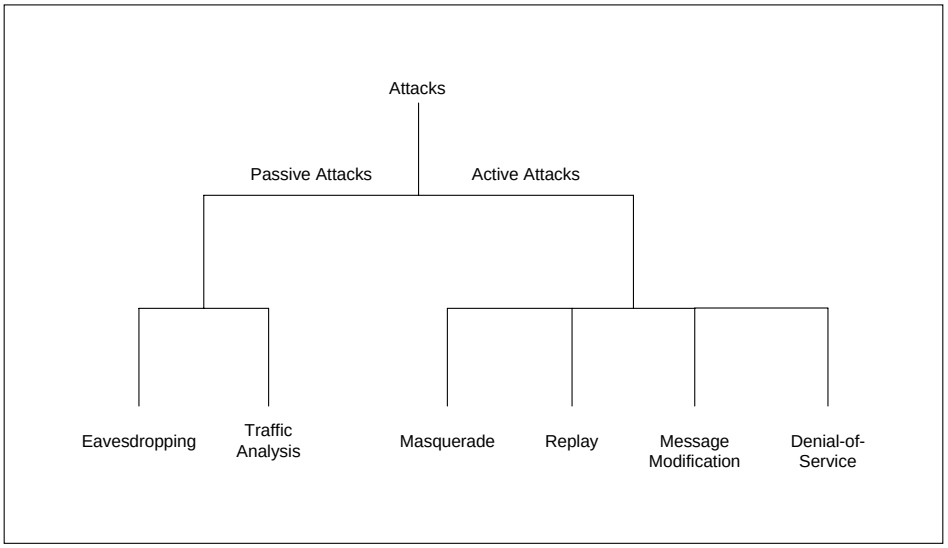


Figure 7.2: Taxonomy of Security Attacks
(following [KO02])

With eavesdropping the attacker monitors transmissions for message content. An example of this attack is a person listening into the transmissions on a LAN between two workstations or tuning into transmissions between wireless devices. With traffic analysis the attacker, in a more subtle way, gains intelligence by monitoring the transmissions for patterns of communication. A considerable amount of information is contained in the flow of messages between communicating parties.

In the following, eavesdropping is used in terms of unauthorized interception of VoWLAN communication.

Confidentiality is the property with which information is not made available or disclosed to unauthorized individuals, entities, or processes. Due to the broadcast and radio nature of wireless technology, confidentiality is a more difficult security requirement to meet in a wireless network. Adversaries do not have to tap into a network cable to access network resources. Moreover, it may not be possible to control the distance over which the transmission occurs. This makes traditional physical security countermeasures less effective.

Passive eavesdropping of native 802.11 wireless communications may cause significant risk. An adversary may be able to listen in and obtain sensitive information including proprietary information, network IDs and passwords, and configuration data. This risk is present because the 802.11 signals may travel outside the building perimeter. Because of the extended range of 802.11 broadcasts, adversaries can potentially detect transmission from a parking lot or nearby roads. This kind of

attack, performed through the use of a wireless network analyzer tool or *sniffer*, is particularly easy for two reasons:

- frequently confidentiality features of WLAN technology are not even enabled
- because of the numerous vulnerabilities in the 802.11 technology security determined adversaries can compromise the system

Wireless packet analyzers, such as AirSnort[1] and WEPcrack[2], are tools that are readily available on the Internet today. AirSnort is one of the first tools created to automate the process of analyzing networks. Unfortunately, it is also commonly used for breaking into wireless networks. AirSnort can take advantage of flaws in the key-scheduling algorithm that was provided for implementation of RC4, which forms part of the original WEP standard. To accomplish this, AirSnort requires only a computer running the Linux operating system and a wireless network card. The software passively monitors the WLAN data transmissions and computes the encryption keys after at least 100 MB of network packets have been sniffed. On a highly saturated network, collecting this amount of data may only take three or four hours; if traffic volume is low, it may take a few days.

Another risk to loss of confidentiality through simple eavesdropping is broadcast monitoring. An attacker can monitor traffic, using a laptop in promiscuous mode, when an access point is connected to a hub instead of a switch. Hubs generally broadcast all network traffic to all connected devices, which leaves the traffic vulnerable to unauthorized monitoring. Switches, on the other hand, can be configured to prohibit certain attached devices from intercepting broadcast traffic from other specified devices. For example, if a wireless access point was connected to an Ethernet hub, a wireless device that is monitoring broadcast traffic could intercept data intended for wired and wireless clients. Consequently, one should consider using switches instead of hubs for connections to wireless access points.

7.5.2 Security Enhancements

As discussed earlier, security in case of wireless speech transmissions might be also considered in the application layer. Integrity and confidentiality, two of the primary security targets, are taken into account by encryption of the payload or hence regarding the multiple description coding approach by encoding parts of it.

The developed WVCS in this work is based on a MDC approach as detailed in chapter 8, hence the speech data is coded in multiple slightly interdependent streams. These streams have to be encrypted in order to provide secure communication in aspect of electronic eavesdropping. As detailed later, the multiple data

[1] `http://airsnort.shmoo.com/`
[2] `http://wepcrack.sourceforge.net/`

streams consist of an enhancement layer build of compressed wavelet-coefficients which are responsible for the higher frequency components an further data streams which describe the lower frequency components of the speech signal by specific information obtained by a CELP resembling procedure.

Generally, two encryption approaches are distinguished. Block- and stream ciphers.

Block ciphers are the most basic, as they use a single key to transform a block of text with a specific length. For example, in the Blowfish [Sch94] system, blocks are always 64 bits. A longer message is encoded by breaking the message up into many blocks and applying the cipher to each block. This stands in contrast to stream ciphers.

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry [MvOV96]. They are also more appropriate when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable. But as no submitted stream cipher was accepted by the NESSIE (New European Schemes for Signatures, Integrity and Encryption) project [Pre03] - none of the six algorithms met the rather stringent security requirements put forward by NESSIE - no recommendation could be given in this case. Most of the proposed schemes are quite secure, but successful attacks are known. So for private communication methods like for example BMGL, SEAL, SCREAM, Leviathan, LILI-128, SNOW or SOBER-t16/-t32 should be sufficient.

Therefore the choice went for block ciphers. The drawback with these algorithms is that they are based on block sizes of 64 or 128 bit, hence, smaller sized payload data is consequently distended. As restructuring of the encoded speech information in order to follow these specific block-sizes is impossible for the reason of MDC the introduction of block ciphers would emphasize additional bandwidth demands as shown in table 7.2 and 7.3.

The overhead (68%/28%) introduced with block size 128 bit is extremely high, as with block size 64 bit the overhead is quite acceptable (5%/12%) - when thinking of improved safety. Two important block ciphers are Rijndael (AES) [DR02, FIP01] and Blowfish [Sch94], which are based on a block size of 128 bit and 64 bit respectively.

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then

frame size (20ms)			
	sub stream [bits/frame]	# blocks [64 bit/block]	overhead [bit]
	48	1	16
	64	1	0
	192	3	0
	304	5	16
total overhead		5.263%	

	sub stream [bits/frame]	# blocks [128 bit/block]	overhead [bit]
	48	1	80
	64	1	64
	192	2	64
	304	4	208
total overhead		68.421%	

Table 7.2: Introduced overhead with 20 ms frame size

frame size (30ms)			
	sub stream [bits/frame]	# blocks [64 bit/block]	overhead [bit]
	64	1	0
	96	2	32
	240	4	16
	400	7	48
total overhead		12.0%	

	sub stream [bits/frame]	# blocks [128 bit/block]	overhead [bit]
	64	1	64
	96	1	32
	240	2	16
	400	4	112
total overhead		28.0%	

Table 7.3: Introduced overhead with 30 ms frame size

(key,block) length	AES CD (ANSI C)		Brian Gladman (Visual C++)	
	speed [Mbps]	# cycles/block	speed [Mbps]	# cycles/block
(128,128)	27.0	950	70.5	363
(192,128)	22.8	1125	59.3	432
(256,128)	19.8	1295	51.2	500

Table 7.4: AES - cipher (and inverse) performance

it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. Blowfish is unpatented and license-free, and is available free for all uses. Many cryptographers have examined Blowfish, although there are few published results. Serge Vaudenay examined weak keys in Blowfish [Vau06]; there is a class of keys that can be detected (although not broken)in Blowfish variants of 14 rounds or less. Vincent Rijmen's Ph.D. thesis includes a second-order differential attack on 4-round Blowfish that cannot be extended to more rounds [Sch06]. Blowfish is one of the fastest block ciphers in widespread use [Vau06], except when changing keys. Each new key requires pre-processing equivalent to encrypting about 4 kilobytes of text, which is very slow compared to other block ciphers. Performance analysis in [ea03] showed that Blowfish is just a little bit slower than AES - (AMD Athlon 700 MHz 512KB cache PC platform; AES 14.02 Mbps, Blowfish 14.53 Mbps). Therefore, table 7.4 could be considered also for Blowfish. Nevertheless, Blowfish is used in this work for encryption as it is slightly faster and it is a 64 bit block cipher introducing less overhead than AES.

The Rijndael Block Cipher is used in AES. Rijndael is an iterated block cipher with a variable block length and a variable key length. The block length and the key length can be independently specified to 128, 192 or 256 bits, moreover Rijndael cipher is suited to be implemented efficiently on a wide range of processors and in dedicated hardware. Table 7.4 gives the figures for the raw encryption, when implemented in C, without counting the overhead caused by the AES API. Speed estimates were originally generated by compiling the code with EGCS (release 1.0.2) and executing it on a 200 MHz Pentium running Linux [DR02].

While trying to enhance the security of the developed WVCS, another approach was considered but it exposed to be unsuccessful. Reducing the extra computational time was one of the main goals while raising walls against eavesdroppers. Therefore, the enhancement stream shouldn't be encrypted but enciphered by parameterized wavelets which were discussed e.g. in [Zib02, Pol89, Sel97].

As the enhancement data is obtained by a wavelet-based compression scheme, the idea was to use parameterized wavelets in order to prohibit for proper decoding, if the exact wavelet is not known by the attacker. But as the differences between the parameterized wavelets are too small, this approach was unsuccessful as figure 7.3 shows.

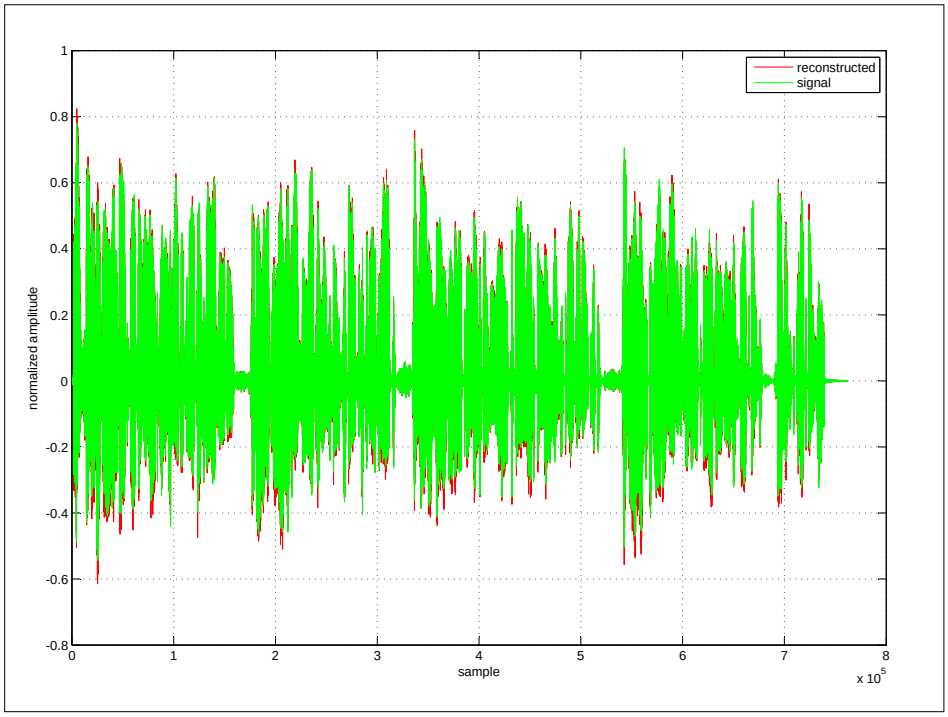


Figure 7.3: Pollen-based (inverse) transformation for different ϕ

A parameterization scheme proposed by Pollen [Pol89] was used. The signal was transformed with $\phi = 0.3$ and inverse transformed without any compression with $\phi = 0.5$. Even schemes with two or more parameters just led to a disturbed and noticeable degradation but the speech is still understandable, hence the anticipated scrambling is too low.

If the degradation would have been that great, that the speech information couldn't be retrieved, the idea was to encode the parameter ϕ within the encrypted data stream. However, if these information couldn't be decrypted properly also the enhancement information couldn't be decrypted as ϕ is unknown, hence, without the proper keys the speech signal should be safe against eavesdroppers.

7.6 Summary

Different strategies to enhance robustness and security of the developed WVCS were presented. Multiple description coding was proposed to enhance communication robustness even with high BERs which are common to wireless links. In order to enhance security, i.e. safety against eavesdroppers, block- and stream ciphers were

discussed.

Modern block ciphers are still unbroken (AES, Blowfish, etc.) and therefore they are suitable to provide secure voice communication. But, with the introduction of block ciphers extra bandwidth must be taken into account. This drawback haven't to be considered if stream ciphers would be used. However, the drawback with stream ciphers is, that no security-proved scheme exist. But, in order to break these kind of ciphers, big efforts have to be made. Finally, the to be used encryption technology depends on the desired security level and the accepted extra bandwidth introduced in case of block ciphers.

The next chapter introduces the whole WVCS and its performance for various conditions.

8 Simulation and Performance Analysis

8.1 Introduction

The final chapter consists of two main sections. The first section deals with the concepts of the fine-grain and channel-adaptive multidescriptive hybrid speech coding scheme, while the second main section deals with the investigation of the whole processing system in different scenarios introducing random error loss or burst error loss.

8.2 Components Setup

In figure 8.1 the complete MDC-based speech coder is depicted. The decoder works just the other way around. In this section each part of the coder is described and if necessary, relationships with the decoder are discussed.

8.2.1 Speech Input and Preprocessing

In order to fit to the specific needs of the developed speech codec, some preprocessing steps have to be accomplished first. This speech codec could be classified as a wideband speech codec with additional features like for example channel adaptivity and fine-grain bit rate scalability. Therefore, the considered frequency range is 0 kHz to 8 kHz. Due to Nyquist's theorem the sampling rate must be at least 16 kHz in order to fully reconstruct the signal. Additionally, the input's quantization resolution must be 16 bits per sample resulting in an uncompressed bit rate of 256 Kbps. Fulfilling final bit rates are between 9 Kbps and 32 Kbps, hence compression ratios of about 32:1 and 8:1. Compared to other wideband speech codecs, these compression ratios are competitive.

Presupposed an input signal sampled at 16 kHz with 16 bits per sample the first process that is applied to the sequence, to be more precise – one speech frame with length of 20 ms, is a psychoacoustic analyzer which classifies which frequency ranges are important and must therefore be encoded with more bits than frequency ranges that are not important in this frame. Details about that process will be discussed in the section about the wavelet packet compressor.

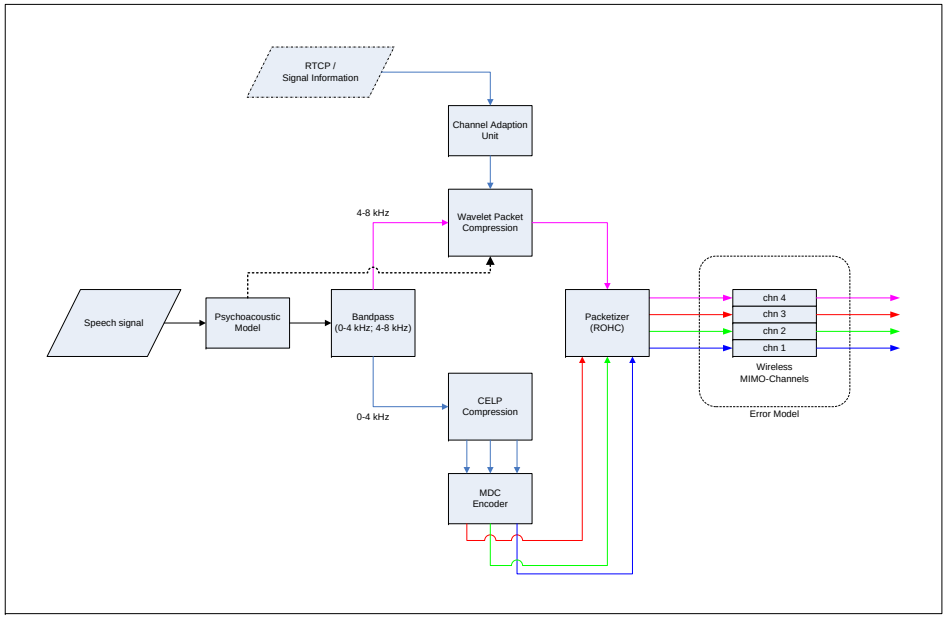


Figure 8.1: MDC Speech Codec

The second process that is applied to the sequence is the split into two subbands. The subband with range from 0 kHz to 4 kHz is send to a CELP-based compressor while the other subband with range from 4-8 kHz is send to a Wavelet-Packet-based compressor.

8.2.2 CELP Compression

The speech signal was split into two subbands. The subband with the lower frequency range (0-4 kHz) is handled by a CELP-based compressor additionally encoded multidescriptive in order to enhance robustness against transmission errors. CELP-based compressors deliver different parameters and coefficients that could be classified regarding their importance for reconstruction. These are for example line spectral frequencies, pitch details, different codebook information, gains, scale factors etc. Two different approaches were investigated:

- Modified G.729 for 20 ms frame size
- Modified iLBC for 20 ms frame size

The decision for a frame size of 20 ms was elucidated while discussing security issues of speech coding schemes in the previous chapter.

	Odd Frame	Even Frame	Sum
Frame Indicator	2(00)	2(01)	4
LSP stage 1	16	16	
LSP stage 2	<10,0>	<0,10>	52
	subframe <1/2>	subframe <1/2>	
Pitch details	<18/10>	<0/0>	28
Fixed Codebook	<26/0>	<26/0>	52
Fixed Codebook Signs	<8/0>	<8/0>	16
Gain Codebook	<14/0>	<14/0>	28
Total			180

Table 8.1: Bit allocation for description I of the modified G.729

	Odd Frame	Even Frame	Sum
Frame Indicator	2(10)	2(11)	4
LSP stage 1	16	16	
LSP stage 2	<0,10>	<10,0>	52
	subframe <1/2>	subframe <1/2>	
Pitch details	<0/0>	<18/10>	28
Fixed Codebook	<0/26>	<0/26>	52
Fixed Codebook Signs	<0/8>	<0/8>	16
Gain Codebook	<0/14>	<0/14>	28
Total			180

Table 8.2: Bit allocation for description II of the modified G.729

8.2.2.1 Modified G.729

A new multiple description coder based on the G.729 [Sta96] speech codec was designed. The MD coder creates two balanced descriptions, i.e. each description is of the same rate, and speech decoded from either description is of similar quality. The idea behind the coder is to take an SD coder (G.729) and split the bitstream into two sub-streams. This is similar to the no-excess joint rate case of MD coding, where the individual descriptions can be combined to give an optimal joint description. Since dividing the bitstream into two non-overlapping portions cannot give acceptable quality at the side decoders, some redundancy by replicating vital information in both the description is injected.

The ITU-T G.729 is a CS-ACELP based codec for encoding narrowband speech at the rate of 8 Kbps. The G.729 codec encodes 10 ms speech frames using 80 bits at a resultant bit rate of 8 Kbps. The encoder of the MD codec divides the G.729 bitstream into two overlapping bitstreams. Table 8.1 and 8.2 show the bit allocation for odd and even frames in each of the descriptions.

To keep the effective average bit rate of each description the same, odd and

even numbered frames in each description are coded with a different number of bits. This is achieved by including the bits corresponding to the pitch delay only in alternate frames. The pitch delay for the second subframe in each frame is differentially encoded with respect to the first subframe. Without the first subframe pitch delay, the second subframe pitch delay cannot be decoded. Therefore, pitch delay information for both subframes has to be always included together in one description.

When both descriptions are received at the decoder the two descriptions are combined to give the bitstream of G.729 modified for 20 ms frame size. If both descriptions are lost, then the frame error concealment algorithm of G.729 is used to conceal the lost frame. If one of the descriptions is received, then the decoder substitutes the missing information by using received parameters in the description or information from the most recently correctly received frame. When only one of the descriptions is received, the LSP vectors are constructed from the received first stage vector and one of the received subvectors. The missing second stage subvector is assumed to be zero. The pitch delay in even (odd) frame in first (second) description is constructed from the previous received frame's pitch delay increased by 1. This process is the same as that used for frame error concealment in the G.729 codec [Sta96]. Finally, the modified G.729 codec encodes 20 ms speech frames using 180 bits at a resultant bit rate of 9 Kbps.

8.2.2.2 iLBC

The second approach is based on a new multiple description coder based on the iLBC [ADA+04] speech codec. The MD coder creates three descriptions according to their bit error or loss sensitivity with extra redundancy of the most sensitive bits (class 1), which are also encoded in successive frames into classes 2 and 3 - the less and least sensitive bits of each frame. A brief version of the 20 ms standard bit allocation scheme is shown in table 8.3.

The standard distribution enables the use of uneven/unbalanced multiple description coding. In order to improve robustness different redundancy schemes are introduced. In case of the 20 ms version of iLBC the effective bit rate is 15.2 Kbps. The following table 8.4 shows investigated robustness levels based on introducing redundancy.

To increase robustness the most sensitive bits are additionally encoded. Intermediate stages encode either just the LSF of class 1 information (c1.lsf) or in the next higher level the full class 1 (c1) information. This approach yields to final bit rates from 15.2 Kbps up to 20.0 Kbps for highest possible robustness.

Two different MDC schemes were introduced. The first is a balanced coding approach while the latter is an unbalanced coder. The balanced coder should be used in environments, where channels couldn't be prioritized, e.g. protected by QoS

	Class 1	Class 2	Class 3
LSF	20	0	0
Block Class	2	0	0
Position sample segment	1	0	0
Scale factor state coder	6	0	0
Quantized residual state samples	0	57	114
Codebook for sample blocks	6	0	15
Gain for sample blocks	3	1	8
Indices for CB sub-blocks	7	0	39
Gains for sub-blocks	3	6	15
Empty frame indicator	0	0	1
Sum	48	64	192
Total		304	

Table 8.3: Bit allocation for iLBC

Robustness Level	description <1,2,3>	Redundancy [bits]	Bit rate [Kbps]
0	<c1,c2,c3>	0	15.2
1	<c1,c1.lsf+c2,c3>	20	16.2
2	<c1,c1+c2,c3>	48	17.6
3	<c1,c1+c2,c1.lsf+c3>	68	18.6
4	<c1,c1+c2,c1+c3>	96	20.0

Table 8.4: iLBC-based robustness levels

mechanisms. As every channel has the same bandwidth, error probability etc. the balanced version should be preferred.

As in whatever way single channels could be promoted regarding different characteristics like for example delay, jitter, bandwidth or applied security mechanisms the unbalanced version should be preferred as the unleveled environment better fit to the unbalanced scheme. Hence, the most sensitive information could be send over the *promoted* channel while the less/least information then could be send over *normal* channels.

8.2.3 Wavelet Packet Compression

The speech signal was split into two subbands. The subband with the higher frequency range (4-8 kHz) is handled by a Wavelet-based compressor with an applied psychoacoustic model. If we regard the wavelet transform as a filter bank, then we can consider wavelet transforming a signal as passing the signal through this filter bank. The outputs of the different filter stages are the wavelet- and scaling function transform coefficients. Analyzing a signal by passing it through a filter bank is not a new idea and has been around for many years under the name subband coding. It is used for instance in computer vision applications. The (classical) dyadic wavelet

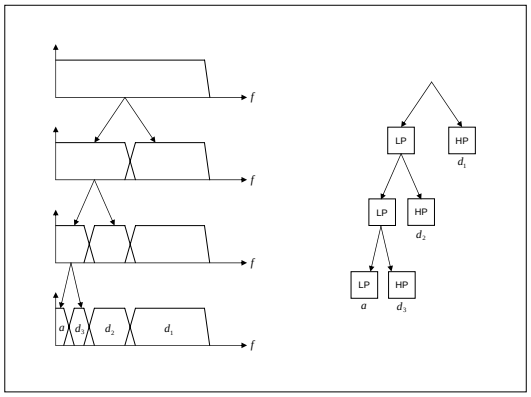


Figure 8.2: A dyadic filter tree for a level-3 DWT

transform is depicted in figure 8.2.

After the repeated spectrum splitting a series of bandpass bands with doubling bandwidth and one lowpass band is left. In order to apply a psychoacoustic model a more fine-grain frequency resolution especially for specific ranges is needed as detailed later in this section. Therefore the dyadic approach, where just the lowpass coefficients are repeatedly handled while the highpass coefficients were left untouched is unuseful for the purpose of psychoacoustic enhanced wavelet compression.

To overcome this disadvantage, the wavelet packet transformation (WPT) with frequency ordered coefficients is used. With the WPT it is possible to control the frequency resolution for specific ranges. This is exactly what is required in order to fit to psychoacoustic modeling. As seen later in this section, the quantization error by reducing the bit resolution for specific frequency ranges could be increased inaudible to improve the overall compression ratio. This is quite similar to current high quality audio compression codecs like for example MP3, MPC, OGG or ACC.

The balanced WPT decomposes the lowpass and highpass coefficients equally. This results in equidistant frequency ranges with fine resolution at higher decomposition levels as depicted in figure 8.3.

Due to psychoacoustic modeling the balanced version of the WPT is not optimal. Therefore an unbalanced version which fits the Bark scale was developed. The Bark scale is a psychoacoustical scale, that ranges from 1 to 24 and corresponds to the first 24 critical bands of hearing. The subsequent band edges are (in Hz) 0, 100, 200, 300, 400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500. Let $\{\psi_n(t) : n \in \mathbb{Z}_+\}$ denote a wavelet packet family, and let $E \subset \{(l,n) : 0 \leq l < L, 0 \leq n < 2^l\}$ represent the terminal nodes of a WPD tree. Then disjoint covers of $[0,1)$ by dyadic intervals of the form

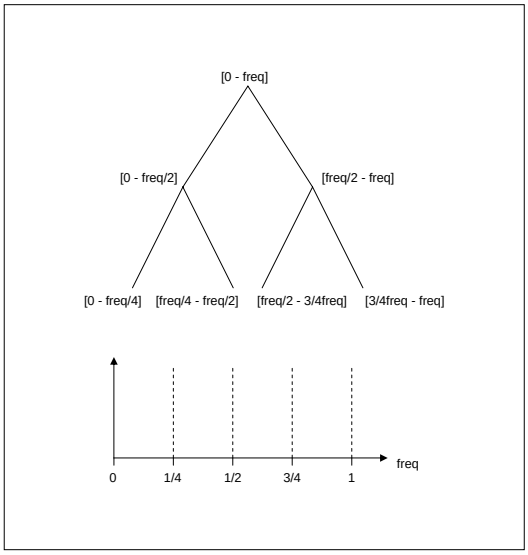


Figure 8.3: Balanced wavelet packet transform for a 2-level decomposition

$I_{l,n} = \left[2^{-l}n, 2^{-l}(n+1)\right)$ correspond to specific wavelet packet expansions (specific sets of terminal nodes E). In particular, $\{\psi_{l,n,k} : (l,n) \in E, k \in \mathbb{Z}\}$ where $\psi_{l,n,k}(t) \triangleq 2^{-l/2}\psi_n(2^{-l}t - k)$, form a basis for the signal space $\overline{span}\,\{\psi_0(t-k) : k \in \mathbb{Z}\}$. A terminal node $(l,n) \in E$ is associated with a subband whose center frequency and bandwidth are roughly given by [Wic00]:

$$f_{l,n} = 2^{-l}\left[GC^{-1}(n) + 0.5\right] \cdot F_s/2$$

$$\Delta_{l,n} = 2^{-l} \cdot F_s/2$$

where GC^{-1} is the inverse Gray code permutation and F_s is the sampling frequency in the signal space. To obtain critical bands with the WPD, a decomposition tree such that the distance between the center frequency of one subband to the center frequency of the next subband is 1 Bark. The relation between frequency f in Hertz and critical band rate z in Bark is approximately given by [Tra90] and depicted in figure 8.4:

$$z = \frac{26.81}{1 + \frac{1960}{f}} - 0.53.$$

Table 8.5 shows an approximation of the Bark scale by critical-band WPD (CB-WPD). Different nodes are grouped together in order to fit the respective Bark ranges. The corresponding decomposition tree is depicted in figure 8.5. The root

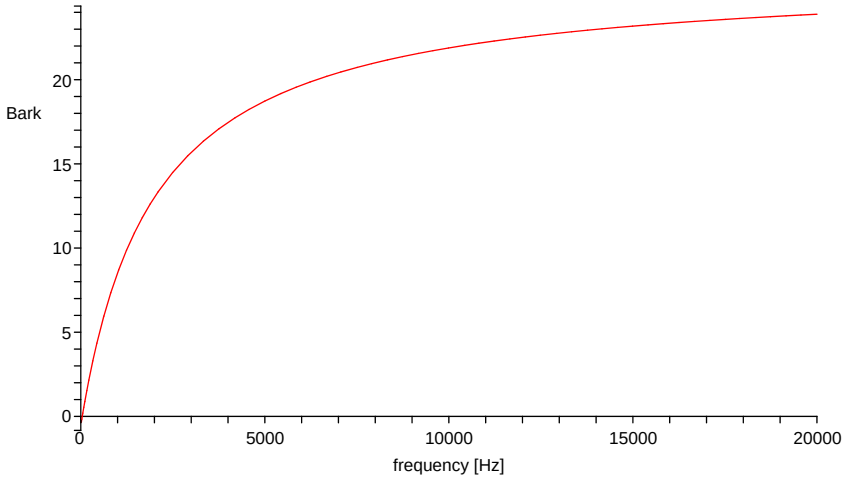


Figure 8.4: Relation between frequency in Hertz and critical band rate in Bark

of the tree $(l, n) = (0, 0)$ refers to the signal space. Each internal node in the tree $(l, n) \notin E$ is split into children-nodes $(l + 1, 2n)$ and $(l + 1, 2n + 1)$. The left and right branches, connecting a given node to its children-nodes, denote respectively lowpass and highpass wavelet filtering followed by a 2:1 downsampling.

The CB-WPD splits the frequency range 4 to 8 kHz into 15 subbands which are mapped to 5 disjunctive Bark scales. For the application of speech enhancement, it is useful to increase the number of subbands, hence refine the frequency resolution, and to allow some degree of redundancy (oversampling).

The psychoacoustic model provides for high quality lossy signal compression by describing which parts of a given digital audio signal can be removed (or aggressively compressed) safely – that is, without significant losses in the quality of the sound. Psychoacoustic is based heavily on human anatomy, especially the ear's limitations in perceiving sound as outlined previously. To summarize, these limitations are:

- Absolute threshold of hearing
- Absolute threshold of pain
- Temporal masking
- Simultaneous masking
- High frequency limit

Node	Node index	Start [Hz]	Center [Hz]	End [Hz]	Bark
(2,0)	3	0	1000	2000	out of range
(3,3)	10	2000	2500	3000	
(4,5)	20	3000	3250	3500	
(5,9)	40	3500	3625	3750	
(5,8)	39	3750	3875	4000	17
(5,24)	55	4000	4125	4250	
(6,51)	114	4250	4312.5	4375	
(6,50)	113	4375	4437.5	4500	18
(4,13)	28	4500	4750	5000	
(5,30)	61	5000	5125	5250	
(5,31)	62	5250	5375	5500	19
(4,14)	29	5500	5750	6000	
(5,20)	51	6000	6125	6250	
(6,43)	106	6250	6312.5	6375	
(6,42)	105	6375	6437.5	6500	20
(4,11)	26	6500	6750	7000	
(4,9)	24	7000	7250	7500	
(5,17)	48	7500	7625	7750	
(5,16)	47	7750	7875	8000	21

Table 8.5: Approximation of the Bark scale by critical-band WPD

Given that the ear will not be at peak perceptive capacity when dealing with these limitations, a compression algorithm can assign a lower priority to sounds outside the range of human hearing. By carefully shifting bits away from the unimportant components and toward the important ones, the algorithm ensures that the sounds a listener can hear most clearly are of the highest possible quality.

To determine which components are important and which are not, the following steps have to be calculated in order to finally receive the signal-to-mask ratio (SMR).

- Time-frequency conversion (FFT)
- Determine the sound pressure level
- Find tonal (sine like) and non-tonal (noise-like) components
- Eliminate all irrelevant maskers
- Compute individual masking thresholds
- Compute the global masking threshold
- Determine the minimum mask threshold
- Compute the signal-to-mask ratio (SMR)

These steps are visualized in figure 8.6. The psychoacoustic model analyzes the audio signal and computes the amount of noise masking available as a function of

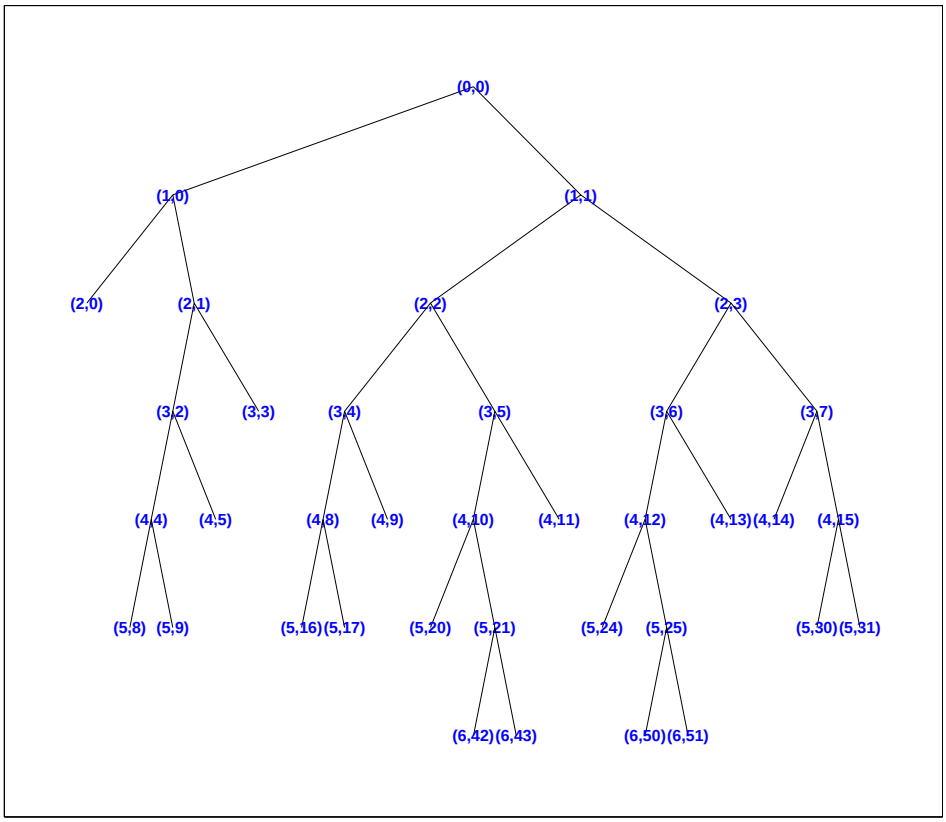


Figure 8.5: Bark scale approximation by critical-band WPD

frequency. The masking ability of a given signal component depends on its frequency position and its loudness. The encoder uses this information to decide how best to represent the input audio signal with its limited number of code bits. There is one psychoacoustic evaluation per frame. The psychoacoustic model uses a separate, independent, time-to-frequency mapping instead of the wavelet packet transform, because it needs finer frequency resolution for an accurate calculation of the masking thresholds. The psychoacoustic model use a Fourier transform for this mapping. A standard Hann weighting, applied to the audio data before Fourier transformation, conditions the data to reduce the edge effects of the transform window. The model uses a 512-sample analysis window. Because there are only 320 samples in a frame, a 512-sample window provides adequate coverage.

The model identifies and separates the tonal and noise-like components of the audio signal because the masking abilities of the two types of signal differ. The model identifies tonal components based on the local peaks of the audio power

spectrum. After processing the tonal components, the remaining spectral values are summed into a single nontonal component per critical band. The frequency index of each concentrated non-tonal component is the value closest to the geometric mean of the enclosing critical band.

The masking ability of a given signal spreads across its surrounding critical band. The model determines the noise-masking thresholds by first applying an empirically determined masking function to the signal components. The model includes an empirically determined absolute masking threshold, the threshold in quiet. This threshold is the lower bound on the audibility of sound. The model selects the minimum masking threshold within each subband. While this approach works well for the lower frequency subbands where the subband is narrow relative to a critical band, it might be inaccurate for the higher frequency subbands because critical bands for that frequency range span several subbands. These inaccuracies arise because the model concentrates all the non-tonal components within each critical band into a single value at a single frequency. In effect, the model converts non-tonal components into a form of tonal component. A subband within a wide critical band but far from the concentrated non-tonal component will not get an accurate non-tonal masking assessment. This approach is a compromise to reduce the computational loads.

The psychoacoustic model computes the signal-to-mask ratio as the ratio of the signal energy within the subband to the minimum masking threshold for that subband. This value is then passed to the bit- (or noise-) allocation section of the encoder.

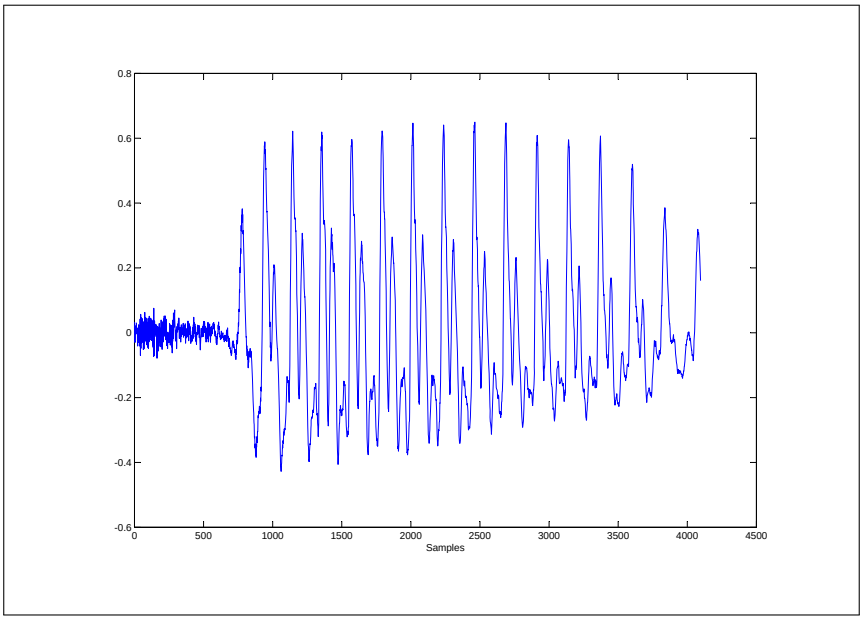
This approach is similar to the MPEG Layer I psychoacoustic model [Pan95].

Figure 8.6 shows the SMR for the calculated example. Subband number 5 to 11 represent the respective Bark scales from 17 to 21. In detail, the following mapping could be derived:

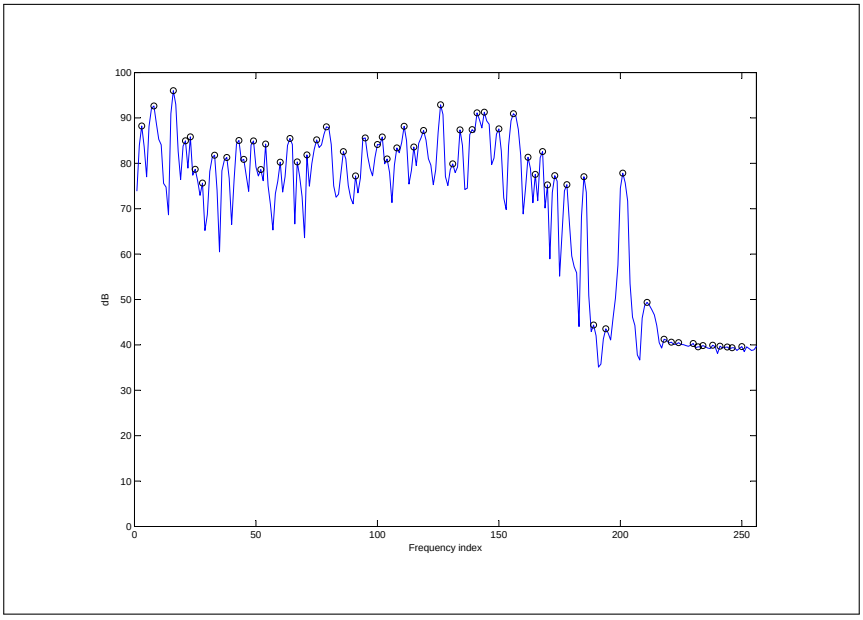
- subband $5 \longrightarrow 17$
- subband $6, 7 \longrightarrow 18$
- subband $8 \longrightarrow 19$
- subband $9, 10 \longrightarrow 20$
- subband $11 \longrightarrow 21$

With this information the dynamic quantization process could be initiated and respective wavelet packet coefficients could be efficiently encoded using the *waterfilling* algorithm.

Figure 8.7 depicts the situation for bit allocation. On the left is shown the situation before waterfilling. We have P bits to pour into the oddly shaped container.

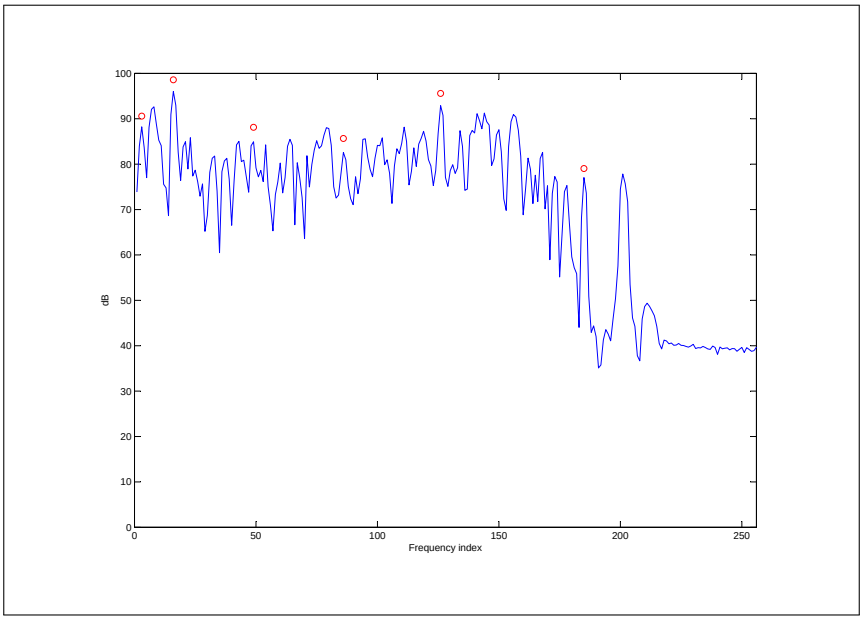


(a) Signal

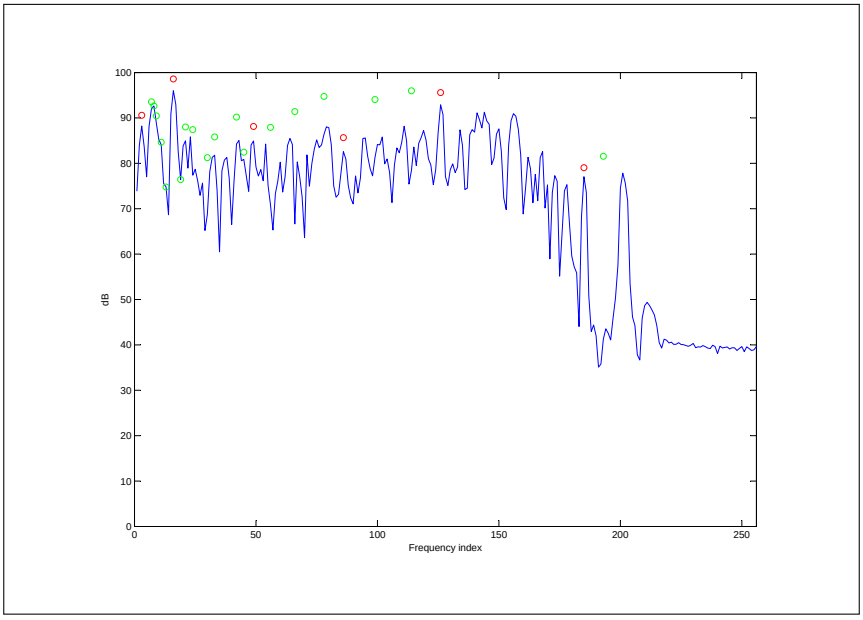


(b) Local maxima

Figure 8.6: Computing the signal-to-mask ratio (SMR)

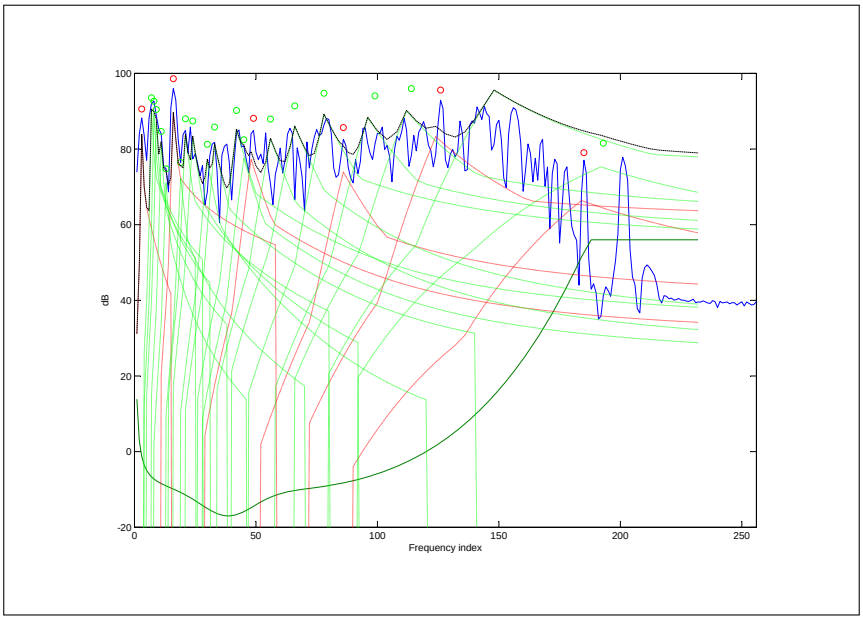


(c) Tonal Components

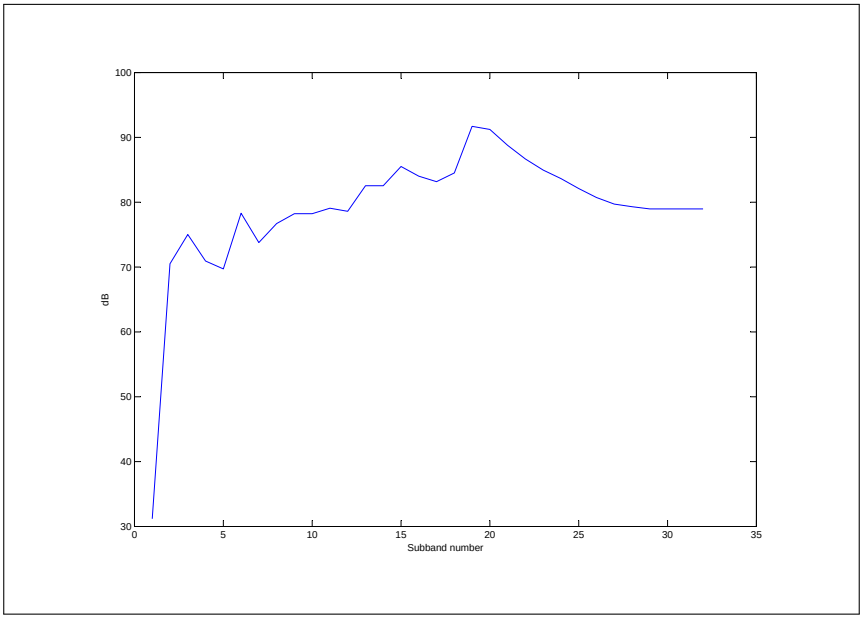


(d) Non-tonal Components

Figure 8.6: Computing the signal-to-mask ratio (SMR) (cont.)

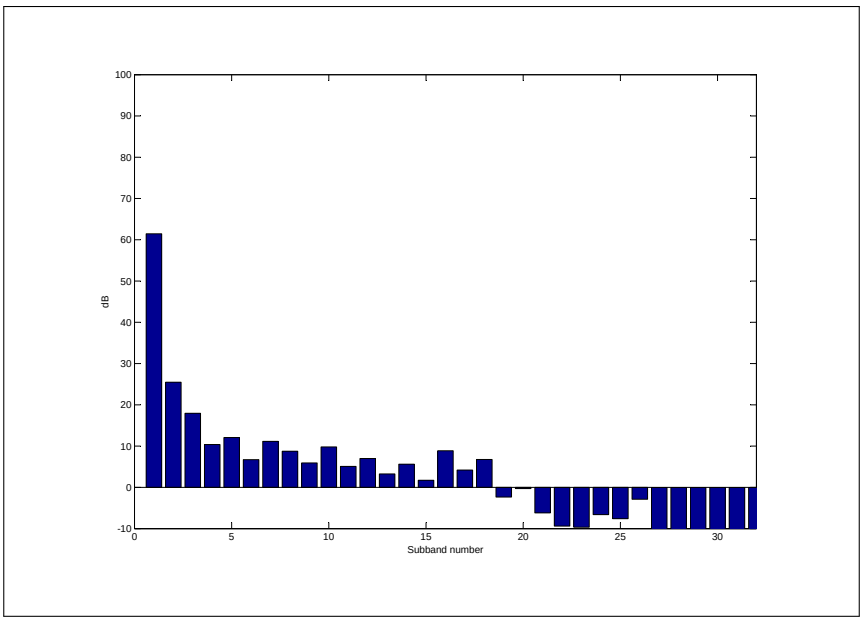


(e) Masking



(f) Minimum Masking Threshold

Figure 8.6: Computing the signal-to-mask ratio (SMR) (cont.)



(g) SMR

Figure 8.6: Computing the signal-to-mask ratio (SMR) (cont.)

The container has several levels, and each level corresponds to a subband. As shown in the figure, the relative depth of a level b is $\mathrm{SMR}_b/6$, and the cross section area is N_b. The dashed lines shows the target height. If water is filled over that line, SNR > SMR for all subbands. I.e., quantization noise is masked. Oppositely, the situation as shown on the right indicates that we run out of bits to mask the noise.

The algorithm for bit allocation is as follows:

- `STEP 1:` Sort subbands according to their SMR
- `STEP 2:` Seek to fill the subband with largest SMR, if enough bits are available
  - in each round, attempt to fill a subband with 2 bits per line if the subband is empty, and 1 bit per line if not
  - if no bits left, `BREAK`
- `STEP 3:` Decrease the SMR of that band by 6 dB (if 1 bit per line is just allocated) or 12 dB (if 2 bits per line are just allocated)
- `STEP 4:` Decrease P by 1 or 2 times N_b. Go to `STEP 1`

With the information of the psychoacoustic model, the wavelet packet coefficient groups are encoded. But before the coefficients are thresholded in order to fit

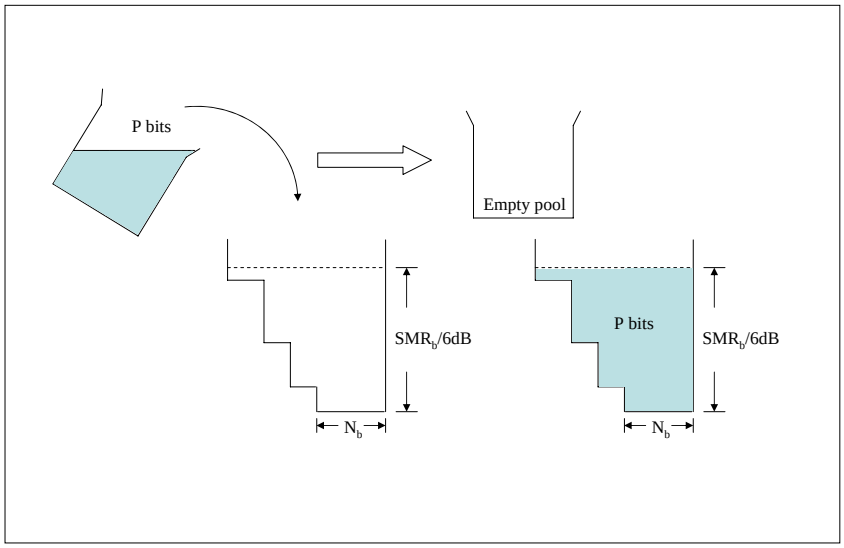


Figure 8.7: Dynamic bit allocation - the waterfilling algorithm

to the channel adaption unit information, hence, thresholding is aggressive, if the transmission tends to be erroneous while in contrary the thresholding is soft, if the communication channels tend to be clear, the sequence is requantized as described above.

The next step is the further compression by entropy coding. First of all, the re-quantized and thresholded wavelet packet coefficients are recursively RLE-0 encoded. After that the following successive processes are applied to the sequence. Burrows-Wheeler-Transformation (BWT), Move-To-Front (MTF), recursive RLE-0 and finally adaptive arithmetic compression. These steps yield to an additional compression of approximately 2:1, see figure 8.8.

8.2.3.1 Channel Adaption Unit

As discussed in chapter 5 the developed channel adaption mechanism delivers a configuration parameter for the fine-grain bit rate scalable compressor unit, which is based on the bark scaled wavelet packet transform. The configuration parameter is given by a percentage value which is used to configure the current bit rate of the top-layer compressor enabling bit rate scalability.

In order to accomplish this, the compressor uses the configuration parameter from the channel adaption unit to adaptively control the threshold for WPT-coefficient selection. The higher the percentage value is, the more coefficients will be kept, hence the resulting speech signal is of better quality.

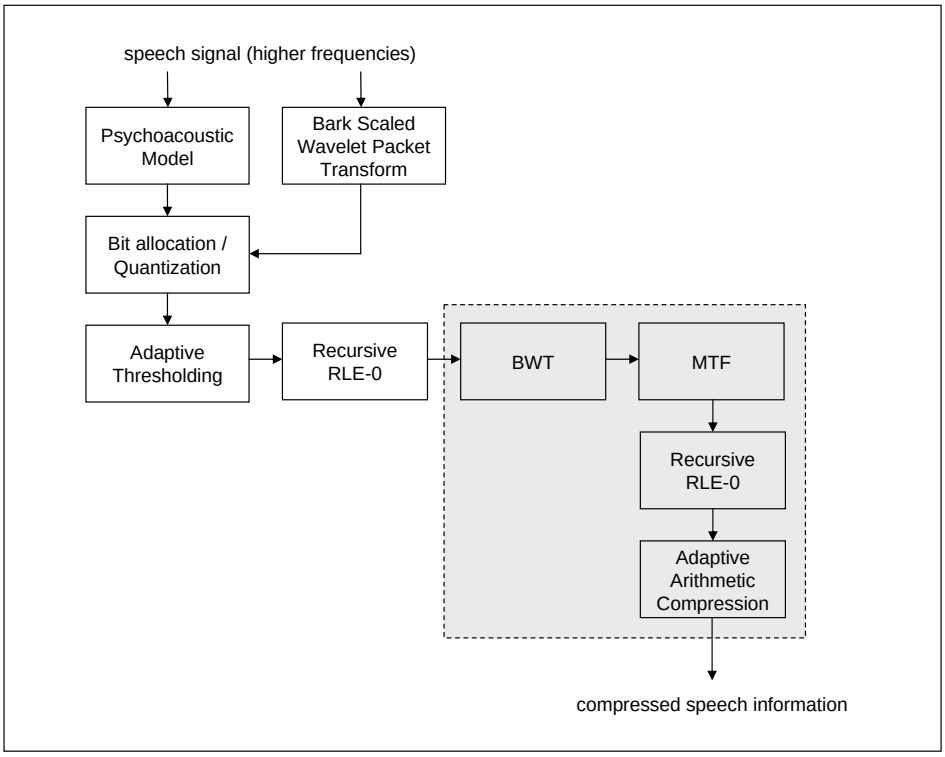


Figure 8.8: Processes of the wavelet compressor

If the configuration parameter by the channel adaption unit falls below a certain threshold, the enhancement layer formed by the WPT-compressor is omitted, as the extra computation for the frequency range of 4-8 kHz is disproportionate to the speech quality improvement by introducing a very low bit rate encoded enhancement layer.

As compression ratios of approximately 8:1 are transparently possible, the threshold of omittance is below 10 %. Simulations showed, that this value yields to unnoticeable degradations of the final speech signal enabling the wanted speech quality enhancement by introducing the WPT-compressed frequency range of 4-8 kHz.

8.2.4 Packetizer

If both compressors finished, the final compressed data stream will be prepared in the packetizer unit. This unit is in direct communication with the underlying hardware for checking how many transmission channels are currently available. If necessary, the data stream has to be prepared for one or multiple transmission channels, hence a further merging process must be applied to the data sequence if

necessary, hence less transmission channels are currently available.

As seen above, in case of the modified G.729 codec two base and one enhancement data stream is generated by default. In case of the iLBC codec three base and one enhancement data stream is send to the packetizer. As a consequence, at least 3 transmission channels must be available in order to benefit from the multiple description coding approach. Current MIMO-based solutions provide up to 4 transmission channels - see for example [PHea05, Wil04] - hence the requirements are fulfilled by default.

Additionally, if header compression is supported by the system, the respective ROHC interface has to be implemented in the packetizer unit, as this unit builds the final packets enabling the current available transmission channels. Therefore the packetizer unit needs access to a device driver in order to communicate with the ROHC interface which is usualy found between link- and network layer.

8.2.5 MIMO-based Transmission

It is well known that by transmitting independent information streams in parallel over different antennas, data rates over MIMO channels can be increased [GSS+03]. As mentioned earlier, in a wireless fading channel with sufficiently rich scattering, it is possible to achieve capacities with MIMO systems that were unthinkable even a decade ago. When the wireless channel has sufficient degrees of freedom, the data streams transmitted from multiple transmit antennas can be separated, thus leading to parallel data paths. This technique is known by *spatial multiplexing*. The capacity of the radio channel under these conditions grows with $min(M_T, M_R)$, that is, linearly with the number of antennas, where M_T is the number of transmit antennas and M_R the number of receive antennas [SN04].

As detailed in the last section, the multiple data streams consisting of base layers and possibly an enhancement layer were transmitted over parallel MIMO data paths. This approach combines two advantages:

- higher channel capacity → higher transmission speed compared to SISO systems
- increased robustness, due to multiple descriptive encoding

Therefore, such a system should lead to higher quality wireless speech communication. To prove the concept a comprehensive simulation was modeled and implemented. Before the final results are presented, the last unit of the simulation is presented - error modeling and simulation.

8.2.5.1 Error Modeling and Simulation

The improvement of robustness regarding to various transmission errors was one of the main aims while the development of a fine-grain scalable and channel-adaptive hybrid speech coding scheme. To prove the concept and to show the behaviour for various different scenarios, a comprehensive error modeling and simulation environment was developed.

Two types of simulations could be carried out:

- a simple random-based error model characterizing a specific
  1. BER (bit error rate) or
  2. LIP (loss in percent)
- a comprehensive error model based on types of different Gilbert-Elliott models

Both approaches have their right to exist. Fast and quite easy error modeling could be accomplished with the simple random-based error models. They deliver results which are comparable to others work in this field of research as quite a number of publications rely on these basic type error models.

With the simple models random packet losses occur due to random bit errors in the channel. For each bit error rate (BER) considered, the packet loss probability p using equation 8.1 is determined. For each p, 500 different trace files were created using different seeds for the random number generator and frames corresponding to the lost packets in the trace files were dropped in the encoded speech files. The packet loss probability for a given BER is given by,

$$p = 1 - (1 - \mathrm{BER})^L \tag{8.1}$$

With the latter more realistic scenarios for specific needs could be modeled. Bursty packet loss has a severe impact on Voice over IP call quality. Even if the average packet loss rate for a call is low (say one percent), the lost packets are likely to occur during short dense periods, resulting in short periods of degraded quality. Gilbert-Elliott-based error models enable to simulate different types of error bursts. Packet loss may occur due to buffer overflow within the network, deliberate discard as a result of some congestion control scheme (e.g. Random Early Detection) or transmission errors. Several of the mechanisms that can lead to packet loss are of a transient nature and hence the resulting packet loss is bursty in nature.

Bolot [BFPT99] studied the distribution of packet loss in the Internet and concluded that this could be represented by a Markovian loss model such as the Gilbert or Elliott models. Jitter (or packet delay variation) also has an effect however the use of a jitter buffer generally replaces jitter by delay and packet loss. Incoming packets are buffered and then read out at a constant rate; if packets are excessively

late in arriving then they are discarded. For this reason it is advisable to measure packet loss (or rather frame loss) between the jitter buffer and the codec. Jitter buffers are often adaptive and adjust their depth dynamically based on either the current packet discard rate or current jitter level.

If the rate of packet loss varies during a VoIP call then the perceived call quality will also vary. The term *instantaneous quality* may be used to denote the measured or calculated quality due to packet loss or other impairments and the term *perceived quality* may be used to denote the quality that the user would report at some instant in time.

Intuitively, if instantaneous quality changes from *good* to *bad* at some moment in time then the listener would not immediately notice the change. As time progresses the user would become progressively more annoyed or distracted by the impairment. This leads to the idea that the perceived quality changes more slowly than instantaneous quality.

In tests reported by Barriac et al [R&00] the packet loss rate during a 3 minute call was varied from 0% to 25%. In the example shown below (see figure 8.9) the packet loss was set to 25% for most of the call and reduced to 0% for a 30 second period mid-call. Listeners were asked to move a slider to indicate their assessment of quality during the call and then asked to rate the call at the end. This showed the effect described above, with an approximately exponential curve with a time constant of 5 seconds for the good-to-bad transition and 15 seconds for the bad-to-good transition. The *recency* effect reflects the way that a listener would remember call quality.

In tests conducted by AT&T [Ros98] a 15 second burst of noise was moved from the beginning to the end of a 60 second call. When the noise was at the start of the call, users reported a MOS score of 3.82 whereas when the noise was at the end of the call users reported a MOS score of 3.18, giving a change in MOS score of 0.64. Tests reported by France Telecom [R&00] showed a similar effect. An improvement in MOS score of 0.68 was reported when a period of high packet loss was moved from the end to the beginning of a 60 second call. The effect is believed to be due to the tendency for people to remember the most recent events [Bad97] or possibly due to auditory memory which typically decays over a 30 second interval [R&00].

A 4-state Markov model is used to represent the burst packet loss characteristics of the call - see figure 8.10. A Markov model is a general multi-state model in which a system switches between states i and j with some transition probability $p(i, j)$. A 2-state Markov model has some merit in that it is able to capture very short term dependencies between lost packets, i.e. consecutive losses. These are generally very short duration events (say 1-3 packets in length) but occasional link failures can result in very long loss sequences extending to tens of seconds. By combining the 2-state model with a Gilbert-Elliott model it is possible to capture both very short duration consecutive loss events and longer lower density events.

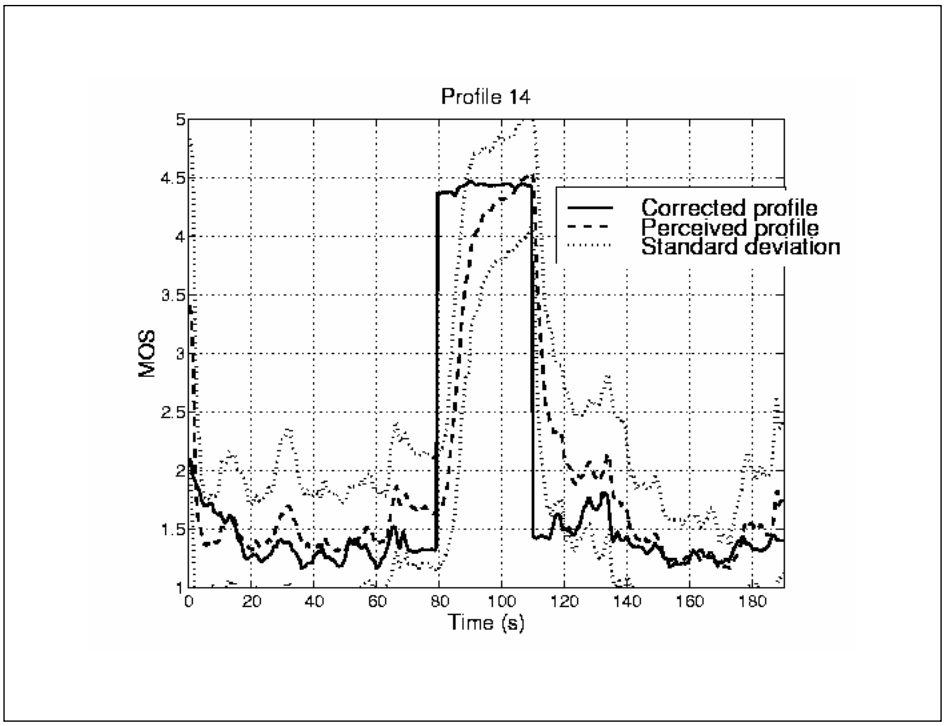


Figure 8.9: Relationship between instantaneous and perceived quality metrics (following [R&00])

This 4-state Markov model represents burst periods, during which packets are received and lost according to a first 2-state model and gap periods during which packets are received and lost according to a second 2-state model. The four states have the following definition [Cla01]:

- State 1 - Gap state - receive packet
- State 2 - Burst state - receive packet
- State 3 - Burst state - lose packet
- State 4 - Gap state - receive packet

This model is similar to the more normal Gilbert or Elliott models however includes a state representing the loss of an isolated packet within a gap. The rationale for this is that packet loss concealment (e.g. replay last packet), can mask the effects of isolated lost packets.

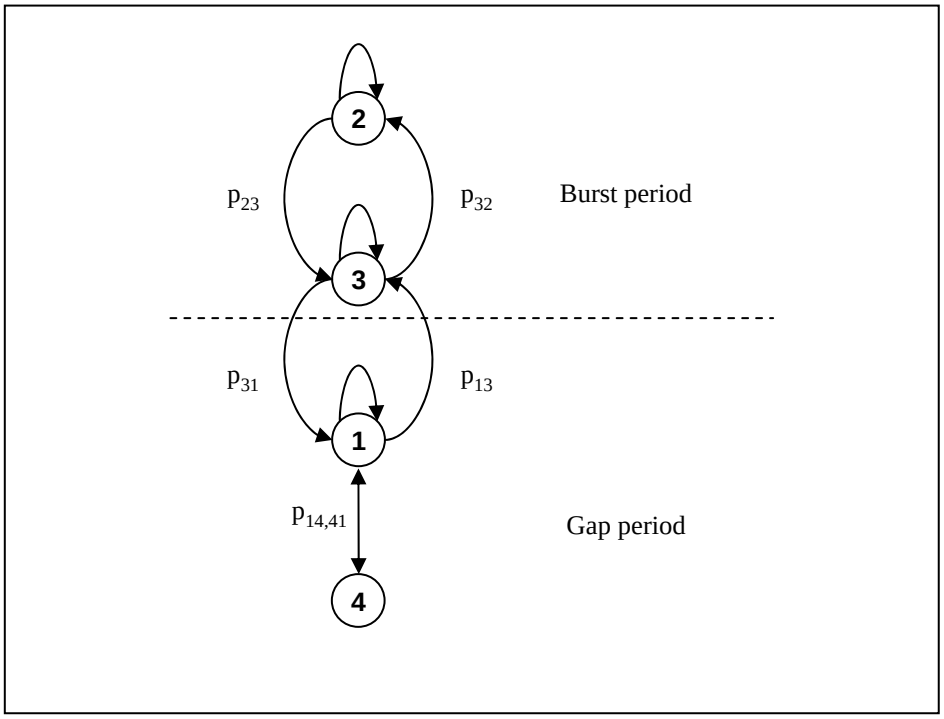


Figure 8.10: 4-State Markov Model
(following [Cla01])

It is common to define a gap state with respect to some criteria, for example a loss rate lower than some limit or some consecutive number of received packets. A convenient definition is that a burst must be a longest sequence beginning and ending with a loss during which the number of consecutive received packets is less than some value $G_{\min}$ (a suitable value for $G_{\min}$ for use with Voice over IP services would be 16 whereas for use with video services a higher value like 64 or 128 would be preferable).

Thereby every component was presented. Finally, the next section will show how the system performs under various conditions. To prove the developed concept, different scenarios were simulated.

8.3 Test Patterns

In this section various simulations and their respective results will be discussed. Improvements by the developed speech coding scheme are shown by the following analyses:

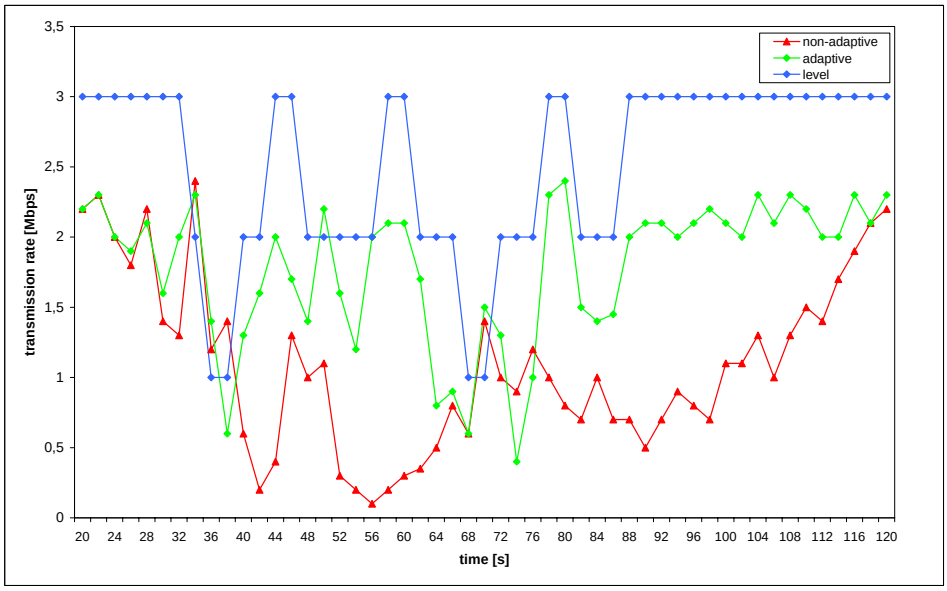


Figure 8.11: Channel-adaptive transmission outperforms non-adaptive transmission

- Channel adaptivity
- Bit rate scalability
- Error robustness
- Objective and perceived speech quality

8.3.1 Channel Adaptivity

Wireless channels condition and wireless communication in general is known to be error prone and permanently changing. In order to provide high quality communication regarding continuous transmissions, channel adaption mechanisms must be integrated. In this thesis a two-stage channel adaption scheme was developed. In both stages different channel feedback information were evaluated in order to provide a control information for the fine-grain scalability unit of the developed speech codec.

Figure 8.11 shows the performance of the channel-adaption unit for a particular example. As stated earlier in this thesis, even more tests under realistic conditions must be done, in order to get proper configuration parameters for the adaption unit. The example depicted in figure 8.11 was processed with weighting factors $\omega_{qsi} = 0.75$ and $\omega_{rtcp} = 0.25$, hence, more attention is paid to the signal information which were received from the wireless NIC.

Due to the difficulties while trying to get stable conditions regarding comparable measurements, this unit must regretfully be entitled as - *to be further investigated*. Nevertheless, various analyses and measurements were promising. The combination and therefore the continuous feedback from either the wireless NIC, the RTCP or both is capable to improve the transmission rate about approximately 20% to 30%. The side effect of the higher transmission rate is, that the waste of network capacity is reduced while transmission robustness could be increased, hence higher speech quality regarding voice over wireless could be achieved.

8.3.2 Bit Rate Scalability

Fine-grain bit rate scalability is achieved by introducing psychoacoustic enhanced wavelet packet based compression for the frequency range between 4 kHz and 8 kHz. This frequency range differentiates narrow- and wideband speech coding schemes. With narrowband speech coding only the range from approximately 0 kHz to 4 kHz is considered while encoding the speech information. With wideband speech coding the speech is more transparent and intelligible as the range from approximately 0 kHz to 8 kHz is considered.

Narrowband speech codecs like for example the G.711, G.729a and iLBC are widely used. They are implemented in hardware solutions as well as in softphones like Skype or X-Lite. Although much better speech codecs exist - all the wideband speech codecs like for example AMR-WB, they couldn't be found for various reasons that often in standard solutions. Compatibility and acceptance play a key role in spreading new and possibly advanced codecs.

Obviously it might be a good idea to reuse or improve standard codecs instead of introducing just another superior wideband speech coding scheme. This is roughly done in this thesis by modifying standard codecs and improve them with an enhancement layer which is fine-grain scalable through wavelet packet based compression improved by a psychoacoustic model with final optional entropy coding to further improve the compression ratio.

As analyses showed, the enhancement layer could be compressed in the range from about 4 Kbps (with entropy coding) to 32 Kbps (without additional entropy coding) improving the speech quality compared to the version without the enhancement layer. Figure 8.12 depicts the averaged objective speech quality measured with different speech signals (male and female speakers - german language).

The Perceptual Evaluation of Speech Quality (PESQ) tool, as defined by ITU-T recommendation G.862.2 [RBHH01, Sta05] was used for measurement.

If the enhancement layer is encoded with less than 4 Kbps, no enhancement could be subjectively notified. Describing audio quality with subjective measurements is quite difficult, but nevertheless with less than 4 Kbps the speech gets the typical *wideband sound* but the current version of the codec also introduces unwanted

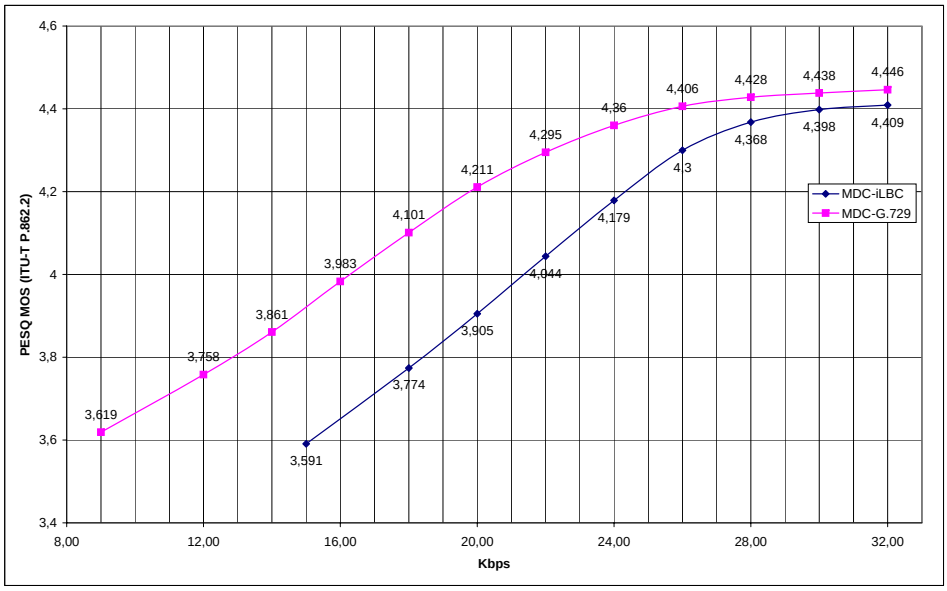


Figure 8.12: P.862.2 PESQ without packet loss

peaks and a small amount of noise which is nearly zero if encoded with 4 Kbps or more. Summarized, fine-grain scalability is also available below 4 Kbps but with no subjective enhancement to the overall speech quality.

Additionally, in the current version of the codec the introduction of entropy coding as described above (WPT-coefficients→rRLE-0→BWT→MTF→rRLE0→adaptive Huff/arith. coding) no constant output bit rate is implemented. The entropy coding process and its output rate depends on the input data, hence, a typical extra compression ratio of approximately 2:1 could be achieved.

8.3.3 Error Robustness

To show the improved error robustness of both coding approaches two classes of simulations were accomplished. The first introduces random packet losses to the communication channels ranging from 0% to 30%. The second class is based on typical wireless error characteristic - *error bursts*. Error bursts clearly appear as bit sequences with errors at their ends but that allow few corrected bits within them. Bit-level errors are, in fact, a consequence of errors in decoding symbols of the complementary code keying (CCK) modulation scheme used for the 11 Mbps data rate, where each symbol represents 8 bits of information. The mean error burst length in 802.11 wireless networks was analyzed by Servetti and De Martin [SM05] to be 14.81 bits. Burst lengths in the range of 5 to 30 bits were measured -

the vast majority of bit error bursts last from 13 to 16 bits. Consequently, for the simulation described in this thesis, error bursts with the described characteristic were investigated.

8.3.3.1 Random Loss [0-30%]

The first class of accomplished simulations deals with random packet loss. Packet loss rates in the range from 0% to 30% were investigated for the following configurations

- iLBC - unmodified narrowband version
- MDC-iLBC (without enhancement layer)
- MDC-iLBC (with additional enhancement layer 3 Kbps to 17 Kbps)
- G.729 - unmodified narrowband version
- MDC-G.729 (without enhancement layer)
- MDC-G.729 (with additional enhancement layer 3 Kbps to 23 Kbps)

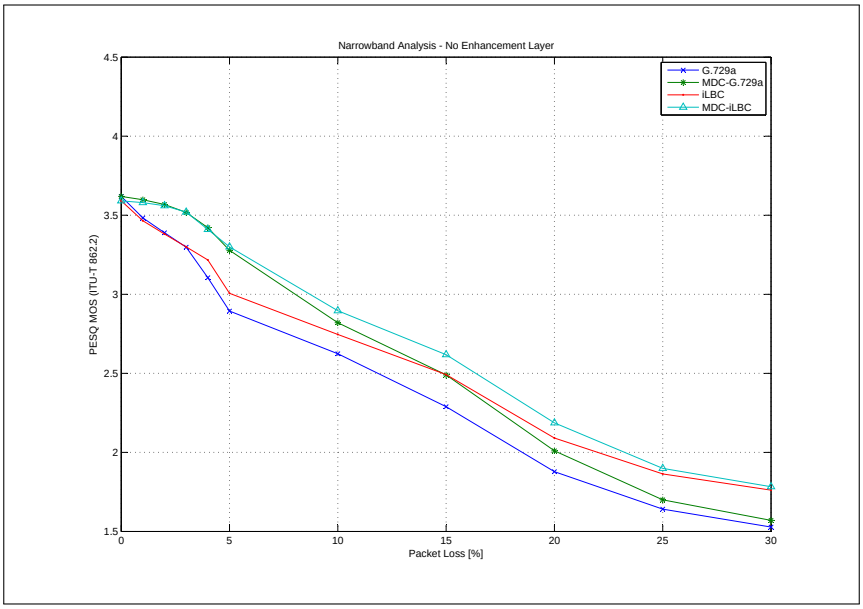
with the respective modifications to the original versions of iLBC and G.729. The first figure 8.13a depicts the situation with no enhancement layer. Therefore the measurement with the ITU-T PESQ 862.2 tool was accomplished in narrowband mode, while the following figures 8.13b - 8.13l depicts the respective situation with an enhancement layer that consequently the measurement was done in wideband mode.

8.3.3.2 Error Bursts [0-8%]

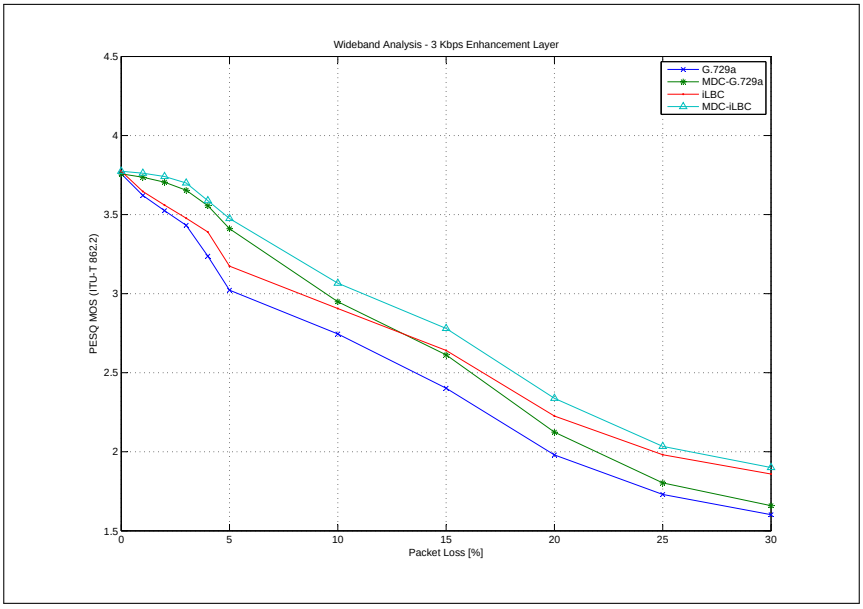
The second class of accomplished simulations deals with error bursts with the characteristic as described above. These bursts were applied to the communication channels for the following configurations

- MDC-iLBC (without enhancement layer)
- MDC-iLBC (with additional enhancement layer 3 Kbps to 17 Kbps)
- MDC-G.729 (without enhancement layer)
- MDC-G.729 (with additional enhancement layer 3 Kbps to 23 Kbps)

with the respective modifications to the original versions of iLBC and G.729.

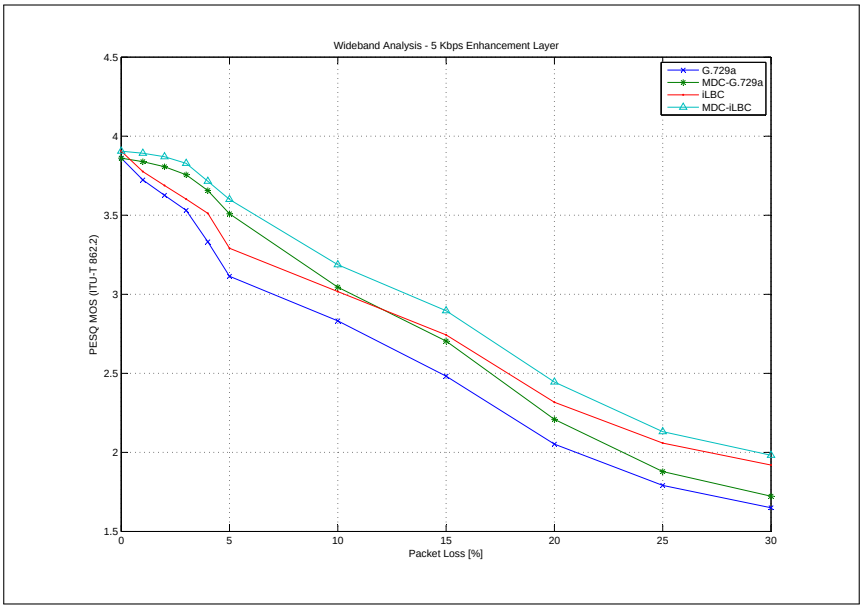


(a) NOE

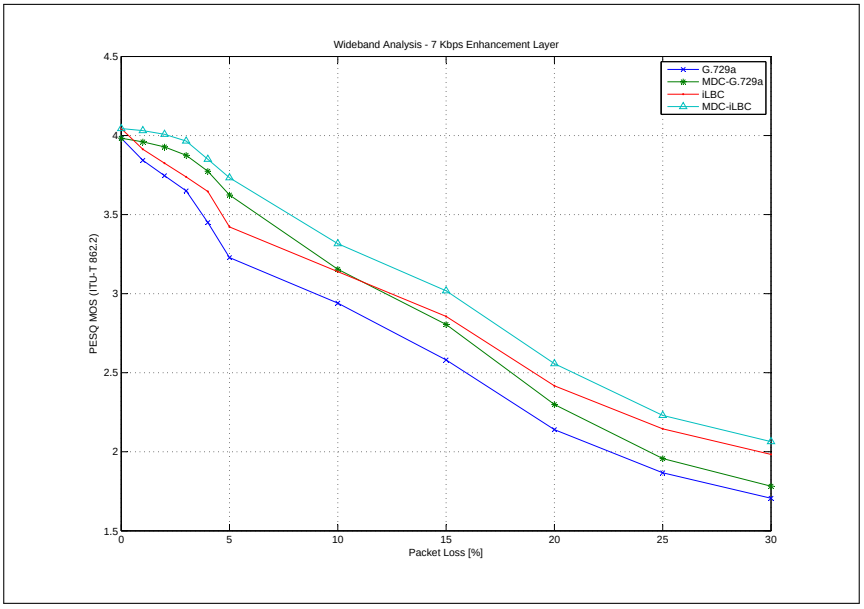


(b) 3kE

Figure 8.13: Random Loss Analyses

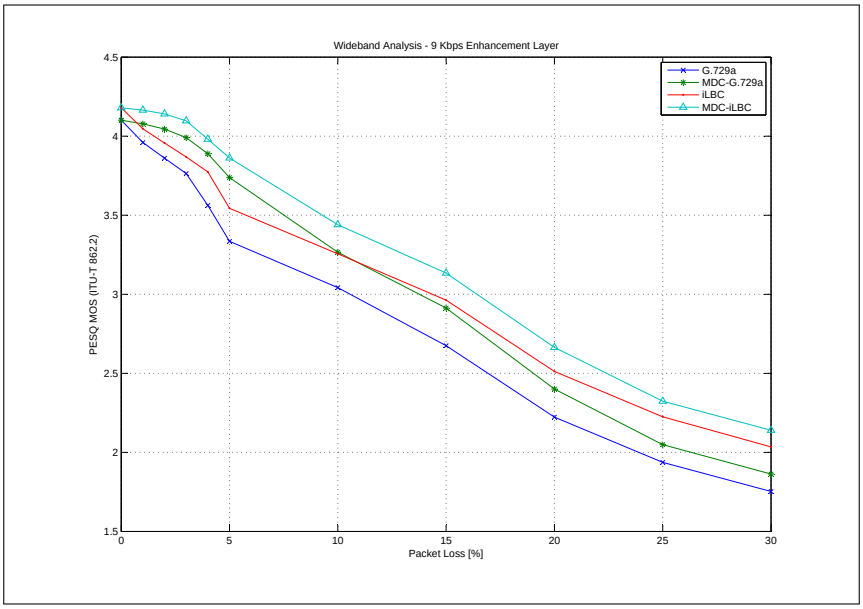


(c) 5kE

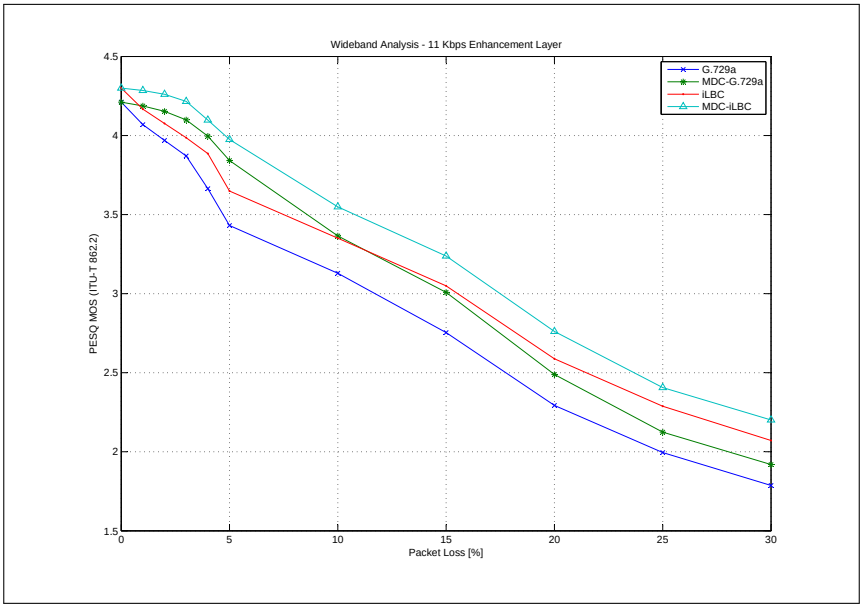


(d) 7kE

Figure 8.13: Random Loss Analyses (cont.)

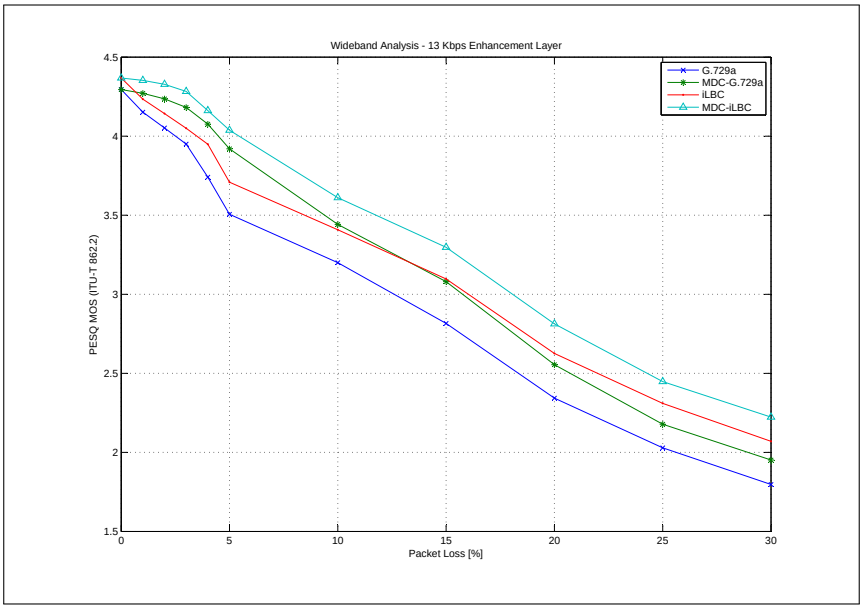


(e) 9kE

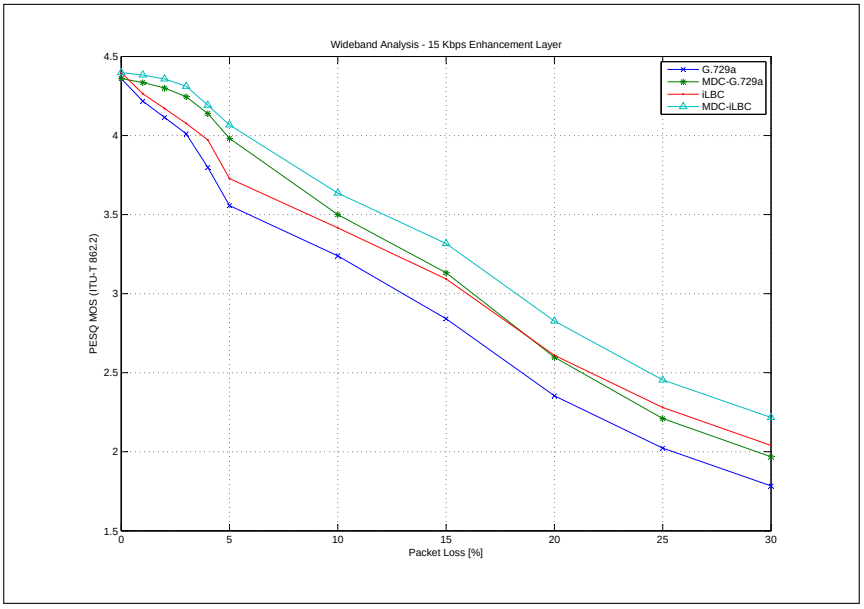


(f) 11kE

Figure 8.13: Random Loss Analyses (cont.)

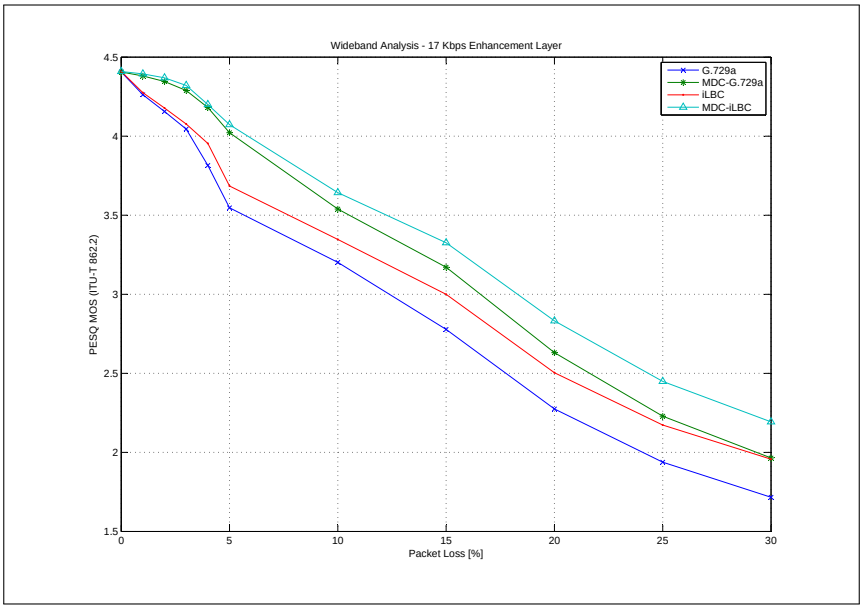


(g) 13kE

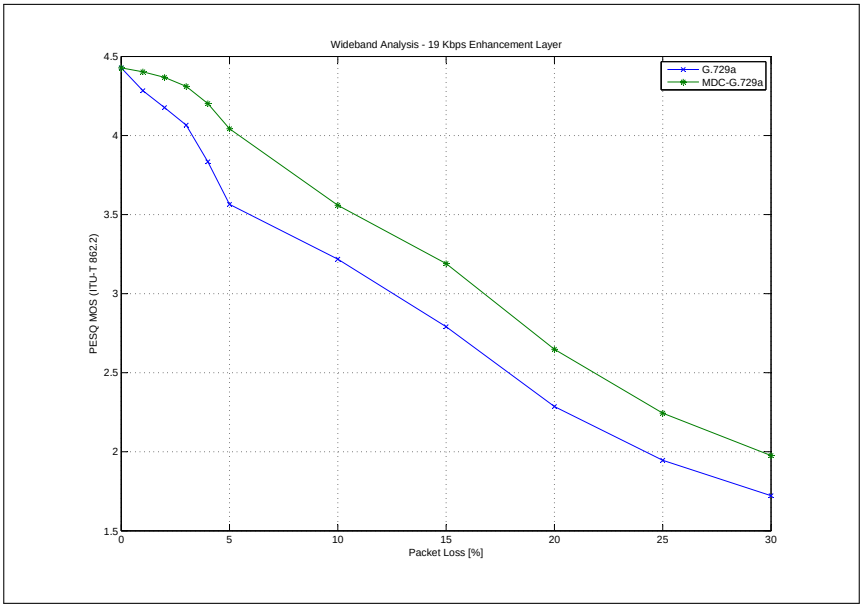


(h) 15kE

Figure 8.13: Random Loss Analyses (cont.)

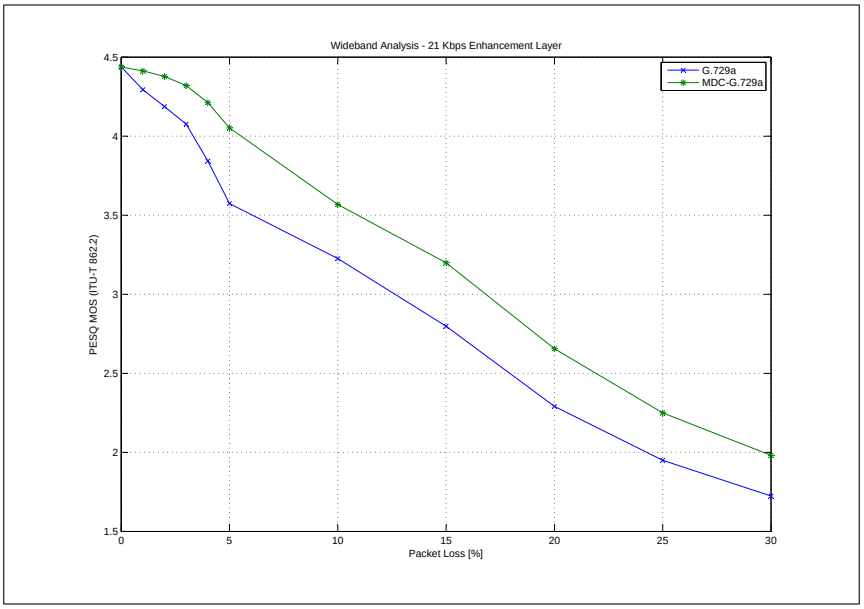


(i) 17kE

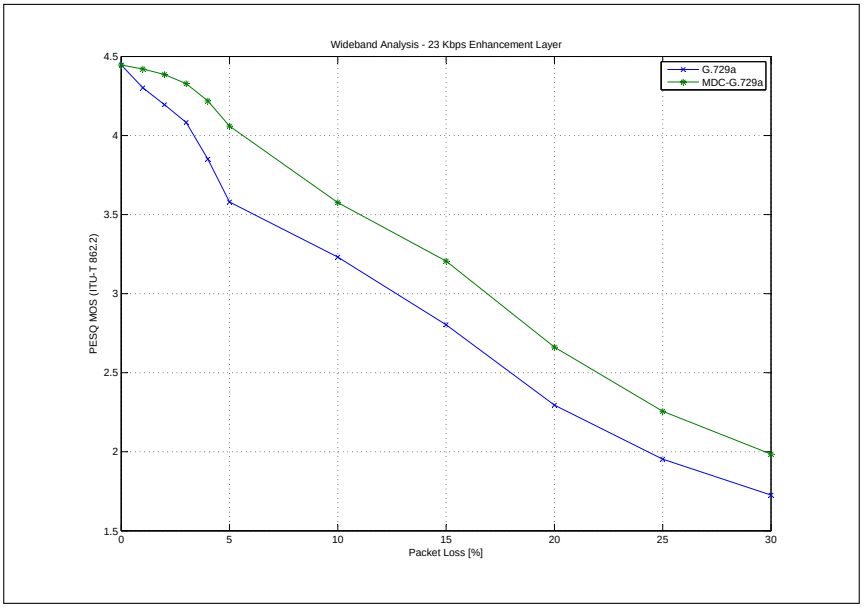


(j) 19kE

Figure 8.13: Random Loss Analyses (cont.)



(k) 21kE



(l) 23kE

Figure 8.13: Random Loss Analyses (cont.)

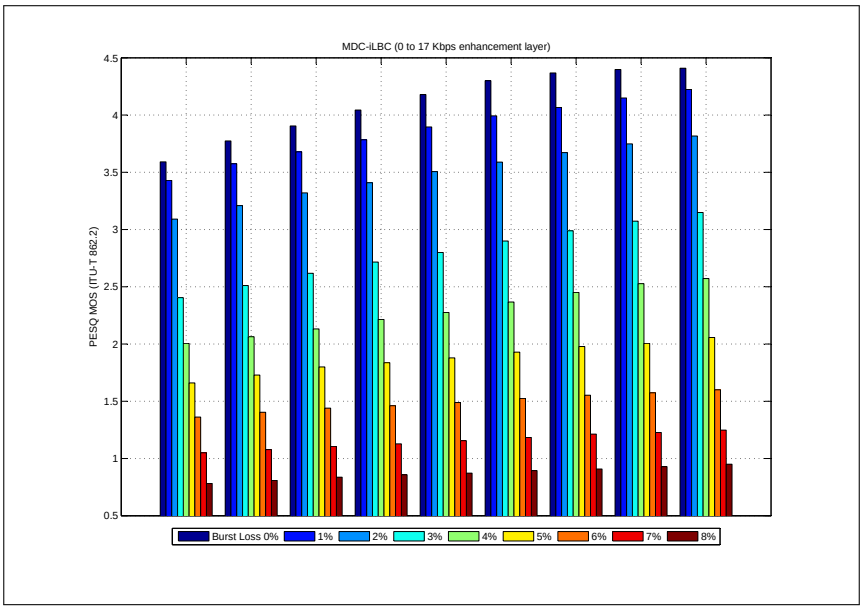


Figure 8.14: Error Bust Analysis for MDC-iLBC

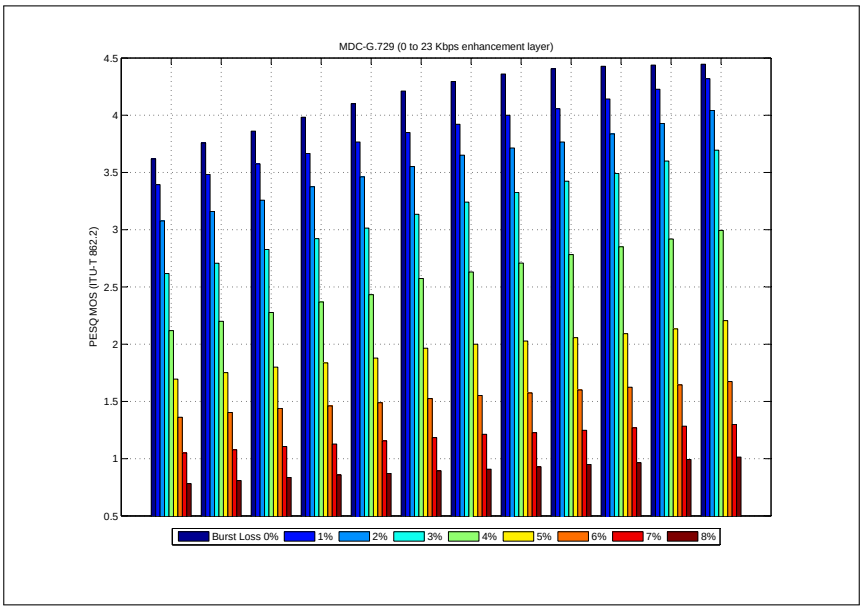


Figure 8.15: Error Bust Analysis for MDC-G.729

8.4 Summary

The simulations as detailed above were accomplished with different source material. German language female and male speech data recorded at 44.1 kHz/16 bit. Due to the initial higher PESQ MOS of the G.729 without loss the MDC-G.729 could reach a slightly higher final PESQ MOS score with full enhancement compared to the MDC-iLBC. The second important fact to mention is, that the G.729/MDC-G.729 performs better at equal bit rates, but compared to iLBC/MDC-iLBC it couldn't profit that fast of additional enhancement informations. Furthermore the error concealment with the G.729 is not that successful as the techniques used for iLBC-based encoding, hence, the G.729-based codecs are more error prone. Nevertheless both encoding approaches provide superior quality compared to the none MDC-versions. Especially with error probability less or equal to 5% the preeminence is unambiguous.

Also subjective impressions confirm the objective measurements with the ITU-T P.862.2. Both MDC approaches yield to enhanced speech quality in case of random and burst errors. Although simulations were accomplished with enhancement layers up to 23 Kbps the gained quality improvement is not clearly noticeable subjectively above a PESQ MOS score of 4.1 - but to be clear - this is my very own impression that is not confirmed by others so far. Finally, the fine-grain scalability of the enhancement layer could effectively be used to control the bit rate in order to adapt to changing channel conditions.

9 Conclusions

This thesis proposes methods to enhance the robustness of packetized voice communication over wireless links. Novel and substantial achievements have been accomplished as summarized in the following:

A universal network simulation environment with focus to wireless error simulation capabilities was developed. This system is universal in kind of the generality in which different applications on top of this unit could be analysed. As built on the `iptables` Linux-Kernel filter package, various configurations regarding to the network packet flow could be managed. Therefore this encapsulated and simulated wireless world could be projected to any kind of wired networks with more reliable channel characteristics for analyses purposes. The error-generator unit is able to process a multitude of diverse network situations by XML configuration files. The behaviour is controllable by MAC- and respective error-models which are fed by trace files which characterize specific situations and their typical error bearing.

Chanel-adaption was the next important approach. Analyses were made in order to model a unit which acts in response to technology constrained network quality changes in order to enable high quality voice communication over wireless links even for poor channel conditions. In combination with bit rate fine-grain scalability which is accomplished by WPT-based encoding these two elements of the system could lead to better quality voice communication in difficult environments.

Moreover the developed speech coding scheme is hybrid. Hence, the speech input signal is split up in to two parts which are handled by two different kind of encoders. The lower frequency part is processed by an CELP-based MDC encoder while the higher frequency part is processed by an WPT-based encoder. This hybrid approach enables easy adaption to current industry used codec packages for the public market as with slight modifications to convert SDC-based to MDC-based codecs it will be possible to use existent hard- and software which yields to modest costs by introducing this technology developed in this thesis.

MIMO promises improvements in WLAN throughput, range and reliability that will broaden the usefulness of wireless for applications that demand ever-greater performance. IEEE 802.11n, which is still not standardized, will introduce MIMO-based communication. Basically, while 802.11a/b/g technologies use single transmitting and dual receiving antennas, MIMO uses multiple transmitting antennas, multiple receiving antennas and a lot of signal processing on both ends to create a complex, 3-D radio-frequency transmission. Hence, MIMO adds a third *spatial*

dimension, and in fact depends upon multipath, previously an impediment, for it to work properly. This is exactly, where MDC-based speech coding must lead to higher robustness and therefore improved speech quality over wireless links. The developed speech coding scheme bridges the gap between innovative technologies and highly improved classical speech coding reinforced with new ideas and concepts which perfectly fits MIMO-based communication.

Summarized the developed speech coding scheme was analyzed to be more robust for various channel conditions in the range from 0% to 30% random packet loss than comparable SDC variants.

In many areas, further research and enhancements is possible because in this new field of research only a fraction of all questions were answered. For example, the following issues can be addressed in future research:

In order to further improve robustness in wireless voice communication investigations in the field of channel adaptivity have to be accomplished. The proposed method looks promising but more practical studies must lead to suitable input parameters for the proposed channel adaption unit.

Additionally, more enhanced psychoacoustic models must be investigated to improve the compression ratio. As discussed earlier, the robustness in the situation of burst errors is still weak for rates above 7%. If the compression ratio could be increased without intelligible distortion, the robustness and efficiency might be further improved even in the situation of error bursts.

By the end of this work, the Ph.D. thesis of C. Hoene. was encountered. C. Hoene et al. developed a lite version of the ITU-T PESQ standard which could be used for realtime measurement of speech quality [Hoe05]. Further investigations with the developed fine-grain scalable and channel-adaptive MDC speech coding scheme should be accomplished with this new tool. Adjustments to the speech processing system and the channel adaption unit could be analyzed more efficiently in order to improve the whole concept.

A most interesting question is whether this thesis helps the development of new products. A promising area for patenting are the WPT-MDC algorithm modifications. Patenting these algorithms and selling them to producers of wireless telephone equipment promises economical success because they can improve the robustness of wireless communication. Thus, founding a start-up company, which focuses on the development of MDC based speech codecs and algorithms, would be a worthwhile goal.

Bibliography

[3GP01] 3GPP TS 26.190. AMR Wideband Speech Codec; Transcoding Functions, 2001.

[ACRG99] A. Aggarwal, V. Cuperman, K. Rose, and A. Gersho. Perceptual zerotrees for scalable wavelet coding of wideband audio. In *1999 IEEE Workshop on Speech Coding Proceedings*, pages 16–18, 1999.

[ADA+04] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, and J. Linden. Internet Low Bit Rate Codec (iLBC). RFC 3951, 2004.

[AK03] J. Aráuz and P. Krishnamurthy. Markov Modeling of 802.11 Channels. In *Proceedings of The* 58^{th} *IEEE Semiannual Vehicular Technology Conference*, volume 2, pages 771–775, 2003.

[AMV00] A.K. Anandakumar, A.V. McCree, and V. Viswanathan. Efficient CELP-based diversity schemes for VoIP. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, volume 6, pages 3682–3685, 2000.

[APL+95] E. Ayanoglu, S. Paul, T.F. LaPorta, K.K. Sabnani, and R.D. Gitlin. AIRMAIL: A Link-Layer Protocol for Wireless Networks. *ACM ACM/Baltzer Wireless Networks Journal*, 1:47–60, 1995.

[Bad97] A. Baddeley. *Human Memory.* Taylor & Francis Ltd., 1997.

[Bar02] J. Bardwell. Converting Signal Strength Percentage to dBm Values. `http://www.wildpackets.com`, 2002. Last request: Jan 12, 2006.

[BB95] A. Bakre and B.R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proc. 15th International Conf. on Distributed Computing Systems (ICDCS)*, 1995.

[BBea01] C. Bormann, C. Burmeister, and M. Degermark et al. RObust Header Compression (ROHC): Framework and four profiles. RFC 3095, 2001.

[BDV00] R. Balan, I. Daubechies, and V. Vaishampayan. The analysis and design of windowed fourier frame based multiple description source coding schemes. *IEEE Trans. Inform. Th.*, 46(7):2491–2536, 2000.

[BFPT99] J.C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive FEC based Error Control for Internet Telephony. In *INFOCOM 1999*, pages 1453–1460, 1999.

[Bis95] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[BK98] K. Brandenburg and M. Kahrs. *Applications of Digital Signal Processing to Audio and Acoustics.* Kluwer Academic Publishers, 1998.

[BSK95] H. Balakrishnan, S. Seshan, and R.H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, 1(4), 1995.

[BSL+02] B. Besette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen. The Adaptive Multirate Wideband Speech Codec (AMR-WB). *IEEE Trans. on Speech and Audio Processing*, 10(8):620–636, 2002.

[Bun85] J.R. Bunch. Stability of methods for solving Toeplitz systems of equations. *SIAM J. Sci. Stat. Comput.*, 6:349–364, 1985.

[BVG96] J.C. Bolot and A. Vega-Garcia. Control mechanisms for packet audio in the Internet. In *INFOCOM 1996*, volume 1, pages 232–239, 1996.

[BW94] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical Report SRC-RR-124, "Systems Research Center, CA", 1994.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205, 1997.

[BZZ05] H. Bai, Y. Zhao, and C. Zhu. Optimized Multiple Description Image Coding Using Lattice Vector Quantization. In *Proc. of ISCAS'05*, 2005.

[CA04] K. Carlberg and R. Atkinson. General Requirements for Emergency Telecommunication Service (ETS). RFC 3689, 2004.

[Cal05] Calleveryone. User Agreement. `http://www.calleveryone.com/terms.shtml`, 2005. Last request: Dec 09, 2005.

[Can91] C. N. Canagarajah. A single-input hearing aid based on auditory perceptual features to improve speech intelligibility in noise. In *Proc. IEEE Workshop on Audio and Acoustics*, 1991.

[CCHC01] W.-T. Chen, D.-W. Chuang, and H.-C.Hsiao. Enhancing CRTP by retransmission for wireless networks. In *Proceedings of the Tenth International Conference on Computer Communications and Networks*, pages 426–431, 2001.

[CFW+01] A. Cellatoglu, S. Fabri, S. Worral, A. Sadka, and A. Kondoz. Robust header compression for real-time services in cellular networks. In *Proceedings of the IEE3G 2001*, pages 124–128, 2001.

[CHSSP03] G. Camponovo, M. Heitmann, K. Stanoevska-Slabeva, and Y. Pigneur. Exploring the WISP industry: Swiss case study. In *Proceedings of the 16th Bled Electronic Commerce Conference*, 2003.

[CI95] R. Caceres and L. Iftode. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. *IEEE Journal on Selected Areas in Communications*, 13(5), 1995.

[CJ99] S. Casner and V. Jacobson. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. RFC 2508, 1999.

[Cla01] A. Clark. Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality. IPtel 2001 Workshop, 2001.

[CSR04] R. Chandramouli, K.P. Subbalakshmi, and N. Ranganathan. Stochastic channeladaptive rate control for wireless video transmission. *Pattern Recognition Letters*, 25(7):793–806, 2004.

[CW84] J.G. Cleary and I.H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Trans. Comm.*, COM-32(4):396–402, 1984.

[CWKS97] B.P. Crow, I. Widjaja, J.G. Kim, and P.T. Sakai. IEEE 802.11 Wireless Local Area Netoworks. *IEEE Communications Magazine*, 35(9):116–126, 1997.

[DCBR+04] H. Dong, I. Chakeres, E. Belding-Royer, A. Gersho, and J. Gibson. Selective bit-error checking at the MAC layer for Voice over Mobile Adhoc Networks with IEEE 802.11. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2004.

[dCK05] A.F. da Conceição and Fabio Kon. Adaptive Streaming Based on IEEE 802.11 Signal Quality. `http://gsd.ime.usp.br`, 2005. Last request: Jan 12, 2006.

[DF99] I. Dalgic and H. Fang. Comparison of H.323 and SIP for IP Telephony Signaling. In *Proceedings of SPIE. Multimedia Systems and Applications II, ser. Proceedings of Photonics East, Tescher, Vasudev, Bove, and Derryberry, Eds.*, volume 3845, 1999.

[DG+04] H. Dong, , A. Gersho, , V. Cuperman, and J. Gibson. A multiple description speech coder based on AMR-WB for mobile ad hoc networks. In *Proceedings of the IEEE ICASSP'04*, 2004.

[DNP99] M. Degermark, B. Nordgren, and S. Pink. IP Header Compression. RFC 2507, 1999.

[Dol67] R.M. Dolby. An audio noise reduction system. *J. Audio Eng. Soc.*, 15(4):383–388, 1967.

[Don00] S. Donovan. The SIP INFO Method. RFC 2976, 2000.

[DR02] J. Daemen and V. Rijmen. *The Design of Rijndael, AES - The Advanced Encryption Standard.* Springer Verlag, 2002.

[Dun01] C. Dunn. Efficient audio coding with fine-grain scalability. AES 111th Convention, NY, USA, preprint 5492, 2001.

[ea97] C.E. Perkins et al. RTP Payload for Redundant Audio Data. RFC 2198, 1997.

[ea03] M.F. Tolba et al. TORNADO - A Novel 256-bit Block Cipher. In *Proc. of ITI First International Conference on Information & Communication Technology*, pages 77–90, 2003.

[ea05] K.H. Garbe et al. VoIPSEC - Studie zur Sicherheit von Voice over Internet Protocol. Bundesamt für Sicherheit in der Informationstechnik, 2005.

[EW99] J.P. Ebert and A. Willig. A Gilbert-Elliott Bit Error Model and the Efficient Use in Packet Level Simulation. Technical report, Telecommunication Network Group, Technische Universität Berlin, March 1999.

[FE99] M. Fleming and M. Effros. Generalized multiple description vector quantization. In *Proc. Data Compr. Conf.*, 1999.

[Fen96] P. Fenwick. Block Sorting Text Compression — Final Report, 1996.

[FF96] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno, and SACK TCP. *Computer Communications Review*, 1996.

[FHSR03] F. Fitzek, S. Hendrata, P. Seeling, and M. Reisslein. Header compression schemes for wireless internet access, 2003.

[FIP01] FIPS PUB 197. Specification for the advanced encryption standard (aes). Federal Information Processing Standards Publication 197, 2001. Last request: Feb 20, 2006.

[Fra05] J.M. François. `http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm`, 2005. Last request: Jan 05, 2006.

[Gab06] C. Gabriel. WiMAX and 3G: forget war, focus on solid steps to workable networks. *WiMAX Watch*, 2(21):2–11, 2006.

[GC82] A.A. El Gamal and T.M. Cover. Achievable rates for multiple descriptions. *IEEE Trans. Inform. Theory*, IT-28:851–857, 1982.

[Gil60] E.N. Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech. Journal*, 39:1253–1256, 1960.

[GK01] V.K. Goyal and J. Kovaĉević. Generalized multiple descriptions coding with correlating transforms. *IEEE Trans. Inform. Th.*, 47(6):2199–2224, 2001.

[Gök05] M. Gök. Entwicklung und Implementierung eines kanaladaptiven Algorithmus zur dynamischen Steuerung der Datenrate für VoIP auf Basis von RTP/RTCP. Bachelor's thesis, Georg-August Universität Göttingen, 2005.

[GKK01] V.K. Goyal, J. Kovaĉević, and J. Kellner. Quantized frame expansions with erasures. *Journal of Appl. and Comput. Harmonic Analysis*, 10(3):200–203, 2001.

[GKK02] V.K. Goyal, J.A. Kelner, and J. Kovaĉević. Multiple description vector quantization with a coarse lattice. *IEEE Trans. Inform. Th.*, 48(3):781–788, 2002.

[GN04] S. Grech and J. Nikkanen. A Security Analysis of Wi-Fi Protected Access. The 9th Nordic Workshop on Secure IT-systems, Finland, 2004.

[Goo86] D. Goodman. Waveform substitution techniques for recovering missing speech segments in packet voice communications. *IEEE Trans. Acoust., Speech, Signal Processing*, pages 1440–1448, 1986.

[Goy98] V.K. Goyal. Optimal multiple description transform coding of gaussian vectors. In *Proc. IEEE Data Compression Conf.*, pages 388–397, 1998.

[GR98] S.J. Godsill and P.J.W. Rayner. *Digital Audio Restoration - a statistical model based approach.* Springer-Verlag London, 1998.

[Gra84] R.M. Gray. Vector Quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

[GRC98] S.J. Godsill, P.J.W. Rayner, and O. Cappé. *Digital audio restoration.* In K. Brandenburg and M. Kahrs, editors, Applications of Digital Signal Processing to Audio and Acoustics. Kluwer Academic Publishers, 1998.

[Gro05] Visiongain Research Group. The 4G Report 2005: Changing The Wireless Value Chain . `http://www.electronics.ca/reports/wireless/4G.html`, 2005. Last request: Jan 16, 2006.

[GSS+03] D. Gesbert, M. Shafi, D.s. Shiu, P.J. Smith, and A. Naguib. From theory to practice: an overview of MIMO space-time coded wireless systems. *IEEE Journal on Selected Areas in Communications*, 21(3):281–302, 2003.

[GV93] M.W. Garrett and M. Vetterli. Joint source/channel coding of statistically multiplexed real time services on packet networks. *ACM/IEEE Trans. Networking*, 1(1):71–80, 1993.

[GWAR05] A. Ghosh, D.R. Wolter, J.G. Andrews, and R.Chen. Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential. *IEEE Communications Magazine*, 43:129–136, 2005.

[HJ98] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327, 1998.

[Hoe05] C. Hoene. *Internet Telephony over Wireless Links.* PhD thesis, Technische Universität Berlin, Dezember, 2005.

[HS93] K. Hamied and G.L. Stuber. A non-iterative algorithm for estimating the impulse response of ISI channels. *Wireless Personal Communications*, pages 175–186, 1993.

[HSSR99] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. SIP: Session Initiation Protocol. RFC 2543, 1999.

[Huf52] D.A. Huffman. A Method for the Construction of Minimum Redundancy Codes. In *Proceedings of IRE 40*, pages 1098–1101, 1952.

[Hwa01] J. Hwang. Internet or Circuit Switched Telephony: Cost and QoS of Large Scale Integrated Service Networks. IPNet 2001 Conference, Paris, France, 2001.

[Ins96] European Telecommunications Standards Institute. Digital cellular telecommunications systems; enhanced full rate (efr) speech transcoding (gsm 06.60), 1996.

[ISP05] D.T. Ives, D.R.R. Smith, and R.D. Patterson. Discrimination of speaker size from syllable phrases. *J. Acoust. Soc. Am.*, 118(6):3816–3822, 2005.

[ITU02] ITU-T Recommend. G.722.2. Wideband Coding of Speech at Around 16 Kbps Using Adaptive Multi-Rate Wideband (AMRWB), 2002.

[IV95] A. Ingle and V.A. Vaishampayan. DPCM system design for diversity systems with applications to packetized speech. *IEEE Trans. Speech and Audio Processing*, 3(1):48–58, 1995.

[Jac88] V. Jacobson. Congestion Avoidance and Control. In *ACM SIGCOMM '88*, 1988.

[Jac90] V. Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. RFC 1144, 1990.

[JB88] V. Jacobson and R. T. Braden. TCP Extensions for Long Delay Paths. RFC 1072, 1988.

[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 Trans. Commun.*, COM-29(2):101–109, 1981.

[Jel68] F. Jelinek. *Probabilistic Information Theory.* New York: McGraw-Hill, 1968.

[JG73] G.R. Kinzie Jr. and D.W. Gravereaux. Automatic detection of impulse noise. *J. Audio Eng. Soc.*, 21(3):331–336, 1973.

[JlCH01] A. Jensen and A. la Cour-Harbo. *Ripples in Mathematics - The Discrete Wavelet Transform.* Springer Verlag, 2001.

[JO00] W. Jiang and A. Ortega. Multiple description speech coding for robust communication over lossy packet networks. In *Proc. IEEE Int. Conf. Multimedia and Expo*, volume 1, pages 444–447, 2000.

[Jor86] E.C. Jordan. *Reference Data for Engineers: Radio, Electronics, Computer, and Communications*. Howard W. Sams and Co., 1986.

[JV97] K. Järvinen and J. Vainio. GSM enhanced full rate speech codec. In *IEEE International conference on acoustics, speech and signal processing*, volume 2, pages 771–774, 1997.

[Kap02] I. Kaplan. Frequency Analysis Using the Wavelet Packet Transform. `http://www.bearcave.com/misl/misl_tech/wavelets`, 2002. Last request: Nov 15, 2005.

[Kar99] A. Karim. H.323 and Associated Protocols. `http://www.cse.wustl.edu/~jain/cis788-99/ftp/h323/index.html`, 1999. Last request: Nov 29, 2005.

[KC91] P. Karn and C.Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. *ACM Transactions on Computer Systems*, 9(4):364–373, 1991.

[KCea03] T. Koren, S. Casner, and J. Geevarghese et al. Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering. RFC 3545, 2003.

[KCZH00] L. Khiem, C. Clanton, L. Zhigang, and Z. Haihong. Efficient and robust header compression for real-time services. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 924–928, 2000.

[KD88] P. Kroon and F. Deprettere. A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16 Kbps. *IEEE Journal on selected areas in communications*, 6(2):353–363, 1988.

[KE97] R. Kattenbach and T. Englert. Wideband statistical modelling of indoor radio channels based on the Time-variant Transfer Function. COST 259, TD(97)071, Lisbon, Portugal, 1997.

[KGK00] J.A. Kelner, V.K. Goyal, and J. Kovaĉević. Proc. IEEE Data Compression Conf. In , pages 480–489, 2000.

[KM97] S. Keshav and S. Morgan. SMART Retransmission: Performance with Overload and Random Losses. In *Proc. Infocom '97*, 1997.

[KO02] T. Karygiannis and L. Owens. Wireless Network Security 802.11, Bluetooth and Handheld Devices. Recommendations of the National Institute of Standards and Technology - Special Publication 800-48, 2002.

[Kon99] A.M. Kondoz. *Digital Speech: Coding for Low Bit Rate Communications Systems*. John Wiley & Sons, 1999.

[KV98] J. Kivinen and P. Vainikainen. Analysis of wideband indoor propagation measurements at 5.3 GHz. COST 259, TD(98)067, Bradford, UK, 1998.

[KV99] J. Kivinen and P. Vainikainen. Indoor propagation measurements at 5 GHz band. COST 259, TD(99)064, Vienna, Austria, 1999.

[KW69] A. Kurtenbach and P. Wintz. Quantizing for noisy channels. *IEEE Transactions on Communications*, 17(2):291–302, 1969.

[KZJL03] A. Konrad, B.Y. Zhao, A.D. Joseph, and R. Ludwig. A Markov-Based Channel Model Algorithm for Wireless Networks. Wireless Networks 9, 189–199, Kluwer Academic Publishers, 2003.

[Lan84] G. Langdon. An Introduction to Arithmetic Coding. *IBM Journal of Research and Development*, 28:135–149, 1984.

[Lax97] P.D. Lax. *Linear Algebra*. John Wiley & Sons, New York, 1997.

[LDS99] B. Leslie, C. Dunn, and M. Sandler. Developments with a Zero Tree Audio Codec. In *Proc. AES 17th International Conf. High Quality Audio Coding*, pages 251–257, 1999.

[Lem02] John J. Lemmon. Wireless Link Statistical Bit Error Model. NTIA Report 02-394, 2002.

[LH97] M. Lorber and R. Hoeldrich. A combined approach for broadband noise reduction. In *Proc. IEEE Workshop on Audio and Acoustics*, 1997.

[Lim83] J.S. Lim. *Speech Enhancement*. Prentice-Hall, 1983.

[LK02] Z. Liu and L.J. Karam. Quantifying the intra and inter subband correlations in the zerotree-based wavelet image coders. In *Conf. Record of the 36th Asilomar Conf. on Signals, Systems and Computers*, pages 1730–1734, 2002.

[LN04] P. Lo and S. Ngai. Characterizing Errors in Wireless LANs. Thesis, University of New South Wales, 2004.

[LO79] J.S. Lim and A.V. Oppenheim. Enhancement and bandwidth compression of noisy speech. In *Proc. IEEE*, 67(12), 1979.

[Loe60] M. Loeve. *Probability Theory, 2nd ed.* Van Nostrand Reinhold, Princeton, NJ, 1960.

[LP98] Z. Lu and A. Pearlman. An efficient, low-complexity audio coder delivering multiple levels of quality for interactive applications. In *IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, pages 529–534, 1998.

[Lut01] P. Luthi. RTP Payload Format for ITU-T Recommendation G.722.1. RFC 3047, 2001.

[McE03] C. McElroy. Principles of Speech Coding. Jaynta Ltd., `http://www.jaynta.com/`, Ireland, 2003. Last request: Nov 06, 2003.

[MMFR96] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018, 1996.

[MT05] E.E. Mier and R.B. Tarplay. Getting a grip on wireless. *Business Communications Review 11/05*, pages 34–39, 2005.

[MvOV96] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[Nak60] M. Nakagami. The m-distribution – a general formula of intensity distribution of rapid fading. In W.C. Hoffman (editor): Statistical methods of radio wave propagation, Pergamon Press, 1960.

[NAS86] M. Nishiguchi, K. Akagiri, and T. Suzuki. A New Audio Bit Rate Reduction System for the CD-I Format. Preprint 2375, 81st Audio Engineering Society Convention, 1986.

[Nel96] M. Nelson. *The Data Compression Book.* M&T Books, New York, 1996.

[New05] D. Newman. Review: Voice over Wireless LAN. Network World: whitepaper, 2005.

[Nol93] P. Noll. Wideband speech and audio coding. *IEEE Communication Magazine*, 26:34–44, 1993.

[Ort04] F. Orthman. *Voice over 802.11.* Artech House, 2004.

[OWVR97] M.T. Orchard, Y. Wang, V. Vaishampayan, and A.R. Reibman. Redundancy ratedistortion analysis of multiple description coding using pairwise correlating transforms. In *Proc. IEEE Conf. on Image Proc.*, volume 1, pages 608–611, 1997.

[Oza80] L. Ozarow. On a source-coding problem with two channels and three receivers. *Bell Syst. Tech. Journal*, 59(10):1909–1921, 1980.

[Pan93] D.Y. Pan. Digital Audio Compression. *Digital Technical Journal*, 5(2), 1993.

[Pan95] D. Pan. A tutorial on MPEG audio compression. *IEEE multimedia magazine*, 2(2):60–74, 1995.

[Pas76] R. Pasco. *Source coding algorithm for fast data compression.* PhD thesis, Dept. Elect. Eng., Stanford Univ., CA, U.S.A., 1976.

[PBea03] R. Price, C. Bormann, and J. Christoffersson et al. Signaling Compression (SigComp). RFC 3320, 2003.

[Per96] C.E. Perkins. IP Mobility Support. RFC 2002, 1996.

[Per03] C.E. Perkins. *RTP: Audio and Video for the Internet.* Addison Wesley, 2003.

[PHea05] D. Perels, S. Haene, and P. Luethi et al. ASIC Implementation of MIMO-OFDM Transceiver for 192 Mbps WLANs. Technical Report 13/2005, "Integrated Systems Laboratory, ETH Zürich, Switzerland", 2005.

[PHH98] C. Perkins, O. Hodson, and V. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Media. *IEEE Network Magazine, September/October*, pages 40–48, 1998.

[PKS97] S.H. Park, Y.B. Kim, and Y.S. Seo. Multi-Layer Bit-Sliced Bit Rate Scalable Audio Coding. Presented at the 103rd Convention of the Audio Engineering Society, New York (preprint 4520), 1997.

[PMLA88] W.B. Pennbaker, J.L. Mitchell, G. Langdon, and R.B. Arps. An Overview of the Basic Principles of the Q-Coder Binary Arithmetic Coder. *IBM Journal of Research and Development*, 32(6):717–726, 1988.

[Pol89] D. Pollen. Parametrization of compactly supported wavelets. Technical report, Aware Inc., U.S.A., 1989.

[PPR01] S. Pradhan, R. Puri, and K. Ramchandran. MDS source-channel codes. In *Proc. IEEE Int. Symp. on Inform. Th.*, 2001.

[PPR04] S. Pradhan, R. Puri, and K. Ramchandran. n-channel symmetric multiple descriptions - part I: (n, k) source-channel erasure codes. *IEEE Trans. Inform. Th.*, 50(1):47–61, 2004.

[Pre03] B. Preneel. NESSIE Project Announces Final Selection of Crypto Algorithms. `https://www.cosic.esat.kuleuven.be/nessie`, 2003. Last request: Feb 21, 2006.

[PSM04] M. Petracca, A. Servetti, and J.D. Martin. Voice transmission over 802.11 wireless networks using analysis-by-synthesis packet classification. In *First International Symposium on Control, Communications and Signal Processing*, 2004.

[PV90] I. Pitas and A.N. Venetsanopoulos. *Nonlinear Digital Filters.* Kluwer Academic Publishers, 1990.

[R&00] France Telecom R&D. Study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences: Influence of a recency effect. ITU Study Group 12, Contribution D.139, 2000.

[RBHH01] A. Rix, J. Beerends, M. Hollier, and A. Hekstra. Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[RHA04] M. Raya, J.P. Hubaux, and I. Aad. DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots. In *Proceedings of ACM MobiSys'04*, 2004.

[Ris76] J.J. Rissanen. Generalized Kraft inequality and arithmetic coding. *IBM J. Res. Dev.*, 20:198–203, 1976.

[RMV97] T.S. Rappaport, R. Muhammed, and V.Kapoor. *The Communications Handbook: Propagation Models.* John Wiley, New York, 1997.

[ROea05] A. Riedel, B. Ortelbach, and M. Zibull et al. WiMAX: Marktpotenziale in Deutschland. Mediaconomy Arbeitsbericht 01/2005, Göttingen, 2005.

[ROea06] A. Riedel, B. Ortelbach, and M. Zibull et al. Wirtschaftliche Erfolgspotenziale von WiMAX-Geschäftsmodellen in Deutschland. In *Proceedings of Multikonferenz Wirtschaftsinformatik 2006, Teilkonferenz Mobilität und Mobile Informationssysteme (MMS 2006)*, 2006.

[Ros98] J. Rosenbluth. Testing the Quality of Connections having Time Varying Impairments. Committee T1 Standards Contribution, T1A1.7/98-031, 1998.

[RS78] L.R. Rabiner and R.W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall Signal Processing Series, 1978.

[RS98] L.R. Rabiner and R.W. Schafer. *Rechnernetze*. Oldenbourg-Verlag, 1998.

[RS99] J. Rosenberg and H. Schulzrinne. An RTP Payload Format for Generic Forward Error Correction. RFC 2733, 1999.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889 (Proposed Standard), 1996.

[SCFJ03] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, 2003.

[Sch94] B. Schneier. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In *Proc. of the First Fast Software Encryption workshop in Cambridge*, pages 191–204, 1994.

[Sch96] H. Schulzrinne. RTP Profile for Audio and Video Conferences with Minimal Control. RFC 1890, 1996.

[Sch06] B. Schneier. The Blowfish Encryption Algorithm. `http://www.schneier.com/blowfish.html`, 2006. Last request: Feb 21, 2006.

[Sea05] A. Schliep and B. Georgi et al. `http://www.ghmm.org/`, 2005. Last request: Jan 05, 2006.

[Sel97] I.W. Selesnick. Maple and the Parameterization of Orthogonal Wavelet Bases. `http://taco.poly.edu/selesi/theta2h/`, 1997. Last request: Feb 20, 2006.

[Sha93] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Processing*, 41(12):3445–3462, 1993.

[Sha05] M. Shakouri. The WiMAX Forum - Showcases Equipment and Breadth of Applications, Opens Test Lab. `http://www.wimaxforum.org`, 2005. Last request: Apr 12, 2006.

[SHJD01] K. Svanbro, H. Hannu, L.-E. Jonsson, and M. Degermark. Wireless Real-time IP Services Enabled by Header Compression. In *Proceedings of the IEEE VehicularTechnology Conference (VTC)*, volume 2, pages 1150–1154, 2001.

[Sim94] W. Simpson. The Point-to-Point Protocol (PPP). RFC 1661, 1994.

[Sim97] W. Simpson. PPP Vendor Extensions. RFC 2153, 1997.

[SLMC94] K. Sklower, B. Lloyd, G. McGregor, and D. Carr. The PPP Multilink Protocol (MP). RFC 1717, 1994.

[SM03] A. Servetti and J.D. Martin. Adaptive interactive speech transmission over 802.11 wireless LANs. In *Proc. IEEE Int. Workshop on DSP in mobile and Vehicular Systems*, 2003.

[SM04] A. Servetti and J.D. Martin. Link-level unequal error detection for speech transmission over 802.11 networks. In *Proc. Special Workshop in Maui (SWIM)*, 2004.

[SM05] A. Servetti and J.D. Martin. 802.11 MAC Protocol with Selective Error Detection for Speech Transmission. M. Ajmone Marsan et al. (Eds.): QoS-IP 2005, LNCS 3375, pp. 509–519, 2005.

[SN04] S. Sanayei and A. Nosratinia. Antenna Selection in MIMO Systems. *IEEE Communications Magazine, October 2004*, pages 68–73, 2004.

[SO99] R. Singh and A. Ortega. Erasure recovery in predictive coding environments using multiple description coding. In *Proc. IEEE Multimedia Signal Processing Workshop*, pages 333–338, 1999.

[SP96] A. Said and W.A. Pearlman. A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology*, 6(3):243–250, 1996.

[Spa94] A. Spanias. Speech Coding: A tutorial review. In *Proceedings of the IEEE*, volume 82(10), pages 1541–1582, 1994.

[SR98] H. Schulzrinne and J. Rosenberg. A Comparison of SIP and H.323 for Internet Telephony. In *Proceedings of The 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, pages 83–86, 1998.

[SRL98] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). RFC 2326, 1998.

[SRVN98] S.D. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt. Multiple-description wavelet based image coding. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 659–663, 1998.

[ST89] J. Suzuki and M. Taka. Missing packet recovery techniques for low-bit-rate coded speech. *IEEE Journal on Selected Areas in Communications*, 7(5):707–717, 1989.

[ST93] D. Sinha and A. Tewfik. Low bit rate transparent audio compression using adapted wavelets. *IEEE Trans. Signal Processing*, 41(12):3463–3479, 1993.

[Sta96] Standardization Sector of ITU. Coding of speech at 8 Kbps using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP), 1996.

[Sta03a] Standardization Sector of ITU. ITU-T Recommend. G.114, 2003.

[Sta03b] Standardization Sector of ITU. ITU-T Recommend. H.235.0 - H.235.9, 2003.

[Sta03c] Standardization Sector of ITU. ITU-T Recommendation P.862.1, 2003.

[Sta05] Standardization Sector of ITU. ITU-T Recommendation P.862.2, Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs, 2005.

[Sto88] J.A. Storer. *Data Compression Methods and Theory*. Computer Science Press, 1988.

[SVS99] S.D. Servetto, V.A. Vaishampayan, and N.J.A. Solane. Multiple description lattice vector quantization. In *Proc. IEEE Data Compression Conf.*, pages 3–12, 1999.

[SZM05] S. Strahl, Huan Zhou, and Alfred Mertins. An adaptive tree-based progressive audio compression scheme. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.

[Tel04] Inc. TeleGeography. TeleGeography 2005. TeleGeography Research, PriMetrica Inc, `www.telegeography.com`, 2004.

[TPB97] T. Turletti, S.F. Parisis, and J. Bolot. Experiments with a Layered Transmission Scheme over the Internet. INRIA Research Report no 3296, 1997.

[Tra90] H. Traunmüller. Analytical expressions for the tonotopic sensory scale. *J. Acoust. Soc. Am.*, 88:97–100, 1990.

[Tuc93] B. Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Technical Journal*, pages 27–37, 1993.

[Tuk71] J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1971.

[TYYA89] Y. Takahashi, H. Yazawa, K. Yamamoto, and T. Anazawa. Study and Evaluation of a New Method of ADPCM Encoding. Preprint 2813, 86th Audio Engineering Society Convention, 1989.

[Vai93] V.A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39(3):821–834, 1993.

[Val03] J.M. Valin. The Speex Codec Manual. `http://www.speex.org/manual2/manual.html`, 2003. Last request: Feb 02, 2006.

[Vas96] S.V. Vaseghi. *Advanced Signal Processing and Digital Noise Reduction*. Wiley, 1996.

[Vau06] S. Vaudenay. *A Classical Introduction to Cryptography - Applications for Communications Security*. Springer, 2006.

[VBC98] V.A. Vaishampayan, J.-C. Batllo, and A. Calderbank. On reducing granular distortion in multiple description quantization. In *Proc. IEEE Int. Symp. on Inform. Th.*, page 98, 1998.

[VCM01] M. Veeraraghavan, N. Cocker, and T. Moors. Support of voice services in IEEE 802.11 wireless LANs. In *Proceedings of INFOCOM '01*, pages 488–496, 2001.

[VD94] V.A. Vaishampayan and J. Domaszewicz. Design of entropy constrained multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 40(1):245–250, 1994.

[WCH+02] M.A. West, L.W. Conroy, R.E. Hancock, R. Price, and A.H. Surtees. IP header and signalling compression for 3G systems. In *Proc. of 3G Mobile Communication Technologies*, pages 102–106, 2002.

[Wic00] M. V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. AK Peters, Ltd., 2000.

[Wie49] N. Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. MIT Press, 1949.

[Wie05] M. Wienecke. Konzeption und Implementierung eines Fehlergenerators zur Simulation drahtloser IP-basierter Netzwerke. Bachelor's thesis, Georg-August Universität Göttingen, 2005.

[Wil04] J. M. Wilson. The Next Generation of Wireless LAN Emerges with 802.11n. Technology@Intel Magazine, August, 2004, 2004. Last request: Sep 28, 2005.

[WO01] X. Wang and M.T. Orchard. Multiple description coding using trellis coded quantization. *IEEE Trans. Image Proc.*, pages 391–394, 2001.

[Won05] L.C. Wong. An Overview of 802.11 Wireless Network Security Standards and Mechanisms. SANS Institute, 2005.

[WSC+02] B. Wang, H. Schwefel, K. Chua, R. Kutka, and C. Schmidt. On implementation and improvement of robust header compression in UMTS. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1151–1155, 2002.

[WV97] R.A. Wannamaker and E.R. Vrscay. Fractal Wavelet Compression of Audio Signal. *J. Audio Eng. Soc.*, 45(7/8):540–553, 1997.

[WWP88] R.M. Warren, J.M. Wrightson, and J. Puretz. Illusory continuity of tonal and infratonal periodic sounds. *Journal of the Acoustical Society of America*, 84(4):1338–1342, 1988.

[WWZ80] J.K. Wolf, A.D. Wyner, and J. Ziv. Source coding for multiple descriptions. *Bell Syst. Tech. Journal*, 59(8):1417–1426, 1980.

[YB94] R. Yavatkar and N. Bhagwat. Improving End-to-End Performance of TCP over Mobile Internetworks. In *Mobile '94 Workshop on Mobile Computing Systems and Applications*, 1994.

[YOVR01] W. Yang, M.T. Orchard, V. Vaishampayan, and A.R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Transactions on Image Processing*, 10(3):351–365, 2001.

[YR98] X. Yang and K. Ramchandran. Optimal multiple description subband coding. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 654–658, 1998.

[Zib02] M. Zibull. Digitale Wasserzeichen. Diploma thesis, TU Clausthal, 2002.

[ZRH05] M. Zibull, A. Riedel, and D. Hogrefe. Voice over Wireless LAN - A Fine-Scalable Channel-Adaptive Speech Coding Scheme. In *The Third ACM*

International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH'05), pages 111–114, 2005.

Curriculum Vitae

Marco Zibull

Persönliche Daten

Geburt	28. Juni 1976 in Kiel
Staatsangehörigkeit	deutsch

Wissenschaftlicher Werdegang

08/1982 - 06/1986	Grundschule Altenholz-Stift
08/1986 - 06/1993	Realschule Altenholz (Abschluß: Mittlere Reife)
08/1993 - 06/1996	Fachgymnasium - Technik - (BST) Kiel (Abschluß: Allgemeine Hochschulreife)
10/1997 - 11/2002	Informatik-Studium an der TU Clausthal (Abschluß: Diplom-Informatiker (Dipl.-Inf.))
seit 12/2002	Wissenschaftlicher Mitarbeiter am Institut für Informatik der Georg-August-Universität zu Göttingen

---