Ein Einblick in digitale Kompressionsverfahren für Audiodaten

Audio CoDecs

Bernd Schelling
Mat.-Nr.: 812133

Fachhochschule Furtwangen,
Studiengang Communication Engineering / Computer Networking

April 1999

Inhalt

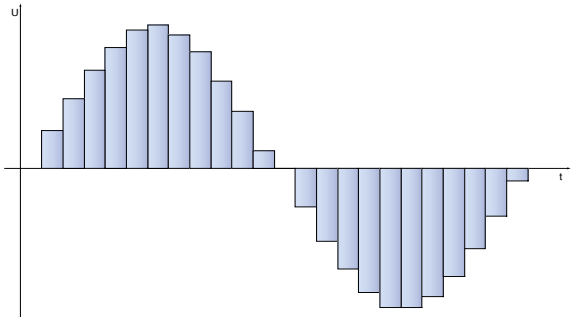
1. Einleitung

Digitale Netze werden immer häufiger für die Übertragung von nicht-textbasierten Daten verwendet. Die verfügbare Bandbreite wächst jedoch langsamer, als der Bedarf danach. Beispielsweise beträgt die Datenrate von CD-Audio über 1,3 Mbps, eine Videosequenz in PAL-TV-Qualität benötigt sogar fast 170 Mbps. Eine ISDN-Leitung bietet jedoch nur 128 kbps. Um solche Datenmengen mit vernünftigem Aufwand übertragen zu können, muß man Verfahren zur Reduktion dieser Datenmengen finden.

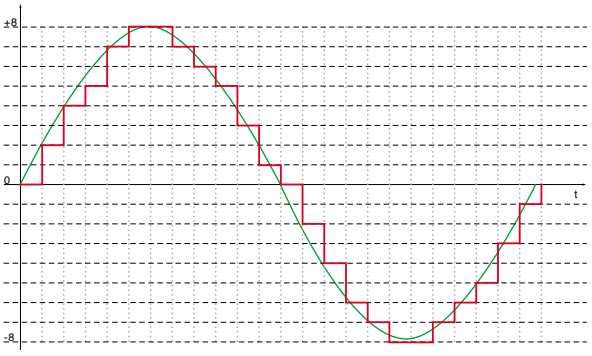
Hier sollen einige zugrundeliegende Konzepte und Datenreduktionsverfahren - insbesondere für Audiodaten - erläutert werden.

1.1 benötigte Bandbreite für unkomprimierte Tondaten

Um eine Schwingung mit beispielsweise 1kHz zu digitalisieren, benötigt man mindestens 2.000 Abtastungen in pro Sekunde, da eine **Schwingung** durch mindestens **2 Punkte** (Hoch- und Tiefpunkt)

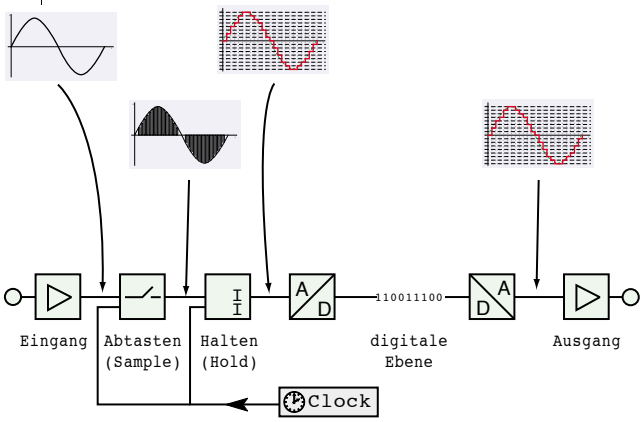


definiert werden kann. Frequenzen oberhalb der Hälfte der Abtastrate (= Abtastungen pro Sekunde) müssen ausgefiltert werden - sie können nicht dargestellt werden. Dies wird auch das 'Nyquist-Theorem' genannt. Das menschliche Ohr ist in der Lage, Frequenzen bis etwa 17kHz zu hören sowie weitaus höhere wahrzunehmen.



Deshalb hat sich eine **Abtastrate von 44,1 kHz** als Standard für digital Audio durchgesetzt.

Desweiteren muß man sich Gedanken machen, mit welcher Genauigkeit (Wortbreite) diese einzelnen Werte (Samples) gespeichert werden sollen. Je genauer die Werte gespeichert werden, desto feiner läßt sich der Unterschied zwischen lautestem und



leisestem Signal abbilden. Man nennt dies den **Dynamikumfang**.

Durch die Wandlung in ein wertdiskretes Signal (**Quantisierung**) müssen (kontinuierliche) Meßwerte, die zwischen zwei zulässigen (diskreten) Werten liegen, auf- oder abgerundet werden. Der daraus resultierende Fehler (Quantisierungsfehler) ist bei natürlichen, nichtdeterministischen Signalen wie Musik oder Sprache statistischer Natur und macht sich daher als gleichmäßiges **Rauschen** bemerkbar. Da es nur im Zusammenhang mit dem Nutzsignal auftritt (kein Quantisierungsrauschen bei Null-Pegel -> kein hörbares Rauschen) und ansonsten weitgehend vom (meist lauteren) Nutzsignal verdeckt wird, fällt das Quantisierungsrauschen nicht weiter auf. Problematisch sind jedoch leise Nutzsignale, da hier das Rauschen störend in den Vordergrund treten kann, bzw. im Extremfall das Nutzsignal vollständig verdeckt.

Das Quantisierungsrauschen verhält sich proportional zur Auflösung, wobei für jedes Bit das man in der Auflösung wegläßt (Halbierung der Auflösung), 6 dB Rauschen entsteht (was einer Verdopplung der Rauschleistung entspricht). Das bedeutet umgekehrt, das **für jedes Bit mehr 6 dB weniger (verdeckendes) Rauschen, und damit 6 dB mehr nutzbare (unverdkckte) Dynamik** zur Verfügung steht. Damit ergibt sich bei (den zur Zeit üblichen) 16 Bit Auflösung ein Dynamikumfang von etwa 96 dB.

Dieser scheinbar überlegene Wert ist jedoch nicht mit der Dynamik analoger Systeme vergleichbar, da hier nicht der effektive, sondern der Spitzen-Störgeräuschpegel bewertet wird (DIN 45405). Mit

diesem Meßverfahren ergibt sich eine Anhebung des Störgeräuschs um 14 dB. Die Aussteuerung digitaler Systeme ist nach unten begrenzt, weil sich sonst bei leisen Passagen das Quantisierungsrauschen störend bemerkbar macht. Man setzt hierfür einen Sicherheitsabstand (Footroom) von 20 dB an. Weiterhin benötigt man eine Übersteuerungs-Reserve (Headroom) von ca.10 dB, da selbst leichte Übersteuerungen (im Gegensatz zu analogen Systemen) zu unangenehmen Verzerrungen führen. Es bleiben also noch 98 dB - 14 dB - 20 dB - 10 dB = 54 dB, was (bei besserem Störabstand) etwa der Nutzdynamik analoger Systeme entspricht.

Für Sprachqualität sollten mindestens 4 Bits bei 8.000 Abtastungen/s gespeichert werden. Eine herkömmliche Audio-CD enthält Daten mit 16 Bits und 44,1 kHz Abtastrate. Dies ist ausreichend, um das gesamte hörbare Spektrum mit ausreichender Dynamik darzustellen. Trotzdem hat es sich erwiesen, daß weder 16 Bits noch 44,1 kHz ausreichen, um alle tonalen Details aufzuzeichnen. Mittlerweile strebt man 24 Bits bei 96 kHz für DVD-Audio (HDCD) an.

Führt man sich diese Datenmengen vor Augen erkennt man schnell, daß die Bandbreiten heutiger Telekommunikationsnetze (z.B. 128 kBit/s bei ISDN gegenüber 1,4 MBit/s für CD-Audio, s.o.) schnell erschöpft sind.

1.2 Was ist ein CoDec?

Für die Komprimierung von Datenströmenmuß jeweils ein Mechanismus vorhanden sein, der die Signale in beide Richtungen umwandelt. Kodieren und Dekodieren - CoDec steht für ‘coder/decoder’.

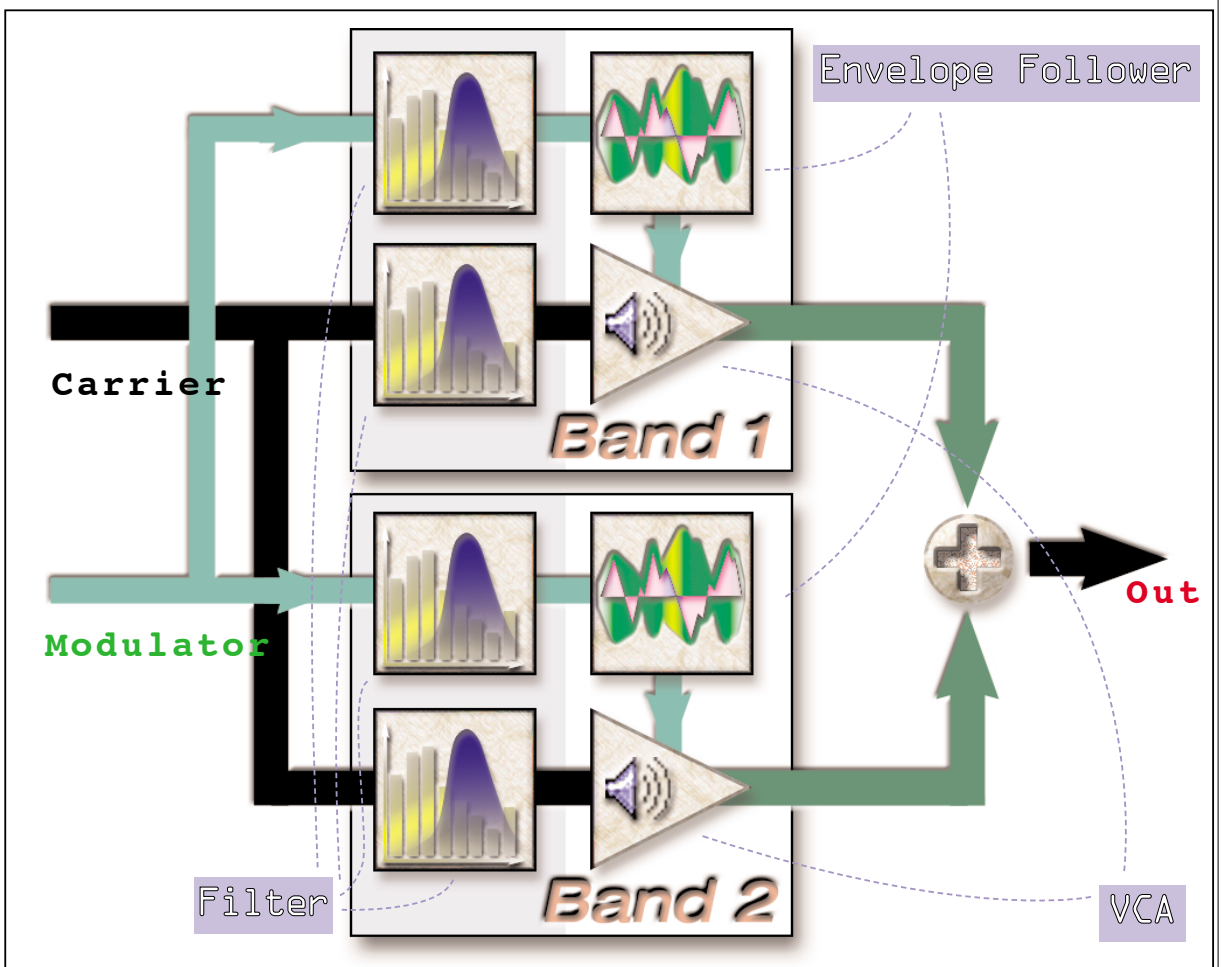
Diese lassen sich in Bezug auf den Dateninhalt zwei Kategorien unterteilen:

- verlustbehaftet
- nicht verlustbehaftet

Streng genommen handelt es sich bei der Digitalisierung von Audiosignalen oder analoge Aufzeichnung auf Magnetband auch um CoDecs.

1.3 analoge Verfahren

Auf analoge Datenreduktionsverfahren soll hier nicht näher eingegangen werden. Anbei nur ein



Vocoder (Schema mit 2 Bändern)

Beispiel, das konzeptionell auch in digitalen Kompressionsverfahren wie z.B. MPEG-Audio verwendet wird.

Der Vocoder

Ursprünglich suchte man zu militärischen Zwecken nach einem Verfahren, Sprachübertragungen per Funk abhörsicher zu machen. In den Bell Laboratories wurde der Vocoder während des zweiten Weltkrieges entwickelt.

Das **Nutzsignal (Modulator)** wird in einzelne (z.B. 10) Frequenzbänder unterteilt. Die Amplitude jedes dieser Bänder wird dann übertragen. Auf der Gegenseite filtern genau gleich viele (10) Bandpaß-Filter, in ihrer Stärke gesteuert von dem übermittelten Signal, aus einem obertonreichen **Trägersignal (Carrier)** entsprechende Bänder aus. Dieser Effekt erzeugt verständliche Sprache, die allerdings sehr roboterähnlich klingt. Die Musikgruppe Kraftwerk hat diesen Effekt um 1970 bekannt gemacht.

1.4 digitale Verfahren

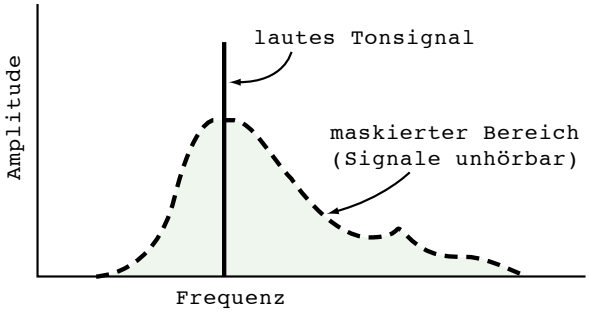
Liegen die Daten erst einmal auf digitaler Ebene vor, lassen sich mittels Algorithmen mathematische Verfahren anwenden, um die Daten 'platzsparender' umzurechnen. Für Computeranwendungen lautet das typische Ziel eines Codecs, Kodierseitig möglichst hohe Kompressionsraten zu erzielen und Dekodierseitig möglichst wenig Rechenaufwand zu benötigen, damit die Daten schnell abgerufen werden können bzw. auch weniger leistungsfähige Rechner diese Formate abspielen können. Für Echtzeit- und Embed-Anwendungen (also Hardwarebasiert) kann jedoch auch ein ausgewogeneres Verfahren wünschenswert sein. Sinnvoll beispielsweise bei digitalen Mobilfunktelefonen.

Verlustfreie Verfahren bedienen sich normalerweise komplexer Algorithmen oder Tabellen, die die Signalverläufe mittels einer geringeren Menge an Variablen darstellen können.

Bei **verlustbehafteten Verfahren** spielen neben diesen datenreduzierenden Algorithmen und Beschränkungen der Auflösung auch psychoakustische und physikalische Effekte eine Rolle.

Die **Psychoakustik** beschäftigt sich mit Zusammenhängen zwischen Schallereignis (Reiz) und Hörereignis (Empfindung), also Wahrnehmung durch das menschliche Ohr. Ein Reiz wird vom Gehör nicht so empfunden, wie er ihm dargeboten wird. Die Lautstärke-Wahrnehmung zum Beispiel ist (unter anderem) sehr stark frequenzabhängig.

Ein weiteres Phänomen ist der sogenannte **Verdeckungseffekt**. Werden dem Ohr zwei ähnlich (Tonhöhe, Spektrum) klingende Geräusche dargeboten, so wird das leisere Geräusch vom Lauteren verdeckt, d.h. das Ohr nimmt das leisere Geräusch nur teilweise bzw. überhaupt nicht wahr. Aus dem Alltag ist dieser Effekt gut bekannt: Für ein Gespräch in einem vollbesetztem Café, in dem sich alle Anwesenden angeregt unterhalten, muß man die Stimme schon etwas anheben, um vom Gesprächspartner verstanden zu werden (Anderenfalls geht die Stimme im



Hintergrundgeräusch unter). Im Gegensatz dazu genügt in einer Bibliothek ein leiser Flüsterton. Der Verdeckungseffekt ist von mehreren Dingen abhängig. Allgemein gilt, daß ein höherfrequenter Schall einen tieferfrequenten Schall nur dann verdeckt, wenn der Frequenzabstand gering ist. Ein Schall tieferer Frequenz verdeckt einen höherfrequenten Schall nur dann, wenn der tieffrequente Schall vergleichsweise große Intensität besitzt. Desweiteren spielt die Art des Schalls eine Rolle: Tonale (schmalbandige, klanghafte) Signale sind stärker verdeckend als geräuschhafte (breitbandige) Signale. Verdeckungserscheinungen gibt es nicht nur für gleichzeitige Signale, sondern auch für zeitlich aufeinanderfolgende Signale. Die **Nachverdeckung** läßt Signale, die auch bei simultaner Darbietung verdeckt werden, nach abschalten des verdeckenden Signals eine kürzere Zeit unhörbar bleiben. Die Zeitspanne der Nachverdeckung hängt sehr von den Signalarten und -dauern ab; sie liegt im Bereich einiger ms bis einiger 10ms. Auch eine **Vorverdeckung** für bis etwa 20ms voreilende Signale kann beobachtet werden.

Derartige, für das Hörempfinden irrelevante Signale lassen sich herausfiltern und müssen dann nicht verarbeitet werden.

2. online/offline

2.1 Einsatzgebiete für komprimierte Audiodaten

Für die professionelle Produktion von Ton sind im Allgemeinen Kompressionsverfahren weniger sinnvoll, da auf die Daten möglichst schnell und einfach zugegriffen werden muß. Außerdem muß der Anwender dazu in der Lage sein, beliebige Punkte innerhalb der Aufzeichnungen anzufahren. Wenn beispielsweise ein Live-Mitschnitt ausgebessert werden soll - z.B. eine Pause herausschneiden - muß diese angefahren werden und der entsprechende Bereich gelöscht werden, ohne die Integrität der Datei zu gefährden. Da aber heutige Kompressionsverfahren meist Paketbasiert und mit zusätzlichen Quersummen und Lookup-Tables ausgestattet sind, fällt eine genaue Positionierung und vor Allem eine Bearbeitung der Daten schwer.

Trotz Allem gibt es Verfahren, die auch das ermöglichen. Diese finden im Moment aber keinen nennenswerten Einsatz. Dies mag unter Anderem damit zusammenhängen, daß für die Komprimierung und Dekomprimierung Prozessorzeit benötigt wird, die natürlich den Bearbeitungsprozeß merklich verlangsamt.

Zur **Archivierung** von Tonmaterial werden gerne verlustfreie Verfahren angewendet, da diese den benötigten Speicherplatz reduzieren. Die Daten müssen ohnehin auf Backupmedien (DAT-Bänder/Streamer-Bänder/CD-Rs) gesichert werden, was bedeutet, daß nicht sofort darauf zugegriffen werden kann (zumindest muß man die entsprechenden Datenträger aus dem Archiv holen) und sich somit ein Dekomprimieren mit in den Arbeitsablauf integrieren lassen kann.

Standardverfahren für Computerdaten, wie ZIP/Huffman-Komprimierung eignen sich hauptsächlich für Programmcode o.ä. und bieten auf Audiodaten angewandt nur geringe Einsparungen (etwa 10%) gegenüber speziell für Audiodaten angepaßte Verfahren (ca. 40%).

Verlustbehaftete Verfahren eignen sich selbstverständlich auch für Archivierungszwecke. Man sollte aber bedenken, daß jede Bearbeitung eine Dekompression und erneute, verlustbehaftete, Kompression erfordert. Einsatzgebiete sind beispielsweise Radiosendungen oder Sprachaufnahmen. Bei diesen ist die Tonqualität und eventueller Verlust bei Rekompression nicht ganz so wichtig, da die Wahrscheinlichkeit, daß man diese Daten für spätere Bearbeitungen benötigt, eher gering ist.

Die Gruppe dieser Verfahren bezeichnet man auch als **offline-Verfahren, da die Komprimierung und Speicherung nicht zwangsläufig in Echtzeit geschieht.**

2.2 Echtzeitverfahren

Als **online-Verfahren werden jene bezeichnet, die direkt während der Aufzeichnung** greifen. Als prominentes Beispiel sei hier die Übertragung von Sprache über digitale Funktelefonnetze mittels GSM zu nennen.

Hauptanwendungsgebiet sind Live-Übertragungen aller Art. Also: Telefon, Reportagen, Chat, Internet-Telefonie etc. Außerdem verwenden digitale Consumer-Aufnahmegeräte derartige CoDecs. Zu nennen wäre hier Sonys ATRAC-Verfahren, das in deren MiniDisc-Systemen Verwendung findet. Auch bei digital-Video werden häufig derartige Mechanismen eingesetzt.

Die meisten heutigen Verfahren sind für Sprachgebrauch optimiert, da die Prozessoren nicht genügend Leistung bieten, um Echtzeitkompression in höherer Qualität bereitzustellen. Zumindest erfreuen sich solche Verfahren momentan noch keines breiten Einsatzes in Standardanwendungen.

3. Formatieren von Audiodaten

Wie bereits unter 2.1 angesprochen, lassen sich Formate auch in der Art unterscheiden, wie die Daten abgelegt werden. Ton ist als ein zeitdiskretes, kontinuierliches Signal zu verstehen. Und hier sollen Formatierungen für beliebige Toninformationen dargestellt werden. Am Rande sei trotzdem zu erwähnen, daß es auch andere Formatierungskonzepte gibt - insbesondere ‘SA’, structured Audio. Hierbei handelt es sich um ein Format, das mehrere Ereignisse - nach einer bestimmten Vorschrift abgespielt - enthält. Vereinfacht gesagt ähnelt SA etwa MIDI-Files, bei dem die Instrumente mitenthalten sind bzw. einem selbstspielenden Pianoautomaten, bei dem die Partitur auf einer Papierrolle gespeichert ist und die Generierung in Echtzeit erfolgt.

Bei herkömmlichem Ton (z.B. Konzertmitschnitt) macht eine Unterteilung - im Gegensatz zu SA - in einzelne 'forks', also ein hierarchisches 'Steuerung<->Ereignis'-Konzept, wenig Sinn.

Es lassen sich drei grundsätzliche Konzepte für die Formatierung von digitalem Ton festlegen:

Streaming, packeting und broadcasting - wobei broadcasting die Mechanismen des packeting verwendet und packeting die des streaming. Jedes dieser Formate läßt sich mit verlustfreien und/oder verlustbehafteten Kompressionsverfahren erweitern.

3.1 streaming

Der Datenstrom wird in einer vereinbarten, kontinuierlichen Form von Bitdaten übertragen. Beispielsweise überträgt man ein Mono-Signal mit 8 Bits Genauigkeit einfach als eine Folge von Bytes.

Vorteil: einfach zu realisieren, einfach zu verändern (Effektberechnung / Modulation), Daten lassen sich ohne Umwandlung des Datenstroms im D/A-Wandler in Analogsignale zusücksetzen.

Nachteil: unflexibel, schwer erweiterbar (mehrere Kanäle, höhere Bit-Tiefen etc.), Übertragungskanal kann nicht zusätzlich für andere Daten verwendet werden

3.2 packeting

Um mehrere Kanäle oder größere Bit-Tiefen übertragen zu können, muß das Signal in einzelne Pakete unterteilt werden, die alle einer Vorschrift folgen.

Ein Beispiel für 16 Bits, 2 Kanäle (Stereo Links und Rechts):

- die 16 Bits werden in 2 Bytes High-Endian unterteilt (das bedeutet, daß zuerst die 8 letzten Stellen des 16-Bit-Wortes, dann die ersten 8 Stellen übertragen werden)
- Jeder Kanal überträgt 128 Samples. Dann findet ein Kanalwechsel statt (Links und Rechts wechseln sich ab)
- Alle 16 Kanalwechsel wird ein Header mit Prüfsummen und Format-Informationen (z.B. timecode) mit übertragen

Vorteil: Übertragungskanal kann für verschiedene Daten mitverwendet werden. Denkbar wäre z.B. ein Transskript einer Reportage oder die Unterlegung von Videobildern mit Ton. Es besteht die Möglichkeit, Prüfsummen zu bilden.

Nachteil: Daten müssen in einen Puffer zwischengespeichert und 'neu zusammengesetzt' werden. Eingeschränkter direkter Zugriff (abhängig von der Art des Formates - Paketgröße abhängig bzw. unabhängig vom letzten Paket)

3.3 broadcasting

Beim broadcasting wird der Datenstrom von vielen verschiedenen Clients gleichzeitig empfangen. Da die Leitungsqualität (Bandbreite, Fehlerrate) sich von Client zu Client unterscheiden kann, muß ein hier verwendetes Format verschiedene, aufeinander aufbauende 'Qualitätsebenen' übertragen.

Beispiel 1: Übertragung einer Radiosendung über das Internet

Der Client nimmt vor dem Empfang eine Analyse der verfügbaren Bandbreite zum Server vor und teilt ihm seine Messung mit. Daraufhin wählt der Server einen passenden Datenstrom aus (unter verschiedenen Bandbreiten - z.B. 14.4, 28.8 und 56 kBit/s) und übermittelt die Übertragungsinformationen an den Client.

Beispiel 2: Übertragung mit maximaler Qualität 16 Bits, 44,1 kHz, Stereo

Hier erfolgt eine Übertragung des Signals in mehreren Stufen, die voneinander abhängig sind:

- Eine gesicherte Übertragung der Basisdaten (8 Bits, 11,025 kHz, Stereo)
  Jeder Client muß, um am Broadcast teilnehmen zu können, diese Daten erfolgreich empfangen (Minimalanforderung).
- ungesicherte Übertragung der restlichen Daten in mehreren Stufen (nur die der darunterliegenden Stufe fehlenden Daten werden übertragen). Zum Beispiel 3 Stufen:
  1.) erhöhte Samplerate: 22,05 kHz (Qualität hier: 8 Bits, 22,05 kHz, Stereo)
  2.) erhöhte Wortbreite: 16 Bits (Qualität hier: 16 Bits, 22005 kHz, Stereo)
  3.) erhöhte Samplerate: 44,1 kHz (Qualität hier: 16 Bits, 44,1 kHz, Stereo)

Somit ist sichergestellt, daß jeder Client eine 'Grundversorgung' erhält. Läßt die Leitungsqualität es zu, wird das Signal entsprechend feiner aufgelöst.

4. Arbeitsweise verschiedener CoDecs

4.1 raw data

Das grundlegenste 'Kompressionsverfahren' ist selbstverständlich die **angemessene Wahl der Bandbreite und Auflösung** der einzelnen Werte.

4.1.1 Verringerung der Nutzsignal-Bandbreite

Das menschliche Ohr hört Frequenzen bis etwa 15 kHz und nimmt Frequenzen bis ca. 20 kHz bewußt wahr. Am Wichtigsten sind jedoch Frequenzen von ca. 600 Hz bis 2,5 kHz, Frequenzen unterhalb von etwa 150 Hz sind für das menschliche Ohr nicht räumlich ortbar.

Enthält das zu übertragende Signal z.B. nur Sprache, lassen sich ohne hörbare Qualitätsverluste alle Frequenzen oberhalb etwa 10 kHz ausfiltern. Wenn es nur darum geht, daß die Sprache verständlich ist, kann man sogar Frequenzen oberhalb 2.5-4 kHz ausfiltern. Dadurch läßt sich die Samplerate von ursprünglich über 40 kHz (volle Bandbreite) auf 8 kHz reduzieren.

4.1.2 nichlineare Dynamik

Leise Signale (natürlich oberhalb der Hörschwelle) werden vom Ohr empfindlicher wahrgenommen als laute. Um dieser Charakteristik Rechenschaft zu tragen kann man anstatt der heute üblichen Wandlung in 16 Bit tiefe Worte, bei der die Werte linear verteilt sind (ein Bit entspricht stets 6 dB), eine logarithmische Verteilung der Werte vornehmen. Große Amplitudenwerte werden gröber als kleine aufgezeichnet. **Eine logarithmische Skala mit 12 Bits ist in Etwa vergleichbar mit 16 Bits linear.**

4.2 PWM (Pulse Width Modulation)

Hierbei wird der Datenstrom nicht in Worten mit einer definierten Wortbreite übertragen (z.B. 16 Bits), sondern mit einem einzigen Bit. Dafür wird die Samplerate erhöht (Oversampling). Ein Sample mit 44,1 kHz und 16 Bit wird dann z.B. bei 16-fach Oversampling zu 16 Samples mit 705,6 kHz. Jedes einzelne Sample wird von 16 auf 1 Bit konvertiert, quasi in viele kleine Informationen 'lauter' oder 'leiser' verwandelt. 'Signallevel halten' drückt sich dann in ständig 'lauter' 'leiser' abwechselnden Bits aus deren Samplingrate weit über der Shannon-Frequenz [das ist die höchste zu übertragende Frequenz mit 2 multipliziert] liegt.

Man kann nun, ähnlich wie bei der unter *4.1.2* besprochenen veränderten Relation zwischen Amplitude und Bitstrom, eine andere Kennlinie anlegen. Dadurch läßt sich die Samplingrate des Bitstroms wieder verringern.

4.3 ADPCM (Adaptive Differential PCM)

Das adaptive PCM-Verfahren arbeitet, ähnlich wie PWM unter *4.2*, nicht mit absoluten Amplitudenwerten, sondern mit Unterschieden zwischen zwei Amplitudenwerten bzw. Samples. Es wird also die **Steigung des Signals, nicht das Signal selbst** aufgezeichnet.

Da bei Audiosignalen sehr große Dynamiksprünge zwischen zwei Samples sehr viel seltener vorkommen als Kleine, macht man sich beim ADPCM genau diese Eigenschaft zunutze.

Man beachte, daß (statistisch gesehen) **tieffrequente Signale** (= langsame Veränderung der Amplitude) **mehr Schallenergie** (höherer Amplituden-Spitzenwert) haben, als **hochfrequente**. Diese ändern ihre Amplitude sehr schnell, aber haben nur **geringe Schallenergie**, verursachen also kleinere Veränderungen der Amplitude als tieffrequente Signale.

Um der Häufigkeit verschiedener Dynamiksprünge Rechenschaft zu tragen, wird hier eine **logarithmische Kennlinie** zugrunde gelegt (vgl. 4.1.2). Das bedeutet, daß größere Dynamiksprünge gröber aufgelöst werden als Kleinere. Es stehen wenige Bits für große Sprünge und viele Bits für kleine Sprünge zur Verfügung.

Im Gegensatz zu herkömmlichen PCM-Signalen wird also zuerst eine logarithmische Kennlinie angelegt, danach wird die Steigung des Signalverlaufs - NICHT dessen zeitdiskreter Absolutwert - gespeichert.

Audiomaterial in CD-Qualität läßt sich mit diesem Verfahren ohne merklichen Verlust **von 16 Bits/Sample auf 4 Bits/Sample** reduzieren. **Weder große Pufferspeicher noch nennenswert großer Rechenaufwand sind hierzu erforderlich.**

Nachteilig ist, daß aufgrund der Tatsache, daß jedes Sample sich auf das Vorhergehende bezieht, eine **beliebige Positionierung** im Stream **nicht möglich** ist. Eine mögliche Abhilfe wäre hier, Pakete zu schaffen, die ihren Startsamplewert speichern und danach nur

noch Pegeländerungen. 50 Pakete pro Sekunde dürften für Audioanwendungen mehr als ausreichend sein. Dies ergäbe pro Kanal ein Paket mit einem diskreten Samplewert und ca. 1.000 Offsets bei CD-Samplerate. Von ca. 2 Kilobytes pro Paket kommt man durch die Anwendung von ADPCM auf ca. 512 Bytes.

4.3.1 DVI

DVI ist ein geschwindigkeitsoptimiertes ADPCM-Kompressionsverfahren mit einer Kompressionsrate von 4:1.

4.3.2 IMA

'IMA' steht für Interactive Multimedia Association, der unter anderen Apple und Intel angehören. Der IMA-Algorithmus ist eine Weiterführung von DVI und komprimiert ausschließlich 16 Bit breite Samples mit Raten von 4:1 oder 2:1.

4.3.3 AIFF-C

Eins der Formate von AIFF-C ist IMA-komprimiertes ADPCM. Unterstützt werden 16-Bit-Samples mit beliebigen Sampleraten bei einer Kompressionsrate von wahlweise 4:1 oder 2:1.

4.4 RealAudio [.ra/.ram]

RealAudio unterscheidet sich von den anderen Formaten zuerst einmal in einer Hinsicht grundlegend: es wurde von Anfang an als Internet-Broadcasting-Format konzipiert, also gibt es sowohl Codierer als auch Player, die in Echtzeit arbeiten.

Wie der Name RealAudio schon sagt, wird hier der Sound in **(Fast)Echtzeit** abgespielt. Ein kurzer Klick in einem WWW-Browser auf den zugehörigen Link genügt: Der Real-Audio-Player wird automatisch gestartet und schon kann man den Sportnachrichten oder sonstigen Klängen lauschen.

Die **Ausgabequalität paßt sich automatisch der verfügbaren Netzleistung an**. Bei Kompressionsraten von ca. **20:1 erreicht man** in etwa die **Qualität eines Mittelwellen-Radios**.

Da es sich bei RealAudio um ein Broadcasting-Format handelt, muß man natürlich auch das Verhalten betrachten, wenn Pakete ausfallen. **RealAudio übermittelt Daten über UDP, also mit einer ungesicherten Verbindung. Fällt ein Paket aus, wird es interpoliert. Bei einer Ausfallrate von 15% sind die Verluste somit schon deutlich hörbar.** Mick Jagger klingt dann schon ein klein wenig nach Marsrauschen.

4.5 MPEG I/II Audio

Das wohl derzeit bekannteste digitale Audio-Format überhaupt dürfte 'mp3' sein. Es steht für MPEG Audio Layer III kodierung.

Die MPEG [Moving Picture Experts Group, unter Anweisung der ISO und der IEC] wollte für digitale Videoanwendungen (DVB - Digital Video Broadcast, DVD - Digital Video Disc, digitales Kino) einen Standard für Kompressionsverfahren verabschieden. Da die unkomprimierte Bandbreite für Video extrem hoch ist (z.B.: PAL-Fernsehbild ohne Ton ca. 250 MBit/s!, Stereo-Ton ca. 1.5 MBit/s), wurden entsprechend effiziente und qualitativ befriedigende Datenreduktionsverfahren benötigt.

Die MPEG hat bis jetzt drei Verfahren verabschiedet, Layer I, II und III. Diese werden in den MPEG-Videoformaten 1 und 2 benutzt. Ausgeschrieben heißt das dann z.B.: 'MPEG 2 Audio Layer III'.

Alle drei Layer arbeiten **Verlustbehaftet mit schmalbandigen adaptiven Verfahren**, teilweise auch **zusätzlich mit Datenkompression und Häufigkeitstabellen**. Außerdem machen sich die MPEG-Verfahren **zwei psychoakustische Eigenschaften** des menschlichen Hörapparates, den **Verdeckungseffekt** und den **Maskierungseffekt**, zu Nutze.

Das menschliche Ohr kann zwei frequenzmäßig eng beieinanderliegende Ereignisse nicht auflösen. Es nimmt nur das Lautere wahr. Außerdem nimmt es leisere Ereignisse, die zeitlich in der Nähe eines lauten Ereignisses liegen, nicht wahr. Diese psychoakustischen Phänomene, Verdeckungs- oder Maskierungseffekte genannt (*vgl. 1.4*), machten sich die MPEG-CoDecs zunutze. 'Maskierte' oder 'verdkckte' Ereignisse werden durch (speicherplatzsparende) Verringerung der Auflösung platzsparend ausgelassen.

Die MPEG-Spezifikation sieht eine Betriebsart für Monosignale und drei für Stereosignale vor:

- **Dual** Channel (zwei unabhängige Kanäle, z.B. für bilinguale Filme)
- **Stereo** (technisch identisch mit Dual Channel, Kanäle 'links' und 'rechts')
- **Joint-Stereo** (Redundanzen durch gleiche Ereignisse auf beiden Kanälen werden ausgenutzt - als MS [Mitte-Seite] bekannt)

MPEG 1 und MPEG 2 unterscheiden sich in ihren Audio-Spezifikationen im Wesentlichen durch die zulässigen Samplingraten. MPEG-1 läßt 32, 44,1 und 48 kHz zu - das waren zum Verabschiedungszeitpunkt die Standard-Raten für professionelle Produktionen (88,2 und 96 kHz wurden nur in einigen wenigen High-End-Mastering-Studios verwendet). MPEG-2 läßt zusätzlich auch niedrigere Samplingraten zu: 16, 22,05 und 24 kHz. Es gibt auch einen MPEG-2.5-'Standard', der aber nie offiziell verabschiedet wurde. Die Samplingraten von MPEG-2.5 liegen bei 8, 11,025 und 12 kHz.

Zudem können mit MPEG-2 mehr als zwei Kanäle codiert werden. Für Surround-Zwecke können bis zu 5 verschiedene Kanäle benutzt werden, zusätzlich noch ein Kanal mit ausschließlich tieffrequentem (bis 100 Hz) Material.

Die höhere Anzahl an Kanälen ermöglicht wesentlich komplexere Codierverfahren. Als zusätzliche Optionen für MPEG-2 sind zu nennen:

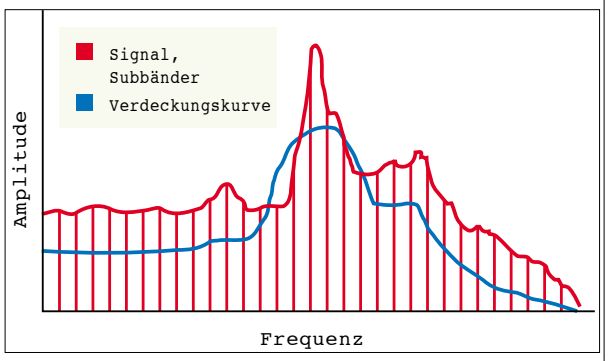
- ISC [Intensity-Stereo-Coding] (die räumliche Positionierung wird eher abstrakt, nicht Kanalgebunden gespeichert)
- PCC [Phantom-Coding of Center] (Ein Kanal wird aus der Summe oder Differenz von Anderen gewonnen)
- Dynamic Transmission Channel Switching (Zusatzinformationen werden in 'wenig ausgelasteten' Kanälen zu Gunsten der Qualität ausgelagert)
- Dynamic Cross Talk (man stelle sich eine Art 'dynamisches Joint-Stereo für Mehrkanal' vor)
- Adaptive Multi-Channel Prediction (Ereignisse auf verschiedenen Kanälen werden 'vorhergeschätzt' und mit dem tatsächlichen Ereignis verglichen. Es wird nur die entstehende Differenz abgespeichert)

Außerdem gibt es bei MPEG-2 auch zusätzlich Prüfsummen.

Ziel der drei MPEG-Algorithmen ist, **bei 384 kBit/s eine mit der CD vergleichbaren Qualität** zu bieten. Layer I ist am einfachsten und Layer III am komplexesten aufgebaut. Layer III bietet die beste Tonqualität bei geringen Datenraten.

Allen drei Layers gemein ist das Grundprinzip: die lineare **Unterteilung des Spektrums in 32 Bänder**. Jedes Band ist somit 625 Hz breit. Das gesamte Spektrum wird mit einer **FFT** analysiert (die Fast-Fourier-Transformation, FFT, liefert die Amplituden enthaltener Frequenzen - ein Frequenzspektrum also) und dem Kompressionsmodell zugeführt. Dieses **Modell 'entscheidet'** - neben anderen Methoden - **welches Band mit welcher Genauigkeit abgespeichert** wird.

Der komprimierte Datenstrom wird zusammen mit einigen Steuerinformationen in wenige 100 Bytes lange Blöcke (**Frames**) unterteilt. Der in einem Frame enthaltene Header dient unter anderem zur Synchronisation auf den fortlaufenden



Datenstrom, falls beispielsweise der **Dekodiervorgang mitten im MPEG-Strom** beginnt.

4.5.1 Layer I und II [.mpg], [.mp2]

Die erste Reduktion besteht nun darin, alle Subbänder wegzulassen deren Pegel unterhalb der Verdeckungsfunktion liegen. Da die Stellenzahl von 16 Bit pro Sample zur Darstellung der Pegel meist nicht komplett benötigt wird, liegt eine weitere Möglichkeit darin, die **führenden Nullen wegzulassen** (Beispiel: 0000001101101011 => 1101101011). Für das Verständnis der letzten Reduktion muß man sich noch einmal die Begleiterscheinungen der Quantisierung ins Gedächtnis rufen: Für jedes Bit weniger in der Auflösung ergeben sich 6 dB Rauschen. Man kann jedoch einige Bits in der **Auflösung weglassen** (gröbere Auflösung der Amplitude), **solange das dadurch entstehende Quantisierungsrauschen verdeckt wird**. Wenn ein Sample (Subband) beispielsweise 30 dB verdecken würde, könnte man also die unteren (Least Significant) 5 Bits einsparen. Konkret bedeutet dies, daß man die unteren 5 Bits (implizit) auf Null setzt (gröbere Auflösung) und sie dann abschneidet, da sie sowieso keinerlei Information mehr enthalten (1101101011 => 1101100000 => 11011).

Von unseren ursprünglichen 16 Bits bleiben also nur noch 5 übrig. Um aus diesen 5 Bits das original-Datenwort rekonstruieren zu können, benötigt man **zusätzlich einen Skalierfaktor**, der die Bits wieder in die richtige Position schiebt (11011 => 1101100000). Da man außerdem nicht mehr mit einer festen Stellenzahl von 16 Bit arbeiten kann, muß man die aktuelle Wortlänge (in unserem

Beispiel 5) ebenfalls mit abspeichern (Bit Allocation). Damit die gemachten Einsparungen nicht durch die zusätzlichen Kosten für Scalefactor und Bit Allocation wieder aufgezehrt werden, **teilen sich jeweils 12 Samples pro Subband** diese beiden Werte.

Selbst schwieriges Tonmaterial läßt sich auf diese Weise ohne hörbare Verluste mit einer Rate von (mindestens) 4:1 komprimieren. Bei stärkerer Kompression (max. 22:1) kann sich das Quantisierungsrauschen allmählich bemerkbar machen. Eine **weitere Möglichkeit** zur starken Kompression ist durch das **Weglassen der hohen Frequenzbänder** (obere Subbänder) gegeben, was jedoch mit klanglichen Einbußen verbunden ist (dumpfer Klang). Für eine verständliche Sprachübertragung beispielsweise ist jedoch eine obere Grenzfrequenz von ca. 3 kHz völlig ausreichend (Telefon-Standard). Man würde also mit 5 der ursprünglichen 32 Subbänder auskommen, was die Datenmenge auf ein Minimum reduziert.

Erreichbare Kompressionsraten liegen bei etwa 1:4.

4.5.2 Layer II [.mp2]

Durch bessere Nutzung des Psychoakustischen Modells läßt sich eine Steigerung der Kompression ohne merkliche Qualitätseinbußen erzielen. Layer II hat größere Blocklängen, nämlich 36 statt 12 Samples. Es lassen sich nun nicht nur frequenzmäßige sondern auch zeitliche Verdeckungseffekte mit berücksichtigen. Es wird notwendig, je nach Signal mehrere zeitlich aufeinanderfolgende Skalenfaktoren innerhalb eines Blocks zu verwenden. Layer II verwendet eine **dynamische Zuweisung der Faktoren zu den einzelnen Teilbändern. Da hochfrequente Signale eine wesentlich geringere Signalenergie haben, werden hier gezielt weniger Quantisierungsabstufungen angewandt.**

In Kombination mit dem **Packen der kodierten Bits** läßt sich die Datenrate weiter reduzieren.

Erreichbare Komprimierungsraten liegen etwa bei 1:8.

4.5.3 Layer III [.mp3]

Um den Kompressionsfaktor noch weiter zu erhöhen, erfolgt bei Layer III eine weitere **Verfeinerung** der Kompression, indem die Unterteilung des Signals in Frequenzbänder **von** ursprünglich **32** gleich breiten **auf 576 Bänder** mit variabler Breite erhöht wird. Desweiteren werden - der Charakteristik des menschlichen Ohrs nachempfunden - **nichtlineare Kennlinien für verschiedene Frequenzen** verwendet (*vgl. 4.1.2 - nichlineare Dynamik*). Das menschliche Ohr nimmt nämlich unterschiedliche Frequenzen unterschiedlich gut (also empfindlich) wahr. Abschließend werden die Daten mit einem **Huffman-Algorithmus** komprimiert (ein Verfahren aus der Computerdatenreduktion, das redundante Informationen 'einspart').

MPEG Audio Layer III basiert auf dem ASPEC-Verfahren, das vom Fraunhofer-Institut in Erlangen maßgeblich mitentwickelt wurde und ist bei kommerziellem Einsatz lizenzpflichtig. Brauchbare Klangqualität erreicht man oberhalb von 64 kBit/s pro Kanal.

Erreichbare Kompressionsraten liegen bei etwa 1:12.

4.6 QuickTime

QuickTime bietet eine große Anzahl verschiedener Video- und Audio-CoDecs an. Exemplarisch seien hier einige genannt.

4.6.1 AIFF-C

AIFF bedeutet 'Audio Interchange File Format' und wurde von der Firma Apple entwickelt. Firmen wie SGI (Silicon Graphics) und einige Andere haben es für ihre Anwendungen lizensiert.

Es erlaubt beliebige Sampleraten, Wortlängen, Anzahl an Kanälen und das Hinzufügen von Anwendungsspezifischen Informationen wie z.B. eine Einbettung von Wellenformübersichten, Kommentaren oder Cue-Listen.

Das 'C' bei AIFF-C steht für 'Compressed', also komprimiert. Es werden verschiedene verlustbehaftete Kompressionsarten unterstützt:

- ACE 2:1 und 8:3
- MACE 3:1 und 6:1
- µLaw
- IMA
- MPEG

ACE und MACE basieren auf ADPCM, verwenden aber nichtlineare Kennlinien (*vgl. 4.1.2*), µLaw ist ein Format mit logarithmischen Samplewerten (also nichtlinearer Kennlinie - *vgl. 4.1.2*) und IMA ist ein reines ADPCM-Verfahren. AIFF-C unterstützt IMA mit Kompressionsraten von 4:1 und 2:1.

Auf das von AIFF-C unterstützte MPEG-Verfahren wird im Punkt *4.5* genauer eingegangen.

4.6.2 QuickTime 3

Die Version 3 von QuickTime wurde um einige CoDecs erweitert.

4.6.2.1 Qualcomm PureVoice

Ein CoDec zur **Sprachkompressoin** mit sehr geringen Übertragungsraten (**um 1 kByte/s**), das sowohl psychoakustische als auch algorithmische Verfahren - ähnlich wie MPEG unter *4.5* und GSM unter *4.7* - einsetzt, das sich CELP nennt (Codebook Excited Linear Prediction). Es ist streamingfähig und für Sprachkommunikation über das Internet gedacht.

PureVoice bietet zwei Bandbreiteun zur Auswahl: 6,2 und 13,3 kBit/s, die in Echtzeit kodiert und dekodiert werden können. Es ist recht anfällig für schlechtes Ausgangsmaterial (verrauscht bzw. Hintergrundgeräusche), hat allerdings recht gute Qualität.

4.6.2.2 QDesign Music

Ein Kompressionsverfahren, das sich - ähnlich wie die MPEG-Verfahren - **psychakustischer** Effekte bedient. Bei QDesign kommen **permutative** Effekte hinzu. Das bedeutet, daß der **Signalverlauf vorher 'geschätzt'** wird und somit eine noch höhere Kompressionsrate möglich wird.

Außerdem erfolgt die Codierung der Daten erst nach einer sorgfältigen **automatischen Vorbereitung** (premastering) der Daten. Es wird zum Beispiel die Samplingfrequenz dynamisch verändert.

QDesign ist optimiert auf 24 kBit/s und bietet hervorragende (sofern man bei solchen Bandbreiten davon sprechen kann) Audioqualität bis ca. 64 kBit/s. Oberhalb von 64 kBit/s gibt es bessere Verfahren.

4.7 GSM

GSM steht für 'Global Systems for Mobile telecommunication'. Gemeint ist hier der Kompressionsalgorythmus - entwickelt für sehr schmalbandige digitale Funktelefonnetze in Europa - bedient sich **RPE-LTP,** was für '**residual pulse extraction - long term prediction**' steht. Die Funktionsweise läßt sich in etwa so erklären: aus dem Signal werden - anhand eines Modells der Eigenschaften des menschlichen Stimmtraktes - 'wichtige', also charakterisierende von 'unwichtigen' Daten getrennt. Die 'wichtigen' Daten (residual pulses) werden als Parameter einer mathematischen Funktion gespeichert, die 'unwichtigen' Daten werden geschätzt (**prediction**).

Die Tonsignale werden mono mit 8 kHz gesampled. Die Größe eines Paketes ist hierbei 160 Samples zu 13 Bit. Daraus ergibt sich eine Framerate von 50 Hz.

Das Resultat ist ein Datenstrom 33 Bytes großer Frames mit 13 kBit/s.

Etwas technischer erklärt:

Der GSM-Encoder arbeitet auf Blöcken zu je 160 samples des digitalisierten Eingangssignals. Diese werden skaliert, durch einen Hochpaß- und einen Preemphasisfilter bearbeitet. Wie bei ADPCM wird nun innerhalb des sogenannten LPC Blocks eine Voraussage der nächsten samples getroffen und die Differenz beider Werte gespeichert. Die Sequenz der Differenzwerte des Blockes (schon korreliert) wird dann als Filterparameter für einen Linear Prediction Filter benutzt, quantisiert und kodiert, dann als Logarithmic Area Ratios (LAR) bezeichnet und zum Dekodierer gesandt. Aus der Sequenz der LAR-Parameter wird dann ein sogenanntes Short Term Residual Signal (STR) interpoliert das den Fehler der Vorhersage für die 160 samples darstellt. Dann werden die STR-Signale mit den Samples kreuzkorreliert und ergeben die sogenannten Long Term Prediction Signale (LTP). Aus diesen läßt sich dann der sogenannte gain-Parameter berechnen, der angibt, wie die LTP-Signale skaliert werden müssen, um die STR-Signale möglichst gut abzubilden. Die LTP-Signale werden anhand von dem vorher bei der Berechnung angefallen lag-Parametern verschoben und mit dem gain-Parameter skaliert. Die entstehenden Signale heißen dann Long Term Estimate (LTE) und sind eine langfristige Vorhersage des Signalverlaufs. Es wird von STR abgezogen, dann wird noch ein Tiefpaßfilter darauf angewandt, schlußendlich findet noch ein Downsampling mit Faktor 3 (nur jeder dritte Wert wird benutzt) statt und anschließend werden sie mit einem APCM-Verfahren kodiert. Im Bitstrom werden zuerst frameweise alle berechneten Parameter übertragen, dann folgen die APCM-Werte.

Wenn Sie dieses Verfahren verstanden haben, haben sie sich einen Tüte Gummibärchen redlichst verdient ;-)

Quellen:

Terran Interactive: CodecCentral (http://www.terran.com/CodecCentral/index.html) und SondApp Formats (http://www-cs-students.stanford.edu/~franke/SoundApp/formats.html#mod)