Audio Coding and Standards

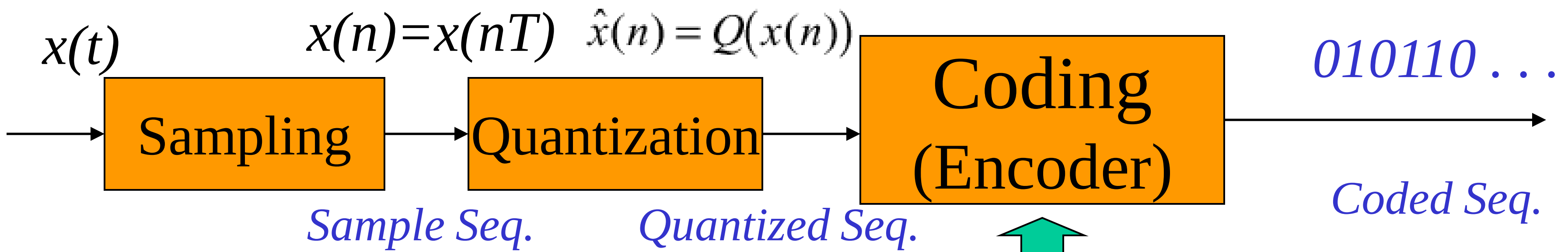
- Models, Techniques & Requirements of Sound Coding
- Entropy Coding: Run length Coding & Huffman coding
- Differential Coding – DPCM & ADPCM
- LPC and Parametric Coding
- Sound Masking Effect and Sub-band Coding
- ITU G.72x Speech/Audio Standards
- ISO MPEG-1/2/4 Audio Standards
- MIDI and Structured Audio
- Common Audio File Formats

PCM Audio Data Rate and Data Size

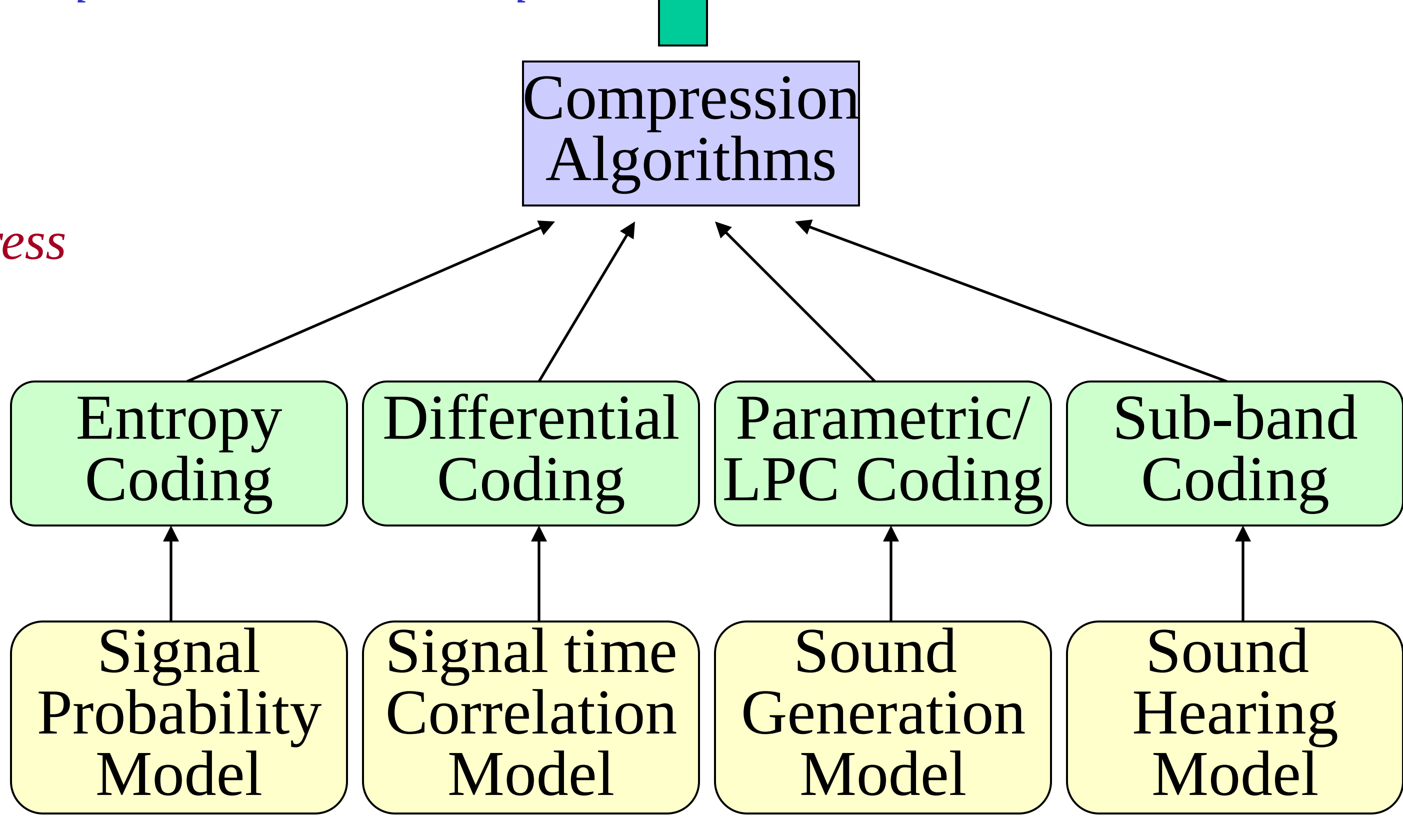
Quality	Sampling Rate (KHz)	Bits per Sample	Data Rate Kbits/s KBytes/s	Data Size in 1 minute 1 hour
Telephone	8	8 (Mono)	64Kbps 8KB/s	480KB 28.8MB
AM Radio	11.025	8 (Mono)	88.2Kbps 11.0KBps	660KB 39.6MB
FM Radio	22.050	16 (Stereo)	705.6Kbps 88.2KBps	5.3MB 317.5MB
CD	44.1	16 (Stereo)	1.41Mbps 176.4KBps	10.6MB 635MB

Conclusion → Need better coding for compressing sound data

Models & Techniques of Sound Compression

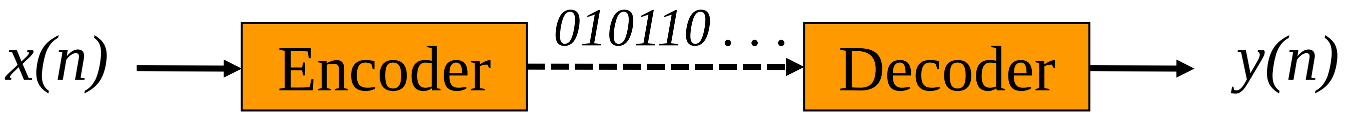


Terms:
Coding & Decoding
Encoder & Decoder
Compress & Decompress
Codec = Co + Dec



$$\textit{Compression Ratio} = \frac{\textit{Orig. Data Amount}}{\textit{Comp. Data Amount}}$$

Requirements for Compression Algorithms



- Lossless compression: $y(n) = x(n)$
  - Decoded audio is mathematically equivalent to the original one
  - Drawback : achieves only a small or modest level of compression
- Lossy compression: $y(n) \doteq x(n)$
  - Decoded audio is worse than the original one → *Distortion*
  - Advantage: achieves very high degree of compression
  - Objective: maximize the degree of compression in certain quality
- General compression requirements:
  - Ensure a good quality of decoded/uncompressed audio
  - Achieve high compression ratios
  - Minimize the complexity of the encoding and decoding process
  - Support multiple channels
  - Support various data rates
  - Give small delay in processing

Entropy Coding

- Entropy encoding (lossless): Ignores semantics of input data and compresses media streams ***x(n)*** by regarding them as sequences of digits or symbols
  - Examples: run-length encoding, Huffman encoding , ...
- Run-length encoding:
  - A compression technique that replaces consecutive occurrences of a symbol with the symbol followed by the number of times it is repeated
    - a a a a a => ax5
    - 000000000000000000001111111 => 0x20 1x7
  - Most useful where symbols appear in long runs: e.g., for images that have areas where the pixels all have the same value, fax and cartoons for examples.

Huffman Coding

Huffman encoding:

- A popular compression technique that

  → assigns *variable length binary codes* to symbols, so that the most frequently occurring symbols have the shortest codes
- Huffman coding is particularly effective where the data are dominated by a small number of symbols, e.g. *{x(n)} = hfeeeegheeegdeeehehcfbeeeeeqghf…*
- Suppose to encode a source of *N* =8 symbols: X(n)→{a,b,c,d,e,f,g,h}
- The probabilities of these symbols are: *P(a) = 0.01, P(b)=0.02, P(c)=0.05, P(d)=0.09, P(e)=0.18, P(f)=0.2, P(g)=0.2, P(h)=0.25*
- If assigning 3 bits per symbol (000~111), the average length of symbols is: $\overline{L} = \sum_{i=1}^{8} 3P(i) = 3 \text{ bits/symbol}$
- The theoretical lowest average length – **Entropy**

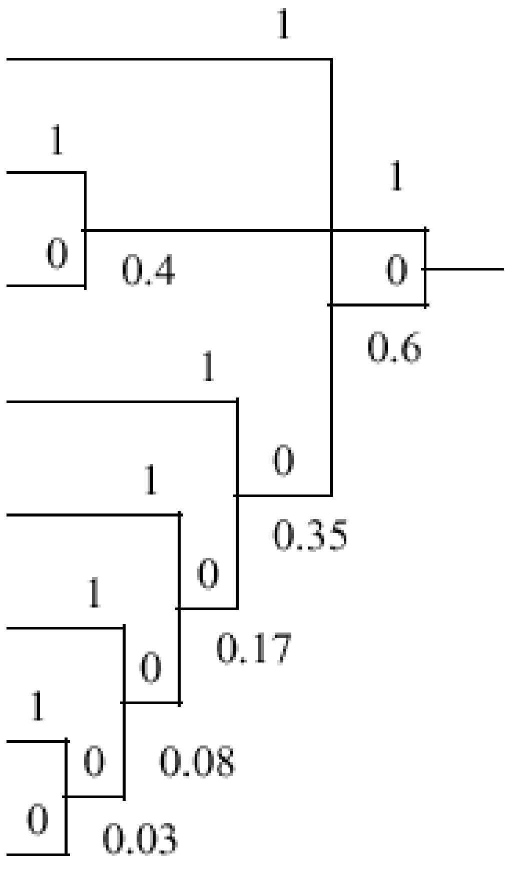
  $$H(P) = -\sum_{i=0}^{N} P(i)\log_2 P(i) = 2.57 \text{ bits /symbol}$$
- If we use Huffman encoding, the average length = 2.63 bits/symbol

Huffman Coding (Cont…)

- The Huffman code assignment procedure is based on a *binary tree* structure. This tree is developed *by a sequence of pairing operations* in which the *two least probable symbols* are joined at a node to form two branches of a tree. More precisely:
  - 1. The list of probabilities of the source symbols are associated with the leaves of a binary tree.
  - 2. Take the two smallest probabilities in the list and generate an intermediate node as their parent and label the branch from parent to one of the child nodes 1 and the branch from parent to the other child 0.
  - 3. Replace the probabilities and associated nodes in the list by the single new intermediate node with the sum of the two probabilities. If the list contains only one element, quit. Otherwise, go to step 2.

Huffman Coding (Cont…)

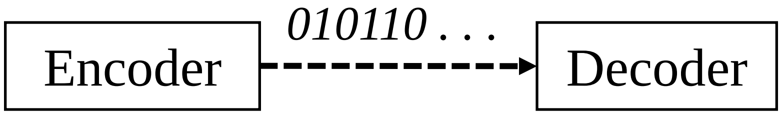
Original codewords	Huffman codewords	
111	01	P(h)=0.25
110	11	P(g)=0.2
101	10	P(f)=0.2
100	001	P(e)=0.18
011	0001	P(d)=0.09
010	00001	P(c)=0.05
001	000001	P(b)=0.02
000	000000	P(a)=0.01



Huffman Coding (Cont…)

Huffman Table

h:01	d:0001
g:11	c:00001
f: 10	b:000001
e: 001	a:0000001



- The new average length of the source

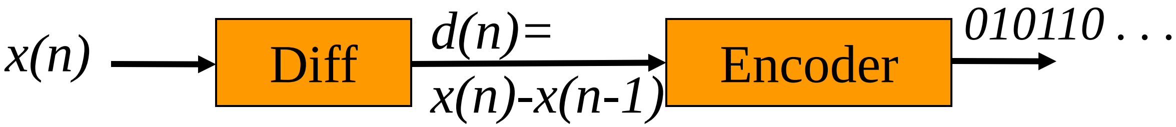
$$\bar{L}_{\mathrm{Huf}} = 0.25\text{x}2 + 0.2\text{x}2 + 0.2\text{x}2 + 0.18\text{x}3 + 0.09\text{x}4 + 0.05\text{x}5 + 0.02\text{x}6 + 0.01\text{x}6 = 2.63 \text{ bits/symbol}$$

- The efficiency of this code is $H/\bar{L}_{\mathrm{Huf}} = 98\%.$
- How do we estimate the *P(i)* ? Relative frequency of the symbols
- How to decode the bit stream ? Share the same Huffman table
- How to decode the variable length codes ? Prefix codes have the property that no codeword can be the prefix (i.e., an initial segment) of any other codeword. Huffman codes are prefix codes !
  - 00000100100110 => ? beef
- Does the best possible codes guarantee to always reduce the size of sources? No. Worst case exists. Huffman coding is better averagely.
- Huffman coding is particularly effective where the data are dominated by a small number of symbols

Differential Coding – DPCM & ADPCM

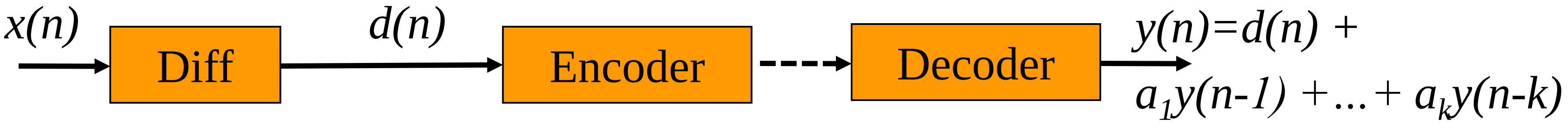
- Based on the fact that neighboring samples … *x(n-1), x(n), x(n+1)*, … in a discrete audio sequence changing slowly in many cases
- A differential PCM coder (DPCM) quantizes and encodes the difference

$d(n) = x(n) – x(n-1)$

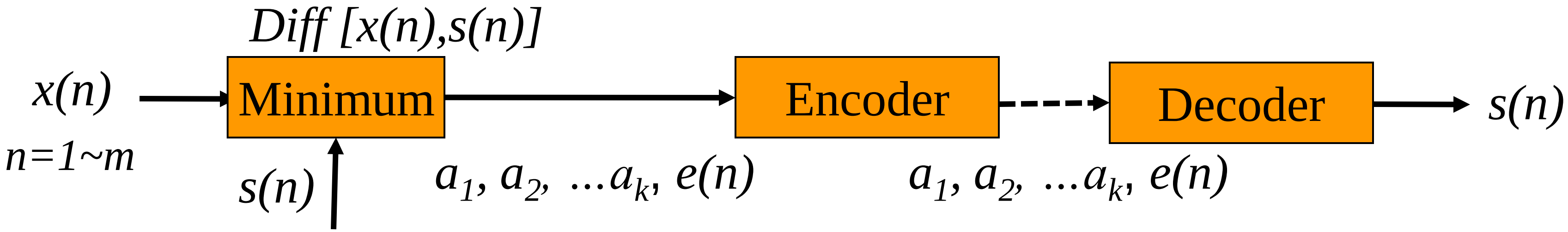


- Advantage of using difference *d(n)* instead of the actual value *x(n)*
  - Reduce the number of bits to represent a sample
- General DPCM: $d(n) = x(n) – a_1x(n-1) - a_2x(n-2) -…- a_kx(n-k)$

  $a_1, a_2, …a_k$ *are fixed*
- Adaptive DPCM: $a_1, a_2, …a_k$ *are dynamically changed with signal*



LPC and Parametric Coding



LPC (Linear Predictive Coding)

- Based on the human utterance organ model

$$s(n) = a_1 s(n-1) + a_2 s(n-2) + \ldots + a_k s(n-k) + e(n)$$

- Estimate $a_1, a_2, \ldots a_k$ and $e(n)$ for each piece (frame) of speech
- Encode and transmit/store $a_1, a_2, \ldots a_k$ and type of $e(n)$
- Decoder reproduce speech using $a_1, a_2, \ldots a_k$ and $e(n)$

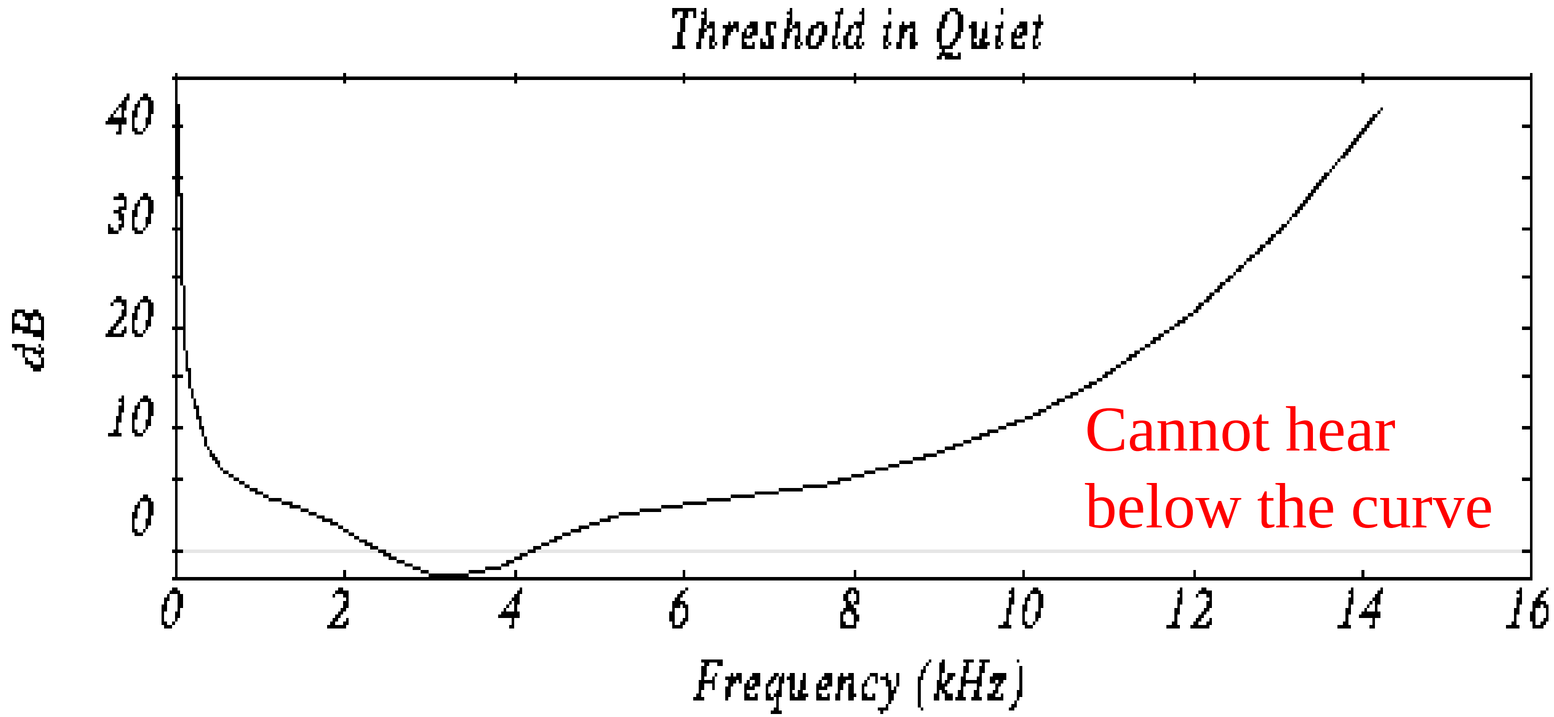
  - very low bit rate but relatively low speech quality

Parametric coding:

- Only coding parameters of sound generation model
- LPC is an example where parameters are $a_1, a_2, \ldots a_k$, $e(n)$
- Music instrument parameters: pitch, loudness, timbre, …

Sub-band Coding

- Human auditory system has limitations
  - – Frequency range: 20 Hz to 20 kHz, sensitive at 2 to 4 KHz.
  - – Dynamic range (quietest to loudest) is about 96 dB



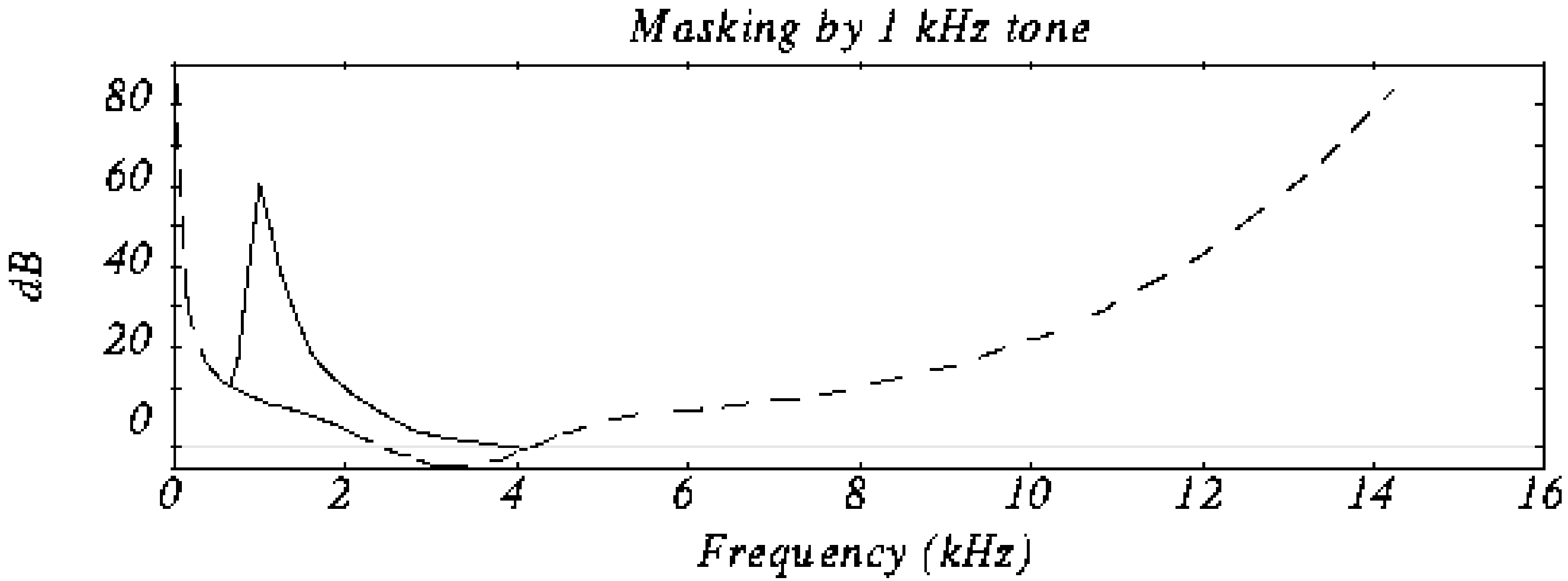
Cannot hear below the curve

- Moreover, based on psycho-acoustic characteristics of human hearing, algorithms perform some tricks to further reduce data rate

Masking Effects

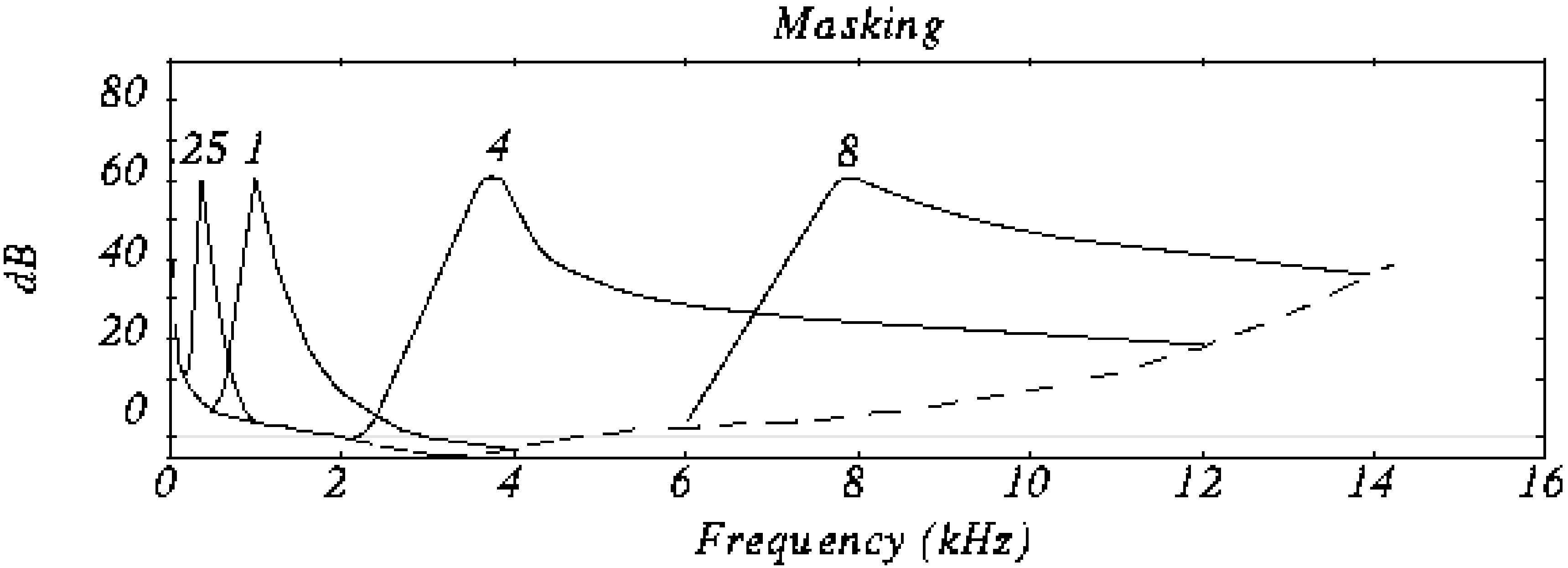
- Frequency Masking: If a tone of a certain frequency and amplitude is present, then other tones or noise of similar frequency cannot be heard by the human ear
- the louder tone (masker) makes the softer tone (maskee)

=> no need to encode and transfer the softer tone



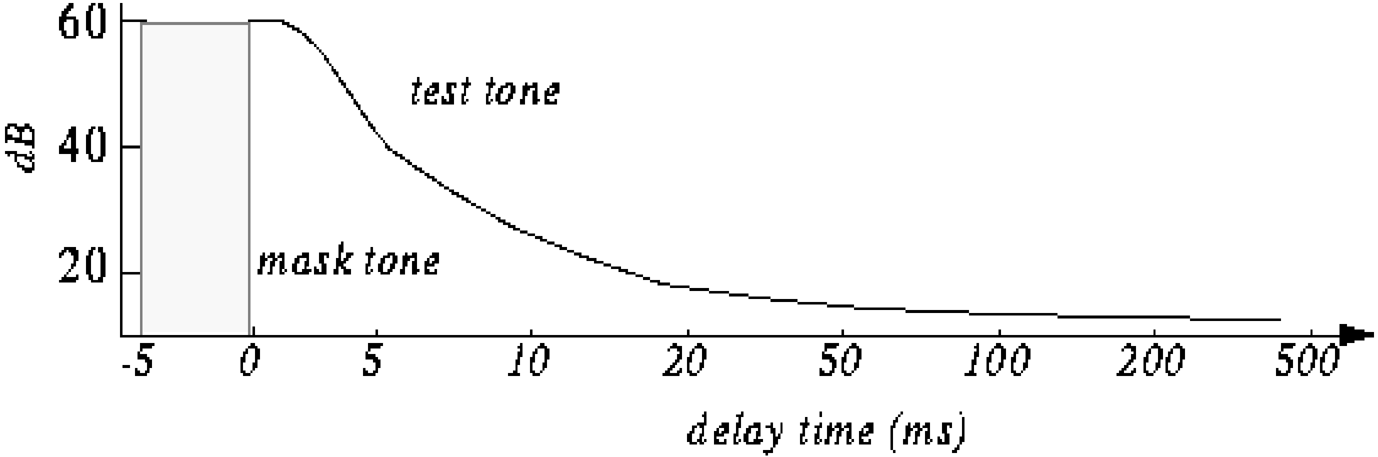
Masking Effects (Cont…)

- Repeat for various frequencies of masking tones
- **Masking Threshold**: Given a certain masker, the maximum non-perceptible amplitude level of the softer tone



Masking Effects (Cont…)

- Temporal Masking: If we hear a loud sound, then it stops, it takes a little while until we can hear a soft tone nearby.



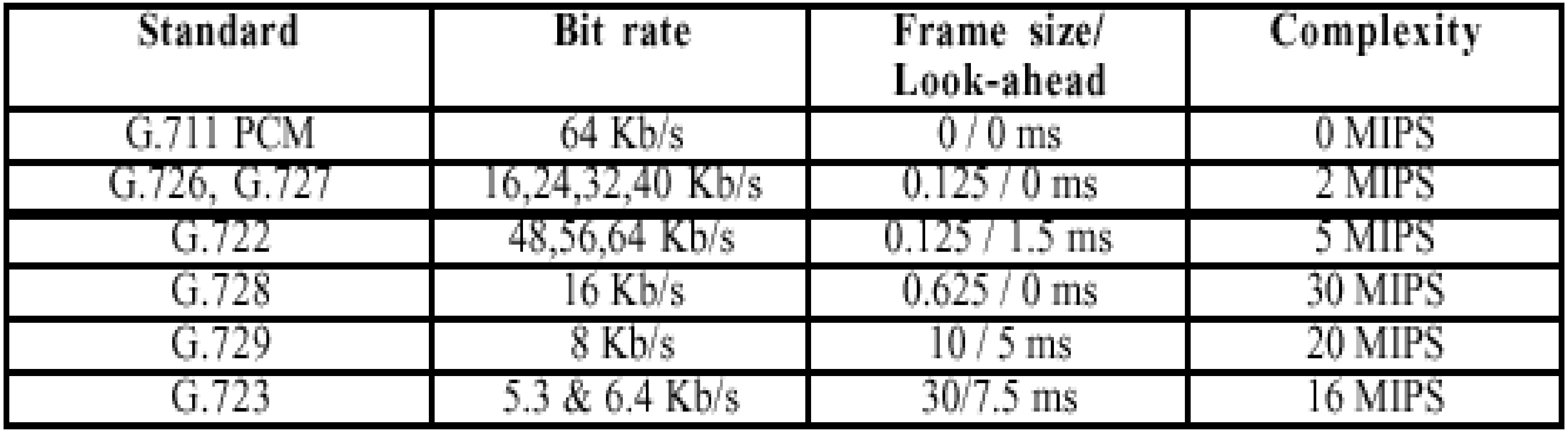
- The Masking Threshold is used by the audio encoder to determine the maximum allowable quantization noise at each frequency to minimize noise perceptibility: remove parts of signal that we cannot perceive

Speech Compression

- Handling speech with other media information such as text, images, video, and data is the essential part of multimedia applications
- The ideal speech coder has a low *bit-rate*, high perceived *quality*, low signal *delay*, and low *complexity*.
- **Delay**
  - Less than 150 ms one-way end-to-end delay for a conversation
  - Processing (coding) delay, network delay
  - Over Internet, ISDN, PSTN, ATM, …
- **Complexity**
  - Computational complexity of speech coders depends on algorithms
  - Contributes to achievable bit-rate and processing delay

G.72x Speech Coding Standards

- **Quality**
  - “intelligible” → “natural” or “subjective” quality
  - Depending on bit-rate
- **Bit-rate**



Standard	Bit rate	Frame size/ Look-ahead	Complexity
G.711 PCM	64 Kb/s	0 / 0 ms	0 MIPS
G.726, G.727	16,24,32,40 Kb/s	0.125 / 0 ms	2 MIPS
G.722	48,56,64 Kb/s	0.125 / 1.5 ms	5 MIPS
G.728	16 Kb/s	0.625 / 0 ms	30 MIPS
G.729	8 Kb/s	10 / 5 ms	20 MIPS
G.723	5.3 & 6.4 Kb/s	30/7.5 ms	16 MIPS

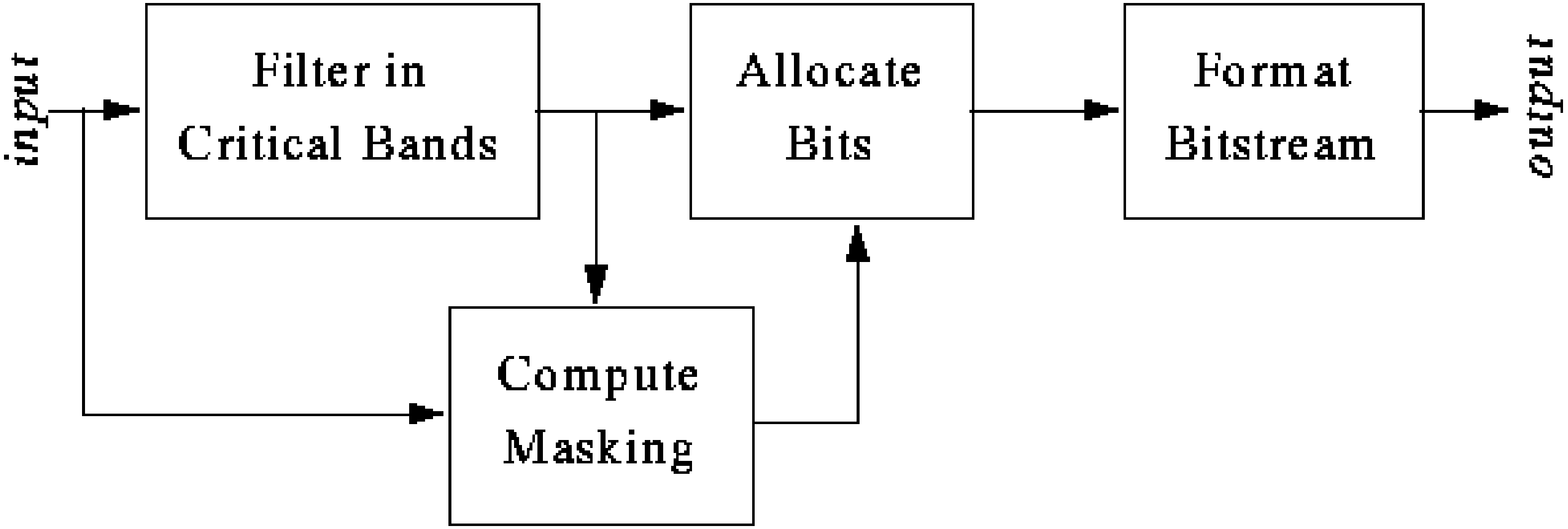
G.72x Audio Coding Standards

- **Silence Compression** - detect the "silence", similar to run-length coding
- **Adaptive Differential Pulse Code Modulation (ADPCM)** e.g., in CCITT G.721 -- 16 or 32 Kb/s.
  - (a) Encodes the difference between two or more consecutive signals; the difference is then quantized → hence the loss *(speech quality becomes worse)*
    (b) Adapts at quantization so fewer bits are used when the value is smaller.
  - It is necessary to predict where the waveform is headed → difficult
- **Linear Predictive Coding (LPC)** fits signal to speech model and then transmits parameters of model

  → sounds like a computer talking, 2.4 Kb/s.

MPEG-1/2 Audio Compression

1. Use filters to divide the audio signal (e.g., 20-20kHz sound) into 32 frequency subbands --> *subband filtering*.
2. Determine amount of masking for each band caused by nearby band using the *psycho-acoustic model*.
3. If the power in a band is below masking threshold, don't encode it.
4. Otherwise, determine no. of bits needed to represent the coefficient such that noise introduced by quantization is below the masking effect (one fewer bit of quantization introduces about 6 dB of noise).
5. Format bitstream



MPEG Audio Compression Example

- After analysis, the first levels of 16 of the 32 bands are these:

Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Level(db)	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

- If the level of the 8th band is 60dB, it gives a masking of 12 dB in the 7th band, 15dB in the 9th.
- Level in 7th band is 10 dB (< 12 dB), so ignore it.
- Level in 9th band is 35 dB (> 15 dB), so send it.
  [Only the amount above the masking level needs to be sent, so instead of using 6 bits to encode it, we can use 4 bits -- saving 2 bits (= 12 dB).]

MPEG Audio Layers

- MPEG defines 3 layers for audio. Basic model is same, but codec complexity increases with each layer.
- Divides data into frames, each of them contains 384 samples, 12 samples from each of the 32 filtered subbands.
- Layer 1: DCT type filter with one frame and equal frequency spread per band. Psycho-acoustic model only uses frequency masking.
- Layer 2: Use three frames in filter (before, current, next, a total of 1152 samples). This models a little bit of the temporal masking.
- Layer 3: Better critical band filter is used (non-equal frequencies), psycho-acoustic model includes temporal masking effects, takes into account stereo redundancy, and uses Huffman coder
- MP3: Music compression format using MPEG Layer 3

MPEG Audio Layers (Cont…)

Layer	**Target Bit-rate** per channel	**Ratio**	**Quality at 64 kb/s**	**Quality at 128 kb/s**	**Theoretical Min. Delay**
Layer 1	192 kb/s	4:1	---	---	19 ms
Layer 2	128 kb/s	6:1	2.1 to 2.6	4+	35 ms
Layer 3	64 kb/s	12:1	3.6 to 3.8	4+	59 ms

- Quality factor: 5 - perfect, 4 - just noticeable, 3 - slightly annoying,
  2 - annoying, 1 - very annoying
- Real delay is about 3 times of the theoretical delay

MPEG-1 Audio Facts

- MPEG-1: 64K~320Kbps for audio
  - Uncompressed CD audio => 1.4 Mb/s
- Compression factor ranging from 2.7 to 24.
- With Compression rate 6:1 (16 bits stereo sampled at 48 KHz is reduced to 256 kb/s) and optimal listening conditions, expert listeners could not distinguish between coded and original audio clips.
- MPEG audio supports sampling frequencies of 32, 44.1 and 48 KHz.
- Supports one or two audio channels in one of the four modes:
  1. Monophonic -- single audio channel
  2. Dual-monophonic -- two independent chs, e.g., English and French
  3. Stereo -- for stereo channels that share bits, but not using Joint-stereo coding
  4. Joint-stereo -- takes advantage of the correlations between stereo channels

MPEG-2 Audio Coding

- MPEG-2/MC: Provide theater-style surround sound capabilities
  - Five channels: left, right, center, rear left, and rear right
  - Five different modes: mono, stereo, three ch, four ch, five ch
  - Full five channel surround stereo: 640 Kb/s
  - 320 Kb/s for 5.1 stereo (5 channels+sub-woofer ch)
- MPEG-2/LSF (Low sampling frequency: 16k, 22K, 24k)
- MPEG-2/AAC (Advanced Audio Coding)
  - 7.1 channels
  - More complex coding
- Compatibility
  - Forward: MPEG-2 decoder can decode MPEG-1 bitstream
  - Backward: MPEG-1 decoder can decode a part of MPEG-2

MPEG-4 Audio Coding

- Consists of natural coding and synthetic coding
- Natural coding

  - General coding: AAC and TwinVQ based arbitrary audio
  twice as good as MP3

  - Speech coding:

    * CELP I: 16K samp., 14.4~22.5Kbps

    * CELP II: 8K & 16K samp., 3.85~23.8Kbps

    * HVXV: 8M samp., 1.4~4Kbps
- Synthetic coding: structured audio
  - Interface to Text-to-Speech synthesizers
  - High-quality audio synthesis with Structured Audio
- AudioBIFS: Mix and postproduce multi-track sound streams

Structured Audio

- A description format that is made up of semantic information about the sounds it represents, and that makes use of high-level (algorithmic) models.
  - E.g., MIDI (Musical Instrument Digital Interface).
- Normal music digitization: perform *waveform coding* (we sample the music signal and then try to reconstruct it exactly)
- MIDI: only record *musical actions* such as the key depressed, the time when the key is depressed, the duration for which the key remains depressed, and how hard the key is struck (pressure).
- MIDI is an example of **parameter** or **event-list representation**
  - An event list is a sequence of control parameters that, taken alone
  - Do not define the quality of a sound but instead specify the ordering and characteristics of parts of a sound with regards to some external model.

Structured Audio Synthesis

- **Sampling synthesis**
  - Individual instrument sounds are digitally recorded and stored in memory
  - When the instrument is played, the note recording are reproduced and mixed (added together) to produce the output sound.
  - This can be a very effective and realistic but requires a lot of memory
  - Good for playing music but not realistic for speech synthesis
  - Good for creating special sound effects from sample libraries

Structured Audio Synthesis (Cont…)

- Additive and subtractive synthesis
  - synthesize sound from the superposition of sinusoidal components (additive)
  - Or from the filtering of an harmonically rich source sound - typically a periodic oscillator with various form of waves (subtractive).
  - Very compact representation of the sound
  - the resulting notes often have a distinctive “analog synthesizer” character.

Applications of Structured Audio

- Low-bandwidth transmission
  - transmit a structural description and dynamically render it into sound on the client side rather than rendering in a studio on the server side
- Sound generation from process models
  - the sound is not created from an event list but rather is dynamically generated in response to evolving, non-sound-oriented environments such as video games
- Music applications
- Content-based retrieval
- Virtual reality together with VRML/X3D

Common Audio File Formats

- Mulaw (Sun, NeXT) .au
- RIFF Wave (MS WAV) .wav
- MPEG Audio Layer (MPEG) .mp2 .mp3
- AIFC (Apple, SGI) .aiff .aif
- HCOM (Mac) .hcom
- SND (Sun, NeXT) .snd
- VOC (Soundblaster card proprietary standard) .voc
- AND MANY OTHERS!

Demos of Audio Coding and Formats