Digital speech coding

Ary Shiddiqi

ary.shiddiqi@if.its.ac.id

What's the need for speech coding ?

- Necessary in order to represent human speech in a digital form
- Applications: mobile/telephone communication, voice over IP
- Code efficiency (high quality, fewer bits) is a must

Analog to digital conversion

- The speech sounds can be converted into electrical signals by a transducer, such as a microphone, which transforms the acoustic waves into an electrical current.
- the electrical current can be sampled (analog-to-digital converted) as discrete data, with each sample typically represented by eight bits.

Type of Speech Coders

- Waveform codecs
  - Sample and code
  - High-quality and not complex
  - Large amount of bandwidth
- source codecs (vocoders)
  - Match the incoming signal to a math model
  - Linear-predictive filter model of the vocal tract
  - A voiced/unvoiced flag for the excitation
  - The information is sent rather than the signal
  - Low bit rates, but sounds synthetic
  - Higher bit rates do not improve much

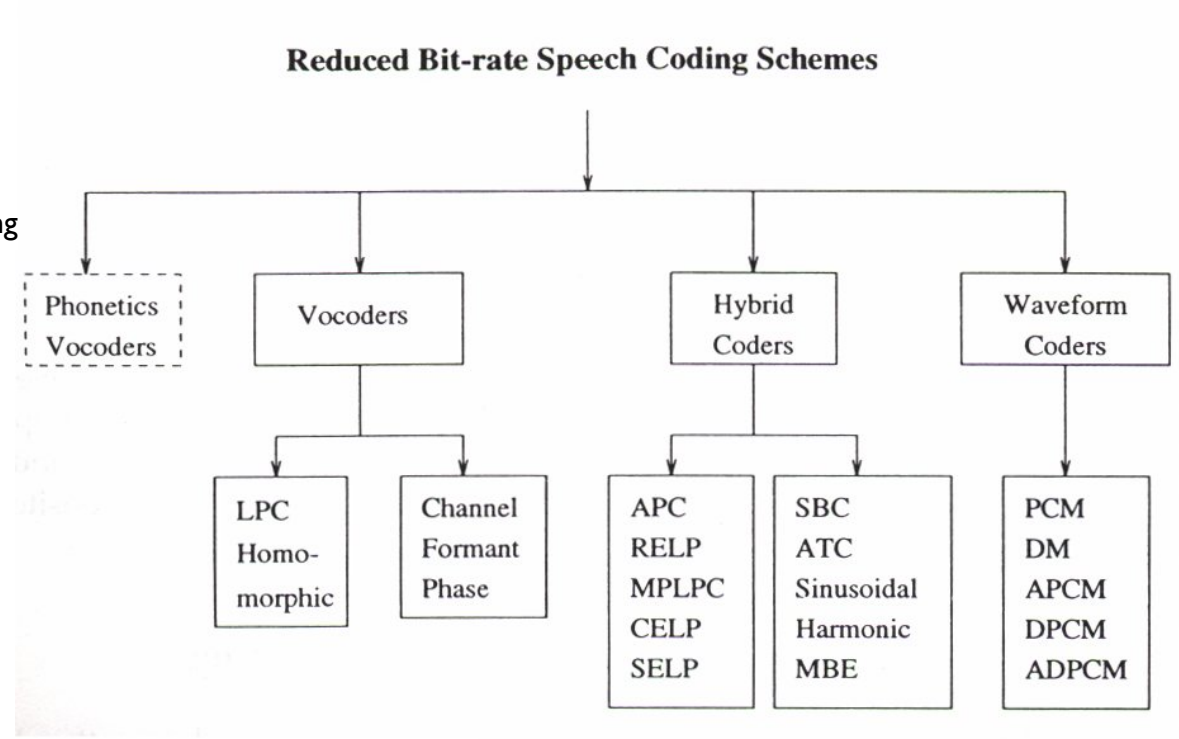
Type of Speech Coders

- Hybrid codecs
  - Attempt to provide the best of both
  - Perform a degree of waveform matching
  - Utilize the sound production model
  - Quite good quality at low bit rate

The goal of speech coding strategies

- The strategies aim to analyse the signal, remove the redundancies, and efficiently code the non-redundant parts of the signal in a perceptually acceptable manner.

- SBC = sub-band coding
- ATC = adaptive transform coding
- MBE = multiband excitation
- APC = adaptive coding
- RELP = residual excited linear predictive coding (LPC)
- MPLPC = multi-pulse LPC
- CELP = code-excited LPC
- SELP = self-excitation LPC.



Waveform Codec

- Waveform codec's attempt, without using any knowledge of how the signal to be coded was generated, to produce a reconstructed signal whose waveform is as close as possible to the original.
- This means that in theory they should be signal independent and work well with non-speech signals.
- Generally they are low complexity codec's which produce high quality speech at rates above about 16 kbits/s.
- When the data rate is lowered below this level the reconstructed speech quality that can be obtained degrades rapidly

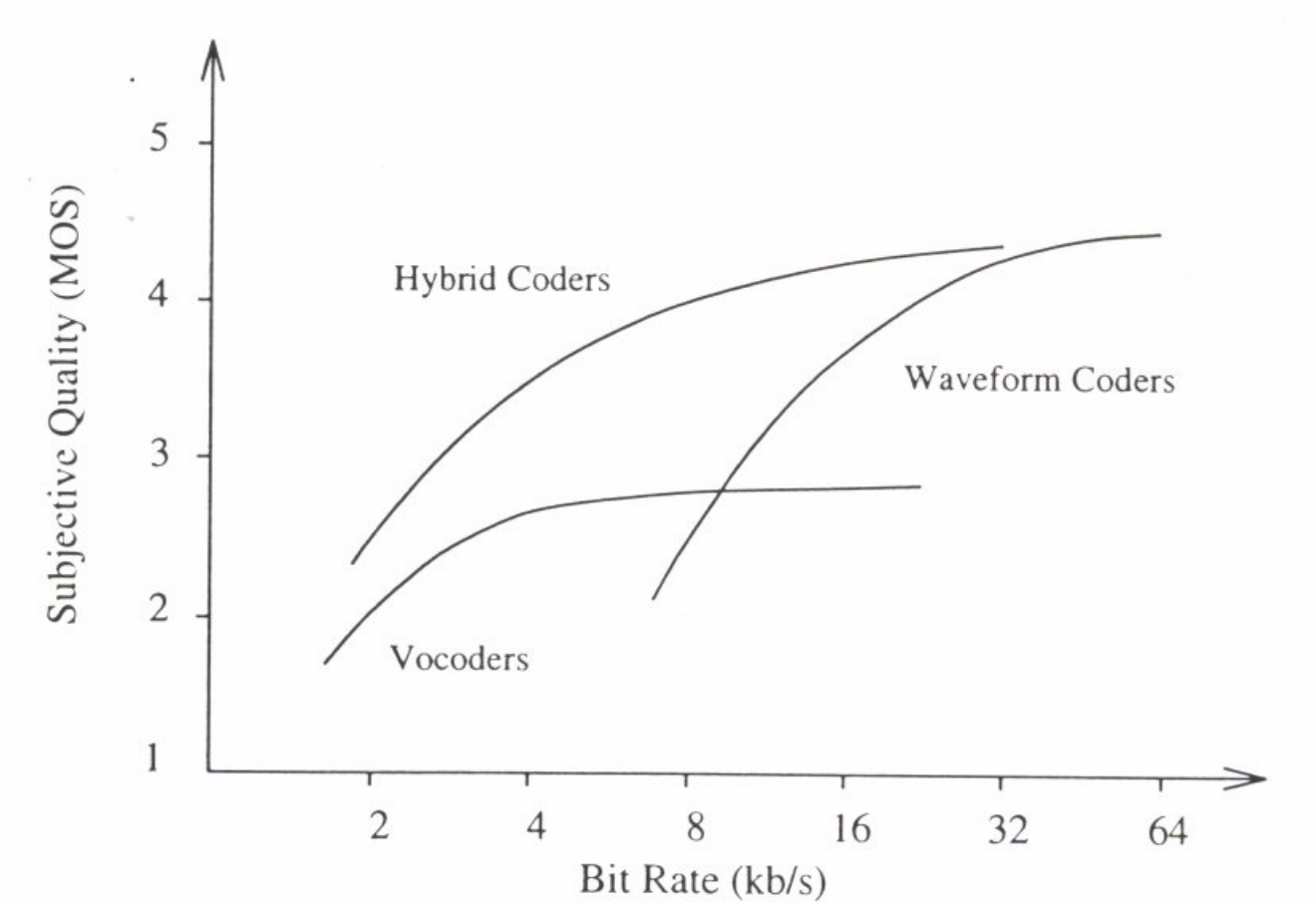
Source Codec

- Source coders operate using a model of how the source was generated, and attempt to extract, from the signal being coded, the parameters of the model.
- It is these model parameters which are transmitted to the decoder.
- Source coders for speech are called **vocoders**, and work as follows.
- The vocal tract is represented as a time-varying filter and is exited with either a white noise source for unvoiced speech segments, or a train of pulses separated by the pitch period for voiced speech.
- Therefore the information which must be sent to the decoder is the *filter specification, a voiced/unvoiced flag, the necessary variance of the excitation signal, and the pitch period for voiced speech.*

Hybrid Codec

- Hybrid codecs attempt to fill the gap between waveform and source codecs.
- Waveform coders are capable of providing good quality speech at bit rates down to about 16 kbits/s, but are of limited use at rates below this.
- Source coders on the other hand can provide intelligible speech at 2.4 kbits/s and below, but cannot provide natural sounding speech at any bit rate.
- Although other forms of hybrid codecs exist, the most successful and commonly used are time domain Analysis-by-Synthesis (AbS) codecs.

Quality of speech coding schemes



Quality measurement of coded audio

- The quality of audio is measured using
  - Signal to noise ratio (SNR)

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}$$

  - Total block distortion (TBD)
- Perceptual objective measure the quality of audio is predicted based on a specific model of hearing.

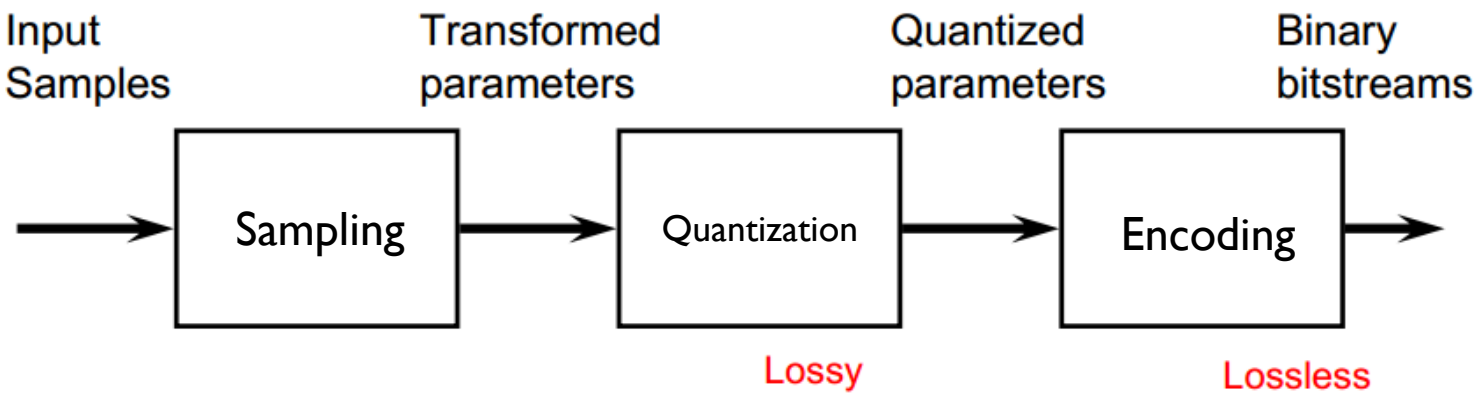
Coding dilemma

- In practical audio codec design, it is always a trade-off between the following two important factors:
  - Data rate and system complexity limitation
  - Audio quality

Pulse Code Modulation (PCM)

- a method used to digitally represent sampled analog signals.
- It is the standard form of digital audio in computers, compact discs, digital telephony and other digital audio applications
- two basic properties:
  - the sampling rate, which is the number of times per second that samples are taken
  - the bit depth, which determines the number of possible digital values that can be used to represent each sample.

The steps of the PCM speech coding system



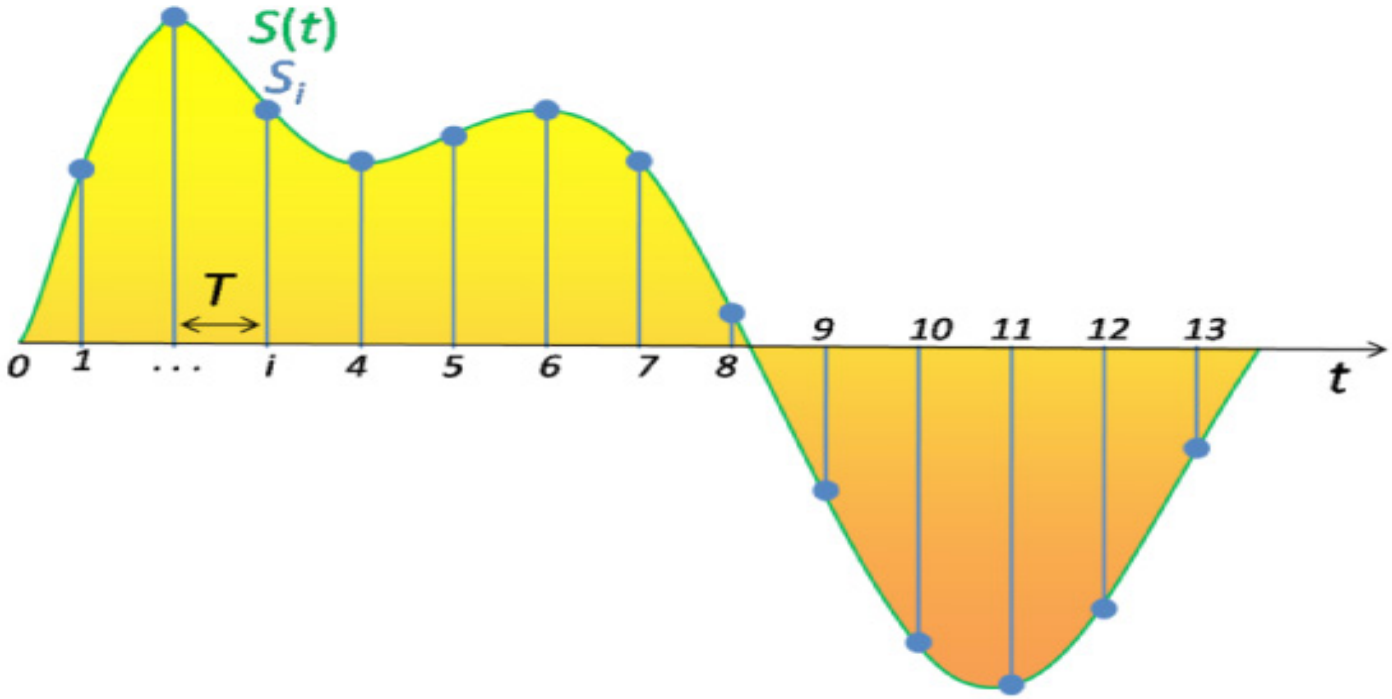
Taking sample within a range of *T*

Scalar Q
Vector Q

$Y(t) = Q(x(t))$

Fixed length
Variable length (Huffman, arithmetic, LZW)

Voice Sampling

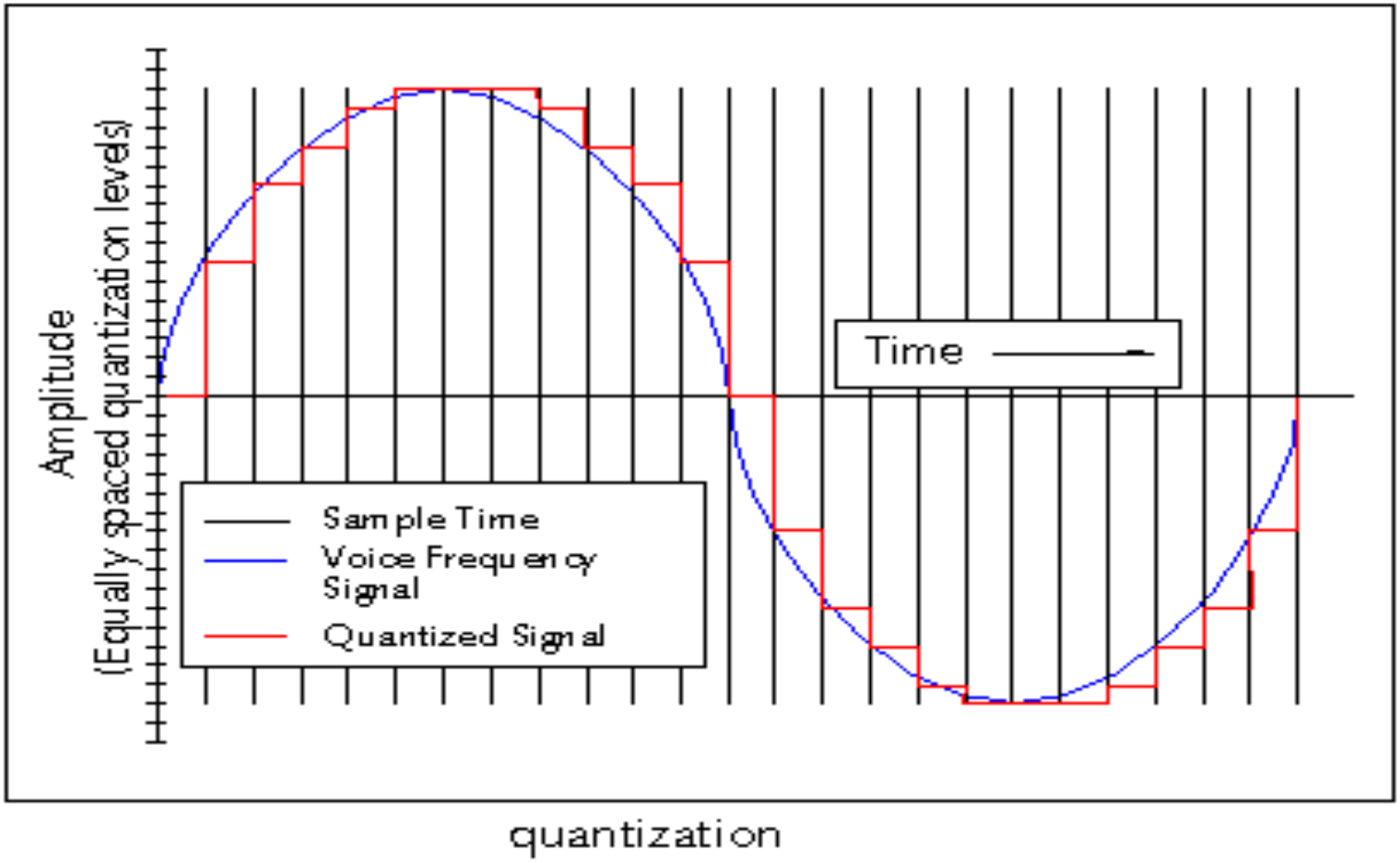


An analog waveform is converted to digital audio samples. Each sample is taken at a fixed time interval.

Sampling Rate

- Nyquist–Shannon sampling theorem shows PCM devices can operate without introducing distortions within their designed frequency bands if they provide a sampling frequency **at least twice** that of the highest frequency contained in the input signal
- For example:
  - Speech frequency range is 300 – 3400 cycles/second
  - So for conversational speech the maximum would be 4000 cycles/second
  - The sampling rate would then be at least 8000 samples per second

Quantization

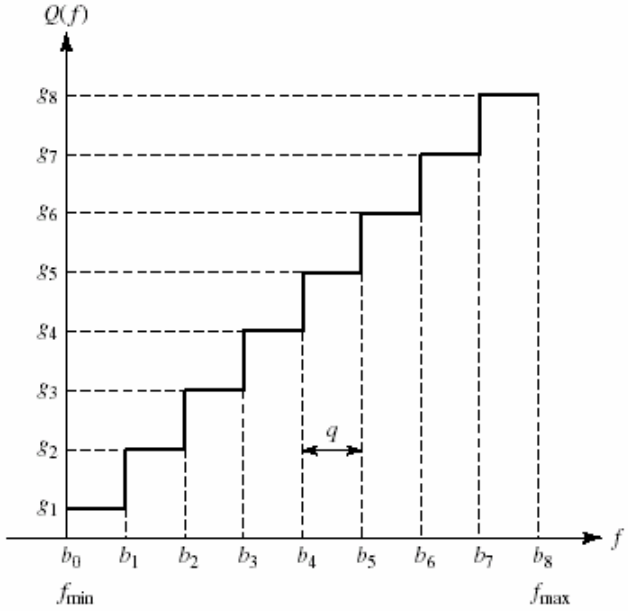


Quantization Noise

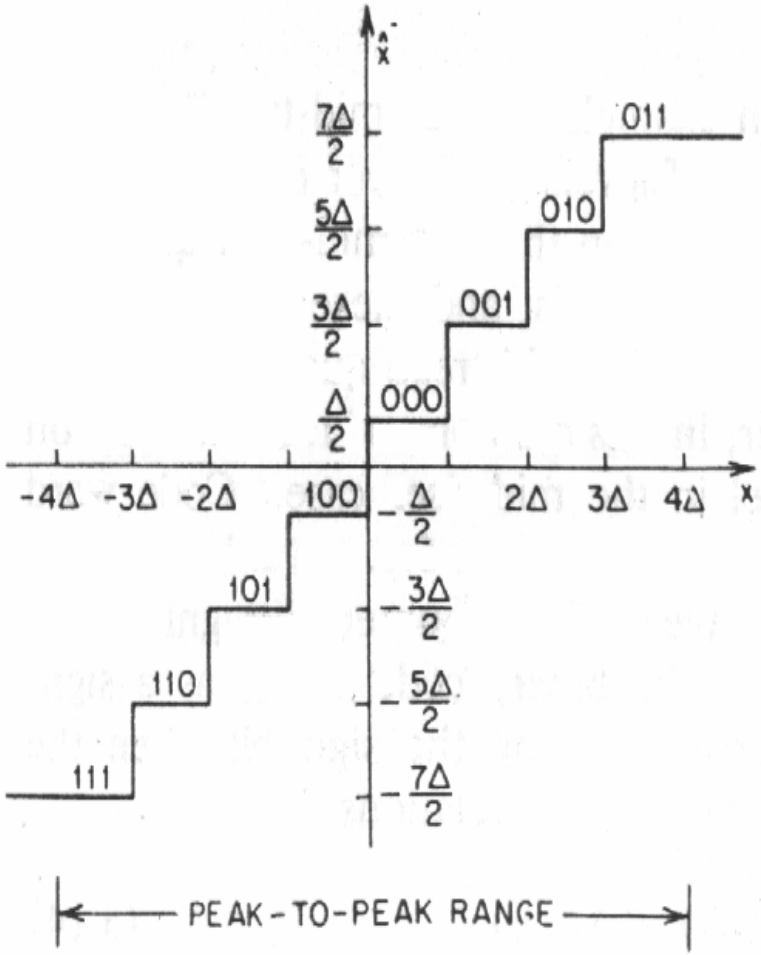
- The number of bits used determines the number of levels
- The number of levels determines the accuracy of our representation of the original signal
- The bits can not precisely model a level of a voice signal, some differences might occur
- The difference between the actual signal and the digital reproduction is known as Quantization Noise

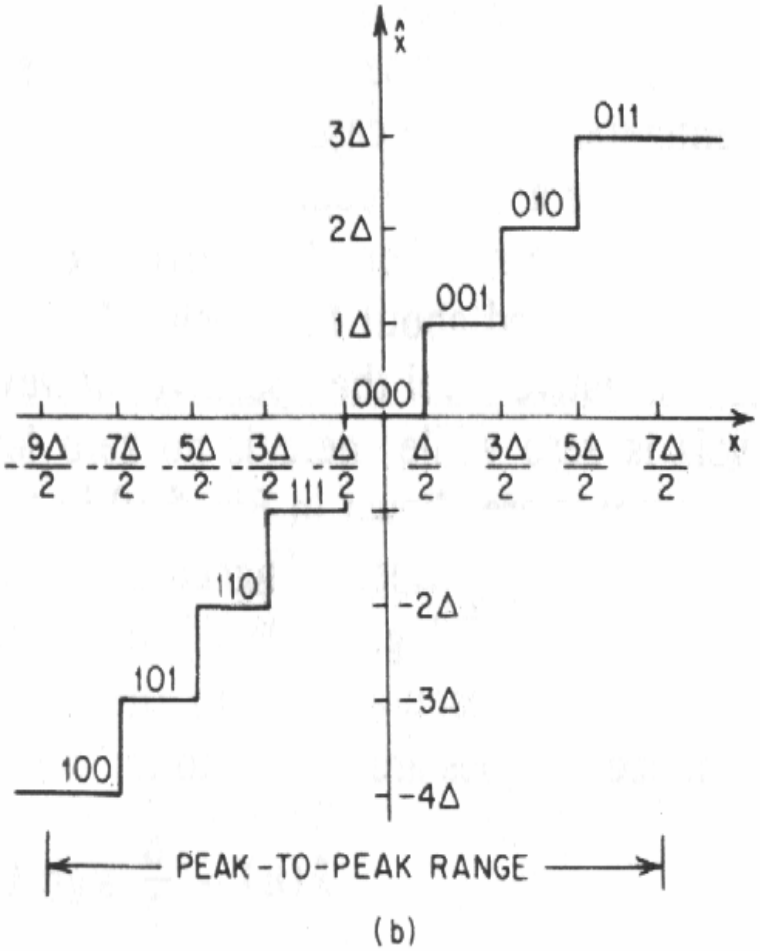
Uniform Quantization

- Applicable when the signal is in a finite range (*fmin, fmax)*
- The entire data range is divided into *L equal intervals of length q (known as quantization interval or quantization step-size)*
- *q=(fmax-fmin)/L*



Symmetric Signal Range in Linear Quantization



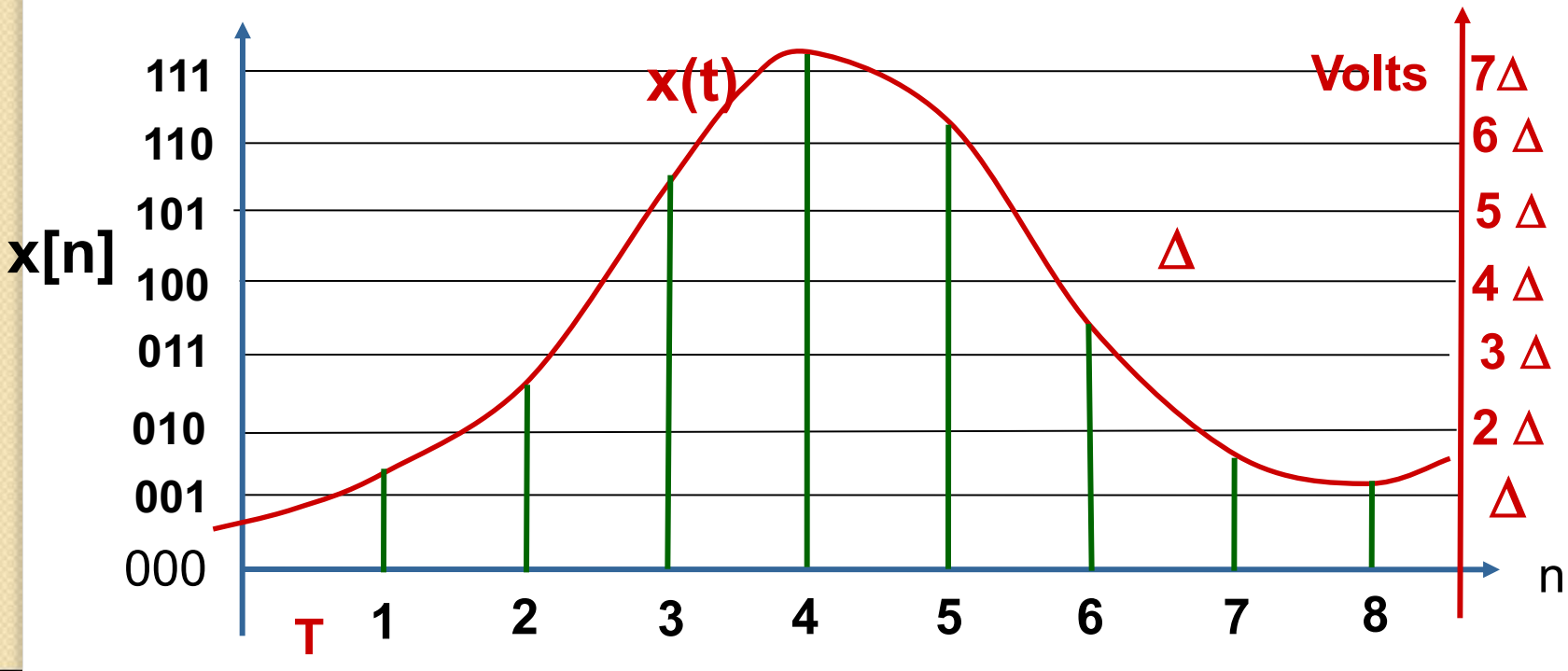


Errors

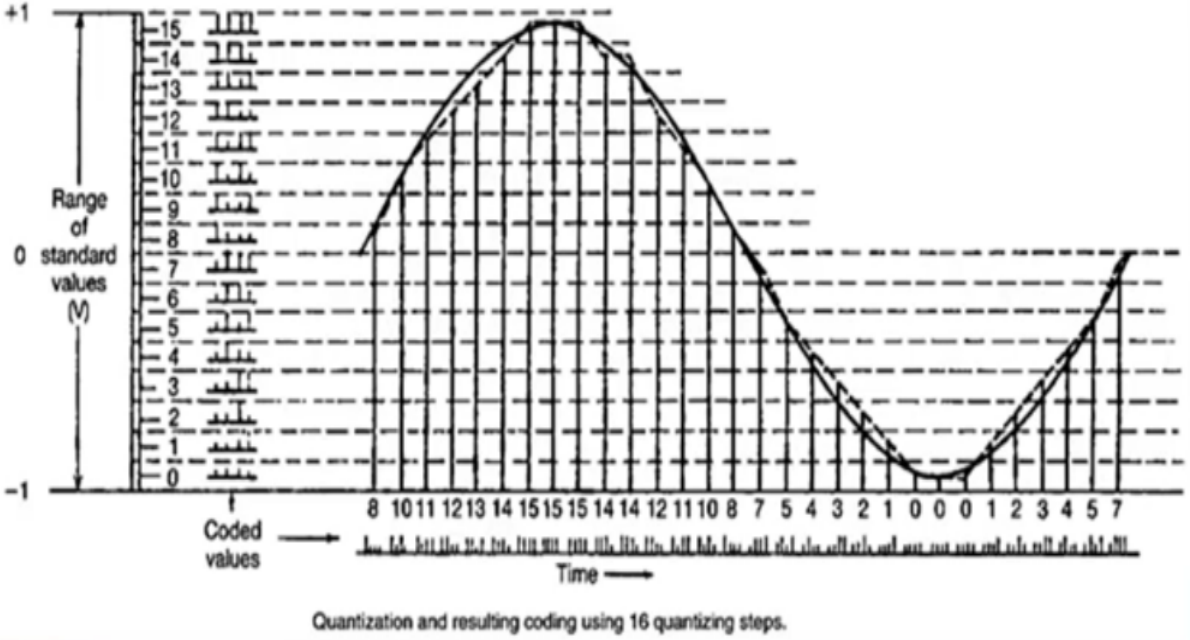
- If smaller steps are taken the quantization error will be less
- However, increasing the steps will complicate the coding operation and increase bandwidth requirements.
- Quantizing noise depends on step size and not on signal amplitude
- Amplitudes at the same level are coded using precisely the same bits

Uniform quantisation

- Each sample of speech x(t) is represented by a binary number x[n].
- Each binary number represents a quantisation level.
- With uniform quantisation there is constant voltage difference between levels.



Problems with the uniform quantization



High Level Signal

0.05V/5V = 1%

Low Level Signal

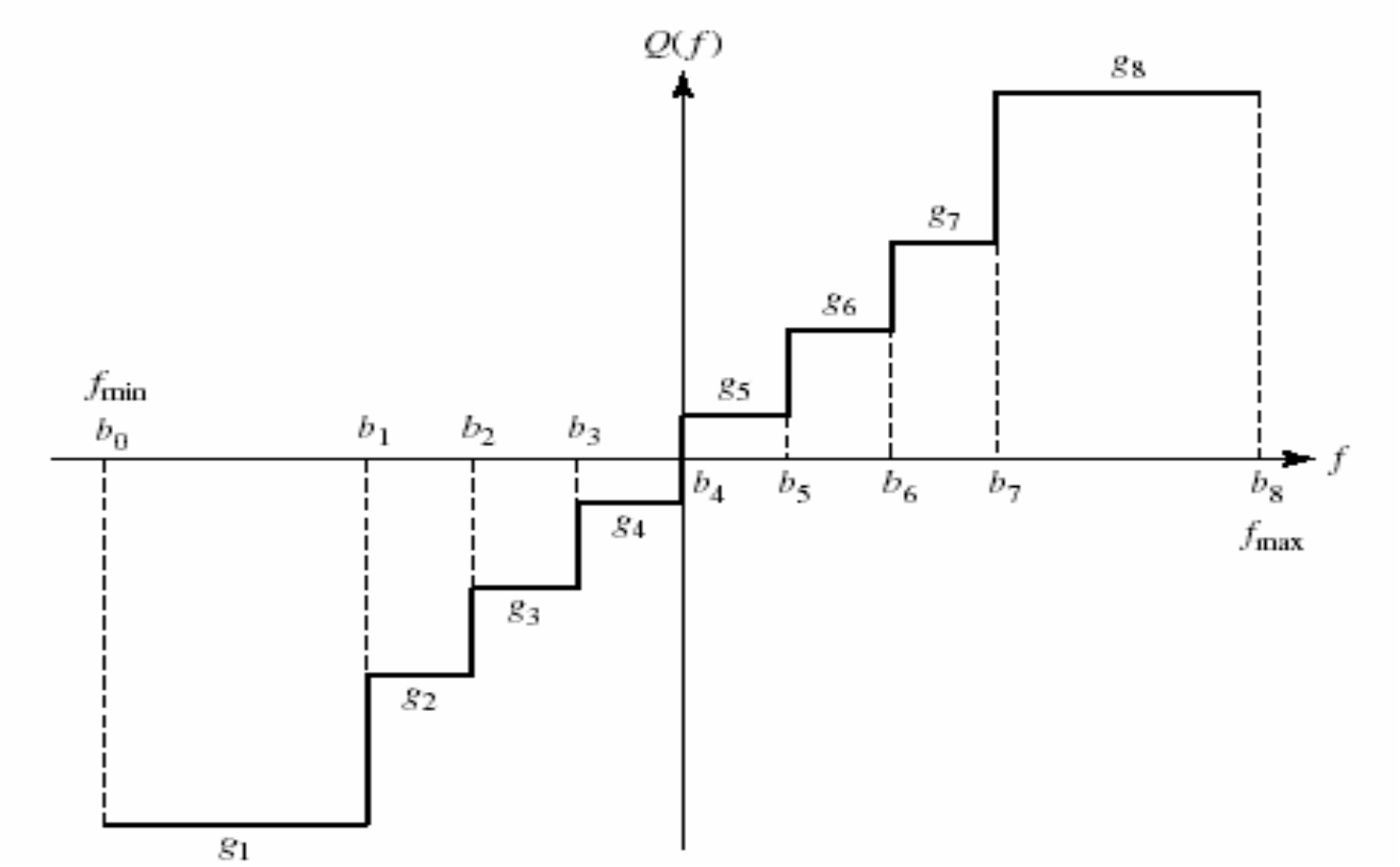
0.05V/0.5V = 10%

- Conclusion
  - Good for high level signals, but bad for low level signals

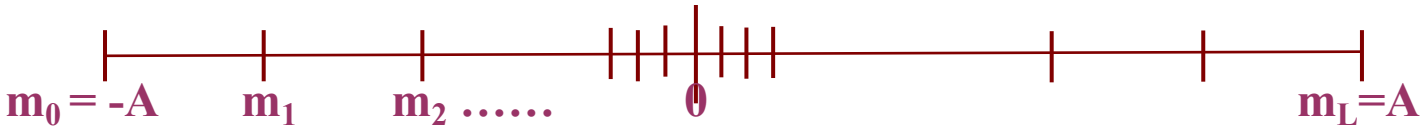
Non-Uniform Quantization

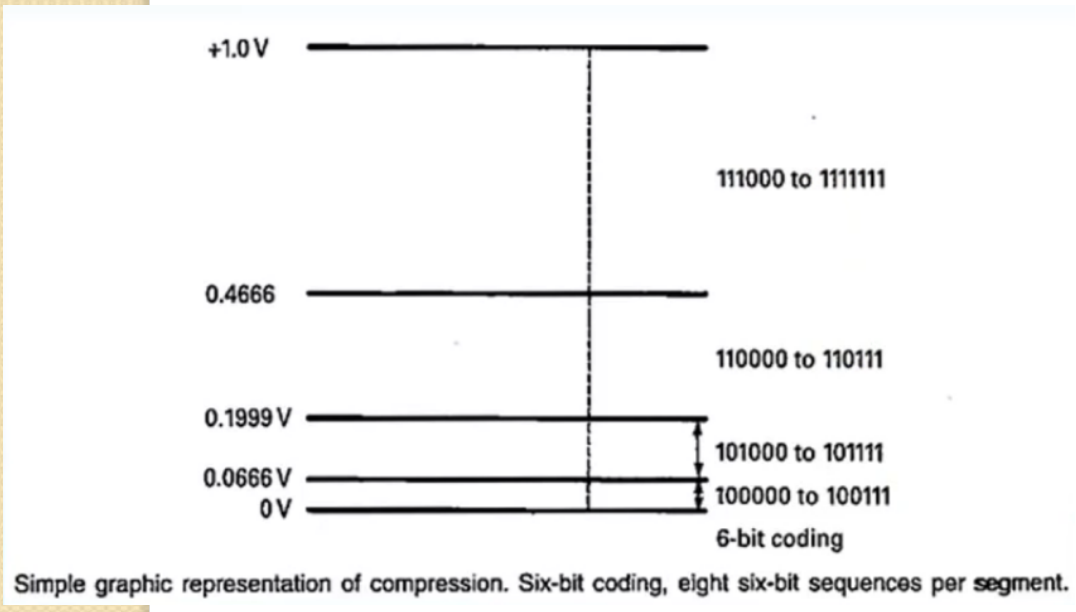
- The quantizing intervals are not of equal size
- Small quantizing intervals are allocated to small signal values (samples) and large quantization intervals to large samples so that the signal-to-quantization distortion ratio is nearly independent of the signal level
- S/N ratios for weak signals are much better but are slightly less for the stronger signals

Non-Uniform Quantization function



Non-Uniform Quantization





Simple graphic representation of compression. Six-bit coding, eight six-bit sequences per segment.

Concept : small quantization levels for small *x*

large quantization levels for large *x*

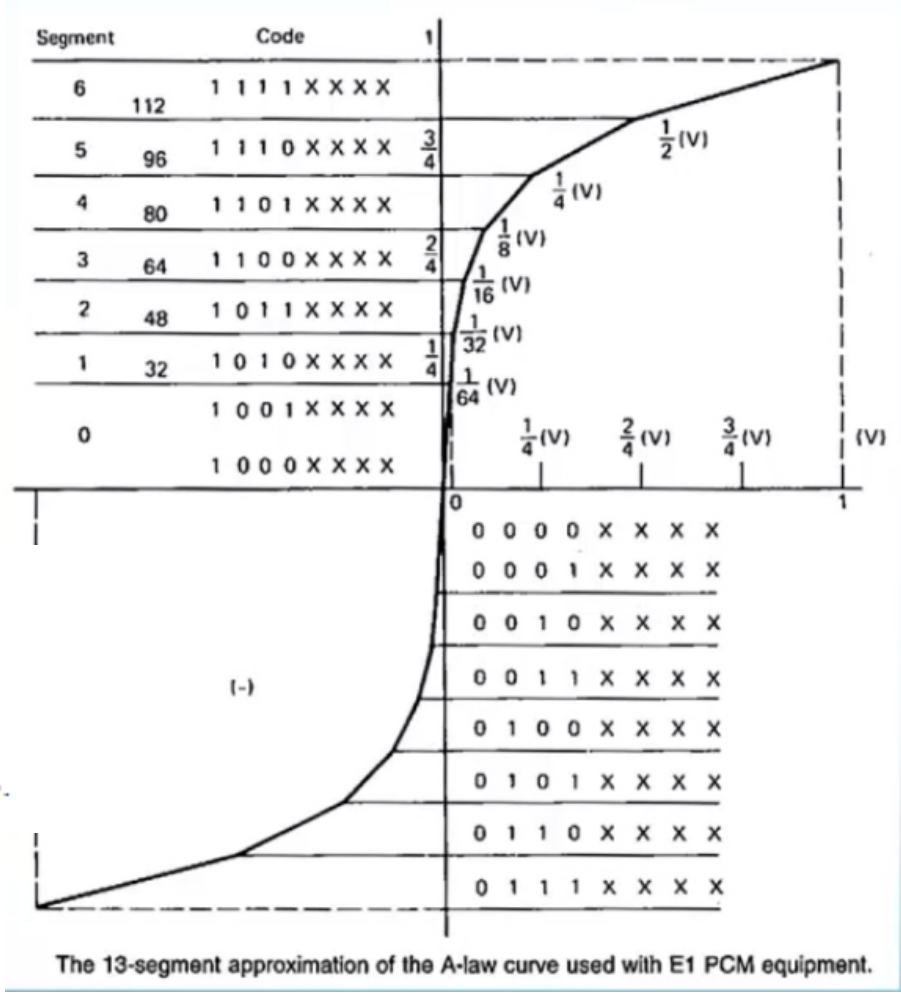
The quantization favors the low level signals

Goal: constant SNR_Q for all *x*

A-law algorithm

$$F(x) = \operatorname{sgn}(x) \begin{cases} \frac{A|x|}{1+\ln(A)}, & |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \le |x| \le 1, \end{cases}$$

where A is the compression parameter. In Europe, $A = 87.6$.

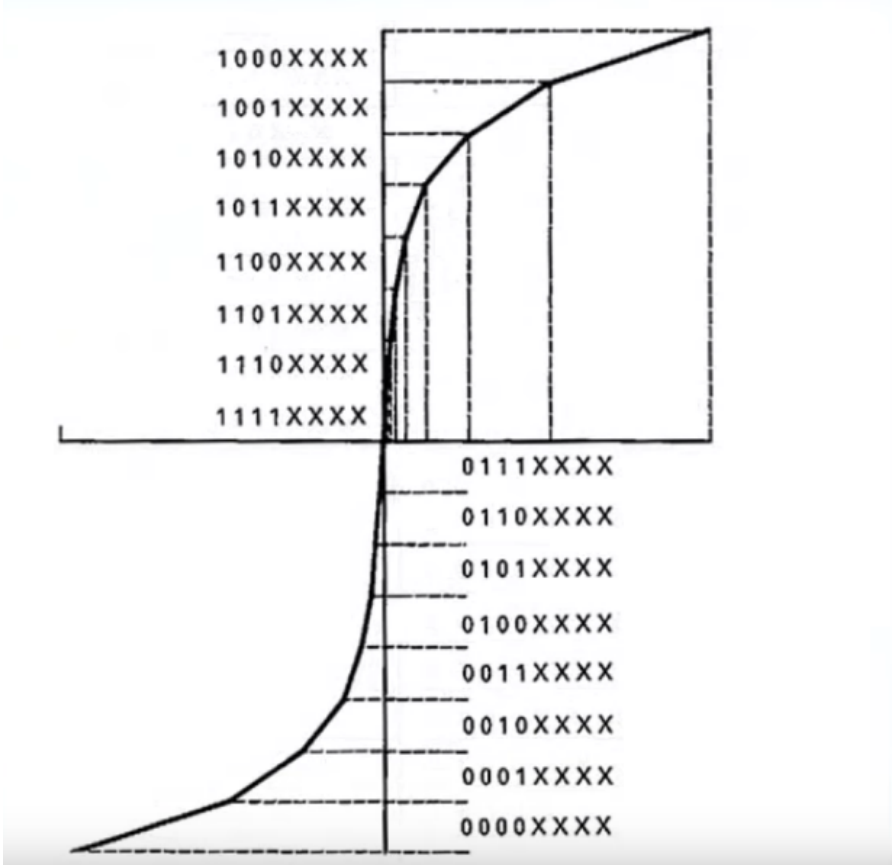


The 13-segment approximation of the A-law curve used with E1 PCM equipment.

The μ-law algorithm

$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \qquad -1 \le x \le 1$$

where $\mu = 255$ in the North American and Japanese standards



Binary encoding

- **Binary encoding**: to represent a finite set of symbols using binary codewords.
- **Fixed length coding**: N levels represented by (int) $\log_2(N)$ bits.
- **Variable length coding (VLC)**: more frequently appearing symbols represented by shorter codewords (Huffman, arithmetic, LZW=zip).
- The minimum number of bits required to represent a source is bounded by its entropy

A simple binary coding

- Assign a binary number to each quantization level
- For example 2's complement, sign magnitude, etc
- Given b bits, then the range of quantization level is 2^b
- If L is the range between max – min, then the q = $\frac{L}{2^b}$

Entropy bound on bitrate (Shannon theory)

- A source with finite number of symbols $\{s_1, s_2, \ldots, s_N\}$
- Symbol s_n has probability (frequency) $P(s_n) = p_n$
- If symbol s_n is given a codeword with l_n bits, the average bitrate (bits/symbol) would be:

$$l_{avg} = \sum p_n * l_n$$

- Average bitrate is bounded by the entropy of the source (H):

$$H \leq l_{avg} \leq H + 1$$

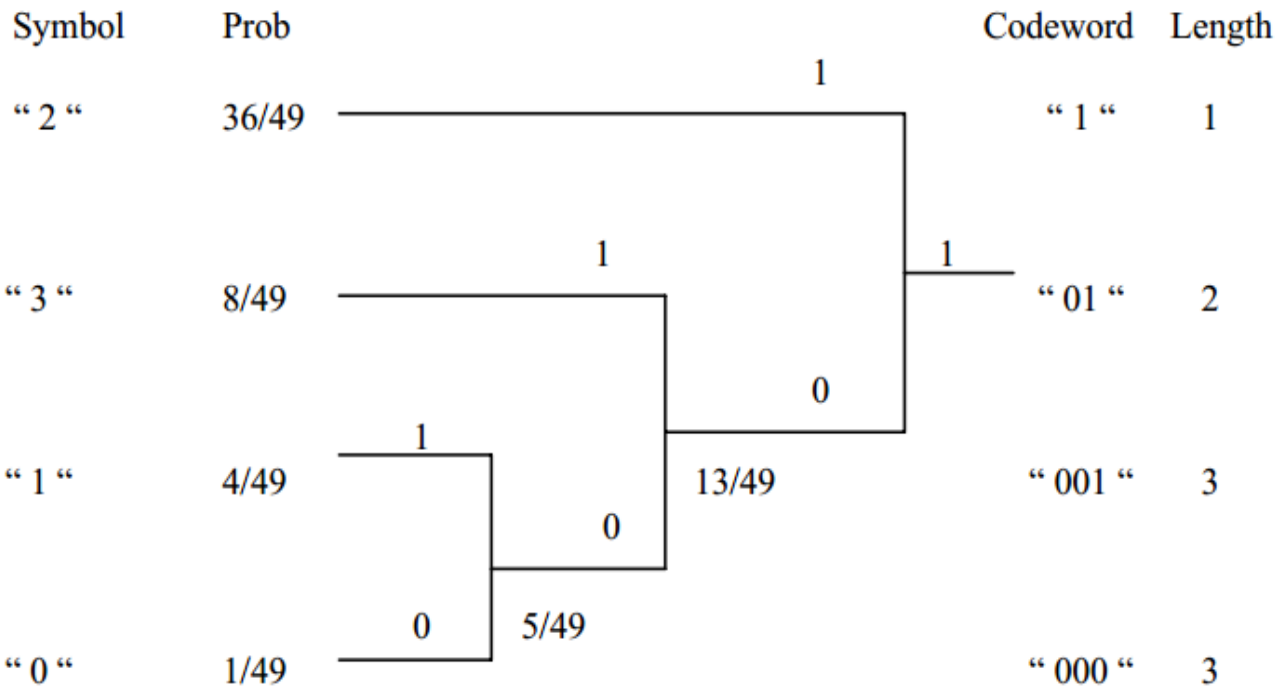
$$H = -\sum p_n * \log_2 p_n$$

- For this reason, variable length coding is also known as entropy coding

Huffman encoding algorithm

- **Step 1**: arrange the symbol probabilities in a decreasing order and consider them as leaf nodes of a tree
- **Step 2**: while there are more than one node:
  - Find the two nodes with the smallest probability and assign the one with the lowest probability a "0", and the other one a "1" (or the other way, but be consistent)
  - Merge the two nodes to form a new node whose probability is the sum of the two merged nodes.
  - Go back to Step 1
- **Step 3**: For each symbol, determine its codeword by tracing the assigned bits from the corresponding leaf node to the top of the tree. The bit at the leaf node is the last bit of the codeword

Huffman encoding example



$$l = \frac{36}{49}\cdot 1 + \frac{8}{49}\cdot 2 + (\frac{4}{49}+\frac{1}{49})\cdot 3 = \frac{67}{49} = 1.4; \quad H = -\sum p_k \log p_k = 1.16.$$

Huffman encoding example (2)

- Huffman encode the sequence of symbols {3,2,2,0,1,1,2,3,2,2} using the codes from previous slide
- Code table:

Symbol	Codeword
0	000
1	001
2	1
3	01

- Coded sequence: {01,1,1,000,001,001,1,01,1,1}
  - Average bit rate: 18 bits/10=1.8 bits/symbol
  - Fixed length coding rate: 2 bits/symbol
  - Saving is more obvious for a longer sequence of symbol
- Decoding: table lookup

More on Huffman encoding

- Huffman coding achieves the upper entropy bound
- One can code one symbol at a time (scalar coding) or a group of symbols at a time (vector coding)
- If the probability distribution is known and accurate, Huffman coding is very good (off from the entropy by 1 bit at most).

Exercise

- Construct a codebook for encoding the message ‘mississippi’?
- What will be the binary coding for a word “mips”?