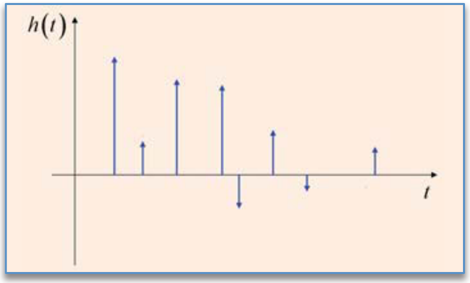
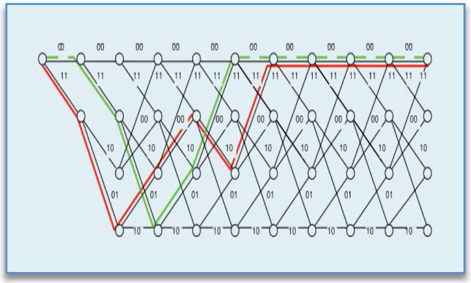
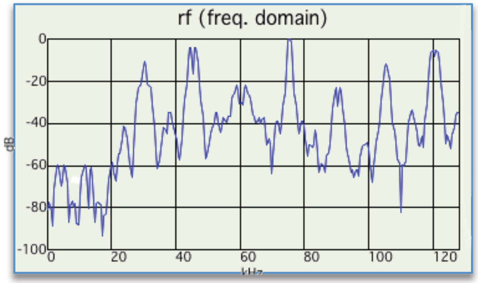
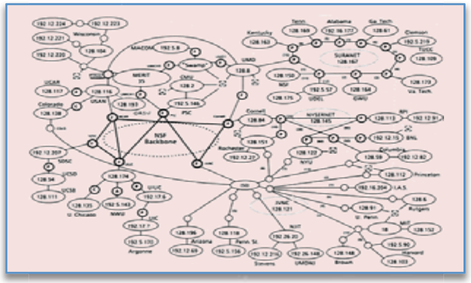








INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Fall 2012
Lecture #1

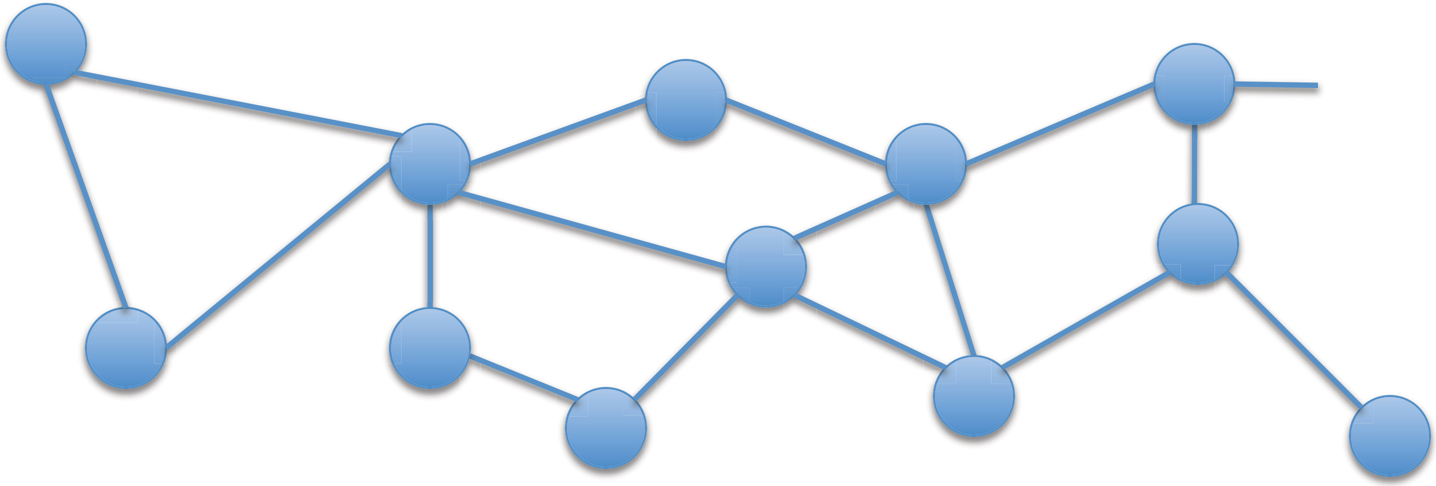
- Digital vs. analog communication
- The birth of modern digital communication
- Information and entropy
- Codes, Huffman coding

Digital vs. Analog Communication

- ANALOG Communicating a continuous-time waveform (e.g., voltage from a microphone), via amplitude modulation (AM) or frequency modulation (FM) or ...
  - Analog electronics
  - Fidelity to the waveform

- DIGITAL Communicating a message comprising a discrete-time sequence of symbols from some source alphabet
  - Often coded onto some other sequence of symbols that's adapted to the communication channel, e.g., binary digits, 0 and 1.
  - Often involving analog communication across the physical channel
  - Fidelity to the message
  - Well suited to riding the staggering growth in computational power, storage, big data, …

6.02 Syllabus



Point-to-point communication channels (transmitter→receiver):

- Encoding information BITS
- Models of communication channels SIGNALS
- Noise, bit errors, error correction
- Sharing a channel

Multi-hop networks:

- Packet switching, efficient routing PACKETS
- Reliable delivery on top of a best-efforts network

Samuel F.B. Morse

- Invented (1832 onwards, patent #1,647 in 1840) the most practical form of electrical telegraphy, including keys, wire arrangements, electromagnets, marking devices, relays, …, and Morse code!

- Worked tirelessly to establish the technology

- After initial struggles, telegraphy was quickly adopted and widely deployed
  - Trans-Atlantic cable attempts 1857 (16 hours to send 98 words from Queen Victoria to President Buchanan!), 1858, 1865, finally success in 1866 (8 words/minute)
  - Trans-continental US in 1861 (effectively ended the Pony Express)
  - Trans-Pacific 1902

- Telegraphy transformed communication (trans-Atlantic time from 10 days by ship to minutes by telegraph) and commerce, also spurred major developments in EE theory & practice (Henry, Kelvin, Heaviside, Pupin, …)

Fast-forward 100 years

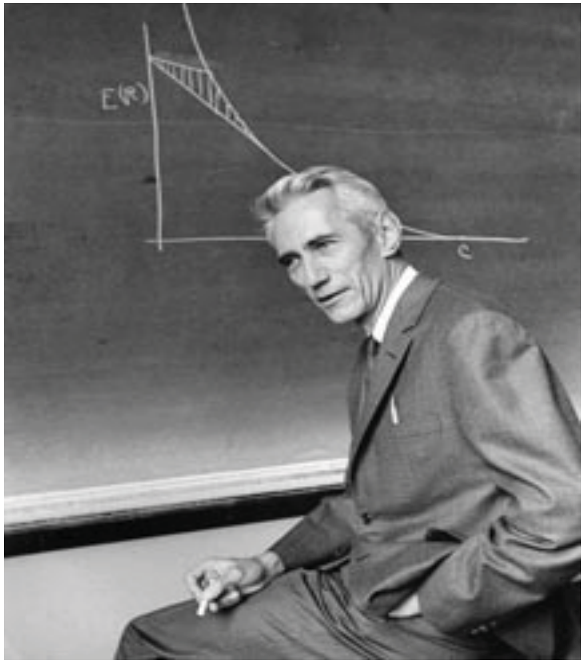
- Via
  - Telephone ("Improvement in Telegraphy", patent # 174,456, Bell 1876)
  - Wireless telegraphy (Marconi 1901)
  - AM radio (Fessenden 1906)
  - FM radio (Armstrong 1933)
  - Television broadcasting by the BBC (1936)
  - …
- Bell Labs galaxy of researchers
  - Nyquist, Bode, Hartley, …

Claude E. Shannon, 1916-2001

1937 Masters thesis, EE Dept, MIT
A symbolic analysis of relay and switching circuits
Introduced application of Boolean algebra to logic circuits, and vice versa. Very influential in digital circuit design. "Most important Masters thesis of the century"

1940 PhD, Math Dept, MIT
An algebra for theoretical genetics
To analyze the dynamics of Mendelian populations.

Joined Bell Labs in 1940.



MIT faculty
1956-1978

"A mathematical theory of cryptography" 1945/1949
"A mathematical theory of communication" 1948

A probabilistic theory requires at least a one-slide checklist on Probabilistic Models!

- Universe **U** of elementary outcomes s_1, s_2, …, s_N, … . One and only one outcome in each experiment or run of the model.
- Events A, B, C, … are subsets of outcomes. We say event A has occurred if the outcome of the experiment lies in A.
- Events form an "algebra" of sets, i.e., A *or* B (union, also written A+B) is an event, A *and* B (intersection, also written AB) is an event, *not* A (complement, also written A^c) is an event. So **U** and the null set **0** are also events.
- Probabilities are defined on events, such that 0 ≤ P(A) ≤ 1, P(**U**)=1, and P(A+B)=P(A)+P(B) if A and B are mutually exclusive, i.e. if AB=**0**. More generally, P(A+B)=P(A)+P(B)-P(AB).
- Events A, B, C, D, E, etc., are said to be (mutually) independent if joint probability of every combination of these events factors into product of individual probabilities, so P(ABCDE)=P(A)P(B)P(C)P(D)P(E), P(ABCD)=P(A)P(B)P(C)P(D), P(ADE)=P(A)P(D)P(E), etc.
- Conditional probability P(A, given that B has occurred) = P(A|B) = P(AB)/P(B).
- Expected value of a random variable is the probability-weighted average.

Measuring Information

Shannon's (and Hartley's) definition of the information obtained on being told the outcome s_i of a probabilistic experiment S :

$$I(S = s_i) = \log_2\left(\frac{1}{p_S(s_i)}\right)$$

where $p_S(s_i)$ is the probability of the event $S = s_i$.

The unit of measurement (when the log is base-2) is the **bit** (**b**inary **i**nformation uni**t** --- not the same as binary digit!).

Image by MIT OpenCourseWare.

1 bit of information corresponds to $p_S(s_i) = 0.5$. So, for example, when the outcome of a *fair* coin toss is revealed to us, we have received 1 bit of information.

"*Information is the resolution of uncertainty*"
Shannon

Examples

We're drawing cards at random from a standard N=52-card deck. Elementary outcome: card that's drawn, probability 1/52, information $\log_2(52/1) = 5.7$ bits.

For an event comprising M such (mutually exclusive) outcomes, the probability is M/52.

Q. If I tell you the card is a spade ♤, how many bits of information have you received?
A. Out of N=52 equally probable cards, M=13 are spades ♤, so probability of drawing a spade is 13/52, and the amount of information received is $\log_2(52/13) = 2$ bits.
This makes sense, we can encode one of the 4 (equally probable) suits using 2 binary digits, e.g., 00=♡, 01=♢, 10=♧, 11=♤.

Q. If instead I tell you the card is a seven, how much info?
A. N=52, M=4, so info = $\log_2(52/4) = \log_2(13) = 3.7$ bits

Properties of Information definition

- A lower-probability outcome yields higher information
- A highly informative outcome does not necessarily mean a more valuable outcome, only a more surprising outcome, i.e., there's no intrinsic value being assessed (can think of information as degree of surprise)
- Often used fact: The information in independent events is additive. (Caution: though independence is sufficient for additivity, it is not necessary, because we can have P(ABC)=P(A)P(B)P(C) even when A,B,C are not independent --- independence requires the pairwise joint probabilities to also factor.)

Expected Information as Uncertainty or Entropy

Consider a discrete random variable S , which may represent the set of possible symbols to be transmitted at a particular time, taking possible values $s_1, s_2, \ldots, s_N$, with respective probabilities $p_S(s_1), p_S(s_2), \ldots, p_S(s_N)$.

The *entropy* $H(S)$ of S is the expected (or mean or average) value of the information obtained by learning the outcome of S :

$$\boxed{H(S) = \sum_{i=1}^{N} p_S(s_i) I(S = s_i)} = \sum_{i=1}^{N} p_S(s_i) \log_2 \left(\frac{1}{p_S(s_i)} \right)$$

When all the $p_S(s_i)$ are *equal* (with value $1/N$), then

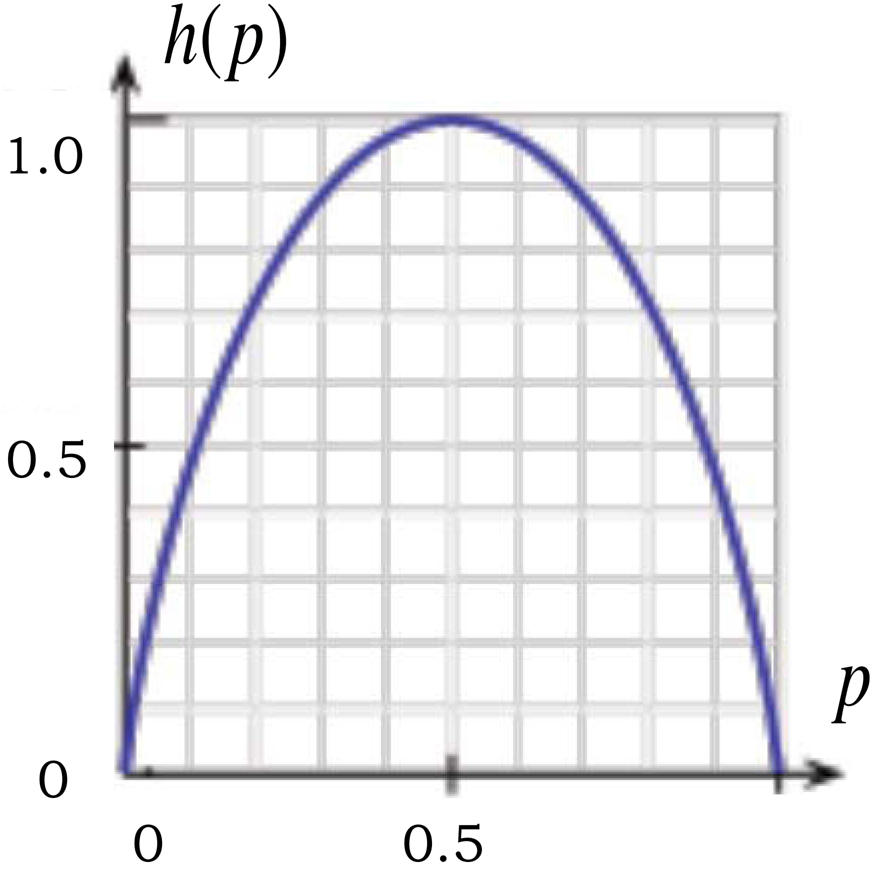
$$H(S) = \log_2 N \qquad \text{or} \qquad \boxed{N = 2^{H(S)}}$$

This is the maximum attainable value!

e.g., Binary entropy function $h(p)$

Heads (or C=1) with probability p

Tails (or C=0) with probability $1-p$



$$H(C) = -p\log_2 p - (1-p)\log_2(1-p) = h(p)$$

Connection to (Binary) Coding

- Suppose p=1/1024, i.e., very small probability of getting a Head, typically one Head in 1024 trials. Then

$$h(p) = (1/1024)\log_2(1024/1) + (1023/1024)\log_2(1024/1023)$$

$= .0112$ bits of uncertainty or information per trial on average

- So using 1024 binary digits (C=0 or 1) to code the results of 1024 tosses of this particular coin seems inordinately wasteful, i.e., 1 binary digit per trial. Can we get closer to an average of .0112 binary digits/trial?

- Yes!

Confusingly, a binary digit is also referred to as a bit!

- Binary coding: Mapping source symbols to **binary digits**

Significance of Entropy

Entropy (in bits) tells us the average amount of information (in bits) that must be delivered in order to resolve the uncertainty about the outcome of a trial. This is a lower bound on the number of binary digits that must, on the average, be used to encode our messages.

If we send fewer binary digits on average, the receiver will have some uncertainty about the outcome described by the message.

If we send more binary digits on average, we're wasting the capacity of the communications channel by sending binary digits we don't have to.

Achieving the entropy lower bound is the "gold standard" for an encoding (at least from the viewpoint of information compression).

Fixed-length Encodings

An obvious choice for encoding equally probable outcomes is to choose a fixed-length code that has enough sequences to encode the necessary information

- 96 printing characters → 7-"bit" ASCII
- Unicode characters → UTF-16
- 10 decimal digits → 4-"bit" BCD (binary coded decimal)

Fixed-length codes have some advantages:

- They are "random access" in the sense that to decode the n^{th} message symbol one can decode the n^{th} fixed-length sequence without decoding sequence 1 through n-1.
- Table lookup suffices for encoding and decoding

Now consider:

$choice_i$	p_i	$log_2(1/p_i)$
"A"	1/3	1.58 bits
"B"	1/2	1 bit
"C"	1/12	3.58 bits
"D"	1/12	3.58 bits

The expected information content in a choice is given by the entropy:

= (.333)(1.58) + (.5)(1) + (2)(.083)(3.58) = 1.626 bits

Can we find an encoding where transmitting 1000 choices requires 1626 binary digits on the average?

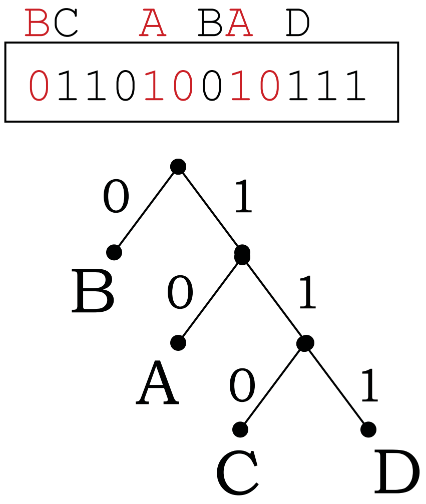
The "natural" fixed-length encoding uses two binary digits for each choice, so transmitting the results of 1000 choices requires 2000 binary digits.

Variable-length encodings

(David Huffman, in term paper for MIT graduate class, 1951)

Use shorter bit sequences for high probability choices, longer sequences for less probable choices

choice$_i$	p_i	*encoding*
"A"	1/3	10
"B"	1/2	0
"C"	1/12	110
"D"	1/12	111



Huffman Decoding Tree

Expected length

=(.333)(2)+(.5)(1)+(2)(.083)(3)

= 1.666 bits

Transmitting 1000 choices takes an average of 1666 bits… better but not optimal

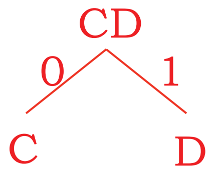
Note: The symbols are at the leaves of the tree; necessary and sufficient for instantaneously decodability.

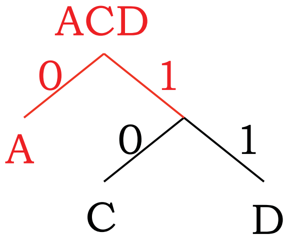
Huffman’s Coding Algorithm

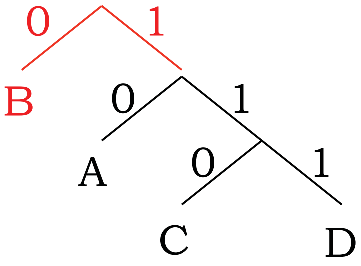
- Begin with the set S of symbols to be encoded as binary strings, together with the probability p(s) for each symbol s in S.
- Repeat the following steps until there is only 1 symbol left in S:
  - Choose the two members of S having lowest probabilities. Choose arbitrarily to resolve ties.
  - Remove the selected symbols from S, and create a new node of the decoding tree whose children (sub-nodes) are the symbols you've removed. Label the left branch with a “0”, and the right branch with a “1”.
  - Add to S a new symbol that represents this new node. Assign this new symbol a probability equal to the sum of the probabilities of the two nodes it replaces.

Huffman Coding Example

- Initially S = { (A, 1/3) (B, 1/2) (C, 1/12) (D, 1/12) }
- First iteration
  - Symbols in S with lowest probabilities: C and D
  - Create new node
  - Add new symbol to S = { (A, 1/3) (B, 1/2) (CD, 1/6) }
- Second iteration
  - Symbols in S with lowest probabilities: A and CD
  - Create new node
  - Add new symbol to S = { (B, 1/2) (ACD, 1/2) }
- Third iteration
  - Symbols in S with lowest probabilities: B and ACD
  - Create new node
  - Add new symbol to S = { (BACD, 1) }
- Done







Another Variable-length Code (not!)

Here's an alternative variable-length for the example on the previous page:

Letter	*Encoding*
A	0
B	1
C	00
D	01

Why isn't this a workable code?

The expected length of an encoded message is

$$(.333+.5)(1) + (.083 + .083)(2) = 1.22 \text{ bits}$$

which even beats the entropy bound ☺

Huffman Codes – the final word?

- Given static symbol probabilities, the Huffman algorithm creates an optimal encoding when each symbol is encoded separately. (optimal ≡ no other encoding will have a shorter expected message length). It can be proved that

  expected length L satisfies $H \leq L \leq H+1$

- Huffman codes have the biggest impact on average message length when some symbols are substantially more likely than other symbols.

- You can improve the results by adding encodings for symbol pairs, triples, quads, etc. From example code:
-

  Pairs: 1.646 bits/sym, Triples: 1.637, Quads 1.633, …

- But the number of possible encodings quickly becomes intractable.

- Symbol probabilities change message-to-message, or even within a single message. Can we do adaptive variable-length encoding?