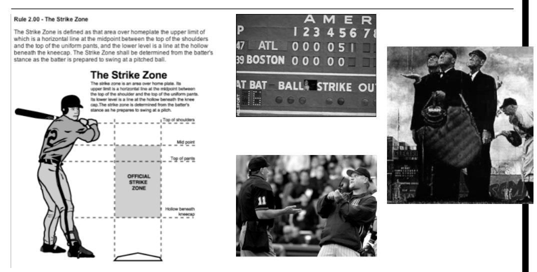
Perfect Bits

Analog = Continuous, Digital = Discrete

States of a Digital System

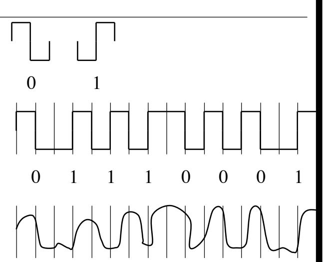
- Digital states represent analog reality
- Digital state is an abstraction, with "irrelevant" detail ignored
- $60.00 vs. 

What Digital State for that Analog Reality? Discretization



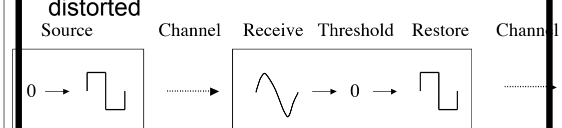
Ideal Bits vs. Real Bits

- Ideal 0 and 1
  (Manchester Coding)
- Ideal 01110001
- Reality



Restoration

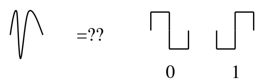
- If we know a signal must represent 0 or 1, it can be restored if it has not been too distorted



Pulse code modulation

But what about threshold ambiguities?

- Sometimes there is just too much noise



Error-Detecting and -Correcting Codes

- Add extra bits to the data bits for the sole purpose of detecting when errors have been introduced and correcting the errors if possible

Repetition Code

- Repeat every bit 3 times
- 0110 ==> 000,111,111,000
- Error detected if all 3 bits are not the same
- 000,110,111,000

Analysis of Repetition Code

- 200% overhead (code is 3x size of data)
- Error can go undetected only if 3 consecutive bits are in error
- 0110 => 000,111,111,000 => 000,000,111,000
- If probability of one-bit error is p, then probability of undetected error is p^3
- E.g. one-bit error = 10^{-5} => undetected error = 10^{-15}
- (Assumes independence)

What if an Error is Detected?

- Strategy #1: Throw the data out and ask for it to be sent again
- Slow but very low odds of erroneous data
- Strategy #2: Majority rules
- 0110 => 000,111,111,000 => 000,110,111,000 => 000,111,111,000
- Quicker, but higher odds of error
- Was it actually 0010 => 000,000,111,000 => 000,110,111,000?

Parity Codes

- Add one bit to every block of, say, 4 bits
- Parity bit = 0 or 1 so that total number of 1 bits is even
- Detects all 1-bit errors, no 2-bit errors

00000	00011	00101	00110
01001	01010	01100	01111
10001	10010	10100	10111
11000	11011	11101	11110

Hamming Codes
(Richard W. Hamming, 1915-1998)

- (4,7) Hamming Code detects all one- and two-bit errors
- Corrects all 1-bit errors
- Magic: Any two different codewords differ in at least 3 places!

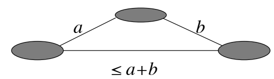
0000000	0001011	0010111	0011100
0100110	0101101	0110001	0111010
1000101	1001110	1010010	1011001
1100011	1101000	1110100	1111111

Hamming Distance

- Number of places in which two bit strings differ
- = Hamming distance 3

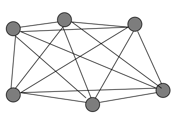
1	0	0	0	1	0	1
1	0	0	1	1	1	0

- Acts like a distance:

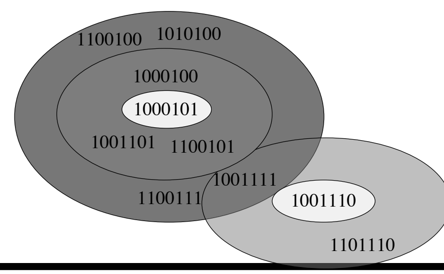


Error Correcting Codes

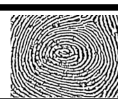
- ECC design is a kind of geometry problem: Find 16 bit strings of length 7, no two of which are separated by distance less than 3



Hamming Distance as Geometry



Fingerprinting Data



- How to check quickly if data are corrupted?
- Transmit large data packet + small fingerprint computed from the data packet
- Take fingerprint of received data and see if it matches transmitted fingerprint
- Match ==> uncorrupted data *with high probability but not certainty*

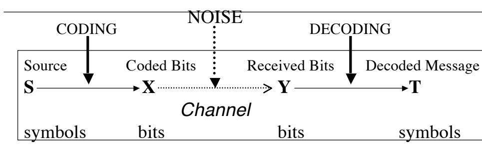
Idea of Cyclic Redundancy Check

- Data packet is, say, 1KB = 8192 bits
- Treat it as one 8192-bit binary numeral
- Divide this number by some big constant K
- Fingerprint is the remainder, $0 \leq r < K$
- Transmit packet and the value of *r*
- At other end compute fingerprint dividing by K and compare remainder to *r*

Analysis of Cyclic Redundancy Check

- Suppose K is a 100-bit number
- If K is well-chosen, probability of false negative (fingerprints match but error in the data) is only 1/K or around 2^{-100}
- No possibility of false positive
- So adding a 13-byte fingerprint to the 1000 byte packet lowers the odds of an undetected error to much less than once in the lifetime of the universe

Shannon's Model



• Shannon's source coding theorem was that the source can be coded so that the number of bits per symbol is as close as we wish to the entropy of the source, but no less

• Shannon's channel coding theorem has to do with reducing the likelihood of errors in the presence of noise

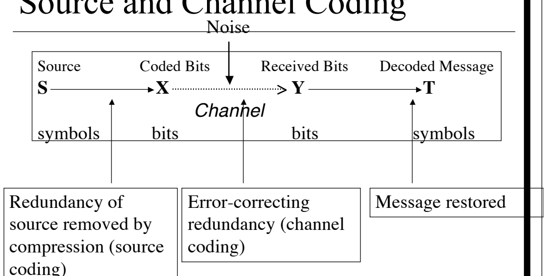
Shannon's Channel Coding Theorem

- For any channel there is a *channel capacity,* a certain number *C* of bits/second
- As long as the source is producing less than *C* bits per second, messages can be coded so they will be received at the other end of the channel with arbitrarily low probability of error
- If the source is producing bits at a rate higher than *C* bits/second, it is impossible to transmit bits with low probability of error

The Surprise of the Channel Coding Theorem

- Until Shannon, people thought that the only way to lower the error rate was to slow down the source
- Shannon showed that the channel capacity is an absolute measure of the rate at which bits can be transmitted *correctly* through the channel
- Absolute assurance of correctness is never possible but with more complex codes one can come as close as one wants to guaranteed correctness

Source and Channel Coding



Source Coding and Channel Coding