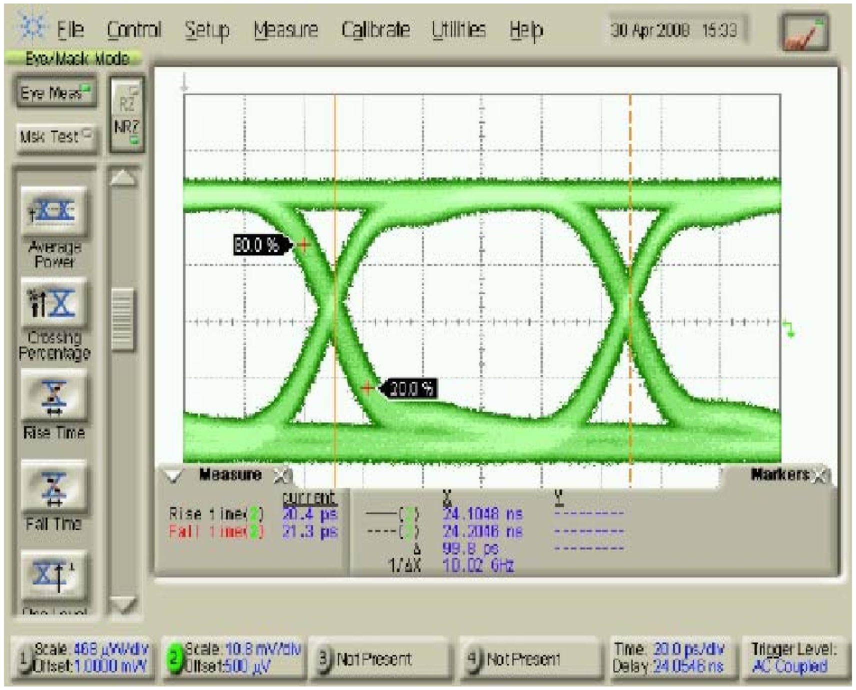
ELEC1200: A System View of Communications: from Signals to Packets
Lecture 10

- **Error Correction**
- **Codewords**
- **Hamming Distance**
- **Error Detection: Adding a Parity bit**
- **Single-bit Error Correction**

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

There's good news and bad news…

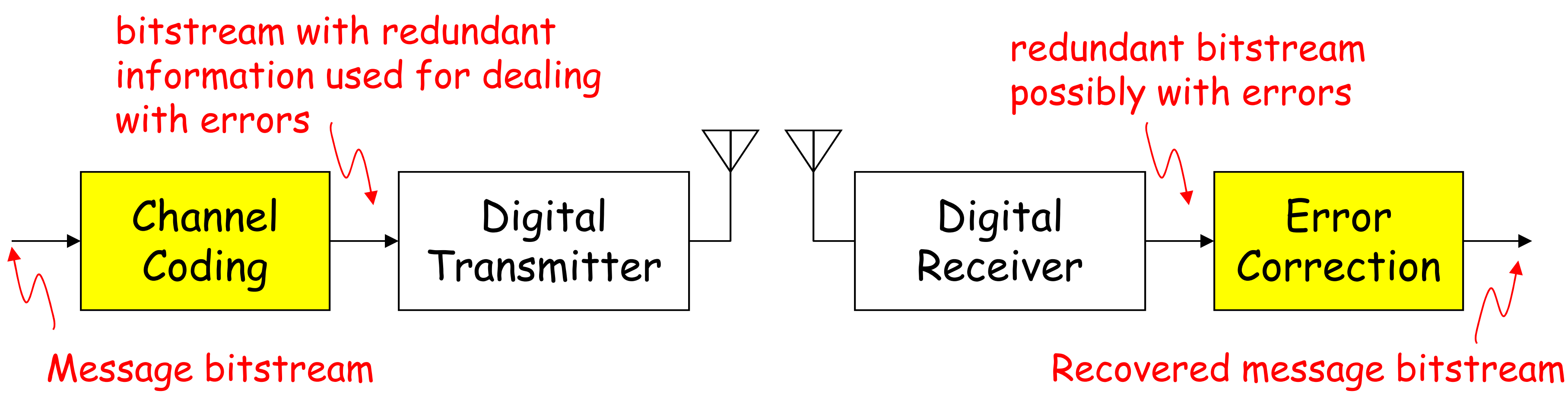


The good news: Our digital modulation scheme usually allows us to recover the original signal despite small amplitude errors introduced by the components and channel.

The bad news: larger amplitude errors (hopefully infrequent) that change the signal irretrievably. These show up as bit errors in our digital data stream.

Channel coding

Our plan to deal with bit errors:

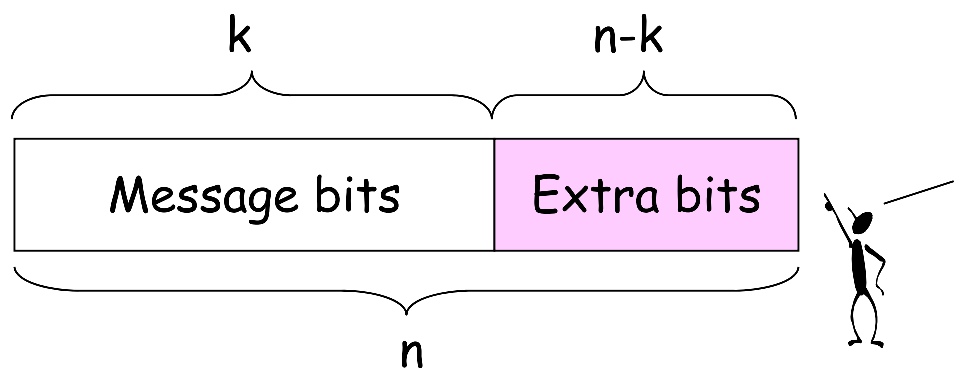


We'll add redundant information to the transmitted bit stream (a process called channel coding) so that we can detect errors at the receiver. Ideally we'd like to correct commonly occurring errors, e.g., error bursts of bounded length. Otherwise, we should detect uncorrectable errors and use, say, retransmission to deal with the problem.

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

(n,k) Block Codes

- **Split message into *k*-bit blocks**
- **Add (*n-k*) extra bits to each block, making each block *n* bits long.**



The entire block is called a "codeword"

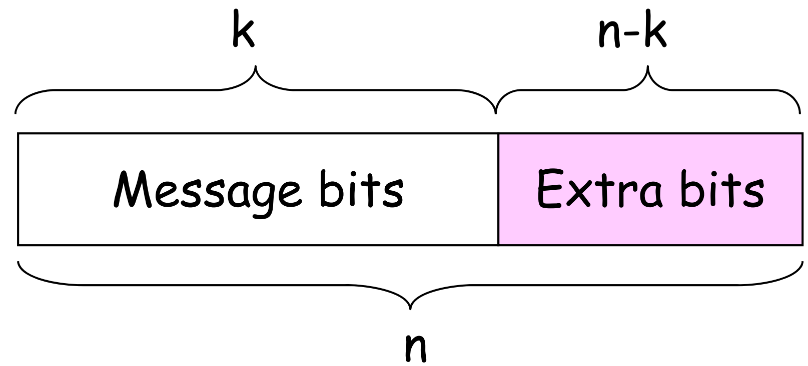
- **By adding a few extra bits, we can *detect* when an error has occurred.**
- **By adding even more bits, we can actually *correct* the errors.**

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Code Rate

- The <u>code rate</u> indicates what fraction of the bits that are sent actually contain useful information (i.e. the message). We want this to be close to one.
- For the (n,k) block code,

$$\text{code rate} = \frac{k}{n}$$



- Related terms
  - The <u>gross bitrate</u> is the rate at which bits (useful or not) are sent. Also called the or data signaling rate.
  - The <u>net bitrate</u> is the rate at which useful bits are sent.

$$\text{gross bitrate} = \frac{F_s}{SPB} \qquad \text{net bitrate} = \frac{k}{n} \cdot \frac{F_s}{SPB}$$

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

More good news, bad news…

Shannon's Noisy Channel Coding Theorem:

A noisy channel has a channel capacity C, which is the maximum rate at which useful information can be transmitted. This capacity depends upon the physical properties of the channel.

For any rate $R < C$ and for large enough N, there exists a code of length N such that the probability of error (after error correction) is arbitrarily small.

- **Good news:**
  - **theoretically it is possible to transmit information *without error* at any rate below C**
- **Bad news:**
  - **the proof doesn't tell us how to construct the error-correcting code!**

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Claude Elwood Shannon

Claude Elwood Shannon
Father of Information Theory

Electrical engineer, mathematician, and native son of Gaylord. His creation of information theory, the mathematical theory of communication, in the 1940s and 1950s inspired the revolutionary advances in digital communications and information storage that have shaped the modern world.

This statue was donated by the
Information Theory Society of the Institute of Electrical and Electronics Engineers,
whose members follow gratefully in his footsteps.

Dedicated October 6, 2000
Eugene Daub, Sculptor

Channel Coding

1. Take an input message stream: **011011101101**
2. Break the message stream into k-bit blocks (e.g. k = 4).

   0110
   1110
   1101

3. Add (n-k) parity bits to form n-bit codeword (e.g. n = 8)

   01101111
   11100101
   11010110

4. Transmit data through noisy channel and receive codewords with some errors

   01100111
   11110101
   11000110

5. Perform error correction
6. Extract the k=4 message bits from each corrected codeword.

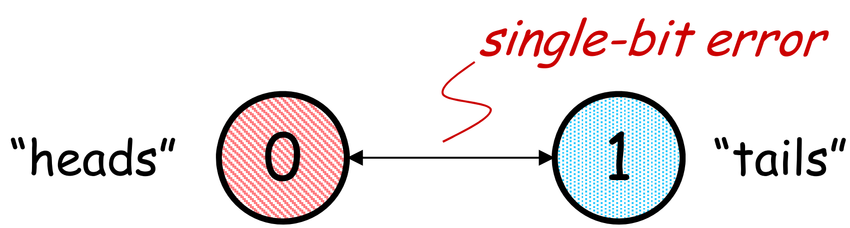
   0110 1110 1101

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Hamming Distance

(Richard Hamming, 1950)

- The Hamming Distance between two codewords is the number of bit positions where the corresponding bits are different.
- For example,
  - The Hamming distance between (00) and (10) is 1.
  - What is the Hamming distance between (00000000) and (11110011)?
- The Hamming distance
  - Measures the distance between two codewords
  - Indicates how many bit errors it takes transform one codeword to another.
- If we use no coding for a 1 bit block, the two code words are ("0" and "1") . Since the Hamming distance is 1, a single-bit error changes one code word the other.



Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

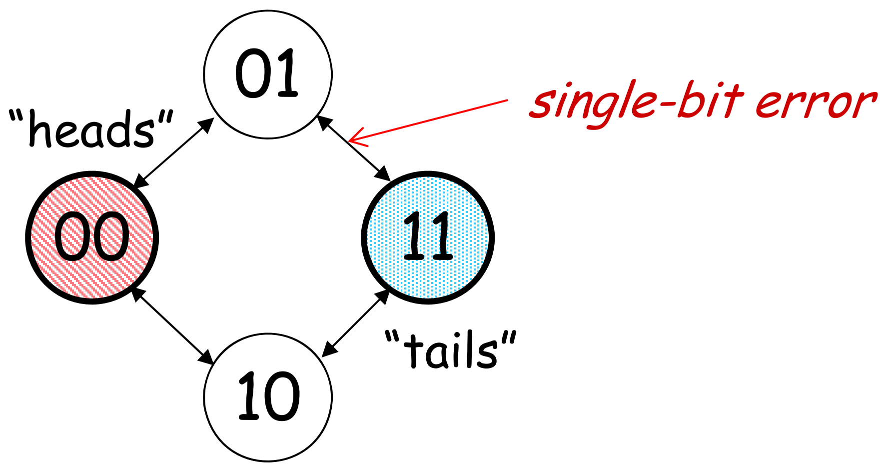
Detection vs Correction

- **Error detection means that we can detect that an error has occurred. However, we don't know how to fix it.**
- **Error correction means that we can correct the errors that we detect.**

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Error <u>Detection</u> in a (2,1) block code

- In order to detect single bit errors, we need an encoding where a single-bit error doesn't produce another valid code word. In other words, the Hamming distance between code words must be at least 2.



code rate = $\frac{1}{2}$

- In the diagram above, each code word has an even number of "1" bits. We refer to this as "even parity".
- There is an error if the number of received "1" bits in the code word is odd.

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Error Detection by adding a parity bit

- A parity bit can be added to any length message and is chosen to make the total number of "1" bits even.
- To check for a single-bit error (actually any odd number of errors), count the number of "1"s in the received word. If it's odd, there's been an error.

parity bit

- 0 1 1 0 0 1 0 → original word
  0 1 1 0 0 1 0 1 → codeword = original word with parity
  0 1 1 0 0 0 0 1 → single-bit error (detected)

code rate = $\frac{7}{8}$

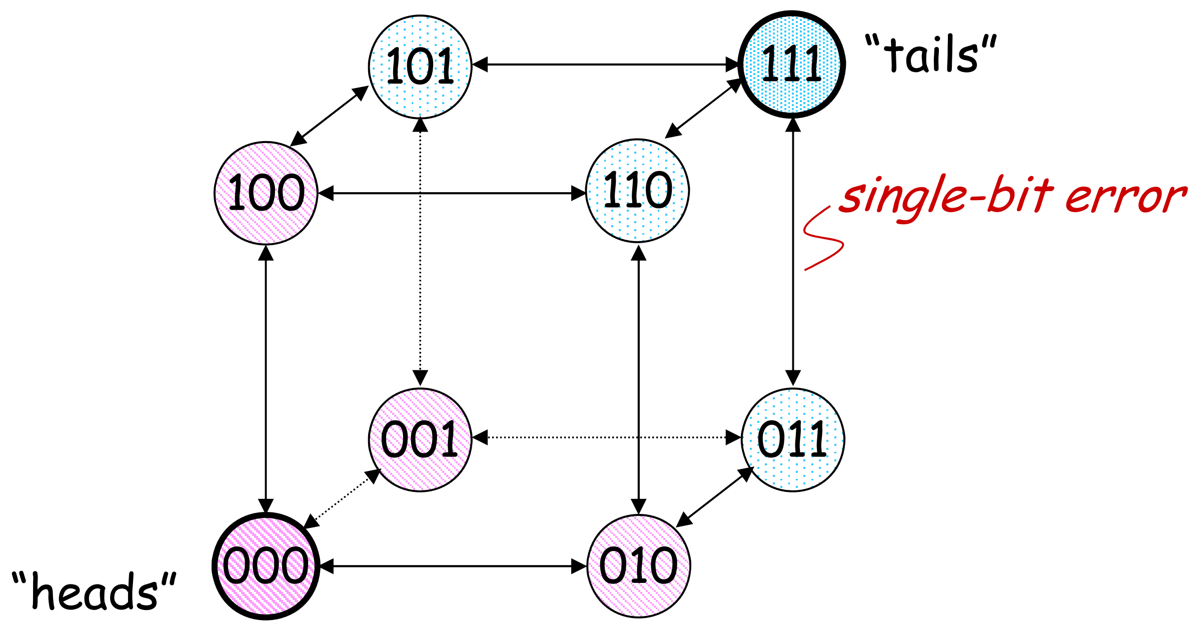
- 1 0 1 1 1 0 0 → original word
  1 0 1 1 1 0 0 0 → codeword = original word with parity
  1 0 1 1 1 1 1 0 → 2-bit error (not detected)

(8,7) code

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Error <u>Correction</u> in a (3,1) block code

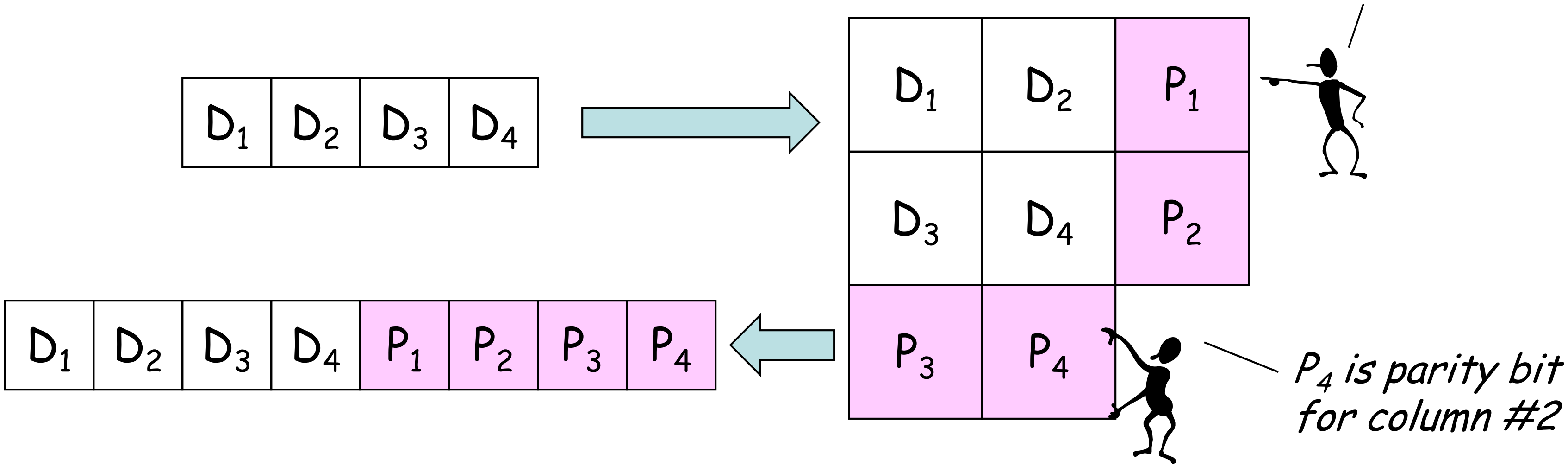
- By increasing the Hamming distance between valid code words to 3, we guarantee that the sets of words produced by single-bit errors don't overlap. So if we detect an error, we can perform *error correction* since we can tell what the valid code was before the error happened.
- Do we always need to triple the number of bits?



code rate = $\frac{1}{3}$

A simple (8,4) code

P_1 is parity bit for row #1



- **The parity bits (P) are computed by first arranging the message bits (D) as shown, and then adding a parity bit to each each row or column so that it has even parity.**
- **For example,**

Message block: 0 1 1 1

Codeword: 0 1 1 1 1 0 1 0

0	1	1
1	1	0
1	0	

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Performing Error Correction

- Take the code word, and rearrange it

D_1	D_2	D_3	D_4	P_1	P_2	P_3	P_4

D_1	D_2	P_1
D_3	D_4	P_2
P_3	P_4	

- Add a syndrome bit to each row/column to make it even parity

D_1	D_2	P_1	S_1
D_3	D_4	P_2	S_2
P_3	P_4		
S_3	S_4		

- Check the syndrome bits.
  - If all are 0, there is no error
  - If only syndrome bits for column i and row j are 1 (e.g. S_1 and S_3), there is a bit error in the data bit at i,j (e.g. D_1)
  - If only one syndrome bit is 1 (e.g. S_2), there is an error in the corresponding parity bit (e.g. P_2).

Examples of Error Correction

Received Codeword	Rearranged Codeword	Computed Syndrome	Corrected Data
01111010	0 1 1 1 1 0 1 0	0 1 1 0 1 1 0 0 1 0 0 0	0111 no errors
01101010	0 1 1 1 0 0 1 0	0 1 1 0 1 0 0 1 1 0 0 1	0111 D_4 incorrect
01111110	0 1 1 1 1 1 1 0	0 1 1 0 1 1 1 1 1 0 0 0	0111 P_2 incorrect

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Channel Coding

1. Take an input message stream:
2. Break the message stream into k-bit blocks (e.g. k = 4).
3. Add (n-k) parity bits to form n-bit codeword (e.g. k = 4)
4. Transmit data through noisy channel and receive codewords with some errors
5. Perform error correction
6. Extract the k=4 message bits from each corrected codeword.

011011101101

↓

0110
1110
1101

↓

01101111
11100101
11010110

↓

0110<u>0</u>111
111<u>1</u>0101
110<u>0</u>0110

↓

0110 1110 1101

Some content taken with permission from material for the course EECS6.02 by C. Sodini, M. Perrot and H. Balakrishnan at MIT

Summary

- Noise is always present in communication systems and this leads to bit errors
- Error Correcting Codes (ECC) can be used to reduce BER
- (n,k) Block codes are one type of ECC that can be used
- Minimum Hamming distance between codewords in the code can be used to measure how many bit errors the code can correct and detect