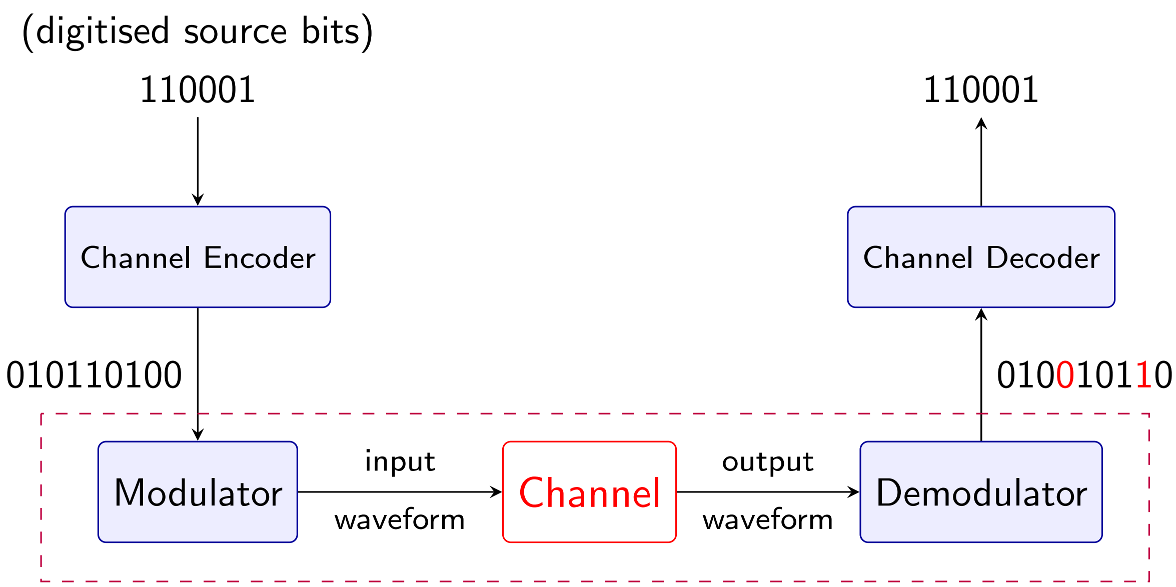
1B Paper 6: Communications

Handout 6: Channel Coding

Ramji Venkataramanan

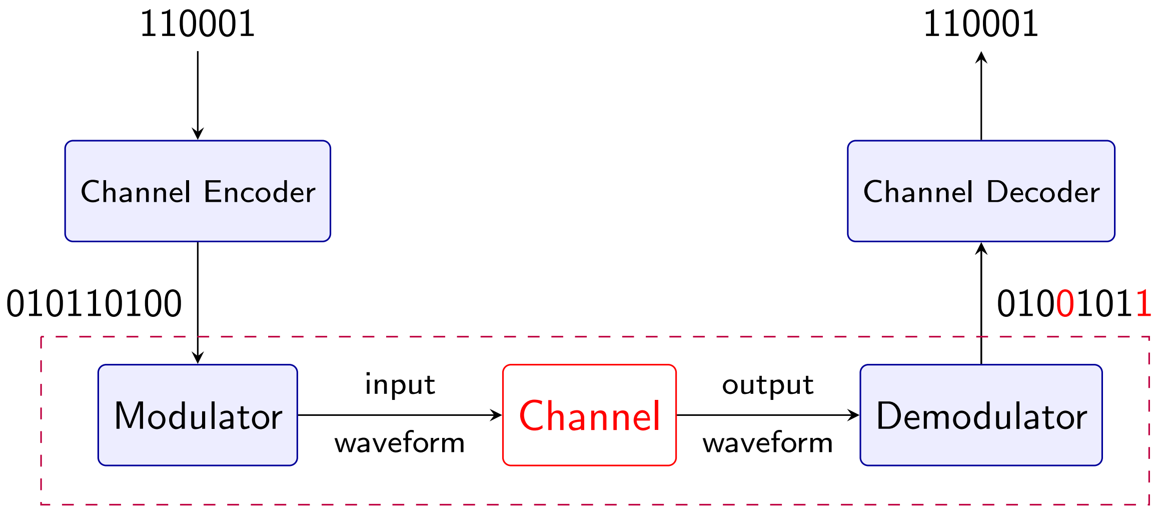
Signal Processing and Communications Lab
Department of Engineering
`ramji.v@eng.cam.ac.uk`

Lent Term 2016



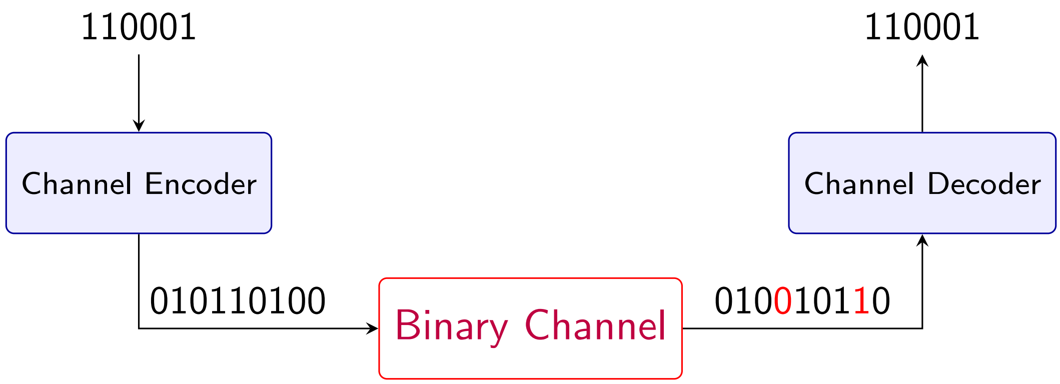
- So far, we focused on the mod & demod blocks, and studied two modulation schemes – PAM and QAM
- We also calculated the probability of symbol error for some of these schemes
- Thus, for a *fixed* modulation scheme (e.g. QPSK), we can estimate the probability that that a bit will be in error at the output of the demodulator/detector

Binary Channel



- Every modulation scheme has an associated *probability of bit error*, say p, that we can estimate theoretically or empirically
- For a fixed modulation scheme, the part of the system enclosed by dashed lines can thus be considered an overall binary channel with bit error probability p

Thus an equivalent representation of the communication system for a fixed modulation scheme is



If the modulation scheme has a bit error probability p:

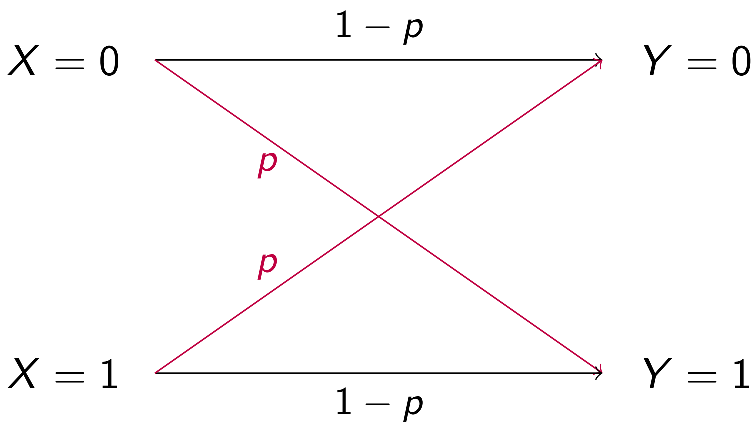
- A 0 input is flipped by the binary channel to a 1 with probability p
- A 1 input is flipped by the binary channel to a 0 with probability p

It is important to remember that the binary channel

- *Is not* the actual physical channel in the communication system
- *Is* the overall channel assuming that the modulation scheme is fixed and we have estimated its bit error probability p

Binary Symmetric Channel (BSC)

As the binary channel flips each bit (0/1) with equal probability p, it is called a Binary Symmetric Channel. Represented as:

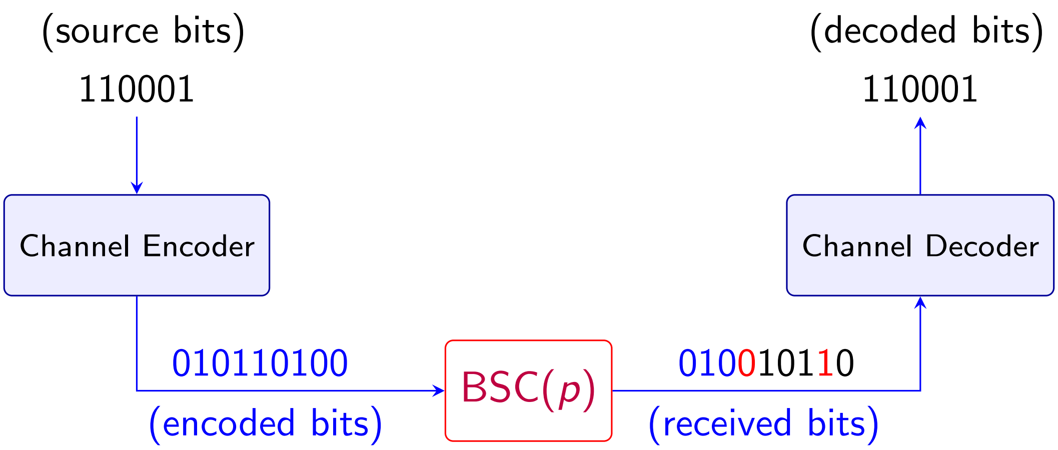


$$P(Y=0|X=0) = 1-p, \quad P(Y=1|X=1) = 1-p$$
$$P(Y=1|X=0) = p, \quad P(Y=0|X=1) = p$$

p is the "crossover probability"; the channel is called BSC(p)

Channel Coding

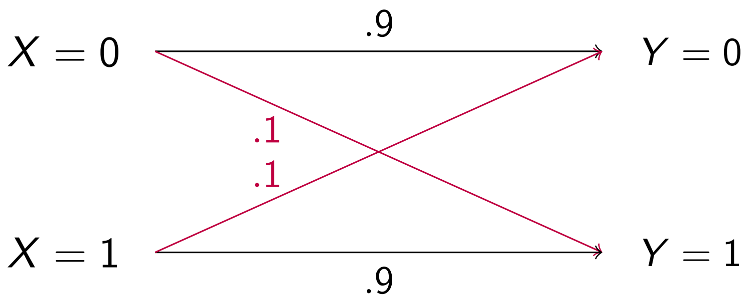
Thus the system is now:



We will now study *channel coding*, which consists of

- Encoding: Adding redundancy to the source bits in a *controlled* manner
- Decoding: Recovering the source bits from the noisy bits by exploiting the redundancy

Repetition Code



The simplest channel code for the BSC is a $(n, 1)$ repetition code:

- Encoding: Simply repeat each source bit n times (n is odd)
- Decoding: By "majority vote". Declare 0 if greater than $n/2$ of the received bits are 0, otherwise decode 1

Example: $(3, 1)$ Repetition Code

Source bits:	0	1	1	0	0 ...
Encoded bits:	000	111	111	000	000 ...
Received bits:	001	101	111	011	000 ...
Decoded bits:	0	1	1	1	0 ...

Decoding Errors and Data Rate

Q: With a $(3, 1)$ repetition code, when is a decoded bit in error ?

A: When the channel flips two or more of the three encoded bits

The probability of decoding error when this code is used over a BSC(0.1) is $\binom{3}{2}(.1)^2(.9) + \binom{3}{3}(.1)^3 = 0.028$

The **rate** of the code is $\frac{1}{3}$ (3 encoded bits for each source bit)

Q: With a $(5, 1)$ repetition code, when is a decoded bit in error ?

A: When the channel flips three or more of the five encoded bits

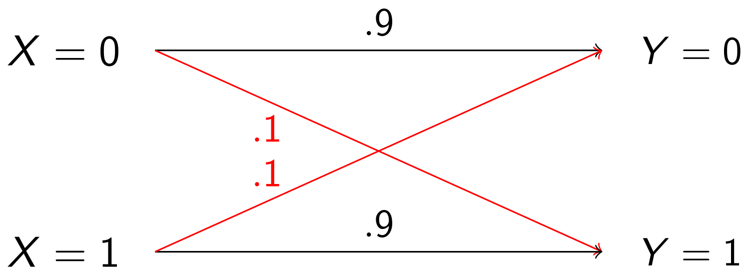
The probability of decoding error is 0.0086 (Ex. Paper 9, Q.5)

The **rate** of the code is $\frac{1}{5}$

- We'd like the rate to be as close to 1 as possible, i.e., fewer redundant bits to transmit
- We'd also like the probability of decoding error to be as small as possible

These two objectives are seemingly in tension ...

Probability of Error vs Rate



$(n, 1)$ Repetition Code

As we increase repetition code length n:

- A decoding error occurs only if at least $(n+1)/2$ bits are flipped $\Rightarrow$ Probability of decoding error goes to 0 as $n \to \infty$ ☺
- Rate $= \frac{1}{n}$, which also goes to 0 ☹

Can we have codes at strictly +ve code rate whose P(error) $\to 0$?

In 1948, it was proved that the answer is yes! (by Clause Shannon)

Block Codes

We'll look at Shannon's result shortly, but let's first try to improve on repetition codes using an idea known as *block coding*.

- In a block code, every block of K source bits is represented by a sequence of N code bits (called the codeword)
- To add redundancy, we need $N > K$
- In a linear block code, the extra $N - K$ code bits are *linear functions* of the K source bits

Example: The $(N = 7, K = 4)$ Hamming code

Each 4-bit source block $\mathbf{s} = (s_1, s_2, s_3, s_4)$, is encoded into 7-bit codeword $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7)$ as follows:

- $c_1 = s_1, \quad c_2 = s_2, \quad c_3 = s_3, \quad c_4 = s_4$

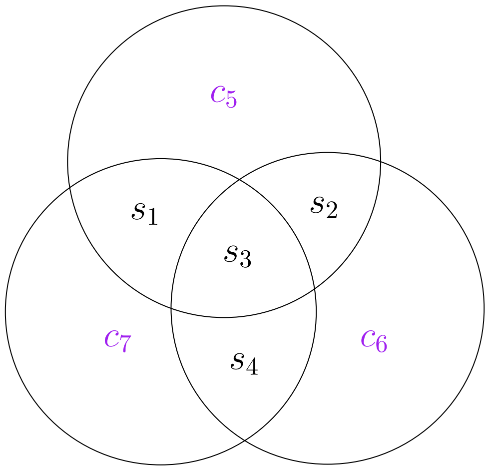
  $c_5 = s_1 \oplus s_2 \oplus s_3, \quad c_6 = s_2 \oplus s_3 \oplus s_4, \quad c_7 = s_1 \oplus s_3 \oplus s_4$

  where $\oplus$ denotes modulo-2 addition
- c_5, c_6, c_7 are called *parity check bits*, and provide the redundancy

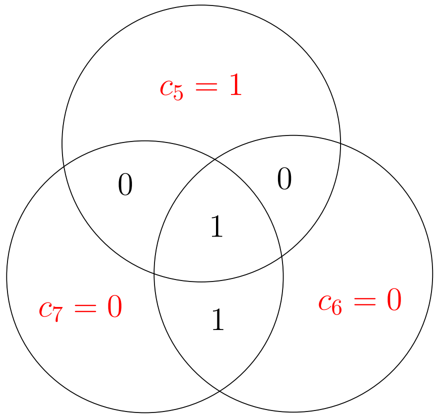
The $(7, 4)$ Hamming Code

E.g.:
For $\mathbf{s} = (0, 0, 1, 1)$, the codeword is $(0, 0, 1, 1, 1, 0, 0)$
For $\mathbf{s} = (0, 0, 0, 0)$, the codeword is $(0, 0, 0, 0, 0, 0, 0)$

The encoding operation can be represented pictorially as follows:



Example:



- For any Hamming codeword, the *parity* of each circle is *even*, i.e., there must be an even number of ones in each circle
- For encoding, first fill up $s_1, \ldots, s_4$, then c_5, c_6, c_7 are easy

Rate and Encoding

- The rate of any (K, N) block code is $\frac{K}{N}$
- The rate of a $(7, 4)$ Hamming code is $\frac{4}{7} = 0.571$
- Note that the $(N, 1)$ repetition code is a block code with $K = 1$ and rate $1/N$

Q: How do you encode a long sequence of source bits with a (K, N) block code?
A: Chop up the source sequence into blocks of K bits each; transmit the N-bit codeword for each block over the BSC.

E.g., For the $(7, 4)$ Hamming code, the source sequence

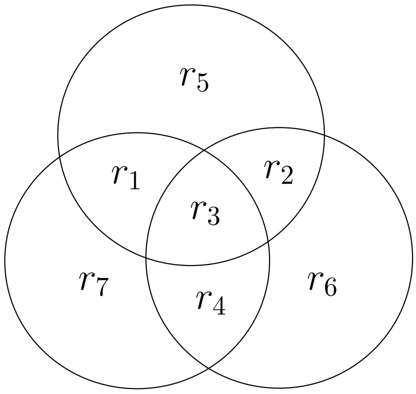
$$\mathbf{s} = \ldots \underbrace{1001}\ \underbrace{0010}\underbrace{1111}\underbrace{1010}\underbrace{0000} \ldots$$

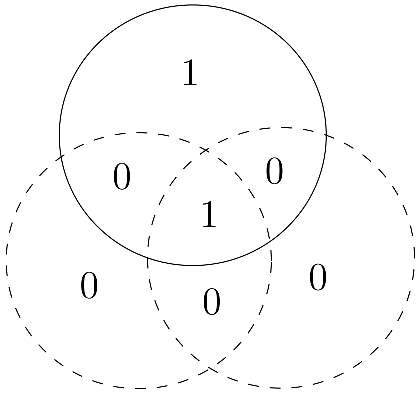
is divided into blocks of 4 bits; for each 4-bit block, the 7-bit Hamming codeword can be found using the parity circles

Error Correction for the Hamming Code

The $(7, 4)$ Hamming code can correct *any* single bit error (flip) in a codeword.

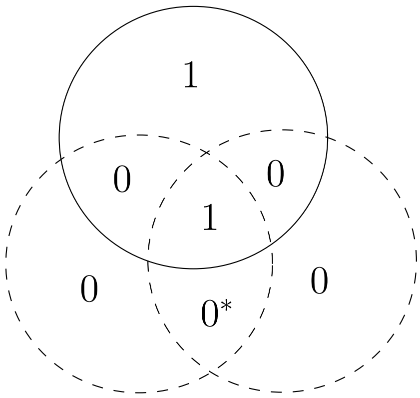
Example: The codeword $(0, 0, 1, 1, 1, 0, 0)$ (corresponding to source bits $(0, 0, 1, 1)$) is transmitted over the BSC. Suppose the channel flips the fourth bit so that the receiver gets $\mathbf{r} = (0, 0, 1, \mathbf{0}, 1, 0, 0)$.





Fill $\mathbf{r} = (r_1, \ldots, r_7)$ into the parity circles. We see that the dashed circles have odd parity.

Decoding Rule: If any circles have odd parity, flip *exactly one bit* to make all of them have even parity



Flipping the starred bit would make all the circles have even parity

We thus recover the transmitted codeword $(0, 0, 1, 1, 1, 0, 0)$

- When the channel flips a single bit, there is at least one circle that becomes "dashed"
- This shows that there is a bit error, which we can correct by flipping it back

Q: When does the $(7, 4)$ Hamming code make a decoding error?
A: When the channel flips two or more bits (Ex. Paper 9, Q.5b)

Thus Hamming codes have good rate $(= 4/7)$, but also rather high probability of decoding error

It's natural to wonder:

- How to design better block codes than repetition/Hamming?
- How many errors can the best (N, K) block code correct?

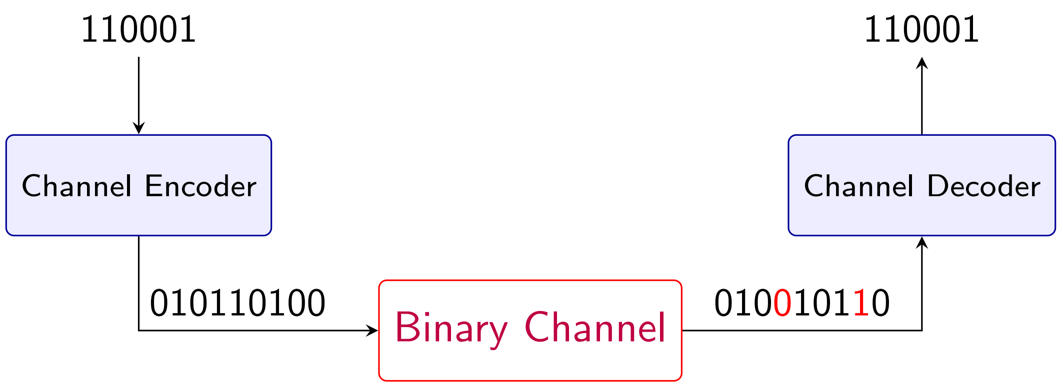
Shannon in 1948 ...

1. Showed that any communication channel has a **capacity**, which is the maximum rate at which the probability error can be made *arbitrarily small*.
2. Also gave a formula to compute the channel capacity

For example, Shannon's result implies that for the BSC(0.1):

- There exist (N, K) block codes with rate $\frac{K}{N} \approx 0.53$ such that you can *almost always* recover the correct codeword from the noisy output sequence of the BSC(0.1)
- But N has to be very large — the block length has to be several thousand bits long
- Practical codes with close-to-capacity performance have been discovered in the last couple of decades (discussed in 3F7, 4F5)

Channel Coding – The Key Points



- Once we fix a modulation scheme, we have a binary-input, binary-output channel
- *Channel coding* is the act of adding redundancy to the source bits to protect against bit errors introduced by the channel
- (N, K) block code: K source bits $\longrightarrow$ N code bits; $(N - K)$ bits provide redundancy
- The rate of a block code is K/N. We want the code rate to be high, but also correct a large number of errors
- We studied two simple block codes (repetition, Hamming) and their encoding and decoding