EBU5504: Networks and Protocols
(Week 4)

Dr Zhijin Qin

Lecturer

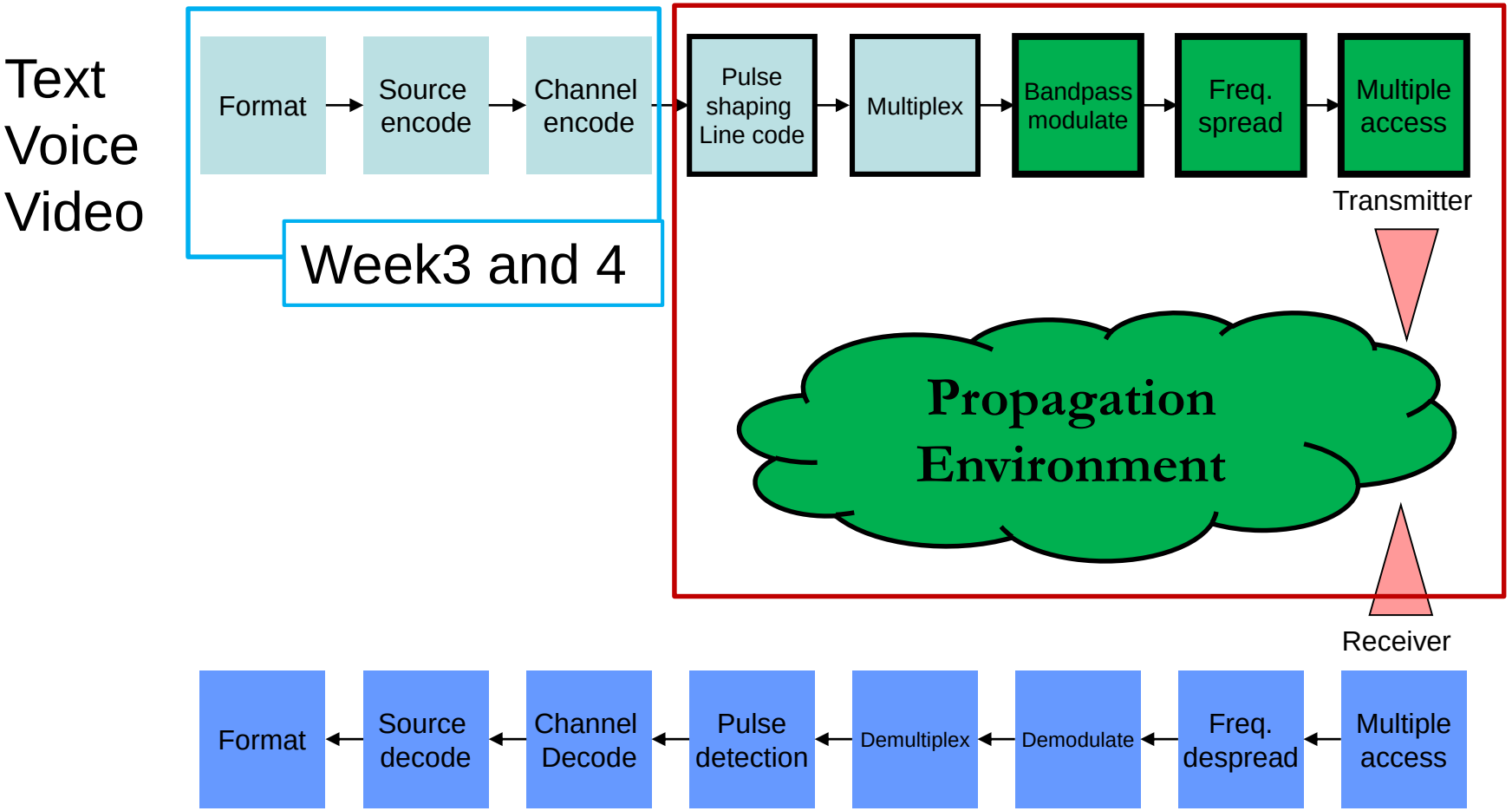
z.qin@qmul.ac.uk

Queen Mary
University of London

Review

- Modern communications
- Channel impairment
- Sampling
- Quantization
- Information Theory

Overview of Wireless Communication System



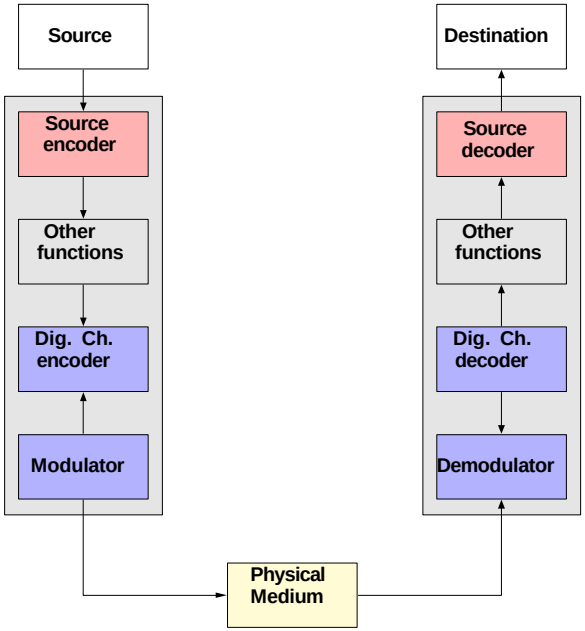
Outline

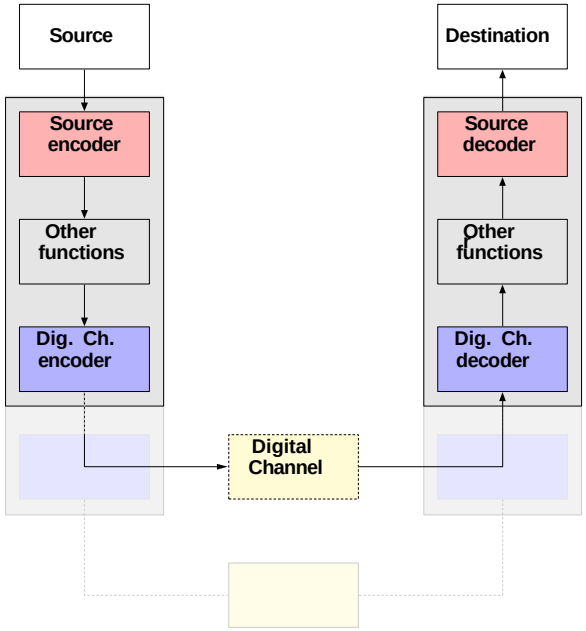
- Digital channel model
  - PAM
- Channel capacity
- Channel coding

DIGITAL CHANNELS

The digital channel

- The digital channel is a **mathematical abstraction**.
- It describes the relationship between the sequence of symbols that leaves the digital channel encoder and the one that reaches the digital channel decoder.





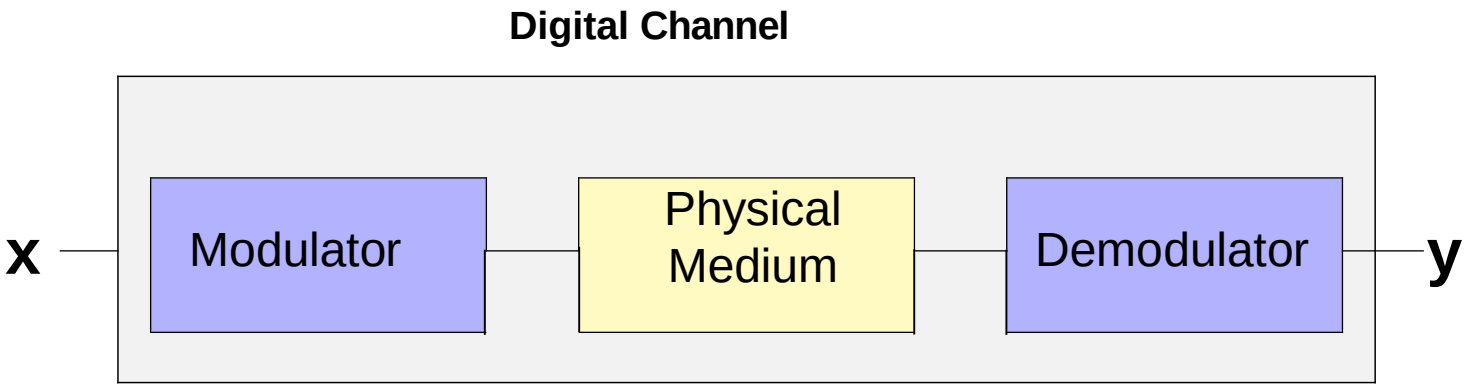
The digital channel

- Mathematically, a digital channel establishes a relationship between two sequences of symbols, the input sequence $x = [x_1, x_2, \ldots, x_N]$ and the output sequence $y = [y_1, y_2, \ldots, y_N]$.

- This relationship is described statistically by the probability of observing **y** at the output when **x** is at the input: $p(\mathbf{y}|\mathbf{x})$.

Mathematical definition of a digital channel

- Digital channel is an abstraction encompassing:
  1. the modulator at the transmitter
  2. the physical medium
  3. demodulator at the receiver.



- Hence, the modulator, the physical medium and the demodulator **determine** the relationship between the input symbol sequence x and the output symbol sequence y.

Mathematical definition of a digital channel

- There are many factors that can contribute to errors in the output sequence y ($y \neq x$)
  - Attenuation
  - Distortion
  - Noise
  - Bandwidth limitations
  - Multipath propagation
- Errors cannot be anticipated and hence, there is no deterministic relationship between the output sequence y and the input sequence x.
- A digital channel will be defined by a probabilistic relationship between the output and input sequence, **namely the probability of observing y when we use x as the input, p(ylx).**

The digital memoryless channel

- **Digital memoryless channels** have the property that the probability of observing a symbol y_i in the output sequence y, **only depends on the input symbol at the same time instant, x_i.**
- Conditional probability defining the channel can be expressed as:

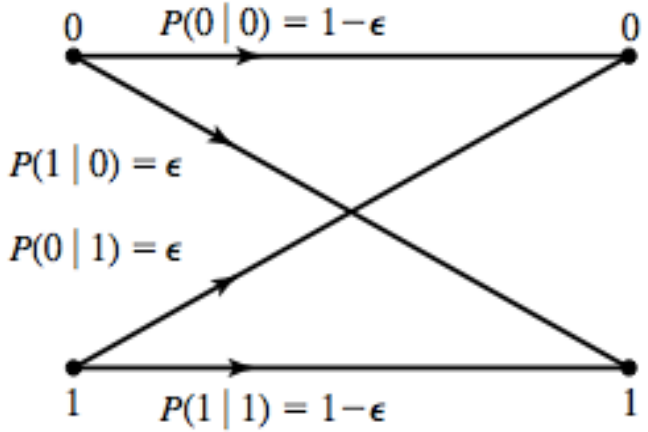
$$p(y|x) = \prod_{i=1}^{N} p(yi|xi)$$

where $p(y_i|x_i)$ is the probability of observing symbol y_i at the output when symbol x_i is present at the input.

The binary symmetric channel

- The **binary symmetric channel** is a special case of memoryless channel for which the input and output alphabet are binary $X = 0, 1$ and $Y = 0, 1$.
- The symbol conditional probabilities are defined as:

$$p(1|1) = p(0|0) = 1 - \varepsilon$$
$$p(1|0) = p(0|1) = \varepsilon$$



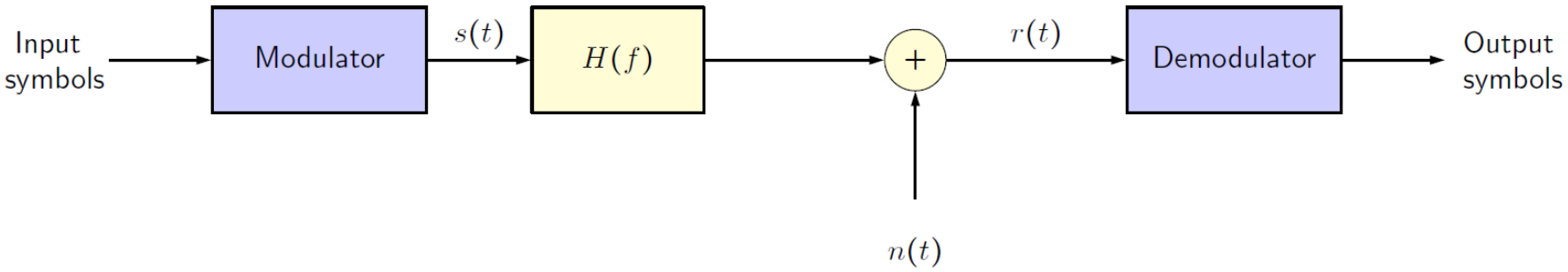
- The value ε is known as the **crossover probability**.
- $p(1|0) + p(0|0) = 1$ and $p(1|1) + p(0|1) = 1$.

Review: Modulation

- What is the purpose of modulation?
- What is the basic process of modulation?
- What are the different types of modulation?

Model of communication system - Modulator

- The **modulator** takes a sequence of symbols and produces an **analog signal s(t)** that is transmitted. This signal is known as the **modulated signal**.

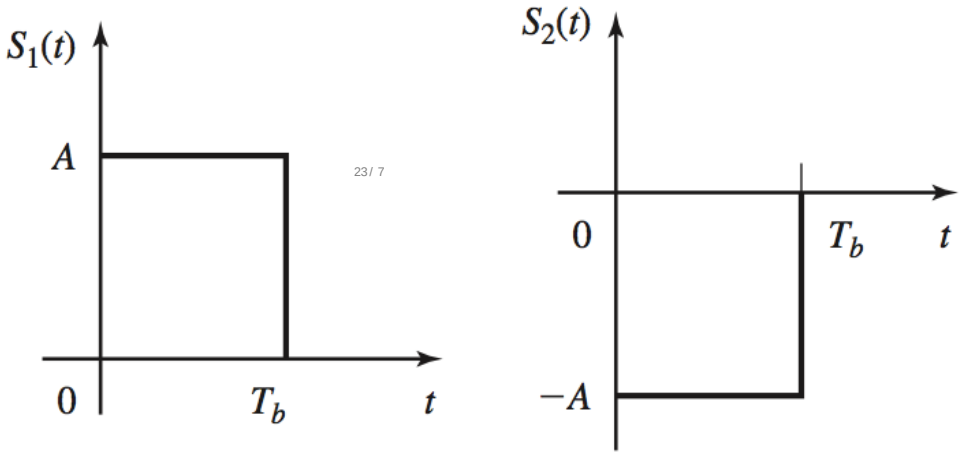


- The channel is characterised by a transfer function H(f) (or an impulse response h(t) and an additive noise source n(t).
- Its output is **r(t) = s(t) * h(t) + n(t).**

- The demodulator receives the output of the channel r(t) and recovers the sequence of symbols.

Example: Binary PAM and AWGN channel - Modulator

- Binary PAM is the simplest digital modulation method.
- Symbol 1 is converted into a pulse waveform $S_1(t)$ of amplitude A, whereas symbol 0 is converted into a pulse waveform $S_2(t)$ of amplitude $-A$.



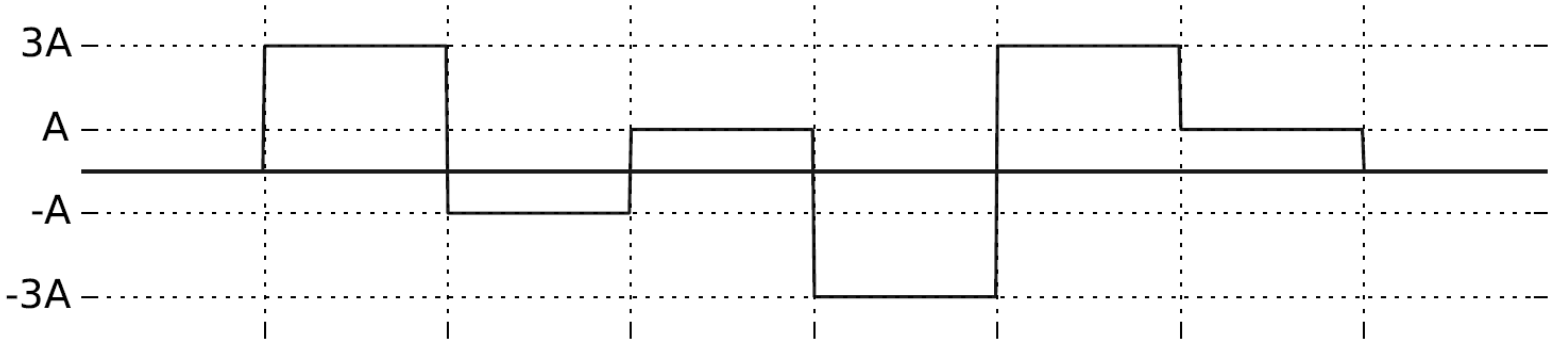
- A sequence of bits is transmitted as a sequence of the corresponding waveforms at a bit rate $R_b = 1/T_b$, where T_b is the bit interval.
- The energy of a symbol is $E_b = A^2 T_b$.

PAM signals

- **Pulse Amplitude Modulation (PAM):** When the waveforms representing each symbol only difference in the amplitudes**.**
- A PAM signal can be mathematically expressed as

$$s(t) = \sum_{n} a_n p(t - nT_s)$$

where a_n is the **amplitude** of the n-th transmitted symbol, $p(t)$ is the PAM waveform (a rectangular pulse, for instance) and T_s is the symbol time.



Energy and average power in PAM signals

- The waveforms of a PAM signal are energy signals. In the case of a rectangular pulse p(t) of duration Ts and amplitude *A*, the energy is

$$E_P = A^2 T_S$$

- The energy of the waveform $a_m p(t)$ corresponding to symbol *m* is

$$E_m = a_m^2 A^2 T_S = a_m^2 E_P$$

- The average power of a PAM signal can be obtained as

$$P_{PAM} = \frac{1}{T_S} \sum_m P\left(symbol\ m\right) E_m$$

where P(symbol *m*) is the probability of transmitting the symbol *m*.

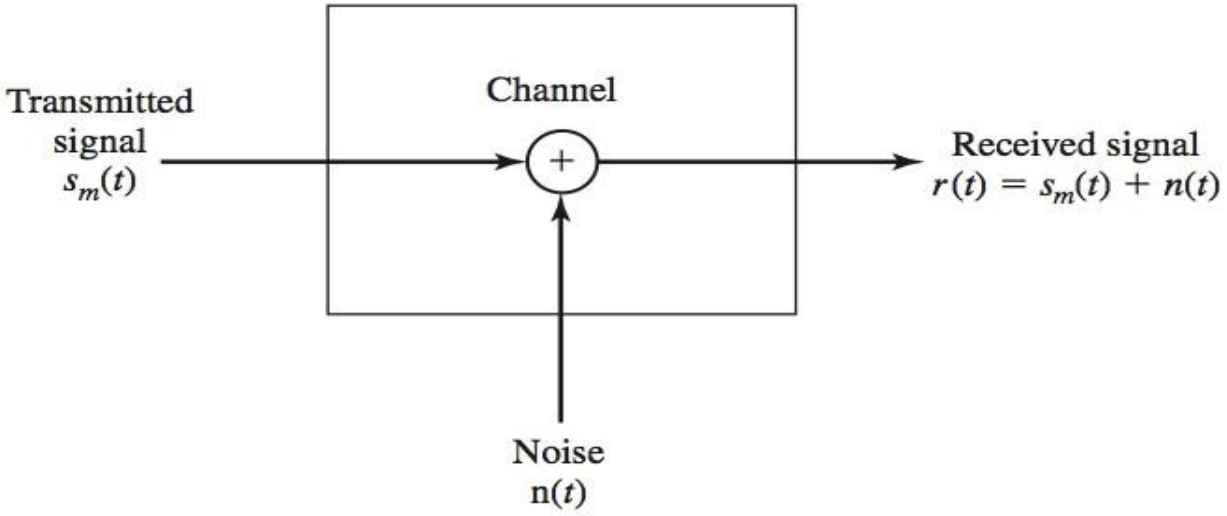
Transmission rate

The following concepts are used in digital communications:

- Symbol time T_S is the **time interval** between the transmission of two consecutive symbols.
- Symbol transmission rate (or baud rate) R_S is defined as the **inverse of the symbol time**, $R_s = 1/T_S$
- In the **binary case**, T_S and R_S are equivalent respectively to the bit time T_B and the bit transmission rate R_B.
- In other cases, we need to consider the number of bits that we need to represent each symbol, N_B. Then, $\boldsymbol{R_B = N_B \times R_S}$.
- The bandwidth W is related to the symbol transmission rate R_S: the higher the transmission rate R_S, the wider bandwidth W.

Example: Binary PAM and AWGN channel - Channel

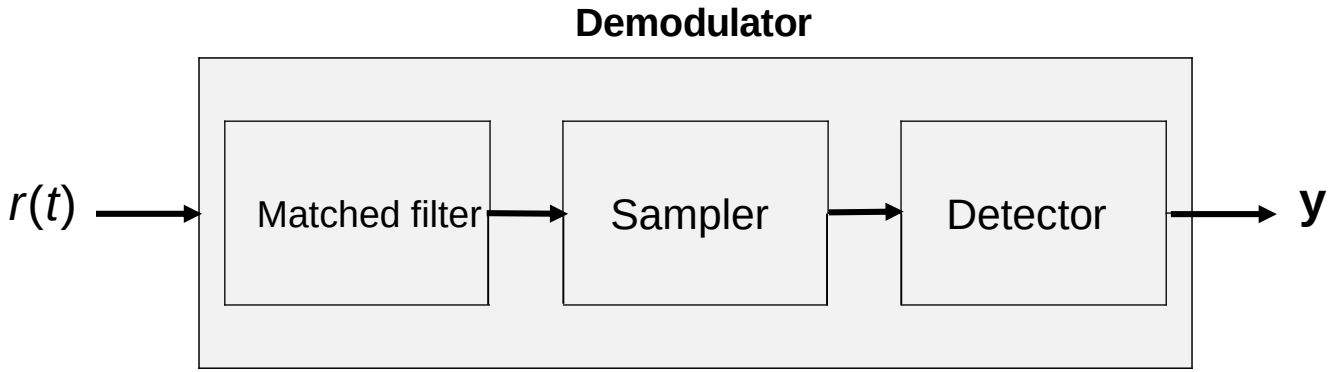
In the additive white Gaussian noise (AWGN) channel, the signal at the output $r(t)$ can be expressed as the message signal at input $s_m(t)$ plus white Gaussian noise, $n(t)$.



The white Gaussian noise $n(t)$ has a flat power spectrum characterized by a density $N_o/2$.

Example: Binary PAM and AWGN channel- Demodulator

- The binary PAM demodulator consists of three steps, namely a matched filter, a sampler and a detector.



- The **matched filter** compares the input waveforms with the expected waveform.
- The **sampler** provides a numerical value of how similar the received waveform and the expected one are.
- Based on the value, the **detector** will decide wether symbol 1 or symbol 0 was received.

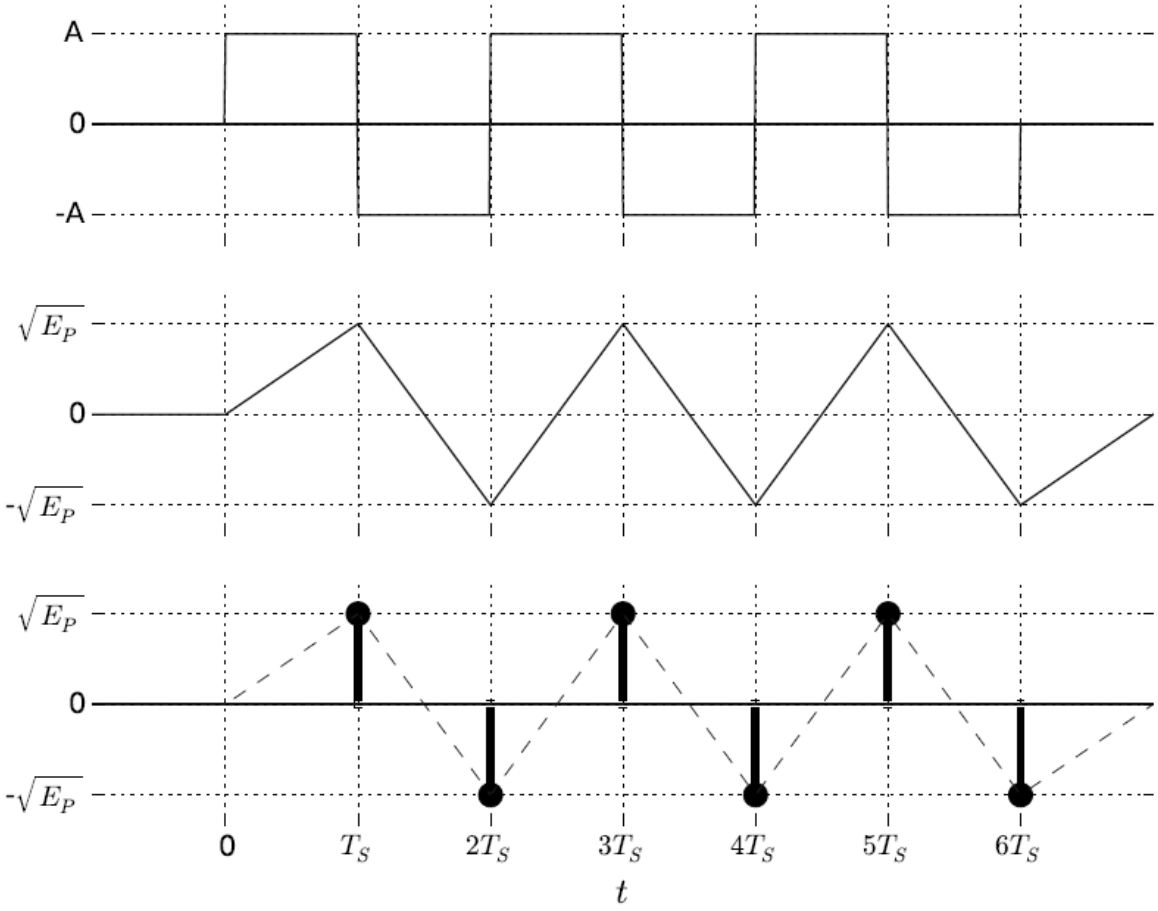
PAM Demodulator

- In the case of a PAM demodulator for waveforms $S_m(t) = a_m p(t)$ can be designed with the following elements:
  - The **matched filter** is a system with impulse response

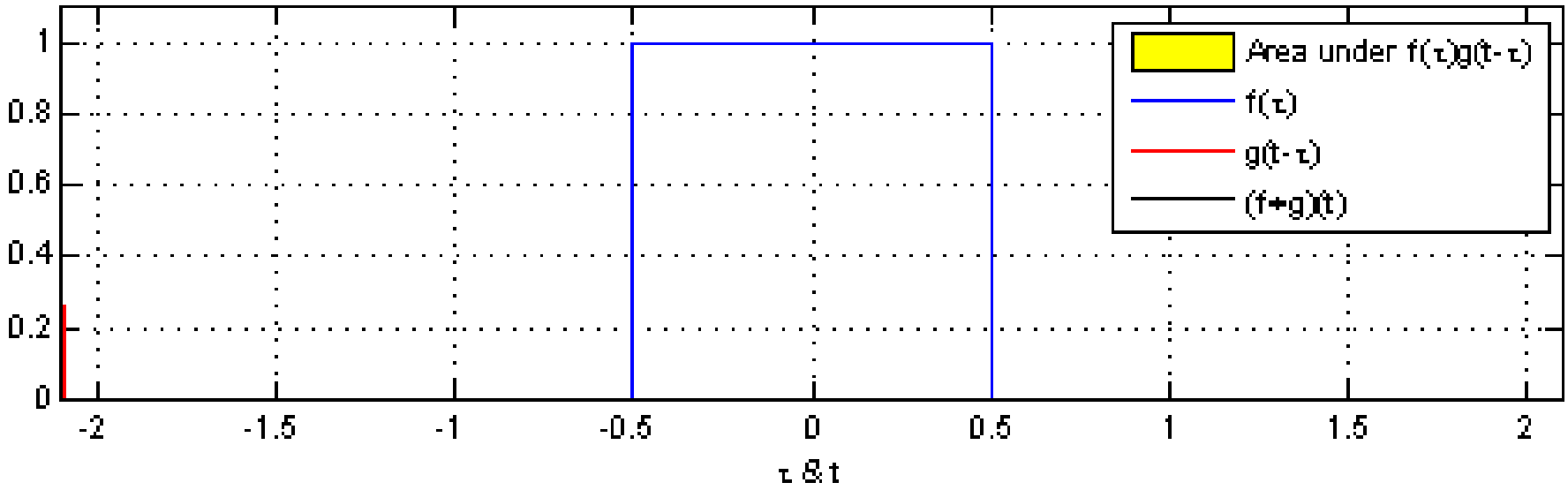
$$h(t) = p\,(T_s - t)/\sqrt{E_P}$$

  - The **sampler** extracts amplitude samples every T_s units of time.
  - The **detector** uses the constellation of the PAM to detect each symbol.

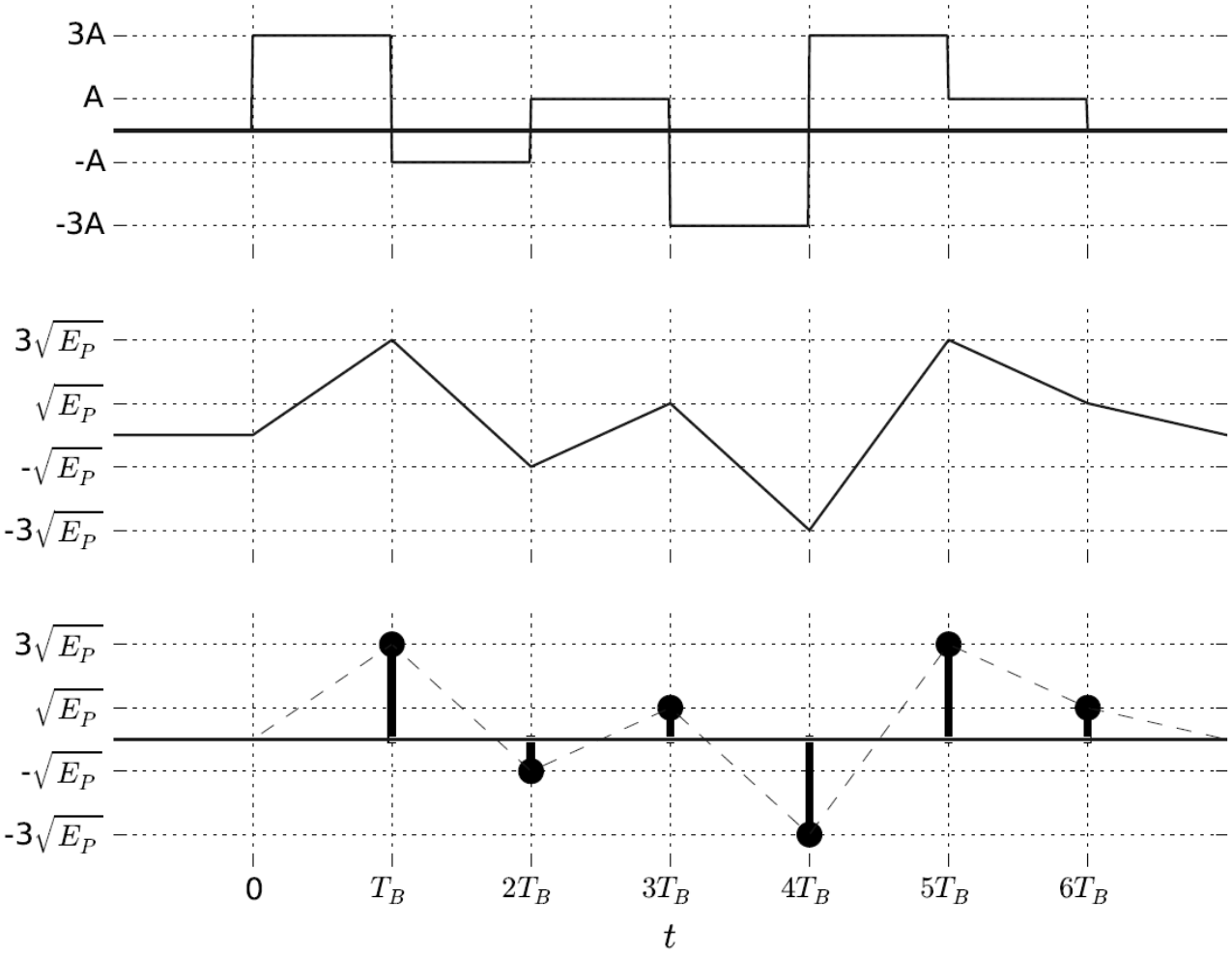
2 PAM demodulator



Convolution



4 PAM demodulator

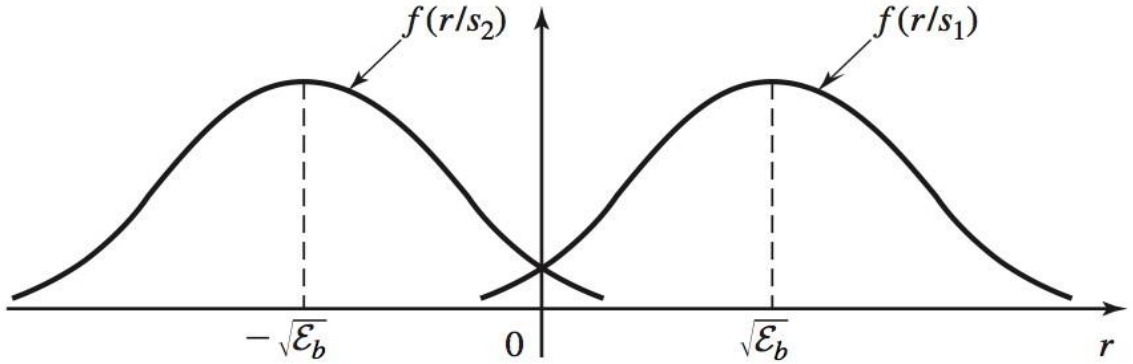


Example: Binary PAM and AWGN channel - Demodulator

It can be proved that the sample values have the following probability density functions that depend on the transmitted waveform, $S_1(t)$ or $S_2(t)$, and the noise power density $N_0/2$:

$$f(r \mid s_1) = \frac{1}{\sqrt{\pi N_0}} e^{-(r-\sqrt{\mathcal{E}_b})^2/N_0}$$

$$f(r \mid s_2) = \frac{1}{\sqrt{\pi N_0}} e^{-(r+\sqrt{\mathcal{E}_b})^2/N_0}$$



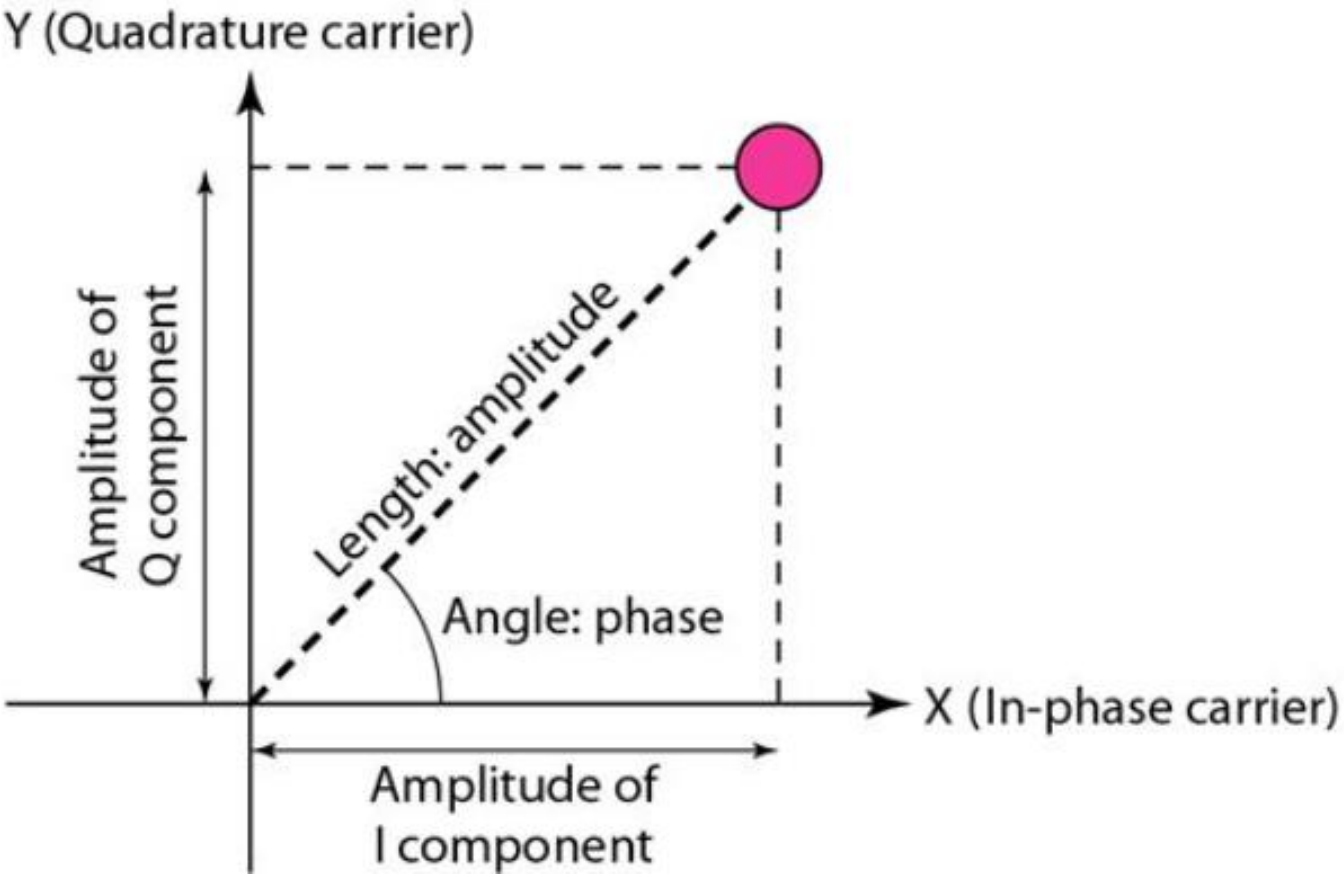
Example: Binary PAM and AWGN channel - Demodulator

Based on the sample values PDF, the detector will decide symbol 1 was transmitted whenever the sample value is positive; otherwise it will decide symbol 0 was transmitted. The probability of error when $S_1(t)$ is transmitted will then be:

$$
\begin{aligned}
P(e \mid s_1) &= \int_{\infty}^{0} p(r \mid s_1)\, dr \\
&= \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^{0} e^{-(r-\sqrt{\mathcal{E}_b})^2/N_0}\, dr \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\sqrt{2\mathcal{E}_b/N_0}} e^{-x^2/2}\, dx \\
&= \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2\mathcal{E}_b/N_0}}^{\infty} e^{-x^2/2}\, dx
\end{aligned}
$$

The same probability of error will be obtained when $S_2(t)$ is transmitted.

Constellation diagram

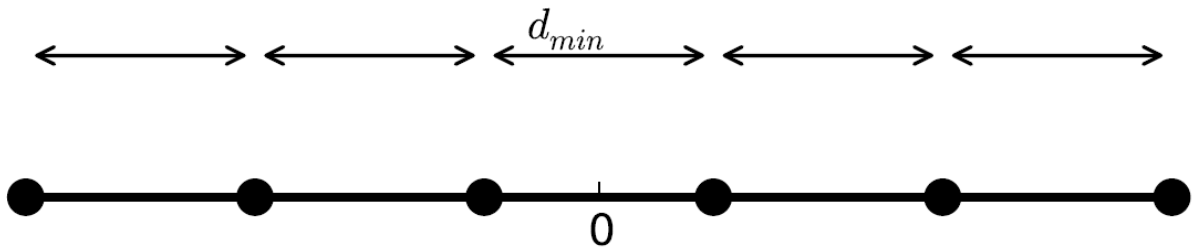


Constellation of PAM systems

- Each transmitted waveform $S_m(t) = a_m p(t)$ will produce, at the output of the sampler, the value

$$s_m = a_m \sqrt{E_P}$$

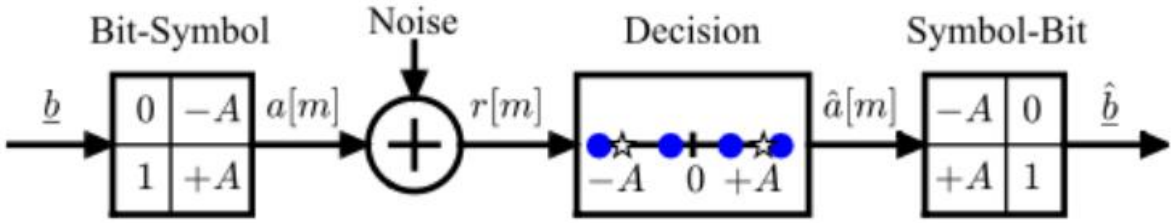
- The set of the values produced by each transmitted waveform denes the constellation of the PAM modulation and a distance between symbols *m* and *l*, $d_{m.l} = (a_m - a_l)\sqrt{E_P}$



- Based on the constellation, the detector will dene the so-called detection regions.

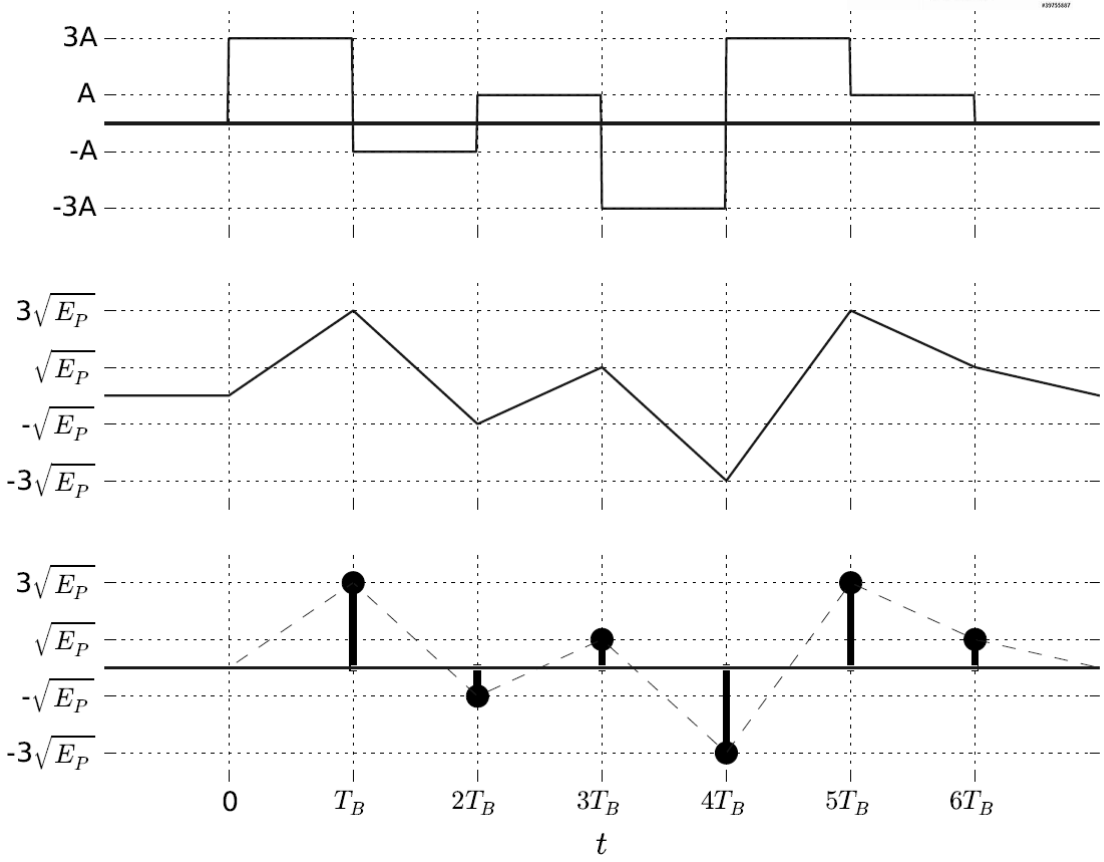
A simple communication system

m	0	1	2	3
b	1	0	1	0
$a[m]$	+2	-2	+2	-2
$r[m]$	+2.2	-0.4	+1.1	-2.4
$\hat{a}[m]$	+2	-2	+2	-2
$\hat{b}$	1	0	1	0



Example

- For 4-PAM shown right answer the following questions
  1. Draw the signal constellation
  2. Calculate the distance between symbols.
  3. What are the transmitt sequences?



CHANNEL CAPACITY

Mathematical definition of a digital channel

- A digital channel is a system that establishes a relationship between two **sequences of symbols**, namely the **input sequence x** = $[x_1, x_2, \ldots, x_N]$ and the **output sequence y** = $[y_1, y_2, \ldots, y_N]$.

- The symbols x_i in the input sequence belong to an alphabet X and the symbols at the output y_i to an alphabet Y.

Information-theory analysis of the digital channel

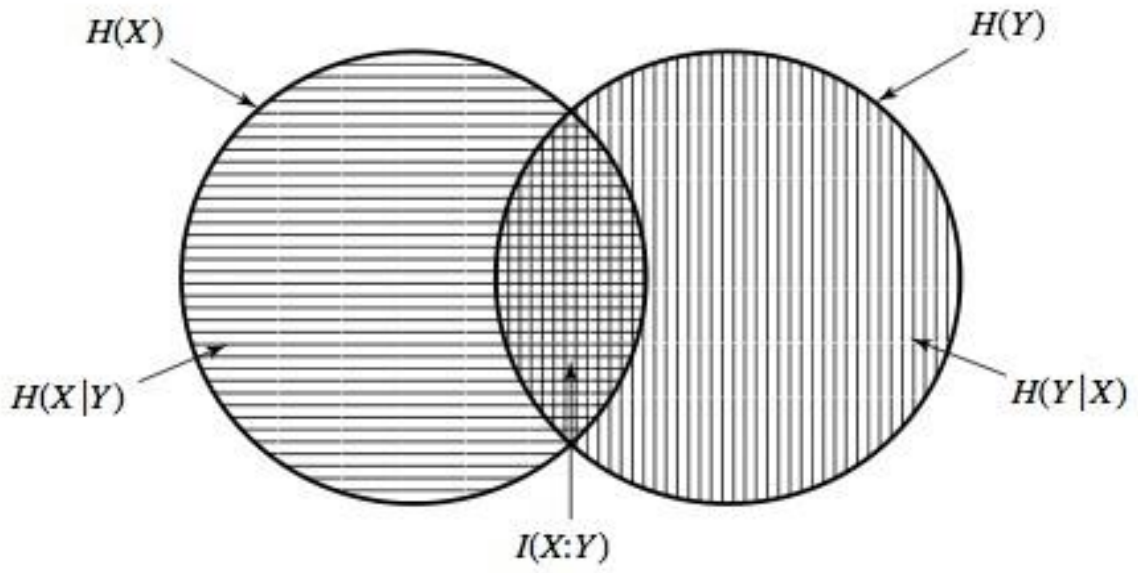
If we treat X and Y as information sources, we can define the following quantities:

- The entropies $H(X)$ and $H(Y)$: the **information content** of each source.
- The entropy $H(X, Y)$: the information content of both sources.
- The conditional entropies $H(X|Y)$ and $H(Y|X)$: the new information provided by one source if the other source is known.
- **The mutual information** $I(X; Y)$: the information shared by both sources.

These quantities can be used to describe the relationship between the input and the output of a digital channel.

Information-theory analysis of the digital channel

Information theory quantities $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$ and $I(X;Y)$ can be represented as follows:



Question: Which quantity are we interested in when transmitting information?

Information-theory analysis of the digital channel

The entropy H(X) is defined as

$$H(X) = -\sum_{x} P(x) \log p(x)$$

and the conditional entropy $H(X|Y)$ is defined as

$$H(X|Y) = -\sum_{x,y} P(x,y) \log p(x|y)$$

The mutual information $I(X;Y)$ can be defined as

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

Question: It can be proved that $0 \leq I(X;Y) \leq \min(H(X), H(Y))$, can you see why?

Additional info.

$$
\begin{aligned}
\mathrm{H}(Y|X) &\equiv \sum_{x \in \mathcal{X}} p(x)\, \mathrm{H}(Y|X=x) \\
&= -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x)\ \log\, p(y|x) \\
&= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y)\ \log\, p(y|x) \\
&= -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log\, p(y|x) \\
&= -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)}. \\
&= \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log \frac{p(x)}{p(x,y)}.
\end{aligned}
$$

Additional info.

$$
\begin{aligned}
\mathrm{H}(Y|X) &= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\left(\frac{p(x)}{p(x,y)}\right) \\
&= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log(p(x,y)) + \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log(p(x)) \\
&= \mathrm{H}(X,Y) + \sum_{x\in\mathcal{X}} p(x)\log(p(x)) \\
&= \mathrm{H}(X,Y) - \mathrm{H}(X).
\end{aligned}
$$

Interlude: The medical diagnosis machine

You are a hospital manager and want to buy a diagnosis machine that tells whether a patient suffers from a certain illness or not. Three different companies offer you their machines, M_A, M_B and M_C.

Before buying any of the machines, you try them on patients whose diagnosis you know in advance and get the following percentage of correctly diagnosed patients:

- M_A: 80%.
- M_B: 50%.
- M_C: 2%.

Which machine would you buy? Why?

Channel Model

- **Noiseless case:** The channel in this case transmits symbols without causing any errors. One would need to exploit the redundancy in the source to economize the length of the transmission. This is done through data compression, also called source coding. The information is decompressed at destination.

- **Noisy case:** The channel in this case introduces noise that causes errors in the received symbols at the destination. To reduce the errors incurred due to noise, one should add systematic redundancy to the information to be sent. This is done through channel coding.

It is known that reliable communication is possible in the above model if the entropy of the source, i.e., the amount of non-redundant information it generates per unit of time, is less than the "capacity" of the channel, i.e., the maximum number of information bits that can be communicated reliable per channel use.

Channel capacity

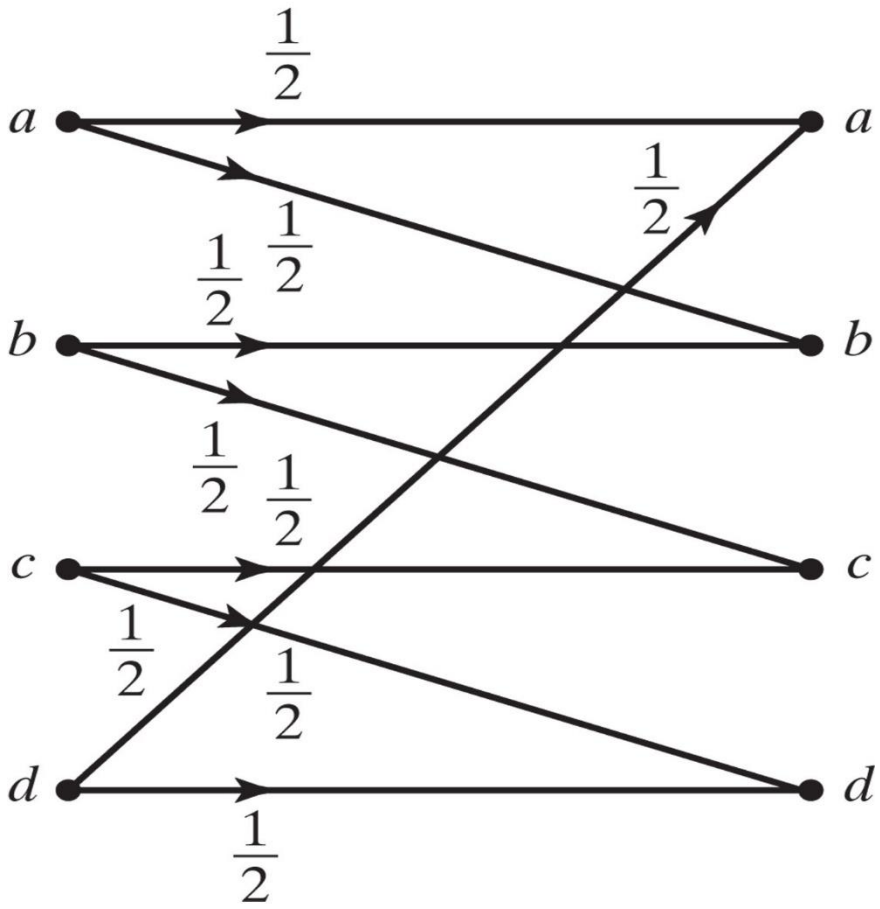
The main objective when transmitting information over a channel is **reliability**, which is measured by **the probability of correct reception** at the receiver.

Information theory tells us that this probability can be increased as much as we want as long as the transmission rate is less than the **channel capacity**.

Noisy channel coding theorem (Shannon 1948) says that the basic limitation that noise causes in a communication channel is not on the reliability of communication, but on the speed of communication.

The channel capacity imposes a **theoretical limit** on the transmission speed. Hence, **the probability of error affects the speed of the communication**.

An example of a discrete channel



- Four Inputs/Outputs
  - What if *b* is received?
  - What if *d* is received?
- Two Inputs/Outputs (a/c)
  - What if *b* is received?
  - What if *d* is received?

→ Using only those inputs whose corresponding possible outputs are disjoint, and thus do not cause ambiguity

Channel capacity of memoryless binary channels

Let us consider a memoryless binary digital channel with input sequence $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ and output sequence $\mathbf{y} = [y_1, y_2, \ldots, y_N]$.

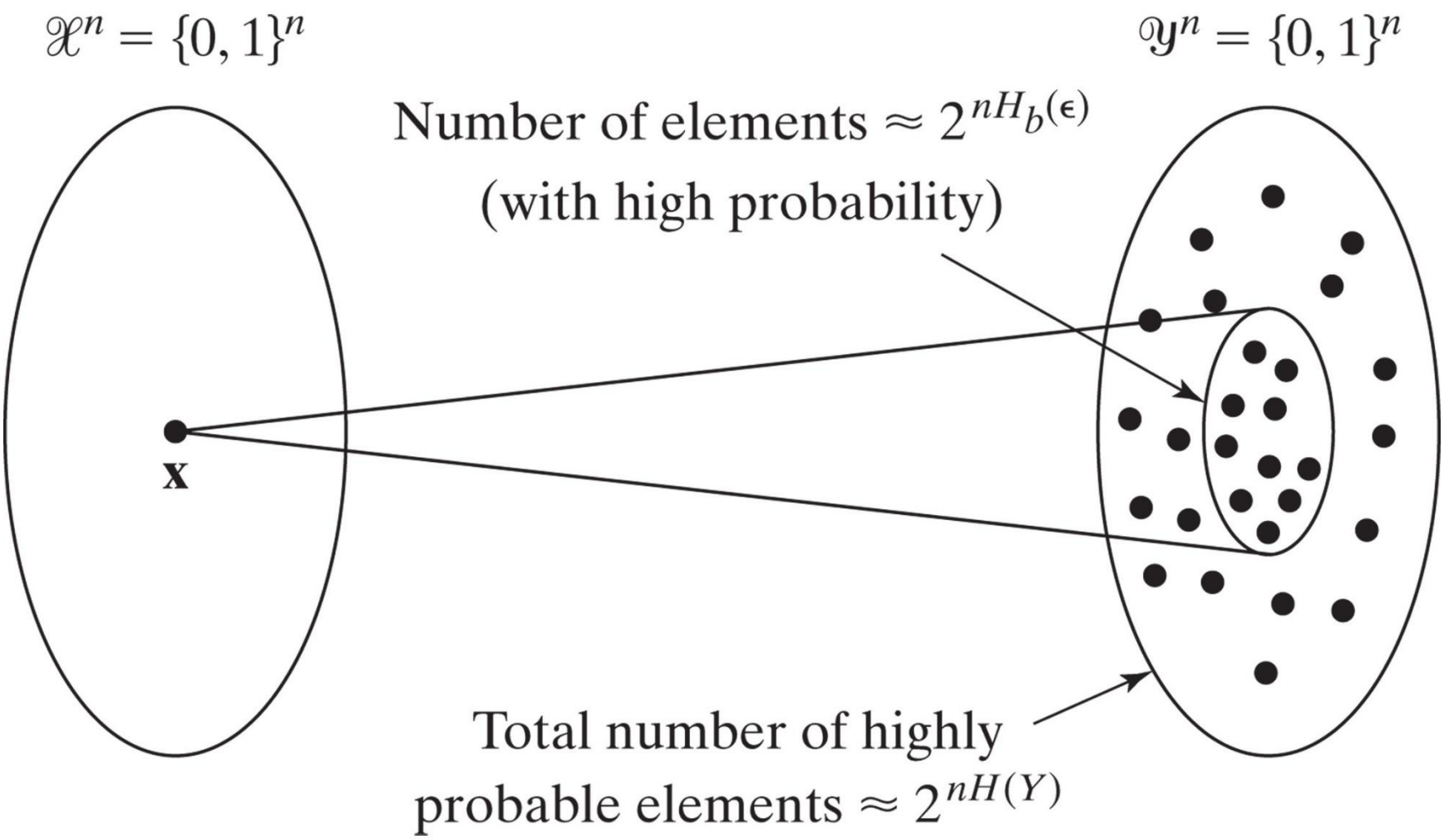
Let ε be the **crossover (error) probability**.

For N *is* large enough, we will expect $N \times \varepsilon$ errors in the output sequence $\mathbf{y}$.
Using Stirling's approximation for factorials, the number of possible output sequences $\mathbf{y}$ that disagree with $\mathbf{x}$ in $N \times \varepsilon$ positions is

$$\binom{N}{N\varepsilon} \approx 2^{NH(\varepsilon)}$$

where $H(\varepsilon) = -\varepsilon \log \varepsilon - (1-\varepsilon) \log(1-\varepsilon)$. This means that for every input sequence $\mathbf{x}$ of length N there will be approximately $2^{N\,H(\varepsilon)}$ different output sequences $\mathbf{y}$ of length N.

Channel capacity of Binary Symmetric Channels



Channel capacity of memoryless binary channels

When we analyzed information sources, we concluded that for an information source Y with entropy $H(Y)$ there are $2^{N\,H(Y)}$ highly probable sequences **y** of length N.

Hence, the **quantity**

$$M = \frac{2^{NH(Y)}}{2^{NH(\epsilon)}} = 2^{N(H(Y)-H(\epsilon))}$$

corresponds to the maximum number of input sequences **x** that can produce different output sequences **y**.
Then, in theory, **if we choose wisely** M **different input sequences we can always identify them without error by looking at the output sequence**.

Channel capacity of memoryless binary channels

If we restrict ourselves to M different binary input sequences of length N, the transmission rate R will be:

$$\boldsymbol{R = \frac{logM}{N} = H(Y) - H(\varepsilon)}\ \textbf{bits/transmission (bits/symbol)}$$

How can we increase the transmission rate? Either by reducing $H(\varepsilon)$ or by increasing $H(Y)$:

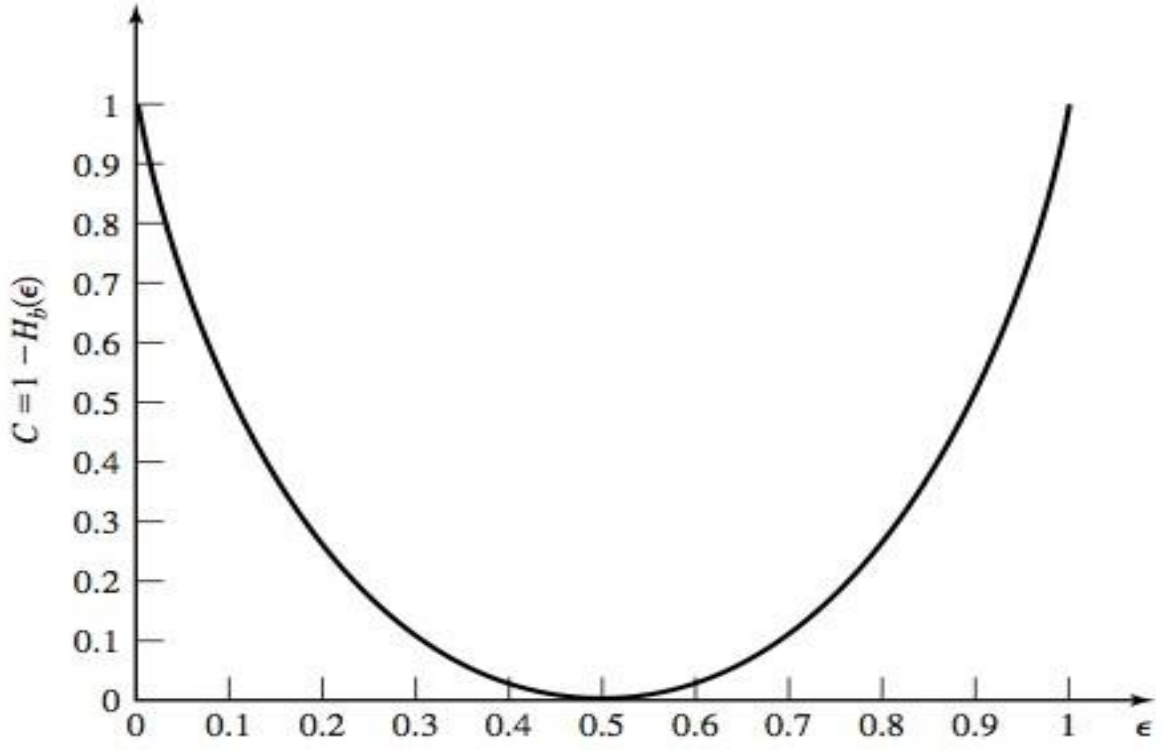
- The quantity $H(\varepsilon)$ cannot be controlled, since it is a property of the channel.
- **The entropy $\boldsymbol{H(Y)}$ can however be maximized by wisely choosing $\boldsymbol{p(x)}$.**

The resulting maximum transmission rate C will be:

$$C = 1 - H(\varepsilon) \text{ bits/transmission}$$

and it is known as the **channel capacity**.

Channel capacity of memoryless binary channels



Question: Why is $C = 0$ when $\varepsilon = 0.5$? Why is $C = 1$ for both $\varepsilon = 0$ and $\varepsilon = 1$?

The noisy channel coding theorem

The capacity of a digital memoryless channel is given by

$$C = \max_{p(x)} I(X;Y)$$

$$I(X;Y) = \sum_{x,y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)}\right)$$

If the transmission rate R is less than the channel capacity C, there will exist a code that will result in an error probability as small as desired.
If $R > C$, the error probability will be bounded away from 0 (error will happen definitely).

CAPACITY OF AWGN CHANNEL

Shannon's formula

The capacity of a channel defines a **limit for reliable communications**. Only when the information rate is below the capacity of the channel, error-free transmission **can** be achieved (if we design our system properly!).

According to Shannon's formula, the capacity of an additive white Gaussian noise channel is

$$C_{bit/s} = Wlog(1 + SNR) = Wlog\left(1 + \frac{P}{N_0 W}\right) bit/\text{sec}$$

where W is the channel bandwidth, the noise spectral density is $N_0/2$ (double-side bandwidth)and P is the signal power.

Notice that the noise power is $P_N = N_0 W$, hence $SNR = \frac{P}{P_N} = \frac{P}{N_0 W}$

Question:

- Find the capacity of a telephone channel with bandwidth W = 3 KHz and SNR of 39 dB.

Transmission rate and spectral bandwidth

In digital systems, the speed of communication is measured by the **bit transmission rate** R_B (number of bits per second) or, in general, by the **symbol transmission rate** R_S (number of symbols per second).

Digital information is transmitted through a physical medium by means of analog waveforms that occupy a bandwidth W. It can be proved that **the maximum symbol transmission rate is** $2W$ **symbol/sec**. Hence, Shannon's formula can also be expressed as

$$C_{bit/sym} = \frac{C_{bit/s}}{2W} = \frac{1}{2}\log(1 + SNR) \ \ bit/symbol$$

Consequently, we can measure the channel's capacity both in bits/sec and bits/symbol.

Spectral efficiency

The spectral efficiency η is used to measure **how efficiently the available bandwidth is used**. It is defined as

$$\eta = \frac{R_B}{W} \; bps/Hz$$

where bps=bits/s.

(For instance, in GSM R_B = 104Kbps and W = 200KHz, hence η_{GSM} = 0.52bps/Hz. In LTE R_B = 81Mbps and W = 20MHz, hence η_{LTE} = 4.08bps/Hz.)

The maximum spectral efficiency η_{max} is then

$$\eta_{max} = \frac{C_{bit/s}}{W} = \log\left(1 + \frac{P}{N_0 W}\right) bps/Hz$$

Signal to noise ratio

In digital communication systems, the bit energy E_B is the average energy that we use to transmit one bit. Since $T_B = 1/R_B$ is the duration of one bit, the signal power P can be obtained as

$$P = E_B R_B$$

By using the bit energy, the SNR can be expressed as

$$SNR = \frac{P}{P_N} = \frac{E_B R_B}{N_0 W} = \eta \frac{E_B}{N_0}$$

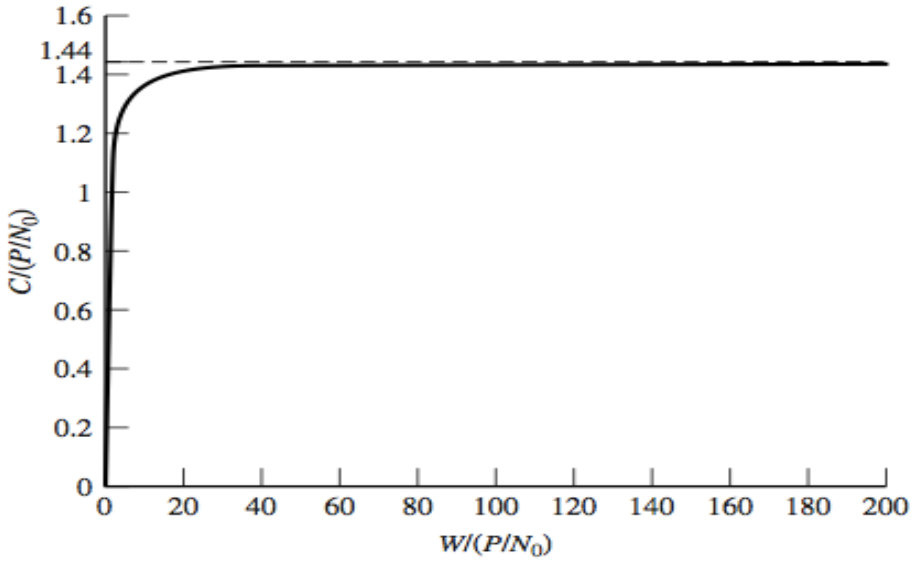
and Shannon's formula as

$$C_{bit/s} = W log(1 + \eta \frac{E_B}{N_0}) \text{ bit/sec.}$$

Bandwidth effects

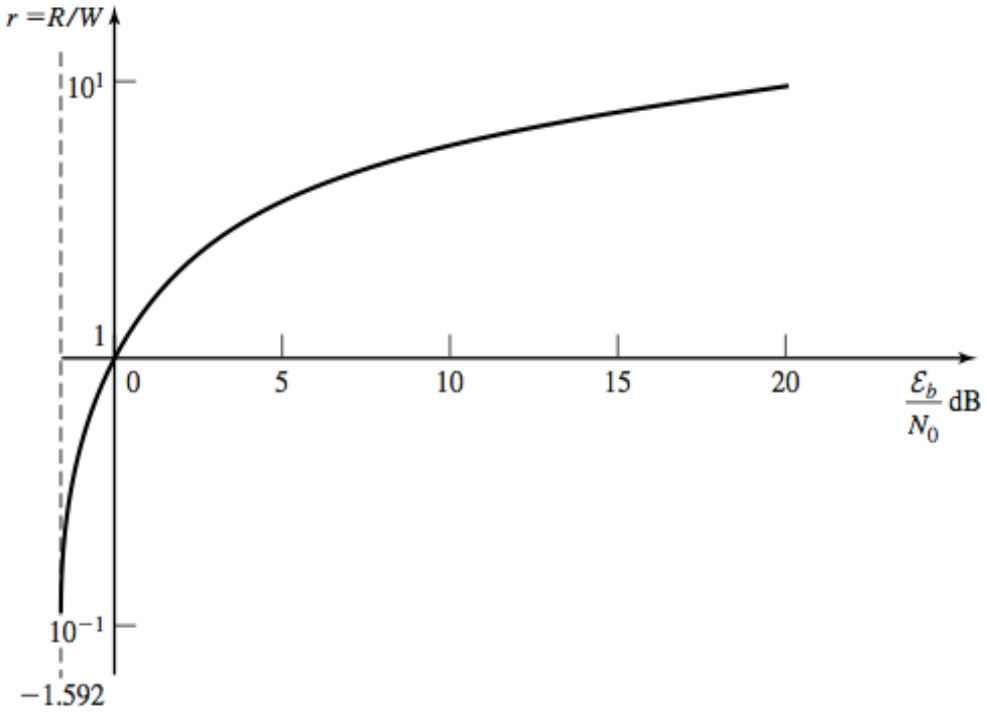
With higher bandwidths the transmission rate can be increased. However, higher bandwidths also imply higher noise power. The following limit can be derived:

$$\lim_{W \to \infty} C = 1.44 \frac{P}{N_0}$$



SNR effects

By increasing the signal to noise ratio (SNR) the maximum information rate for reliable communications increases.



CHANNEL CODING

Channel coding and redundancy

Coding is a process that produces a sequence of symbols from another sequence of symbols.

In **source coding**, given a sequence of symbols we produce a new, shorter sequence that contains the same information. Hence, **eliminate redundancy**.

By contrast, in **channel coding** our aim is to protect information against errors and for that we **introduce redundancy**, producing longer sequences of symbols.

Are we undoing in channel coding what we did in source coding? **Not really**, in the former we eliminate unnecessary redundancy and in the latter we introduce suitable redundancy!

A simple example of channel coding

Let us define the following code:

- $0 \rightarrow 00$.
- $1 \rightarrow 11$.

By using this code, we are sending every bit of our message twice. Hence, we would expect to see sequences of 00 and 11 but not 10 nor 01 since **they are not codewords**. Unless, of course, there is one error.

In the sequence $00 - 11 - 00 - 11 - 11 - \mathbf{10} - 00 - 00$ we have been able to detect one error, but we do not know whether the corrupted two-bits sequence was 11 or 00.

Another simple example of channel coding

Let us define the following code:

- $0 \rightarrow 000$.
- $1 \rightarrow 111$.

Now we are sending every bit of our message three times. If we detect a 3-bit sequence that is not 000 nor 111 we will know there has been an error.

In the sequence $000 - 111 - 000 - 111 - 111 - \mathbf{101} - 000 - 000$ we have been able to detect one error. In this case, we will correct 101 to 111 and not to 000, because it is more probable to observe one error than two errors.

Coding rate

From now on, we will assume that we are dealing with binary sequences. Let k be the length of the original binary sequence and n the length of the sequence after coding. Hence, we are introducing $m = n - k$ redundancy bits.

We define the code rate R_C as

$$R_C = \frac{k}{n}$$

As we can see, only 2^k binary sequences out of 2^n binary sequences are valid code words! If we receive a sequence that is not a code word, it means that errors have occurred during transmission.

Question: What was the coding rate in the previous two examples?

Effects of coding on the bandwidth and the bit rate

If our code rate is $R_C = k/n$, for every k bits of our message, we will be transmitting n bits. Hence:

- If our bandwidth is fixed, then the information rate will decrease by R_C. In other words, we transmit fewer bits of our message per second.
- If our information rate is fixed, then the transmission rate will increase by $1/R_C$ and so will the necessary bandwidth. In other words, we need more bandwidth to accommodate more bits in the same time interval.

Error detection and correction

We have seen that the purpose of channel coding is to protect our information against errors. Assume that we are using a code rate R_C.

- **Forward error correction** (FEC) protects our message against up to N_C errors. It is convenient in those cases where we do not have a feedback link.
- If we detect errors, what should we do? We can ask for the message to be retransmitted. **Automatic repeat request** (ARQ) consists of asking the sender to retransmit the message.
- For the same code rate R_C, $N_D > N_C$

Types of channel codes

Here exist two main families of channel coding techniques:

- **Block codes**. In a block code, an information sequence is broken into blocks of length k and each block is mapped into channel inputs of length n. Each block is independent from any other block.
- **Convolutional codes**. In a convolutional code, k bits of the information sequence enter a $k \times L$ shift registry. The bits are linearly combined to produce n bits. Hence, each n-bit output depends on the previous $k \times (L - 1)$ bits. In other words, convolutional codes have memory.

In this lecture, we will restrict ourselves to block codes.

Definition of linear block codes

An (n, k) block code $C = \{\mathbf{c}_1, \mathbf{c}_2 \ldots, \mathbf{c}_M\}$ is defined by a collection of $M = 2^k$ binary sequences of length n called *code words*. Instead of sending the original block of k bits, we send a code word.

- **Definition**: A block code is said to be **linear** if any linear combination of code words is also a code word. In the binary case, **linear combinations are defined as the component-wise modulo 2 addition** (i.e. 0+0=0, 1+1=0, 1+0=1, 0+1=1).

One consequence is that the zero sequence **0** is always a code word of any linear block code.

The (5,2) code defined by the following mapping is linear

$$00 \rightarrow 00000$$
$$01 \rightarrow 01111$$
$$10 \rightarrow 10100$$
$$11 \rightarrow 11011$$

Generator matrix

Given an (n, k) linear block code, any information sequence $\mathbf{x}$ can be mapped into its code word $\mathbf{c}$ by multiplying it by the generator matrix $\mathbf{G}$

$$\boldsymbol{G} \overset{\text{def}}{=} \begin{bmatrix} \boldsymbol{g_1} \\ \boldsymbol{g_2} \\ \vdots \\ \boldsymbol{g_k} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{11} & \cdots & g_{1n} \\ g_{21} & g_{22} & & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}$$

so that $\mathbf{c} = \mathbf{xG}$.

The generator matrix in our previous example is

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Systematic codes

In a systematic code, the code word consists of the information sequence followed by a sequence of $m = n - k$ bits, known as the parity bits. Hence, the generator matrix has the form

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}]$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix and $\mathbf{P}$ is the parity check matrix.

Parity check matrix

The parity check matrix **H** allows us to check whether a code word belongs to our code or not. It has the property that

$$\boldsymbol{cH}^t = 0$$

If the code is systematic, the parity check matrix can be obtained as

$$\mathbf{H} = [\, -\mathbf{P}^t \mid \mathbf{I}_{n-k} \,]$$

where *t* denotes transposition. **In the binary case, $-\mathbf{P}^t = \mathbf{P}^t$.** Hence, parity check matrices allow us to **detect errors** by determining whether a given received sequence is a code word or not.

Parity check matrix

In our example:

$$00 \rightarrow 00000$$
$$01 \rightarrow 01111$$
$$10 \rightarrow 10100$$
$$11 \rightarrow 11011$$

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{H} = \left[\begin{array}{cc|ccc} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Principles of block decoding

Decoding consists of recovering the initial information sequence based on the observation of a (probably) corrupted sequence of code words. During decoding we essentially **compare each received sequence with all the code words** defining the code and then **choose the most similar code word**.

The question is, how do we define similarity between binary sequences? Our **definition of distance is given by the Hamming distance**. The notion of similarity will also allow us to determine the maximum number of errors that we can correct.

Hamming distance

Two more definitions:

- **Definition**: The Hamming distance between two code words $\mathbf{c}_i$ and $\mathbf{c}_j$, $d(\mathbf{c}_i, \mathbf{c}_j)$ is the number of components at which they differ.
- **Definition**: The minimum distance of a code d_{min} is the **minimum Hamming distance** between any two code words.

The Hamming distance between code words 01111 and 10100 is $d(01111, 10100) = 4$. However, the minimum distance of the code is $d_{min} = 2$.

Hamming distance

How can we use the Hamming distance for decoding?

- Given an output sequence y, we can obtain the distance between this sequence and all of the code words, $d(y, c_i)$.
- We will decode y as the sequence $c_{\min}$ whose distance to y is shortest, i.e. as the most similar sequence.

In terms of error correction, a good code book will then be one such that its minimum distance $d_{\min}$ is high. Why?

Because the maximum number of errors that we are able to correct by this procedure, N_C, is related to $d_{\min}$ by $\boldsymbol{d_{min} \geq 2N_C + 1.}$

The **number of errors that can be detect**, N_D, is by contrast related to $d_{\min}$ by $\boldsymbol{d_{min} \geq N_D + 1}$.

Block decoding algorithm: syndrome decoding

Let us denote by **e** the error binary sequence. The output sequence y that we obtain when code word c is transmitted can be expressed as

$$\mathbf{y} = \mathbf{c} + \mathbf{e}.$$

If there are no errors during transmission, e = 0, if there is an error in the first bit, **e** = (10 . . . 0), if there is an error in the first and third bite = (1010 . . . 0), and so on.

If we apply the parity check to **y**, we get:

$$\mathbf{y}H^t = \mathbf{c}H^t + \mathbf{e}H^t = \mathbf{e}H^t$$

Notice that the result of this operation **depends on the error sequence e and not on the code word c that we have transmitted.**

Block decoding algorithm: syndrome decoding

The sequence $\mathbf{s} = \mathbf{e}\boldsymbol{H}^{\boldsymbol{t}}$ is called the syndrome. How many different syndrome sequences are there?

For **a *(n, k)*** code, $\boldsymbol{H}^{\boldsymbol{t}}$ has *n* rows and $\mathbf{m} = \mathbf{n} - \mathbf{k}$ columns. Hence, s is a $\mathbf{1} \times \mathbf{m}$ vector and there will exist $\mathbf{2}^{\boldsymbol{m}}$different syndrome sequences **s**.

If we can relate an error sequence e to one syndrome sequence ***s***, we can determine e based on the calculation $\mathbf{s} = \mathbf{y}\boldsymbol{H}^{\boldsymbol{t}}$ and the transmitted code word will be obtained as **c = y + e.**

Hence, we need as many syndrome sequences **s** as error sequences **e** we want to identify. For instance, if we want to be able to correct one single error in our sequence we need

$$\mathbf{2}^{\boldsymbol{m}} \geq \mathrm{n} + 1.$$

Example 1

For a (6,3) systematic linear block code, the three parity check digits are:

$$P_1 = 1 \times I_1 \oplus 1 \times I_2 \oplus 1 \times I_3$$
$$P_2 = 1 \times I_1 \oplus 1 \times I_2 \oplus 0 \times I_3$$
$$P_3 = 0 \times I_1 \oplus 1 \times I_2 \oplus 1 \times I_3$$

1. Construct the generator matrix G for this code.
2. Construct all the possible codewords generated by this matrix
3. Determine the error –correcting capabilities for this code
4. Write down the syndrome table for this code.
5. Decode the received words 101100,000110 and 101010.

Example 2

PROBLEM ♯8. Let $\mathbf{x}_i$, for $i = 1, 2, 3, 4$ be 4-bit sequences. The sequence of 16 bits $\mathbf{x} = \mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4$ is coded by a $(7, 4)$ Hamming code defined by the parity check matrix

$$\mathbf{H} = \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

After transmission, the sequence $\mathbf{r} = 0001110001010111000110000110$ is demodulated. Determine $\mathbf{x}_i$, for $i = 1, 2, 3, 4$ by decoding $\mathbf{r}$.

Solution-1

SOLUTION ♯8.

Firstly, we are going to identify the received sequences corresponding to each transmitted codeword. Let $\mathbf{r_i}$ be the received sequence corresponding to the transmitted codeword $\mathbf{c_i}$. By grouping $\mathbf{r}$ into 7-bits sequences we get:

$$\begin{aligned} \mathbf{r_1} &= 0001110 \\ \mathbf{r_2} &= 0010101 \\ \mathbf{r_3} &= 1100011 \\ \mathbf{r_4} &= 0000110 \end{aligned}$$

Next, we will calculate the syndrome sequence $\mathbf{s_i} = \mathbf{r_i H}^T$ corresponding to each received sequence. If the syndrome is zero, the received sequence is a codeword. Otherwise, it will contain errors.

$$\begin{aligned} \mathbf{s_1} &= 001 \\ \mathbf{s_2} &= 000 \\ \mathbf{s_3} &= 110 \\ \mathbf{s_4} &= 110 \end{aligned}$$

Since we are using a Hamming code, the minimum distance is 3 and the maximum number of errors that can be corrected is 1. Assuming that there has been up to one error, based on the

Solution-2

syndrome sequences, we can determine the error sequences by identifying them in $\mathbf{H}^T$:

$$\begin{aligned} \mathbf{e_1} &= 0000001 \\ \mathbf{e_2} &= 0000000 \\ \mathbf{e_3} &= 1000000 \\ \mathbf{e_4} &= 1000000 \end{aligned}$$

Hence, the codewords will be obtained as $\mathbf{c_i} = \mathbf{r_i} + \mathbf{e_i}$

$$\begin{aligned} \mathbf{c_1} &= 0001111 \\ \mathbf{c_2} &= 0010101 \\ \mathbf{c_3} &= 0100011 \\ \mathbf{c_4} &= 1000110 \end{aligned}$$

Finally, since this is a systematic $(7, 4)$ code, the first 4 bits correspond to the original information sequence:

$$\begin{aligned} \mathbf{x_1} &= 0001 \\ \mathbf{x_2} &= 0010 \\ \mathbf{x_3} &= 0100 \\ \mathbf{x_4} &= 1000 \end{aligned}$$

Summary

- Digital channel model
- PAM
- Channel capacity
  - Noiseless
  - Noisy
- Channel coding