通訊系統 (II)

國立清華大學電機系暨通訊工程研究所
蔡育仁
台達館 821 室
Tel: 62210
E-mail: yrtsai@ee.nthu.edu.tw

Chapter 7
Information Theory

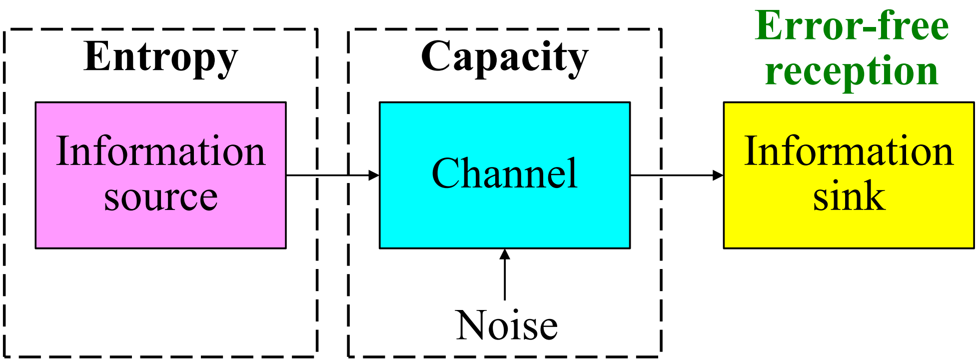
Introduction

Introduction

- In communications, **information theory** deals with modeling and analysis of a **communication system**
- In particular, it provides answers to two fundamental questions:
  - **Signal Source:** What is the irreducible **complexity**, below which a signal **cannot be compressed**?
  - **Channel:** What is the ultimate **transmission rate** for **reliable communication** over a **noisy channel**?
- The answers to these two questions lie in the **entropy of a source** and the **capacity of a channel**, respectively:
  - **Entropy:** the **probabilistic behavior** of a **source of information**
  - **Capacity:** the **intrinsic ability** of a **channel** to convey information (related to the **noise characteristics**)

Introduction

- If the entropy of the source is **less than** the capacity of the channel, then, ideally, **error-free communication** over the channel can be achieved.
- If the entropy of the source is **more than** the capacity of the channel, then, error-free communication over the channel is **impossible**.



Uncertainty, Information and Entropy

Entropy

- Suppose that a **probabilistic experiment** involves observation of the output emitted by a **discrete source** during every signaling interval.
- A **sample** of the source output is denoted by the **discrete random variable** S
  - with the fixed finite **alphabet** $\mathbb{S} = \{s_0, s_1, \cdots, s_{K-1}\}$
  - with probabilities $P(S = s_k) = p_k, \quad k = 0, 1, \cdots, K-1$
- We assume that the symbols emitted by the source during successive signaling intervals are **statistically independent**.
- How much **information** is produced by such a source?
  - The amount of information is closely related to that of **uncertainty**

Entropy (Cont.)

- Consider the event $S = s_k$, describing the emission of symbol s_k by the source with probability p_k
  - If the probability $p_k = 1$ and $p_i = 0$ for all $i \neq k$, then
    - There is **no information** when symbol s_k is emitted
  - If $0 < \boldsymbol{p_k} < \boldsymbol{p_i} < 1$, then there is **more information** when symbol s_k is emitted than when symbol s_i is emitted
    - The occurrence of a **rare event** implies more information
  - **Before** the event $S = s_k$ occurs, there is an amount of **uncertainty**. $\Rightarrow$ **After** the occurrence of the event $S = s_k$, there is gain in the amount of **information**.
- Most importantly, the amount of information is related to **the inverse of the probability of occurrence** of the event $S = s_k$.

Entropy (Cont.)

- The **amount of information** gained after observing the event $S = s_k$, which occurs with probability p_k, is defined as

$$I(s_k) = \log(1/p_k), \quad k = 0, 1, \cdots, K-1$$

- This definition exhibits the following important properties:
  - **Property 1**: $I(s_k) = 0, \quad \text{for } p_k = 1$
  - **Property 2**: $I(s_k) \geq 0, \quad \text{for } 0 \leq p_k \leq 1$
    - The occurrence of an event **never** brings about a **loss** of information
  - **Property 3**: $I(s_k) > I(s_i), \quad \text{for } p_k < p_i$
  - **Property 4** (the **additive property**): If s_k and s_l are **statistically independent**

$$I(s_k, s_l) = I(s_k) + I(s_l)$$

Entropy (Cont.)

- The base of the logarithm specifies the **units of information measure** (e.g., in bits)
- For **binary signaling** and with the information measure in **bits**
  - We use a logarithm of **base 2**

$$I(s_k) = \log_2(1/p_k) = -\log_2(p_k), \quad k = 0, 1, \cdots, K-1$$

  - When $p_k = 1/2$, we have $I(s_k) = 1$ bit
- **One bit** is the amount of information that we gain when one of two **equally likely** (i.e., **equiprobable**) events occurs.
  - s_0 = "0" with $p_0 = 0.5$ and s_1 = "1" with $p_1 = 0.5$

Entropy (Cont.)

- During **an arbitrary signaling interval**, the amount of information $I(s_k)$ depends on the symbol s_k emitted by the source at the time.
- $I(s_k)$ is a **discrete random variable** that takes on the values $I(s_0), I(s_1), \cdots, I(s_{K-1})$ with probabilities $p_0, p_1, \cdots, p_{K-1}$
- The **entropy** of the source is defined as the **expectation** of $I(s_k)$ over all the probable values taken by the random variable S

$$H(S) = E\left[I(s_k)\right] = \sum_{k=0}^{K-1} p_k I(s_k) = \sum_{k=0}^{K-1} p_k \log_2 (1/p_k)$$

  – It is a measure of the **average information content per source symbol**
- $H(S)$ is **independent** of the alphabet $\mathbb{S}$; it depends only on the **probabilities** of the symbols in the alphabet $\mathbb{S}$ of the source.

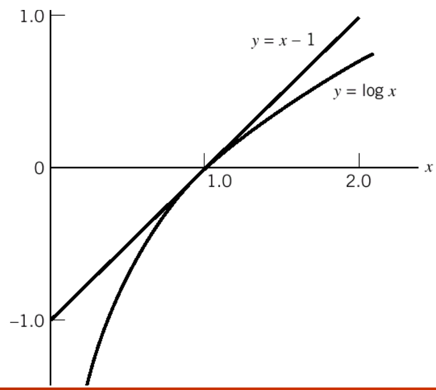
Properties of Entropy

- The entropy of the discrete random variable S is bounded

$$0 \le H(S) \le \log_2 K$$

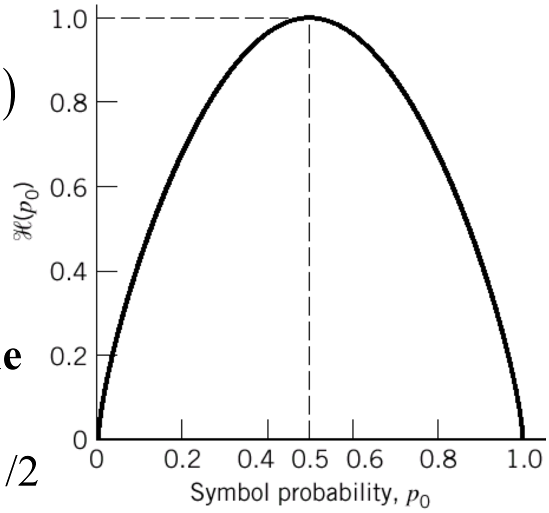
  – where K is the number of symbols in the alphabet $\mathbb{S}$
- $H(S) = 0$: if, and only if, the probability $p_k = 1$ for some k, and the remaining probabilities in the set are all zero
  – This **lower bound** corresponds to **no uncertainty**
- $H(S) = \log_2 K$: if, and only if, $p_k = 1/K$ for all k (i.e., all the symbols in the source alphabet $\mathbb{S}$ are **equiprobable**)
  – This **upper bound** corresponds to **maximum uncertainty**
- The proof needs the inequality:

$$\ln x \le x - 1, \quad x \ge 0$$



Example: Entropy of Binary Memoryless Source

- Consider a binary source for which symbol 0 occurs with probability p_0 and symbol 1 with probability $p_1 = 1 - p_0$
- We assume that the source is **memoryless** so that successive symbols are **statistically independent**
- $H(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$
  $= -p_0 \log_2 p_0 - (1-p_0)\log_2(1-p_0)$
  - When $p_0 = 0$, $H(S) = 0$
    - $x\log_2 x \to 0$ as $x \to 0$
  - When $p_0 = 1$, $H(S) = 0$
  - $H(S)$ attains its **maximum value** $H_{max} = 1$ bit when $p_1 = p_0 = 1/2$
  - $H(S)$ is **symmetric** about $p_0 = 1/2$



Extension of a Discrete Memoryless Source

- For high order modulation, one modulation symbol may contains **multiple** source symbols
  - To consider **blocks** rather than individual symbols
  - Each block consisting of ***n*** **successive** source symbols
- We may view each such block as being produced by an **extended source** with a **source alphabet** described by the **Cartesian product** of a set $\mathbb{S}^n$ that has K^n distinct blocks
  - where K is the number of distinct symbols in $\mathbb{S}$

$$\mathbb{S}^n = \left\{ \left(s^{(1)}, s^{(2)}, \cdots, s^{(n)} \right) \middle| s^{(i)} \in \mathbb{S} = \left\{ s_0, s_1, \cdots, s_{K-1} \right\}, 1 \le i \le n \right\}$$

- The probability of a symbol in $\mathbb{S}^n$ is equal to the **product** of the probabilities of the n source symbols in $\mathbb{S}$
  - The entropy of the **extended source**, is equal to n times $H(S)$

$$H(S^n) = nH(S)$$

Example: Entropy of Extended Source

- Consider a discrete memoryless source with source alphabet $\mathbb{S} = \{s_0, s_1, s_2\}$,
  - with the probabilities: $p_0 = 1/4$, $p_1 = 1/4$, $p_2 = 1/2$
- The entropy of the discrete random variable S is

$$H(S) = p_0 \log_2(1/p_0) + p_1 \log_2(1/p_1) + p_2 \log_2(1/p_2)$$
$$= 1/2 + 1/2 + 1/2 = 3/2 \text{ bits}$$

- Consider the second-order extension of the source: S^2
  - with source alphabet $\mathbb{S}^2$

Symbols of S^2	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8
Corresponding sequences of symbols of S	s_0s_0	s_0s_1	s_0s_2	s_1s_0	s_1s_1	s_1s_2	s_2s_0	s_2s_1	s_2s_2
Probability $P(\sigma_i)$	1/16	1/16	1/8	1/16	1/16	1/8	1/8	1/8	1/4

Example: Entropy of Extended Source (Cont.)

- Accordingly, the entropy of the extended source is

$$H(S^2) = \sum_{i=0}^{8} P(\sigma_i) \log_2(1/P(\sigma_i))$$
$$= \frac{1}{16}\log_2(16) + \frac{1}{16}\log_2(16) + \frac{1}{8}\log_2(8) + \frac{1}{16}\log_2(16)$$
$$+ \frac{1}{16}\log_2(16) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{4}\log_2(4)$$
$$= 3 \text{ bits} = 2 \times 3/2 \text{ bits}$$

Source-Coding

Source Coding

- **Source encoding**: The process used to **represent** the data generated by a discrete source of information
- The device that performs the representation is called a **source encoder**.
- We generally assume that the **statistics** of the source output are **known** for source encoding
  - For **frequent** source symbols: assigning **short** codewords
  - For **rare** source symbols: assigning **long** codewords
  - In order to **minimize** the **average symbol length**
  - We refer to such a source code as a **variable-length code**
- The **Morse code** is an example of a variable-length code.
  - Used in **telegraphy**

Source Coding (Cont.)

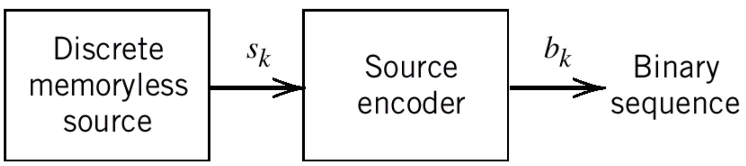
- For **digital communications**, a **source encoder** must satisfy two requirements:
  - The codewords produced by the encoder are in **binary form**
  - The source code is **uniquely decodable** (one-to-one mapping), so that the original source sequence can be **reconstructed** perfectly from the encoded binary sequence
- The second requirement is particularly important for a **perfect source code**
  - Otherwise, the transmission is **inherent in errors** even for a correct reception

Source Coding Efficiency

- Consider a discrete memoryless source whose output s_k is converted by the source encoder into a binary sequence b_k
  - The symbol s_k occurs with **probability** p_k, $k = 0, 1, \cdots, K-1$
  - The **length** of the binary codeword assigned to symbol s_k by the encoder is l_k (measured in **bits**)
- The **average codeword length** of the source encoder is

$$\bar{L} = \sum_{k=0}^{K-1} p_k l_k$$

  - The **average number of bits per source symbol** used in the source encoding process



Source Coding Efficiency (Cont.)

- Let $L_{\min}$ denote the **minimum possible** value of L.
  - But how to determine $L_{\min}$?
- According to **Shannon's source-coding theorem**:
  - Given a discrete memoryless source whose output is denoted by the random variable S, the **entropy** $H(S)$ imposes the following bound on the **average codeword length** $\bar{L}$ for any source encoding scheme: $\bar{L} \geq H(S)$
  - $H(S)$ represents a **fundamental limit** (**lower bound**) on $\bar{L}$
- The **coding efficiency** of the source encoder is defined as

$$\eta = L_{\min} / \bar{L} = H(S) / \bar{L}$$

  - Because $\bar{L} \geq L_{\min}$, we clearly have $\eta \leq 1$
- The source encoder is said to be **efficient** when $\eta \to 1$

Data Compaction

- A common characteristic of signals generated by **physical sources** is that, in their natural form, they contain a significant amount of **redundant information**
  - Direct transmission is **wasteful** of communication resources
- For **efficient signal transmission**, the redundant information should be **removed** from the signal prior to transmission.
  - This operation, with **no loss of information**, is ordinarily performed on a signal **in digital form**
  - Known as **data compaction** or **lossless data compression**
  - The source output is **efficient** in terms of the average number of bits per symbol
  - The original data can be reconstructed **with no loss of information**

Data Compaction (Cont.)

- Consider a discrete memoryless source of alphabet $\{s_0, s_1, \cdots, s_{K-1}\}$ and respective probabilities $\{p_0, p_1, \cdots, p_{K-1}\}$.
- After source coding, the code has to be **uniquely decodable**
  - For each **finite sequence of symbols**, the corresponding **sequence of codewords** is **different from** the sequence of codewords corresponding to any other source sequence
- Basically, data compaction is achieved by assigning **short (long) codewords** to the **most (less) frequent** outcomes
- We discuss some source-coding schemes for data compaction:
  - **Prefix coding**
  - **Huffman Coding**
  - **Lempel–Ziv Coding**

Prefix Coding

- In the **prefix coding** scheme, the codewords must satisfy a restriction known as the **prefix condition**.
- Let the codeword assigned to source symbol s_k be denoted by

$$\left(m_{k_1}, m_{k_2}, \cdots, m_{k_n}\right) \overset{e.g.}{=} \left(1, 0, 0, \cdots, 1\right)$$

  - where each element is **0 or 1** and n is the **codeword length**
- The **initial part** of the codeword is represented by the elements

$$m_{k_1}, m_{k_2}, \cdots, m_{k_i} \quad \text{for some } \; i \leq n$$

  - Any sequence made up of the initial part of the codeword is called a **prefix of the codeword**.

Prefix Prefix

Codeword **0 1 1 0 1 1 1 0 0 0** ⋯

- A **prefix code** is defined as a code in which **no codeword** is the prefix of **any other codeword**.

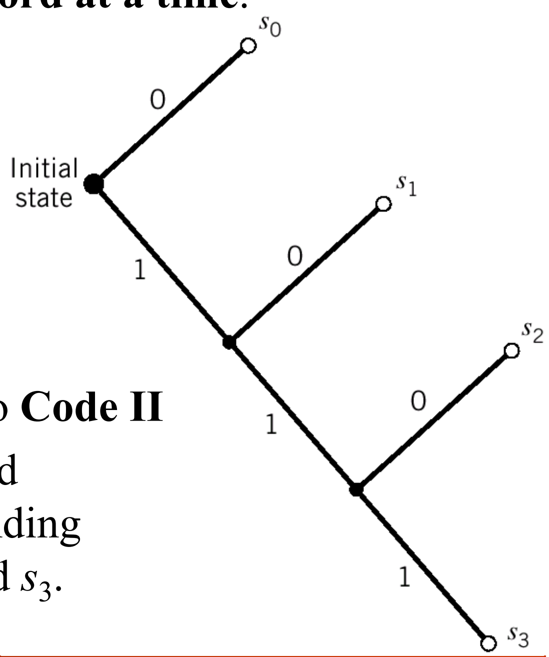
Prefix Coding (Cont.)

- Prefix codes are **distinguished from** other **uniquely decodable codes** by the fact that **the end of a codeword** is always **recognizable**.
  - The decoding can be accomplished **as soon as** the binary sequence representing a source symbol is **fully received**
  - Prefix codes are also referred to as **instantaneous codes**
- In the following example, Code II is a prefix code, but Code I and Code III are not. Code I is not a uniquely decodable code.

Source symbol	Probability of occurrence	Code I	Code II	Code III
s_0	0.5	0	0	0
s_1	0.25	1	10	01
s_2	0.125	00	110	011
s_3	0.125	11	111	0111

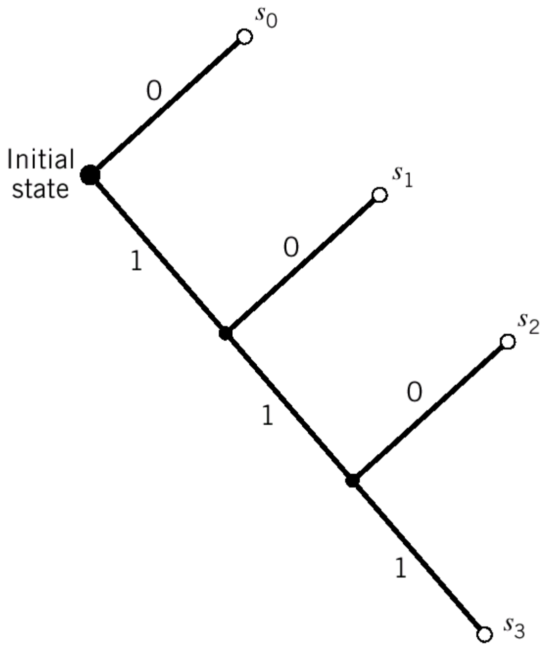
Decoding of Prefix Code

- The **source decoder** simply starts at the beginning of the sequence and **decodes one codeword at a time**.
- Specifically, it sets up what is equivalent to a **decision tree**
  - Starts at the **initial state**
  - Once a terminal state **emits** its symbol, the decoder is **reset** to its initial state
- The decision tree corresponding to **Code II**
  - The tree has an **initial state** and four **terminal states** corresponding to source symbols s_0, s_1, s_2, and s_3.



Decoding of Prefix Code (Cont.)

- Consider, for example, the following encoded sequence:
  - **1 0 1 1 1 1 1 0 0 0** ···
  - This sequence is readily decoded as the source sequence: $\boldsymbol{s_1\ s_3\ s_2\ s_0\ s_0}$ ···



Huffman Coding

- **Huffman codes**: an important class of **prefix codes**
  - A simple algorithm that computes an **optimal** prefix code for a **given distribution**
  - In the sense that the code has the **shortest expected length**
  - Construct a source code whose average codeword length approaches the fundamental limit set by **the entropy *H*(*S*)**
- The Huffman **encoding algorithm** proceeds as follows:
  - The **splitting stage**:
    - The source symbols are listed **in order of decreasing probability**.
    - The two source symbols of the **lowest probability** are assigned 0 and 1.

Huffman Coding (Cont.)

- The **reduction stage**:
  - These two source symbols are then **combined** into a **new source symbol**
  - The **probability** of the new symbol is equal to **the sum** of the two original probabilities.
  - The list of **source symbols**, as well as **source statistics**, is **reduced in size** by one.
- The procedure, the **splitting** and **reduction** stages, is **repeated** until **a final list** contains only **two** source statistics,
  - For which **(0, 1)** is an **optimal code**
- The code for each (**original**) source is found by working **backward** and **tracing the sequence of 0s and 1s**

Huffman Code Construction

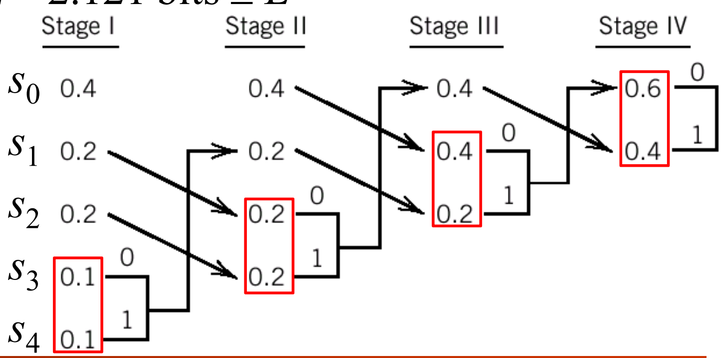
- Consider a discrete memoryless source with five symbols
- Following the Huffman algorithm, it reaches the end in **four steps**, resulting in a **Huffman tree** as follows
- The **average codeword length** is

$$\bar{L} = 0.4\times 2 + 0.2\times 2 + 0.2\times 2 + 0.1\times 3 + 0.1\times 3 = 2.2 \text{ bits/symbol}$$

- The entropy of the discrete memoryless source is

$$H(S) = 2.121 \text{ bits} \le \bar{L}$$

Symbol	Probability	Codeword
s_0	0.4	00
s_1	0.2	10
s_2	0.2	11
s_3	0.1	010
s_4	0.1	011



Huffman Code Construction (Cont.)

- It is noteworthy that the Huffman encoding process (i.e., the Huffman tree) is **not unique** (multiple sets of codewords)
- There are two variations in the process that are responsible for the **non-uniqueness** of the Huffman code:
  - First, **at each splitting stage**, there is arbitrariness in the assignment of "0" and "1" to the last two source symbols.
  - Second, ambiguity arises when the **probability** of a **combined** symbol is equal to another probability in the list.
    - We may proceed by placing the probability of the new symbol as **high** as possible or as **low** as possible.
- For different Huffman codes, the codewords of the same source symbol may have **different lengths**.
  - But the average codeword length **remains the same**

Lempel–Ziv Coding

- A **drawback** of the Huffman code is that it requires knowledge of a **probabilistic model of the source**
  - In practice, source statistics are not always known a priori
- In the modeling of **text**, capturing **higher-order relationships** between words and phrases requires an **extremely large** codebook $\Rightarrow$ Extremely high **storage requirement**
- To overcome these practical limitations, we may use the **Lempel–Ziv algorithm**
  - It is **adaptive** and **simpler** to implement than Huffman coding.
  - The source data stream is **parsed into segments**
    - The **shortest subsequences** not encountered previously

Lempel–Ziv Code Construction

- Consider the example of the binary sequence:
  - **0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 1 0 1 …**
- We assume that "**0**" and "**1**" are already stored in the codebook
  - Subsequences stored: **0, 1**
  - Data to be parsed: **0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 1 0 1 …**
- The **shortest subsequence** of the data stream encountered for the **first time and not seen before** is "**00**"
  - Subsequences stored: **0, 1, 00**
  - Data to be parsed: **0 1 0 1 1 1 0 0 1 0 1 0 0 1 0 1 …**
- The **second** shortest subsequence not seen before is "**01**"
  - Subsequences stored: **0, 1, 00, 01**
  - Data to be parsed: **0 1 1 1 0 0 1 0 1 0 0 1 0 1 …**

Lempel–Ziv Code Construction (Cont.)

- The **next** shortest subsequence not encountered before is "**011**"
  - Subsequences stored: **0, 1, 00, 01, 011**
  - Data to be parsed: **1 0 0 1 0 1 0 0 1 0 1 …**
- We continue in the process **until the given data stream has been completely parsed**.
  - Thus, we get the **codebook** of binary subsequences
- The data stream "00" is made up of the **first** codebook entry "0"
  - Therefore, it is represented by the number 11
- The subsequence, "01" consists of the **first** codebook entry "0" concatenated with the **second** codebook entry "1"
  - Therefore, it is represented by the number 12
- The subsequence "011" consists of "01" and "1" $\Rightarrow$ 42

Lempel–Ziv Code Construction (Cont.)

- The **last row** shows the **binary encoded representation** of the different subsequences of the data stream.
  - The **last symbol** of each **subsequence** in the codebook is an **innovation symbol** to distinguishes it from all previous subsequences stored in the code book
    - "00" and "01"; "011" and "010"; "100" and "101";
  - The **remaining bits** provide the "**pointer**" to the **root subsequence**
    - "001" for position 1 "0"; "100" for position 4 "01"

Numerical Positions:	1	2	3	4	5	6	7	8	9
Subsequences:	**0**	**1**	**00**	**01**	**011**	**10**	**010**	**100**	**101**
Numerical representations:			**11**	**12**	**42**	**21**	**41**	**61**	**62**
Binary encoded blocks:			**0010**	**0011**	**1001**	**0100**	**1000**	**1100**	**1101**

Lempel–Ziv Code Decoding

- The **Lempel–Ziv decoder** is just as simple as the encoder.
  - It uses the **pointer** to identify the **root subsequence**
  - Then appends the **innovation symbol**
- For example, the binary encoded block "1101" is received
  - "110" points to the root subsequence "10" in position 6
  - The last bit "1" is the innovation symbol
  - The decoded symbol is "101", which is correct.

Numerical Positions:	1	2	3	4	5	6	7	8	9
Subsequences:	**0**	**1**	**00**	**01**	**011**	**10**	**010**	**100**	**101**
Numerical representations:			**11**	**12**	**42**	**21**	**41**	**61**	**62**
Binary encoded blocks:			**0010**	**0011**	**1001**	**0100**	**1000**	**1100**	**1101**

Lempel–Ziv Coding VS Huffman Coding

- In contrast to Huffman coding, the Lempel–Ziv algorithm uses **fixed-length codes** to represent a variable number of source symbols
  - This feature makes the Lempel–Ziv code suitable for **synchronous transmission**.
- In practice, fixed blocks of **12 bits** long are used
  - A code book of $2^{12} = 4096$ entries.
- **Huffman coding** is still optimal, but in practice it is **hard to implement**.
- For practical implementation, the **Lempel–Ziv algorithm** has taken over almost completely from the Huffman algorithm.
  - **Lempel–Ziv algorithm** is the standard algorithm for file compression.

Discrete Memoryless Channels

Discrete Memoryless Channels

- Information generation: Discrete memoryless sources
- Information transmission: **Discrete memoryless channels**
- A discrete memoryless channel is a statistical model with an **input** X and an **output** Y
  - Y is a **noisy version** of X; both X and Y are random variables
- Every unit of time, the channel accepts an input symbol X selected from an alphabet $\mathcal{X}$ and, in response, it emits an output symbol Y from an alphabet $\mathcal{Y}$.
- The channel is said to be "**discrete**" when both of the alphabets $\mathcal{X}$ and $\mathcal{Y}$ have **finite sizes**.
- It is said to be "**memoryless**" when the current output symbol depends **only** on the current **input symbol**
  - **Not** any previous or future symbol

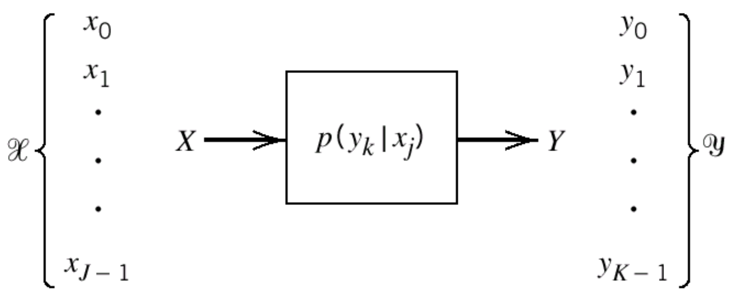
Discrete Memoryless Channels (Cont.)

- A discrete memoryless channel is described in terms of an
  input alphabet $\mathcal{X} = \{x_0, x_1, \cdots, x_{J-1}\}$
  an **output alphabet** $\mathcal{Y} = \{y_0, y_1, \cdots, y_{K-1}\}$
  and a set of **transition probabilities**

$$p(y_k | x_j) = P(Y = y_k | X = x_j), \quad 0 \le j \le J-1, 0 \le k \le K-1$$

  - According to probability theory, we naturally have

$$0 \le p(y_k | x_j) \le 1; \quad \sum_{k=0}^{K-1} p(y_k | x_j) = 1, \text{ for a fixed } j$$



Discrete Memoryless Channels (Cont.)

- A discrete memoryless channel can be described in the form of a **channel matrix** or **transition matrix**

$$\mathbf{P}=\begin{bmatrix} p(y_0|x_0) & p(y_1|x_0) & \cdots & p(y_{K-1}|x_0) \\ p(y_0|x_1) & p(y_1|x_1) & \cdots & p(y_{K-1}|x_1) \\ \vdots & \vdots & \ddots & \vdots \\ p(y_0|x_{J-1}) & p(y_1|x_{J-1}) & \cdots & p(y_{K-1}|x_{J-1}) \end{bmatrix}$$

- Suppose the event that the input $X=x_j$ occurs with probability (**prior probability**) $p(x_j)=P(X=x_j)$ for $j=0, 1, \cdots, J-1$
- The **joint probability distribution** of X and Y is given by

$$p(x_j, y_k)=P(X=x_j, Y=y_k)=p(y_k|x_j)\,p(x_j)$$

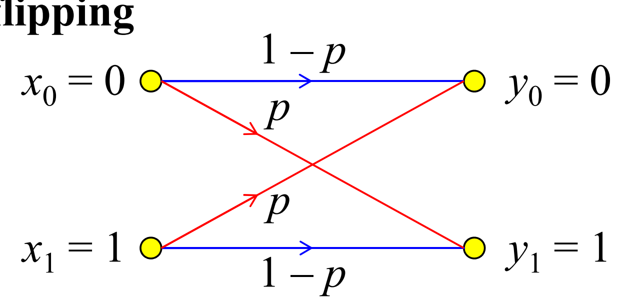
- The **marginal probability distribution** of the output Y is

$$p(y_k)=P(Y=y_k)=\sum_{j=0}^{J-1} p(y_k|x_j)\,p(x_j)$$

Binary Symmetric Channel

- The **binary symmetric channel** is a special case of the discrete memoryless channel with $J = K = 2$.
  - Two input symbols: $x_0 = 0$, $x_1 = 1$
  - Two output symbols: $y_0 = 0$, $y_1 = 1$
- The channel is **symmetric**: "the probability of receiving 1 if 0 is sent" is the same as "the probability of receiving 0 if 1 is sent"
- The **conditional probability of error** is denoted by p
  - The probability of a **bit flipping**
- The **transition matrix** is

$$\mathbf{P}=\begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$



Binary Symmetric Channel (Cont.)

- To describe the probabilistic nature of this channel, we need
  - The **a priori probabilities** of sending symbols '0' and '1'

$$P(x_0) = p_0; \quad P(x_1) = p_1; \quad p_0 + p_1 = 1$$

  - The **conditional probabilities of error**

$$P(y_1 | x_0) = P(y_0 | x_1) = p$$

- The probability of receiving symbol '0' (or '1') is given by

$$P(y_0) = P(y_0 | x_0) P(x_0) + P(y_0 | x_1) P(x_1) = (1-p) p_0 + p p_1$$
$$P(y_1) = P(y_1 | x_0) P(x_0) + P(y_1 | x_1) P(x_1) = p p_0 + (1-p) p_1$$

- Then, applying Bayes' rule, we obtain the two ***a posteriori probabilities***

$$P(x_0 | y_0) = P(y_0 | x_0) P(x_0) / P(y_0) = [(1-p) p_0] / [(1-p) p_0 + p p_1]$$
$$P(x_1 | y_1) = P(y_1 | x_1) P(x_1) / P(y_1) = [(1-p) p_1] / [p p_0 + (1-p) p_1]$$

Mutual Information

Conditional Entropy

- In a discrete memoryless channel, we know that
  - The channel output Y (selected from alphabet $\mathcal{Y}$) is a **noisy version** of the channel input X (selected from alphabet $\mathcal{X}$).
  - The entropy $H(X)$ is a measure of the **prior uncertainty** about the **discrete source output** X.
- How can we measure the uncertainty about X after **observing** Y?
  - The **conditional entropy** of X given that $Y = y_k$ is observed

$$H\left(X \middle| Y = y_k\right) = E_X\left[I\left(x_j \middle| Y = y_k\right)\right] = \sum_{j=0}^{J-1} p\left(x_j \middle| y_k\right)\log_2\left(1\middle/ p\left(x_j \middle| y_k\right)\right)$$

  - Depending on the value of $Y = y_k$
  - Because Y is a **random variable**, it takes on the values $H(X|Y = y_0)$, $H(X|Y = y_1)$, $\cdots$, $H(X|Y = y_{K-1})$
    - with probabilities $p(y_0)$, $p(y_1)$, $\cdots$, $p(y_{K-1})$, respectively.

Conditional Entropy (Cont.)

- The **conditional entropy** $H(X|Y)$ is the **expectation** of entropy $H(X|Y = y_k)$ over the output alphabet $\mathcal{Y}$

$$H\left(X \middle| Y\right) = E_Y\left[H\left(X \middle| Y = y_k\right)\right] = \sum_{k=0}^{K-1} H\left(X \middle| Y = y_k\right) p\left(y_k\right)$$

$$= \sum_{k=0}^{K-1}\sum_{j=0}^{J-1} p\left(x_j \middle| y_k\right) p\left(y_k\right)\log_2\left(1\middle/ p\left(x_j \middle| y_k\right)\right)$$

$$= \sum_{k=0}^{K-1}\sum_{j=0}^{J-1} p\left(x_j, y_k\right)\log_2\left(1\middle/ p\left(x_j \middle| y_k\right)\right)$$

- The conditional entropy, $H(X|Y)$, is the **average amount of uncertainty** remaining **about the channel input** X after the channel output Y has been **observed**.
  - Some **uncertainty** has been removed through transmission

Mutual Information

- The conditional entropy $H(X|Y)$ relates the **channel output** Y to the **channel input** X.
- The **entropy** $H(X)$ accounts for the uncertainty about the channel input **before observing** the channel output.
- The **conditional entropy** $H(X|Y)$ accounts for the uncertainty about the channel input **after observing** the channel output.
- The definition the **mutual information** of the channel:

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

  - It is a measure of the uncertainty about the **channel input**, which is **resolved by observing the channel output**.

Mutual Information (Cont.)

- The **mutual information** of a channel can also be defined as

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

  - It is a measure of the uncertainty about the **channel output** that is **resolved by sending the channel input**.
- Although the two definitions look different, but they could be used **interchangeably**.

Properties of Mutual Information

- **PROPERTY 1:** The mutual information of a channel is **symmetric** in the sense that

$$I(X;Y) = I(Y;X)$$

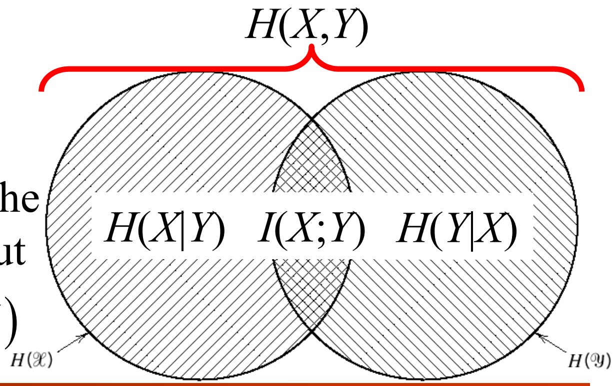
- **PROPERTY 2:** The mutual information is always **nonnegative**

$$I(X;Y) \geq 0$$

  – We **cannot lose** information, on the average, by observing the output of a channel.

- **PROPERTY 3:** The mutual information of a channel is related to the **joint entropy** of the channel input and channel output

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$



Channel Capacity

Channel Capacity

- Consider a discrete memoryless channel with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $p(y_k|x_j)$
- The **mutual information** of the channel is defined by

$$I(X;Y)=\sum_{k=0}^{K-1}\sum_{j=0}^{J-1} p(x_j,y_k)\log_2\left(p(y_k|x_j)\big/p(y_k)\right)$$

- Then, according to

$$p(x_j,y_k)=p(y_k|x_j)\,p(x_j);\quad p(y_k)=\sum_{j=0}^{J-1} p(y_k|x_j)\,p(x_j)$$

- Finally, we have

$$I(X;Y)=\sum_{k=0}^{K-1}\sum_{j=0}^{J-1} \underline{p(y_k|x_j)}\,\underline{p(x_j)}\log_2\left(\underline{p(y_k|x_j)}\Big/\sum_{j=0}^{J-1}\underline{p(y_k|x_j)}\,\underline{p(x_j)}\right)$$

- The mutual information $I(X;Y)$ depends on
  - The probability distribution of the **channel input** and
  - The transition probability distribution of the **channel**

Channel Capacity (Cont.)

- The two probability distributions $p(x_j)$ and $p(y_k|x_j)$ are obviously **independent** of each other.
- Given the channel's transition probability distribution $\{p(y_k|x_j\}$, the **channel capacity** is defined in terms of **the mutual information** between the channel input and output:

$$C=\max_{\{p(x_j)\}} I(X;Y) \quad \text{bits per channel use}$$

$$\text{Subject to}\quad p(x_j)\ge 0,\quad \forall\, j;\quad \sum_{j=0}^{J-1} p(x_j)=1$$

- The **channel capacity** of a discrete memoryless channel is defined as the **maximum mutual information** $I(X;Y)$ in **any single use of the channel** (i.e., signaling interval)
  - where the maximization is over **all possible** input probability distributions $\{p(x_j)\}$ on X.

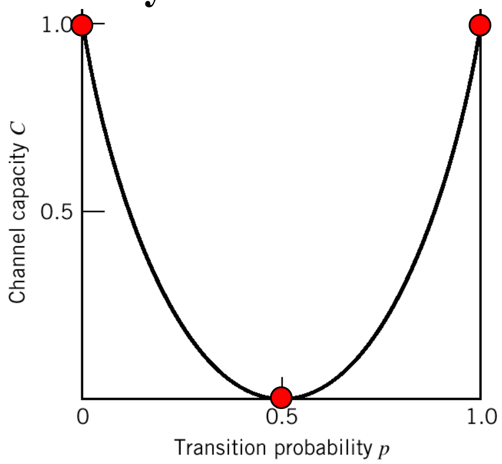
Channel Capacity of Binary Symmetric Channel

- Consider the **binary symmetric channel** defined by the conditional probability of error p.
- The entropy $H(X)$ is maximized when the channel input probability $p(x_0) = p(x_1) = 1/2$
  - The mutual information $I(X;Y)$ is **similarly maximized**
- Thus, the **channel capacity** is

$$C = I\left(X;Y\right)\Big|_{p(x_0)=p(x_1)=1/2}$$
$$= 1 + p\log_2 p + \left(1-p\right)\log_2\left(1-p\right)$$

- Using the definition of the entropy function, the channel capacity is

$$C = 1 - H\left(p\right)$$



Channel Capacity of BSC (Cont.)

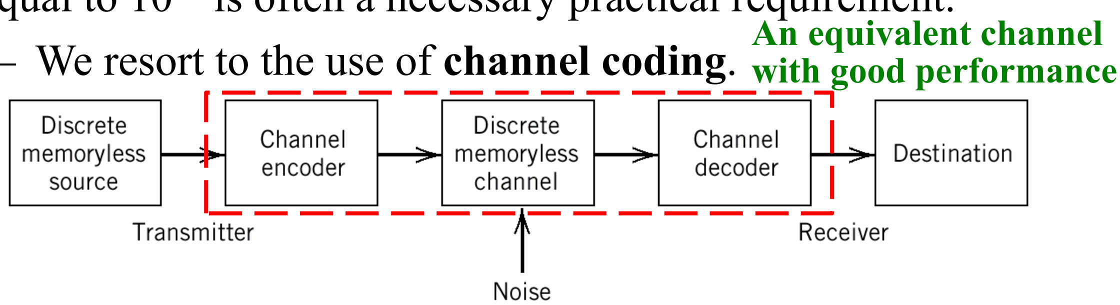
- The channel capacity C varies with the probability of error (i.e., transition probability) p in a **convex** manner
  - It is **symmetric** about $p = 1/2$
- When the channel is **noise free**, permitting us to set $\boldsymbol{p = 0}$
  - The channel capacity C attains its **maximum value** of **one bit** per channel use
  - Which is exactly the information in each channel input
- When the conditional probability of error $p = 1/2$
  - The channel capacity C attains its **minimum value** of **zero**
  - The channel is said to be **useless** in the sense that the channel input and output become **statistically independent**
- How about the condition with $\boldsymbol{p = 1}$?

Channel-Coding Theorem

Prof. Tsai

Channel Coding

- The inevitable presence of **noise** in a channel causes **detection errors** at the receiver of a digital communication system.
- In a relatively noisy channel with a low SNR (e.g., **wireless communication channels**), the probability of error may higher than 10^{-1} .
  - This level of **reliability** is generally **unacceptable**
- For many applications, a probability of error lower than or equal to 10^{-6} is often a necessary practical requirement.
  - We resort to the use of **channel coding**.



Channel Coding (Cont.)

- The design goal of channel coding is to **increase the resistance** of a digital communication system **to channel noise**.
- Specifically, **channel coding** consists of
  - At the **transmitter**: **Mapping** the incoming data sequence into a channel input sequence $\Rightarrow$ **channel encoder**
  - At the **receiver**: **inverse mapping** the channel output sequence into an output data sequence $\Rightarrow$ **channel decoder**
  - **Goal**: the overall effect of channel noise on the system is **minimized**
- The approach taken is to introduce **redundancy** in the channel encoder **in a controlled manner**, so as to **reconstruct** the original source sequence in the channel decoder **as accurately as possible**.

Channel Coding (Cont.)

- Consider one class of channel-coding: **block codes**
  - The message sequence is **subdivided** into sequential blocks
    - Each block contains k bits
  - Each k-bit block is **mapped into** an n-bit block, where $n > k$
  - The number of **redundant** bits added by the encoder to each transmitted block is $n - k$ bits
- The ratio k/n is called the **code rate**

$$r = k/n$$

  - where r is **less than unity**
  - For a prescribed k, the code rate r (and, therefore, the system's **coding efficiency**) **approaches zero** as the block length n **approaches infinity**.

Channel Coding (Cont.)

- To accurately reconstruct the original source sequence, the **average probability of symbol error** of the decoded data sequence must be **arbitrarily low**.
- Does a channel-coding scheme **exist**? such that,
  – The probability that a message bit will be in error is less than **any positive number** ε, and
  – The channel-coding scheme is **efficient** in that the **code rate** need not be too small
- The answer to this fundamental question is **"yes."**
  – The answer to the question is provided by **Shannon's second theorem** in terms of the **channel capacity** C

Channel-Coding Theorem

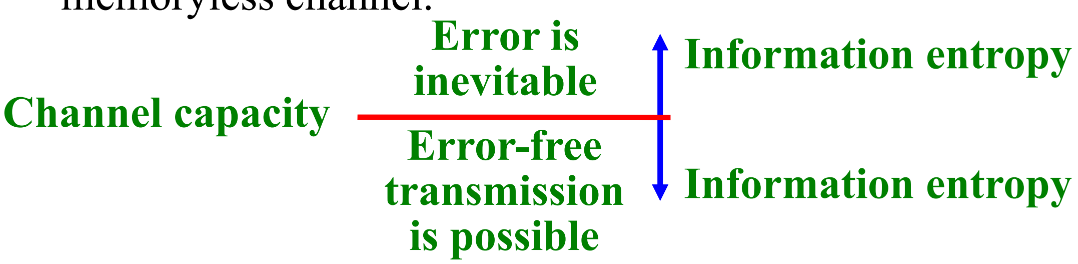
- Consider a discrete memoryless source that has the source alphabet $\mathbb{S}$ and entropy $H(S)$ bits per source symbol.
- Assume that the source **emits symbols** once every T_s seconds
  – The **average information rate**: $H(S)/T_s$ bits per second
  – The decoder delivers decoded symbols to the destination at **the same source rate** (i.e., one symbol every T_s seconds)
- The discrete memoryless channel has a **channel capacity** equal to C bits per use of the channel.
- Assume that the channel can be used once every T_c seconds
  – The **channel capacity per unit time**: C/T_c bits per second
  – The **maximum rate** of information transfer over the channel to the destination: C/T_c bits per second

Channel-Coding Theorem (Cont.)

- Shannon's second theorem: the **channel-coding theorem**
- Let a **discrete memoryless source** with an alphabet $\mathbb{S}$ have entropy $H(S)$ for random variable S and produce symbols once every T_s seconds.
- Let a **discrete memoryless channel** have capacity C and be used once every T_c seconds.
- Then, if $$H(S)/T_s \le C/T_c$$ there exists a **coding scheme** for which the source output can be transmitted over the channel and be reconstructed with an arbitrarily small probability of error.
- The parameter C/T_c is called the **critical rate**.
  - When $H(S)/T_s = C/T_c$, the system is said to be signaling at the critical rate.

Channel-Coding Theorem (Cont.)

- Conversely, if $$H(S)/T_s > C/T_c$$ it is **not possible** to transmit information over the channel and reconstruct it **with an arbitrarily small probability of error**.
- The channel-coding theorem is the single **most important** result of information theory.
  - The theorem specifies the channel capacity C as a **fundamental limit** on the rate at which the transmission of reliable **error-free** messages can take place over a discrete memoryless channel.



Channel-Coding Theorem (Cont.)

- However, it is important to note **two limitations** of the theorem:
- The channel-coding theorem **does not** show us **how to construct a good code**.
  - The theorem should be viewed as an **existence proof** in the sense that
  - If $H(S)/T_s \le C/T_c$ is satisfied, then good codes do exist.
- The theorem **does not** have a **precise result** for the probability of symbol error after decoding the channel output.
  - The theorem only tells us that the probability of symbol error **tends to zero** as the length of the code **increases**
    - Providing that the condition $H(S)/T_s \le C/T_c$ is satisfied

Channel-Coding Theorem for BSC

- Consider a **discrete memoryless source** that emits equally likely binary symbols (0s and 1s) once every T_s seconds.
  - The source entropy: one bit per source symbol
  - The information rate: $1/T_s$ bits per second
- The source sequence is applied to a **channel encoder**
  - The **code rate**: r
  - The **encoded symbol rate**: $1/T_c$ symbols per second
- The channel encoder engages a **binary symmetric channel** once every T_c seconds.
  - The channel capacity per unit time: C/T_c bits per second
  - C is determined by the channel transition probability p

$$C = 1 + p\log_2 p + (1-p)\log_2(1-p)$$

Channel-Coding Theorem for BSC (Cont.)

- According to the channel-coding theorem, if

$$1/T_s \le C/T_c$$

  - The probability of error can be made **arbitrarily low** by the use of a suitable **channel encoding scheme**
- The ratio T_c/T_s equals the **code rate** of the channel encoder

$$r = T_c/T_s$$

$\boldsymbol{k}$ bits $\Rightarrow kT_s$ sec
$\Rightarrow \boldsymbol{kT_s/T_c}$ bits $= n$ bits
$\Rightarrow r = k/n = T_c/T_s$

- Hence, we may restate the condition

$$1/T_s \le C/T_c \quad \Rightarrow \quad r \le C$$

- That is, for $r \le C$, there exists a code (**with code rate $\boldsymbol{r}$ less than or equal to channel capacity $\boldsymbol{C}$**) capable of achieving an arbitrarily low probability of error.

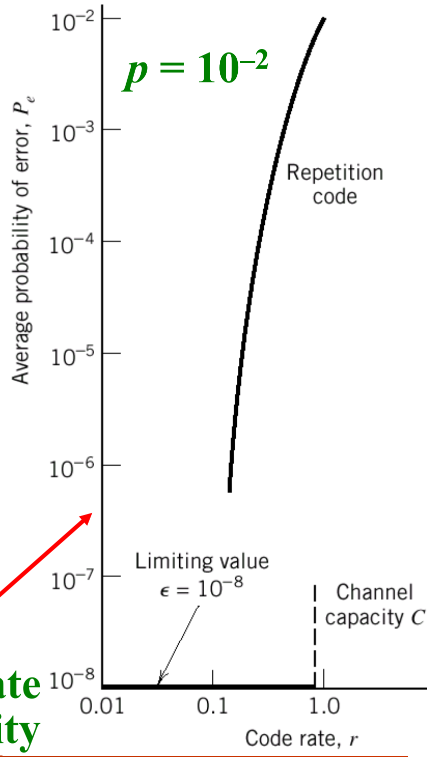
Example: Repetition Code

- Consider a BSC with transition probability $p = 10^{-2}$
  - The channel capacity: $C = 0.9192$
- Hence, for any $\varepsilon > 0$ and $r \le C = 0.9192$, there exists a code of **large enough** length n, code rate r, and an appropriate decoding algorithm, such that,
  - When the coded bit stream is sent over the given channel, the average probability of decoding error is less than ε.
- In the following, we consider a simple coding scheme that involves the use of a **repetition code**.
  - Each bit of the message is **repeated several times**.
  - Let each bit (0 or 1) be repeated n (an **odd** integer) times.
  - For example, for $n = 3$, we transmit **0** and **1** as **000** and **111**

Example: Repetition Code (Cont.)

- Intuitively, the channel decoder uses a **majority rule** for decoding:
  - In a block of n **repeated bits**, if the number of 0s exceeds the number of 1s, the decoder decides in favor of a 0
  - Otherwise, it decides in favor of a 1
- An **error** occurs when $m + 1$ or more bits out of $n = 2m + 1$ bits are received **incorrectly**.
- The **average probability of error** is

$$P_e = \sum_{i=m+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$



Exchange of code rate for message reliability

Differential Entropy and Mutual Information for Continuous Ensembles

Differential Entropy

- In addition to the **discrete sources**, we extend these concepts to **continuous** random variables.
  - For the description of another **fundamental limit** in information theory
- Consider a continuous random variable X with the **probability density function** $f_X(x)$.
- We define the **differential entropy** of X as

$$h(X) = \int_{-\infty}^{\infty} f_X(x) \log_2 \left[1 / f_X(x) \right] dx$$

  - To distinguish from the ordinary or absolute **entropy**
  - Although $h(X)$ is a useful mathematical quantity to know, it is **not**, in any sense, **a measure of the randomness** of X.

Differential Entropy (Cont.)

- In a limiting form, the **continuous** random variable X is viewed as a **discrete** random variable
  - $x_k = k\Delta x$, where $k = 0, \pm 1, \pm 2, \cdots$, and Δx approaches zero
- By definition, X is in the interval $[x_k, x_k + \Delta x]$ with **probability** $f_X(x_k)\, \Delta x$
- The **ordinary entropy** of the **continuous random variable** X takes the limiting form:

$$H(X) = \lim_{\Delta x \to 0} \sum_{k=-\infty}^{\infty} f_X(x_k) \Delta x \log_2 \left(\frac{1}{f_X(x_k) \Delta x} \right)$$

$$= \lim_{\Delta x \to 0} \left[\sum_{k=-\infty}^{\infty} f_X(x_k) \log_2 \left(\frac{1}{f_X(x_k)} \right) \Delta x - \sum_{k=-\infty}^{\infty} f_X(x_k) \Delta x \times \log_2 \Delta x \right]$$

Differential Entropy (Cont.)

$$H(X) = \int_{-\infty}^{\infty} f_X(x_k) \log_2 \left(\frac{1}{f_X(x_k)} \right) dx - \lim_{\Delta x \to 0} \left[\log_2 \Delta x \int_{-\infty}^{\infty} f_X(x_k)\, dx \right]$$
$$= h(X) - \lim_{\Delta x \to 0} \log_2 \Delta x$$

- In the limit as Δx **approaching zero**, the term "– $\log_2 \Delta x$" approaches **infinity**.
  - The entropy $H(X)$ of a **continuous** random variable X is **infinity**
- The evaluation of entropy for a **continuous** random variable is **infeasible**
- We only adopt the **differential entropy** $h(X)$ as a measure
  - The term "– $\log_2 \Delta x$" is regarded as **a reference**

Differential Entropy (Cont.)

- When we have a **continuous** random vector **X** consisting of n random variables $X_1, X_2, \cdots, X_n$, we define the differential entropy of **X** as the ***n*-fold integral**

$$h(\mathbf{X}) = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log_2 \left(\frac{1}{f_{\mathbf{X}}(\mathbf{x})} \right) d\mathbf{x}$$

  - where $f_{\mathbf{X}}(\mathbf{x})$ is the joint probability density function of **X**

Channel capacity Based on Differential Entropy

- **Channel capacity** is the information transmitted over a channel
  - The **mutual information** between channel input and output
  - Evaluation of **channel capacity** based on entropy for a **continuous channel** is impossible
- For the **difference** between **two entropy terms** that have a **common reference**, the information will be **the same** as the difference between the corresponding **differential entropy terms**

$$H(X)-H(Y)=\left[h(X)-\lim_{\Delta x\to 0}\log_2 \Delta x\right]-\left[h(Y)-\lim_{\Delta y\to 0}\log_2 \Delta y\right]$$
$$=h(X)-h(Y) \quad \text{for } \Delta x=\Delta y$$

  - "$-\log_2 \Delta x$" is the common reference for input and output
- **Channel capacity** is evaluated based on **differential entropy**

Example: Uniform Distribution

- Consider a random variable X uniformly distributed over the interval $(0, a)$.
- The probability density function of X is

$$f_X(x)=\begin{cases} 1/a, & 0<x<a \\ 0, & \text{otherwise} \end{cases}$$

- The **differential entropy** of X is

$$h(X)=\int_0^a (1/a)\times\log_2(a)\,dx=\log_2(a)$$

- Note that $\log_2 a<0$ for $a<1$.
- Unlike a discrete random variable, the **differential entropy** of a **continuous** random variable can be a **negative** value.
  - The value of a **pdf** could be **larger than 1**

Relative Entropy of Continuous Distributions

- Consider a pair of continuous random variables X and Y whose respective **probability density functions** are denoted by $f_X(x)$ and $f_Y(x)$ for the same dummy variable (argument) x.
- The **relative entropy** of the random variables X and Y is defined by

$$D\left(f_Y \middle| f_X\right) = \int_{-\infty}^{\infty} f_Y(x) \log_2 \left[f_Y(x) / f_X(x) \right] dx$$

  – where $f_X(x)$ is viewed as the **"reference" distribution**
- Based on some fundamental properties, we have $D(f_Y | f_X) \geq 0$
- Hence, we have the **differential entropy** of Y

$$D\left(f_Y \middle| f_X\right) = -\int_{-\infty}^{\infty} f_Y(x) \log_2 \left[1/ f_Y(x) \right] dx + \int_{-\infty}^{\infty} f_Y(x) \log_2 \left[1/ f_X(x) \right] dx \geq 0$$

$$\Rightarrow h(Y) = \int_{-\infty}^{\infty} f_Y(x) \log_2 \left[1/ f_Y(x) \right] dx \leq \int_{-\infty}^{\infty} f_Y(x) \log_2 \left[1/ f_X(x) \right] dx$$

Example: Gaussian Distribution

- Suppose two random variables, X and Y
  – X and Y have the common mean μ and variance σ^2
  – X is **Gaussian distributed**

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right]$$

- By changing the base of the logarithm from 2 to $e = 2.7183$

$$h(Y) \leq \log_2 e \times \int_{-\infty}^{\infty} f_Y(x) \left[\frac{(x-\mu)^2}{2\sigma^2} + \ln\left(\sqrt{2\pi}\sigma\right) \right] dx$$

  – where e is the base of the **natural logarithm**
- Given the mean μ and variance σ^2 of Y, we have

$$\int_{-\infty}^{\infty} f_Y(x)\, dx = 1; \quad \int_{-\infty}^{\infty} (x-\mu)^2 f_Y(x)\, dx = \sigma^2$$

Example: Gaussian Distribution (Cont.)

- Therefore, $$h(Y) \le \frac{1}{2}\log_2\left(2\pi e\sigma^2\right) = h(X)$$
- If X is a **Gaussian** random variable and Y is a **non-Gaussian** random variable, then $h(Y) < h(X)$

$$\boxed{\begin{aligned} h(Y) &\le \log_2 e \times \int_{-\infty}^{\infty} f_Y(x)\left[\frac{(x-\mu)^2}{2\sigma^2} + \ln\left(\sqrt{2\pi}\sigma\right)\right]dx \\ &= \log_2 e \times \left[\frac{1}{2\sigma^2}\int_{-\infty}^{\infty}(x-\mu)^2 f_Y(x)\,dx + \ln\left(\sqrt{2\pi}\sigma\right)\int_{-\infty}^{\infty} f_Y(x)\,dx\right] \\ &= \log_2 e \times \left[\frac{1}{2}\ln(e) + \frac{1}{2}\ln\left(2\pi\sigma^2\right)\right] = \frac{1}{2}\log_2 e \times \ln\left(2\pi e\sigma^2\right) \\ &= \frac{1}{2}\log_2\left(2\pi e\sigma^2\right) = h(X) \end{aligned}}$$

Example: Gaussian Distribution (Cont.)

- Two entropic properties of a random variable:
  - For any finite variance, a Gaussian random variable has the **largest differential entropy** attainable by any other random variable.
  - The entropy of a Gaussian random variable is **uniquely determined** by its **variance** (i.e., independent of the mean).

$$\boxed{h(Y) \le \frac{1}{2}\log_2\left(2\pi e\sigma^2\right) = h(X)}$$

- The **Gaussian channel model** is widely used as a **conservative model** in the study of digital communication systems.

Mutual Information of Continuous Distributions

- The **mutual information** between a pair of **continuous** random variables X and Y is defined as follows:

$$I(X;Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2\left[\frac{f_{X|Y}(x|y)}{f_X(x)}\right]dx\,dy$$

  – where $f_{X|Y}(x|y)$ is the **conditional pdf** of X given $Y = y$

- The mutual information between the pair of **Gaussian** random variables has the following properties:

$$I(X;Y)=I(Y;X);\quad I(X;Y)\geq 0;$$

$$I(X;Y)=h(X)-h(X|Y)=h(Y)-h(Y|X)$$

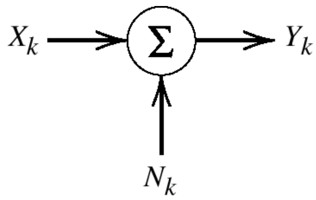
  – where the **conditional differential entropy** of X given Y is

$$h(X|Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\log_2\left[\frac{1}{f_{X|Y}(x|y)}\right]dx\,dy$$

Information Capacity Theorem

Band-limited, Power-limited Gaussian Channel

- In the following, we formulate the **information capacity** for a **band-limited**, **power-limited Gaussian channel**
- To be specific, consider a zero-mean stationary process $X(t)$ that is **band-limited** to B Hz
  - X_k, $k = 1, 2, \cdots, K$: the **continuous** random variables obtained by uniformly sampling at a rate of $2B$ samples per second
  - $2B$ samples per second: the Nyquist rate
- Suppose that the K samples are transmitted in T seconds over a noisy channel, also **band-limited** to B Hz $\Rightarrow K = 2BT$
- The channel output is perturbed by **additive white Gaussian noise** (**AWGN**) of zero mean and power spectral density $N_0/2$

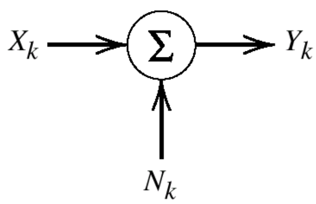


Band-limited, Power-limited Gaussian Channel

- The corresponding samples of the channel output Y_k are

$$Y_k = X_k + N_k, \quad k = 1, 2, \cdots, K$$

  - N_k is **Gaussian** distributed with zero mean and variance N_0B
  - Y_k, $k = 1, 2, \cdots, K$, are **statistically independent**
  - A **discrete-time, memoryless Gaussian channel**
- Typically, the transmitter is **power limited**
  - Define the cost as $E[(X_k)^2] = P$, $k = 1, 2, \cdots, K$
    - where P is the **average transmitted power**
- The **power-limited Gaussian channel** models many communication channels
  - Including **line-of-sight (LOS) radio** and **satellite links**



Information Capacity

- The **information capacity** of the channel is defined as the **maximum** of the **mutual information** between the channel input X_k and the channel output Y_k
  - Over the distributions of X_k that satisfy the **power constraint**

$$C = \max_{f_{X_k}(x)} I(X_k;Y_k), \quad \text{Subject to} \quad E\left[X_k^2\right] = P$$

  - $I(X_k;Y_k)$: the **mutual information** between X_k and Y_k

$$I(X_k;Y_k) = h(Y_k) - h(Y_k | X_k)$$

- Since X_k and N_k are **independent** random variables

$$h(Y_k | X_k) = h(N_k)$$

$$\Rightarrow I(X_k;Y_k) = h(Y_k) - h(N_k)$$

Information Capacity (Cont.)

$$I(X_k;Y_k) = h(Y_k) - h(N_k)$$

- With $h(N_k)$ being **independent** of the distribution of X_k
  - **Maximizing** $I(X_k; Y_k)$ is equivalent to **maximizing** the **differential entropy** $h(Y_k)$
- To maximize $h(Y_k)$, Y_k has to be a **Gaussian** random variable
  - The channel output must be a **noise-like** process
  - Since N_k is **Gaussian** by assumption, the sample X_k of the channel input must be **Gaussian** too
- We may state that **maximizing** the **information capacity** is
  - To choose samples of the channel input from a **noise-like Gaussian-distributed** process of average power P

$$C = I(X_k;Y_k) : X_k \text{ Gaussian}, \quad \text{Subject to} \quad E\left[X_k^2\right] = P$$

Information Capacity (Cont.)

- For the evaluation of the **information capacity** C
  - The variance of output sample Y_k equals $P + \sigma^2$
    - Because X_k and N_k are statistically **independent**
    - The **differential entropy** is

$$h(Y_k) = \frac{1}{2}\log_2\left[2\pi e\left(P+\sigma^2\right)\right]$$

  - The variance of the noisy sample N_k equals σ^2; hence,

$$h(N_k) = \frac{1}{2}\log_2\left(2\pi e\sigma^2\right)$$

  - Accordingly, the **information capacity** of the channel, in bits **per channel use**, is

$$C' = \frac{1}{2}\log_2\left[2\pi e\left(P+\sigma^2\right)\right] - \frac{1}{2}\log_2\left[2\pi e\sigma^2\right] = \frac{1}{2}\log_2\left(1+P/\sigma^2\right)$$

Information Capacity (Cont.)

- With the channel used K times for the transmission of K samples of the process $X(t)$ in T seconds, we find that
  - The information capacity **per unit time** is (K/T) times C'
  - The number K equals $2BT$
  - The information capacity of the channel **per unit time** is

$$C = B\log_2\left[1+P/\left(N_0B\right)\right]$$

    - where N_0B is the total noise power at the channel output
- The **information capacity** of a continuous channel of bandwidth B Hz, perturbed by AWGN of power spectral density $N_0/2$ and limited in bandwidth to B, is given by

$$C = B\log_2\left(1+\frac{P}{N_0B}\right)$$

Information Capacity (Cont.)

- The **information capacity law** is one of the most remarkable results of Shannon's information theory.
- The information capacity C **depends** on three key system parameters: **channel bandwidth**, **average transmitted power**, and **power spectral density of channel noise**.
  - The dependence of C on **channel bandwidth** B is **linear**
  - The dependence of C on **signal-to-noise ratio** $P/(N_0B)$ is **logarithmic**

$$C = B\log_2\left(1 + \frac{P}{N_0 B}\right)$$

Information Capacity (Cont.)

- To **increase** the information capacity of a continuous communication channel
  - By **expanding the bandwidth**: much **easier**
  - By **increasing the transmitted power**: harder
  - **Bandwidth** and **power** are the two major resources
- It is **not possible** for **error-free transmission** at a rate **higher than** C bits per second by **any encoding system**.
- Hence, the channel capacity law defines the **fundamental limit** on the permissible rate of **error-free transmission** for a **power-limited**, **band-limited Gaussian** channel.
  - To approach this limit, the transmitted signal must have statistical properties approximating those of **white Gaussian noise**.

Sphere Packing

Channel Capacity for Digital Modulation

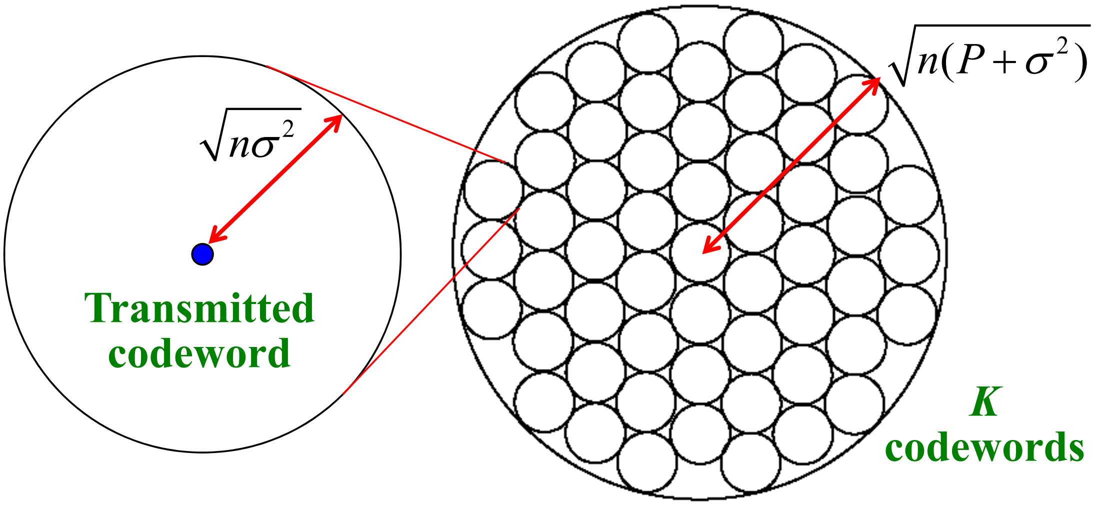
- Are the conventional digital modulation signals **noise-like Gaussian-distributed** signals?
  - **No!** The achieved capacity is **lower** than the information capacity of a continuous communication channel.

Sphere Packing

- Suppose that we use an encoding scheme that yields K **codewords**, one for each sample of the transmitted signal.
  - n: the number of bits per codeword
  - P: the average power per bit (the power constraint)
  - σ^2: the noise variance
  - The **transmission power** of each codeword with n bits is nP
  - The **coding scheme** is designed to produce an acceptably low probability of symbol error
  - The **received signal vector** of n bits at the channel output is **Gaussian distributed** with a mean equal to the transmitted codeword and a variance equal to $n\sigma^2$

Sphere Packing (Cont.)

- With a high probability, the received signal vector lies inside a sphere of radius $\sqrt{n\sigma^2}$, centered on the transmitted codeword.
- This sphere is contained in a larger sphere of radius $\sqrt{n(P+\sigma^2)}$
  - $n(P + \sigma^2)$ is the average power of the received signal vector



Sphere Packing (Cont.)

- The probability that the received signal vector will lie inside the **correct "decoding" sphere** is high.
- The key question is: "**How many** decoding spheres can be packed inside the larger sphere of received signal vectors?"
- Basic assumptions:
  - **No overlapping** between the decoding spheres (**uniqueness**)
  - The volume of an ***n*-dimensional sphere** of radius r as $A_n r^n$
    - where A_n is a **scaling factor**
- The volume of the sphere of received signal vectors (i.e., the larger sphere) is $A_n [n(P + \sigma^2)]^{n/2}$
- The volume of the decoding sphere (i.e., a small sphere) is $A_n (n\sigma^2)^{n/2}$

Sphere Packing (Cont.)

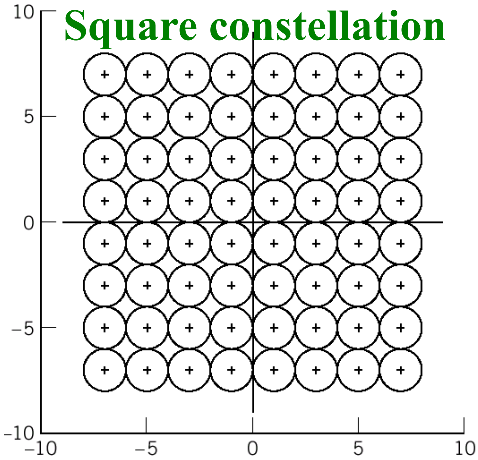
- The **maximum** number of **nonintersecting** decoding spheres is
  - The **number of codewords** that can be packed inside the sphere of possible received signal vectors

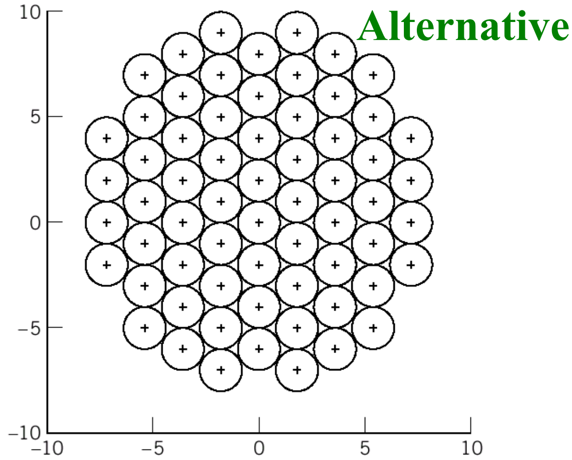
$$\frac{A_n\left[n(P+\sigma^2)\right]^{n/2}}{A_n\left[n\sigma^2\right]^{n/2}} = \underbrace{\left(1+\frac{P}{\sigma^2}\right)^{n/2}}_{K} = 2^{\underbrace{(n/2)\log_2\left(1+P/\sigma^2\right)}_{\text{Number of bits}}}$$

- The **maximum number of bits** per transmission for a low probability of error is indeed as the **information capacity**.
  - However, it is under the ideal assumption that the received signal vector lies inside a sphere of radius $\sqrt{n\sigma^2}$
  - It provides an **upper bound** on the physically realizable information capacity of a communication channel.

Reconfiguration of 64-QAM Constellation

- The **alternative** constellation packs the decoding spheres **as tightly as possible**
  - While maintaining **the same minimum Euclidean distance**
  - A **smaller** average transmitted signal energy per symbol for **the same** bit error rate over an AWGN channel





Implications of the Information Capacity Law

Information Capacity Law

- Consider an **ideal system** that transmits data at a bit rate R_b **equal to** the **information capacity** C.
- The average transmitted power is $P = E_b C$
  - where E_b is the **transmitted energy per bit**
- Accordingly, the ideal system is defined by the equation

$$\frac{C}{B} = \log_2\left(1 + \frac{E_b C}{N_0 B}\right)$$

- The **signal energy-per-bit** to **noise power spectral density ratio**, E_b/N_0, in terms of the ratio C/B for the ideal system is

$$\frac{E_b}{N_0} = \frac{2^{C/B} - 1}{C/B}$$

- **Bandwidth-efficiency diagram**: A plot of the bandwidth efficiency R_b/B versus E_b/N_0

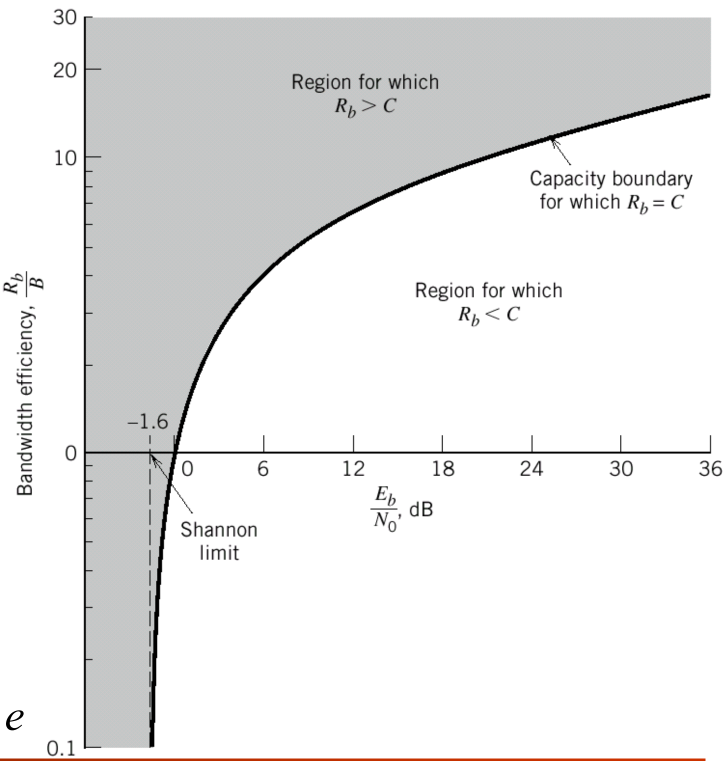
Bandwidth-Efficiency Diagram

- **1.** For **infinite channel bandwidth**, the SNR approaches the limit

$$\left(E_b/N_0\right)_\infty = \lim_{B\to\infty}\left(E_b/N_0\right)$$
$$= \ln 2 = 0.693 = -1.6 \text{ dB}$$

  - **Shannon limit** for an AWGN channel
- Minimum required SNR for efficient transmission
- The corresponding limiting value of the channel capacity

$$C_\infty = \lim_{B\to\infty} C = \left(P/N_0\right)\log_2 e$$



Bandwidth-Efficiency Diagram (Cont.)

- **2.** The **capacity boundary** is defined by the curve for the critical bit rate $R_b = C$.
  - For any point on this boundary, we have **error-free transmission** or not with probability of 1/2.
- **3.** The diagram highlights potential **trade-offs** among three quantities:
  - E_b/N_0, R_b/B, and the probability of symbol error P_e
  - For the operating point along a **horizontal line**: trading P_e versus E_b/N_0 for a fixed R_b/B
  - For the operating point along a **vertical line**: trading P_e versus R_b/B for a fixed E_b/N_0

Example: M-ary PCM

- Consider an M-ary PCM (pulse-code modulation) system
  - n: the number of **code elements** in each codeword
  - There are M^n **different codewords**
- The average transmission power is

$$P = \frac{2}{M}\left[\left(\frac{1}{2}\right)^2 + \left(\frac{3}{2}\right)^2 + \cdots + \left(\frac{M-1}{2}\right)^2\right](k\sigma)^2 = k^2\sigma^2\left(\frac{M^2-1}{12}\right)$$

  - k is a constant for successfully decoding
  - $\sigma^2 = N_0B$: the noise variance measured in a bandwidth B
- Suppose that the bandwidth of the message signal is W and the number of quantization levels is L
- The **maximum rate** of information transmission over the PCM system is $R_b = 2W\log_2 L$ bits per second

Example: *M*-ary PCM (Cont.)

- For a unique encoding process, we have $L = M^n$
- The rate of information transmission is

$$R_b = 2Wn\log_2 M \quad \text{bits per second}$$

- Solving the **number of discrete amplitude levels** under the average transmission power *P*, we have

$$M = \left(1 + \frac{12P}{k^2 N_0 B}\right)^{1/2}$$

- Therefore,

$$R_b = Wn\log_2\left(1 + \frac{12P}{k^2 N_0 B}\right) \quad \text{bits per second}$$

- The channel bandwidth *B* required to transmit a rectangular pulse of duration $1/(2nW)$ is $B = \kappa nW$
  - κ is a constant between 1 and 2 $\Rightarrow$ the minimum value is 1

Example: *M*-ary PCM (Cont.)

- Therefore, the minimum required channel bandwidth is $B = nW$
- Hence,

$$R_b = B\log_2\left(1 + \frac{12P}{k^2 N_0 B}\right) \quad \text{bits per second}$$

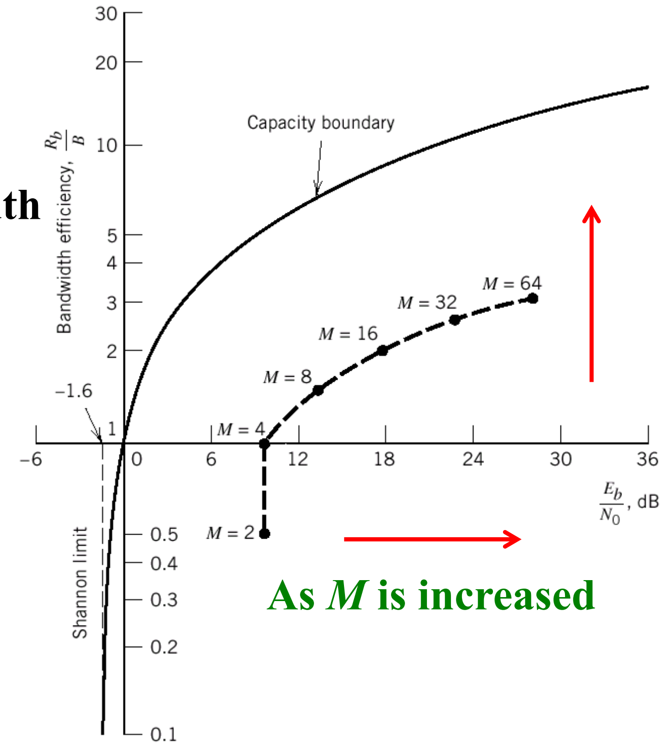
- If the average transmission power in the PCM system is **increased** by a factor of $k^2/12$,
  - The **maximum rate** of information transmission is identical to the capacity of the ideal system

Example: *M*-ary PSK

- Consider a coherent ***M*-ary PSK** system
  - Using the **null-to-null** bandwidth, the **bandwidth efficiency** is

$$\frac{R_b}{B} = \frac{\log_2 M}{2}$$

- The operating points correspond to an average probability of symbol error $P_e = 10^{-5}$
  - $M = 2, 4, 8, 16, 32, 64$

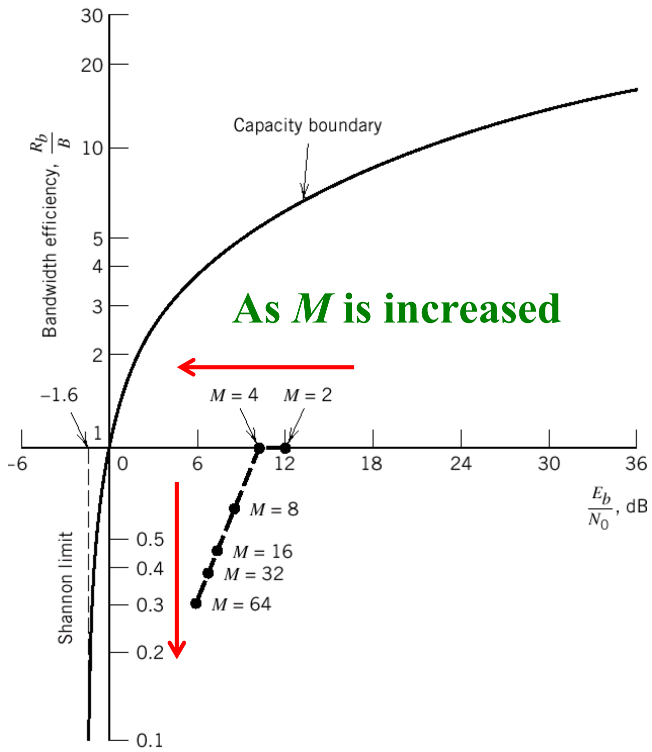


Example: *M*-ary FSK

- Consider a coherent ***M*-ary FSK** system
  - With the separation between adjacent frequencies $1/2T$, the **bandwidth efficiency** is

$$\frac{R_b}{B} = \frac{2\log_2 M}{M}$$

- The operating points correspond to an average probability of symbol error $P_e = 10^{-5}$
  - $M = 2, 4, 8, 16, 32, 64$



Capacity of Binary-Input AWGN Channel

- Consider the capacity of an **AWGN channel** using **encoded** (channel coding) binary antipodal signaling (i.e., '0': –1; '1': +1)
  - To determine the **minimum achievable bit error rate** as a function of E_b/N_0 for **varying code rate** r
- Let the random variables X and Y denote the **channel input** and **channel output** respectively
  - X is a **discrete** variable, whereas Y is a **continuous** variable
- The **mutual information** between X and Y

$$I(X;Y) = h(Y) - h(Y|X)$$

  - For Gaussian distributed noise with a variance σ^2

$$h(Y|\mathrm{X}) = \frac{1}{2}\log_2\left(2\pi e\sigma^2\right)$$

Capacity of Binary-Input AWGN Channel(Cont.)

- The **probability density function** of Y is a mixture of two Gaussian distributions (given $X = x$) with common variance

$$f_Y(y) = \frac{1}{2}\left\{\frac{\exp\left[-(y+1)^2/2\sigma^2\right]}{\sqrt{2\pi}\sigma} + \frac{\exp\left[-(y-1)^2/2\sigma^2\right]}{\sqrt{2\pi}\sigma}\right\}$$

- Then, the differential entropy of Y is

$$h(y) = -\int_{-\infty}^{\infty} f_Y(y)\log_2\left[f_Y(y)\right]dy$$

  - No closed form is available
- The mutual information is solely a function of the **noise variance** $\sigma^2 \Rightarrow I(X;Y) = M(\sigma^2)$
- For **error-free** transmission over the **AWGN channel**, the **code rate** r must be **smaller** than the channel capacity C (i.e., $I(X;Y)$)

$$r < M\left(\sigma^2\right)$$

Capacity of Binary-Input AWGN Channel(Cont.)

- A robust measure of the ratio E_b/N_0 is

$$\frac{E_b}{N_0} = \frac{PT_b}{N_0} = \frac{PT_s/r}{N_0} = \frac{P}{N_0 r/T_s} = \frac{P}{(N_0/T_s)r} = \frac{P}{2\sigma^2 r}$$

  - where P is the average transmitted power, T_b is the bit duration, $T_s = T_b r$ is the coded symbol duration, and $N_0/2$ is the two-sided power spectral density of the channel noise
- For the **maximum** code rate r,

$$r = M(\sigma^2) \Rightarrow \sigma^2 = M^{-1}(r)$$

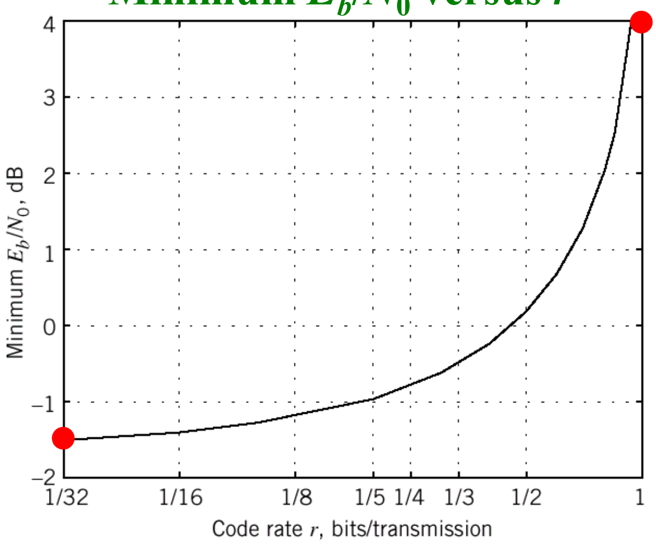
  - where $M^{-1}(r)$ is the **inverse** of the mutual information between the channel input and channel output
- By setting $P = 1$, the desired relation between E_b/N_0 and r is

$$\frac{E_b}{N_0} = \frac{P}{2\sigma^2 r} = \frac{1}{2rM^{-1}(r)}$$

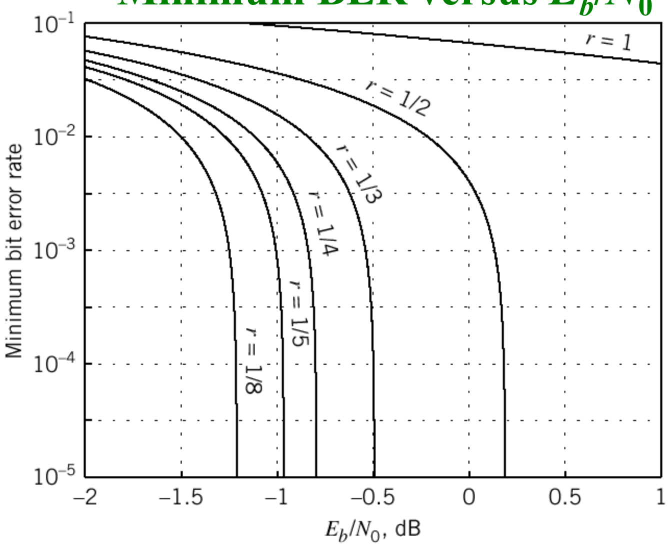
Capacity of Binary-Input AWGN Channel(Cont.)

- Using the **Monte Carlo method** to estimate the differential entropy $h(Y)$ and therefore $M^{-1}(r)$

Minimum E_b/N_0 versus r



Minimum BER versus E_b/N_0



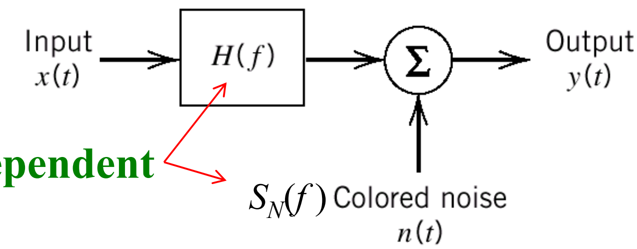
Capacity of Binary-Input AWGN Channel(Cont.)

- From previous results, we have the following conclusions:
  - For **uncoded** binary signaling (i.e., $r = 1$), an **infinite** E_b/N_0 is required for **error-free** communications
  - The **minimum** E_b/N_0, **decreases** with decreasing code rate r
    - For example, for $r = 1/2$, the minimum value of E_b/N_0 is slightly less than 0.2 dB
  - As r approaches **zero**, the **minimum** E_b/N_0 approaches the limiting value of **–1.6 dB** (**Shannon limit**)

Information Capacity of Colored Noisy Channel

Colored Noisy Channel

- Previous discussion is under the assumption of a **band-limited white noise channel**.
- Consider the more general case of a **non-white**, or **colored**, **noisy channel**.
  - $H(f)$: **transfer function** (frequency response) of the channel
  - $n(t)$: channel noise
    - A **stationary Gaussian process** of zero mean and power spectral density $S_N(f)$



Frequency-dependent

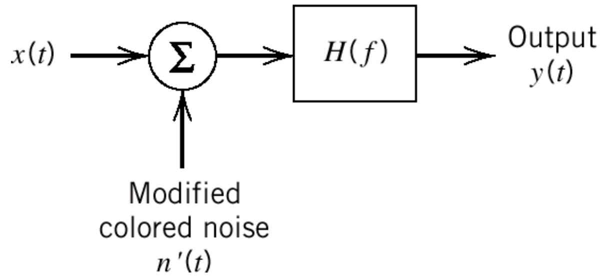
Colored Noisy Channel (Cont.)

- The goals of this study:
  - Find the **input ensemble**, described by the PSD $S_X(f)$ that
    - **Maximizes** the **mutual information** between the channel output $y(t)$ and the channel input $x(t)$
    - Subject to the **average power constraint** P of $x(t)$
  - Determine the **optimum information capacity** of the channel

Colored Noisy Channel (Cont.)

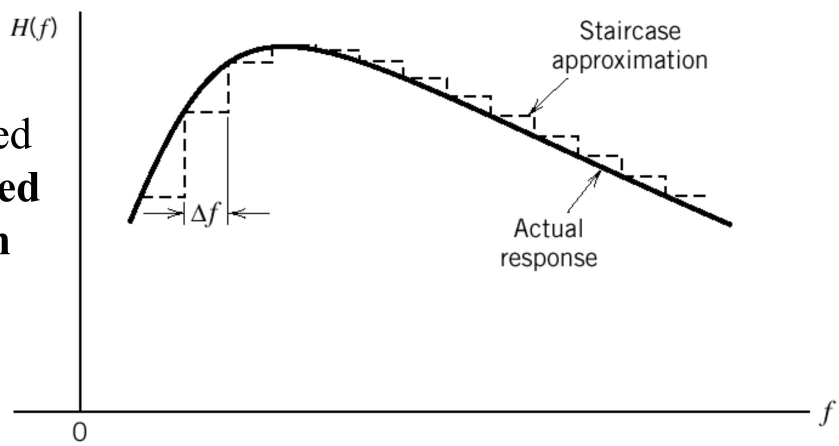
- Because the channel is linear, the channel model can be replaced with an equivalent model
  - From the viewpoint of the **spectral characteristics** of the **signal plus noise** measured at the **channel output**
  - The power spectral density of the noise $n'(t)$ is defined as

$$S_{N'}(f) = \frac{S_N(f)}{|H(f)|^2}$$



Colored Noisy Channel (Cont.)

- To simplify the analysis, the **continuous** $|H(f)|$ is approximated in the form of a **staircase**
  - The channel is divided into a large number of adjoining **frequency slots** $\Rightarrow$ Slot width: Δf (one-sided)
- The original model is replaced by the **parallel combination** of a finite number of **subchannels**, N
  - Each is corrupted by "**band-limited white Gaussian noise**"



Capacity of Colored Noisy Channel

- The k-th subchannel in the approximation is described by

$$y_k(t) = x_k(t) + n_k(t), \quad k = 1, 2, \cdots, N$$

- The average power of the signal component $x_k(t)$ is

$$P_k = S_X(f_k) \times \underline{2\Delta f}, \quad k = 1, 2, \cdots, N \longleftarrow \text{Positive \& Negative}$$

  - where $S_X(f_k)$ is the PSD of the input signal evaluated at $f = f_k$
- The variance of the noise component $n_k(t)$ is

$$\sigma_k^2 = \frac{S_N(f_k)}{|H(f_k)|^2} \times 2\Delta f, \quad k = 1, 2, \cdots, N$$

  - where $S_N(f_k)$ and $|H(f_k)|$ are the noise spectral density and the channel's magnitude response evaluated at $f = f_k$
- The **information capacity** of the k-th subchannel is

$$C_k = \Delta f \log_2\left(1 + P_k / \sigma_k^2\right), \quad k = 1, 2, \cdots, N$$

Capacity of Colored Noisy Channel (Cont.)

- All the N subchannels are **independent** of one another.
- The total capacity of the **overall channel** is approximately given by the summation

$$C = \sum_{k=1}^{N} C_k = \sum_{k=1}^{N} \Delta f \log_2\left(1 + P_k / \sigma_k^2\right)$$

- We want to **maximize** the overall information capacity C subject to the **total power constraint**

$$P = \sum_{k=1}^{N} P_k$$

- The method of **Lagrange multipliers** is used to solve a constrained optimization problem (The solving is omitted)
- To satisfy this optimizing solution, we have the requirement:

$$P_k + \sigma_k^2 = K \times 2\Delta f, \quad k = 1, 2, \cdots, N$$

  - where K is a **constant** chosen to satisfy the power constraint

Capacity of Colored Noisy Channel (Cont.)

- Inserting the defining values of P_k and σ_k, we get

$$S_X(f_k)\times 2\Delta f+\frac{S_N(f_k)}{|H(f_k)|^2}\times 2\Delta f=2K\Delta f \Rightarrow S_X(f_k)=K-\frac{S_N(f_k)}{|H(f_k)|^2}$$

- Let $\mathcal{F}_A$ denote the **frequency range** for which the constant K satisfies the condition

$$K\geq S_N(f_k)\Big/|H(f_k)|^2 \quad \text{to ensure that } S_X(f_k)\geq 0$$

- As the incremental frequency interval **approaches zero** and the number of subchannels N **goes to infinity**
  - The PSD of the input ensemble that achieves the **optimum information capacity** is a **nonnegative quantity** defined by

$$S_X(f)=\begin{cases} K-S_N(f)\Big/|H(f)|^2, & f\in\mathcal{F}_A \\ 0, & \text{otherwise}\end{cases}$$

Capacity of Colored Noisy Channel (Cont.)

- The average power of the channel input $x(t)$ is

$$P=\int_{f\in\mathcal{F}_A} K-S_N(f)\Big/|H(f)|^2\, df$$

  - The constant K is set to the value satisfying the constraint
- For the optimum information capacity, we obtain

$$C=\sum_{k=1}^{N}C_k=\sum_{k=1}^{N}\Delta f\log_2\left(1+P_k/\sigma_k^2\right)$$
$$=\sum_{k=1}^{N}\Delta f\log_2\left(K|H(f_k)|^2\Big/S_N(f_k)\right)$$

$$\boxed{\begin{aligned} P_k+\sigma_k^2&=K\times 2\Delta f\\ P_k&=S_X(f_k)\times 2\Delta f\\ \sigma_k^2&=\frac{S_N(f_k)}{|H(f_k)|^2}\times 2\Delta f\end{aligned}}$$

- When the incremental frequency interval approaches zero, we have the limiting form

$$C=\int_{-\infty}^{\infty}\log_2\left(K|H(f)|^2\Big/S_N(f)\right)df$$

Water-filling Interpretation of the Information Capacity Law

Water-filling Interpretation

- According to the **PSD of the input ensemble** that achieves the **optimum information capacity** and the **average power** of the channel input $x(t)$

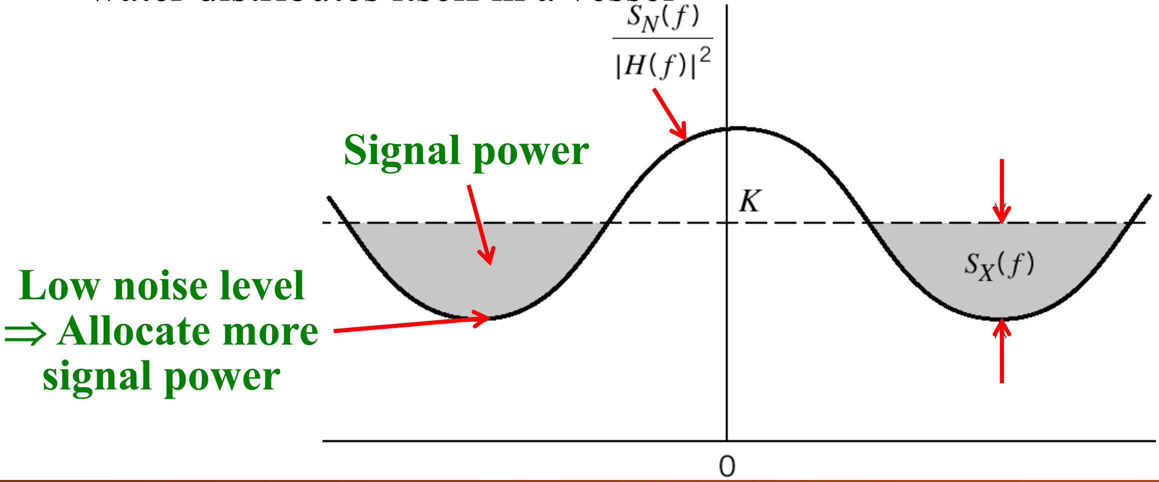
$$S_X(f) = \begin{cases} K - S_N(f) / |H(f)|^2, & f \in \mathscr{F}_A \\ 0, & \text{otherwise} \end{cases}$$

$$P = \int_{f \in \mathscr{F}_A} K - S_N(f) / |H(f)|^2 \, df$$

- We have the following observations:
  - The appropriate **input power spectral density** $S_X(f)$ is the bottom regions of the function $S_N(f)/|H(f)|^2$ that lie below the constant level K (which are shown shaded).
  - The **input power** P is defined by the **total area** of these shaded regions.

Water-filling Interpretation (Cont.)

- The shown spectral-domain picture is called the **water-filling (pouring) interpretation**, in the sense that
  - The process of **distributing** the **input power** across the function $S_N(f)/|H(f)|^2$ is identical to "The way in which water distributes itself in a vessel"



Water-filling Interpretation (Cont.)

- Consider the idealized case of a band-limited signal in **AWGN channel** of power spectral density $N(f) = N_0/2$
  - The transfer function $H(f)$: an **ideal band-pass filter**

$$H(f) = \begin{cases} 1, & 0 \le f_c - B/2 \le |f| \le f_c + B/2 \\ 0, & \text{otherwise} \end{cases}$$

  - f_c: the **midband frequency**; B: the **channel bandwidth**
- The **average input signal power** and the **optimum information capacity** become

$$P = 2B\left(K - N_0/2\right)$$
$$C = B\log_2\left(2K/N_0\right)$$

Homework

- **You must give detailed derivations or explanations, otherwise you get no points.**
- Communication Systems, Simon Haykin **(4th Ed.)**
- 9.2; 9.3;
- 9.5; 9.10;
- 9.12; 9.17;
- 9.22; 9.23;
- 9.29; 9.30;