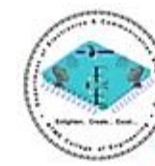




A T M E
College of Engineering



DIGITAL COMMUNICATION

BEC503

Credits: 04

Pradeep Kumar Y

Assistant Professor
Department of ECE



WELCOME TO

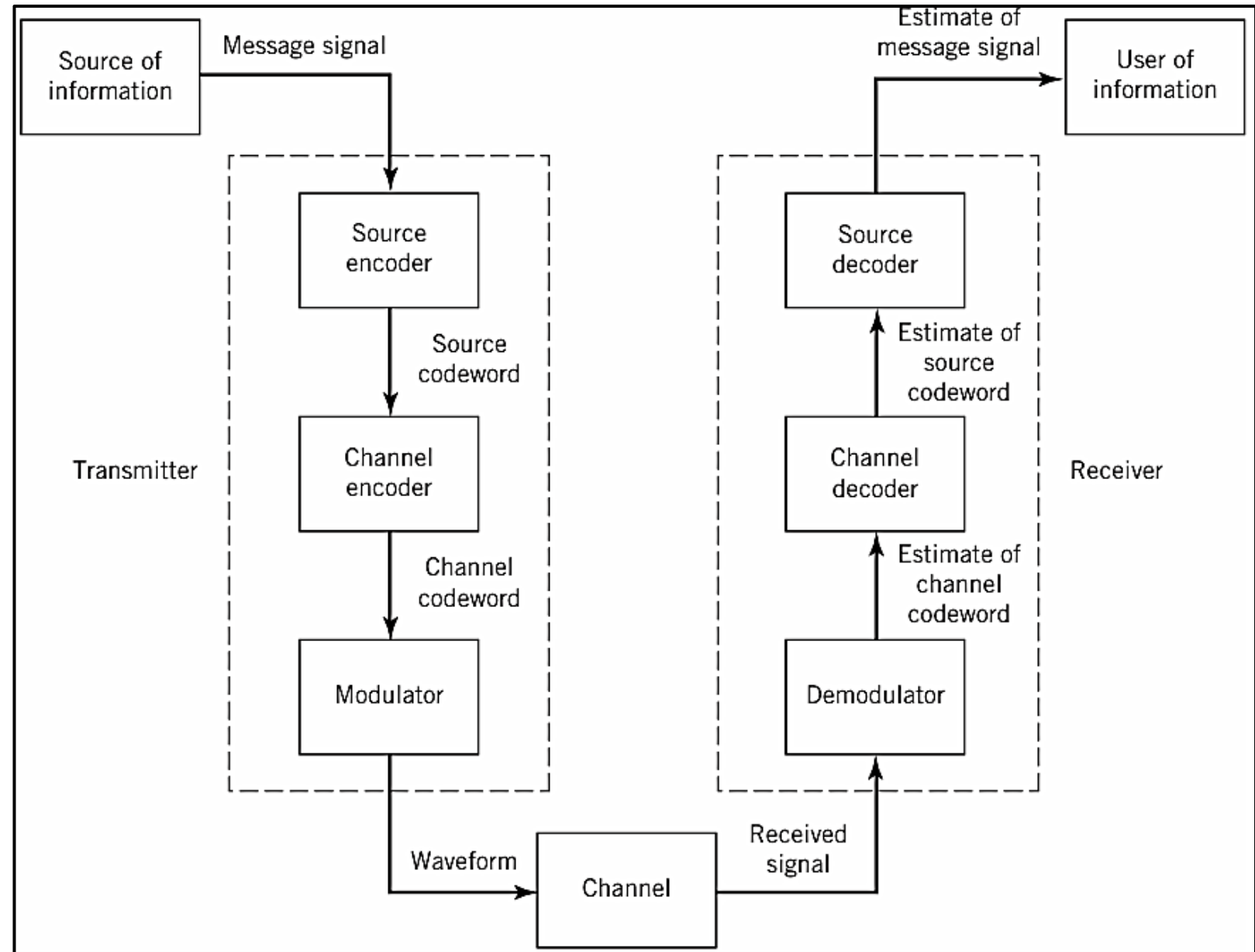
DIGITAL COMMUNICATION

Connecting Ideas. Sharing Information. Building Tomorrow.

TRANSMIT | SIGNAL | DATA | RECEIVE | COMMUNICATE

◀ Every Bit. Every Signal. Every Connection Matters. ▶

Block diagram of a digital communication system



Course Objectives:

This course will enable students to:

1. **Understand** the concept of **signal processing** of digital data and **signal conversion** to symbols at the transmitter and receiver.
2. **Compute** performance metrics and parameters for **symbol processing** and recovery in ideal and corrupted channel conditions.
3. **Understand** the principles of **spread spectrum communications**.
4. **Understand** the basic principles of **information theory** and various **source coding techniques**.
5. **Build** a comprehensive knowledge about various **Source and Channel Coding** techniques.
6. **Discuss** the different types of **errors and error detection and controlling codes** used in the communication channel.
7. **Understand** the concepts of **convolution codes** and analyze the code words using time domain and transform domain approach.

Course Outcomes

1. **Apply** the concept of **signal conversion to vectors** in communication transmission and reception.
2. **Perform** the **mathematical analysis** of digital communication systems for different modulation techniques.
3. **Apply** the **Source coding and Channel coding principles** for the discrete memoryless channels.
4. **Compute** the codewords for the **error correction and detection** of a digital data using Linear Block Code, Cyclic Codes and Convolution Codes.
5. **Design encoding and decoding** circuits for Linear Block Code, Cyclic Codes and Convolution Codes.

Module 3- Information Theory & Source Coding

- 1. Introduction to Information Theory**
- 2. Entropy & Information**
- 3. Source Coding Theorem**
- 4. Lossless Data Compression**
 1. Huffman Coding
 2. Shannon–Fano Coding
- 5. Discrete Memoryless Channels (DMC)**
- 6. Mutual Information**
- 7. Channel Capacity**
- 8. Channel Coding Theorem**
- 9. Information Capacity Law – Statement**

1. Introduction to Information Theory

- Mathematical framework for **information and communication systems**
- Deals with **mathematical modelling and analysis**
- Uses **probability theory**
- Gives a precise meaning to **information**
- Forms the basis for:
 - Source Coding
 - Channel Coding
 - Data Compression
 - Reliable Communication

1. Information Theory – Introduction

Two Key Questions

- Minimum possible compression? → **Entropy**
- Maximum reliable transmission rate? → **Channel Capacity**

Fundamental Result

$H(S) < C \Rightarrow$ Ideally error-free communication

$H(S) \rightarrow$ Source Entropy

$C \rightarrow$ Channel Capacity

1. Information Theory – Introduction

Two Key Questions

- Minimum possible compression? → **Entropy**
- Maximum reliable transmission rate? → **Channel Capacity**

Fundamental Result

$H(S) < C \Rightarrow$ Ideally error-free communication

$H(S) \rightarrow$ Source Entropy

$C \rightarrow$ Channel Capacity

1. Information Theory – Introduction

Entropy and Capacity

- **Entropy** represents the average information content or the uncertainty associated with a source.
- **Channel Capacity** represents the intrinsic ability of a communication channel to convey information reliably and is related to the noise characteristics of the channel.

2. Entropy and Information

- The entropy of a discrete random variable, representing the output of a source of information, is a measure of the average information content per source symbol.
- The entropy is represented by **$H(S)$** .

2. Entropy and Information

Information, Uncertainty and Surprise

- In a digital communication system, the source produces information in the form of symbols. Since different symbols may occur with different probabilities, there is uncertainty about which symbol will be produced next.
- Information is closely related to uncertainty and surprise.
- If an event is certain, there is no surprise and hence no new information.
- If an event is unlikely, its occurrence gives us more information.
- Therefore, the amount of information is related to the inverse of the probability of occurrence.

2. Entropy and Information

Discrete Source Representation

- Consider a discrete source represented by a discrete random variable S .
- Its alphabet is: $S = \{s_0, s_1, s_2, \dots, s_{K-1}\}$
where K is the number of symbols in the source alphabet.
- The probability of occurrence of symbol s_k is:
$$P(S = s_k) = p_k, \quad k = 0, 1, \dots, K-1$$
- The probabilities satisfy the normalization condition:
$$\sum_{k=0}^{K-1} p_k = 1, p_k \geq 0$$
- The source symbols emitted during successive signaling intervals are assumed to be statistically independent.

2. Entropy and Information- **Self-Information**

- The amount of information gained when the event $S = s_k$ occurs with probability p_k is called the self-information of the event.

- It is defined as $I(s_k) = \log\left(\frac{1}{p_k}\right)$

- For binary signaling, the logarithm is normally taken to the base 2:

$$I(s_k) = \log_2\left(\frac{1}{p_k}\right) = -\log_2 p_k$$

- The unit of information is then the **bit**

2. Entropy and Information- Important Properties of Self-Information

❑ **Property 1: Certain event** If $p_k = 1$ Then $I(s_k) = 0$

This means that a completely certain event provides **no new information**.

❑ **Property 2: Information is non-negative**

$$\text{For } 0 < p_k \leq 1$$
$$I(s_k) \geq 0$$

Thus, an event provides some information or no information, but not negative information.

2. Entropy and Information- Important Properties of Self-Information

□ **Property 3: Less probable event gives more information:**

If $p_k < p_i$ then $I(s_k) > I(s_i)$

Therefore, an unlikely event carries more information than a highly probable event.

□ **Property 4: Additive property**

For statistically independent events s_k and s_l $I(s_k, s_l) = I(s_k) + I(s_l)$

This additive property results from the logarithmic definition of self-information

2. Entropy and Information- **Unit of Information – Bit**

When the logarithm is taken to base 2,

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

the resulting unit is called a **bit**.

Example: For two equally likely events,

$$p_k = \frac{1}{2}$$

therefore,

$$I(s_k) = \log_2 2 = 1$$

Hence: **One bit is the amount of information obtained when one of two equally likely events occurs.**

2. Entropy and Information- Entropy

- The self-information depends on which symbol is actually produced. Therefore, self-information itself can be considered a **discrete random variable**.
- The average information produced by the source is obtained by taking the expectation of self-information.

2. Entropy and Information- Entropy

The **entropy** of a discrete random variable S is defined as

$$H(S) = E[I(s_k)]$$

Therefore,

$$H(S) = \sum_{k=0}^{K-1} p_k I(s_k)$$

Substituting

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

2. Entropy and Information- **Entropy**

we obtain

$$H(S) = \sum_{k=0}^{K-1} p_k \log_2 \left(\frac{1}{p_k} \right)$$

or equivalently,

$$H(S) = - \sum_{k=0}^{K-1} p_k \log_2 p_k$$

2. Entropy and Information- Entropy

- ❑ Entropy is the average information content per source symbol produced by a source.
- ❑ Entropy depends only on the **probabilities of the source symbols**, not on the actual names or values of the symbols.
- ❑ Source having K possible symbols,

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

2. Entropy and Information- **Properties of Entropy**

1. Minimum Entropy $H(S) = 0$

when one symbol has probability 1 and all other symbols have probability 0.

For example, $p_0 = 1, p_1 = p_2 = \dots = 0$

There is no uncertainty because the source output is completely predictable.

2. Maximum Entropy $H(S) = \log_2 K$

when all K symbols are **equiprobable**: $p_k = \frac{1}{K}, k = 0, 1, \dots, K - 1$

Maximum entropy therefore corresponds to **maximum uncertainty**.

2. Entropy and Information- Properties of Entropy

Condition	Entropy
Completely certain source	$H(S) = 0$
Maximum uncertainty	$H(S) = \log_2 K$
All symbols equally probable	Maximum entropy

2. Entropy and Information Example1: Entropy of a Bernoulli Random Variable

A **Bernoulli random variable** has two possible symbols, 0 and 1.

Let $P(S = 0) = p_0$

and $P(S = 1) = p_1 = 1 - p_0$

The entropy is $H(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$

Since $p_1 = 1 - p_0$,

$$H(S) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0)$$

The corresponding **entropy function** is written as

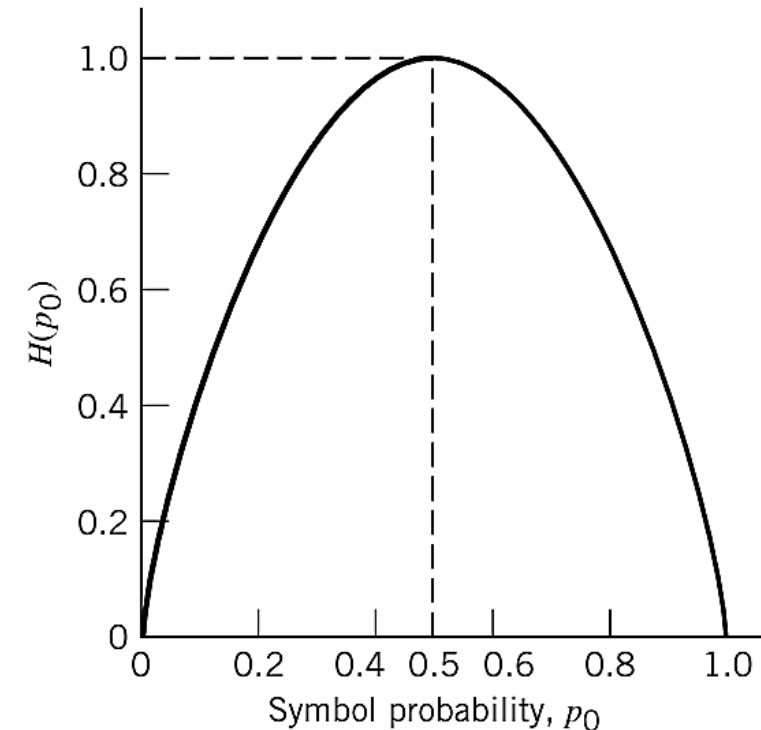
$$H(p_0) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0)$$

2. Entropy and Information

Example1: Entropy of a Bernoulli Random Variable

Important Observations

Condition	Entropy
$p_0 = 0$	$H(S) = 0$
$p_0 = 1$	$H(S) = 0$
$p_0 = p_1 = \frac{1}{2}$	$H_{\max} = 1$



The entropy function is **symmetric about** $p_0 = \frac{1}{2}$

2. Entropy and Information

Extension of a Discrete Memoryless Source

A source is called **memoryless** when the symbol emitted at any time is statistically independent of the symbols emitted at previous and future signaling intervals.

Instead of considering individual source symbols, we can consider a block of n successive source symbols.

If the original source alphabet contains K symbols, then the extended source has K^n possible blocks.

Since the source symbols are statistically independent, the probability of a particular block is equal to the **product of the probabilities** of the symbols forming that block.

The entropy of the extended source is $H(S^n) = nH(S)$

where:

- S = original source
- S^n = n -th order extension of the source
- $H(S)$ = entropy per source symbol
- $H(S^n)$ = entropy of the n -symbol block.

3. SOURCE-CODING THEOREM

Source Encoding:

- It is the process of representing data generated by a discrete source in a more efficient form, usually as a sequence of binary digits (0's and 1's).
- The device that performs the representation is called a source encoder.
- When encoding symbols from a source, some symbols may occur more frequently than others.
- By assigning shorter codewords to more frequent symbols and longer codewords to less frequent symbols, we can achieve a more efficient encoding scheme.
- This method is known as variable-length coding.

3. SOURCE-CODING THEOREM

Key Requirements for Source Codes:

- The codewords produced by the encoder are in Binary form.
- The source code is uniquely decodable, so that the original source sequence can be reconstructed perfectly from the encoded binary sequence.

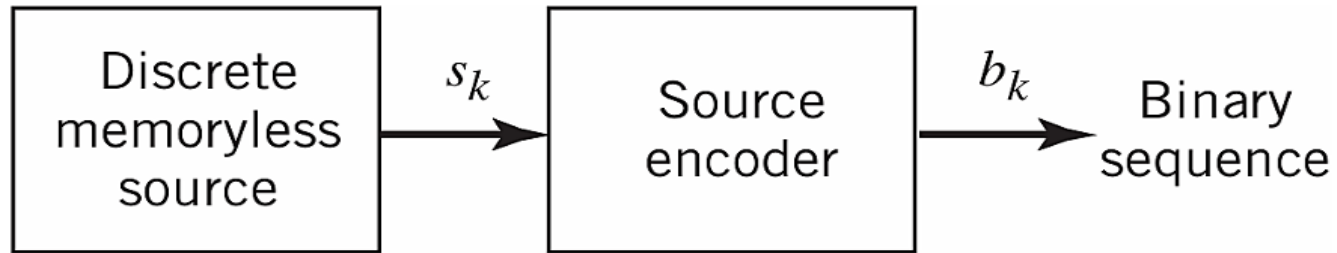
3. SOURCE-CODING THEOREM

Key Requirements for Source Codes:

- The codewords produced by the encoder are in Binary form.
- The source code is uniquely decodable, so that the original source sequence can be reconstructed perfectly from the encoded binary sequence.

3. SOURCE-CODING THEOREM

Consider a discrete memoryless source whose output s_k is converted by the source encoder into a sequence of 0s and 1s, denoted by b_k .



- Let the binary codeword assigned to symbol s_k by the encoder have length l_k , measured in bits. The average codeword length ($\bar{L}$) of the source encoder is:

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

- The parameter $\bar{L}$ represents the average number of bits per source symbol used in the source encoding process.
- The coding efficiency (η) is: $\eta = \frac{H(S)}{\bar{L}}$

4. LOSSLESS DATA COMPRESSION ALGORITHMS

- Lossless data compression is a process of removing redundancy from a signal without any loss of information.
- This process, typically applied to digital signals, results in an efficient representation that minimizes the average number of bits per symbol.

4. LOSSLESS DATA COMPRESSION ALGORITHMS

Prefix Coding

- A prefix code is defined as a code in which no codeword is the prefix of any other codeword.
- It is a type of source code used for lossless data compression.
- These codes are uniquely decodable.
- Prefix codes are distinguished from other uniquely decodable codes by the fact that the end of a codeword is always recognizable.
- Hence, the decoding of a prefix can be accomplished as soon as the binary sequence representing a source symbol is fully received.
- For this reason, prefix codes are also referred to as instantaneous codes.

4. LOSSLESS DATA COMPRESSION ALGORITHMS

Example of Prefix Coding.

Symbol source	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

- Code-I is not a prefix code because the bit 0, the codeword for S_0 , is a prefix of 00, the codeword for S_2 . Likewise, the bit 1, the codeword for S_1 , is a prefix of 11, the codeword for S_3 .
- Similarly, we may show that Code-III is not a prefix code, but Code-II is a prefix code because no codeword is a prefix of another.

3. LOSSLESS DATA COMPRESSION ALGORITHMS

Example of Prefix Coding.

Symbol source	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

- Code-I is not a prefix code because the bit 0, the codeword for S_0 , is a prefix of 00, the codeword for S_2 . Likewise, the bit 1, the codeword for S_1 , is a prefix of 11, the codeword for S_3 .
- Similarly, we may show that Code-III is not a prefix code, but Code-II is a prefix code because no codeword is a prefix of another.

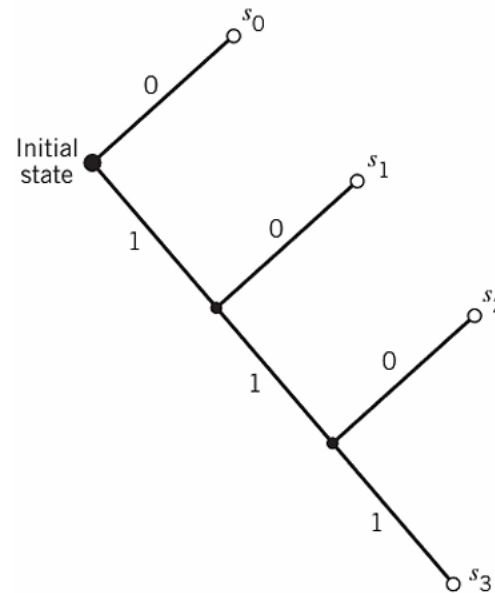
4. LOSSLESS DATA COMPRESSION ALGORITHMS

Decoding of a Prefix Code:

To decode a sequence of codewords from a prefix code, a decision tree is often used to represent the code.

Decision Tree for Code-II of Table-1:

- The decoding process involves traversing this tree based on the bits received in the encoded sequence.
- The decoder starts at the initial state and moves down the tree based on the bits it encounters, until it reaches a terminal state, which corresponds to one of the source symbols.
- After decoding a symbol, the decoder resets to the initial state for the next codeword.



Example: Consider the following encoded sequence:

1011111000 ...

This sequence can be decoded using Code-II of Table-1 as follows:

$\underbrace{10}_{s_1}$
 $\underbrace{111}_{s_3}$
 $\underbrace{110}_{s_2}$
 $\underbrace{0}_{s_0}$
 $\underbrace{0}_{s_0}$
 ...

4. HUFFMAN CODING

Huffman coding is an optimal prefix coding algorithm for lossless data compression. It minimizes average code length by assigning shorter binary codes to more frequent symbols and longer codes to less frequent ones.

Encoding Algorithm Steps:

1. **Sort & Split:** List source symbols in order of decreasing probability. Assign 0 and 1 to the two lowest-probability symbols (Splitting stage).
2. **Combine:** Add the two lowest probabilities together to form a new combined symbol and reinsert it into the sorted list.
3. **Repeat:** Continue combining the two lowest probabilities until only two symbols remain (assigned 0 and 1).
4. **Trace Codewords:** Work backward from the final stage, tracing the assigned 0s and 1s to construct each symbol's codeword.

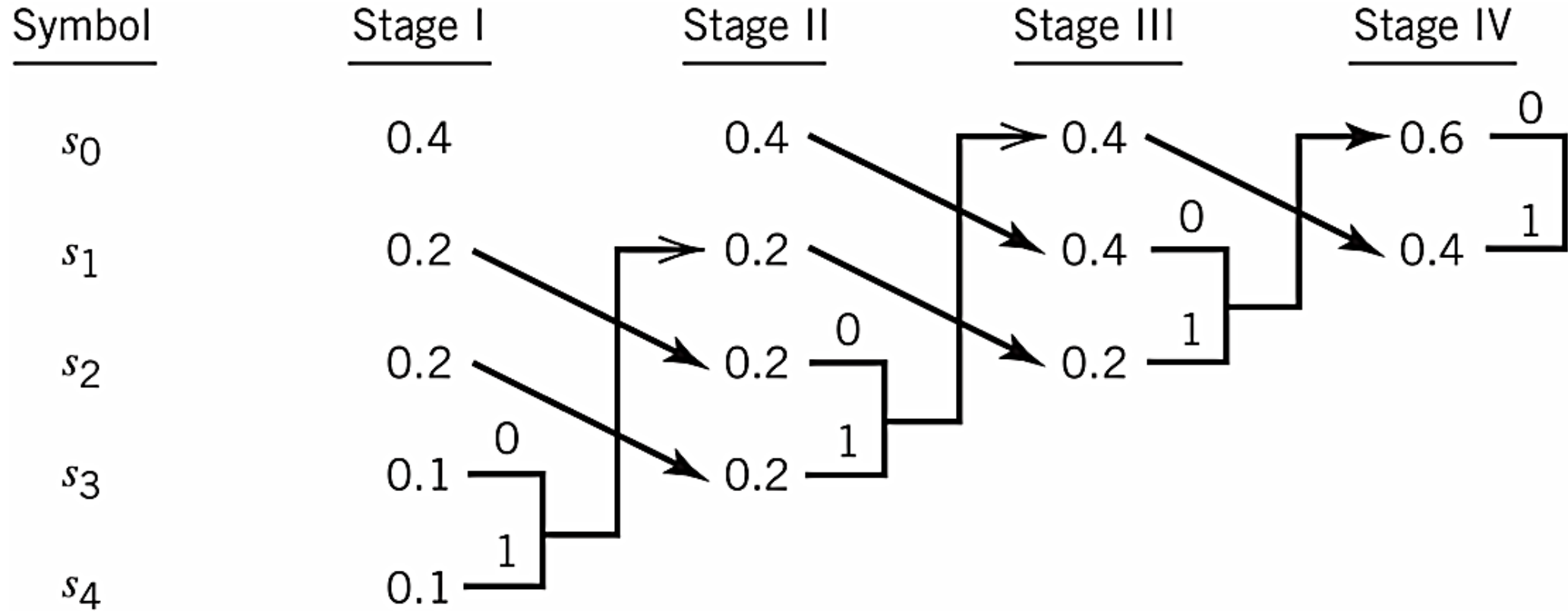
4. HUFFMAN CODING

Given a discrete memoryless source with five symbols:

Symbol	s0	s1	s2	s3	s4
Probability	0.4	0.2	0.2	0.1	0.1

Compute the Huffman code (placing combined symbols as high as possible) and verify $\bar{L} \geq H(S)$.

4. HUFFMAN CODING



4. HUFFMAN CODING

Resulting Source Code:

Symbol	Probability	Codeword	Length (lk)
s_0	0.4	00	2
s_1	0.2	10	2
s_2	0.2	11	2
s_3	0.1	010	3
s_4	0.1	011	3

4. HUFFMAN CODING

Average Codeword Length ($\bar{L}$):

$$\bar{L} = \sum_{k=0}^4 P_k l_k = 0.4(2) + 0.2(2) + 0.2(2) + 0.1(3) + 0.1(3) = \mathbf{2.2 \text{ binary symbols}}$$

Entropy $H(S)$:

$$\begin{aligned} H(S) &= 0.4 \log_2 \left(\frac{1}{0.4} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right) \\ &= \mathbf{2.121 \text{ bits}} \end{aligned}$$

Conclusion:

$$\bar{L} \geq H(S) \Rightarrow 2.2 \geq 2.121 \quad \mathbf{(Verified)}$$

4. LEMPEL-ZIV CODING

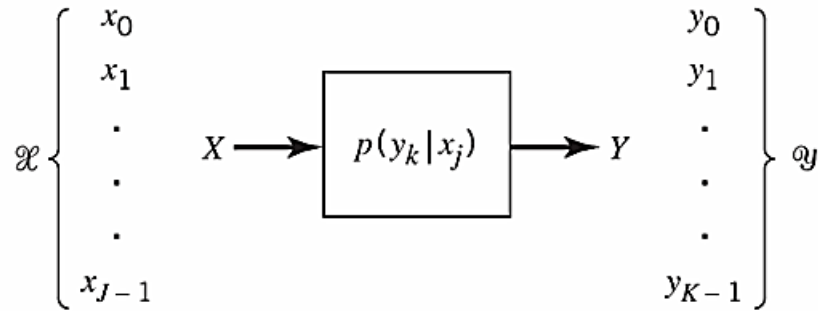
Lempel-Ziv coding is an adaptive, dictionary-based algorithm used for lossless data compression.

The source data stream is parsed into segments that are the shortest subsequences not encountered previously.

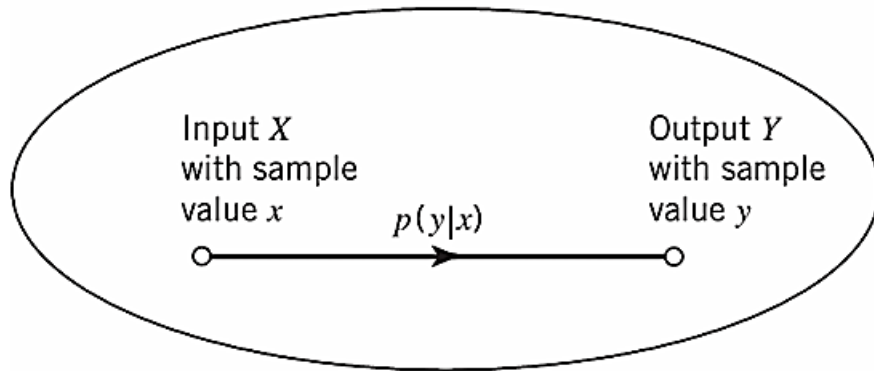
Parsing Example: Consider the binary sequence: 000101110010100101...

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

5. DISCRETE MEMORYLESS CHANNELS



(a)



$$0 \leq p(y_k|x_j) \leq 1 \quad \text{for all } j \text{ and } k$$

$$\sum_K p(y_k|x_j) = 1 \quad \text{for fixed } j$$

5. DISCRETE MEMORYLESS CHANNELS

- A discrete memoryless channel is a statistical model with an input $\mathbf{X}$ and an output $\mathbf{Y}$ that is a noisy version of $\mathbf{X}$; Both $\mathbf{X}$ & $\mathbf{Y}$ are random variables.
- Every unit of time, the channel accepts an input symbol $\mathbf{X}$ selected from an alphabet $\mathcal{X}$ and, in response, it emits an output symbol $\mathbf{Y}$ from an alphabet $\mathcal{Y}$.

$$\mathcal{X} = \{x_0, x_1, \dots, x_{J-1}\}$$
$$\mathcal{Y} = \{y_0, y_1, \dots, y_{K-1}\}$$

5. DISCRETE MEMORYLESS CHANNELS

CHANNEL MATRIX & PROBABILITIES

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

- The J -by- K matrix $\mathbf{P}$ is called the channel matrix, or stochastic matrix.
- Note that each row of the channel matrix $\mathbf{P}$ corresponds to a fixed channel input, whereas each column of the matrix corresponds to a fixed channel output.
- The probabilities of the input symbols is $p(x_j)$ for $j = 0, 1, \dots, J-1$.
- The joint probability distribution of the random variables $\mathbf{X}$ and $\mathbf{Y}$ is given by:

$$p(x_j, y_k) = p(y_k|x_j)p(x_j)$$
- The marginal probability distribution of the output random variable $\mathbf{Y}$ is obtained by averaging out the dependence of $p(x_j, y_k)$ on x_j , obtaining:

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

$$p(y_k) = \sum_{j=0}^{J-1} p(y_k|x_j)p(x_j) \quad \text{for } k = 0, 1, \dots, K-1$$

6. MUTUAL INFORMATION

- **Definition of $I(X; Y)$:** The mutual information $I(X; Y)$ is a measure of the uncertainty about the channel input, which is resolved by observing the output.

$$I(X; Y) = H(X) - H(X|Y) \rightarrow (2)$$

where,

- $H(X)$ – entropy of the channel input X .
 - $H(X|Y)$ – conditional entropy.
- **Definition of $I(Y; X)$:** The mutual information $I(Y; X)$ is a measure of the uncertainty about the channel output that is resolved by sending the channel input.

$$I(Y; X) = H(Y) - H(Y|X) \rightarrow (3)$$

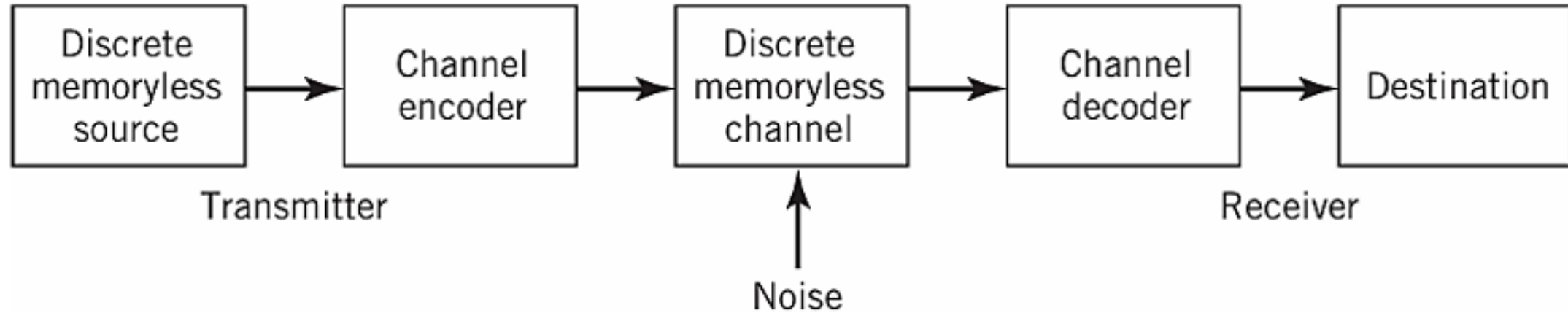
7. CHANNEL CAPACITY

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

$$C = \max_{\{p(x_j)\}} I(X; Y) \text{ bits per channel use.} \rightarrow (14)$$

- **Constraints:** Equation (14) is subjected to two input probabilistic constraints:
 - $p(x_j) \geq 0$ for all j
 - $\sum_{j=0}^{J-1} p(x_j) = 1$

8. CHANNEL CODING THEOREM



- Direct Theorem
- Converse theorem
- Information Capacity law



ATME
College of Engineering

