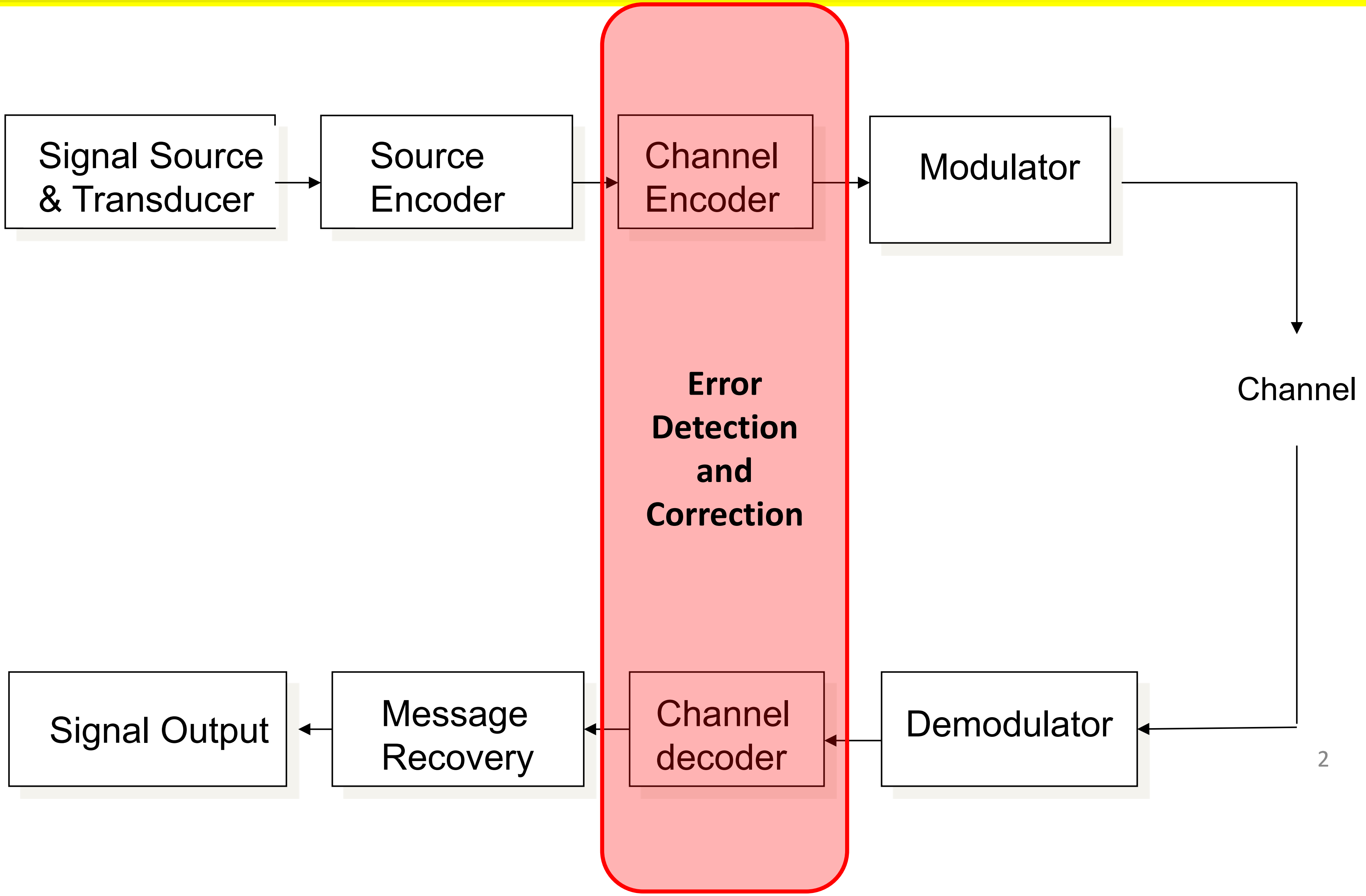
ERROR DETECTION & CORRECTION

EEEN 464 – DIGITAL COMMUNICATION

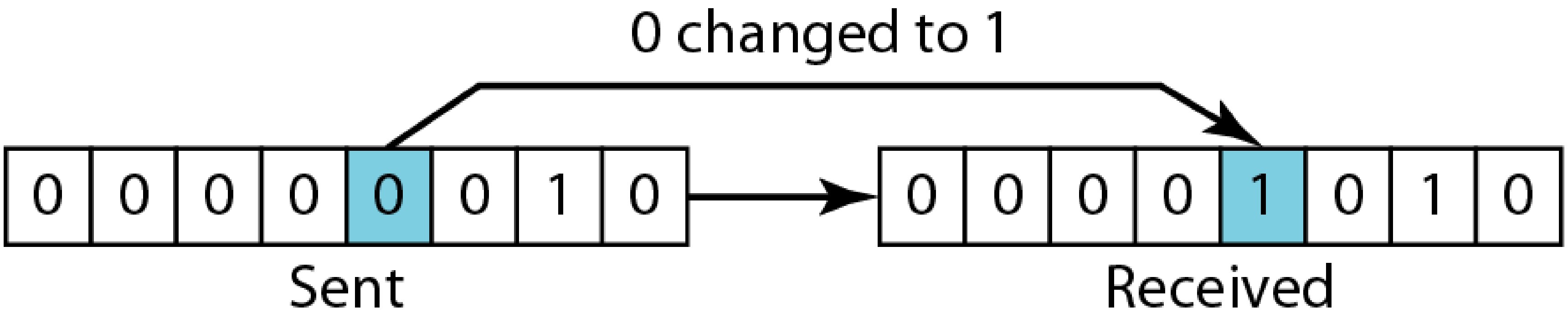
Friday, February 27, 2026

ERROR DETECTION & CORRECTION IN A DIGITAL COMMUNICATION SYSTEM

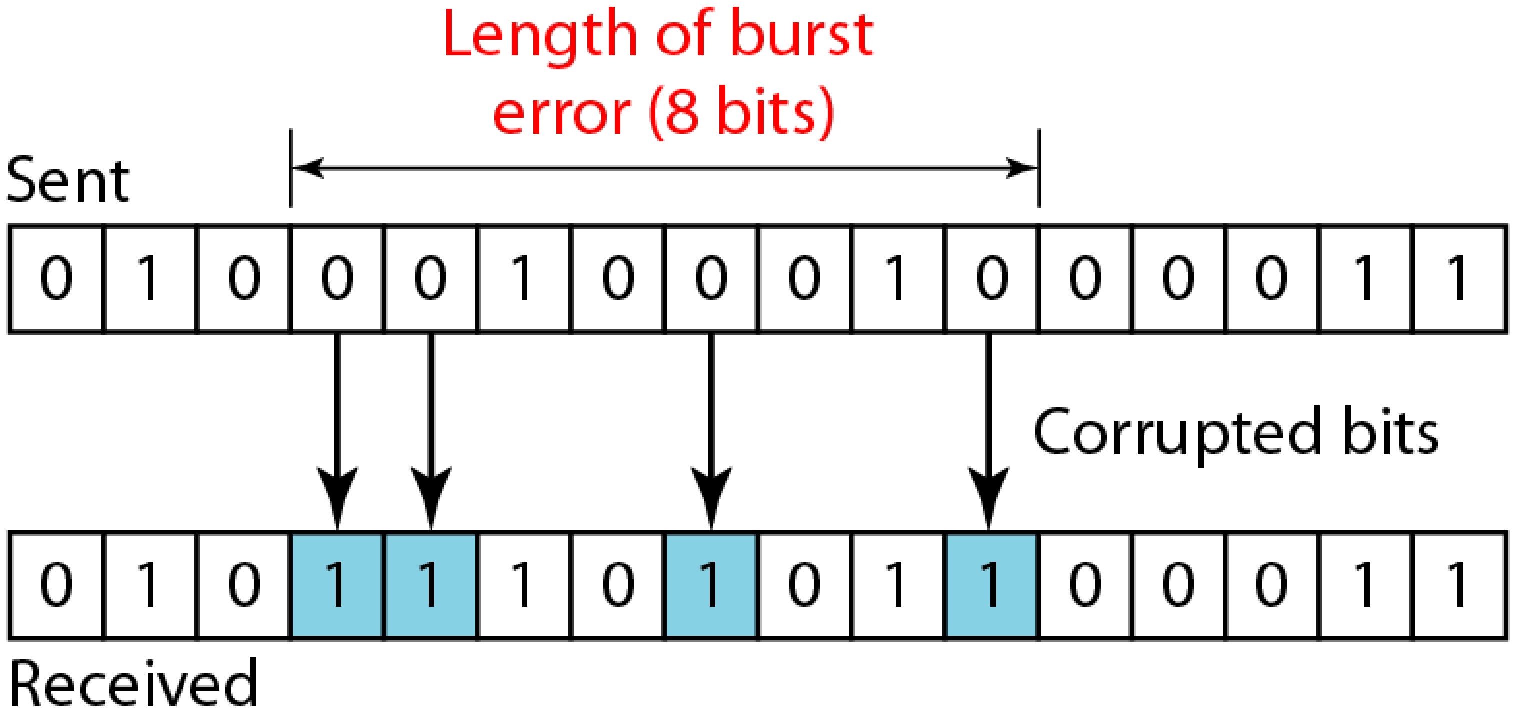


CLASSIFICATION OF ERRORS/01

(a) Single-bit error



(b) Multiple bit /burst Error

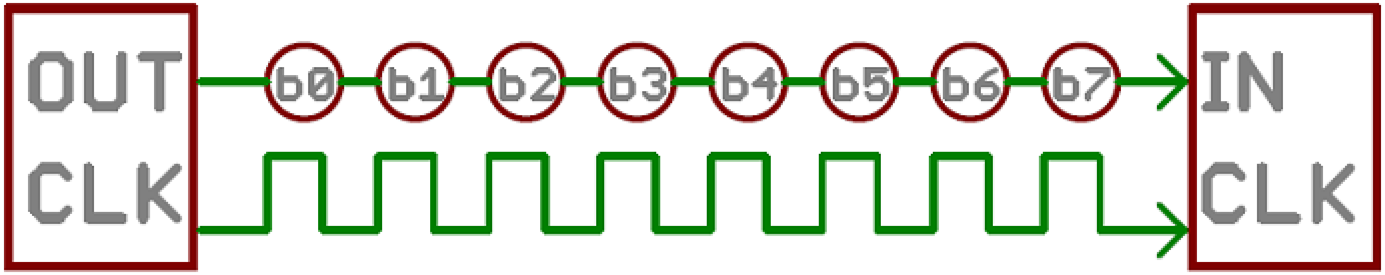


SINGLE-BIT ERRORS IN SERIAL COMMUNICATION

- Single-bit errors are the least likely type of error in serial data transmission.

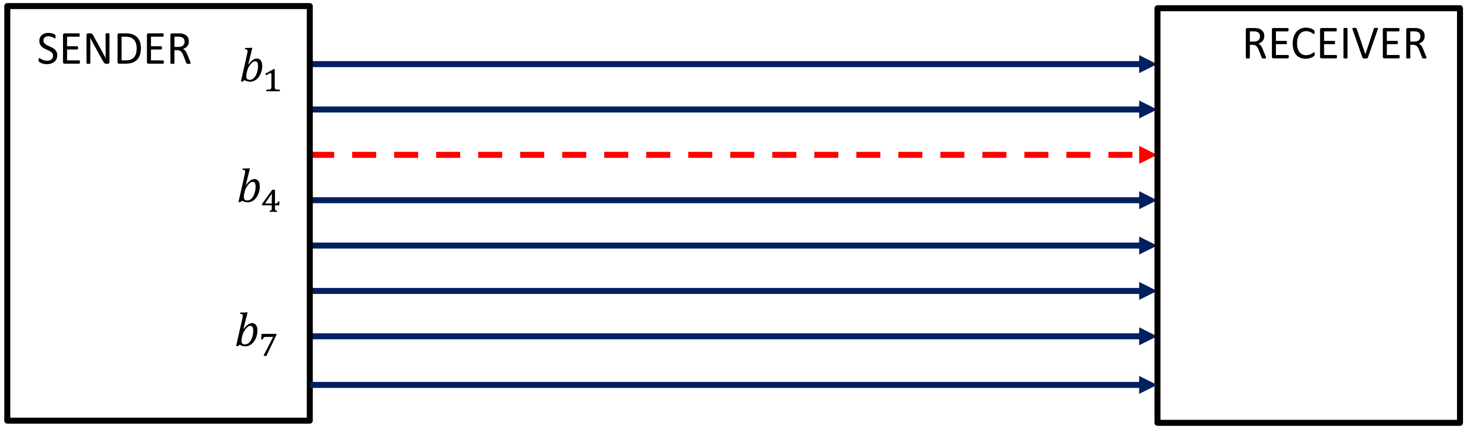
Illustration

- Suppose a sender sends data at 1 Mbps.
- This means that the bit duration is 1 micro sec.
- For a single bit error to occur the noise must have a duration of 1 micro sec which is very rare.



SINGLE-BIT ERRORS IN PARALLEL TRANSMISSION

- Single bit errors can occur if we are sending data using parallel transmission.
- **Illustration**
- Suppose 8 wires are used to send all 8 bits of 1 byte at the same time
- One of the wires is noisy, one bit can be corrupted in each byte.



BURST ERRORS

1. **Burst error** occurs when two or more bits in the data unit have changed.
2. The burst error does not necessarily mean that errors occur in consecutive bits.
3. **The length of a burst is measured from the first corrupted bit to the last corrupted bit.**
4. **Some bits in between may not be corrupted.**

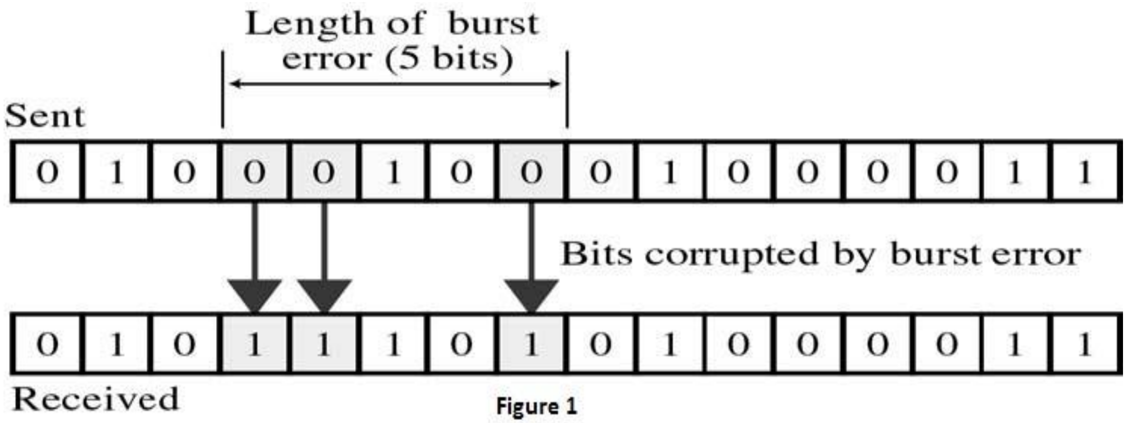
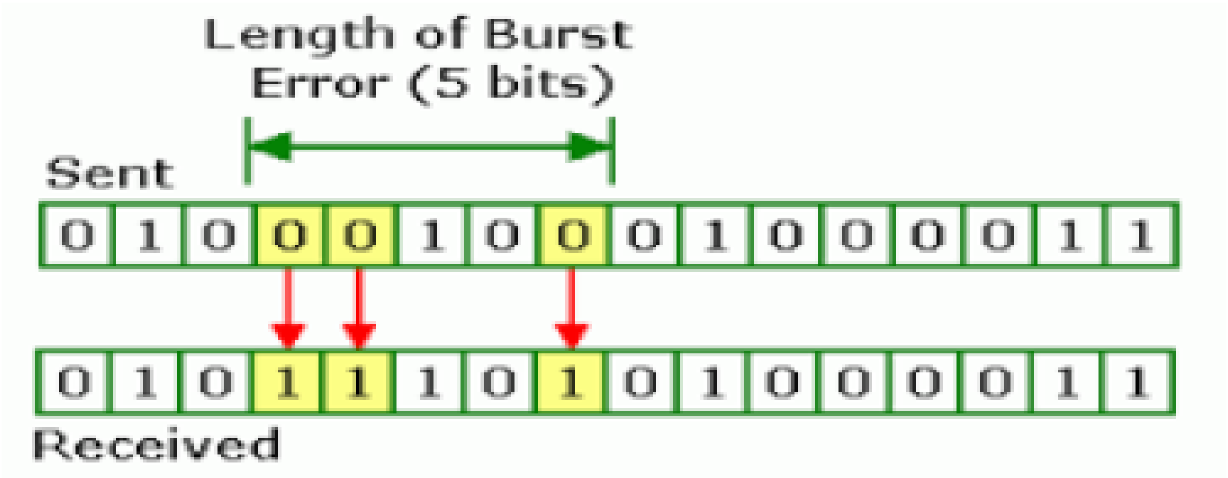


Figure 1

BURST ERRORS IN SERIAL COMMUNICATION

1. Burst error is most likely to occur in serial transmission.
2. The duration of noise is normally longer than the duration of one bit.
3. Which means that when noise affects data, it affects a set of bits.
4. The number of bits affected depends on the data rate and duration of noise.

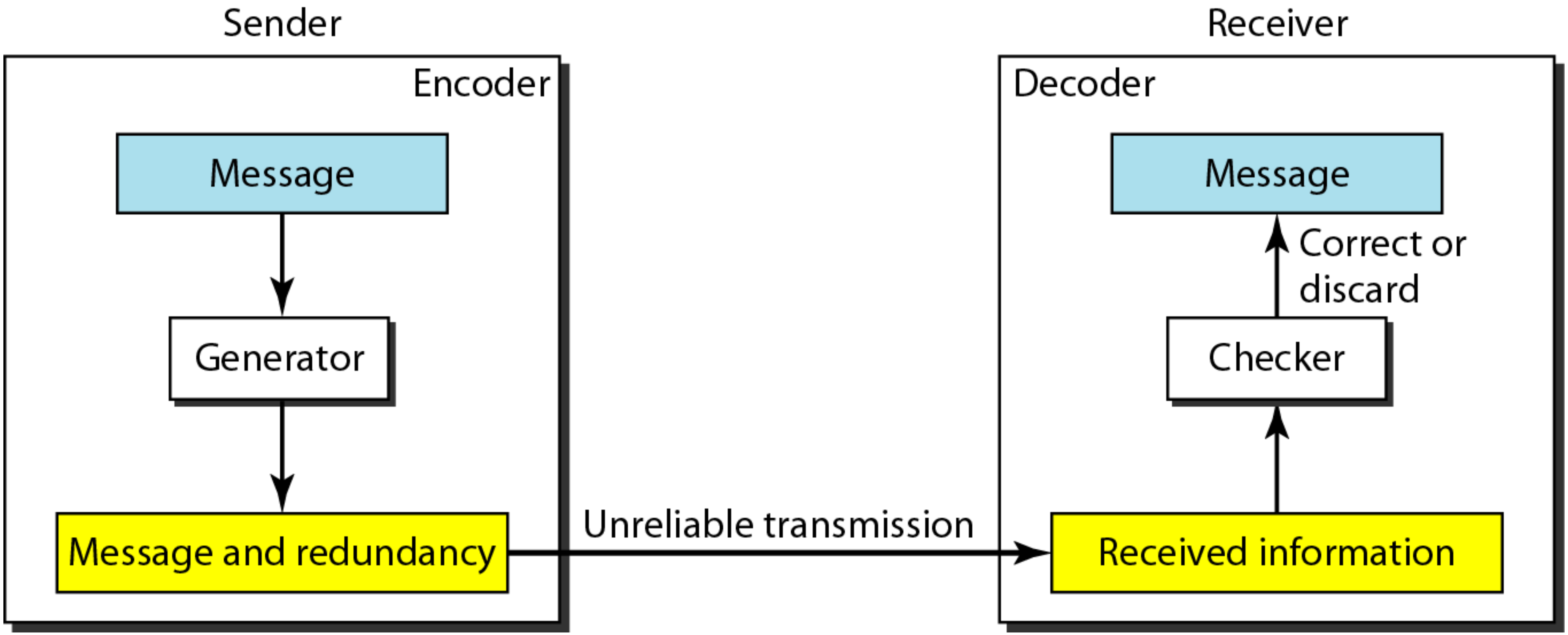


Sent: 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Received: 0 1 0 1 1 1 0 1 0 1 0 0 0 0 1 1

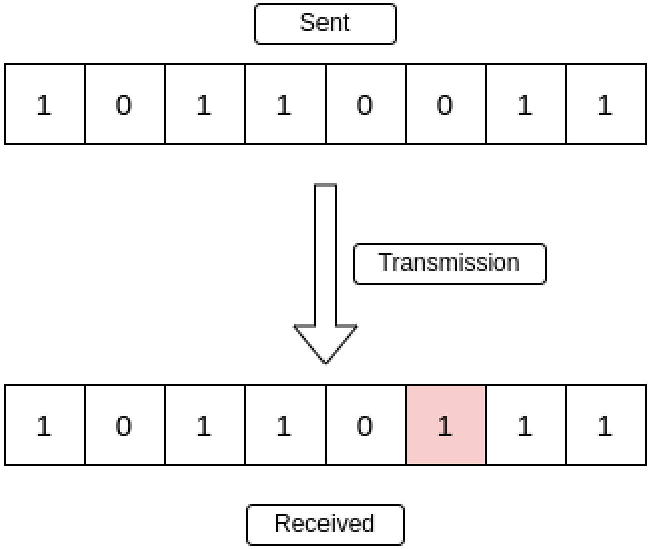
ERROR DETECTION & CORRECTION

To detect or correct errors, we need to send the message data together with redundant data.



ERROR DETECTION AND CORRECTION

1. **Error detection** is the process of identifying whether a transmitted data message contains errors caused by noise or interference during transmission.
2. **Error correction** refers to techniques used to identify and correct errors in data transmission or storage without requiring retransmission of the data.
3. The most common error detection methods are:
   - **(a)** Parity Checking
   - **(b)** Checksum error detection
   - **(c)** Cyclic Error Check (CRC)



Single-Bit Error

PARITY CHECKING

1. **Parity checking** methods introduce an additional bit, called a parity bit which is added to each data word.
2. **A parity bit** is added to ensure that the number of bits with the value one in a set of bits is even or odd.
   a) **Even parity** is when the number of ‘1’ bits in the word is even
   b) **Odd parity** is when the number of ‘1’ bits is odd.

EVEN PARITY

1. The number of ‘1’ bits in the data word is 4, i.e Even, therefore Parity bit is 0.

PARITY BIT	DATA WORD
0	1001011

01001011

2. The number of ‘1’ bits in the data word is 3, i.e Odd, therefore Parity bit is 1.

PARITY BIT	DATA WORD
1	0010011

10010011

ODD PARITY

1. The number of ‘1’ bits in the data word is 4, i.e Even, therefore Parity bit is 1.

PARITY BIT	DATA WORD
1	1001011

11001011

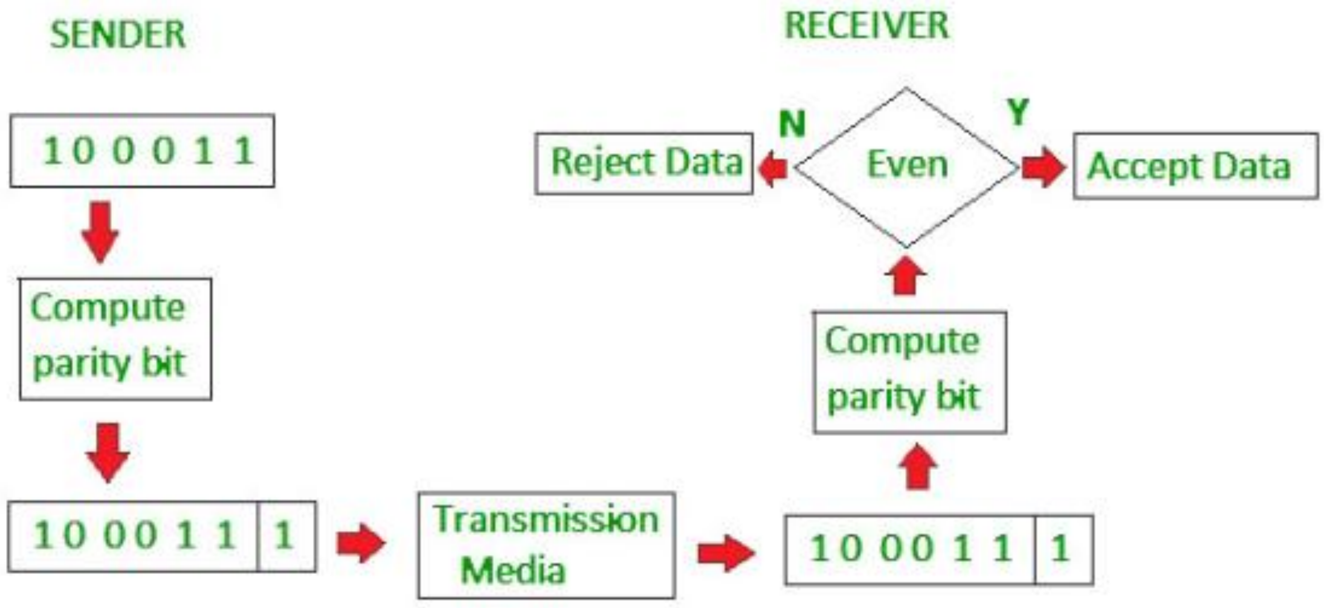
2. The number of ‘1’ bits in the data word is 3, i.e Odd, therefore Parity bit is 0.

PARITY BIT	DATA WORD
0	1000110

01000110

ERROR DETECTION

- **A parity check at the receiver** detects an error if the parity of the received word is different from the expected parity.



FAILURE OF PARITY CHECKING METHOD – EVEN PARITY

1. If the number of errors introduced in a transmitted word is two or multiple of two, then the parity does not change and the receiver will fail to detect the presence of errors.
2. If the number of errors introduced in a transmitted word is one or any odd number, then the parity changes and the receiver can detect the presence of errors.

EXAMPLE 1: PARITY

1. Write the ASCII code of the word TALE using even parity.

SOLUTION (1)

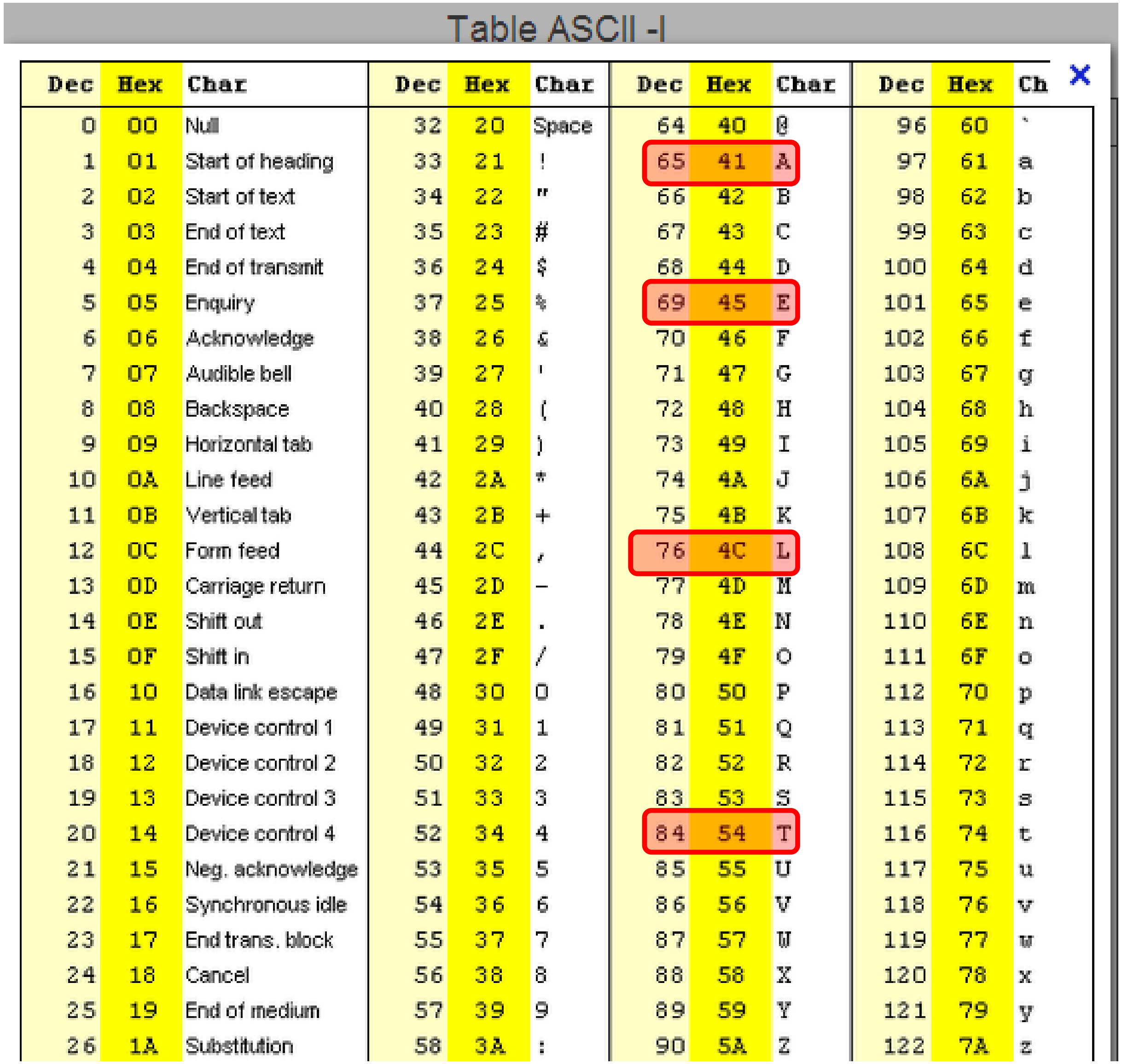


Table ASCII -I

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Ch
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	−	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z

SOLUTION (2)

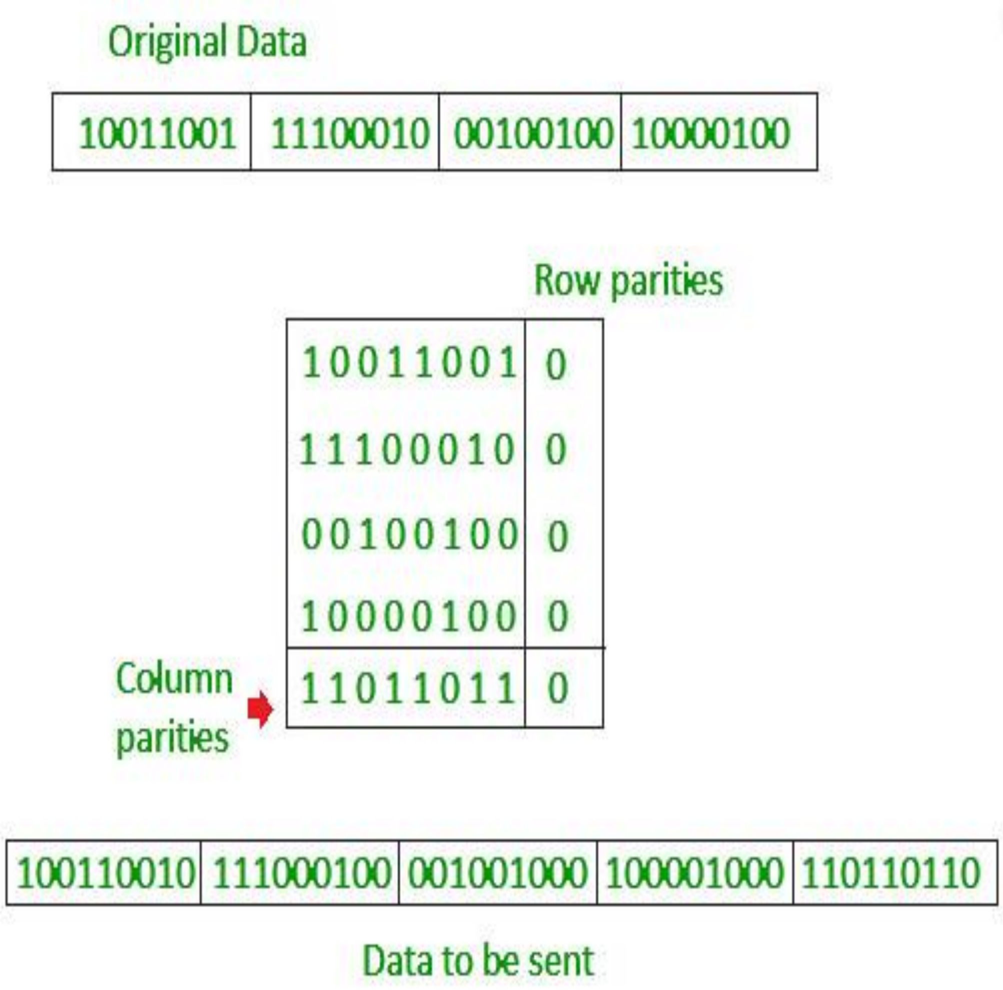
D		P	7	6	5	4	3	2	1
84	T	1	1	0	1	0	1	0	0
65	A	0	1	0	0	0	0	0	1
76	L	1	1	0	0	1	1	0	0
69	E	1	1	0	0	0	1	0	1

ADVANTAGES OF PARITY ERROR DETECTION

1. **Implementation:** Simple Parity Check is easy to implement in both hardware and software.
2. **Minimal Extra Data:** Only one additional bit (the parity bit) is added per data unit (e.g., per byte).
3. **Fast Error Detection:** The process of calculating and checking the parity bit is quick, which allows for rapid error detection without significant delay in data processing or communication.
4. **Single-Bit Error Detection:** It can effectively detect single-bit errors within a data unit, providing a basic level of error detection for relatively low-error environments.

TWO-DIMENSIONAL PARITY CHECK

1. **In two-dimensional Parity check, bits are calculated for each row**, which is equivalent to a simple parity check bit.
2. **Parity check bits are also calculated for all columns**, then both are sent along with the data.
3. **At the receiving end**, these are compared with the parity bits calculated on the received data.



Original Data

10011001	11100010	00100100	10000100

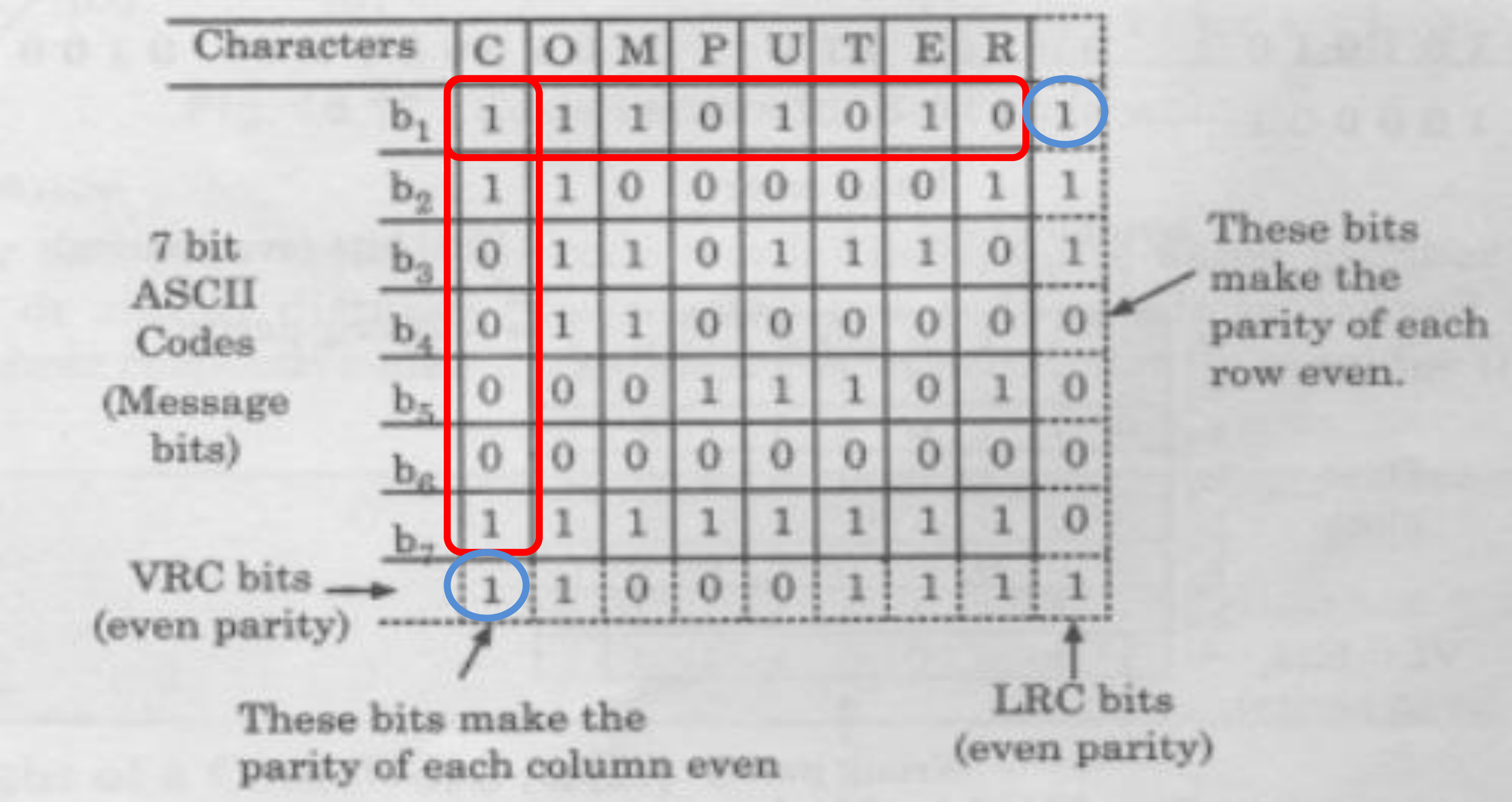
Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

100110010	111000100	001001000	100001000	110110110

Data to be sent

TWO-DIMENSIONAL PARITY CHECK EXAMPLE



Characters	C	O	M	P	U	T	E	R	
b_1	1	1	1	0	1	0	1	0	1
b_2	1	1	0	0	0	0	0	1	1
b_3	0	1	1	0	1	1	1	0	1
b_4	0	1	1	0	0	0	0	0	0
b_5	0	0	0	1	1	1	0	1	0
b_6	0	0	0	0	0	0	0	0	0
b_7	1	1	1	1	1	1	1	1	0
VRC bits (even parity)	1	1	0	0	0	1	1	1	1

7 bit ASCII Codes (Message bits)

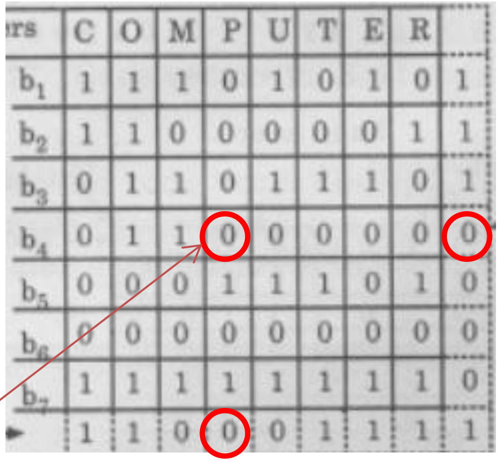
These bits make the parity of each row even.

These bits make the parity of each column even

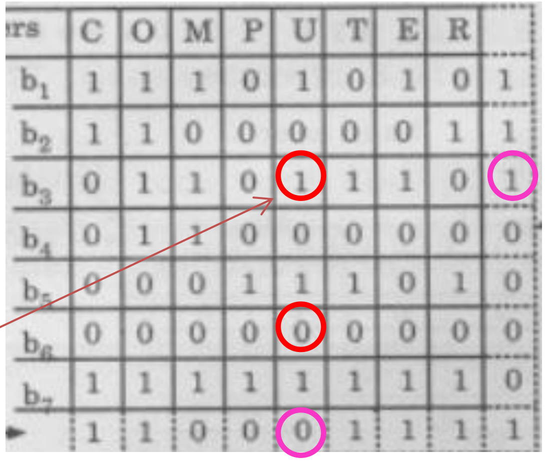
LRC bits (even parity)

IDENTIFICATION OF ERRORS IN LRC & VRC

1. A single error in any row will result in an incorrect LRC.
2. Similarly, a single error in any column will result in an incorrect VRC.
3. A bit at the intersection of the incorrect VRC and LRC can then be identified as the incorrect bit.
4. Unfortunately, multiple errors in rows and columns can only be detected but not corrected.



rs	C	O	M	P	U	T	E	R	
b_1	1	1	1	0	1	0	1	0	1
b_2	1	1	0	0	0	0	0	1	1
b_3	0	1	1	0	1	1	1	0	1
b_4	0	1	1	0	0	0	0	0	0
b_5	0	0	0	1	1	1	0	1	0
b_6	0	0	0	0	0	0	0	0	0
b_7	1	1	1	1	1	1	1	1	0
►	1	1	0	0	0	1	1	1	1



rs	C	O	M	P	U	T	E	R	
b_1	1	1	1	0	1	0	1	0	1
b_2	1	1	0	0	0	0	0	1	1
b_3	0	1	1	0	1	1	1	0	1
b_4	0	1	1	0	0	0	0	0	0
b_5	0	0	0	1	1	1	0	1	0
b_6	0	0	0	0	0	0	0	0	0
b_7	1	1	1	1	1	1	1	1	0
►	1	1	0	0	0	1	1	1	1

LOW-DENSITY PARITY CHECK (LDPC)

1. **Low-Density Parity Check (LDPC)** is a type of error correction technique characterized by a parity-check matrix with very few "1"s, meaning most of the matrix is filled with "0"s, which allows for efficient decoding algorithms and makes it particularly effective in noisy communication channels; essentially, the "low density" refers to the sparsity of the "1"s within the matrix.
2. **LDPC** is widely used in modern communication systems including 5G mobile networks, satellite communications (DVB-S2), Ethernet standards (10GBASE-T), and storage systems.

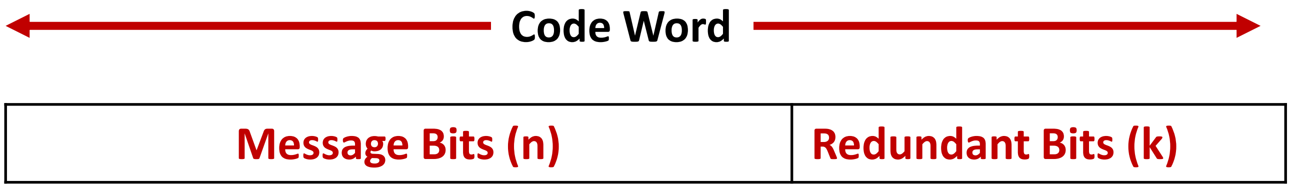
ERROR CORRECTING CODES: TERMINOLOGY

EEEN 464 – DIGITAL COMMUNICATION

Friday, February 27, 2026

CODEWORD

Code Word refers to an n bit block of bits containing message bits, parity or redundant bits.



Code Word

Message Bits (n)

Redundant Bits (k)

DEFINITIONS RELATING TO ERROR CODES /2

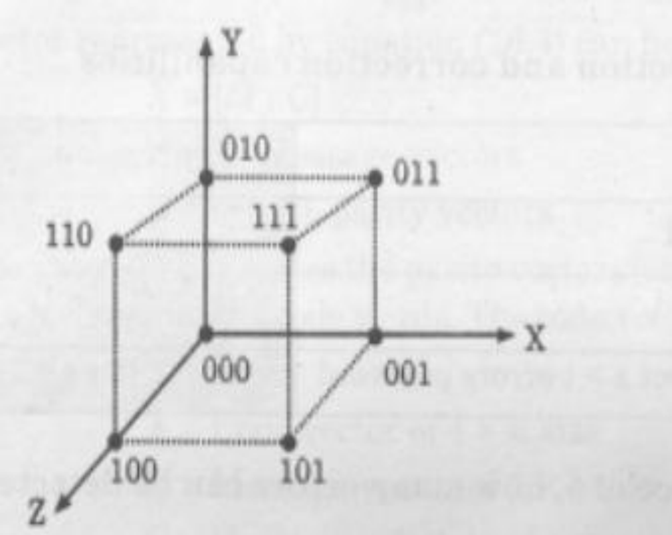
Code rate is the ratio of the number of message bits to the number of bits in the code word.

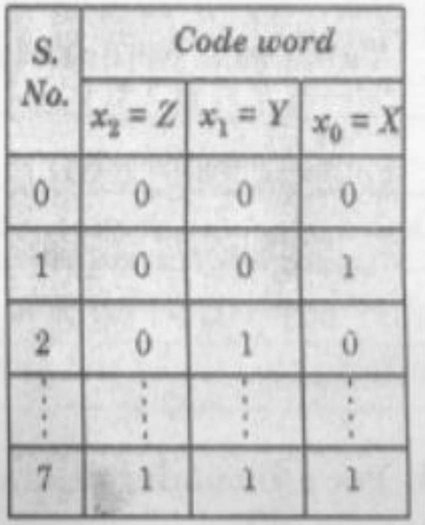
Message Bits (n)	Redundant Bits (k)

$$Code\ Rate = \frac{Number\ of\ message\ bits}{Total\ number\ of\ Bits} = \frac{n}{n+k}$$

DEFINITIONS RELATING TO ERROR CODES /3

Code vector refers to the visualization of an n-bit word in m-dimensional space.

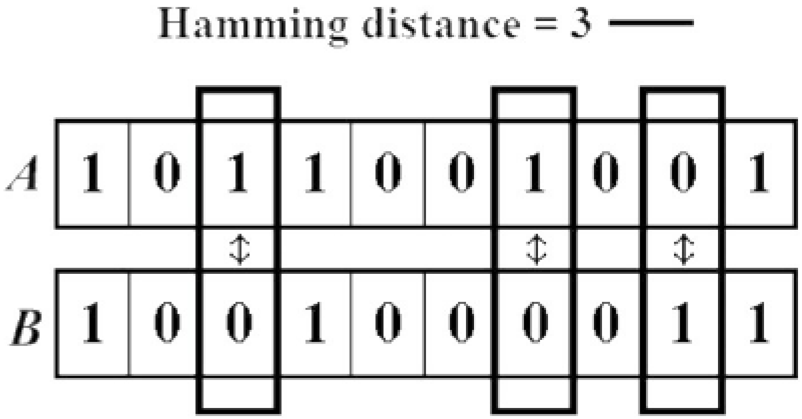




S. No.	Code word		
	$x_2 = Z$	$x_1 = Y$	$x_0 = X$
0	0	0	0
1	0	0	1
2	0	1	0
⋮	⋮	⋮	⋮
7	1	1	1

HAMMING DISTANCE

1. **Hamming distance** is a metric for comparing two binary data strings. While comparing two binary strings of equal length, Hamming distance is the number of bit positions in which the two bits are different.
2. If two code vectors have the same number of elements, Hamming Distance is defined as the number of locations in which the respective elements differ.



Hamming distance = 3

A	1	0	1	1	0	0	1	0	0	1
			↕				↕		↕	
B	1	0	0	1	0	0	0	0	1	1

EXAMPLE

What is the Hamming distance between 1111000 and 1000011?

The Hamming distance between "11111000" and "1000011" is 5.

1	1	1	1	0	0	0
1	0	0	0	0	1	1

HAMMING WEIGHT

Hamming Weight of a vector code is the distance between the code word and an all zero-code vectors of the length.

OR

Hamming weight is the number of non-zero symbols in a string or code word.

ERROR CONTROL TECHNIQUES

EEEN 464 – DIGITAL COMMUNICATION

Friday, February 27, 2026

TYPES OF ERROR CONTROL

There are basically two types of error control:

(a) Automatic Request for Retransmission (ARQ): The receiver requests for retransmission of complete or part of the message. This requires a feedback channel.

(b) Forward Error Correction (FEC): The sender adds redundant data to the original information, allowing the receiver to detect and correct errors in the received data without needing to request retransmission.

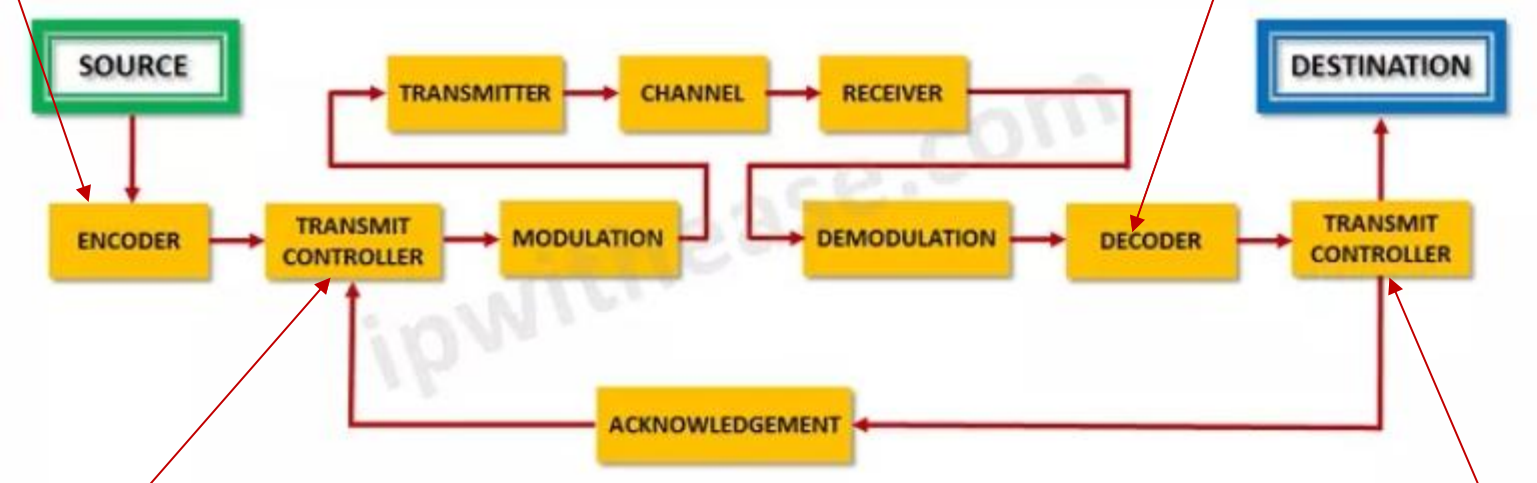
AUTOMATIC REPEAT REQUEST (ARQ)

1. **Automatic Repeat reQuest (ARQ)** uses acknowledgements and timeouts to achieve reliable data transmission over an unreliable channel.
2. If the sender does not receive an acknowledgment before the timeout, it usually re-transmits the frame/packet until it receives an acknowledgment or exceeds a predefined number of re-transmissions.

WORKING PRINCIPLE OF ARQ SYSTEM

Encoder splits the message into Code Words

The receiver decodes the code words and checks for errors

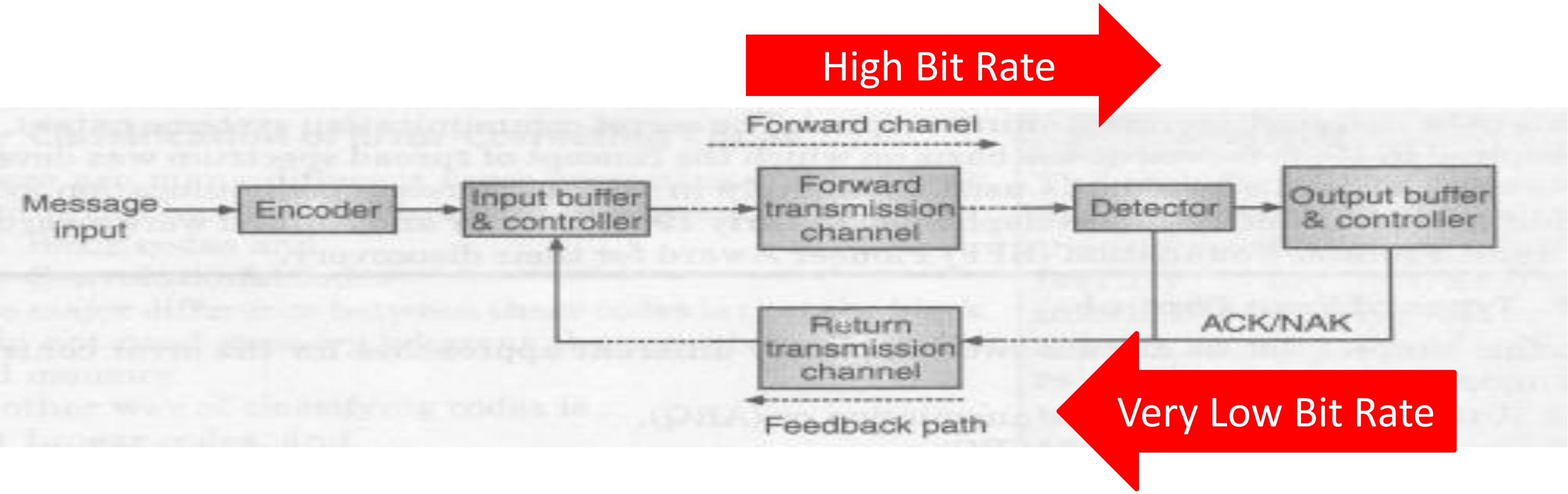


Each transmitted Code Word is stored temporarily

Controller sends positive ACKnowledgement(ACK) or Negative Acknowledgement (NAK) to the sender.

PROBABILITY OF ERROR ON FEEFBACK PATH

1. **The data rate for the feedback transmission is low** compared with forward transmission since it transmits only ACK or NAK.
2. **The probability of error in the feedback is therefore very small** and is therefore negligible.



High Bit Rate

Very Low Bit Rate

Forward chanel
Message input
Encoder
Input buffer & controller
Forward transmission channel
Detector
Output buffer & controller
ACK/NAK
Return transmission channel
Feedback path

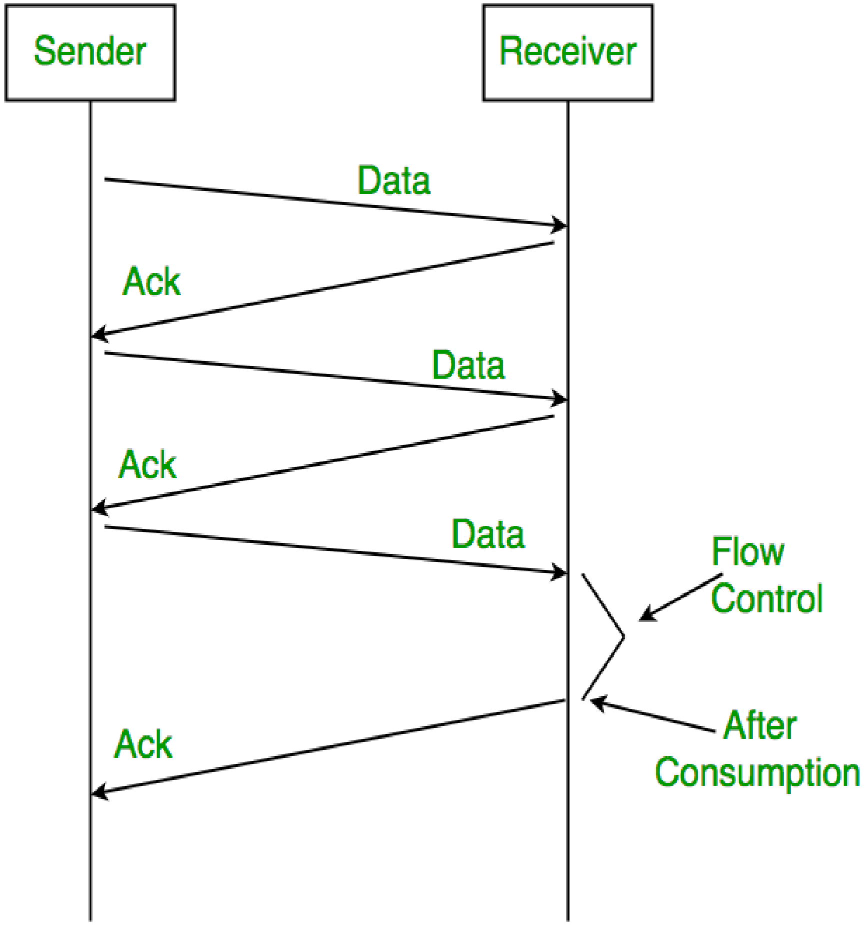
TYPES OF ARQ SYTEMS

There are THREE types of ARQ Systems:

(a) **Stop-and-Wait ARQ**

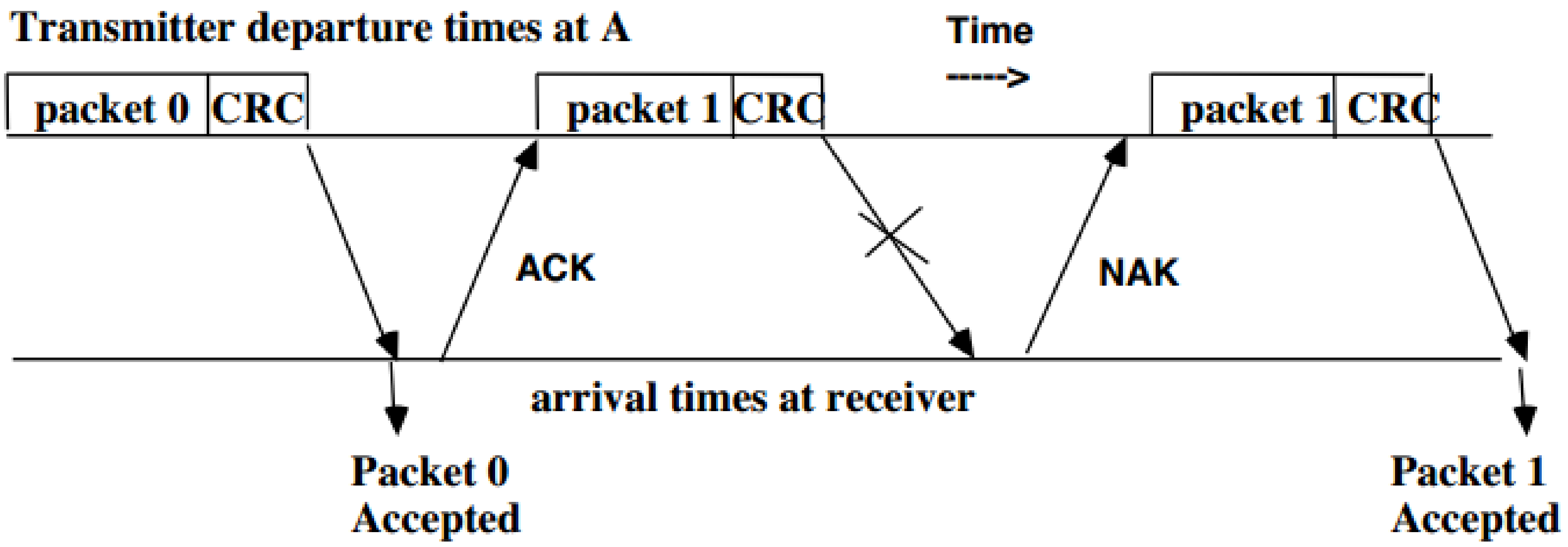
(b) **Go Back N ARQ**

(c) **Selective ARQ**



STOP-AND-WAIT ARQ (1)

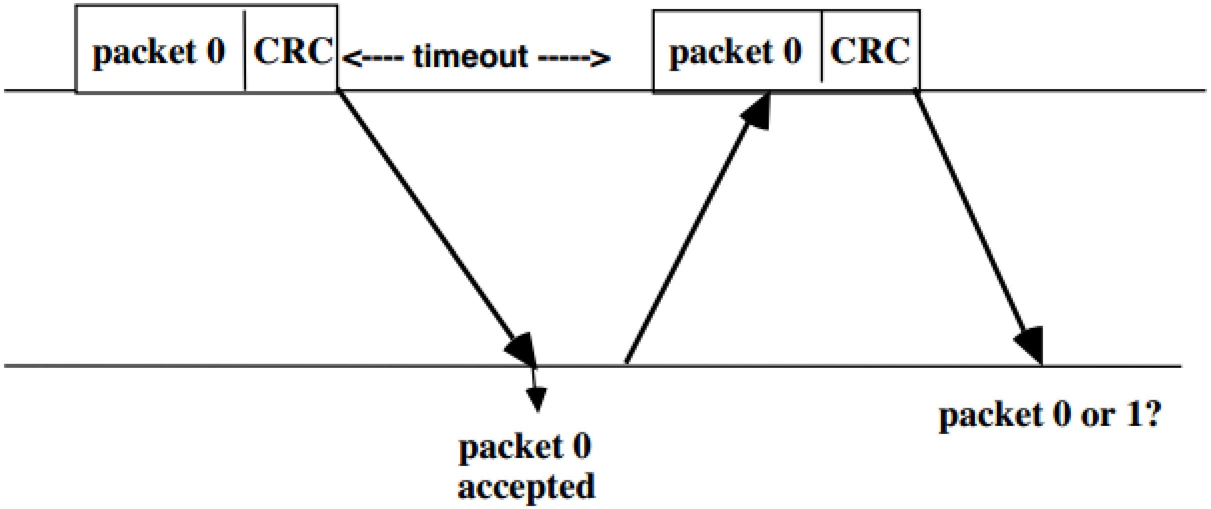
1. **Stop and Wait ARQ** is a simple error control method in data transmission where a sender transmits one data frame/packet at a time and waits for an acknowledgment (ACK) from the receiver before sending the next frame.



2. **If the ACK isn't received within a certain time frame (timeout),** the sender retransmits the same frame.

STOP-AND-WAIT ARQ (2)

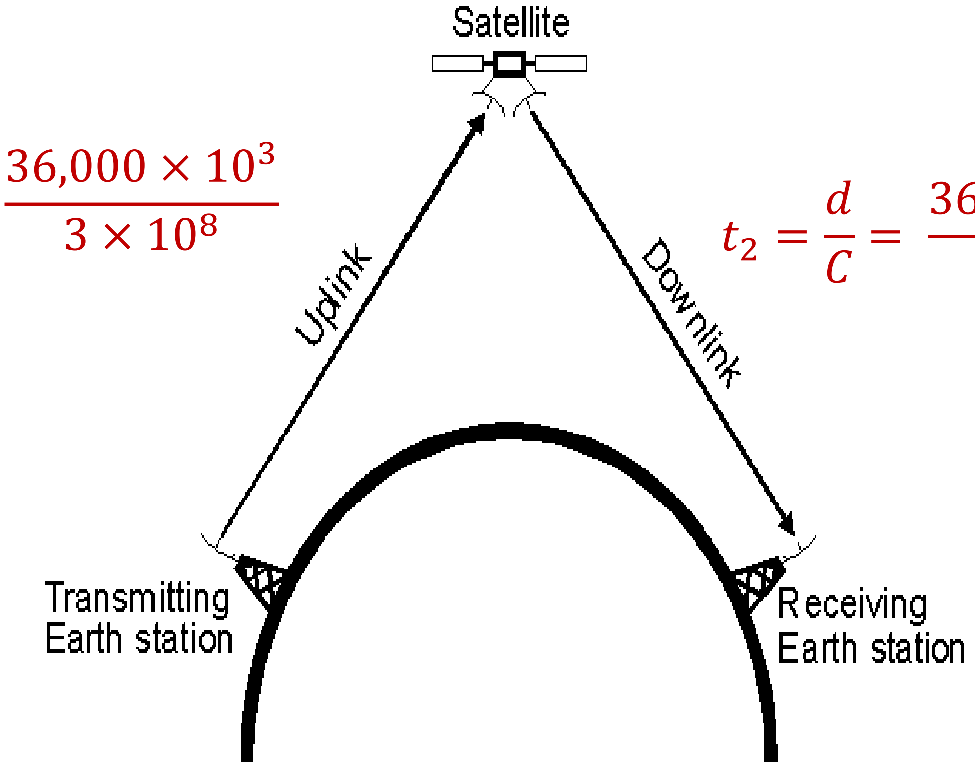
The Use of Timeouts For Lost Packets Requires Sequence Numbers



1. **Problem:** Unless packets are numbered, the receiver cannot tell which packet it received.
2. **Solution:** Stop-and-Wait ARQ uses sequence numbers to number the frames. The sequence numbers are based on modulo-2 arithmetic.

PROBLEMS OF STOP-AND-WAIT

1. **Stop and Wait is inefficient when propagation delay is larger than the packet transmission time**
2. Example is satellite communication where a round trip is nearly 0.5 seconds.

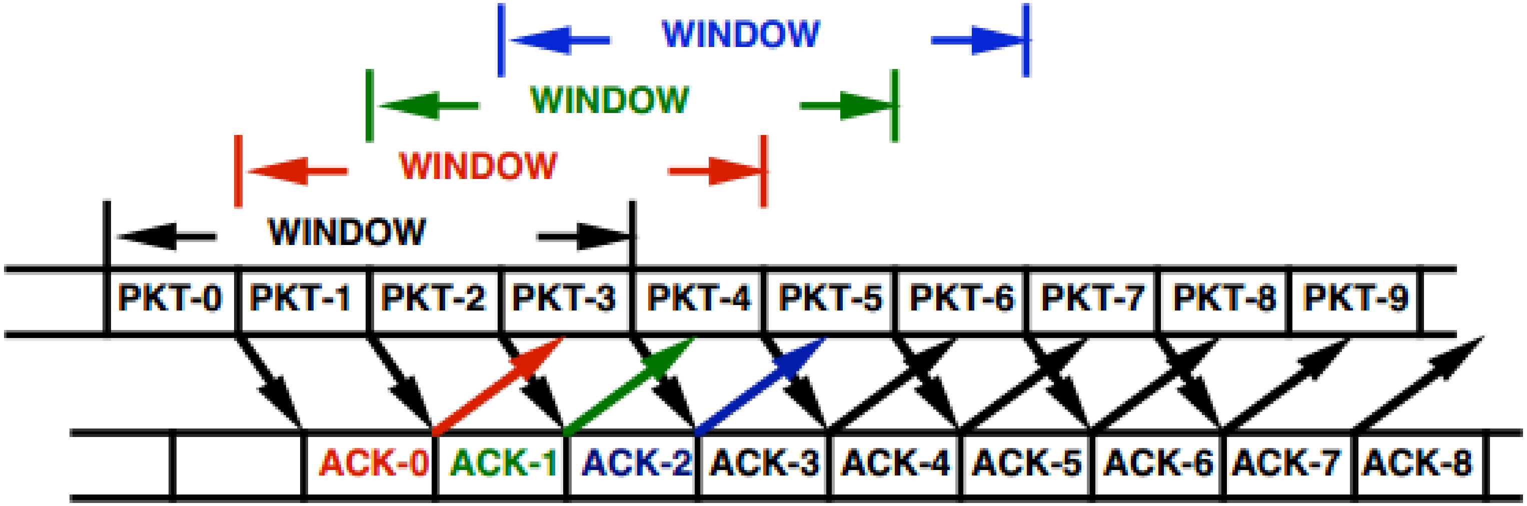


$$t_1 = \frac{d}{C} = \frac{36{,}000 \times 10^3}{3 \times 10^8}$$

$$t_2 = \frac{d}{C} = \frac{36{,}000 \times 10^3}{3 \times 10^8}$$

GO BACK N ARQ

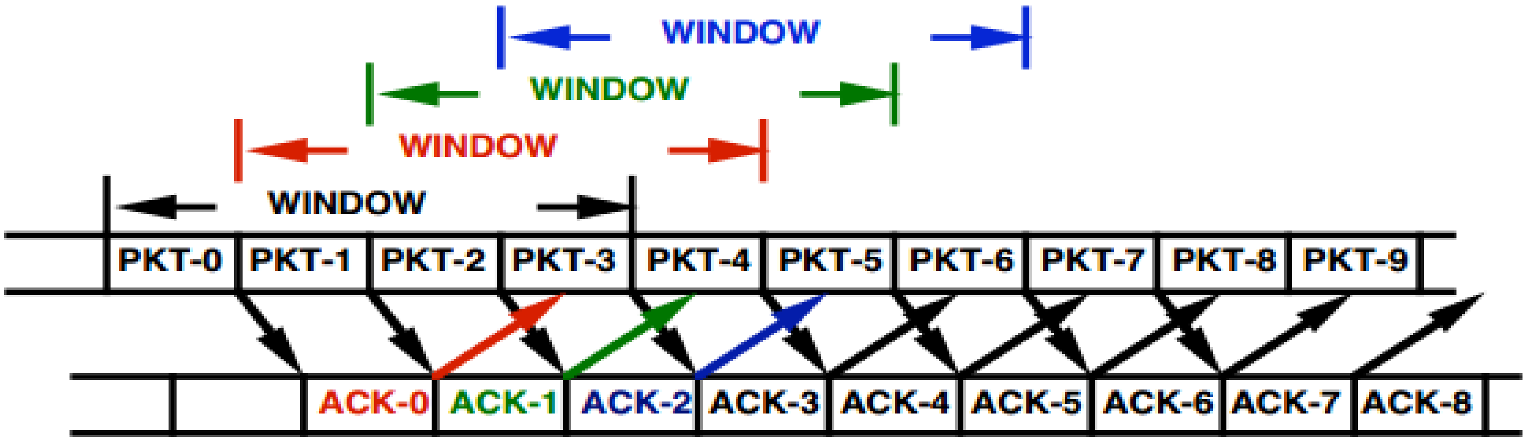
1. **Go-Back-N ARQ** is a specific instance of the automatic repeat request (ARQ) protocol, in which the sending process continues to send a number of frames without receiving an acknowledgement (ACK) packet from the receiver.
2. **Go-Back-N ARQ** uses a window mechanism where the sender can send packets that are within a "window" (range) of packets.
3. **The window advances as acknowledgements for earlier packets are received.**



FEATURES OF GO BACK N ARQ

Window size = N

1. Sender cannot send packet i+N until it has received the ACK for packet i
2. Receiver operates just like in Stop and Wait, i.e it receives packets in order and it cannot accept packet out of sequence



CONDITIONS FOR GO BACK N TO FUNCTION CORRECTLY

Go Back N is guaranteed to work correctly, if

1) System is correctly initialized

2) There are no failures in detecting errors

3) Packets travel in First-Come-First-serve (FCFS) order

4) There is Positive probability of correct reception

5) Transmitter occasionally resends (e.g., upon timeout)

6) Receiver occasionally sends RN

ADVANTAGES & DISADVANTAGES OF GO BACK N ARQ

1. **The advantages** of Go-Back-N ARQ Protocol are simplicity and efficiency.

2. **The Disadvantage** of Go Back N is this need to re-send the entire window when an error occurs. This leads to increased delay and potential for unnecessary retransmissions.

SELECTIVE REPEAT ARQ

1. **Selective Repeat ARQ** uses a sliding window technique to ensure reliable data transmission by only retransmitting individual lost or corrupted data frames.
2. For this to happen,
    1. Receiver must be able to accept packets out of order
3. The receiver, just like the transmitter, must be able to buffer packets.

WHERE IS GO-BACK-ARQ USED?

1. **TCP (Transmission Control Protocol):** The foundation of reliable communication on the internet, TCP heavily relies on a variant of the Go-Back-N ARQ protocol to ensure data integrity.
2. **Serial port communication:** Go-Back-N ARQ can be used to guarantee reliable data delivery, especially in environments with potential noise or interference.
3. **Satellite communication:** Due to the high latency in satellite links, Go-Back-N ARQ is often employed to manage retransmissions efficiently.
4. **Industrial control systems:** When critical data needs to be transmitted reliably in industrial environments, Go-Back-N ARQ can be used to ensure proper control system operation.