Badji Mokhtar University Annaba
Electronics Department

Lecture : Error Detection and correction

Contact:
seifallah.nasri@univ-annaba.org

December, 2019 Annaba, Algeria

Tx

Rx

0101101

Dealing with Errors

- Transmission impairments can lead to bit errors
- Error types at receiver:
  - One or more bit errors in payload (damaged frame)
  - One or more bit errors in header/trailer (damaged frame)
  - Frame not received (lost frame)
  - Frame received out-of-order
- Error detection
  - Attach extra information to data (in header or trailer) to allow receiver to check if received data is correct (Error Detection)
  - Include sequence numbers in header to identify if frames received in correct order (ARQ)
- Error correction
  - Attach extra information or transform data to allow receiver to check and correct bit errors (Forward Error Correction)
  - Receiver asks transmitter to re-transmit lost/damaged frame (ARQ)

Error Detection Example: Odd-Parity Check

- Odd-parity check: append parity bit to block of data; resulting set of bits has odd number of ones
- Receiver detects an error if receiver bits has unexpected number of ones (transmitter and receiver both know parity scheme being used)
- Assume character S is to be sent using odd-parity check. What is transmitted? What happens if the last bit is corrupted? What about the last two bits? What is the overhead?

Example of Digital Data: Text

Last 4 bits \ First 3 bits	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

S = 1010011 , odd parity

A ⟶ B

parity ↘

Tx : 11010011

Errors	Rx	Assume	Result
no	11010011	✓	✓
8th	11010010	X	✓
1st	01010011	X	✓
2nd, 5th	10011011	✓	X failed

$$\text{Efficiency} = \frac{7}{8} = 87.5\%$$

Payload = k eg. 7

Transmit = n eg. 8

$$\text{Efficiency} = \frac{k}{n}$$

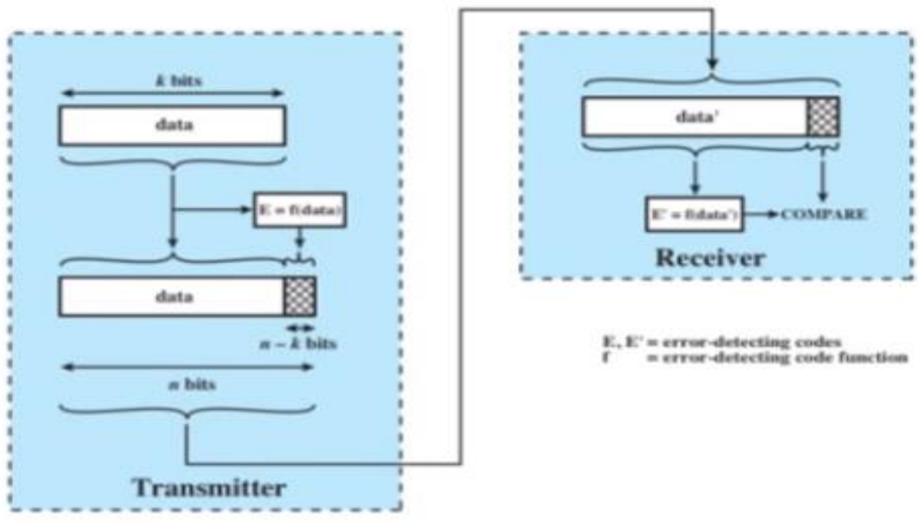
$n-k$ = error detecting code

Larger code → lower efficiency

Larger code → better error detection

Error Detection Concept

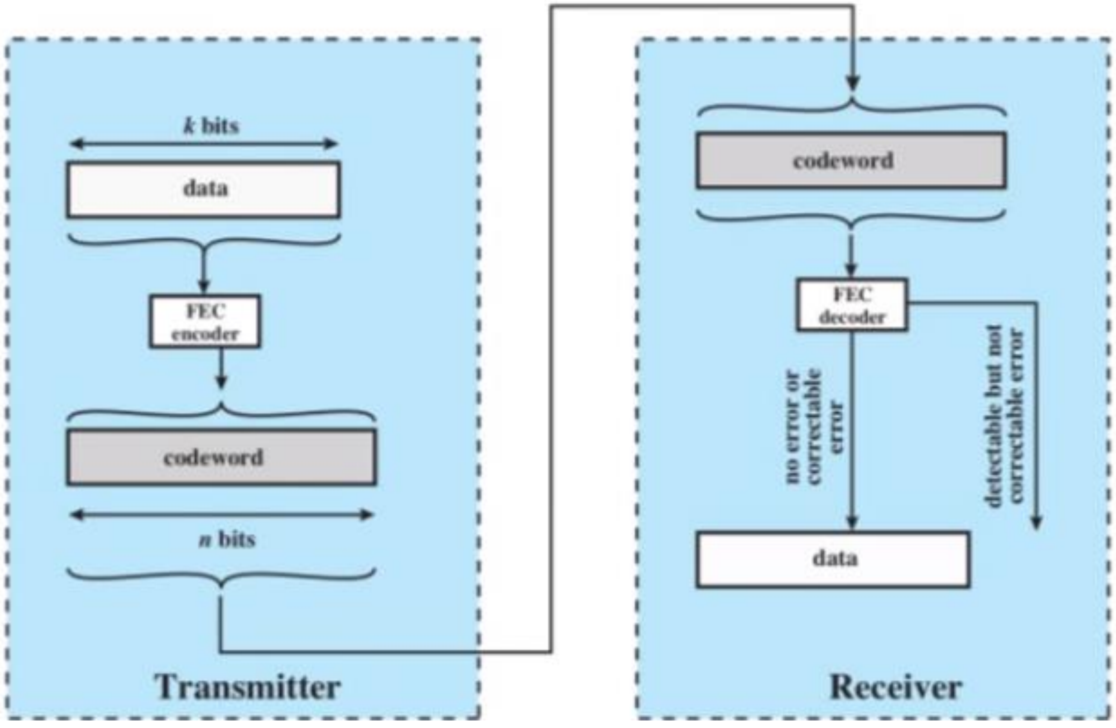
- Transmitter adds extra information to transmitted data, i.e. an error–detecting code
- Receiver recalculates the error–detecting code from received data; compares to received error-detecting code
- If the same, good. If not, then error (in data or code). Still a chance that an error is not detected



- Detection capability depend on algorithm & code length
- Cyclic Redundancy Check (CRC) very common

Forward Error Correction

- Sender sends a codeword (instead of data); codeword chosen such that if error detected, receiver can correct the error without retransmission
- Depending on encoding scheme and pattern of errors, receiver may: detect and correct errors; detect, but not correct errors; not detect errors



Example: FEC with Hamming Distance

Hamming Distance

- Number of bits of two n-bit sequences that differ
- $v_1 = 011011$, $v_2 = 110001$: $d(v1, v2) = 3$

Example FEC Encoder

- 2-bits of data mapped to 5-bit codeword ($k = 2$, $n = 5$)

Data	*Codeword*
00	00000
01	00111
10	11001
11	11110

- If received codeword invalid, assume valid codeword that is unique minimum Hamming distance from received codeword was transmitted

$$V_1 = 011011$$

$$V_2 = 110001$$

1 2 3

$$d(V_1, V_2) = 3$$

Data	Codeword
00	00000
01	00111
10	11001
11	11110

Tx data : 01

Tx codeword : 00111 —Errors→ Rx

Efficiency $= \frac{2}{5} = 40\%$

Errors	Rx codeword		Rx data	
no	00111	no errors	01	✓
3rd	00011	detected	01	✓
1st, 4th	10101	detected	–	✓
3rd, 4th	00001	detected	00	X

1st, 4th (Rx codeword 10101):

00000	3
00111	2] cannot correct
11001	2]
11110	3

3rd, 4th (Rx codeword 00001):

00000	1 ← correction
00111	2
11001	2
11110	5

Thank you for your attention