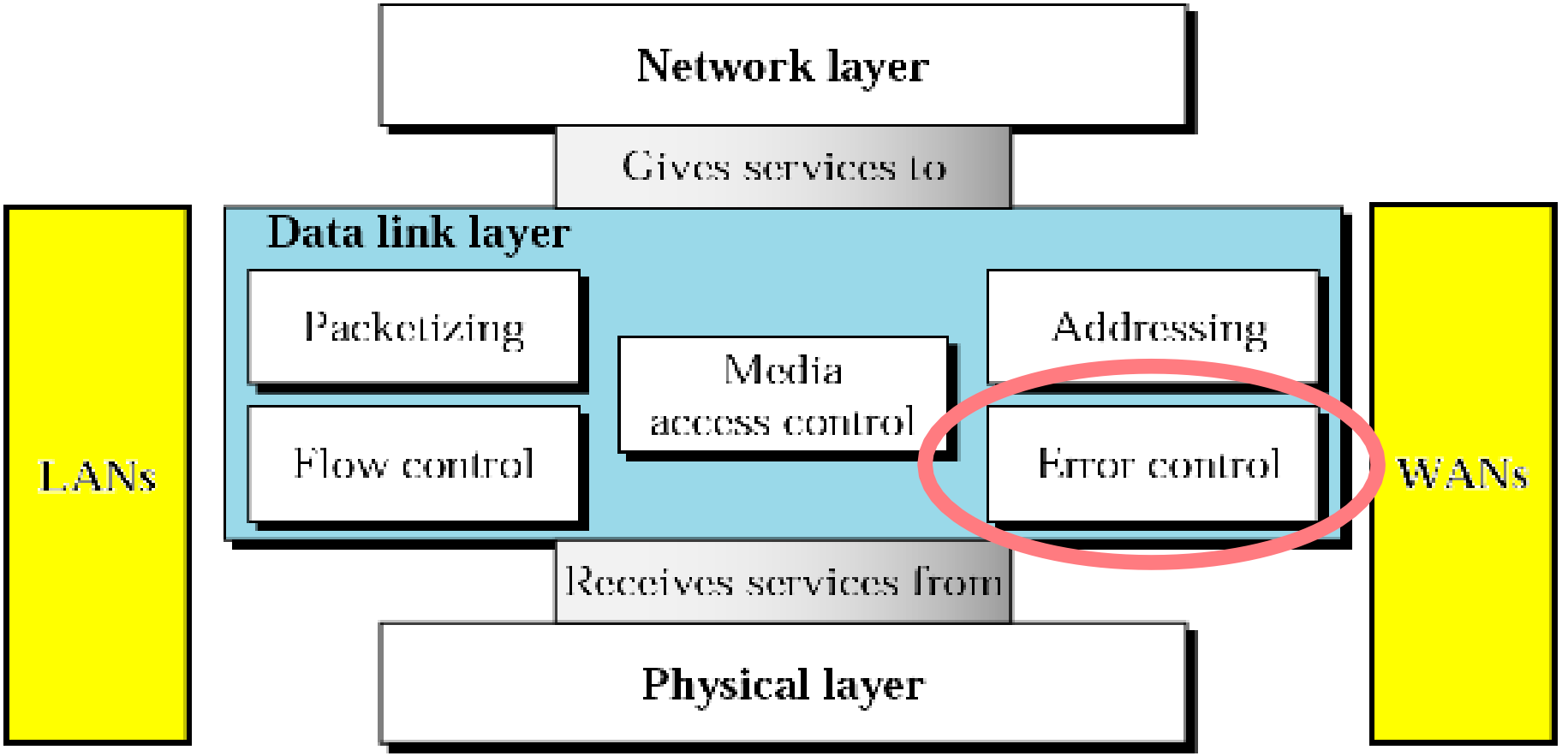
CRC - Error Detection & Error Correction

Name – Sangita Ghosh
Stream – CSE
Year – 3rd
Semester – 6th
Roll No. – 16800117029
Subject – Computer Network Assignment

Data Link Layer

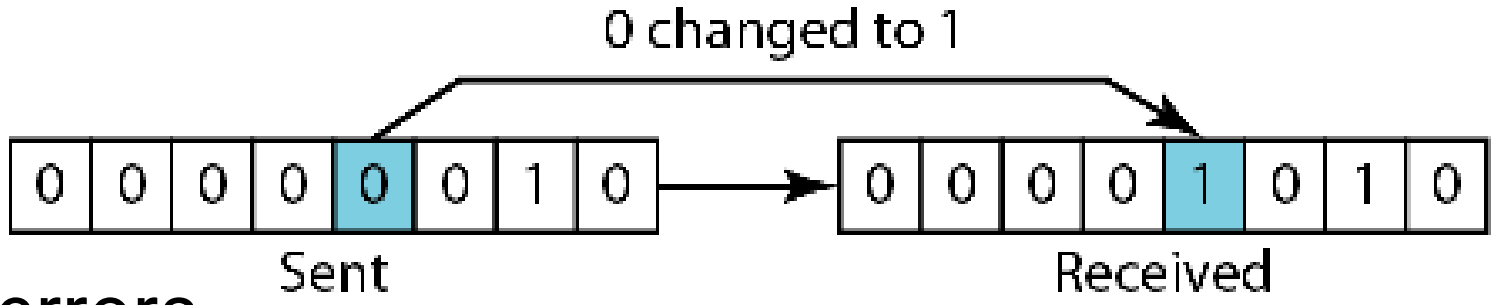


Definition of Error:

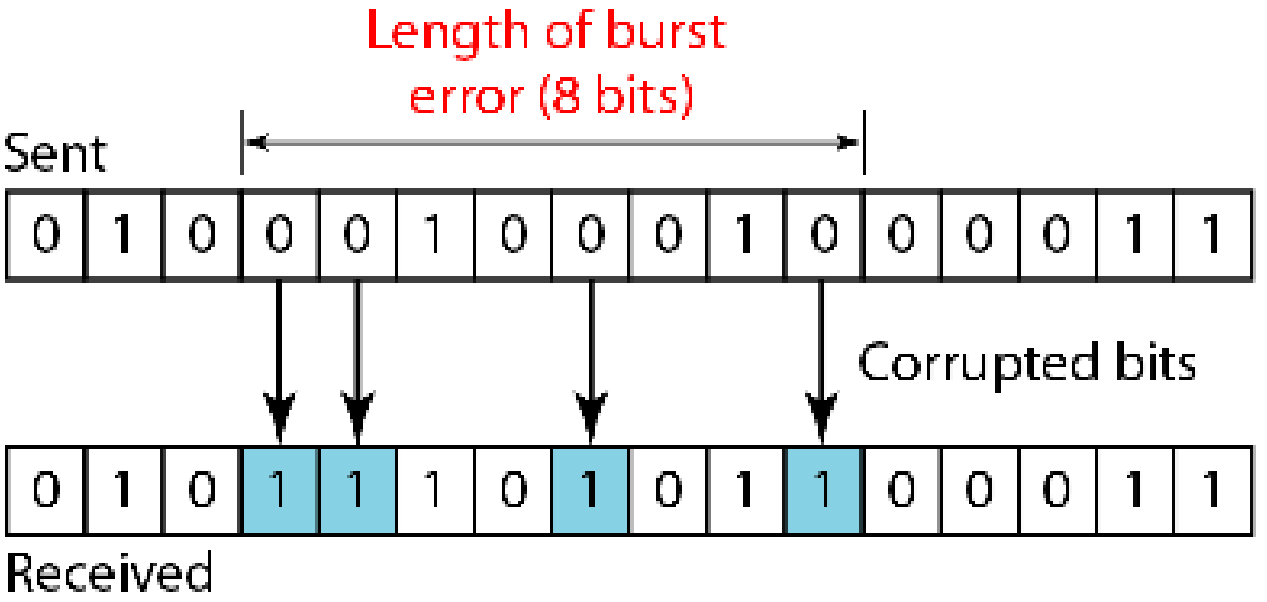
- Networks must be able to transform data from once device to another with complete accuracy. While the transmission data can be corrupted, for reliable communication errors must be detected and corrected.

Types of Errors

- Single-bit errors

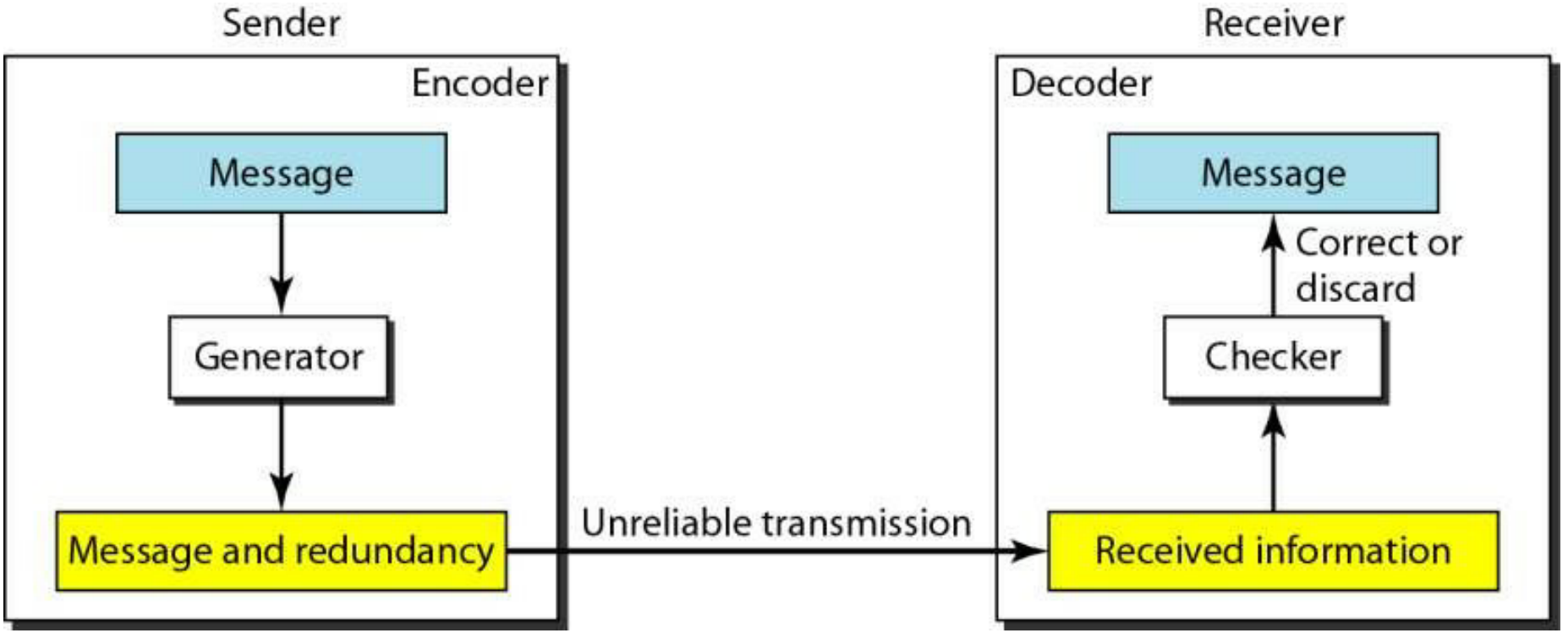


- Burst errors



Redundancy

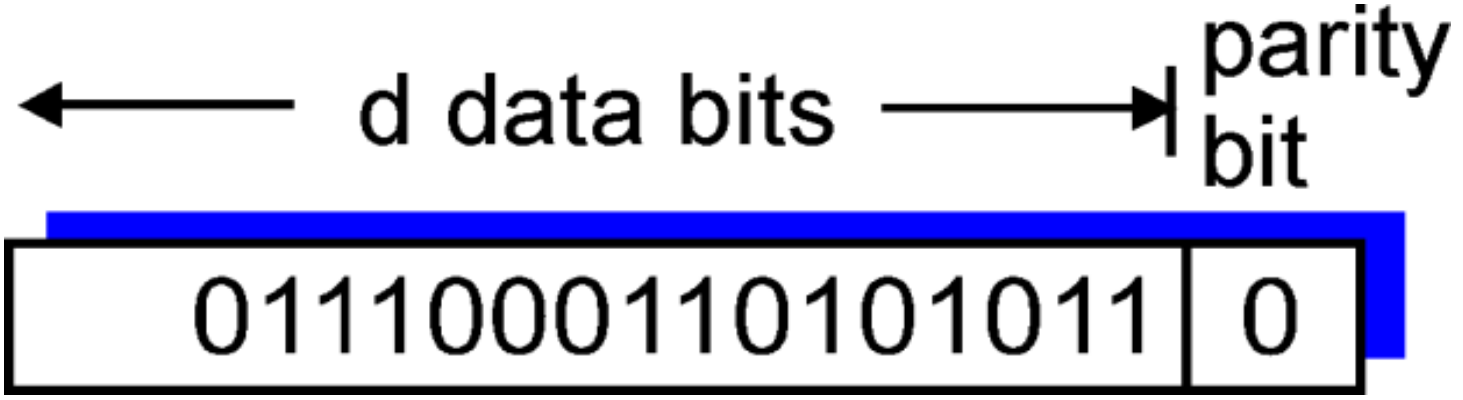
- To detect or correct errors, redundant bits of data must be added



Detection/Correction Techniques

- Parity Checks
- Checksumming methods
- Cyclic redundancy checks

Parity Checks



- Parity Bit (PB)
- One additional bit per character
- *Even parity Odd Parity*
-

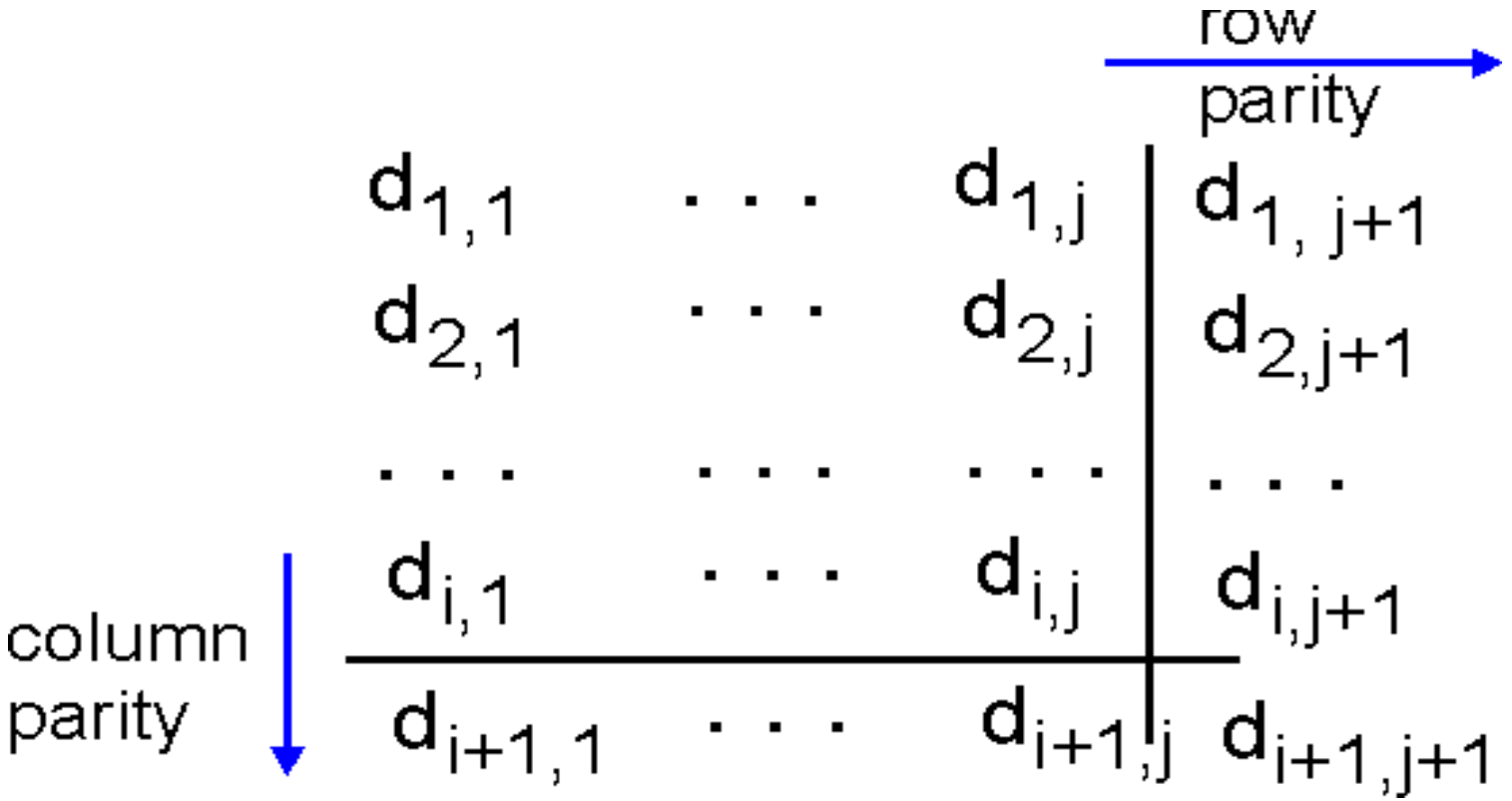
How many bit errors can PB detect ?

10001110 ---□ 10101110 => error !

10001110 ---□ 10100110 => **No error detected** !!!

Conclusion – 1 PB can only detect an odd number of errors !

Single Bit Error Correction



Parity for each character(byte=line) + parity for each column (set of data bytes sent)

Example - Single Bit Error Correction

10101	1
11110	0
01110	1
10101	0

no errors

10101	1
10110	0
01110	1
10101	0

parity error

parity error

Hamming - Correctable single bit error

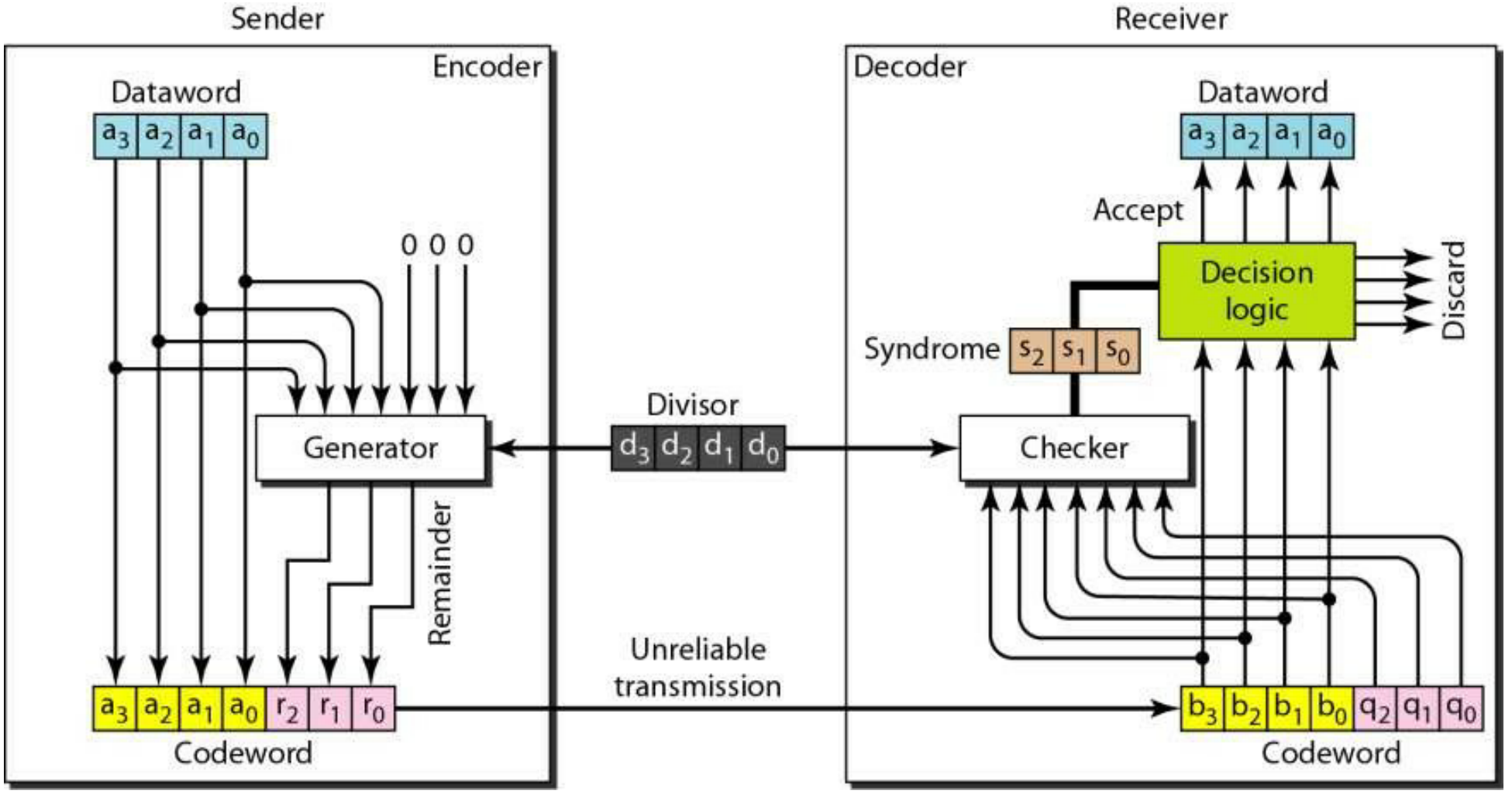
Cyclic Redundancy Checksum (CRC)

- CRC error detection method treats packet of data to be transmitted as a large polynomial
- Transmitter
  - Using polynomial arithmetic, divides polynomial by a given generating polynomial
- Quotient is discarded
  - Remainder is “attached” to the end of message

Cyclic Redundancy Checksum (continued)

- Message (with the remainder) is transmitted to the receiver
- Receiver divides the message and remainder by same generating polynomial
- If a remainder not equal to zero results error during transmission
- If a remainder of zero results error during transmission

CRC Encoder/Decoder

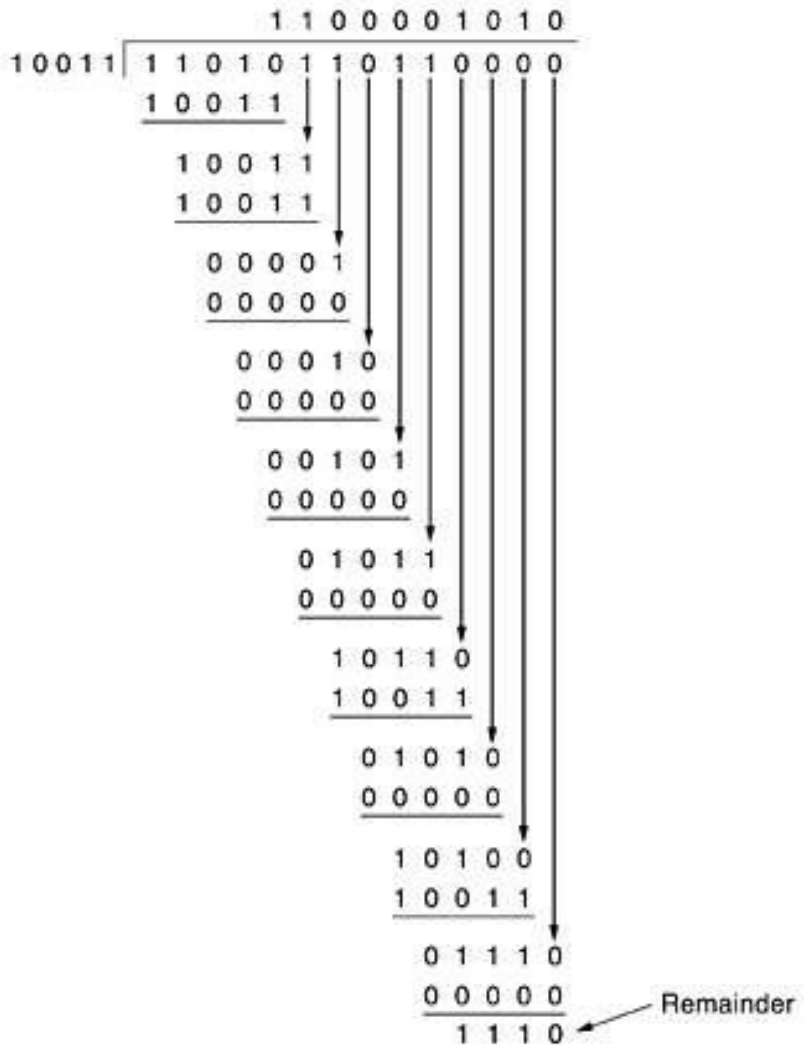


CRC -

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 0

Frame – 1101011011

$G(x)=x^4+x+1$

Transmitted frame:

11010110110000 –

00000000001110

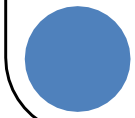
11010110111110

Checksums

- • A checksum “adds” together “chunks” of data
- – The “add” operation may not be normal integer addition
- – The chunk size is typically 8, 16, or 32 bits
- • We’ll discuss:
- – Integer addition “checksum”
- – One’s complement “checksum”
- – Fletcher Checksum
- – Adler Checksum
- – ATN Checksum (AN/466)

Error Correction

- Two methods
  - Retransmission after detecting error
  - Forward error correction (FEC)

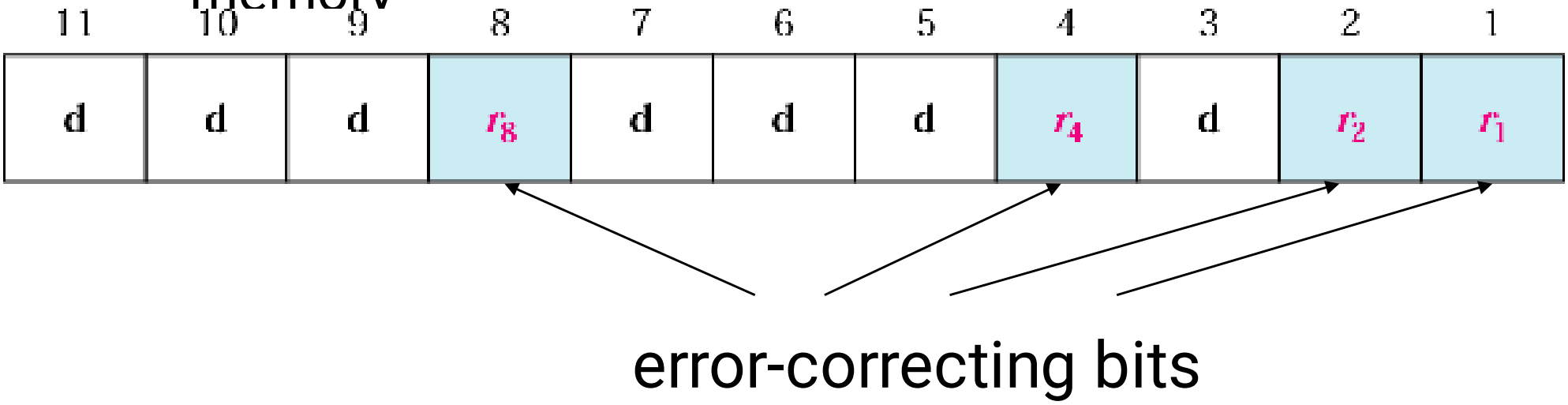


Number of Redundant Bits

Number of data bits k	Number of redundancy bits r	Total bits $k + r$
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

Hamming Code

- Simple, powerful FEC
- Widely used in computer memory
  - Known as ECC memory



Redundant Bit Calculation

r_1 will take care of these bits.

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_2 will take care of these bits.

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

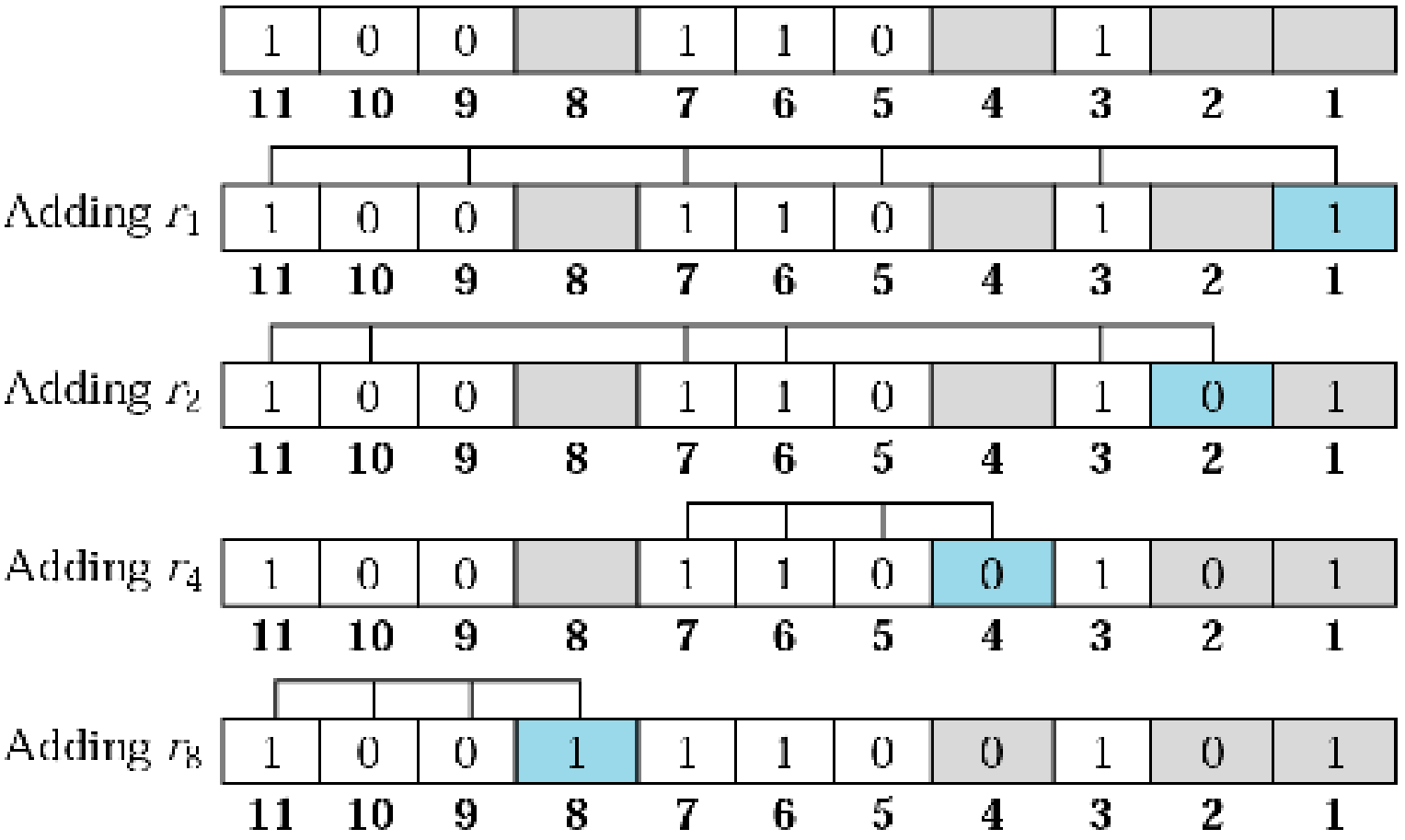
r_4 will take care of these bits.

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_8 will take care of these bits.

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

Example: *Hamming Code*



Data:
1 0 0 1 1 0 1

Code:
1 0 0 1 1 1 0 0 1 0 1

Name – Sangita Ghosh
Stream – CSE
Year – 3rd
Semester – 6th
Roll No. – 16800117029
Subject – Computer Network Assignment

THANK YOU