Chapter 10

Error Detection and Correction

Error

(i) Data are transmitted in the network

(ii) The data can be corrupted during transmission.

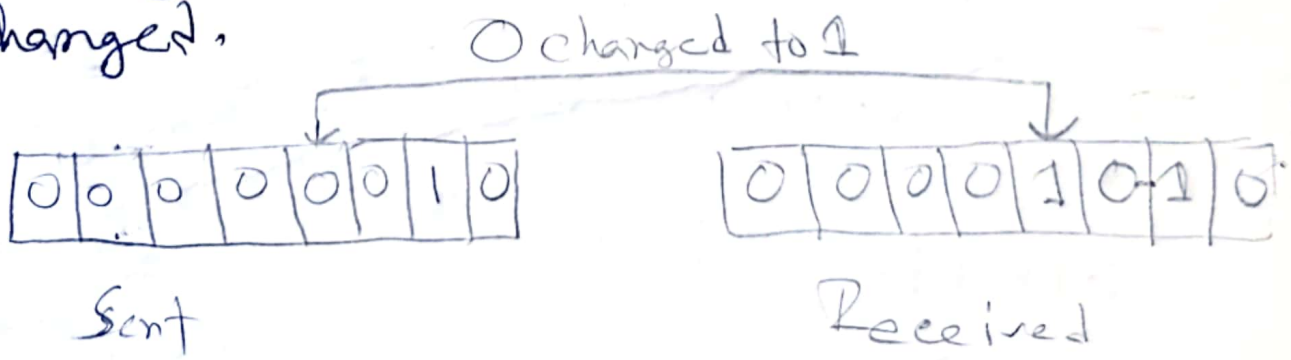
(iii) Handled → Transport or Data link layer

(iv) 2 types of error → (1) Bit error
(2) Burst error

(1) Bit Error

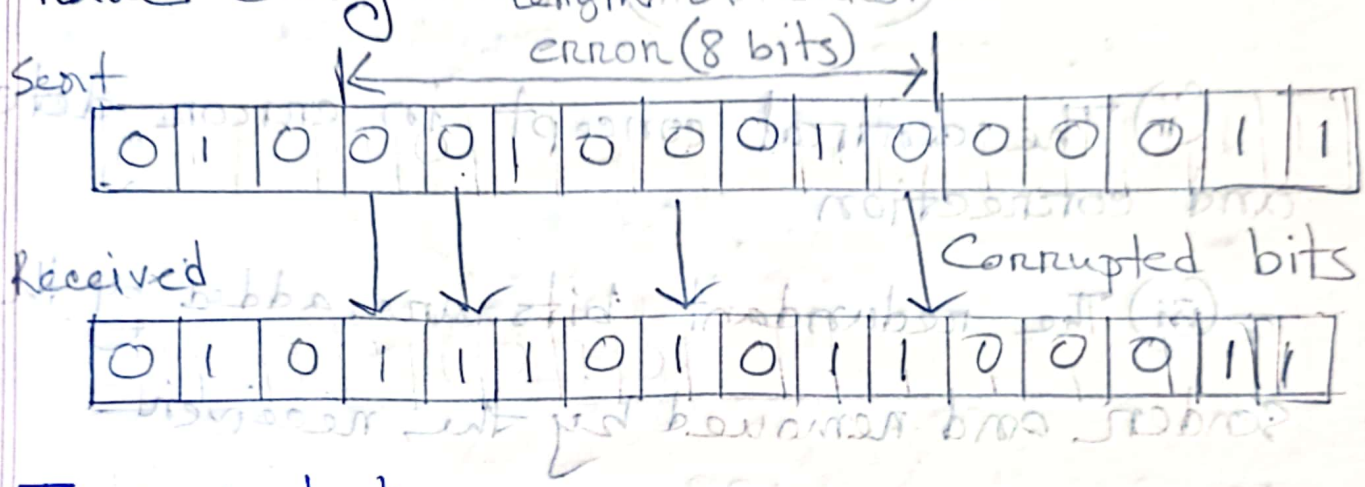
(1) Single Bit Error

(2) Only 1 bit in the data unit has been changed.



(ii) Burst Error

(i) 2 or more ~~ers~~ bits in the data unit have changed.



To detect the errors

(i) Error detection means to decide whether the received data is correct or not without having a copy of the original message

(ii) To detect or correct errors, we need to send some extra bits with the data.

(iii) The extra bits are called as redundant bits

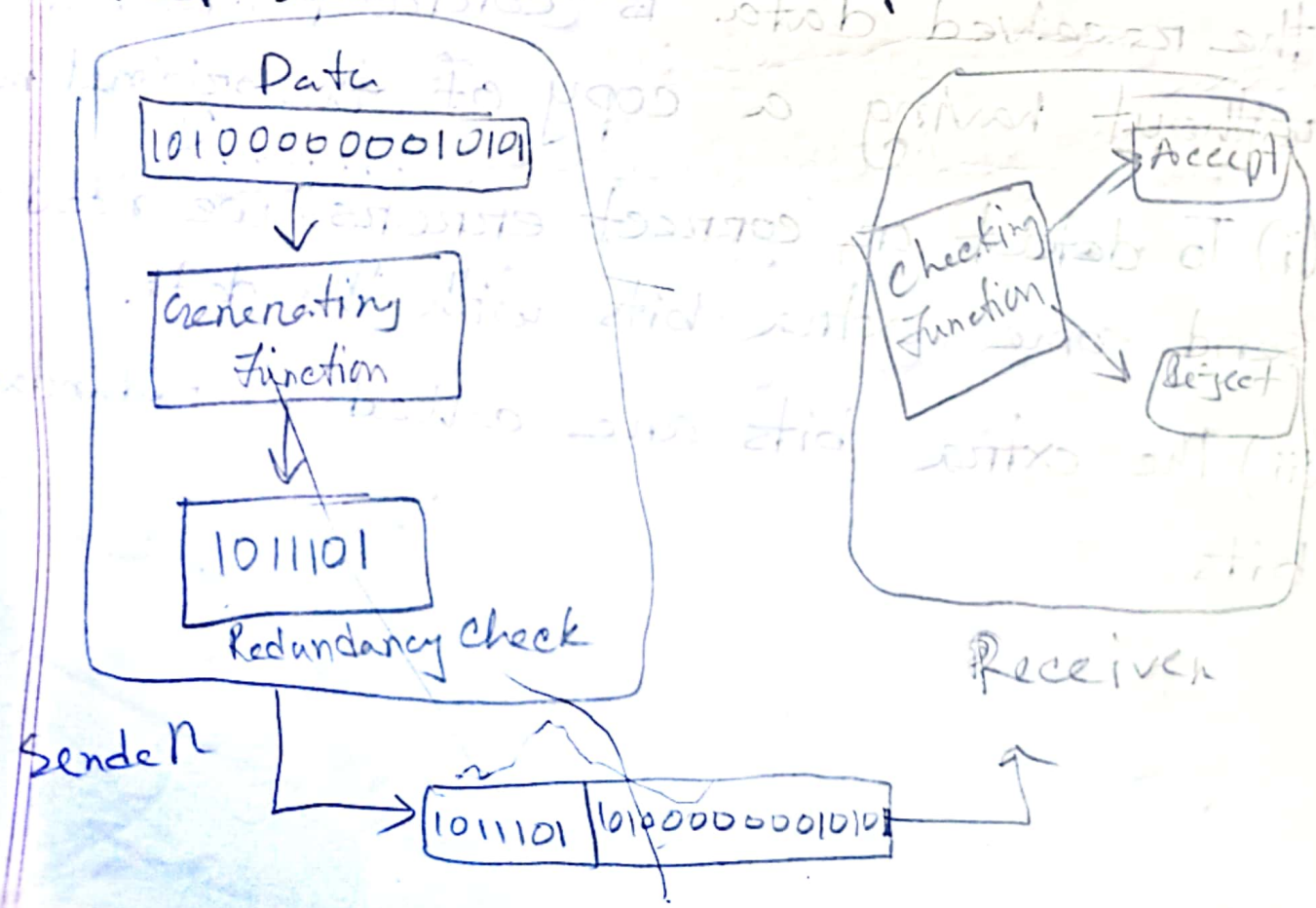
Redundancy

(i) To detect or correct errors, we need to send extra (redundant) bits with data

(ii) The central concept in error detection and correction

(iii) The redundant bits are added by the sender and removed by the receiver.

(iv) Their presence helps the receiver to detect and correct corrupted bits.



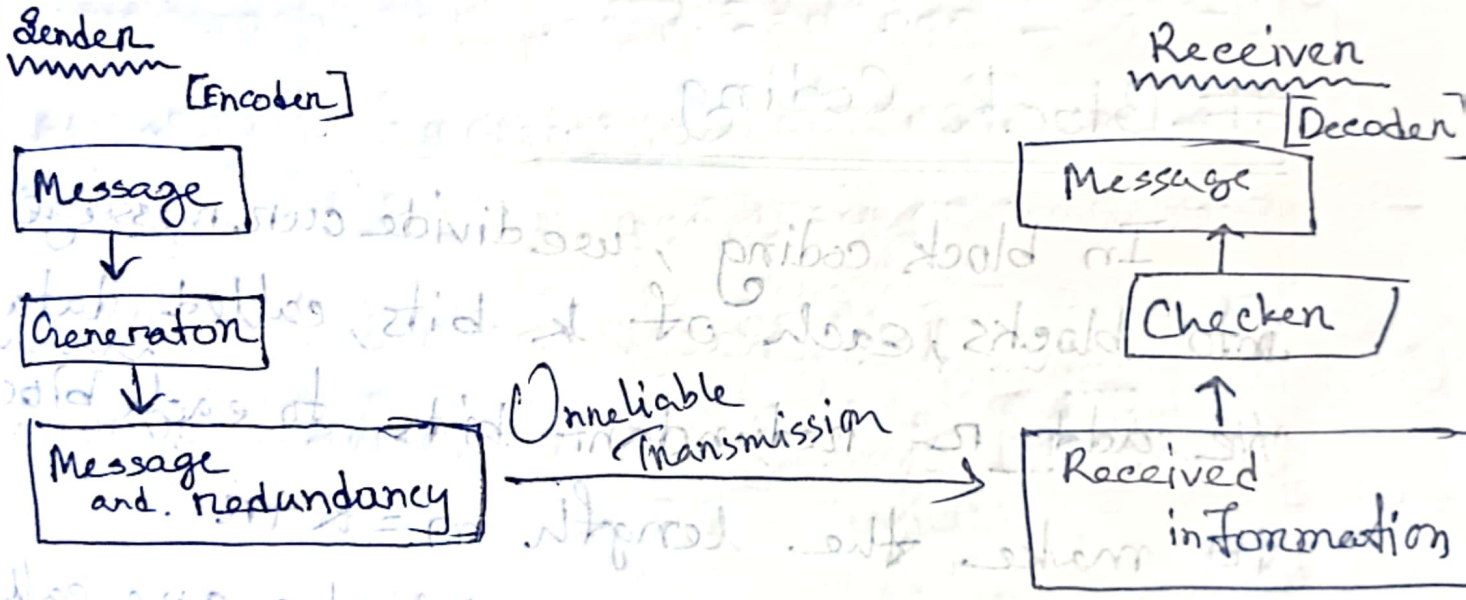
Error Correction

It can be handled in 2 ways.

(i) Receiver can have the sender retransmit the entire data unit.

(ii) The receiver can use an error-correcting code, which automatically corrects certain errors.

Error Detection / Correction



Coding

(i) Redundancy is achieved through various coding schemes.

(ii) The coding schemes are divided into two broad categories

- Block Coding
- Convolution Coding.

Block Coding

In block coding, we divide our message into blocks, each of k bits, called datawords. We add r redundant bits to each block to make the length $n = k + r$.

The resulting n-bits blocks are called codewords.

k bits	k bits	...	k bits

2^k Datawords, each of k bits

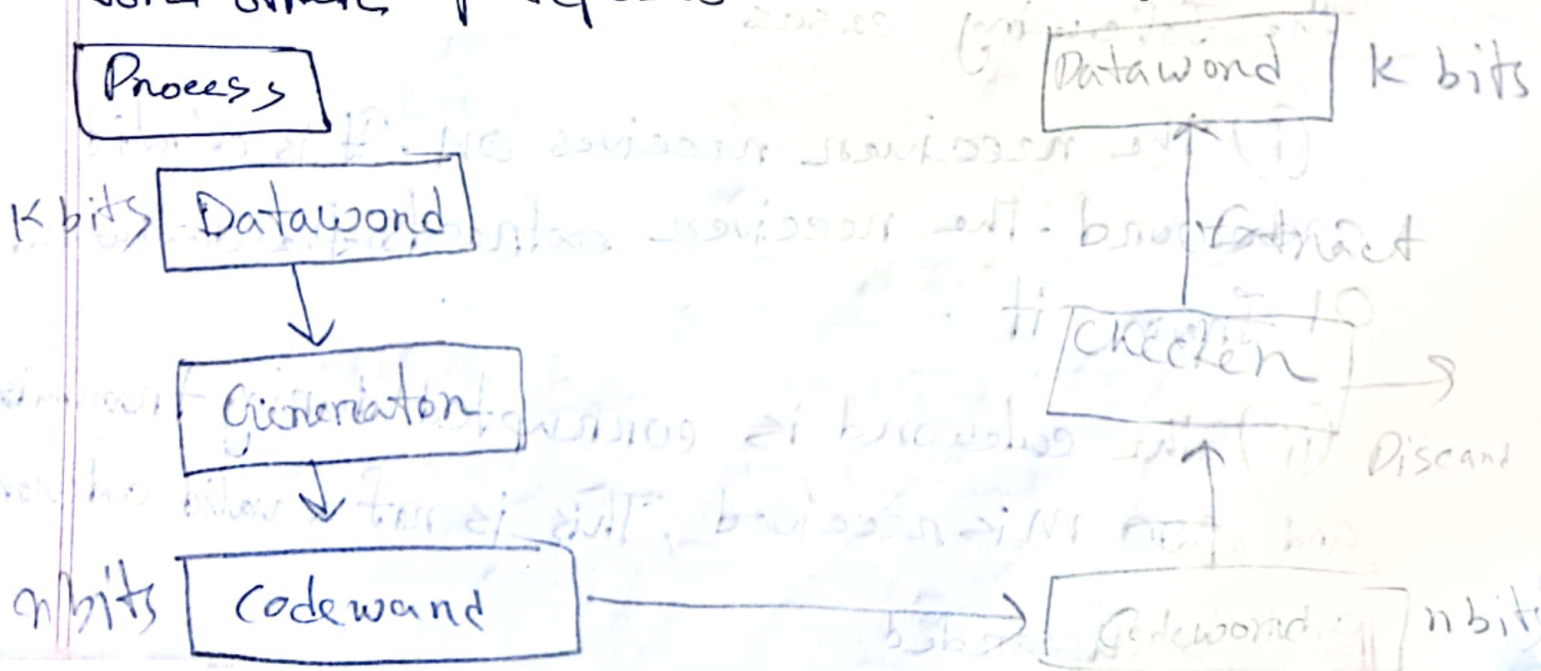
n bits	n bits	...	n bits

2^n Codewords, each of n bits (Only 2^k of them are valid)

Example

4B/5B block coding. In this coding scheme, $k=4$, $n=5$. As we saw, we have $2^k=16$ datawords, and $2^n=32$ codewords. We saw that 16 out of 32 codewords are used for message transfer, and the rest are either used for other purposes or unused.

Process



A code for error detection

Datawords	Codewords
00	000
01	011
10	101
11	110

Let, $k=2$ and $n=3$. The table here shows the list of datawords and codewords. Later, we will see how to derive a codeword from dataword.

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases.

(i) The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.

(ii) The codeword is corrupted during transmission and 111 is received. This is not a valid codeword and is discarded.

(iii) The codeword is corrupted during transmission, and 000 is received. This is a valid codeword. The receiver incorrectly extracts the dataword 00.

These, two corrupted bits have made the error undetectable.

Note:

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

Hamming distance

number of differences between corresponding bits. XOR gate use করতে হবে।

Hamming distance between

1. $d(000, 011) \Rightarrow$

$$\begin{array}{r} 0\ 0\ 0 \\ \oplus\ 0\ 1\ 1 \\ \hline 0\ 1\ 1 \end{array} = 2$$

2. $d(10101, 11110) \Rightarrow$

$$\begin{array}{r} 1\ 0\ 1\ 0\ 1 \\ \oplus\ 1\ 1\ 1\ 1\ 0 \\ \hline 0\ 1\ 0\ 1\ 1 \end{array} = 3$$

Here $d_{min} = 2$

A code for error detection

Datawords	Codewords
00	000
01	011
10	101
11	110

There are 4 difference codewords which can have 6 possible pairs.

(000, 011) (000, 101) (000, 110)

(011, 101) (011, 110) (101, 110)

#

Possible Pairs,
(00000, 01011)
(00000, 10101)
(00000, 11110)
(01011, 10101)
(01011, 11110)
(10101, 11110)

Datawords	Codewords
00	00000
01	01011
10	10101
11	11110

$d_{min} = ?$

$d(00000, 01011) = 3$; $d(00000, 10101) = 3$

$d(00000, 11110) = 4$; $d(01011, 10101) = 4$

$d(01011, 11110) = 3$; $d(10101, 11110) = 3$

$\therefore d_{min} = 3$

To guarantee the detection of upto s errors in all cases, $d_{min} = s+1$

Correction upto t errors, $d_{min} = 2t+1$

$d_{min} = 4$; Error detection, $d_{min} = s+1$

$\Rightarrow 4 = s+1$

$\therefore s = 3$

Error correction, $d_{min} = 2t+1$

$\Rightarrow 4 = 2t+1$

$\therefore t \cong 1$

Cyclic Codes

Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

CRC generation at sender side

(i) Find the length of the divisor "L".

(ii) Append "L-1" bits to the original message.

(iii) Perform binary division operation

(iv) Remainder of the division = CRC

[The CRC must be "L-1" bits]

Find the CRC for the data blocks 100100 with the divisor 1101?

[Here, L=4
∴ So number of 0 should be appe
= L-1 = 4-1 = 3]

```
        111101
1101 ) 100100 000
       1101
       ----
        1000
        1101
        ----
         1010
         1101
         ----
          1110
          1101
          ----
           0110
           0000
           ----
            1100
            1101
            ----
            [001] → Remainder
```

So, the CRC code = 001

∴ Data Transmitted = 100 100 001

∴ Code word = ~~1001110~~ 100100110

Now to detect errors of the same code.

Codeword = 1001110

```
        1010
1011 ) 1001110
       1011
       ----
        0101
        0000
        ----
         1011
         1011
         ----
          0000
          0000
          ----
           000
```

→ if the remainder is 000 then it is acceptable. [which must be L-1]

```
        101011
1011 ) 100100110
       1011
       ----
        0100
        0000
        ----
         1000
         1011
         ----
          0111
          0000
          ----
           1111
           1011
           ----
            1000
            1011
            ----
             011
```

→ Discarded

Datawond = 1001 ; Divisor = 1011

$k = 4$; $n = k-1 = 3$

```
      1010
1011 ) 1001000
       1011
       ----
        0000
        0000
        ----
         1000
         1011
         ----
          0110
          0000
          ----
           110
```

∴ Codeword = 1001 110

Datawond Remainder

```
      1010
1011 ) 1001110
       1011
       ----
        0101
        0000
        ----
         1011
         1011
         ----
          0000
          0000
          ----
           000
```

A polynomial to represent a binary word

a_6	a_5	a_4	a_3	a_2	a_1	a_0
1	0	0	0	0	1	1

$$1x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 1x^1 + 1x^0$$

$$= x^6 + x + 1$$

Here Dataword $= x^3 + 1$; Divison $= x^3 + x + 1$

$$x^3 + 1 = 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

$$= 1001$$

Divison $= x^3 + x + 1$

$$= 1 \times x^3 + 0 \times x^2 + 1 \times x + 1 \times x^0$$

$$= 1011$$

```
        1010
1011 ) 1001000
       1011
       ----
        0100
        0000
        ----
        1000
        1011
        ----
        0110
        0000
        ----
```

remainder = 110

x^2	a_1	a_0
1	1	0

$$x^2 + x + 0$$

$$= x^2 + x$$

Find the CRC for 1110010101 with the divison of x^3+x^2+1

1110010101

a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
1	1	1	0	0	1	0	1	0	1

$$x^9+x^8+x^7+x^4+x^2+1$$

divison = $x^3+x^2+1 = 1\times x^3+1\times x^2+0\times x+1\times x^0$

$= 1101$

Now,

```
                1010000110
1101 ) 1110010101000
       1101
       ----
        0110
        0000
        ----
         1101
         1101
         ----
          0000
          0000
          ----
           0001
           0000
           ----
            0010
            0000
            ----
             0101
             0000
             ----
              1010
              1101
              ----
               1110
               1101
               ----
                0110
                0000
                ----
                 110
```

So remainden

$= 110$

$= a$

a^2	a^1	a^0
1	1	0

x^2+x

(Ans)

Dataword

$= x^9+x^8+x^7+x^4+x^3+x^2+1+x^2+x$

The divisor in a cyclic code is normally called the generator polynomial or simply the generator.

Dataword : $d(x)$; Syndrome : $s(x)$

Codeword : $c(x)$; Error : $e(x)$

Generator : $g(x)$

In a cyclic code,

if $s(x) \neq 0$; one or more bits is corrupted

if $s(x) = 0$; either

a. No bit is corrupted. or

b. Some bits are corrupted, but the decoder failed to detect them.

In a cyclic code, those $e(x)$ errors that are divisible by $g(x)$ are not caught.

CRC-8 → ATM header ; CRC-10 → ATM AAL

CRC-16 → HDLC ; CRC-32 → LANs

Checksum

Operation at sender side

(i) Break the original message in to "k" number of blocks with "n" bits in each block.

(ii) Sum all the "K" data blocks

(iii) Add the carry to the sum, if any.

(iv) Do 1's complement to the sum
= Checksum

Consider the data unit to be transmitted is.

10011001111000100100100100000100

Here, n = 16 ; K = 8

(i)	(ii)	(iii)	(iv)
10011001	11100010	00100100	10000100

Sender

(i) → 10011001

(ii) → 11100010

carry ← (1) 0111 1011

1

0111 1100

(iii) → 00100100

1010 0000

(iv) → 10000100

(1) 0010 0100

1

0010 0101

∴ Checksum = 1101 1010

Now, Receiver

(i) Collect all the data blocks including the checksum.

(ii) Sum all the data blocks and checksum

(iii) If the result is all 1's, Accept, else Reject

So,

Sum of sender data + checksum =>

```
  0 0 1 0 0 1 0 1
  1 1 0 1 1 0 1 0
-----------------
  1 1 1 1 1 1 1 1
```

So, accept data.