

# Forward Error Correction

---

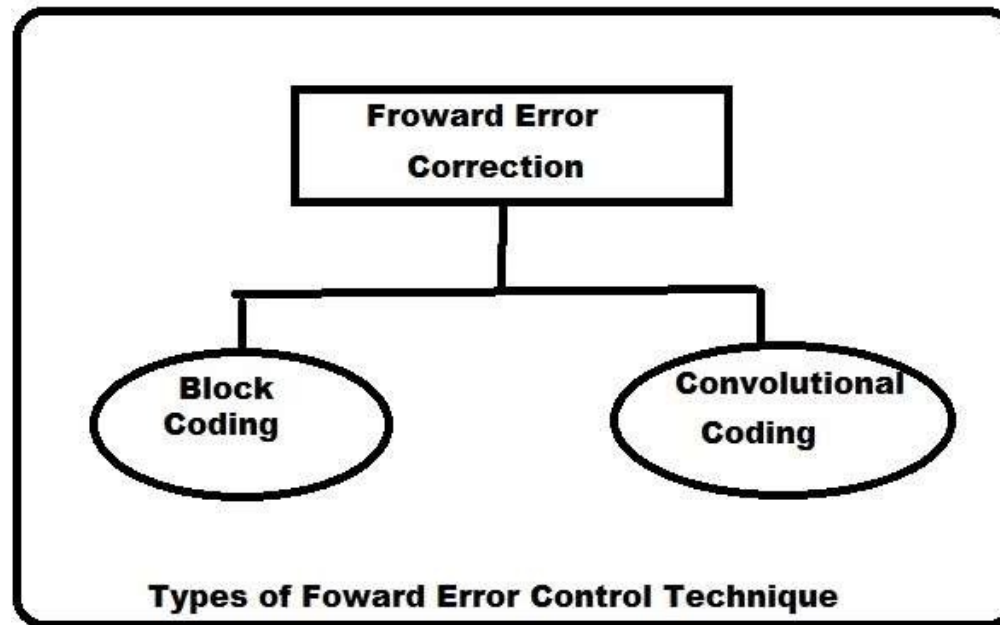
# Introduction

- FEC là hệ thống giúp phát hiện và tự sửa chữa lỗi khi truyền tin
- Bên gửi thêm các dữ liệu dư thừa gọi là Error Correction Code, bên nhận sử dụng các dữ liệu này để tự phát hiện và sửa lỗi mà không cần retransmit.
- Lượng dữ liệu dư thừa càng nhiều thì khả năng sửa lỗi càng tốt nhưng hiệu suất truyền tải dữ liệu càng giảm và ngược lại

# Ưu điểm và ứng dụng

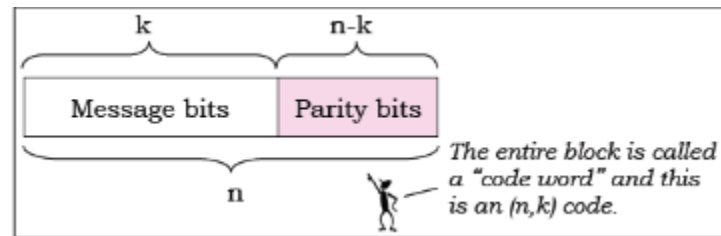
- Ưu điểm chính của FEC là việc truyền dữ liệu được thực hiện theo một chiều từ bên gửi đến bên nhận, không cần phản hồi cho các bản tin báo hiệu (ACK).
- Vì vậy FEC thường được sử dụng khi việc truyền lại dữ liệu là khó khăn hoặc tốn kém.

# Phân loại



# Block Coding

- Luồng dữ liệu vào được chia thành các block có độ dài cố định là  $k$  bit, từ đó sinh ra codeword có chiều dài  $n$  bit với  $n > k$ ,  $r = n - k$  là số bit dư thừa được thêm vào.
- Block code với  $n$  bit codeword và  $k$  bit dữ liệu được kí hiệu là  $(n, k)$
- Code rate:  $k/n$



# Block Coding

■ Biểu diễn toán học:

- $C = d[G]$

■ trong đó:

- C là codeword
- d là block dữ liệu ban đầu
- G là ma trận sinh,
  - $G = [IP]$ , trong đó
    - I là identity matrix, là ma trận đơn vị  $k \times k$
    - P là parity matrix cho biết vị trí của các bit được sử dụng để tính toán giá trị parity bit tương ứng.

# Block Coding

- Ví dụ cho blocking code (7,3)
- $d = (0 \ 1 \ 1 \ 1)$
- 

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$c = d[G] = (0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0)$$

# Block Coding

- Error detection and correction:
- Bên nhận nhận được

- $$\text{Received codeword} = \frac{(0 \ 1 \ 0 \ 1 \ \overbrace{1 \ 0 \ 0}^{\mathbf{c}_p})}{\mathbf{d}'}$$

- Code word hợp lệ khi

$$\mathbf{d}'[\mathbf{P}] + \mathbf{c}_p = [\mathbf{0}]$$



# Block Coding

$$\mathbf{d}' = (0 \ 1 \ 0 \ 1)$$

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{c}_p = (1 \ 0 \ 0)$$

$$\mathbf{d}'[\mathbf{P}] + \mathbf{c}_p = (0 \ 1 \ 0) + (1 \ 0 \ 0) = (1 \ 1 \ 0)$$

# Hamming Code

- Parity bit: cho biết số bit 1 trong một nhóm bit là chẵn hay lẻ. Giá trị của parity bit là XOR của tất cả các bit mà nó đại diện.
- Cho phép sửa 1 bit bị lỗi và phát hiện 2 bit lỗi với số bit dư thừa (parity bit) ít nhất
- Thuật toán:
  - Vị trí bit được đánh theo thứ tự 1, 10, 11, ...
  - Các bit ở vị trí là lũy thừa của 2 (1, 2, 4, 8, 16...) là các parity bit, các bit còn lại mang dữ liệu
    - Parity bit 1 đại diện cho các bit ở vị trí có biểu diễn nhị phân với bit cuối cùng là 1 (1, 11, 101, 111, ...)
    - Parity bit 2 đại diện cho các bit ở vị trí có biểu diễn nhị phân với bit thứ 2 là 1 (10, 11, 110, 111, 1010, ...) ...
  - Parity bit = 1 nghĩa là số bit 1 trong nhóm là 1 và ngược lại

# Hamming Code

- Codeword của chuỗi 1011

*	1	1	0	0	1	0	0	*
7	6	5	4	3	2	1		

# Hamming Code

## ■ Phát hiện và sửa lỗi:

- Giả sử trong quá trình truyền bit số 5 bị lỗi
- $X1 = X3 \text{ xor } X5 \text{ xor } X7 = 1$
- $X2 = X3 \text{ xor } X6 = 0$
- $X4 = X5 \text{ xor } X7 = 1$

Giá trị tính dc là 101 → bit 5 bị lỗi

1	1	1	0	1	0	0
7	6	5	4	3	2	1

# References

<https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470841516.app6>

[https://www.ittc.ku.edu/~frost/EECS\\_562/Error-Detection\\_EECS\\_562\\_Sp\\_2017.pdf](https://www.ittc.ku.edu/~frost/EECS_562/Error-Detection_EECS_562_Sp_2017.pdf)

[https://en.wikipedia.org/wiki/Error\\_correction\\_code](https://en.wikipedia.org/wiki/Error_correction_code)

<https://www.geeksforgeeks.org/hamming-code-in-computer-network/>