



# Vấn đề xử lý lỗi

Bởi:  
unknown

## Vấn đề xử lý lỗi

### Bộ mã phát hiện lỗi

Khi truyền tải một chuỗi các bit, các lỗi có thể phát sinh ra, bit 1 có thể biến thành bit 0 hay ngược lại.

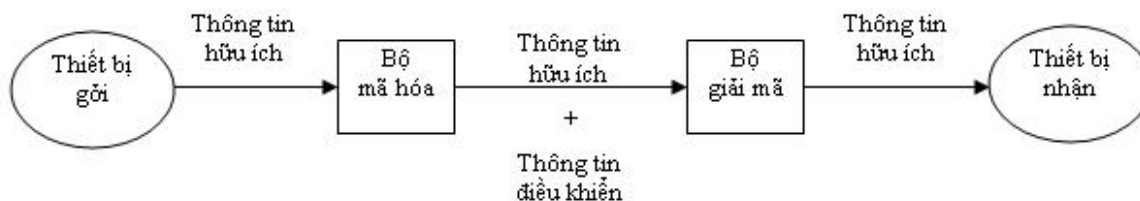
Ta định nghĩa tỷ lệ lỗi bởi tỷ số sau:

$$\tau = \text{Số bit bị lỗi} / \text{Tổng số bit được truyền}$$

Tỷ lệ lỗi này có giá trị từ  $10^{-5}$  đến  $10^{-8}$ . Tùy thuộc vào từng loại ứng dụng, một lỗi có mức độ nghiêm trọng khác nhau, chính vì thế cần có các cơ chế cho phép phát hiện lỗi cũng như sửa lỗi.

Các thống kê cho thấy rằng 88% các lỗi xảy ra do sai lệch một bit và 10% các lỗi xảy ra do sự sai lệch 2 bit kế nhau. Chính vì thế ta ưu tiên cho vấn đề phát hiện các lỗi trên một bit và sửa đổi chúng một cách tự động.

Với ý tưởng như thế, ta sử dụng các mã phát hiện lỗi: bên cạnh các thông tin hữu ích cần truyền đi, ta thêm vào các thông tin điều khiển. Bên nhận thực hiện việc giải mã các thông tin điều khiển này để phân tích xem thông tin nhận được là chính xác hay có lỗi.



*Mô hình xử lý lỗi trong truyền dữ liệu(H4.5)*

Thông tin điều khiển được đưa vào có thể theo 2 chiến lược. Chiến lược thứ nhất gọi là bộ mã sửa lỗi (Error-correcting codes) và chiến lược thứ hai gọi là bộ mã phát hiện lỗi (Error-detecting codes). Bộ mã sửa lỗi cho phép bên nhận có thể tính toán và suy ra

được các thông tin bị lỗi (sửa dữ liệu bị lỗi). Trong khi bộ mã phát hiện lỗi chỉ cho phép bên nhận phát hiện ra dữ liệu có lỗi hay không. Nếu có lỗi bên nhận sẽ yêu cầu bên gửi gửi lại thông tin. Với tốc độ của đường truyền ngày càng cao, người ta thấy rằng việc gửi lại một khung thông tin bị lỗi sẽ ít tốn kém hơn so với việc tính toán để suy ra giá trị ban đầu của các dữ liệu bị lỗi. Chính vì thế đa số các hệ thống mạng ngày nay đều chọn bộ mã phát hiện lỗi.

### **Những bộ mã phát hiện lỗi (Error-Detecting Codes)**

Có nhiều sơ đồ phát hiện lỗi, trong đó có các sơ đồ thông dụng là:

- Kiểm tra chẵn lẻ (Parity checks)
- Kiểm tra thêm theo chiều dọc (Longitudinal redundancy check)
- Kiểm tra phần dư tuần hoàn (Cyclic redundancy check)

#### ***Kiểm tra chẵn lẻ (Parity Check)***

Sơ đồ phát hiện bit lỗi đơn giản nhất là nối một bit chẵn-lẻ vào cuối của mỗi từ trong khung. Một ví dụ tiêu biểu là việc truyền các ký tự ASCII, mà trong đó một bit chẵn-lẻ được nối vào mỗi ký tự ASCII 7 bit. Giá trị của bit này được lựa chọn sao cho có một số chẵn của bit 1, với kiểm tra chẵn (even parity) hoặc một số lẻ của bit 1, với kiểm tra lẻ (odd parity).

Ví dụ, nếu bên gửi đang truyền một ký tự ASCII G ( mã ASCII là 1110001) và đang dùng phương pháp kiểm tra lẻ, nó sẽ nối một bit 1 và truyền đi 11100011. Bên nhận sẽ kiểm tra ký tự nhận được và nếu tổng của các bit 1 là lẻ, nó xem như không có lỗi. Nếu một bit hoặc một số lẻ bất kỳ các bit bị lỗi đảo ngược thì rõ ràng bên nhận sẽ phát hiện được lỗi. Tuy nhiên, nếu hai hay một số chẵn bất kỳ các bit bị lỗi đảo ngược thì nó sẽ không phát hiện được lỗi. Trên thực tế những xung nhiễu lại thường đủ dài để có thể phá hủy hơn một bit, đặc biệt là với tốc độ dữ liệu cao. Do đó, cần phải có một phương pháp cải thiện trường hợp này.



### ***Kiểm tra phần dư tuần hoàn (Cyclic Redundancy Check)***

Để cải tiến hơn nữa các nhà thiết kế đã dùng kỹ thuật mới dễ dàng và hiệu quả được gọi là kiểm tra phần dư tuần hoàn, trong đó có thể sử dụng một số phương pháp cài đặt khác nhau như: modulo 2, đa thức, thanh ghi dịch và các cổng Exclusive-or.

Các thủ tục với modulo 2 diễn ra như sau. Với một thông điệp **M** có **k** bit cần gửi đi, bên gửi sẽ nối vào cuối thông điệp một chuỗi **F** có **r** bit, được gọi là **Chuỗi kiểm tra khung** (FCS: Frame Check Sequence). Chuỗi kiểm tra khung sẽ tính toán sao cho khung kết quả **T** được hình thành từ việc nối **M** với **F** (gồm **k + r** bit) có thể chia hết bởi số **P** nào đó được định trước. Bên gửi sẽ gửi T đi. Khi bên nhận nhận được T, nó sẽ thực hiện phép chia modulo T cho P. Nếu phép chia không hết, tức có số dư, bên nhận xác định rằng khung T đã bị lỗi, ngược lại là không có lỗi. Nếu khung không có lỗi, bên nhận sẽ tách thông điệp M từ T, là k bits trọng số cao của T.

Phương pháp này dùng phép chia modulo 2 trong việc chia T cho P, phép toán modulo 2 dùng một phép cộng nhị phân không nhớ và đó cũng chính là phép toán Exclusive-or. Ví dụ sau mô tả phép toán cộng và nhân modulo 2:

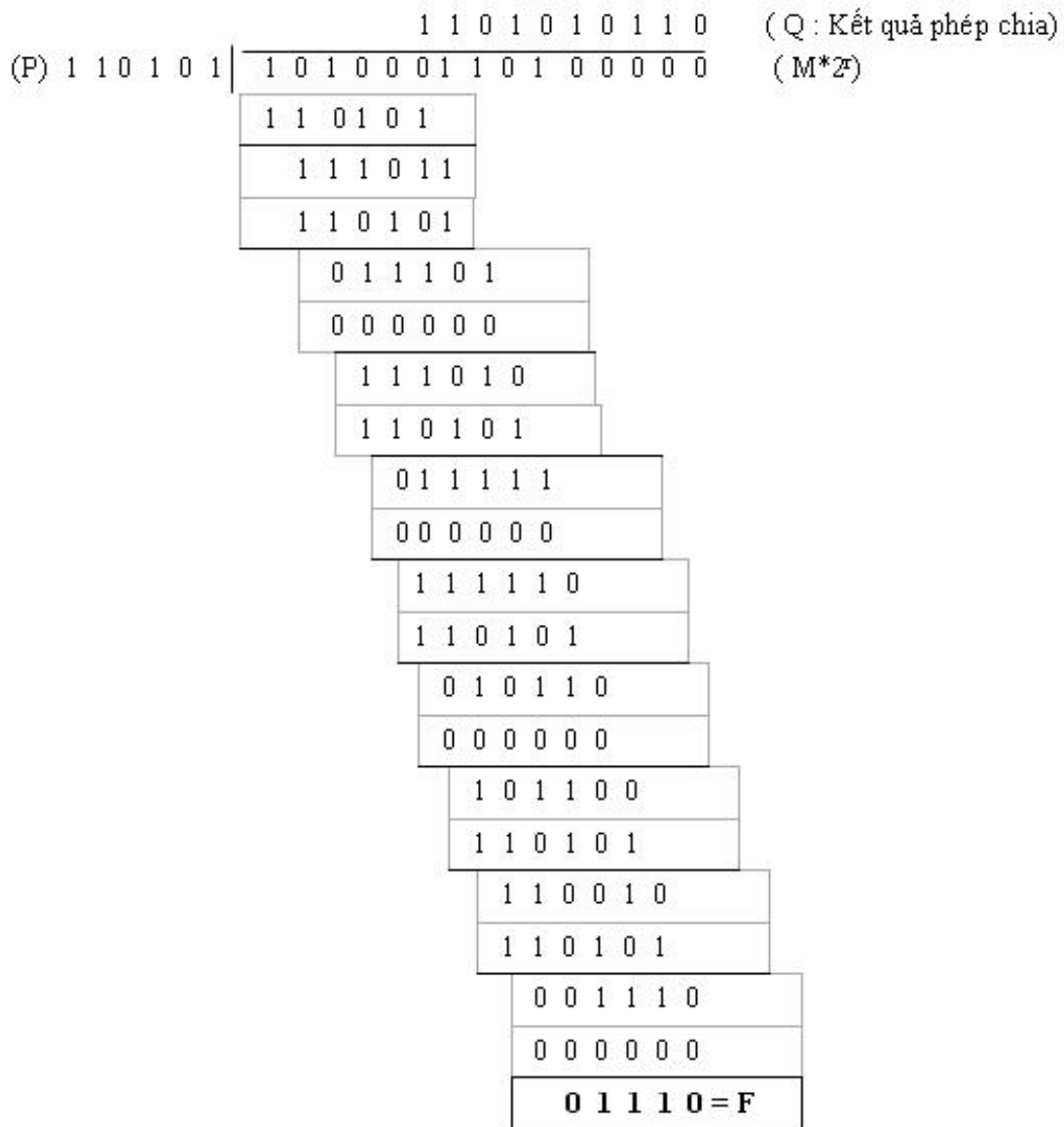
|   |      |   |        |
|---|------|---|--------|
|   | 1111 |   | 11001  |
| + | 1010 | x | 11     |
|   | 0101 |   | 11001  |
|   |      |   | 11001  |
|   |      |   | 101011 |

- Giả sử ta có:
  - M: Thông điệp k bit cần được gửi sang bên nhận.
  - F : Chuỗi kiểm tra khung FCS gồm r bit là thông tin điều khiển được gửi theo M để giúp bên nhận có thể phát hiện được lỗi.
  - $T = MF$  là khung (k + r) bit, được hình thành bằng cách nối M và F lại với nhau. T sẽ được truyền sang bên nhận, với  $r < k$
- Với M (k bit) , P (r+1 bit), F (r bit), T (k+r bit), thủ tục tiến hành để xác định checksum F và tạo khung truyền như sau:
  - Nối r bit 0 vào cuối M, hay thực hiện phép nhân M với  $2^r$
  - Dùng phép chia modulo 2 chia chuỗi bit  $M \cdot 2^r$  cho P.
  - Phần dư của phép chia sẽ được cộng với  $M \cdot 2^r$  tạo thành khung T truyền đi.
  - Trong đó P được chọn dài hơn F một bit, và cả hai bit cao nhất và thấp nhất phải là 1

## Vấn đề xử lý lỗi

Ví dụ:

- Giả sử ta có:
  - $M = 1010001101$  (10 bit)
  - $P = 110101$  (6 bit)
  - FCS cần phải tính toán ( 5 bit)
- Ta lần lượt thực hiện các bước sau:
  - Tính  $M \cdot 2^5 = 101000110100000$ .
  - Thực hiện phép chia modulo  $M \cdot 2^5$  cho  $P$  như hình dưới, ta được phần dư  $F = \mathbf{01110}$
  - Tạo khung gửi đi là  $T = M \cdot 2^r + F = 1010001101\mathbf{01110}$



Ngoài ra người ta còn có thể sử dụng phương pháp đa thức để biểu diễn phương pháp kiểm tra phần dư tuần hoàn. Trong phương pháp này người ta biểu diễn các chuỗi nhị

## Vấn đề xử lý lỗi

phân dưới dạng những đa thức của biến  $x$  với các hệ số nhị phân. Các hệ số tương ứng với các bit trong chuỗi nhị phân cần biểu diễn.

Giả sử ta có  $M=110011$  và  $P = 11001$ , khi đó  $M$  và  $P$  sẽ được biểu diễn lại bằng 2 đa thức sau:

$$M(x) = x^5 + x^4 + x + 1$$

$$P(x) = x^4 + x^3 + 1$$

Những phép toán trên đa thức vẫn là modulo 2. Quá trình tính CRC được mô tả dưới dạng các biểu thức sau:

## Vấn đề xử lý lỗi

$$M(x) = x^5 + x^4 + x + 1$$

$$P(x) = x^4 + x^3 + 1$$

Những phép toán trên đa thức vẫn là modulo 2. Quá trình tính CRC được mô tả dưới dạng các biểu thức sau:

$$1. \frac{X^n M(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

$$2. T(X) = X^n M(X) + R(X)$$

Các version thường được sử dụng của P là :

$$\text{CRC-12} = X^{12} + X^{11} + X^3 + X^2 + X + 1$$

$$\text{CRC-16} = X^{16} + X^{15} + X^2 + 1$$

$$\text{CRC-CCITT} = X^{16} + X^{12} + X^5 + 1$$

$$\begin{aligned} \text{CRC-32} &= X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} \\ &+ X^8 + X^7 + X^5 + X^4 + X^2 + X + 1 \end{aligned}$$

**Vi dụ:**

- Cho:  $M=1010001101$ ,  $P=110101$

- Ta có:  $r=5$

$$M(x) = x^9 + x^7 + x^3 + x^2 + 1$$

$$x^5 M(x) = x^{14} + x^{12} + x^8 + x^7 + x^5$$

$$P(x) = x^5 + x^4 + x^2 + 1$$

- Thực hiện phép toán:

$$\frac{x^5 M(x)}{P(x)} = Q(x) + \frac{F(x)}{P(x)}$$

$$\Rightarrow Q(x) = x^9 + x^8 + x^6 + x^4 + x^2 + x^1$$

$$F(x) = x^3 + x^2 + x^1 \langle \mathbf{01110} \rangle$$

$\Rightarrow$  Khung cần truyền đi là  $T = 1010001101\mathbf{01110}$