

*Ways of Error Control*

Lớp liên kết dữ liệu sử dụng các kỹ thuật kiểm soát lỗi đơn giản chỉ để đảm bảo và xác nhận rằng tất cả các khung dữ liệu hoặc gói tin, tức là chuỗi bit dữ liệu, được truyền từ bên gửi đến bên nhận với độ chính xác nhất định. Việc sử dụng hoặc cung cấp kiểm soát lỗi ở tầng liên kết dữ liệu là một quá trình tối ưu hóa, không bao giờ là một yêu cầu. Kiểm soát lỗi cơ bản là quá trình trong tầng liên kết dữ liệu để phát hiện hoặc xác định và gửi lại các khung dữ liệu có thể bị mất hoặc bị hỏng trong quá trình truyền. Trong cả hai trường hợp này, bên nhận hoặc đích không nhận được khung dữ liệu chính xác và bên gửi hoặc nguồn thậm chí không biết gì về bất kỳ sự mất mát nào liên quan đến các khung dữ liệu. Do đó, trong các trường hợp như vậy, cả bên gửi và bên nhận được cung cấp một số giao thức cần thiết để phát hiện hoặc xác định các loại lỗi như mất mát khung dữ liệu. Lớp Liên kết dữ liệu tuân theo một kỹ thuật được gọi là gửi lại các khung để phát hiện hoặc xác định các lỗi trung chuyển và cũng thực hiện các hành động cần thiết để giảm thiểu hoặc loại bỏ các lỗi đó. Mỗi khi phát hiện lỗi trong quá trình truyền, các khung dữ liệu cụ thể được gửi lại và quá trình này được gọi là ARQ (Yêu cầu Lặp Tự Động).

Có hai cách thực hiện Kiểm soát Lỗi như sau:

1. Phát hiện Lỗi: Phát hiện lỗi, như tên gọi, đơn giản là phát hiện hoặc xác định các lỗi. Những lỗi này có thể xảy ra do nhiễu hoặc bất kỳ ảnh hưởng nào khác trong quá trình truyền từ bộ phát đến bộ nhận, trong hệ thống truyền thông. Đây là một lớp các kỹ thuật để phát hiện dữ liệu hoặc tin nhắn bị méo mó hoặc biến dạng.

2. Sửa Lỗi: Sửa lỗi, như tên gọi, đơn giản là sửa chữa hoặc khắc phục các lỗi. Điều này đơn giản là việc tái tạo và phục hồi dữ liệu ban đầu không có lỗi. Nhưng phương pháp sửa lỗi rất tốn kém và rất khó khăn.

Các kỹ thuật khác nhau cho Kiểm soát Dòng:

1. ARQ Dừng và Chờ: ARQ Dừng và Chờ còn được gọi là giao thức bit xen kẽ. Đây là một trong những kỹ thuật hoặc cơ chế kiểm soát lỗi và dòng đơn giản nhất. Cơ chế này thường được yêu cầu trong viễn thông để truyền dữ liệu hoặc thông tin giữa hai thiết bị kết nối. Bên nhận đơn giản chỉ báo hiệu sẵn sàng nhận dữ liệu cho mỗi khung. Trong đó, bên gửi gửi thông tin hoặc gói dữ liệu cho bên nhận. Bên gửi sau đó dừng lại và đợi ACK (Xác nhận) từ bên nhận. Ngoài ra, nếu ACK không đến trong khoảng thời gian nhất định, tức là thời gian chờ, bên gửi sau đó gửi lại khung và đợi ACK. Nhưng nếu bên gửi nhận được ACK, thì nó sẽ truyền gói dữ liệu tiếp theo cho bên nhận và sau đó lại đợi ACK từ bên nhận. Quá trình dừng và chờ này tiếp tục cho đến khi bên gửi không có khung dữ liệu nào để gửi nữa.

2. ARQ Cửa trượt: Kỹ thuật này thường được sử dụng cho kiểm soát lỗi và truyền liên tục. Nó được phân loại thành hai loại như sau:

- ARQ Quay lại N : ARQ Quay lại N là dạng giao thức ARQ trong đó quá trình truyền tiếp tục gửi hoặc truyền tổng số khung được chỉ định bởi kích thước cửa sổ ngay cả khi không nhận được một gói ACK (Xác nhận) từ bên nhận. Nó sử dụng giao thức kiểm soát dòng cửa trượt. Nếu không có lỗi xảy ra, thì hoạt động là giống như cửa trượt.

- ARQ Lặp lại Lựa chọn : ARQ Lặp lại Lựa chọn cũng là một dạng giao thức ARQ trong đó chỉ các khung dữ liệu nghi ngờ hoặc bị hỏng hoặc bị mất được gửi lại. Kỹ thuật này tương tự như ARQ Quay lại N

**Kiểm soát lỗi trong mạng máy tính**

Khi gửi dữ liệu từ bên gửi đến bên nhận, có khả năng cao rằng dữ liệu có thể bị mất hoặc bị hỏng. Lỗi xảy ra khi dữ liệu của bên gửi không khớp với dữ liệu tại đầu nhận. Khi phát hiện lỗi, chúng ta cần phải gửi lại dữ liệu. Vì vậy, có các kỹ thuật kiểm soát lỗi khác nhau trong mạng máy tính. Trong bài viết này, chúng ta sẽ xem xét tất cả các kỹ thuật này. Vậy, hãy bắt đầu.

**Kiểm soát lỗi (Error control)**

**Kiểm soát lỗi đề cập đến cơ chế phát hiện và sửa lỗi xảy ra trong quá trình truyền khung/gói tin.**

**Dữ liệu được gửi dưới dạng một chuỗi các khung. Các khung đến theo đúng thứ tự mà chúng được gửi và mỗi khung được truyền phải chịu một độ trễ tùy ý và có thể thay đổi trước khi nhận.**

2 Kiểm soát lỗi trong tầng liên kết dữ liệu là quá trình phát hiện và gửi lại dữ liệu đã bị mất hoặc bị hỏng trong quá trình truyền dữ liệu. Bất kỳ hệ thống đáng tin cậy nào cũng phải có một cơ chế để phát hiện và sửa chữa các lỗi như vậy. Phát hiện và sửa chữa lỗi xảy ra ở cả tầng giao vận và tầng liên kết dữ liệu. Ở đây chúng tôi sẽ nói về tầng liên kết dữ liệu và kiểm tra từng bit xem có lỗi nào không.

**Trong quá trình truyền có thể xảy ra 2 loại lỗi:**

**Mất khung: Một khung không đến được phía bên kia. Một vụ nổ nhiễu có thể làm hỏng khung đến mức máy thu không biết rằng khung đã được truyền đi.**

**Khung bị hỏng: Một khung có thể nhận dạng được đã đến nhưng một số bit bị lỗi, tức là chúng đã bị thay đổi trong quá trình truyền.**

**2 Loại lỗi**

Lỗi một bit: Khi có một thay đổi trong chỉ một bit của dữ liệu của bên gửi thì được gọi là lỗi một bit.

Ví dụ: Nếu bên gửi gửi 101(5) cho bên nhận nhưng bên nhận nhận được 100(4) thì đó là lỗi một bit.101 (bit gửi) → 100 (bit nhận được)

Lỗi nổ: Khi có một thay đổi trong hai hoặc nhiều hơn hai bit của dữ liệu của bên gửi thì được gọi là lỗi nổ.

Ví dụ: Nếu bên gửi gửi 1011(11) cho bên nhận nhưng bên nhận nhận được 1000(8) thì đó là lỗi nổ 1011 (bit gửi) → 1000 (bit nhận được)

**3 Các loại lỗi:**Lỗi một bit: Lỗi một bit xảy ra khi chỉ một bit trong đơn vị dữ liệu thay đổi giá trị của nó trong quá trình truyền. Điều này có thể xảy ra do nhiều yếu tố khác nhau như tiếng ồn, nhiễu hoặc lỗi truyền tải. Lỗi bit đơn tương đối phổ biến nhưng thường có tác động tối thiểu đến tính toàn vẹn dữ liệu. Các kỹ thuật như kiểm tra tính chẵn lẻ hoặc tổng kiểm tra có thể giúp phát hiện và sửa các lỗi bit đơn.

Lỗi nhiều bit: Lỗi nhiều bit xảy ra khi có nhiều hơn một bit trong đơn vị dữ liệu thay đổi giá trị đồng thời trong quá trình truyền. Điều này có thể xảy ra do lỗi truyền tải nghiêm trọng hơn hoặc các yếu tố môi trường ảnh hưởng đến nhiều bit. Lỗi nhiều bit ít phổ biến hơn lỗi bit đơn nhưng có thể có tác động lớn hơn đến tính toàn vẹn dữ liệu. Việc phát hiện và sửa lỗi nhiều bit có thể yêu cầu các kỹ thuật phát hiện và sửa lỗi phức tạp hơn.

Lỗi chùm: Lỗi chùm đề cập đến một chuỗi các lỗi bit liên tiếp xảy ra trong một khoảng thời gian ngắn hoặc trên các bit liền kề trong luồng dữ liệu. Lỗi bùng nổ có thể xảy ra do các hiện tượng vật lý như suy giảm tín hiệu, bùng nổ nhiễu hoặc các nguồn nhiễu ảnh hưởng đến nhiều bit ở gần nhau. Lỗi bùng nổ có thể đặc biệt khó khăn để giảm thiểu vì chúng có thể dẫn đến hỏng dữ liệu đáng kể và ảnh hưởng đến độ tin cậy của hệ thống truyền thông.

**Các giai đoạn trong Kiểm soát lỗi**

Các kỹ thuật phổ biến nhất để kiểm soát lỗi dựa trên một số hoặc tất cả những điều sau đây:

Phát hiện lỗi: Sử dụng kiểm tra chẵn lẻ hoặc kiểm tra CRC.

Xác nhận tích cực: Đích trả về một xác nhận tích cực cho các khung được nhận thành công, không có lỗi.

Truyền lại sau khi hết thời gian chờ: Nguồn truyền lại khung chưa được xác nhận sau một khoảng thời gian định trước.

Xác nhận phủ định và truyền lại: Đích trả về xác nhận phủ định cho các khung trong đó phát hiện thấy lỗi. Nguồn truyền lại các khung như vậy.

Nói chung, các cơ chế này được gọi là Yêu cầu lặp lại tự động (ARQ).

Có 3 kỹ thuật ARQ phổ biến:

Dừng – và – chờ ARQ

Đi – quay lại – N ARQ

Chọn lọc – từ chối ARQ

**Có hai phương pháp kiểm soát lỗi chính:** phát hiện lỗi và sửa lỗi, bao gồm các kỹ thuật kiểm tra tính chẵn lẻ, tổng kiểm tra và dự phòng theo chu kỳ (CRC).

Các giao thức Yêu cầu lặp lại tự động (ARQ), bao gồm Dừng và chờ, Quay lại-N và Lặp lại có chọn lọc, thường được sử dụng để kiểm soát lỗi nhằm đảm bảo rằng dữ liệu bị mất hoặc bị hỏng sẽ được truyền lại khi cần.

**2 Phát hiện lỗi:** Đầu tiên, chúng ta cần phải phát hiện ở đầu nhận rằng dữ liệu nhận được có lỗi hay không.

Xác nhận: Nếu phát hiện có bất kỳ lỗi nào, bên nhận gửi một phản hồi tiêu cực (NACK) cho bên gửi.

Gửi lại: Khi bên gửi nhận được một phản hồi tiêu cực hoặc nếu không nhận được bất kỳ phản hồi nào từ bên nhận, bên gửi gửi lại dữ liệu một lần nữa.

**Các cách phát hiện lỗi**

Phát hiện lỗi: Phát hiện lỗi liên quan đến việc xác định sự hiện diện của lỗi trong dữ liệu được truyền. Các kỹ thuật phát hiện lỗi khác nhau, chẳng hạn như Tổng kiểm tra, kiểm tra dự phòng theo chu kỳ (CRC) và kiểm tra tính chẵn lẻ, được sử dụng để phát hiện lỗi bằng cách thêm thông tin dư thừa vào dữ liệu. Nếu dữ liệu nhận được không khớp với tổng kiểm tra hoặc tính chẵn lẻ dự kiến, điều đó cho thấy có lỗi.

Sửa lỗi: Kỹ thuật sửa lỗi được sử dụng để sửa các lỗi được phát hiện trong quá trình truyền dữ liệu. Sửa lỗi chuyển tiếp (FEC) và yêu cầu lặp lại tự động (ARQ) là các cơ chế sửa lỗi phổ biến được sử dụng trong mạng máy tính. FEC thêm tính dự phòng vào dữ liệu được truyền, cho phép người nhận xây dựng lại dữ liệu gốc ngay cả khi xảy ra lỗi. ARQ liên quan đến việc truyền lại các gói dữ liệu bị lỗi để đáp ứng yêu cầu từ người nhận.

**Phát hiện lỗi**

Kiểm tra dư lượng theo chiều dọc

Kiểm tra dư lượng theo chiều dọc

Kiểm tra dư lượng tròn

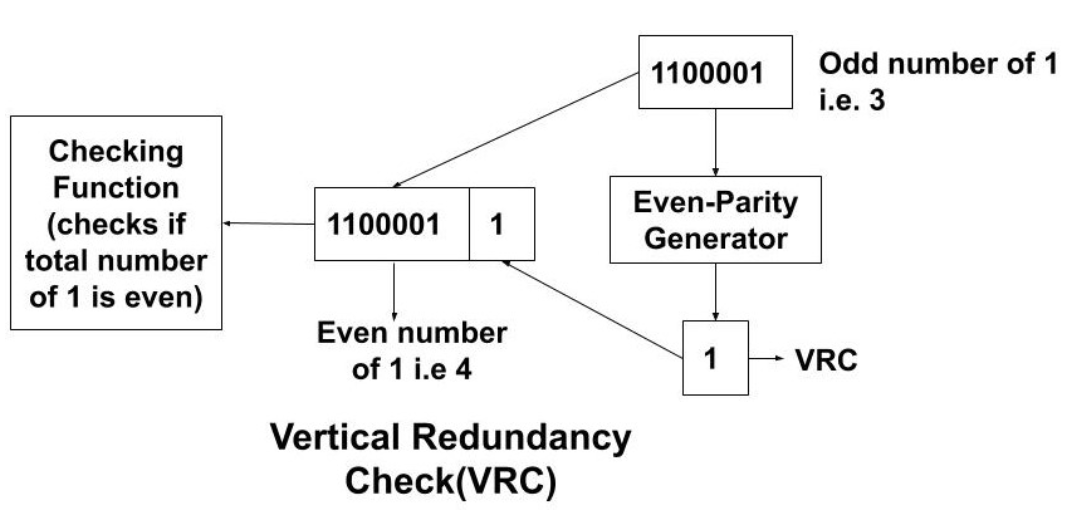
CheckSum

**Kiểm tra dư lượng theo chiều dọc (VRC)**

Trong phương pháp này, một bit dư lượng gọi là bit chẵn lẻ được thêm vào dữ liệu. Bit dư lượng này được thêm vào sao cho số lượng bit 1 trong dữ liệu là chẵn. Điều này gọi là kiểm tra chẵn lẻ. Nếu số lượng bit 1 là chẵn thì bit cần thêm vào là 0. Nếu số lượng bit 1 là lẻ thì bit cần thêm vào là 1.

Một số hệ thống cũng có thể kiểm tra số lượng bit 1 lẻ. Điều này được gọi là kiểm tra chẵn lẻ. Nếu số lượng bit 1 là lẻ thì bit cần thêm vào là 0. Nếu số lượng bit 1 là chẵn thì bit cần thêm vào là 1.

Ví dụ: Chúng ta có một dữ liệu 1100001. Bây giờ, dữ liệu này được gửi đến bộ tạo chẵn lẻ mà thêm một bit dư lượng vào dữ liệu bằng cách kiểm tra số lượng bit 1. Bộ tạo chẵn lẻ sẽ thêm vào 1 vì nó có số lượng bit 1 lẻ. Vì vậy, dữ liệu sẽ được truyền là dữ liệu gốc cùng với bit dư lượng tức là 11000011. Ở phía bên nhận, chúng ta có một hàm kiểm tra kiểm tra xem số lượng bit 1 có phải là chẵn hay không.



**Giới hạn của VRC**

Giả sử trong ví dụ trên, nếu vào thời gian truyền hai bit bị thay đổi sao cho bên nhận nhận được dữ liệu là 10000001. Bên nhận sẽ chấp nhận dữ liệu này một cách thành công. Điều này là do hàm kiểm tra sẽ kiểm tra xem số lượng bit 1 có phải là chẵn hay không và dữ liệu nhận được sẽ thỏa mãn điều kiện này.

Vì vậy, VRC sẽ thất bại khi có một số lượng thay đổi chẵn trong dữ liệu.

**Kiểm tra dư lượng theo chiều dọc (LRC)**

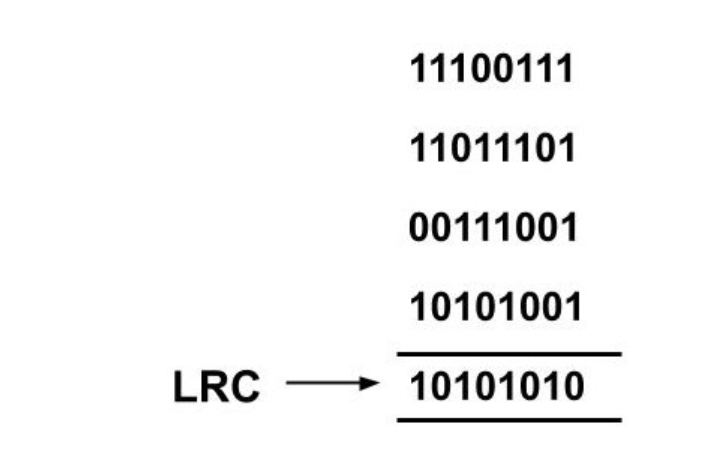
Trong LRC, chúng ta sử dụng một khối mã như bit chẵn lẻ. Chúng ta sẽ lấy mỗi khối dữ liệu và tính toán bit chẵn lẻ theo chiều dọc thay vì theo chiều dọc. Bên gửi sẽ gửi dữ liệu gốc cùng với khối bit chẵn lẻ được tạo ra. Điều này có thể được hiểu thông qua ví dụ dưới đây.

Ví dụ: Giả sử chúng ta cần gửi dữ liệu, 11100111 11011101 00111001 10101001. Vì vậy chúng ta sẽ tính toán bit chẵn lẻ bằng cách sử dụng kiểm tra chẵn lẻ. Chúng ta sẽ bắt đầu từ vị trí 0 của mỗi khối bit và dần dần di chuyển về các vị trí cao hơn. Chúng ta lấy tất cả các bit có mặt ở vị trí 0 tức là 1, 1, 1, 1. Bây giờ, đầu ra được quyết định theo hai quy tắc sau:

Nếu có số lượng bit 1 là chẵn thì đầu ra sẽ là 0.

Nếu có số lượng bit 1 là lẻ thì đầu ra sẽ là 1.

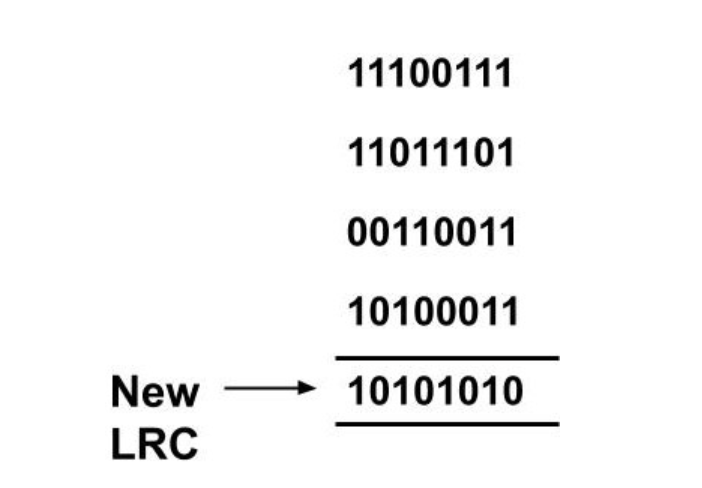
Vì có '4' bit 1 tức là một số chẵn nên đầu ra tại vị trí 0 sẽ là 0 và tiếp tục như vậy cho các vị trí cao hơn.



Dữ liệu được truyền sẽ là 11100111 11011101 00111001 10101001 10101010.

**Giới hạn của LRC**

Giả sử trong ví dụ trên, trong quá trình truyền dữ liệu, một số bit đã bị thay đổi và dữ liệu nhận được là 11100111 11011101 00110011 10100011. Nếu chúng ta tính toán bit chẵn lẻ cho dữ liệu nhận được này thì kết quả sẽ lại là 10101010 ở đầu nhận được.



Vậy nếu các bit dữ liệu đã thay đổi thì lỗi này sẽ không được phát hiện ở đầu nhận.

Vì vậy, nếu hai bit trong một đơn vị dữ liệu bị hỏng và hai bit ở cùng một vị trí chính xác trong **một đơn vị dữ liệu khác cũng bị hỏng, thì LRC sẽ không thể phát hiện ra.**

**Kiểm tra tổng**

Có hai phương pháp xử lý được sử dụng trong việc này. Bên gửi tạo ra kiểm tra tổng và gửi dữ liệu gốc cùng với kiểm tra tổng. Ở đầu nhận, cũng tạo ra kiểm tra tổng từ dữ liệu nhận được. Nếu tổng tạo ra ở phía bên nhận là tất cả các số không thì chỉ dữ liệu mới được chấp nhận.

**Bên gửi thực hiện các bước sau:**

1 Dữ liệu được chia thành k phần hoặc khối có 'n' bit được chỉ định.

2 Tất cả các phần được cộng lại bằng cách sử dụng bù 1 (dữ liệu).

3 Tổng cuối cùng được bù bit theo từng bit (chuyển đổi 0 thành 1 và 1 thành 0) để có được kiểm tra tổng.

4 Bên gửi gửi dữ liệu gốc cùng với kiểm tra tổng.

**Bên nhận thực hiện các bước sau:**

1 Dữ liệu được chia thành k phần hoặc khối có 'n' bit được chỉ định.

2 Tất cả các phần được cộng lại bằng cách sử dụng bù 1 (dữ liệu + kiểm tra tổng).

3 Tổng cuối cùng được bù bit theo từng bit.

4 Nếu kết quả là tất cả các số không thì nó được chấp nhận, nếu không thì bị từ chối.

**Ví dụ:** Chúng ta cần gửi dữ liệu được chia thành bốn phần mỗi phần có 8 bit. Giả sử chúng ta cần gửi 10110011 10101011 01011010 11010101. Vì vậy, có tổng cộng 32 bit và chúng ta sẽ chia toàn bộ dữ liệu 32 bit thành một nhóm 8 bit tức là 4 nhóm.

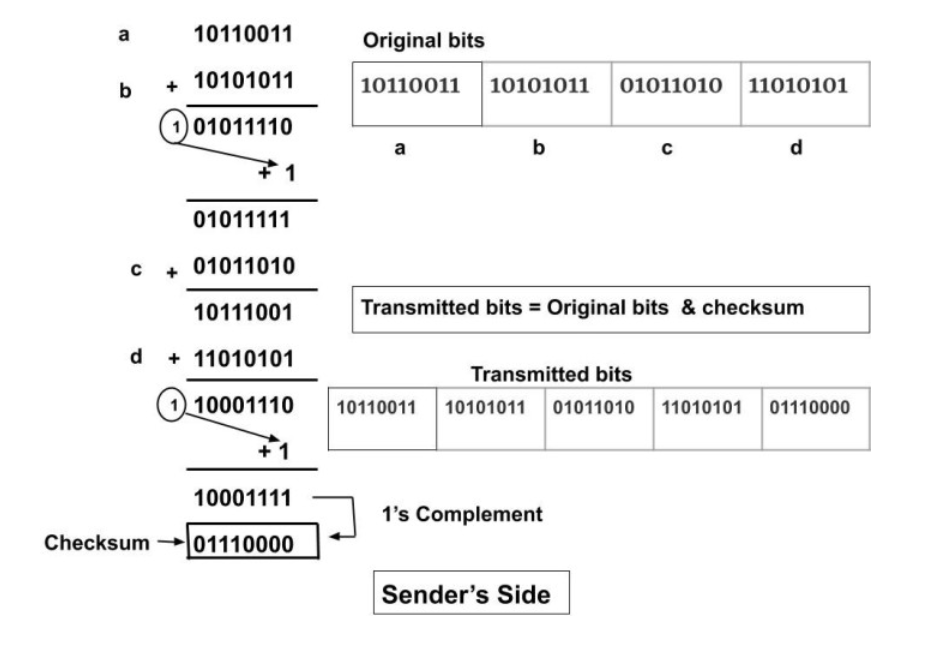
**Ở phía bên gửi**

1 Lấy bất kỳ hai khối dữ liệu nào và thực hiện phép tính bù 1.

2 Lấy khối dữ liệu tiếp theo và cộng nó vào kết quả của phép cộng trước đó. Thực hiện phép cộng cho đến khi tất cả các khối đều đã được cộng.

3 Lấy bù 1 của tổng cuối cùng nhận được. Đó là kiểm tra tổng.

4 Truyền các bit gốc cùng với kiểm tra tổng đến phía bên nhận.

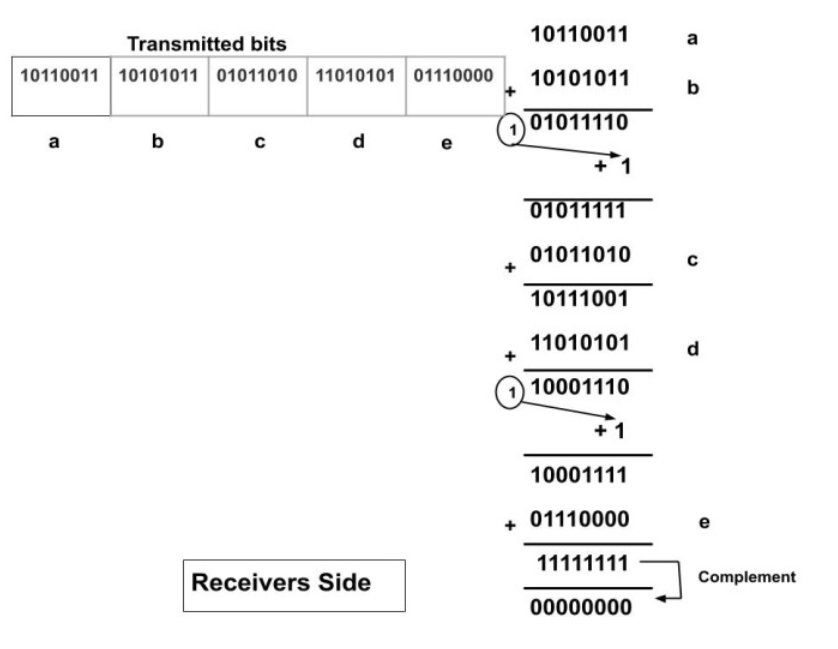


**Ở phía bên nhận**

1 Chọn hai khối dữ liệu bất kỳ và thực hiện phép tính bù 1.

2 Lấy khối dữ liệu tiếp theo và cộng nó vào kết quả của phép cộng trước đó. Thực hiện phép cộng cho đến khi tất cả các khối đều đã được cộng.

3 Lấy bù 1 của tổng cuối cùng nhận được. Nếu bù đều là tất cả các số không thì chỉ dữ liệu mới được chấp nhận. Ở đây, kết quả là tất cả các số không nên bên nhận chấp nhận dữ liệu.

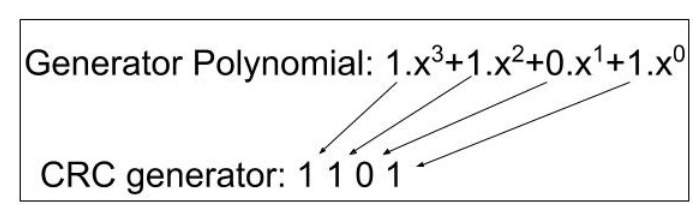


Vì kết quả là 00000000 nên dữ liệu nhận được là chính xác.

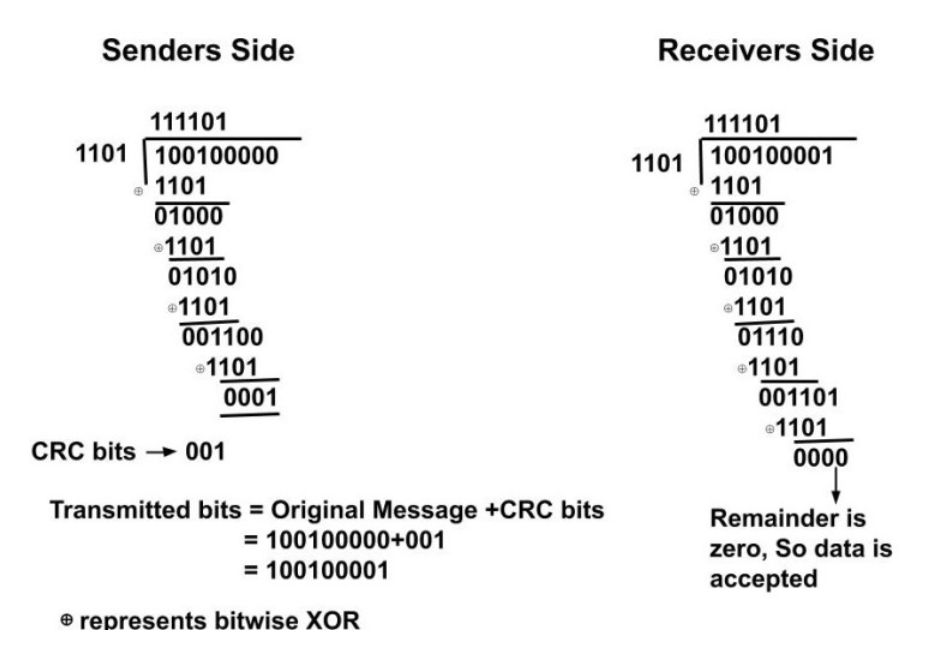
**Kiểm tra dư vòng lặp (CRC)**

Kiểm tra dư vòng lặp dựa trên phép chia nhị phân. Người gửi và người nhận đều đồng ý về một đa thức sinh. Một đa thức sinh có thể là bất kỳ biểu thức đa thức nào. Đa thức sinh này giúp tìm ra đa thức sinh CRC. Bit CRC (có cùng số bit với đa thức sinh CRC) được tạo ra ở phía người gửi bằng cách chia chuỗi bit của dữ liệu cho đa thức sinh CRC. Trước khi chia chuỗi bit cho đa thức sinh CRC, chúng ta sẽ thêm dữ liệu gốc với n-1 bit 0 giả sử rằng đa thức sinh CRC có n bit. Người gửi sau đó sẽ thêm dữ liệu với bit CRC và gửi nó cho người nhận.

Ví dụ: Cho bất kỳ đa thức sinh nào, chúng ta có thể tìm ra đa thức sinh CRC bằng cách lấy hệ số của mỗi biến.



Ở đầu nhận, các bit nhận được được chia lại bởi đa thức sinh CRC. Nếu phần dư của phép chia là không thì dữ liệu được chấp nhận, ngược lại bị từ chối.



**Tái truyền**

Khi phát hiện bất kỳ lỗi nào, các khung được chỉ định sẽ được gửi lại, quá trình này được gọi là yêu cầu lặp tự động (ARQ). Kiểm soát lỗi ở tầng liên kết dữ liệu dựa trên ARQ.

Các kỹ thuật kiểm soát lỗi sau có thể được sử dụng sau khi lỗi được phát hiện.

1 Dừng và chờ ARQ

2 Trượt cửa sổ ARQ

**Dừng và Chờ ARQ**

Một bộ đếm thời gian chờ được duy trì ở phía người gửi. Trước tiên, nếu người gửi không nhận được sự xác nhận của dữ liệu đã gửi trong khoảng thời gian cho trước thì người gửi sẽ giả định rằng dữ liệu đã gửi đã bị mất hoặc sự xác nhận của dữ liệu đã bị mất. Vì vậy, người gửi sẽ gửi lại dữ liệu cho người nhận. Thứ hai, nếu người nhận phát hiện lỗi trong khung dữ liệu chỉ ra rằng nó đã bị hỏng trong quá trình truyền, người nhận sẽ gửi một NACK (phản hồi phủ định). Nếu người gửi nhận được một phản hồi phủ định của dữ liệu thì nó sẽ gửi lại dữ liệu.

**Trượt cửa sổ ARQ**

Trong ARQ trượt cửa sổ, một người gửi có thể gửi nhiều khung dữ liệu cùng một lúc. Người gửi giữ một bản ghi của tất cả các khung đã gửi cho đến khi chúng được xác nhận. Người nhận có thể gửi một ACK (xác nhận) hoặc NACK (phản hồi phủ định) tùy thuộc vào việc khung dữ liệu được nhận đúng cách, nếu có bất kỳ lỗi nào được phát hiện hoặc đã bị mất.

*ARQ trượt cửa sổ được chia thành hai loại:*

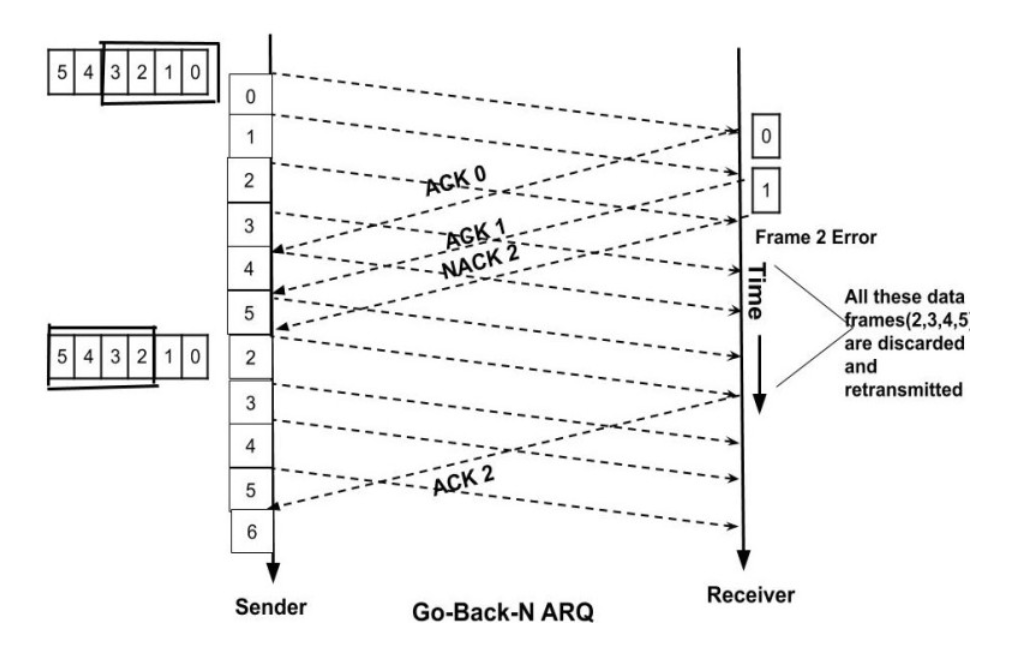
1 Go-Back-N ARQ

2 ARQ Lặp Lựa Chọn

**Go-Back-N ARQ**

Trong giao thức này, nếu bất kỳ khung nào bị mất hoặc bị hỏng thì tất cả các khung kể từ khung cuối cùng được xác nhận sẽ được gửi lại một lần nữa. Kích thước cửa sổ của người gửi là N nhưng kích thước cửa sổ của người nhận chỉ là một.

Ví dụ: Giả sử chúng ta có một kích thước cửa sổ là 4 cho các khung dữ liệu mà chúng ta sẽ gửi. Bây giờ, giả sử trong khi gửi khung dữ liệu 2 có một số lỗi xảy ra và nó đã bị hỏng. Vì vậy, người nhận sẽ gửi một phản hồi phủ định (NACK) của dữ liệu. Tất cả các khung dữ liệu sau các khung đã được xác nhận cuối cùng (ACK) tức là sau khung 1 sẽ được gửi lại.



**Hạn chế của Go-Back-N ARQ**

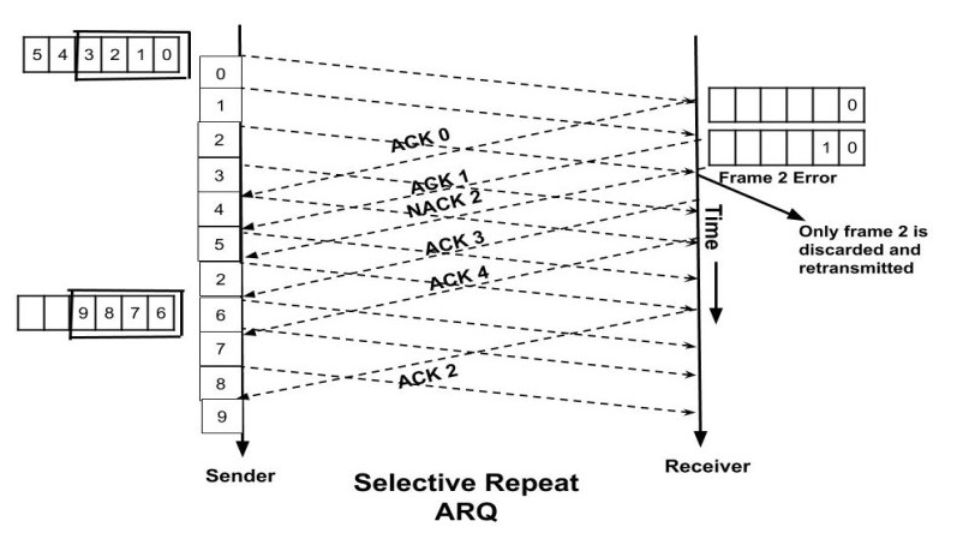
Trong phương pháp này, chúng ta phải gửi lại tất cả các khung một lần nữa ngay cả khi chúng không có lỗi. Trong ví dụ trên, chúng ta đã phải gửi lại tất cả các khung, tức là 2, 3, 4, 5 một lần nữa mặc dù lỗi chỉ xuất hiện trong khung 2. Làm thế nào để vượt qua điều này?

**ARQ Lặp Lựa Chọn**

Trong ARQ này, nếu bất kỳ khung nào bị mất hoặc bị hỏng thì chỉ có khung đó được gửi lại, khung đó có ACK phủ định. Kích thước cửa sổ của người gửi và kích thước cửa sổ của người nhận là như nhau ở đây.

Nó loại bỏ vấn đề của Go-Back-N ARQ vì các khung không có lỗi có thể được chấp nhận khi kích thước cửa sổ của người nhận bằng kích thước của người gửi, khác với Go-Back-N ARQ nơi kích thước cửa sổ của người nhận chỉ là 1. Phương pháp truyền lại được điều chỉnh sao cho chỉ có các khung cá nhân được gửi lại.

Ví dụ: Trong ví dụ trên, nếu có lỗi trong khung 2. Vì vậy, chúng ta sẽ chỉ gửi lại khung số 2.



Tài liệu tìm thêm được có thể bổ sung tùy thích nha

Theo dõi và giám sát lỗi

Nhóm xác định nguyên nhân gốc rễ không nhất thiết là nhóm sẽ định vị và áp dụng giải pháp. Bước đầu tiên của Kiểm soát Lỗi là quan trọng để chuyển giao lỗi một cách trơn tru. Nhóm theo dõi lỗi giữ các nguồn trên ma trận báo cáo chính được cập nhật khi cần thiết.

**Các bước trong việc theo dõi và giám sát lỗi là:**

1. Lấy tình trạng lỗi.

2. Đánh giá lại tình trạng lỗi.

3. Cập nhật ma trận báo cáo chính về tình trạng lỗi.

4. Theo dõi bất kỳ dữ liệu liên quan nào và đảm bảo rằng các bản ghi về vấn đề, lỗi đã biết, sự cố, và thay đổi đều được cập nhật.

5. Dịch vụ quản lý đăng tải cập nhật lên các bản ghi bị ảnh hưởng tự động.

**1**. [**Flow Control**](https://www.geeksforgeeks.org/flow-control-in-data-link-layer/)**:**  **(Kiểm soát dòng chảy):**

Đây là một chức năng quan trọng của Lớp liên kết dữ liệu. Nó đề cập đến một tập hợp các thủ tục cho người gửi biết lượng dữ liệu có thể truyền trước khi chờ xác nhận từ người nhận.

**Mục đích của việc kiểm soát dòng chảy:**

Bất kỳ thiết bị nhận nào cũng có tốc độ giới hạn để có thể xử lý dữ liệu đến và cũng có một lượng bộ nhớ hạn chế để lưu trữ dữ liệu đến. Nếu nguồn đang gửi dữ liệu với tốc độ nhanh hơn khả năng của máy thu thì có khả năng máy thu bị quá tải. Người nhận sẽ tiếp tục mất một số khung hình chỉ vì chúng đến quá nhanh và bộ đệm cũng bị đầy.

Điều này sẽ tạo ra các khung lãng phí trên mạng. Do đó, thiết bị nhận phải có cơ chế nào đó để thông báo cho người gửi gửi ít khung hơn hoặc tạm thời dừng truyền. Bằng cách này, điều khiển luồng sẽ kiểm soát tốc độ truyền khung đến một giá trị mà bộ thu có thể xử lý được.

So sánh **Flow control và Error control**

|  |  |  |  |
| --- | --- | --- | --- |
| Số thứ tự | **Flow control** |  | **Error control** |
| 1 | Kiểm soát luồng chỉ nhằm mục đích truyền dữ liệu từ người gửi đến người nhận. |  | Kiểm soát lỗi nhằm mục đích truyền dữ liệu không có lỗi từ người gửi đến người nhận. |
| 2 | Đối với Kiểm soát luồng, có hai cách tiếp cận: Kiểm soát luồng dựa trên phản hồi và Kiểm soát luồng dựa trên tỷ lệ. |  | Để phát hiện lỗi trong dữ liệu, các phương pháp tiếp cận là: Kiểm tra tổng([Checksum](https://www.geeksforgeeks.org/error-detection-in-computer-networks/)), Kiểm tra dự phòng theo chu kỳ([Cyclic Redundancy](https://www.geeksforgeeks.org/modulo-2-binary-division/) [Check](https://www.geeksforgeeks.org/modulo-2-binary-division/)) và Kiểm tra tính chẵn lẻ([Parity Checking](https://www.geeksforgeeks.org/vertical-redundancy-check-vrc-or-parity-%20check/). ).  Để sửa lỗi dữ liệu, các phương pháp tiếp cận là: Mã Hamming([Hamming code](https://www.geeksforgeeks.org/hamming-code-in-computer-network/)), Mã chập nhị phân(Binary Convolution codes), Mã Reed-Solomon, Mã kiểm tra chẵn lẻ mật độ thấp(Low-Density Parity Check codes). |
| 3 | Nó ngăn ngừa mất dữ liệu và tránh chạy quá mức bộ đệm nhận. |  | Nó được sử dụng để phát hiện và sửa lỗi xảy ra trong code |
| 4 | Ví dụ về các kỹ thuật Kiểm soát luồng là: Giao thức Dừng & Chờ và Giao thức Cửa sổ trượt. |  | Ví dụ về các kỹ thuật Kiểm soát lỗi là: ARQ dừng & chờ và ARQ cửa sổ trượt (ARQ quay lại-N, ARQ lặp lại đã chọn). |

**Tầm quan trọng của kiểm soát lỗi:**Kiểm soát lỗi là một thuật ngữ công nghệ quan trọng vì nó đề cập đến các phương pháp và quy trình phát hiện và sửa lỗi xảy ra trong quá trình truyền và xử lý dữ liệu.

Những lỗi này có thể ảnh hưởng đến tính chính xác, độ tin cậy và tính toàn vẹn của thông tin được truyền hoặc lưu trữ.

Bằng cách triển khai các kỹ thuật kiểm soát lỗi, chẳng hạn như phương pháp phát hiện lỗi và sửa lỗi, hệ thống có thể giảm thiểu hỏng dữ liệu, đảm bảo hoạt động trơn tru của các hệ thống và mạng truyền thông khác nhau cũng như duy trì chất lượng dịch vụ tổng thể.

Do đó, kiểm soát lỗi đóng một vai trò quan trọng trong việc duy trì hiệu quả và độ tin cậy của hệ thống kỹ thuật số, khiến nó trở thành một khía cạnh quan trọng của công nghệ và quản lý dữ liệu.