Truyền Số Liệu
Điều Khiển Liên Kết

Giảng viên: Ths. Vũ Mạnh Khánh

Một số khái niệm

- **Đồng bộ frame (Frame synchronization):** Dữ liệu được gửi trong các khối gọi là các frame. Điểm bắt đầu và kết thúc của frame phải nhận biết được.
- **Điều khiển lưu lượng (Flow control):** trạm phát không được gửi các frame với tốc độ nhanh hơn khả năng nhận của trạm thu.
- **Điều khiển khắc phục lỗi (Error control):** các bit lỗi được phát sinh bởi hệ thống truyền dẫn có thể được khôi phục.

Một số khái niệm

- Gán địa chỉ (Addressing): Trên một đường truyền nhiều điểm (như mạng LAN) thì cần thiết phải định danh các trạm trong quá trình truyền.
- Dữ liệu và việc điều khiển trên cùng liên kết (Control and data on same link): không cần phải có một đường truyền khác biệt cho việc điều khiển thông tin mà chỉ cần trạm thu phải có khả năng phân biệt việc điều khiển đó từ dòng dữ liệu nhận được.
- Quản lý liên kết (Link management): việc khởi tạo, duy trì và kết thúc của quá trình trao đổi dữ liệu đòi hỏi sự sắp xếp và phối hợp giữa các trạm làm việc một cách thích hợp.

I. Điều khiển dòng

- Điều khiển lưu lượng là cơ chế điều khiển nhằm đảm bảo thực thể truyền dữ liệu không làm thực thể nhận bị tràn bộ nhớ đệm.

Stop and Wait

- Trạm nguồn truyền một frame
- Sau khi trạm đích nhận được frame nó sẽ thông báo sẵn sàng nhận frame tiếp theo bằng việc gửi lại một tín hiệu báo nhận rằng đã nhận được frame
- Trạm nguồn sẽ đợi cho đến khi nó nhận được tín hiệu báo nhận trước khi gửi frame tiếp theo.

Stop and Wait

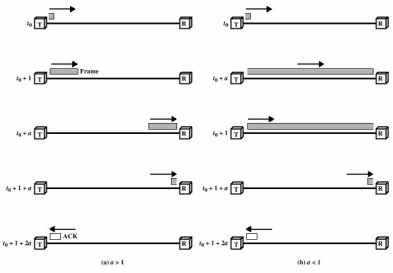


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

Stop and Wait

- Ưu điểm
  - Thủ tục này làm việc khá tốt với số ít các frame có kích cỡ lớn. Trong trường hợp khối dữ liệu lớn cần truyền thì trạm nguồn sẽ chia khối dữ liệu thành các khối nhỏ hơn và truyền bằng các frame.
    - Kích cỡ bộ nhớ đệm của trạm thu có thể được giới hạn
    - Quá trình truyền lâu hơn nên có thể phát sinh lỗi và cần phải truyền lại frame, tuy nhiên do các frame nhỏ nên lỗi có thể phát hiện sớm hơn và lượng dữ liệu truyền lại là nhỏ hơn.
    - Với môi trường chia sẻ (như mạng LAN), nó sẽ không cho phép một trạm chiếm giữ môi trường truyền với thời gian kéo dài, vì sẽ phát sinh thời gian trễ lâu cho việc gửi thông tin của các trạm khác. Tức cơ chế này sẽ ngăn chặn được việc một trạm chiếm dụng đường truyền lâu.

Stop and Wait

- Nhược điểm
  - Với một thông điệp sử dụng nhiều frame thì thủ tục của Stop-and-Wait có thể không đầy đủ. Bản chất của vấn đề là chỉ 1 frame tại một thời điểm được truyền.
  - Trường hợp mà độ dài của liên kết (là số lượng bit lấp đầy trong liên kết) lớn hơn độ dài của frame thì thật sự kết quả không có hiệu quả.

Sliding-Window Flow Control

- Hai trạm A và B được kết nối theo liên kết truyền song công. Trạm B cấp phát bộ nhớ đệm và nhận frame W, còn A gửi các frame W nhưng không cần đợi tín hiệu báo nhận (ACK).
- Để theo dõi các frame vừa báo nhận, mỗi frame được gán nhãn với một số tuần tự.
  - B báo nhận frame bằng việc gửi một tín hiệu báo nhận chứa số tuần tự của frame sẽ nhận tiếp.
  - Cơ chế này cũng có thể dùng để báo nhận nhiều frame. Ví dụ: B nhận frame 2,3 và 4 nhưng không báo
  - Trong khi đó, A phải duy trì một danh sách số tuần tự các frame mà nó đã gửi, và B duy trì một danh sách các số tuần tự mà nó đã sẵn sàng nhận.
  - Mỗi danh sách này có thể truyền qua của sổ các frame. Thao tác trên được hiểu là có chế điều khiển lưu lượng cửa sổ trượt.

Sliding-Window Flow Control

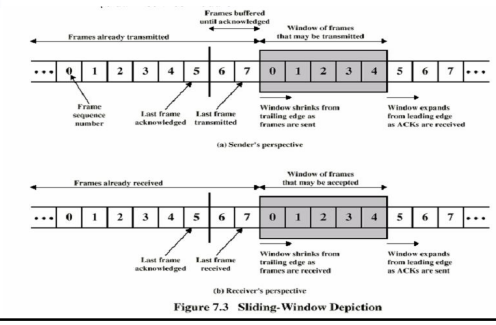


Figure 7.3 Sliding-Window Depiction

Sliding-Window Flow Control

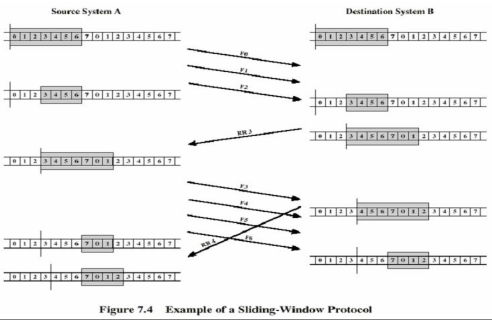


Figure 7.4 Example of a Sliding-Window Protocol

Sliding-Window Flow Control

- Ưu điểm
  - Điều khiển lưu lượng cửa sổ trượt có khả năng hữu ích hơn điều khiển lưu lượng Stop-and-Wait. Vì nó đã xử lý như một đường ống (pipeline) để có thể lấp đầy frame trên đường truyền. Điều này hoàn toàn tương phản với Stop-and-Wait là tại một thời điểm chỉ có 1 frame được truyền trong ống trên đường truyền

II. Dò tìm lỗi

- Quá trình truyền vẫn phát sinh lỗi bất chấp sự thiết kế của các hệ thống truyền dẫn, đó là sự thay đổi một hoặc nhiều bit của frame truyền.

II. Dò tìm lỗi

P_b: Xác suất lỗi một bit đơn, là tỉ lệ bit lỗi (BER – bit error rate).
P_1: Xác suất để 1 frame đến đích mà không bị lỗi
P_2: Xác suất để 1 frame đến đích với 1 hoặc nhiều lỗi không được phát hiện
P_3: Xác suất để 1 frame đến đích với 1 hoặc nhiều lỗi đươch phát hiện, không có lỗi nào không được phát hiện

$$P_1 = (1 - P_b)^F$$

$$P_2 = 1 - P_1$$

Trong đó, F là số bit của mỗi frame

II. Dò tìm lỗi

- Các loại lỗi trong truyền thông
  - Lỗi đơn bit : Trong một đơn vị dữ liệu có một bit bị hỏng
  - Lỗi đa bit : Trong một đơn vị dữ liệu có hai hay nhiều bít bị hỏng
  - Lỗi bit chùm (đảo bit) : Trong một đơn vị dữ liệu có hai hay nhiều bit liên tiếp bị lỗi

Kiểm tra chẵn lẻ(parity bit)

- Thêm một bit chẵn lẻ vào phần cuối của khối dữ liệu.
- Giá trị của bit này được chọn lựa sao cho ký tự có các bit 1 là số chẵn (even parity) hoặc các bit 1 là số lẻ (odd parity).

Kiểm tra chẵn lẻ(parity bit)

- Vd
  - Dữ liệu: 0111010101
  - Thêm bit chẵn lẻ: 0111010101 0
  - Bit chẵn lẻ thêm vào đảm bảo số bit 1 là một số chẵn
- Bên nhận kiểm tra
  - Nếu có số chẵn các bit 1 thì dịch là không có lỗi
  - Nếu là một số lẻ các bít thì có nghĩa là chắc chắn có một lỗi xảy ra. Một lỗi đơn (hoặc một số lẻ của lỗi) có thể được dò tìm. Một số chẵn các lỗi không được dò tìm.

Kiểm tra chẵn lẻ kép

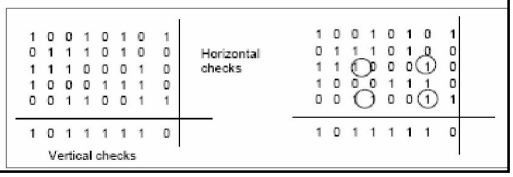
k Data bits	r Check bits

Mỗi lần kiểm tra chẵn lẻ là một modulo tính tổng của một số bít dữ liệu

Example:

$c_1 = x_1 + x_2 + x_3$
$c_2 = x_2 + x_3 + x_4$
$c_3 = x_1 + x_2 + x_4$

Horizontal and Vertical Parity Chẵn lẻ ngang và chẵn lẻ dọc



Mã dư vòng (CRC - Cyclic Redundancy Check)

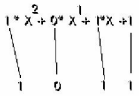
- Bên phát :Lấy toàn bộ dữ liệu được truyền đem chia cho một số chia, phần dư của phép tính này chính là CRC, CRC được đưa vào cuối dữ liệu
- Bên nhận: Lấy toàn bộ dữ liệu nhận được (CRC + DL) đem chia cho một số chia như là bên gửi, kết quả của phép chia này cho biết dữ liệu đúng hay sai
  - Không dư : Dữ liệu nhận đúng
  - Dư : Dữ liệu nhận sai

Mã dư vòng (CRC - Cyclic Redundancy Check)

- Cách lấy số chia: Dùng một số đa thức sinh

CRC-12 $= X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC-16 $= X^{16} + X^{15} + X^2 + 1$
CRC-CCITT $= X^{16} + X^{12} + X^5 + 1$
CRC-32
$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + x + 1$

Vd ta có đa thức sinh : $X^3 + X + 1$

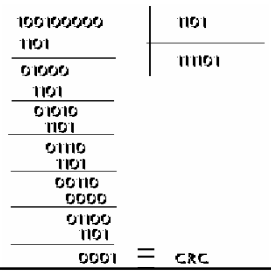


Như vậy đa thức bậc 3 thì số chia có 4 bậc => đa thức bậc n thì số chia có n+1 bậc

Mã dư vòng (CRC - Cyclic Redundancy Check)

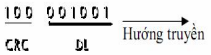
- Cách thực hiện phép chia

Để chia dữ liệu cho một số ta thực hiện như sau: Nếu số chia có n bậc ta thêm vào bên phải dữ liệu n-1 số 0 và thực hiện phép chia nhị phân. Số dư thu được từ phép chia này là CRC sẽ thay thế vào n-1 số 0.
Ví dụ : ta có dữ liệu là 100100
số chia là 1101-> n=4



Mã dư vòng (CRC - Cyclic Redundancy Check)

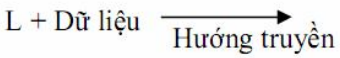
Dữ liệu sẽ gửi đi:



Ưu điểm :
Phương pháp CRC bảo đảm truyền dữ liệu có độ tin cậy cao và nếu sử dụng CRC 32 bảo đảm độ tin cậy rất cao
Nhược điểm : Số bit dư thừa lớn

Check Sum

- Nơi gửi:Người ta chia dữ liệu ra làm n đoạn và cộng những đoạn ấy với nhau theo phương pháp số học bù 1 được tổng là T. Lấy nghịch đảo của T ta được một số L chính là check sum. L được đưa vào cuối đơn vị dữ liệu và gửi đi đến nơi nhận



Check Sum

- Nơi nhận: nhận được L+ dữ liệu cũng thực hiện cộng theo phương pháp số học bù 1 các dữ liệu nhận được . Nếu tổng các bậc đều có giá trị “1” thì giá trị không bị lỗi

Check Sum(ví dụ)

1 00111 01011 10101 01101

Trong trường hợp chia thừa ta có thể thêm các số 0 vào đằng trước cho đủ

Thực hiện cộng :

```
 01101 +
 10101
 100010
     1 +
 -------
 00011 +
 01011
 -------
 01110 +
 00111
 -------
 10101   = T
```

Nghịch đảo = 0 1 0 1 0 = L = Check Sum

Gửi đi :

01010 1 00111 01011 10101 01101 Hướng truyền

Check Sum

Check Sum

Check Sum(ví dụ)

Bên nhận dữ liệu :

```
            T= 10101
CHECK SUM   L= 01010
             -------
               11111
```

⇨ dữ liệu không bị lỗi

III. Sửa lỗi

- Việc phát hiện lỗi và sửa lỗi đều đòi hỏi phải đưa thêm vào các bit dư thừa, về nguyên tắc người ta đều có thể phát hiện và sửa những bit bị lỗi.
- Việc sửa phải đòi hỏi phải biết được vị trí của bit lỗi
- Việc sửa là đảo ngược giá trị của bit đó
- Về nguyên tắc : tự động phát hiện lỗi và sửa lỗi.

Phương pháp Hamming

Các bit dư thừa được đưa vào với mục đích để phát hiện lỗi và sửa lỗi. Nếu bit dư thừa được đưa vào là **r** và dữ liệu là M bit thì phải chọn **r** sao cho

$2^r > M + r + 1$

Thì mới bao quát hết mọi khả năng xảy ra lỗi.

Ví dụ : Với M = 7 thì cần bao nhiêu bit dư thừa

- Nếu r = 1 => $2^1 = 2$ mà M + r + 1 = 7 + 1 + 1 = 9 => không thỏa mãn
- Nếu r = 2 => $2^2 = 4$ mà M + r + 1 = 7 + 2 + 1 = 10 => không thỏa mãn
- Nếu r = 3 => $2^3 = 8$ mà M + r + 1 = 7 + 3 + 1 = 11 => không thỏa mãn
- Nếu r = 4 => $2^4 = 16$ mà M + r + 1 = 7 + 4 + 1 = 12 => thỏa mãn

Vậy với r= 4 là thỏa mãn

Phương pháp Hamming

Giả sử với dữ liệu của mã Asscii có 7 bit dữ liệu thì cần 4 bit dư thừa vậy đặt các bit dữ liệu vào vị trí nào trong dữ liệu

11	10	9	8	7	6	5	4	3	2	1
D6	D5	D4	R8	D3	D2	D1	R4	D0	R1	R0

Ví dụ dữ liệu cần gửi đi là 1101011

11	10	9	8	7	6	5	4	3	2	1
1	1	0	R8	1	0	1	R4	1	R1	R0

Để gán các giá trị cho R0 ,R1,R4,R8 trên cơ sở như sau:

Phương pháp Hamming

R0 : có mặt trên các vị trí bit 1,3,5 ,7,9 ,11
R1 : có mặt trên các vị trí bit 2,3,6 ,7,10 ,11
R4 : có mặt trên các vị trí bit 4,5,6 ,7
R8 : có mặt trên các vị trí bit 8,9,10 ,11

Xác định giá trị

R0:
1 : 2^0 =1
3 : $2^0 + 2^1 = 3$
5 : $2^0 + 2^2 = 5$
7 : $2^0 + 2^1 + 2^2 = 7$
9 : $2^0 + 2^3 = 9$
11 : $2^0 + 2^1 + 2^3 = 11$

R1:
2 : 2^1 =2
3 : $2^0 + 2^1 = 3$
6 : $2^1 + 2^2 = 6$
7 : $2^0 + 2^1 + 2^2 = 7$
10 : $2^1 + 2^3 = 10$
11 : $2^0 + 2^1 + 2^3 = 11$

R4:
4 : 2^2 =4
6 : $2^1 + 2^2 = 6$
5 : $2^0 + 2^2 = 5$
7 : $2^0 + 2^1 + 2^2 = 7$

R8:
8 : $2^3 = 8$
9 : $2^0 + 2^3 = 9$
10 : $2^1 + 2^3 = 10$
11 : $2^0 + 2^1 + 2^3 = 11$

Phương pháp Hamming

R0 có tổng "1" = 6 -> chẵn -> R0 = 0
R1 có tổng "1" = 6 -> chẵn -> R1 = 0
R4 có tổng "1" = 4 -> chẵn -> R4 = 0
R8 có tổng "1" = 4 -> chẵn -> R8 = 0

Dữ liệu gửi đi

11	10	9	8	7	6	5	4	3	2	1
1	1	0	0	1	0	1	0	1	0	0

Phương pháp Hamming

Nơi nhận

11	10	9	8	7	6	5	4	3	2	1
0	1	0	0	1	0	1	0	1	0	0

R0 = 1
R1= 1
R 4 = 0
R8 = 1
Dùng biện pháp tổ hợp cho thấy bit 11 bị lỗi

IV. Điều khiển lỗi

- Mất frame: vì một lý do nào đó mà frame gửi đã không đến được đích
- Hỏng frame: nhận được frame đến nhưng có một số bit bị lỗi

Stop-and-Wait ARQ

- Phương pháp này dựa trên nguyên tắc chung của kỹ thuật điều khiển lưu lượng stop-and-wait.
- Trạm nguồn truyền 1 frame và chờ tín hiệu ACK.
- Hai loại lỗi xuất hiện:
  - frame đến được đích nhưng bị hỏng, tức là trạm đích dùng kỹ thuật phát hiện lỗi thấy lỗi và huỷ frame
    - Để giải quyết vấn đề này, trạm nguồn sử dụng một đồng hồ, và nếu sau khoảng thời gian timeout không nhận được tín hiệu ACK thì frame đó được truyền lại

Stop-and-Wait ARQ

- Tín hiệu ACK khi gửi lại bị hỏng, tức là trạm đích đã nhận đúng frame và báo tin hiệu ACK, nhưng tín hiệu ACK bị hỏng nên trạm nguồn không nhận ra (hoặc không đến được)
  - Trường hợp này trạm nguồn cũng dùng thời gian timeout để truyền lại frame. Giả sử frame truyền lại này cũng đến đích và không bị lỗi, khi đó trạm đích sẽ nhận được 2 frame giống nhau. Để giải quyết vấn đề này, các frame được luân phiên gán nhãn 0 và 1, và tín hiệu ACK tương ứng là ACK1 và ACK0. Ở đây, ACK0 sẽ báo nhận của frame 1 và có với ý nghĩa thông báo là đã sẵn sàng nhận tiếp (frame 0).

Stop-and-Wait ARQ

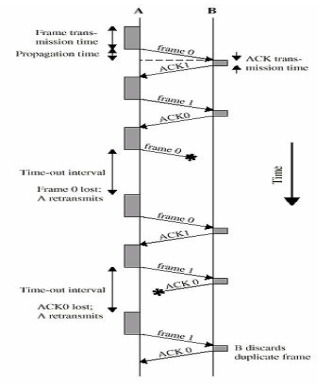


Figure 7.8 Stop-and-Wait ARQ

Go-Back-N ARQ

- Phương pháp này dựa trên kỹ thuật điều khiển lưu lượng cửa sổ trượt (sliding-window).
- Trạm nguồn có thể gửi hàng loạt các frame tuần tự được đánh số. (Số frame không được báo nhận còn lại sẽ được chỉ ra bởi kích cỡ cửa sổ.)
- Trạm đích sẽ báo tín hiệu (tín hiệu RR - receive ready hoặc kèm ACK - piggybacked ACK) các frame đến bình thường nếu không có lỗi nào xuất hiện.

Go-Back-N ARQ

- Khi trạm đích phát hiện một frame lỗi, nó sẽ báo tín hiệu NACK (REJ - reject) cho frame đó và sẽ huỷ bỏ mọi frame đến sau cho đến khi frame lỗi được nhận đúng.
- Còn trạm nguồn khi nhận được tín hiệu REJ nó sẽ truyền lại frame lỗi và tất cả các frame đã truyền sau frame lỗi.

Go-Back-N ARQ

- Sau mỗi frame truyền, trạm nguồn khởi tạo giá trị timeout cho frame đó. Giả sử trạm đích (B) nhận thành công frame i-1 và trạm nguồn (A) gửi tiếp frame i, ta có thể có các trường hợp lỗi sau:
  - Hỏng frame: B sẽ huỷ frame i, ta có 2 trường hợp nhỏ:
  - Hỏng tín hiệu RR: Có 2 trường hợp:

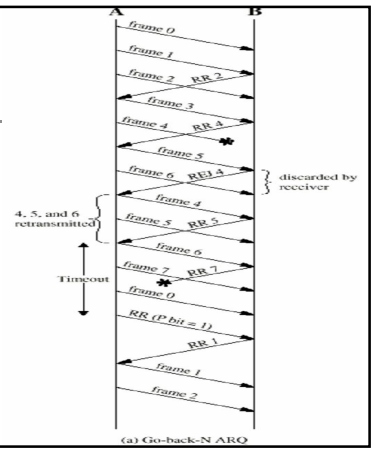
Hỏng frame

- Ta có 2 trường hợp nhỏ:
  - Trong chu kỳ thời gian hợp lệ, A gửi tiếp frame i+1, B nhận được i+1 nhưng không đúng thứ tự (do hỏng i) và gửi REJ. A phải truyền lại i và tất cả các frame sau đó.
  - A không gửi tiếp frame i+1, nên B không nhận được gì thêm và cũng không gửi tín hiệu hoặc RR hoặc REJ. Khi A đạt giá trị timeout, nó sẽ truyền một frame RR gồm 1 bit (được hiểu như bit P - Poll) được thiết lập giá trị 1. Khi đó, B hiểu frame RR với bit P=1 như là một lệnh yêu cầu phải gửi thông báo tín hiệu RR cho biết sẵng sàng nhận frame tiếp theo là frame i. Và khi A nhận được tín hiệu RR nó sẽ truyền lại frame i.

Hỏng tín hiệu RR

- B nhận đúng frame i và báo RR sẵn sàng nhận i+1, nhưng tín hiệu RR bị mất.
- Tức là A chỉ nhận được tuần tự các tín hiệu RR của các frame 1,2,…,i-1,i và vẫn không nhận được i+1 cho đến khi đạt giá trị timeout của i.
- Khi A đạt giá trị timeout nó sẽ truyền một frame RR đặc biệt như trường hợp trên, và thiết lập giá trị timeout của P-bit (P=1). Nếu tín hiệu trả lời RR này của B cũng thất bại thì khi A đạt giá trị timeout của P-bit nó sẽ phát lại lệnh RR. Và tất nhiên sau một số lần nhất định phát lại lệnh RR mà vẫn không nhận được tính hiệu báo nhận thì A sẽ khởi tạo lại thủ tục truyền từ đầu.

Go-Back -N ARQ

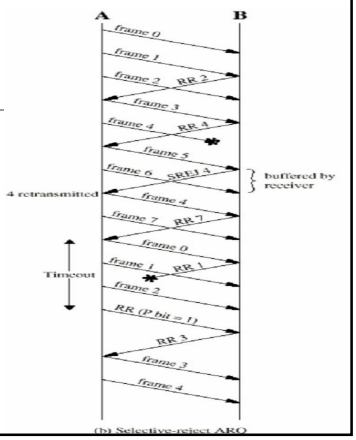


(a) Go-back-N ARQ

Selective-Reject ARQ

- Tương tự Go back N nhưng khi cần truyền lại thì các frame cần truyền lại chỉ là các frame có báo nhận NACK (ta gọi là tín hiệu SREJ) hoặc quá thời gian timeout
  - Ưu điểm: Đầy lại có chọn lọc thực hiện hiệu quả hơn Go-Back-N, bởi vì nó truyền lại các frame là ít nhất.
  - Nhược điểm:duy trì một bộ nhớ tạm đủ để lưu các frame đến cho đến khi frame lỗi nhận đúng và phải tạo vị trí logic để chèn frame lỗi vào đúng thứ tự. Và trạm truyền cũng phải duy trì nhiều logic phức tạp để có thể gửi các frame không theo tuần tự. Chí

Selective-Reject ARQ



(b) Selective-reject ARQ