### CHƯƠNG 2: TÌM HIỂU VỀ GIAO THỨC TRUYỀN TIN Y - MODEM

# 2.1 Giới thiệu về giao thức XMODEM

## 2.1.1 Khái quát chung

Ngày nay khi máy tính cá nhân được phổ biến rộng rãi thì việc kết nối giữa các máy tính với nhau trở lên hết sức cần thiết. Mô hình kết nối đơn giản nhất là thiết lập một giao thức (hay còn gọi là thủ tục) truyền dữ liệu giữa hai máy tính.

Khi ta muốn thiết lập một đường truyền dữ liệu giữa hai hay nhiều máy tính, ta phải chỉ ra một số quy định về khuôn dạng, thời gian, cách thức làm việc trên đường truyền đó để đồng bộ quá trình làm việc giữa các máy tính. Các quy định đó được gọi chung là một giao thức (hay còn gọi là thủ tục) cho việc truyền số liệu.

Một giao thức phải đảm bảo được các tính chất sau:

Tính đơn giản: Giao thức phải đơn giản tới mức có thể dễ dàng lập trình, mã hóa bằng ngôn ngữ máy, đồng thời lỗi và thời gian truyền tối thiểu.

Tính cô đọng: Giao thức sẽ truyền tối đa dữ liệu trên mức tối thiểu tổng phí

Tính tin cậy: Trên thực tế đường truyền không phải là lý tưởng do đó giao thức phải phát hiện được những lỗi xảy ra trên đường truyền và phải có cơ chế chỉ ra chính xác rằng dữ liệu được truyền đến đích một cách đúng đắn.

Tính mềm dẻo: Giao thức phải có khả năng truyền dữ liệu nhanh nhất trong mọi tốc độ của đường truyền.

Tất nhiên không phải bất cứ giao thức nào cũng thỏa mãn được tất cả các tính chất trên nhưng trong trường hợp ứng dụng cụ thể ta phải chọn lấy giao thức thích hợp nhất.

Một trong những giao thức phổ biến nhất là giao thức XMODEM do Ward Christiansen thiết kế. Giao thức Xmodem có những đặc điểm chủ yếu sau:

+ Giao thức tiếp nhận dữ liệu chia nhỏ thành các gói có kích thước là 128 bytes sau đó chuyển gói dữ liệu trên đường truyền đến nơi nhận. Tại nơi nhận các gói được liên kết lại thành một file dữ liệu như lúc ban đầu.

+ Môi trường dùng để thực hiện giao thức rất đơn giản, có thể là qua dây điện thoại (thông qua Modem) hay sử dụng dây nối trực tiếp hai cổng nối tiếp của hai máy tính..

+ Hiệu suất truyền dữ liệu cao.

+ Trong giao thức có sử dụng thuật toán phát hiện lỗi đơn giản giúp kiểm tra độ tin cậy của môi trường sử dụng.

Phương thức hoạt động của giao thức Xmodem như sau:

Dữ liệu được tiếp nhận ở đầu gửi và được chia nhỏ thành các đoạn có độ dài 128 byte. Trong giao thức X-MODEM tất cả các field ngoại trừ field thông tin, cso độ dài là một byte. Sau đó đoạn dữ liệu nói trên được Xmodem đóng gói thành một gói thông tin có cấu trúc như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Đầu gói (SOH) | Số hiệu gói | Phần bổ sung | Dữ liệu | Tổng kiểm tra (CheckSum) |

Trong đó:

- Đầu gói là ký tự đầu gói SOH

- Số hiệu gói là số thứ tự của gói mang giá trị từ 01 đến 255. Khi số hiệu gói đạt giá trị 255 thì gói tiếp theo sẽ mang số hiệu là 0.

- Phần bổ sung là giá trị bằng 255 (0xFF) trừ đi số hiệu gói.

- Dữ liệu bao gồm 128 byte dữ liệu cần truyền.

- Vùng kiểm tra là một byte, dùng phương pháp kiểm tra kiểu tổng và chi tính theo nội dung của vùng thông tin.

Sau này giao thức XMODEM-CRC sử dụng field kiểm tra có độ dài 2 byte và dùng phương pháp kiểm tra CRC với đa thức sinh CCITT là: X16 + X12 + X5 + 1.

## 2.1.2 Kỹ thuật truyền nhận tin của giao thức X-MODEM

|  |  |  |  |
| --- | --- | --- | --- |
| SENDER | | RECEIVER | |
|  |  | | NAK |
| SOH 01 FE Data[128] CkSum |  | |  |
|  |  | | ACK |
| SOH 02 FD Data[128] CKSum |  | |  |
|  |  | | ACK |
| SOH 03 FC Data[128] CkSum |  | |  |
|  |  | | ACK |
| EOT |  | |  |
|  |  | | ACK  C |

**Truyền tập tin bằng giao thức XMODEM:**

Hình là lưu đồ mô tả quá trình truyền mỗi tập tin của giao thức XMODEM. Quá trình này có thể chia làm 3 giai đoạn:

(i) Giai đoạn khởi động:

Như mọi giao thức khác, để khởi động việc truyền phải có sự hợp đồng giữa máy phát và máy thu. Máy phát chờ tín hiệu NAK từ máy thu. Khi có một tín hiệu NAK đầu tiên thì máy phát sẽ bắt đầu thực hiện việc truyền gói đầu tiên.

(ii) Giai đoạn giữa

Máy phát chia tập tin thành từng khối 128 bytes để truyền đi và chờ tín hiệu báo nhận của máy thu đối với từng gói sau khi truyền. Nếu nhận được tín hiệu ACK máy phát sẽ phát gói kế tiếp. Nếu nhận được tín hiệu NAK máy phát truyền lại gói trước đó. Nếu nhận được tín hiệu CAN thì máy phát sẽ kết thúc truyền không điều kiện.

(iii) Giai đoạn kết thúc.

Nếu giai đoạn giữa diễn ra bình thường thì máy phát sẽ truyền thông điệp EOT (End Of Transmission) để báo cho máy thu biết không còn truyền tập tin nào nữa. Lúc này máy thu sẽ đáp lại bằng tín hiệu ACK để cho biết nhận được EOT.

Sau đó là công việc kết thúc việc truyền tập tin như đóng tập tin, giải phóng vùng nhớ chứa tập tin.

Nếu giai đoạn giữa kết thúc không bình thường, do có tín hiệu CAN từ máy thu chẳng hạn thì máy phát kết thúc truyền tập tin mà không gửi tín hiệu EOT.



**Nhận tin bằng giao thức XMODEM:**

Trong quá trình nhận tập tin, ngoài việc nhận các gói dữ liệu còn phải kiểm tra các field chứa thông tin dịch vụ truyền để đảm bảo gói vừa nhận được là đúng số thứ tự cần và không bị sai. Do đó quá trình nhận sẽ phức tạp hơn quá trình truyền.

Quá trình nhận tập tin được xúc tiến qua 3 giai đoạn:

(i) Giai đoạn khởi động.

Sau khi thực hiện các công việc chuẩn bị như cấp phát vùng nhớ để chứa tạm (đệm) các gói nhận, mở các tập tin, đặt tên tập tin, chỉ đường dẫn…để sẵn sàng nhận các gói dữ liệu.

(ii) Giai đoạn giữa

Máy thu sẽ chuyển vào một vòng lặp trong đó thực hiện các thao tác như chờ một gói dữ liệu đến và truyền tín hiệu NAK nếu không có gói dữ liệu đến trong vòng 10 giây.

Khi nhận một byte đến thì máy thu biết là có một gói dữ liệu đến, và nó sẽ thực hiện các thao tác sau:

+ Nếu byte nhận được không phải là SOH mà là EOT thì sẽ kết thúc giai đoạn giữa.

+ Nếu byte nhận được là CAN sẽ kết thúc nhân không điều kiện.

+ Máy thu kiểm tra gói đến. Nó phải biết chắc là vùng thứ hai và thử gói không sai lệch, bằng cách lấy bù 1 của giá trị ở một vùng nào đó (thứ 2 hoặc thứ 3) rồi EXOR với giá trị của vùng kia. Kết quả là 0 nếu khác 0 thì máy thu sẽ truyền tín hiệu NAK về cho máy phát.

+ Kế đến máy thu kiểm tra số thứ tự của gói đến có phải là gói đang muốn nhận. Nếu không phải thì có nghĩa là có một sự cố sai sót xảy ra. Vì máy thu không thể biết sự cố này, nên không thể khắc phục nên truyền lại tín hiệu CAN để yêu cầu máy phát kết thúc việc truyền. chỉ có trường hợp số thứ tự của gói đến trùng với số thứ tự của gói trước đó thì nó giả thiết là máy phát không nhận được ACK của gói này. Lúc này máy thu loại bỏ gói mới đến và truyền một tín hiệu ACK khác và quay lại chờ gói kế tiếp đến.

+ Cuối cùng máy thu sẽ tính vùng theo phương pháp tổng kiểm tra hoặc CRC để so sánh với giá trị trong vùng kiểm tra của gói vừa nhận. Nếu hai giá trị trùng nhau thì xem như không có lỗi và máy thu sẽ truyền tín hiệu ACK, ngược lại sẽ truyền tín hiệu NAK.

(iii) Giai đoạn kết thúc

Nếu quá trình nhận kết thúc bình thường thì máy phát sẽ truyền cho máy thu thông điệp EOT, máy thu sẽ đáp ứng lại bằng một ACK và tiến hành các công việc sau cùng như đóng các tập tin, giải phóng vùng nhớ..



Giao thức XMODEM tương đối đơn giản, dễ lập trình. Tỷ lệ giữa dữ liệu cần truyền và tổng gói dữ liệu phải truyền là lớn chỉ có 4 byte bổ sung cho mỗi gói có 128 byte dữ liệu cần truyền. Giao thức này dễ mã hóa, lập trình nên ta có thể dùng trong các đường truyền tốc độ cao mà không có thời gian trễ trong khi truyền. Điều này làm cho giao thức XMODEM trở nên phổ biến rộng rãi trên hầu hết các mạng thông tin hiện nay.

Độ tin cậy là đủ thích hợp, tuy vậy trong một số trường hợp thì 1 byte tổng kiểm tra không đủ tốt để phát hiện khi xuất hiện nhiều bít lỗi trong một gói.

Giao thức truyền Xmodem làm việc theo chế độ bán song công (half duplex), tức là tại một thời điểm chỉ có một phía truyền thông tin. Ví dụ máy A sau khi gửi một gói thì phải chờ đợi máy B trả lời kết quả thì mới gửi tiếp gói tiếp theo.

## 2.1.3 Đặc điểm của giao thức cải tiến X- MODEM -CRC

Dựa trên giao thức XMODEM, người ta cải tiến một giao thức mới có tên là Xmodem CRC nhầm tăng khả năng phát hiện lỗi trong khi truyền. Điểm cải tiến duy nhất của Xmodem CRC so với Xmodem là:

Thay vì một byte tổng kiểm tra người ta sử dụng hai byte kiểm tra CRC (Cycle Redundancy Check) cho một khối 128 byte dữ liệu. Dữ liệu này làm Xmodem CRC có thể phát hiện được hầu hết các lỗi tới 16 bit. Nó phát hiện được 99,99% các lỗi 1 bit và 98% các lỗi 2 bit. Dữ liệu trong giao thức Xmodem CRC được đóng gói với cấu trúc như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Đầu gói | Số hiệu gói | Phần bổ sung | Dữ liệu | Tổng kiểm tra CRC |

Đối với giao thức Xmodem – CRC các quá trình truyềnvà nhận cũng tương tự XMODEM, chỉ khác ở giai đoạn khởi động, máy thu sẽ không truyền tín hiệu NAK mà truyền ký tự “C” và máy phát cũng chờ ký tự “C” để khởi động. Còn sau đó vẫn dùng tín hiệu ACK và NAK để đồng bộ quá trình truyền nhận gói.

Tóm lại, giống như giao thức xmodem, giao thức Xmodem CRC là một giao thức đơn giản, dễ lập trình, phổ biến trên và có tính tin cậy cao.

## 2.1.4 Ưu nhược điểm của giao thức X-MODEM

Ưu điểm:

Các bản tin cần truyền được chia nhỏ thành các gói có kích thước là 128 byte và được tuần tự truyền đi. Điều này làm cho ta dễ dàng phát hiện lỗi và sửa lỗi trong các gói tin hơn là trong cả một bản tin lớn. Khi có các lỗi không khắc phục được cần truyền lại thì việc truyền lại một gói tin có lỗi mất ít thời gian hơn là cả một bản tin lớn.

Việc gán mã phát hiện lỗi CRC đã cho phép phát hiện được các gói thông tin bị truyền sai trong khi truyền do đó đảm bảo được tin đúng đắn của bản tin.

Các gói thông tin được đánh số trong truyền tin cho phép ta dễ dàng liên kết các gói lại ở phía nhận thành bản tin cần truyền.

Nhược điểm:

Việc truyền các gói thông tin theo giao thức Xmodem được gọi là phương pháp (stop and wait). Theo phương pháp này thì bên phát truyền đi một gói tin sau đó đợi bên trả lời kết quả đã nhận đúng hay chưa sau đó mới truyền gói tiếp theo. Quá trình này phải mất thêm thời gian đáng kể chó việc chờ tín hiệu chấp nhận của máy thu đối với từng gói truyền.

Độ dài của khối ngắn làm cho dữ liệu bị hỏng khi sử dụng với các hệ thống chia đoạn, các mạng chuyển mạch gói, các mạch vệ tinh, và các bộ điều giải đệm (sửa lỗi). Giao thức chỉ cho phép truyền nhận khối dữ liệu có độ dài nhỏ 128 bytes. Chưa có sự linh hoạt trong việc thay đổi độ dài khối dữ liệu cho phù hợp khi chất lượng đường truyền có sự thay đổi.

Chỉ có một tập tin có thể được gửi cho mỗi lệnh. Tên tệp phải được cung cấp hai lần, trước tiên vào chương trình gửi và sau đó lại cho chương trình nhận.

Giao thức này sẽ hủy bỏ việc truyền nhận một tập tin ngay sau khi nhận được mẫu bit của ký tự CAN. Tuy nhiên mẫu bit này rất dễ tạo ra bởi nhiễu trên đường truyền dẫn đến việc truyền nhận tin bằng giao thức này rất hay xảy ra việc dừng hoạt động. Các nhược điểm này sẽ được khắc phục và hạn chế tối thiểu ở giao thức Y-MODEM.

# 2.2 Giao thức Y – MODEM

## 2.2.1 Đặc điểm của giao thức Y-MODEM

a, Yêu cầu tối thiểu giao thức Y-MODEM

Giao thức Y-MODEM là sự mở rộng của giao thức X-MODEM, được đưa ra đầu tiên vào năm 1981.

Tất cả các chương trình yêu cầu hỗ trợ YMODEM phải đáp ứng các yêu cầu tối thiểu sau:

+ Chương trình gửi sẽ gửi tên đường dẫn (tên tập tin) trong khối 0.

+ Tên đường dẫn sẽ là một chuỗi ASCII kết thúc rỗng (null) như được mô tả bên dưới.

         + Hệ thống không phân biệt chữ hoa và chữ thường trong tên tập tin sẽ chỉ gửi tên đường dẫn theo thứ tự chữ thường.

+ Chương trình nhận sẽ sử dụng tên đường dẫn này cho tên tập tin đã nhận, trừ khi được ghi đè rõ ràng.

+ Khi chương trình nhận nhận được khối này và mở tập tin đầu ra thành công, nó sẽ thừa nhận khối này bằng cách gửi một ký tự ACK cho bên phát và sau đó bắt đầu tiến hành truyền tập tin tương tự như giao thức XMODEM khi nhận được ký tự "C" hoặc NAK được truyền bởi người nhận.

+ Chương trình gửi sẽ sử dụng CRC-16 khi nhận được tín hiệu "C" và sau đó gửi tên đường dẫn cho bên thu, nếu không sử dụng checksum 8 bit.

+ Chương trình nhận phải chấp nhận bất kỳ hỗn hợp khối 128 và 1024 byte trong mỗi tập tin nó nhận được. Các chương trình gửi có thể tự ý chuyển đổi giữa các khối 1024 và 128 byte.

+ Chương trình gửi không được thay đổi độ dài của một khối không được công nhận

+ Vào cuối mỗi tập tin, chương trình gửi sẽ gửi EOT đến mười lần cho đến khi nó nhận được một ký tự ACK. (Đây là một phần của thông số XMODEM)

+ Kết thúc phiên truyền sẽ được biểu thị bằng một tên đường dẫn nul (rỗng), khối tên đường dẫn này sẽ được thừa nhận giống như các khối tên đường dẫn khác.

Các chương trình không đáp ứng tất cả các yêu cầu này sẽ không phải là YMODEM tương thích và sẽ không được hỗ trợ như YMODEM.

Việc đáp ứng các yêu cầu tối thiểu này không thể đảm bảo 100% việc truyền tệp tin chính xác, đáng tin cậy hoàn toàn khi gặp tất cả các sự cố, trục trặc trên đường truyền. Các ký tự đơn để thông báo các sự cố lỗi khi gửi tin của giao thức XMODEM dễ bị hỏng do lỗi truyền dẫn thông thường.

b) Cấu trúc gói tin của giao thức Y-MODEM

Y - MODEM đóng gói thành một gói thông tin có cấu trúc như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Đầu gói | Số hiệu gói | Phần bổ sung | Dữ liệu | Tổng kiểm tra CRC |

Trong đó:

Đầu gói là ký tự đầu gói SOH nếu dữ liệu có kích thước là 128byte còn đầu gói là STX thì dữ liệu gửi đi có kích thước là 1024 byte.

Số hiệu gói là số thứ tự của gói mang giá trị từ 0 đến 255. Khi số hiệu gói đạt giá trị 255 thì gói tiếp theo sẽ mang số hiệu là 0.

Phần bổ sung là giá trị bằng 255 (0xFF) trừ đi số hiệu gói.

Dữ liệu bao gồm dữ liệu cần truyền có thể là ở dạng text hoặc nhị phân.

Vùng kiểm tra là sử dụng field kiểm tra có độ dài 2 byte và dùng phương pháp kiểm tra CRC với đa thức sinh CCITT là: X16 + X12 + X5 + 1. CRC được sử dụng trong giao thức Modem là một dạng kiểm tra luân phiên cung cấp khả năng phát hiện lỗi mạnh mẽ hơn so với checksum gốc. CRC-CCITT được sử dụng bởi giao thức Modem sẽ phát hiện ra tất cả các lỗi đơn và đôi, tất cả các lỗi với một số lẻ các bit, tất cả các lỗi burst có chiều dài 16 hoặc nhỏ hơn, 99.997 % số lỗi lỗi 17-bit, và 99.998% của18-bit và dài hơn.

Giao thức Y-MODEM có hai điểm cải tiến cơ bản so với X-MODEM là:

- Cho phép truyền nhận các khối dữ liệu có độ dài 1024 byte (gấp 8 lần khối dữ liệu của X-MODEM). Nếu đường truyền không tốt, Y-MODEM tự động giảm độ dài khối 128 byte để giảm bớt số byte phải truyền lại khi phát hiện sai. Như vậy trong trường hợp xấu nhất thì đội dài khối và chất lượng truyền sẽ bằng giao thức XMODEM.

- Bên cạnh đó, Y - Modem còn có một cải tiến nhằm giảm khả năng hư hỏng khi kết thúc truyền tin. Như đã biết X-MODEM sẽ hủy bỏ việc truyền nhận một tập tin ngay sau nhận được mẫu bít của ký tự Ctrl-X (ký tự ASCII là CAN) mẫu bit này rất dễ tạo ra bởi nhiễu trên đường truyền. Trong khi Y-MODEM yêu cầu cần phải có hai ký tự CAN liên tiếp nhau mới hủy bỏ việc truyền nhận tập tin.

- Y-MODEM còn cung cấp hai tính năng có ý nghĩa cho người sử dụng, đó là việc sử dụng phương pháp phát hiện và kiểm tra sai CRC và truyền các thông tin liên quan đến tập trin truyền cho máy thu. Các thông tin này, gồm có tên tập tin, ngày giờ và kích hoạt tập tin được truyền, các thông tin này ở trong khối đầu tiên khi bắt đầu việc truyền.

## 2.2.2 Kỹ thuật truyền nhận tin của giao thức Y-MODEM

|  |  |  |  |
| --- | --- | --- | --- |
| SENDER  "Gửi chế độ loạt (bacth)." | | RECEIVER  "sb foo.\*<CR>" | |
|  |  | | C (command:rb) |
| SOH 00 FF foo.c NUL[123] CRC CRC |  | |  |
|  |  | | ACK  C C |
| SOH 01 FE Data[128] CRC CRC |  | |  |
|  |  | | ACK |
| STX 02 FD Data[1024] CRC CRC |  | |  |
|  |  | | ACK |
| SOH 03 FC Data[128] CRC CRC |  | |  |
|  |  | | ACK |
| EOT |  | |  |
|  |  | | NAK |
| EOT |  | |  |
|  |  | | ACK  C C |
| SOH 00 FF NUL[128] CRC CRC |  | |  |
|  |  | | ACK |

Truyền tập tin bằng giao thức YMODEM:

Hình là lưu đồ mô tả quá trình truyền mỗi tập tin của giao thức Y-MODEM. Quá trình này có thể chia làm 3 giai đoạn:

(i) Giai đoạn khởi động:

Nhiệm vụ đầu tiên trong bất cứ giao thức truyền tập tin nào cũng phải là thiết lập mối liên kết giữa phía gửi và phía nhận. Giao thức Y - MODEM thuộc dạng giao thức điều khiển từ phía nhận (Receiver driven), tức là phía nhận chịu trách nhiệm kích thích và duy trì truyền thông các gói dữ liệu. Phía phát chỉ giữ vai trò khởi tạo ban đầu hoặc đồng bộ về pha của quá trình truyền.

Như giao thức X-MODEM, để khởi động việc truyền phải có sự hợp đồng giữa máy phát và máy thu. Máy phát chờ tín hiệu ‘C’ từ máy thu. Khi có một tín hiệu ‘C’ đầu tiên thì máy phát sẽ bắt đầu thực hiện việc truyền gói đầu tiên đây được gọi là khối 0:

SOH 00 FF foo.c NUL[123] CRC CRC

Gói số 0 gồm các thông tin sau:

- SOH là ký tự đầu gói

- 00 là số hiệu của gói số 0

- FF là phần bổ sung

- foo.c là tên đường dẫn:

+ Để duy trì khả năng tương thích trở lên, tất cả các byte không sử dụng trong khối 0 phải được đặt thành null (rỗng).

+ Tên đường dẫn (thông thường, tên tệp tin) được gửi dưới dạng chuỗi ASCII kết thúc rỗng. Đây là định dạng tên tập tin được sử dụng bởi các chức năng định hướng MSDOS (TM) và các chức năng của thư viện C fopen. Một ví dụ về ngôn ngữ assembly sau:

DB 'foo.bar', 0

+ Không có dấu cách nào trong tên đường dẫn.

(ii) Giai đoạn giữa

Sau khi máy thu nhận được gói dữ liệu đầu tiên là khối 0, nó sẽ gửi ACK cho bên phát rằng đã nhận đúng gói tin và sau đó gửi tín hiệu ‘C’.

Sau khi, máy phát nhận được ‘C’, bắt đầu truyền gói liệu tiếp theo. Bên thu phải chấp nhận bất kỳ hỗn hợp khối 128 và 1024 byte trong mỗi tập tin nó nhận được. Bên gửi có thể tự ý chuyển đổi giữa các khối 1024 và 128 byte.

Bên thu nhận dữ liệu, sau đó gửi ACK và chờ đợi việc truyền gói dữ liệu tiếp theo. Nếu nhận được tín hiệu ACK máy phát sẽ phát gói kế tiếp. Nếu nhận được tín hiệu NAK máy phát truyền lại gói trước đó. Nếu nhận được hai tín hiệu CAN thì máy phát sẽ kết thúc truyền không điều kiện.

(iii) Giai đoạn kết thúc.

Nếu giai đoạn giữa diễn ra bình thường thì máy phát sẽ truyền thông điệp EOT (End Of Transmission) để báo cho máy thu biết không còn truyền tập tin nào nữa. Lúc này máy thu sẽ đáp lại bằng tín hiệu NAK để cho biết nhận được EOT.

Sau khi nhận được tín hiệu NAK, máy phát truyền lại EOT. Lúc này máy thu tiếp tục gửi tín hiệu ACK cho bên phát.

Cuối cùng, máy thu sẽ gửi ‘C’. Bên phát nhận được tín hiệu ‘C’ và gửi gói dữ liệu sau:

SOH 00 FF NUL[128] CRC CRC

Kết thúc phiên truyền sẽ được biểu thị bằng một tên đường dẫn nul (rỗng), tên đường dẫn khối này sẽ được thừa nhận giống như các khối tên đường dẫn khác.

Bên thu sau khi nhận được gói dữ liệu tiến hành truyền ACK kết thúc chính thức việc truyền tin. Sau đó là công việc kết thúc việc truyền tập tin như đóng tập tin, giải phóng vùng nhớ chứa tập tin. Nếu giai đoạn giữa kết thúc không bình thường, do có hai tín hiệu CAN từ máy thu chẳng hạn thì máy phát kết thúc truyền tập tin mà không gửi tín hiệu EOT.



Quá trình nhận tập tin được xúc tiến qua 3 giai đoạn:

(i) Giai đoạn khởi động.

Sau khi thực hiện các công việc chuẩn bị như cấp phát vùng nhớ để chứa tạm (đệm) các gói nhận, mở các tập tin, đặt tên tập tin, chỉ đường dẫn…để sẵn sàng nhận các gói dữ liệu.

(ii) Giai đoạn giữa

Đầu tiên bên thu sẽ gửi tín hiệu ‘C’ và chờ nhận gói tin số 0 từ bên phát. Sau khi nhận được gói tin đầu tiên, máy thu sẽ tiến hành mở tập tin và xác nhận gói tin là đúng mới tiến hành gửi tín hiệu “ACK” và “C” cho bên phát.

Máy thu sẽ chuyển vào một vòng lặp trong đó thực hiện các thao tác như chờ một gói dữ liệu đến và truyền tín hiệu ’NAK” nếu không có gói dữ liệu đến trong vòng 10 giây. Việc truyền nhận tin sau đó được thực hiện giống với giao thức X-MODEM.

Khi nhận một byte đến thì máy thu biết là có một gói dữ liệu đến, và nó sẽ thực hiện các thao tác sau:

+ Nếu byte nhận được không phải là SOH mà là EOT thì sẽ kết thúc giai đoạn giữa.

+ Nếu byte nhận được là hai tín hiệu CAN sẽ kết thúc nhận không điều kiện.

+ Máy thu kiểm tra gói đến. Nó phải biết chắc là vùng thứ hai và thử gói không sai lệch, bằng cách lấy bù 1 của giá trị ở một vùng nào đó (thứ 2 hoặc thứ 3) rồi EXOR với giá trị của vùng kia. Kết quả là 0 nếu khác 0 thì máy thu sẽ truyền tín hiệu NAK về cho máy phát.

+ Kế đến máy thu kiểm tra số thứ tự của gói đến có phải là gói đang muốn nhận. Nếu không phải thì có nghĩa là có một sự cố sai sót xảy ra. Vì máy thu không thể biết sự cố này, nên không thể khắc phục nên truyền lại hai tín hiệu CAN để yêu cầu máy phát kết thúc việc truyền. Chỉ có trường hợp số thứ tự của gói đến trùng với số thứ tự của gói trước đó thì nó giả thiết là máy phát không nhận được ACK của gói này. Lúc này máy thu loại bỏ gói mới đến và truyền một tín hiệu ACK khác và quay lại chờ gói kế tiếp đến.

+ Cuối cùng máy thu sẽ tính vùng theo phương pháp CRC để so sánh với giá trị trong vùng kiểm tra của gói vừa nhận. Nếu hai giá trị trùng nhau thì xem như không có lỗi và máy thu sẽ truyền tín hiệu ACK, ngược lại sẽ truyền tín hiệu NAK.

(iii) Giai đoạn kết thúc

Nếu quá trình nhận kết thúc bình thường thì máy phát sẽ truyền cho máy thu thông điệp EOT, máy thu sẽ đáp ứng lại bằng một NAK. Sau đó bên thu lại nhận tiếp thông điệp EOT thứ hai và khi nhận được sẽ truyền lại cho bên phát tín hiệu ACK và tiến hành các công việc sau cùng như đóng các tập tin, giải phóng vùng nhớ..



## 2.2.3 Ưu nhược điểm của kỹ thuật truyền tin Y-MODEM

Ưu điểm:

Y-MODEM có nhiều đặc điểm cải tiến hơn so với X-MODEM. Trong đó pha khởi tạo cho phép khả năng nhận biết “được phép phát” của bên phát nên có thể tận dụng để tùy biến thành pha xác thực đầu cuối nhận điện. Đồng thời, nó cho phép truyền nhận các khối dữ liệu có độ dài 1024 bytes (gấp 8 lần khối dữ liệu của X-MODEM).

Ưu điểm sau cùng của YMODEM so với XMODEM là tính năng truyền nhóm tập tin. Có thể dùng đường dẫn và ký tự đại diện để chọn một nhóm các tập tin truyền

Nhược điểm:

Chỉ truyền được ký tự 8 bit. Cũng dùng thủ tục Idle RQ truyền và chờ, sau mỗi gói được truyền đi máy phát sẽ chờ tín hiệu báo nhận tốt. Vì thế giao thức này phải thêm thời gian đáng kể do việc chờ tín hiệu chấp nhận của máy thu đối với từng gói truyền.

Giữa bên phát và bên nhận thực hiện truyền nhận theo hình thức quay số, kết nối và truyền tin sự xác thực được thực hiện thông qua số điện thoại. Vì vậy sự xác thực ở đây chỉ là xác thực nhân công chưa có sự xác thực tự động. Việc sử dụng xác thực tự động, thông qua chèn các gói tin của giao thức xác thực vào các khung dữ liệu của giao thức Y-MODEM ban đầu là một giải pháp có tính khả thi cao khi tùy biến giao thức Y-MODEM.

## Kết luận

Trong phần đầu chương này đã trình bày về khái niệm, đặc điểm kỹ thuật truyền nhận tin của giao thức X-MODEM, đặc điểm cải tiến của giao thức X-MODEM-CRC và ưu nhược điểm của giao thức X-MODEM. Phần hai trình bày về giao thức Y-MODEM bao gồm: đặc điểm, cấu trúc gói tin, kỹ thuật truyền nhận tin và ưu nhược điểm của giao thức Y-MODEM. Từ việc tìm hiểu về đặc truyền tin của giao thức Y-MODEM ta tiến hành tùy biến giao thức để phục vụ việc xác thực tự động hai đầu mối liên lạc ở chương 3.

### CHƯƠNG 3: TÙY BIẾN GIAO THỨC Y – MODEM PHỤC VỤ TRUYỀN TIN CÓ XÁC THỰC TỰ ĐỘNG

Chương này tiến hành trình bày giao thức truyền tin và đề xuất tùy biến giao thức Y-MODEM phục vụ xác thực, xây dựng chương trình và đánh giá kết quả thực hiện.

# 3.1 Chương trình truyền tin bằng giao thức Y – MODEM

## 3.1.1 Lưu đồ thuật toán bên gửi

a) Quá trình gửi gói tin đầu tiên

- Hai bên thu phát tiến hành kết nối với nhau.

- Bên phát chờ nhận ký tự “C” từ bên thu để thực hiện việc truyền gói tin đầu tiên. Trong quá trình truyền gói tin đầu tiên có các trạng thái xảy ra như sau:

+ Nếu bên phát nhận được 2 tín hiệu CAN thì bên phát sẽ tiến hành hủy truyền.

+ Nếu bên phát nhận được tín hiệu ESC thì người dùng sẽ hủy quá trình truyền.

+ Nếu quá trình đợi “C” đạt đến giá trị timeout = 60s, bên phát tiến hành kết thúc truyền.

+ Nếu bên phát nhận được ký tự “C” và gói tin không phải gói cuối. Bên gửi sẽ tiến hành tạo gói tin có chứa tên đường dẫn, số thứ tự gói là seq =0, tính toán CRC và tiến hành gửi gói đầu tiên cho bên thu.



b) Quá trình chờ ACK và C

- Sau khi bên phát gửi gói tin đầu tiên thì sẽ đợi tín hiệu phản hồi ACK và C từ bên thu để chắc chắn rằng gói tin đầu tiên đã được nhận và tiến hành gửi gói tiếp theo. Quá trình chờ ACK và C có các trạng thái sau:

+ Nếu nhận được ký tự ESC thì người dùng tiến hành hủy truyền.

+ Nếu nhận được hai tín hiệu CAN thì người gửi sẽ hủy truyền.

+ Nếu quá trình đợi ACK và C đạt giá trị timeout = 60s và lỗi nhiều hơn 5 sẽ hủy quá trình truyền. Nếu lỗi nhỏ hơn 5 sẽ quay trở lại quá trình chờ.

+ Nếu bên gửi nhận được ký tự “NAK” và số lỗi trên 10 bên gửi sẽ tự động dừng hoạt động còn lỗi nhỏ hơn 10 sẽ quay trở lại quá trình đợi ban đầu.

+ Nếu nhận được tín hiệu ACK và C thì bên gửi tiến hành chuyển trạng thái sang việc truyền gói tin tiếp theo.



c) Quá trình gửi gói tin tiếp theo

- Sau khi nhận được ACK và C từ bên thu thì bên gửi tiến hành truyền gói tin có chứa dữ liệu đầu tiên. Quá trình này xảy ra các trường hợp như sau:

+ Bên gửi tiến hành tạo gói tin: tăng giá trị seq lên 1, tạo 2 byte thuật toán kiểm tra CRC và gói tin ở đây chia làm 2 khả năng. Nếu đường truyền tốt thì gói tin sẽ có dữ liệu kích thước là 1024 byte, đầu gói có ký tự là STX. Nếu đường truyền không tốt bên gửi sẽ tự động chuyển sang kích thước 128 byte với ký tự đầu gói là SOH. Sau khi tạo ra gói tin thì bên phát tiến hành gửi gói tin.

+ Nếu bên phát nhận tín hiệu ESC hoặc 2CAN thì quá trình gửi hủy.

+ Nếu gửi gói tin đạt giá trị timeout = 60s và số lỗi trên 5 thì hủy truyền tin. Nếu số lỗi timeout nhỏ hơn 5 thì bên phát gửi lại gói tin.

+ Nếu bên gửi nhận được tín hiệu “NAK” từ bên thu và gói tin số lỗi lớn hơn 10 thì bên gửi kết thúc truyền tin. Nếu gói tin có số lỗi là nhỏ hơn 10 thì bên phát gửi lại gói tin.

+ Nếu nhận được ký hiệu “ACK” từ bên thu thì bên phát biết được rằng gói tin được gửi thành công và chuyển trạng thái tiếp theo gửi gói tin còn lại.



d) Quá trình gửi gói tin cuối và kết thúc gửi tin

- Khi còn gói tin cuối cùng bên phát sẽ gửi tín hiệu EOT cho bên thu:

+ Nếu bên gửi nhận được tín hiệu 2CAN và ESC thì hủy truyền.

+ Nếu bên thu phản hồi bằng tín hiệu “NAK” và số lỗi lớn hơn 10 thì bên phát hủy quá trình truyền. Ngược lại, số lỗi nhận được nhỏ hơn 10 thì bên phát gửi lại EOT.

+ Nếu quá trình đạt giá trị timeout = 10 và số lỗi lớn hơn 10 thì hủy truyền. Còn nếu số lỗi nhỏ hơn 10 thì yêu cầu gửi lại EOT.

+ Nếu nhận được ACK và C bên phát tiến hành gửi gói tin cuối cùng.



- Sau khi nhận được ký tự “ACK” và “C” thì bên gửi tiến hành tạo ra gói tin cuối:

+ Nếu nhận được tín hiệu 2CAN và ESC thì hủy truyền.

+ Nếu bên thu phản hồi bằng tín hiệu “NAK” và số lỗi lớn hơn 10 thì bên phát hủy quá trình truyền. Ngược lại, số lỗi nhận được nhỏ hơn 10 thì bên phát gửi lại gói tin cuối.

+ Nếu quá trình đạt giá trị timeout = 10 và số lỗi lớn hơn 10 thì hủy truyền. Còn nếu số lỗi nhỏ hơn 10 thì yêu cầu gửi lại gói tin cuối.

+ Nếu nhận được ACK bên phát biết được bên thu đã nhận được gói tin cuối thành công và kết thúc quá trình truyền.



## 3.1.2 Lưu đồ thuật toán bên nhận

a) Quá trình nhận gói tin đầu

- Sau khi 2 bên đã thiết lập kết nối thành công. Bên thu tiến hành cấp phát bộ nhớ tạm (đệm) để chứa gói tin nhận được. Để thực hiện nhận gói tin đầu tiên, bên thu sẽ truyền tín hiệu “C” cho bên phát và chờ nhận gói đầu tiên:

+ Nếu quá trình đợi gói đầu tiên đạt timeout = 5s và số lỗi lớn hơn 20 thì bên nhận hủy nhận gói tin. Còn số lỗi nhỏ hơn 20 thì bên thu quay trở lại quá trình chờ gói tin đầu tiên.

+ Nếu tín hiệu nhận được là “ESC” thì hủy bởi người dùng.

+ Nếu nhận được tín hiệu 2CAN thì chương trình thoát.

- Trường hợp ký tự đầu của gói tin nhận được là SOH:

+ Nếu đạt timeout = 1s và số lỗi lớn hơn 10 thì bên nhận hủy quá trình nhận. Nếu số lỗi nhỏ hơn 10 thì bên thu quay trở lại việc chờ gói tin đầu.

+ Nếu bên thu nhận tín hiệu ESC thì hủy bởi người dùng.

+Gói tin nhận được có kích thước 132byte sẽ xảy ra các trường hợp sau:

+ (Bad CRC) xảy ra lỗi CRC thì quay trở lại timeout.

+ Số thứ tự gói tin sai thì bên thu hủy truyền.

+ Mở file nhận được bị lỗi thì hủy truyền.

+ Không có tên file bên thu gửi ACK và thoát.

+ Nếu bên thu nhận được gói tin đầu tiên thành công sẽ gửi cho bên phát tín hiệu ACK và C để chuyển sang trạng thái tiếp theo.



b) Quá trình nhận gói tiếp theo

- Sau khi bên thu nhận được gói tin đầu tiên chứa tên đường dẫn file sau đó gửi tín hiệu ACK và C cho bên phát thì bên thu sẽ tiến hành việc đợi gói tin tiếp theo:

+ Nếu bên thu nhận được 2CAN thì sẽ hủy việc nhận gói tin.

+ Nếu quá trình chờ gói tin tiếp theo mà đạt giá trị timeout = 10 và số lỗi lớn hơn 10 thì thực hiện hủy nhận. Còn số lỗi nhỏ hơn 10 thì bên thu sẽ gửi ký tự “NAK” để yêu cầu gửi lại gói tin.

+ Nếu nhận được là tín hiệu “ESC” thì sẽ hủy quá trình nhận.

+ Nếu nhận được gói tin có ký tự đầu là SOH thì dữ liệu nhận là 128byte còn STX là 1024 byte:

+ Nếu giá trị timeout = 1s hoặc lỗi CRC sẽ trở lại quá trình timeout(10).

+ Nếu lỗi thứ tự gói bên nhận sẽ gửi tín hiệu “ACK” cho bên thu và chờ nhận lại gói tin.

+ Nếu ghi dữ liệu bị lỗi thì bên nhận hủy gói tin.

+ Bên thu nhận gói tin thành công và gửi ACK cho bên phát để tiếp tục quá trình nhận gói tin tiếp theo.



c) Quá trình nhận gói tin cuối cùng và kết thúc nhận.

Nếu bên thu nhận tín hiệu EOT thì bên thu sẽ gửi tín hiệu NAK và chờ tín hiêu EOT tiếp theo từ bên phát:

+ Nếu bên thu nhận được tín hiệu CAN và ESC thì bên thu sẽ hủy nhận.

+ Nếu quá trình chờ EOT đạt giá trị timeout = 5s và số lỗi lớn hơn 20 thì sẽ hủy truyền và ngược lại số lỗi nhỏ hơn 20 sẽ quay trở lại trạng thái chờ EOT.

+ Nếu bên thu nhận EOT thứ hai thì bên thu sẽ tiến hành truyền ACK và C và nhận gói tin cuối cùng.

Sau khi nhận gói tin cuối cùng sẽ gửi tín hiệu ACK và kết thúc chương trình.



# 3.2 Đề xuất tùy biến giao thức Y-MODEM phục vụ xác thực tự động.

## 3.2.1 Nguyên lý hoạt động của chữ ký số

Chữ ký số là một cơ chế xác thực cho phép người tạo ra thông tin (message creator) dùng khóa riêng của mình để xử lý khôi thông tin theo một thuật toán nào đó giúp người nhận thông tin kiểm chứng được tính toàn vẹn về nội dung và nguồn gốc của thông tin. Chữ ký số còn có công dụng ràng buộc giữa chủ thể với hành vi, nhằm ngăn chặn các tình huống từ chối hành vi (nonrepudiation) trong các giao dịch điện tử.

Chữ ký đơn giản được tạo ra bằng cách áp dụng một hàm băm lên thông gốc, sau đó mã hóa thông tin gốc dùng khóa riêng của người gởi rồi gởi kèm với mã băm. Bên nhận sẽ xác minh bằng cách dùng khóa công khai của bên gởi và hàm băm. Như vậy, chữ ký số là kỹ thuật xác thực thông tin dựa trên cơ sở mật mã bất đối xứng và các hàm băm bảo mật.

Tại sao phải dùng chữ ký số trong khi các cơ chế xác thực thông tin (message authentication) đã thực hiện chức năng xác thực nguồn gốc thông tin? Các cơ chế xác thực thông tin sử dụng các hàm băm một chiều có tác dụng bảo vệ thông tin trao đổi giữa hai đối tượng trao đổi thông tin khỏi sự xâm phạm của một đối tượng thứ 3, tuy nhiên nó không có tác dụng ngăn chặn được sự cố ý gian lận của một trong hai đối tượng chính. Ví dụ:

Bên A gởi một bản tin X cho bên B sử dụng một cơ chế xác thực nào đó, cơ chế này đảm bảo chỉ có A và B dùng chung một khoá bí mật K để tạo ra các mã xác thực từ thông tin gốc. Tuy nhiên, bên B có thể đổi bản tin X thành một bản tin Y, và với khóa bí mật K sẵn có, bên B hòan tòan có thể tạo ra thông tin xác thực mới để gắn vào Y, làm cho nó trở thành một bản tin hợp lệ mặc dù thực chất đây không phải là bản tin do bân A tạo ra.

Một ví dụ khác, bên A có thể từ chối xác nhận việc mình đã gởi bản tin X cho bên B, vì với các cơ chế xác thực như trên, bên B hoàn toàn có khả năng giả mạo thông tin đưa ra từ bên A.

Giống như một chữ ký thông thường (chữ ký bằng tay), một chữ ký số phải có đầy đủ các thuộc tính sau đây:

• Phải xác nhận chính xác người ký và ngày giờ phát sinh chữ ký.

• Phải xác thực nội dung thông tin ngay tại thời điểm phát sinh chữ ký.

• Phải có khả năng cho phép kiểm chứng bởi một người thứ 3 để giải quyết các tranh chấp nếu có.

Như vậy, chức năng của chữ ký số bao gồm chức năng của xác thực thông tin. Các yêu cầu đối với chữ ký số:

• Chữ ký phải gắn liền với khối thông tin cần được xác nhận (thông tin gốc).

• Chữ ký phải chứa thông tin nhận dạng riêng của người ký để tránh giả mạo và tránh phủ nhận.

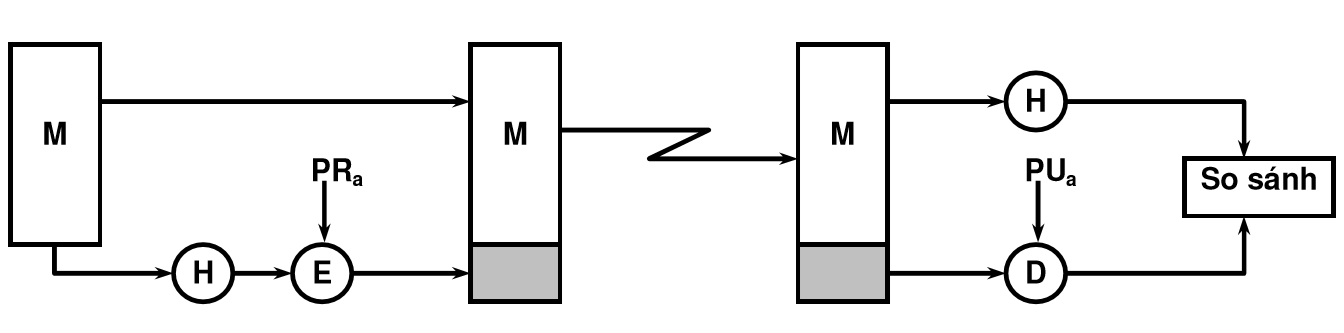
• Quy trình tạo ra chữ ký cũng như xác minh chữ ký phải đơn giản, nhanh chóng

• Chữ ký thông thể bị giả mạo bằng bất cứ cách nào.

• Có thể sao chép một bản sao của chữ ký dành cho mục đích lưu trữ

## 3.2.2 Thuật toán xác thực sử dụng chữ ký số RSA

Trong thuật toán xác thực thông tin dùng mật mã RSA, thông tin gốc được đưa vào hàm băm SHA để tạo ra mã băm (tức message digest) có kích thước cố định. Mã băm này sau đó được mã hóa (bằng thuật toán RSA) dùng khóa riêng của thực thể tạo thông tin (phía gởi). Kết quả của phép mã hóa được gắn vào thông tin gốc và gởi đi. Phía thu nhận được thông tin, tách phần mã băm ra khỏi thông tin gốc và giải mã nó bằng khóa công khai của phía gởi. Chú ý rằng khóa công khai là thông tin được công bố rộng rãi cho bất kỳ thực thể nào có quan tâm. Đồng thời, thông tin gốc cũng được đưa vào hàm băm để tính mã băm, sau đó đem so sánh với mã băm vừa nhận được. Nếu hai mã này giống nhau thì thông tin vừa nhận được chấp nhận như là thông tin hợp lệ



Thuật toán tạo chữ ký số RSA là:

-Tạo khoá công khai và khóa riêng tương ứng cho bên phát:

* Sinh ra hai số nguyên tố lớn ngẫu nhiên p và q cùng kích thước
* Tính n = pq và 
* Chọn số tự nhiên ngẫu nhiên e thỏa mãn điều kiện sau: 1 < e <  và USCLN(e, ) = 1
* Sử dụng thuật toán mở rộng Euclidean để tính số tự nhiên duy nhất sao cho 1 < d <  và ed  1(mod )

Khóa công khai của là (n,e) và khóa riêng là d

-Tạo chữ ký:

* Tính m= H(M) (H là hàm băm 256 bit)
* Tính s = mdmodn
* Chữ ký được tạo ra đó là s

-Kiểm chứng chữ ký:

* Bên thu nhận khóa công khai là (n,e)
* Tính m’= semodn
* Kiểm tra m’= m hay không nếu không sẽ không chấp nhận chữ ký

## 3.2.3 Bổ sung xác thực cho giao thức Y – Modem sử dụng chữ ký số

Tại pha khởi tạo của giao thức Y-MODEM có quá trình truyền gói tin đầu tiên có chứa tên đường dẫn ta có thể tận dụng yếu tố này để bổ sung thành pha xác thực. Ta tiến hành thực hiện các bước như sau:

a) Tại bên phát

- Tạo các thành phần khóa công khai (p,q)

- Tạo cặp khóa công khai (n, e) và khóa riêng d

- Tiến hành băm thông báo M thành mã băm H(M). Sau đó tạo ra chữ ký số dựa trên mã băm vừa tạo s = mdmodn.

- Tiến hành đóng gói gói tin bao gồm :

+ Thông báo rõ M

+ Chữ ký số s

+ Khóa công khai (n,e)

+ Tên file cần truyền

- Sau đó sẽ bên phát sẽ truyền gói tin đầu tiên cho bên thu và chờ phản hồi.



b) Bên nhận

- Sau khi nhận được gói tin đầu tiên bên thu sẽ tiến hành bóc tách gói tin:

+ Bên thu sẽ dùng khóa công khai (n,e) để kiểm tra chữ ký số nếu không giải mã được thì chứng tỏ thông tin gửi không chính xác và yêu cầu bên phát gửi lại. Nếu giải mã bên thu sẽ nhận được đoạn mã băm m, bên thu sẽ băm thông báo M’ nhận được thành m’.

+ Nếu m = m’ thì thông báo được xác thực và đảm bảo tính toàn vẹn.

+ Nếu m khác m’ thì thông báo không được xác thực hoặc bị thay đổi. Bên thu sẽ gửi NAK để yêu cầu bên phát truyền lại gói tin.



# 3.3 Xây dựng chương trình thực hiện giao thức tùy biến truyền tin trên kênh PSTN.

- Kết quả thực hiện và đánh giá.

# Kết luận