

# XÂY DỰNG MỘT THỦ TỤC TRUYỀN SỐ LIỆU SONG CÔNG

NGUYỄN HẢI CHÂU

*Viện Công nghệ Thông tin*

**Summary.** The paper presents the design and implementation of a full-duplex data communication protocol. The main purpose of the design is to build a basic tool for a network. In addition, the protocol can be used in point-to-point communication to increase the usage of transmission line in both two directions.

## I. ĐẶT VẤN ĐỀ

Các thủ tục truyền số liệu đã được thiết kế và cài đặt khá nhiều. Có thể kể đến các thủ tục thường được sử dụng như XMODEM, YODEM, KERMIT. Các thủ tục này đều hoạt động hữu hiệu trong các ứng dụng truyền tin điểm - điểm. Phần lớn các thủ tục này đều hoạt động theo chế độ bán song công, vì vậy chưa tận dụng được nhiều khả năng của các đường truyền song công. Các thủ tục điều khiển truyền thông trên mạng - thường nằm trong bộ thủ tục mạng - được cài đặt hoạt động theo chế độ song công (ví dụ thủ tục điều khiển truyền TCP trong bộ TCP/IP). Các thủ tục truyền số liệu hoạt động theo cả hai chế độ song công và bán song công đều đã được thiết kế, cài đặt và hoạt động tốt. Tuy vậy, chúng tôi thấy cần xây dựng thủ tục này vì các lý do sau:

- Cần phải xây dựng công cụ truyền tin ở mức thấp, trên cơ sở đó phát triển một mạng máy tính.

- Khi cần xây dựng mạng chuyên dụng để phục vụ các bài toán đặc thù cho từng cơ quan, xí nghiệp,... việc tạo công cụ từ mức thấp sẽ giúp cho việc giải quyết các bài toán trở nên gọn nhẹ mà vẫn đáp ứng được các yêu cầu đặt ra. Cũng có thể giải quyết các bài toán này dựa trên các mạng cài đặt các thủ tục như TCP/IP. Tuy vậy có một số điểm cần phải tính đến khi sử dụng các thủ tục này:

- + Giá thành của phần mềm và các trang bị kỹ thuật cần thiết khi sử dụng các bộ thủ tục này là khá cao.

- + Đối với các bài toán đặc thù, không nhất thiết phải sử dụng cả một bộ thủ tục lớn như vậy.

- + Chất lượng môi trường truyền tin ở Việt Nam chưa cao nên có thể là một yếu tố ảnh hưởng không tốt đến hoạt động của các thủ tục đó.

+ Một số công nghệ viễn thông hiện đại, có chất lượng cao đã được đưa vào sử dụng ở Việt Nam, song giá thuê bao và cước phí tương đối cao (tổng đài truyền số liệu X25 đã được lắp đặt và sử dụng ở Bru điện Việt Nam).

+ Một số thủ tục như XMODEM, YMODEM, KERMIT,... khi hoạt động ở môi trường truyền tin ở Việt Nam không có độ tin cậy như mong muốn.

+ Thủ tục song công sẽ tận dụng được nhiều hơn khả năng của các đường truyền song công.

Sau đây là cách xây dựng thủ tục.

## II. XÂY DỰNG VÀ HOẠT ĐỘNG CÁC THỦ TỤC

### 2.1. Khuôn dạng một gói tin

Căn cứ vào mục đích đặt ra, ta thấy thủ tục phải đáp ứng một số yêu cầu sau:

- Các yêu cầu trên mạng đòi hỏi thủ tục hoạt động theo chế độ song công.
- Phân biệt được các dữ liệu thuộc vào bài toán ứng dụng nào trên mạng trong cùng một phiên truyền.

- Để điều khiển quá trình song công dễ dàng hơn và để tăng độ tin cậy, các thông báo về điều khiển quá trình hoạt động của thủ tục như ACK, NAK, CAN,... sẽ là một gói tin (một số thủ tục như XMODEM, YMODEM chỉ dùng một byte để điều khiển quá trình).

Vì vậy, các gói tin được xây dựng gồm hai loại:

- Gói tin dữ liệu (ký hiệu là D) mang các dữ liệu thực sự cần chuyển tải.
- Gói tin điều khiển (ký hiệu là C) mang các thông tin điều khiển quá trình hoạt động của thủ tục. Các gói tin điều khiển gồm có:

- ACK: Xác nhận gói tin D vừa thu được là tốt
- NAK: Gói tin vừa thu được có lỗi
- EOF: Hết một bản tin
- EOT: Hết một phiên truyền
- CAN: Huỷ bỏ phiên truyền do lỗi không khắc phục được

Do các yêu cầu trên, chúng tôi đưa ra một khuôn dạng chung cho tất cả các loại gói tin của thủ tục như sau:

SON    LEN    MSG    PKT    TYPE    DATA...    CHK

trong đó, các trường có ý nghĩa sau:

SON: Bắt đầu một gói tin

LEN: Độ dài của gói tin tính từ trường LEN đến CHK. Việc xác định trường LEN cho phép truyền được cả bản tin text và nhị phân.

MSG: Số hiệu của bản tin được truyền

PKT: Số thứ tự của gói tin thuộc bản tin có số hiệu MSG. Khi số lượng gói tin trong một bản tin quá lớn, số thứ tự PKT của gói tin phải quay về vòng 0.

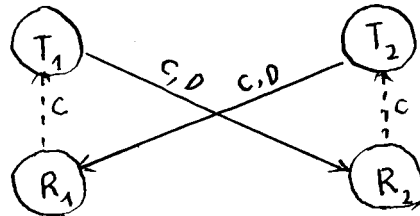
TYPE: Kiểu của gói tin : C hoặc D.

DATA: Dữ liệu của gói tin. Khi gói tin có kiểu C, dữ liệu sẽ là ACK, NAK, EOF,...

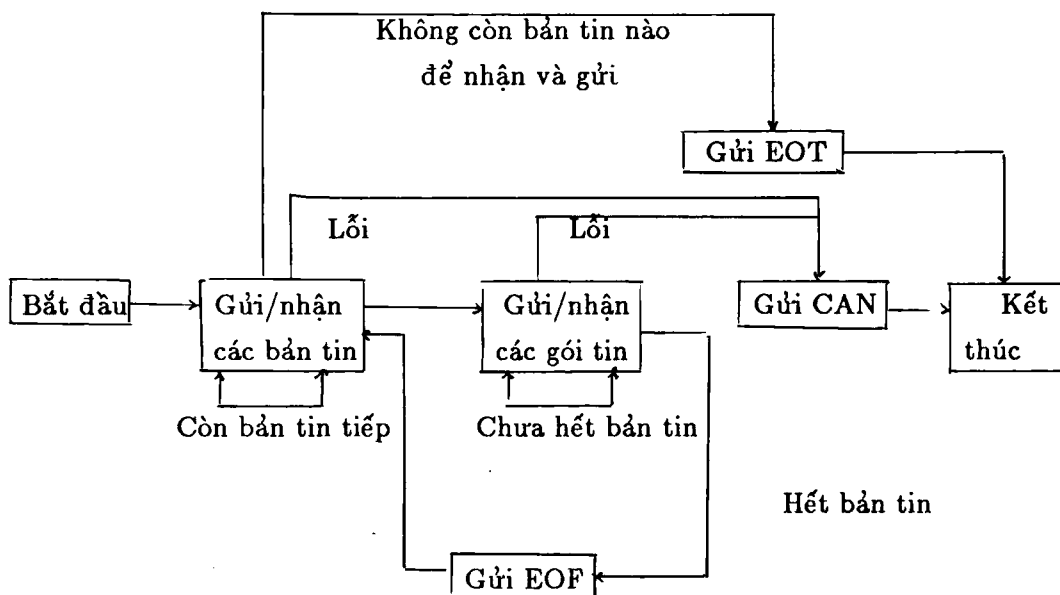
CKH: Trường kiểm tra sự chính xác của gói tin.

## 2.2. Điều khiển phiên liên lạc

Trong các thủ tục bán song công, bên phát điều khiển phiên liên lạc. Bên thu thụ động, chịu sự điều khiển của bên phát. Với các thủ tục song công, điều khiển phiên liên lạc có phức tạp hơn. Ngoài việc đảm bảo cho quá trình thu phát hoạt động tốt, chúng ta còn phải đảm bảo được sự liên lạc và đồng bộ bên trong của mỗi bên tham gia phiên liên lạc. Trong thủ tục này điều khiển phiên liên lạc được xây dựng như sau:



Mỗi bên tham gia liên lạc có hai quá trình hoạt động song song là  $T_i$  và  $R_i$ ,  $T_i$  là quá trình truyền,  $R_i$  là quá trình nhận ( $i = 1, 2$ ). Khi  $R_i$  nhận được một gói tin  $D$  và trả lời ACK, NAK cho  $T_j$  ( $i \neq j$ )  $R_i$  không được trực tiếp truyền gói tin  $C$  mà yêu cầu  $T_j$  truyền gói đó qua liên lạc  $T - i - R_i$ . Phiên liên lạc song công được xem là hai quá trình thu phát bán song công hoạt động đồng thời là  $(T_1, R_2)$ ,  $(T_2, R_1)$  và kết thúc khi cả  $(T_1, R_2)$ ,  $(T_2, R_1)$  kết thúc. Nói cách khác, phiên liên lạc kết thúc khi không còn gói tin  $D$  nào truyền theo hướng  $T_1 \rightarrow R_2$  và  $T_2 \rightarrow R_1$ . Các gói tin điều khiển EOT và CAN đóng vai trò quan trọng trong điều khiển quá trình. Chức năng của hai gói này là thông báo kết thúc phiên truyền. Gói EOT được phát đi khi liên lạc kết thúc bình thường, còn CAN được phát để huỷ bỏ liên lạc do lỗi. Trong điều khiển phiên liên lạc, một yếu tố quan trọng nữa là xác định được trạng thái của từng bên tham gia. Theo cách xây dựng thủ tục, trạng thái của mỗi bên tham gia được mô tả qua sơ đồ sau:



### 2.3. Điều khiển dòng dữ liệu

Như đã nói ở trên, chúng ta xem phiên liên lạc song công là hai quá trình thu phát bán song công  $(T_1, R_2)$  và  $(T_2, R_1)$  hoạt động song song. Mỗi phiên  $(T_i, R_j)$ ,  $i \neq j$  điều khiển dòng dữ liệu theo kiểu stop - end - wait, tức là  $T_i$  phát đi một gói tin D, chờ  $R_j$  trả lời rồi phát gói tin tiếp theo. Ngoài ra  $T_i$  còn phát các gói tin C theo yêu cầu của  $R_i$ . Như vậy các gói tin C và D được  $T_i$  phát đi xen kẽ trên đường truyền. Trong điều khiển dòng dữ liệu, chỉ có các gói tin D sau khi phát đi mới có xác nhận ACK, NAK còn các gói tin C không được xác nhận.  $R_j$  sau khi nhận một gói tin sẽ lọc các gói D đưa vào byffer nhận và chuyển các gói C cho  $T_j$ . Để tránh mất mát dữ liệu khi truyền, các gói tin D được đánh số thứ tự  $0, 1, 2, \dots, n$ . Khi  $n$  vượt quá giới hạn của trường PKT, số thứ tự được đánh lại từ 0.

Nhiều bản tin khác nhau có được truyền trong cùng một phiên liên lạc. Các bản tin không nhất thiết phải truyền nối tiếp nhau hết bản tin này đến bản tin khác. Các gói tin của các bản tin có thể được truyền xen kẽ nhau. Đây là một điểm khác so với các thủ tục YMODEM và KERMIT. Trong hai thủ tục này, các bản tin phải được truyền tuần tự. Chúng tôi đưa ra cơ chế điều khiển dòng dữ liệu này vì:

- Khi xử lý các ứng dụng trên mạng với các độ ưu tiên khác nhau, các bản tin được truyền xen kẽ nhau giúp cho việc xử lý độ ưu tiên dễ dàng hơn. Chẳng hạn khi một bản tin có độ ưu tiên cao đến sau sẽ không phải chờ bản tin có độ ưu tiên thấp được truyền hết mà có thể chen hàng ngay.

- Khi phiên truyền kết thúc bất thường do lỗi, chúng ta không phải truyền lại toàn bộ bản tin mà chỉ những phần chưa được phát.

#### 2.4. Kiểm soát lỗi

Trong thực tế, lỗi trong truyền tin là không thể tránh khỏi. Có nhiều nguyên nhân gây ra lỗi, ví dụ: nhiễu trên đường truyền, hai bên tham gia liên lạc không đồng bộ được với nhau,... Cách kiểm soát lỗi của thủ tục được xây dựng như sau:

- Khi các gói tin D bị sai hỏng hoặc mất:  $R_j$  có thể phát hiện được nhờ vào cơ chế phát hiện lỗi trong gói tin (sẽ trình bày trong phần 5) hoặc căn cứ theo số thứ tự của gói tin. Sau khi phát hiện được lỗi  $R_j$  sẽ yêu cầu  $T_j$  phát lại gói tin lỗi cho đến khi nhận được gói tin tốt. Tuy vậy nếu số lần phát lại vượt quá giới hạn cho phép,  $T_i$  sẽ phát CAN để bỏ phiên truyền.

- Khi các gói tin ACK, NAK bị sai thì  $T_i$  cũng phát lại như trên.

- Khi ACK, NAK, CAN, EOT bị mất trong quá trình liên lạc,  $T_i$  sẽ phải chờ một thời gian  $t$  rồi thực hiện một hành động đáp ứng thích hợp (truyền lại, hủy bỏ...).

Đặc biệt khi gói EOF bị mất, hoặc sai: bên thu  $R_j$  không xác định được bản tin đã kết thúc hay chưa. Để tránh trường hợp này chúng tôi giải quyết vấn đề này như sau:

Gọi  $1_p$  là độ dài phần DATA của một gói tin. Trong phiên liên tục,  $1_p$  được cố định. Như vậy tất cả các gói tin đều có độ dài như nhau, trừ gói cuối cùng. Gọi  $1$  là độ dài phần DATA của gói tin cuối cùng. Ta có:  $1 \leq 1_p$ . Xét hai trường hợp xảy ra:

- a)  $1 < 1_p \Leftrightarrow 1_m \bmod 1_p \neq 0$ , trong đó  $1_m$  là độ dài bản tin chứa gói tin. Căn cứ vào  $1 < 1_p$ , nếu EOF bị mất thì ta vẫn biết được bản tin đã kết thúc vì gói vừa nhận được là gói cuối cùng.

- b)  $1_p = 1 \Leftrightarrow 1_m \bmod 1_p = 0$ . Trong trường hợp này ta phát thêm một gói tin có độ dài phần DATA bằng không. Như vậy khi EOF bị mất ta lại quay về trường hợp a) vì  $0 = 1 < 1_p$ .

#### 2.5. Phát hiện lỗi

Lỗi trong gói tin được phát hiện nhờ trường hợp kiểm tra CHK ở cuối mỗi gói. Khi phát một gói tin,  $T_i$  sẽ dựa vào byte dữ liệu có trong gói tin, tính từ trường LEN đến DATA để tính ra giá trị tương ứng của CHK theo một cách nào đó. Khi  $R_j$  nhận được gói tin, quá trình tính toán trên được lặp lại. Nếu kết quả tính toán trùng với trường CHK của gói tin thì gói tin được xem là tốt và ngược lại. Các phương pháp tính toán để phát hiện lỗi hay được dùng là:

- Checksum

- Mã dư thừa vòng (cyclic redundancy check)

Trong thủ tục này, chúng tôi lựa chọn phương pháp thứ nhất:

$$CHK = d_1 \text{ XOR } d_2 \dots \text{ XOR } d_n,$$

với  $d_i$ ,  $i = 1, 2, \dots, n$  là các byte trong gói tin từ trường LEN đến trường DATA, XOR là phép hoặc loại trừ. Phương pháp này có ưu điểm là đơn giản và hiệu quả.

### III. ĐÁNH GIÁ

Sau đây chúng ta sẽ so sánh khả năng của thủ tục khi hoạt động ở chế độ song công và bán song công.

Giả sử hai bên tham gia quá trình liên lạc là  $A_1$ ,  $A_2$ . Gọi số lượng các gói tin cần truyền tương ứng của hai bên là  $n_1, n_2$  trong đó  $n_1, n_2 \geq 0$ . Gọi  $t$  là thời gian cần thiết để truyền xong một gói tin. Giả sử rằng  $t$  không phụ thuộc vào chế độ hoạt động của thủ tục là song công hay bán song công.

Như vậy, nếu hoạt động theo chế độ bán song công, chúng ta cần một khoảng thời gian

$$t_H = t(n_1 + n_2)$$

để truyền xong toàn bộ dữ liệu.

Khi hoạt động trong chế độ song công, thời gian truyền là

$$t_F = at \max(n_1, n_2)$$

với  $a$  là hệ số trễ sinh ra khi thủ tục hoạt động theo chế độ song công. Ta nhận thấy  $a \geq 1$  vì:

- Hoạt động song công đòi hỏi việc xử lý phức tạp hơn bán song công.
- Các quá trình liên lạc bên trong cá bên tham gia liên lạc  $t_i - R_i$  cũng là một yếu tố gây trễ.

Ta giả sử  $n_1 \geq n_2$  và  $n_1 > 0$ . Khi đó:

$$t_F = at \max(n_1, n_2) = an_1 t.$$

Xét tỷ số  $t_H/t_F$

$$\frac{t_H}{t_F} = \frac{t(n_1 + n_2)}{an_1 t} = \frac{1}{a} \left(1 + \frac{n_2}{n_1}\right).$$

Xét hai trường hợp sau đây

$$a) \quad \frac{1}{a} \left(1 + \frac{n_2}{n_1}\right) > 1 \Leftrightarrow \frac{n_2}{n_1} > a - 1. \quad (1)$$

Trong trường hợp này hoạt động theo chế độ song công có lợi hơn bán song công

$$b) \quad \frac{1}{a} \left(1 + \frac{n_2}{n_1}\right) \leq 1 \Leftrightarrow \frac{n_2}{n_1} \leq a - 1. \quad (2)$$

Trường hợp này cho ta thấy hoạt động theo bán song công lại có lợi hơn. Có nhiều nguyên nhân, trong đó chúng tôi nhận thấy:

- Khả năng của đường truyền có thể bị giảm sút khi hoạt động ở chế độ song công.
- Khả năng xử lý của chương trình cài đặt bị hạn chế.
- Khi xử dụng các hệ điều hành đa nhiệm, cấu hình của máy tính được cài đặt không đủ mạnh để phục vụ các yêu cầu của hệ điều hành dẫn đến các chương trình cài đặt trên đó bị ảnh hưởng.

Như vậy chúng ta có thể thấy rằng không phải lúc nào hoạt động theo chế độ song công cũng có lợi hơn bán song công. Để quyết định khi nào sử dụng chế độ song công và khi nào sử dụng chế độ bán song công, chúng tôi đưa ra một cách ước lượng các tham số cho các công thức (1), (2) như sau:

1. Các tham số  $n_i$ , ( $i = 1, 2$ ) tính được dễ dàng qua công thức

$$n_i = \sum_{j=1}^k \left( \left\lfloor \frac{f_{ij}}{1_p} \right\rfloor + 1 \right)$$

trong đó:

- $k$  là số bản tin cần truyền
- $f_{ij}$  là độ dài các bản tin cần truyền
- $1_p$  là độ dài DATA của một gói tin

2. Hệ số  $a$  có thể ước lượng nhờ vào thống kê. Sau mỗi phiên liên lạc, các tham số sau đây được lưu lại để trên cơ sở đó ước lượng  $a$ :

- + Tốc độ truyền tin ( $s$ )
- + Số lượng các gói tin cần truyền của cả hai bên ( $n_1, n_2$ )
- + Thời gian hoạt động của bên liên lạc ( $T$ )
- + Số lượng các gói tin phải truyền lại do lỗi ( $e_1, e_2$ ). Khi đó ta có tập hợp các tham số  $S = \{s, n_1, n_2, T, e_1, e_2\}$ . Ứng với mỗi loại tốc độ  $s$ , ta có thể tính  $a$  như sau:

$$a_s = \frac{1}{|S|} \sum_{i=1}^{|S|} \left[ (n_1^i + e_1^i + n_2^i + e_2^i)t - T_i \right] + 1, \quad (3)$$

với  $T_i, n_1^i, n_2^i, e_1^i, e_2^i \in S$ ,  $|S|$  là số phần tử của tập  $S$ ,  $t$  là thời gian cần thiết để truyền một gói tin.

**IV. KẾT LUẬN**

Trong bài này đã trình bày cách xây dựng một thủ tục truyền số liệu hoạt động theo chế độ song công. Thủ tục đáp ứng được các yêu cầu đặt ra và đã hoạt động tốt trong thực tế. Ngoài chức năng làm công cụ truyền tin cơ sở cho một mạng máy tính sau này, thủ tục có thể cài đặt để hoạt động trong truyền số liệu điểm - điểm. Các ước lượng (1), (2), (3) cho thấy có thể căn cứ vào một số các tham số của các bên tham gia liên lạc để quyết định sử dụng phương thức truyền song công sao cho có lợi nhất.

**TÀI LIỆU THAM KHẢO**

1. Stalling W., Data computer communication, 3rd edition, Macmillan Publ. Comp., 1991.
2. Abrahamson N., Kuo F. F., (edited) Computer communication networks, Prentice-Hall Inc. 1973.
3. Martin J. System analysis for data transmission, Prentice-Hall Inc. 1972.