

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**NHẬP MÔN MẠNG MÁY TÍNH**

**LỚP: IT005.O118**

**BÁO CÁO BÀI TẬP 6 – NHÓM 12**

*Giảng viên hướng dẫn: ThS. Trần Mạnh Hùng*

**NoName** – “Không tên nhưng không bao giờ vô danh”

## MỤC LỤC

<b>I. DANH SÁCH THÀNH VIÊN .....</b>	<b>1</b>
<b>II. BÁO CÁO BÀI TẬP 6.....</b>	<b>2</b>
<i>Problem 23.....</i>	<i>2</i>
<i>Problem 24.....</i>	<i>3</i>
<i>Problem 27.....</i>	<i>4</i>
<i>Problem 28.....</i>	<i>6</i>
<i>Problem 31.....</i>	<i>7</i>
<b>III. NHẬN XÉT .....</b>	<b>9</b>
<b>IV. THẮC MẮC .....</b>	<b>9</b>
<b>V. NGUỒN THAM KHẢO .....</b>	<b>10</b>

**I. DANH SÁCH THÀNH VIÊN**

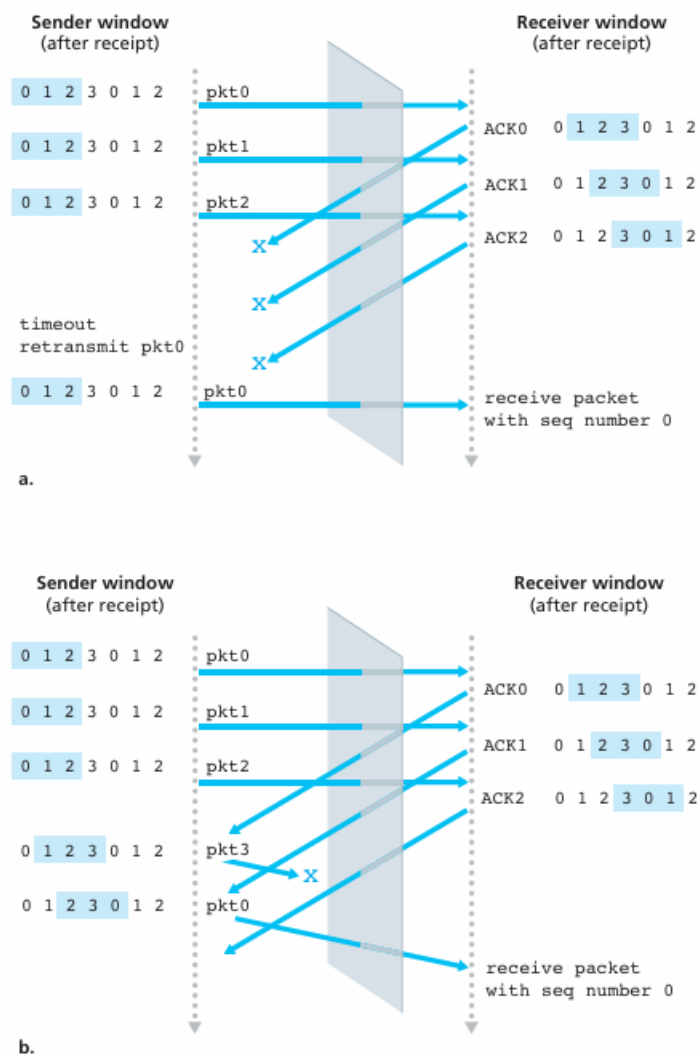
MSSV	Họ và tên	Phân công		Đánh giá
22521301	Mai Văn Tân (nhóm trưởng)	Trình bày báo cáo, P24	Cùng kiểm tra lại tất cả các câu sau khi hoàn thành đáp án.  Nhận xét, nêu thắc mắc tồn đọng.	100%
22520512	Nguyễn Bá Hưng	P24, P27		100%
22521539	Nguyễn Thị Trinh	P27, P28		100%
22521394	Trần Ý Thiên	P28, P29		100%
22520518	Nguyễn Thanh Hùng	P29, P31		100%
22520108	Nguyễn Gia Bảo	P23, P31		100%

## II. BÁO CÁO BÀI TẬP 6

### Problem 23

Consider the GBN and SR protocols. Suppose the sequence number space is of size  $k$ . What is the largest allowable sender window that will avoid the occurrence of problems such as that in Figure 3.27 for each of these protocols?

*Giả sử sequence number space có kích thước là  $k$  trong các giao thức GBN (Go-Back-N) và SR (Selective Repeat). Xác định sender window lớn nhất cho mỗi giao thức mà không gặp các vấn đề như trong hình 3.27?*



**Figure 3.27** ♦ SR receiver dilemma with too-large windows: A new packet or a retransmission?

Để xác định sender window lớn nhất trong các giao thức GBN và SR mà không gặp các vấn đề như trong hình 3.27, tức là chúng ta phải **tránh tình trạng cạnh của receiver window** (là cửa sổ có số thứ tự cao nhất) bao quanh không gian của sequence number và chồng lên cạnh cuối cùng (cửa sổ có số thứ tự thấp nhất)

➔ Điều này có nghĩa là không gian của sequence number **phải đủ lớn để chứa cả receiver and sender windows.**

Trong cả GBN và SR, **không gian sequence number** có kích thước là **k**, đại diện cho tất cả các số thứ tự có thể được sử dụng. Giả sử chúng ta có một **receiver window** với kích thước **w**, tức là nó có thể chứa w gói tin.

**Người nhận sẽ chờ là m.** Do đó, **cửa sổ của người nhận** sẽ từ **[m, m+w-1]**. Đồng thời, người nhận gói tin có số thứ tự nhỏ nhất đã nhận và gửi ACK cho gói tin **m-1** và **w-1 gói tin trước đó**. Nếu tất cả các ACK này chưa được người gửi nhận được, có thể tồn tại các ACK đang lan truyền trong mạng với giá trị từ **m-w đến m-1**. Do đó, cửa sổ của người gửi sẽ là từ **m-w đến m-1**.

Để tránh tình huống cửa sổ của người nhận chồng lên cửa sổ của người gửi, không gian số thứ tự phải đủ lớn để chứa cả cửa sổ của người nhận và cửa sổ của người gửi mà không có sự chồng chéo. Điều đó có nghĩa là không gian số thứ tự phải ít nhất là **k ≥ 2w**.

Vì vậy, để tránh vấn đề trong hình 3.27, sender window lớn nhất trong cả GBN và SR là w, và **không gian số thứ tự phải có kích thước ít nhất là 2w**.

### Problem 24

Answer true or false to the following questions and briefly justify your answer:

- With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
- With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.
- The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.
- The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

*Trả lời đúng hoặc sai cho các câu hỏi sau và giải thích ngắn gọn lý do:*

*a. Với giao thức SR, có khả năng cho bên gửi nhận được một ACK cho một gói dữ liệu nằm ngoài cửa sổ hiện tại của nó.*

**Đúng.** Có khả năng nhận được các xác nhận nằm ngoài cửa sổ hiện tại trong giao thức Selective Repeat vì có trường hợp xác nhận từ bên nhận bị trễ và không đến được bên gửi trong khoảng thời gian đếm ngược được đặt và gửi lại, trong khi trong thời gian đó, xác nhận gói trễ có thể đến đến bên gửi và cửa sổ được mở rộng thêm. Sau đó, các xác nhận của các thông điệp được gửi lại cũng đến, làm cho xác nhận nằm ngoài kích thước cửa sổ hiện tại.

*b. Với GBN, có khả năng cho bên gửi nhận được một ACK cho một gói dữ liệu nằm ngoài cửa sổ hiện tại của nó.*

**Đúng.** Vì trong giao thức Go Back N, có khả năng nhận được các xác nhận nằm ngoài cửa sổ hiện tại vì có trường hợp xác nhận từ bên nhận bị trễ và không đến được bên gửi trong khoảng thời gian đếm ngược được đặt và gửi lại, trong khi trong thời gian đó, xác nhận gói trễ có thể đến đến bên gửi và cửa sổ được mở rộng thêm. Sau đó, các xác nhận của các thông điệp được gửi lại cũng đến, làm cho xác nhận nằm ngoài kích thước cửa sổ hiện tại.

*c. Giao thức alternating-bit giống như giao thức SR với kích thước cửa sổ của bên gửi và bên nhận là 1.*

**Đúng.** Thông thường, trong giao thức alternating bit, số thứ tự luân phiên giữa 0 và 1. Trong khi đó, trong Giao thức Selective Repeat, kích thước cửa sổ của bên gửi và bên nhận sẽ không trùng khớp. Do đó, trong tình huống kích thước cửa sổ của bên gửi và bên nhận là 1, Giao thức Alternating bit được cho là giống với Giao thức SR.

*d. Giao thức alternating-bit giống như giao thức GBN với kích thước cửa sổ của bên gửi và bên nhận là 1.*

**Đúng.** Trong trường hợp kích thước cửa sổ là 1, khái niệm về gói tin nằm ngoài thứ tự bị loại bỏ, làm cho chức năng của giao thức alternating bit giống như Giao thức Go Back N.

### **Problem 27**

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is

302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

- In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?
- If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?
- If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?
- Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

*Host A và B đang giao tiếp qua một kết nối TCP, và Host B đã nhận được từ A tất cả các byte cho đến byte 126. Giả sử Host A sau đó gửi hai segment đến Host B liên tiếp. Segment đầu tiên và thứ hai chứa lần lượt 80 và 40 byte dữ liệu. Trong Segment đầu tiên, số thứ tự là 127, số cổng nguồn là 302 và số cổng đích là 80. Host B gửi một ACK mỗi khi nhận được một đoạn tin từ Host A.*

*a. Trong segment thứ hai được gửi từ Host A đến B, số thứ tự, số cổng nguồn và số cổng đích là gì?*

Trong segment thứ hai từ Host A đến B, số thứ tự là 207, số cổng nguồn là 302 và số cổng đích là 80.

*b. Nếu segment đầu tiên đến trước segment thứ hai, trong ACK của segment đầu tiên được nhận, số ACK, số cổng nguồn và số cổng đích là gì?*

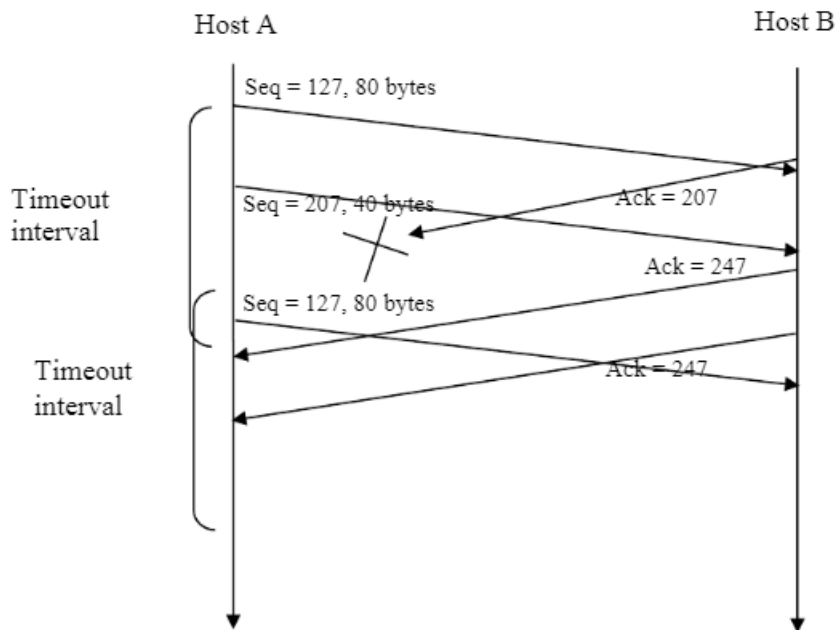
Nếu segment thứ nhất đến trước segment thứ hai, trong ACK của segment đầu tiên đến, số ACK là 207, số cổng nguồn là 80 và số cổng đích là 302.

*c. Nếu segment thứ hai đến trước segment đầu tiên, trong ACK của segment đầu tiên được nhận, số ACK là gì?*

Nếu segment thứ hai đến trước segment thứ nhất, trong ACK của segment đầu tiên đến, số ACK là 127, cho thấy nó vẫn đang chờ đợi từ byte 127 trở đi.

*d. Giả sử hai segment được gửi bởi A đến B theo thứ tự. ACK đầu tiên bị mất và ACK thứ hai đến sau khoảng thời gian chờ đợi đầu tiên. Vẽ sơ đồ thời gian, cho thấy các segment này và tất cả các segment và ACK khác được gửi đi. (Giả sử không có mất gói tin) Đối với mỗi segment trong hình của bạn, cung cấp số thứ tự và số byte dữ liệu; đối với mỗi ACK mà bạn thêm vào, cung cấp số ACK.*

Sơ đồ:



### Problem 28

Host A and B are directly connected with a 100 Mbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection. Host A can send its application data into its TCP socket at a rate as high as 120 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 50 Mbps. Describe the effect of TCP flow control.

*Host A và B kết nối trực tiếp với đường link 100 Mbps. Tồn tại 1 kết nối TCP giữa 2 host, host A gửi đến host B 1 file rất lớn quá tải với kết nối. Host A có thể gửi dữ liệu ứng dụng của nó vào trong TCP socket của chính nó ở tỉ lệ cao là 120 Mbps nhưng host B vẫn nhận biết được buffer chứa TCP của nó ở tỉ lệ tối đa là 50 Mbps. Hãy mô tả kết quả của TCP flow control*



Vì khả năng của đường link là chỉ 100Mbps, vì thế tốc độ gửi của host A tối đa là 100 Mbps. Nhưng, host A gửi dữ liệu (data) vào buffer nhanh hơn là Host B xóa dữ liệu khỏi buffer. Tốc độ đầy của buffer là khoảng 40Mbps. Khi buffer đầy, Host B gửi tín hiệu tới Host A để dừng việc gửi data bằng cách cho  $RcvWindow = 0$ . Host A sau đó dừng gửi cho đến khi nó nhận một TCP segment có  $RcvWindow > 0$ . Do đó, Host A sẽ liên tục lặp quá trình dừng và gửi như là 1 chức năng đối với các giá trị  $RcvWindow$  mà nó nhận được từ Host B. Trung bình thì tốc độ lâu dài của kết nối để Host A gửi dữ liệu tới Host B là không quá 60Mbps.

### Problem 31

Suppose that the five measured **SampleRTT** values (see Section 3.5.3) are 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Compute the **EstimatedRTT** after each of these **SampleRTT** values is obtained, using a value of  $\alpha = 0.125$  and assuming that the value of **EstimatedRTT** was 100 ms just before the first of these five samples were obtained. Compute also the **DevRTT** after each sample is obtained, assuming a value of  $\beta = 0.25$  and assuming the value of **DevRTT** was 5 ms just before the first of these five samples was obtained. Last, compute the TCP **TimeoutInterval** after each of these samples is obtained.

*Giả sử năm giá trị **SampleRTT** được đo (xem Phần 3.5.3) là 106 ms, 120 ms, 140 ms, 90 ms và 115 ms. Tính toán **EstimatedRTT** sau khi thu được mỗi giá trị **SampleRTT** này, sử dụng giá trị  $\alpha = 0,125$  và giả sử rằng giá trị **EstimatedRTT** là 100 mili giây ngay trước khi thu được sample đầu tiên trong số năm sample này. Cũng tính toán **DevRTT** sau khi lấy được mỗi sample, giả sử giá trị  $\beta = 0,25$  và giả sử giá trị **DevRTT** là 5 ms ngay trước khi lấy được sample đầu tiên trong số năm sample này. Cuối cùng, tính toán TCP **TimeoutInterval** sau khi lấy được từng sample này.*

$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * | \text{SampleRTT} - \text{EstimatedRTT} |$$

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

Sau khi thu được giá trị **SampleRTT** đầu tiên 106 ms:

$$\text{DevRTT} = 0.75 * 5 + 0.25 * | 106 - 100 | = 5.25 \text{ ms}$$

$$\text{EstimatedRTT} = 0.875 * 100 + 0.125 * 106 = 100.75 \text{ ms}$$

$$\text{TimeoutInterval} = 100.75 + 4 * 5.25 = 121.75 \text{ ms}$$

Sau khi thu được giá trị **SampleRTT** 120 ms:

$$\text{DevRTT} = 0.75 * 5.25 + 0.25 * |120 - 100.75| = 8.75 \text{ ms}$$

$$\text{EstimatedRTT} = 0.875 * 100.75 + 0.125 * 120 = 103.16 \text{ ms}$$

$$\text{TimeoutInterval} = 103.16 + 4 * 8.75 = 138.16 \text{ ms}$$

Sau khi thu được giá trị **SampleRTT** 140ms:

$$\text{DevRTT} = 0.75 * 8.75 + 0.25 * |140 - 103.16| = 15.77 \text{ ms}$$

$$\text{EstimatedRTT} = 0.875 * 103.16 + 0.125 * 140 = 107.76 \text{ ms}$$

$$\text{TimeoutInterval} = 107.76 + 4 * 15.77 = 170.84 \text{ ms}$$

Sau khi thu được giá trị **SampleRTT** 90ms:

$$\text{DevRTT} = 0.75 * 15.77 + 0.25 * |90 - 107.76| = 16.27 \text{ ms}$$

$$\text{EstimatedRTT} = 0.875 * 107.76 + 0.125 * 90 = 105.54 \text{ ms}$$

$$\text{TimeoutInterval} = 105.54 + 4 * 16.27 = 170.62 \text{ ms}$$

Sau khi thu được giá trị **SampleRTT** 115ms:

$$\text{DevRTT} = 0.75 * 16.27 + 0.25 * |115 - 105.54| = 14.57 \text{ ms}$$

$$\text{EstimatedRTT} = 0.875 * 105.54 + 0.125 * 115 = 106.72 \text{ ms}$$

$$\text{TimeoutInterval} = 106.72 + 4 * 14.57 = 165 \text{ ms}$$

### III. NHẬN XÉT

Thông qua bài báo cáo, chúng em đã nắm rõ hơn các kiến thức liên quan đến giao thức TCP và flow control trong môi trường mạng, cụ thể là:

1. Hiểu về cách quản lý sender window, các xác định kích thước sender window lớn nhất mà không gặp vấn đề.
2. Hiểu cơ bản về cách TCP flow control hoạt động giữa hai hosts, biết cách tính toán và áp dụng các thông số như RcvWindow, EstimatedRTT, DevRTT, TimeoutInterval để quản lý truyền dữ liệu.
3. Hiểu rõ hơn về sự khác biệt giữa các giao thức như Go-Back-N, Selective Repeat và Alternating-Bit Protocol.

### IV. THẮC MẮC

Nhóm chúng em chưa có bất kỳ thắc mắc nào.

## V. NGUỒN THAM KHẢO

1. Slide bài giảng môn học
2. Computer Networking: A Top-Down Approach, 6th Edition By Kurose and Ross
3. <https://www.educative.io/answers/how-to-compute-devrtt-estimated-rtt-time-out-interval-in-ccn>

**HẾT**