**BÀI TẬP 6**



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**







**NHẬP MÔN MẠNG MÁY TÍNH**

**LỚP: IT005.O118**

**Nhóm: CYBER SQUAD**

**Học ít hiểu nhiều**



**MỤC LỤC**

**[BẢNG ĐÁNH GIÁ THÀNH VIÊN. 2](#_Toc24561)**

**[P23: 3](#_Toc31502)**

**[P24: 3](#_Toc10396)**

**[P27: 4](#_Toc24556)**

**[P28: 6](#_Toc1023)**

**[P31: 7](#_Toc23535)**

**[BÀI HỌC RÚT RA 9](#_Toc24854)**

**[NGUỒN THAM KHẢO: 10](#_Toc24166)**

**BẢNG ĐÁNH GIÁ THÀNH VIÊN.**

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Tên thành viên** | **Phân chia công việc** | **Đánh giá** |
| 22521060 | Lê Minh Nhựt(C) | P31 | 100% |
| 22520195 | Trần Đình Khánh Đăng | P23 | 100% |
| 22521189 | Thái Ngọc Quân | P28 | 100% |
| 22521078 | Bùi Nhật Phi | P27 | 100% |
| 22520127 | Võ Ngọc Bảo | P24 | 100% |

**P23:**

**Xét giao thức GBN và SR, giả sử không gian seq number có kích thước là k thì cửa sổ truyền lớn nhất có khả năng tránh những vấn đề xảy ra như bên trên Figure 3.27 ứng với những giao thức kể trên là bao nhiêu?**

Giả sử **seq number** bé nhất mà bên *máy nhận* đang chờ là gói **m**. Trong trường hợp này, cửa sổ là **[m, m+w-1]** và đã nhận được gói **m-1** với **w-1** trước đó, lúc mà **w** là kích thước cửa sổ. Nếu *máy gửi* chưa nhận bất kì **ACK** của gói **w** nào thì **ACK message** với vùng giá trị **[m-w, m-1]** có thể đang được truyền lại.

Nếu *máy gửi* chưa nhận được bất kì **ACK** nào của các **ACK number** kể trên, thì cửa sổ gửi lúc này là **[m-w, m-1]**.

Do đó, cận dưới của cửa sổ gửi là **m-w**, và cận trên của cửa sổ nhận là **m+w-1**. Vậy, để những gói tin gửi từ *máy nhận* không bị ghi chồng với những gói của *máy gửi*, không gian **seq number** phải đủ lớn để chứa **2w seq number**.

**Kết luận:** Không gian **seq number** phải đạt tối thiểu là **2w**, hay **k >= 2w**.

**P24:**

**Trả lời đúng hoặc sai cho các câu hỏi sau và giải thích ngắn gọn cho câu trả lời.**

1. ***Với giao thức SR, người gửi có thể nhận được ACK cho một gói tin nằm ngoài cửa sổ hiện tại của nó***

**Đúng**. Giả sử *bên gửi* có kích thước cửa sổ là *3* và gửi các gói *1, 2, 3* tại **t0**. Tại **t1** (**t1** > **t0**) *bên nhận* **ACK** *1, 2, 3*. Tại **t2** (**t2** > **t1**) *bên gửi* hết thời gian chờ và gửi lại *1, 2, 3*. Tại **t3** *bên nhận* nhận được các bản sao và xác nhận lại *1, 2, 3*. Tại **t4** *bên gửi* nhận được **ACK** mà *bên nhận* đã gửi tại **t1** và đẩy cửa sổ của mình lên *4, 5, 6*. Tại **t5** *bên gửi* nhận được **ACK** *1, 2, 3* *bên nhận* đã gửi tại **t2**. Các **ACK** này nằm ngoài cửa sổ của nó.

1. ***Với GBN, người gửi có thể nhận được ACK cho một gói tin nằm ngoài cửa sổ hiện tại của nó.***

**Đúng**. Kịch bản giống với câu a.

1. **Giao thức bit xen kẽ giống như giao thức SR với người gửi và kích thước cửa sổ máy thu là 1.**

**Đúng**. Với kích thước cửa sổ là *1*, **SR**, **GBN** và giao thức bit xen kẽ về mặt chức năng tương đương nhau. Kích thước cửa sổ *1* ngăn cản khả năng xuất hiện các gói không theo thứ tự (trong cửa sổ). Một **ACK** tích lũy chỉ đơn giản là một **ACK** thông thường trong tình huống này, vì nó chỉ có thể tham chiếu đến gói duy nhất trong cửa sổ.

1. **Giao thức bit xen kẽ giống như giao thức GBN với người gửi và kích thước cửa sổ máy thu là 1.**

**Đúng**. Đã giải thích ở câu c.

**P27:**

**Máy chủ A và B đang liên lạc qua kết nối TCP, B đã nhận được từ A tất cả các byte cho đến byte 126. Máy chủ A sau đó gửi hai phân đoạn liên tiếp tới B. Chúng lần lượt chứa 80 và 40 byte dữ liệu. Trong phân đoạn đầu tiên, số thứ tự là 127, số cổng nguồn là 302 và số cổng đích là 80. Máy chủ B gửi xác nhận bất cứ khi nào nó nhận được phân đoạn từ Máy chủ A.**

1. ***Trong phân đoạn thứ hai được gửi từ Máy chủ A đến B, số thứ tự, số cổng nguồn và số cổng đích là gì?***

Số thứ tự (**sequence number**): 127 + 80 = **207** (Vì phân đoạn đầu tiên chứa *80 bytes* dữ liệu)

Số cổng nguồn (**source port number**): **302** (Số cổng nguồn không thay đổi)

Số cổng đích (**destination port number**): **80** (Số cổng đích không thay đổi)

1. ***Nếu phân đoạn đầu tiên đến trước phân đoạn thứ hai, trong xác nhận của phân đoạn đến đầu tiên, số xác nhận, số cổng nguồn và số cổng đích là gì?***

Số xác nhận (**acknowledgment number**): **207** (Vì Host B đã nhận được tất cả các byte cho đến byte 126, nên số xác nhận sẽ bằng số thứ tự của phân đoạn đầu tiên cộng với độ dài của phân đoạn đầu tiên)

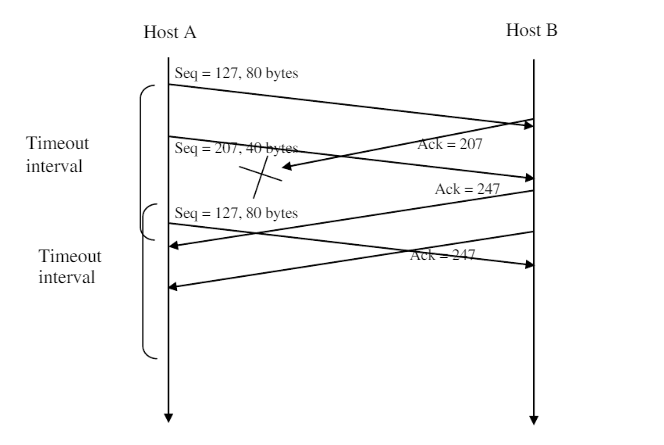
Số cổng nguồn (**source port number**): **80** (Số cổng nguồn của Host B)

Số cổng đích (**destination port number**): **302** (Số cổng đích của Host A)

1. ***Nếu phân đoạn thứ hai đến trước phân đoạn đầu tiên, trong phần xác nhận của phân đoạn đến đầu tiên, số xác nhận là gì?***

Nếu phân đoạn thứ hai đến trước phân đoạn đầu tiên, trong xác nhận của phân đoạn đến đầu tiên, số xác nhận (**ACK number**) là **127**, cho thấy rằng nó vẫn đang chờ các byte 127 trở đi.

1. ***Giả sử A gửi đến B theo thứ tự. Xác nhận đầu tiên bị mất và xác nhận thứ hai đến sau khoảng thời gian chờ đầu tiên. Vẽ sơ đồ thời gian, hiển thị các phân đoạn này cũng như tất cả các phân đoạn và báo nhận khác được gửi. (Giả sử không có thêm gói tin nào bị mất.) Đối với mỗi phân đoạn trong hình của bạn, hãy cung cấp số thứ tự và số byte dữ liệu; đối với mỗi xác nhận mà bạn thêm vào, hãy cung cấp số xác nhận.***



**P28:**

**Host A và B được kết nối trực tiếp thông qua một liên kết 100 Mbps. Có một kết nối TCP giữa hai máy chủ, và Host A đang gửi một tập tin lớn đến Host B qua kết nối này. Host A có thể gửi dữ liệu ứng dụng vào TCP socket của mình ở tốc độ lên đến 120 Mbps, nhưng Host B chỉ có thể đọc từ bộ đệm nhận TCP của mình ở tốc độ tối đa là 50 Mbps. Mô tả tác dụng của điều khiển luồng TCP.**

Vì dung lượng của liên kết chỉ là *100 Mbps*, nên tốc độ gửi của *A* có thể là tối đa *100 Mbps*. Tuy nhiên, *host A* gửi dữ liệu vào bộ đệm nhận nhanh hơn so với khả năng loại bỏ dữ liệu của *host B*. Bộ đệm nhận đầy với tốc độ khoảng *40 Mbps*. Khi bộ đệm đầy, *Host B* thông báo cho *host A* dừng việc gửi dữ liệu bằng cách đặt **RcvWindow** = *0*. *Host A* sau đó dừng việc gửi cho đến khi nhận được một đoạn **TCP** với **RcvWindow** > *0*. Do đó, *host A* sẽ lặp đi lặp lại việc dừng và bắt đầu gửi dữ liệu dựa trên các giá trị **RcvWindow** mà nó nhận từ *host B*. Trung bình, tốc độ dài hạn mà *host A gửi dữ liệu đến host B* trong kết nối này không vượt quá *60 Mbps*.

**P31:**

* **SampleRTT1 = 106ms, SampleRTT2 = 120 ms, SampleRTT3 = 140ms, SampleRTT4 = 90ms, SampleRTT5 = 115ms.**
* **Tính EstimatedRTT sau mỗi SampleRTT thu được, α = 0.125 và EstimatedRTT là 100 ms ngay trước khi mẫu thứ nhất thu được.**
* **Tính DevRTT sau mỗi mẫu được thu được, β = 0.25 và DevRTT = 5ms ngay trước khi mẫu thứ nhất thu được.**
* **Tính TCP TimeoutInterval sau mỗi mẫu thu được.**

\* Ta có các công thức:

*EstimatedRTT = αSampleRTT + (1 - α)EstimatedRTT*

*DevRTT = βSampleRTT - EstimatedRTT + (1 - β)DevRTT*

*TimeoutInterval = EstimatedRTT + 4 \* DevRTT*

***Với SampleRTT thứ nhất = 106ms:***

**EstimatedRTT** = 0.125 \* 106 + 0.875 \* 100 = **100.75ms**

**DevRTT** = 0.25 \* |106 - 100.75| + 0.75 \* 5 = **5.06ms**

**TimeoutInterval** = 100.75 + 4 \* 5.06 = **120.99ms**

***Với SampleRTT thứ hai = 120ms:***

**EstimatedRTT** = 0.125 \* 120 + 0.875 \* 100.75 = **103.15ms**

**DevRTT** = 0.25 \* |120 - 103.15| + 0.75 \* 5.06 = **8ms**

**TimeoutInterval** = 103.15 + 4\*8 = **135.15ms**

***Với SampleRTT thứ ba = 140ms:***

**EstimatedRTT** = 0.125 \* 140 + 0.875 \* 103.15 = **107.76ms**

**DevRTT** = 0.25 \* |140 - 107.76| + 0.75 \* 8 = **14.06ms**

**TimeoutInterval** = 107.76 + 4 \* 14.06 = **164ms**

***Với SampleRTT thứ tư = 90ms:***

**EstimatedRTT** = 0.125 \* 90 + 0.875 \* 107.76 = **105.54ms**

**DevRTT** = 0.25 \* |90 - 105.54| + 0.75 \* 14.06 = **14.42ms**

**TimeoutInterval** = 105.54 + 4 \* 14.12 = **163.22ms**

***SampleRTT thứ năm = 115ms:***

**Với EstimatedRTT** = 0.125 \* 115 + 0.875 \* 105.54 = **106.71ms**

**DevRTT** = 0.25 \* |115 - 106.71| + 0.75 \* 14.42 = **12.88ms**

**TimeoutInterval** = 106.71 + 4 \* 12.88 = **158.23ms**

**BÀI HỌC RÚT RA**

* **Biết được một số điểm khác biệt giữa giao thức GBN và SR.**
* **Rút ra được kích thước cửa sổ truyền có ảnh hưởng đến khả năng tắc nghẽn của giao thức.**
* **Biết cách tính các giá trị sequence number, source port number, destination number.**
* **Biết được cách thức điều khiển luồng của TCP hoạt động và tác dụng của nó.**
* **Nắm được cách tính toán các đại lượng như EstimatedRTT, DevRTT, TimeoutInterval**

**\* Nhóm không có câu hỏi thắc mắc.**

**NGUỒN THAM KHẢO:**

* **Slide bài giảng trên lớp.**
* **Sách Computer.Networking.A.Top-Down.Approach.6th.Edition.**
* [Bard (google.com)](https://bard.google.com/chat?hl=vi)