

## Computer Networking - A Top-Down Approach Featuring the Internet, 5<sup>th</sup> edition Solutions to Review Questions and Problems (Vietnamese version)

Author : K55CC translate team.(K55CC-University of engineering and technology-Vietnam national university)

This document belong to K55CC.All copies or sharing must be allowed by K55CC.

### Chapter1: Introduction-Review Question

**Câu 1: Sự khác nhau giữa 1 host và 1 end system. Liệt kê các kiểu end system. Web server có phải là 1 end system không?**

Không có sự khác nhau. “host” và “end system” là 2 từ có thể thay thế cho nhau.

End system gồm có: PC, máy trạm, web server, mail server, web TV, các PDA có kết nối internet...

**Câu 2: Ví dụ về giao thức ngoại giao (diplomatic protocol)**

Giả sử Alice, một đại sứ của quốc gia A muốn mời Bob, một đại sứ nước B, ăn tối. Alice không chỉ đơn giản là chỉ cần gọi Bob trên điện thoại và nói, "Đến bàn ăn của chúng tôi bây giờ". Thay vào đó, cô gọi Bob và cho thấy một ngày và thời gian. Bob có thể đáp ứng bằng cách nói rằng ông không phải có sẵn mà cụ thể ngày, nhưng anh có sẵn một ngày khác. Alice và Bob tiếp tục gửi "thông điệp" qua lại cho đến khi họ đồng ý vào một ngày và thời gian. Bob sau đó cho thấy tại Đại sứ quán vào ngày đã thoả thuận, hy vọng không quá 15 phút trước khi hoặc sau khi thời gian đã thoả thuận. ngoại giao giao thức cũng cho phép hoặc Alice hoặc Bob lịch sự hủy bỏ sự tham gia nếu họ có lý do hợp lý.

**Câu 3: Chương trình client là gì? Chương trình server là gì? Một chương trình server yêu cầu và nhận dịch vụ từ 1 chương trình client đúng k?**

Một chương trình mạng thường có 2 chương trình, mỗi cái chạy trên 1 host khác nhau, được nối vs nhau. Chương trình mà khởi đầu việc trao đổi thông tin là client. Thông thường, chương trình client yêu cầu và nhận dịch vụ từ chương trình server.

**Câu 4: Liệt kê 6 công nghệ truy cập.**

1. Dial-up modem over telephone line: residential;
2. DSL over telephone line: residential or small office;
3. Cable to HFC: residential;
4. 100 Mbps switched Ethernet: company;
5. Wireless LAN: mobile;
6. Cellular mobile access (for example, WAP): mobile

**Câu 7: Tốc độ truyền của LAN Ethernet là gì? Với 1 tốc độ truyền mỗi user trong mạng LAN có thể truyền liên tục với cùng tốc độ đc k?**

Ethernet LANs có tốc độ truyền là 10 Mbps, 100 Mbps, 1 Gbps and 10 Gbps. VD với X Mbps Ethernet ( $X = 10, 100, 1,000$  or  $10,000$ ), một user có thể truyền liên tục với tốc độ X Mbps nếu chỉ có user đó đang gửi dữ liệu. Nếu có nhiều user hoạt động thì mỗi user không thể truyền liên tục với tốc độ X Mbps.

**Câu 8: Môi trường vật lý mà Ethernet có thể truyền qua là gì?**

Ethernet có thể truyền qua cáp đồng trục mảnh và dây đồng xoắn đôi. Nó cũng có thể truyền qua sợi quang và cáp đồng trục dày.

**Câu 9: Cho biết khoảng tốc độ truyền của Dial up modems, ISDN, ADSL, HFC, FTTH. Chúng thuộc kiểu chia sẻ hay riêng biệt?**

Dial up modems: up to 56 Kbps, bandwidth is dedicated; ISDN: up to 128 kbps, bandwidth is dedicated; ADSL: downstream channel is .5-8 Mbps, upstream channel is up to 1 Mbps, bandwidth is dedicated; HFC, downstream channel is 10-30 Mbps and upstream channel is usually less than a few Mbps, bandwidth is shared. FTTH: 2-10Mbps upload; 10-20 Mbps download; bandwidth is not shared.

**Câu 10: Mô tả các công nghệ truy cập Internet không dây phổ biến hiện nay. So sánh và nêu điểm khác biệt của chúng**

Hiện nay có 2 công nghệ truy cập Internet không dây phổ biến.

- Mạng LAN không dây (Wireless LAN). Trong WLAN, người dùng không dây truyền và các nhận gói tin đến 1 trạm cơ sở (điểm truy cập không dây) trong bán kính khoảng vài chục mét. Trạm cơ sở thường được kết nối Internet bằng dây.
- Mạng truy cập không dây diện rộng: Trong các hệ thống này, các gói tin được truyền qua cùng cơ sở hạ tầng không dây được dùng cho hệ thống điện thoại, trạm cơ sở này được quản lý bởi 1 nhà cung cấp dịch vụ viễn thông. Nó cung cấp sự truy cập không dây cho người dùng trong vòng bán kính vài chục kilomet từ máy trạm cơ sở.

**Câu 11: Thuận lợi của mạng chuyển mạch điện so với chuyển mạch gói là gì?**

Mạng chuyển mạch điện có thể đảm bảo 1 lượng cố định băng thông giữa 2 điểm nối để duy trì 1 cuộc gọi. Hầu hết các mạng chuyển mạch gói hiện nay (bao gồm cả Internet) không tạo sự đảm bảo băng thông giữa 2 điểm nối.

**Câu 13: Giả sử có 1 gói tin được chuyển từ host gửi đến host nhận. Tốc độ truyền giữa host gửi và switch và giữa switch và host nhận lần lượt là  $R_1$  và  $R_2$ . Giả sử switch sử dụng chuyển mạch gói để lưu trữ và chuyển tiếp. Tính tổng thời gian delay giữa 2 điểm nối để gửi 1 packet có độ dài L (bỏ qua hàng đợi độ trễ khi truyền và khi xử lý)**

Tại thời điểm  $t_0$ , host gửi bắt đầu truyền. Tại thời điểm  $t_1 = L/R_1$ , host gửi hoàn thành việc truyền và toàn bộ gói tin được nhận tại router (không có trễ khi truyền). Vì router có toàn bộ gói tin tại thời điểm  $t_1$  nên nó có thể truyền gói tin đến host nhận tại thời điểm  $t_1$ . Tại thời điểm  $t_2 = t_1 + L/R_2$ , router hoàn thành việc truyền và toàn bộ gói tin được nhận tại host nhận. Vì vậy, tổng thời gian trễ là  $L/R_1 + L/R_2$ .

**Câu 15: Giả sử nhiều user chia sẻ 1 link 2Mbps. Và cũng giả sử rằng mỗi user truyền liên tục vs tốc độ 1Mbps khi truyền, nhưng mỗi user chỉ truyền 20% thời gian.**

- Nếu dùng chuyển mạch điện thì có bao nhiêu user được sử dụng?
- Giả sử dùng chuyển mạch gói. Tại sao nếu có ít hơn hoặc bằng 2 user cùng truyền thì không phải đợi? và nếu nhiều hơn 2 user cùng truyền thì phải có hàng đợi?
- Tính xác suất để mỗi user được truyền.
- Giả sử có 3 user. Tính xác suất tại bất kì thời điểm nào, cả 3 user cùng truyền 1 lúc. Tính khoảng thời gian để hình thành hàng đợi.

TL:

- Có 2 user được truyền vì mỗi user cần 1 nửa băng thông đường truyền.
- Theo giả thiết, mỗi user cần 1Mbps khi truyền, nếu có 2 hoặc ít hơn 2 user cùng truyền thì tối đa cần 2Mbps băng thông. Mà đường truyền đã cho có băng thông là 2Mbps nên không có hàng đợi nào cả.  
Ngược lại, nếu 3 user cùng truyền thì băng thông cần sẽ là 3Mbps, nhiều hơn băng thông hiện có. Trong trường hợp này sẽ có hàng đợi trước khi vào đường truyền.
- Xác suất = 0.2
- Xác suất để 3 user cùng truyền cùng lúc =  $C(3,3) \cdot p^3 \cdot (1-p)^{3-3} = (0.2)^3 = 0.008$ . Theo gt, hàng đợi tăng khi tất cả user đang truyền, nên khoảng tgian mà hàng đợi tăng (= xác suất để tất cả 3 user cùng truyền) là 0.008.

**Câu 16:** Giả sử đang gửi 1 gói tin từ 1 host nguồn đến 1 host đích qua 1 đường cố định. Liệt kê các khoảng trễ. Cái nào là cố định, cái nào thay đổi?

Các thành phần trễ gồm: trễ khi xử lý, trễ khi lan tỏa (propagation delay), trễ khi truyền (transmission delay) và trễ khi đợi. Trừ trễ khi đợi là thay đổi, tất cả những khoảng trễ còn lại đều là cố định.

**Câu 18:** Tính khoảng thời gian để truyền 1 gói tin độ dài L qua 1 đường truyền khoảng cách d, tốc độ truyền là s, tốc độ chuyển giao là R bps? Nó có phụ thuộc vào L và R không?

Khoảng thời gian =  $d/s$ .

Nó không phụ thuộc vào độ dài gói tin (L) và tốc độ chuyển R.

**Câu 19:** Giả sử host A muốn gửi 1 file lớn đến host B. Có 3 đường đi từ A đến B, tốc độ lần lượt là  $R_1 = 500$  kbps,  $R_2 = 2$  Mbps và  $R_3 = 1$  Mbps.

- Giả sử không có tắc nghẽn trong mạng. Tính thông lượng truyền file?
- Giả sử file có dung lượng 4 triệu byte. Chia kích thước file bằng thông lượng. Tính thời gian xấp xỉ để truyền file đến host B?
- Làm câu a, b với  $R_2$  giảm còn 100 kbps.

TL:

- 500 kbps. (chọn nhỏ nhất để đảm bảo cho  $R_1$  đáp ứng được)
- $4.000.000 \text{ byte} = 4.000 \text{ KB} = 32.000 \text{ Kbit}$ . Tgian =  $32000/500 = 64 \text{ s}$

c. thông lượng = 100 kbps. Tgian = 320 s.

**Câu 22: Liệt kê 5 nhiệm vụ mà 1 tầng có thể thực hiện được. Có thể xảy ra trường hợp một hay nhiều nhiệm vụ được thực hiện bởi nhiều tầng được k?**

5 nhiệm vụ chung là: kiểm soát lỗi, điều khiển luồng, phân đoạn và lắp ráp lại, dòn kênh và cài đặt kết nối. Có thể xảy ra trường hợp 1 hay nhiều nhiệm vụ được thực hiện bởi nhiều tầng. VD: kiểm soát lỗi thường được cung cấp ở nhiều tầng.

**Câu 24: Một message ở tầng ứng dụng là gì? Một segment ở tầng giao vận? Một datagram ở tầng network? Một frame ở tầng liên kết?**

- Message: dữ liệu mà 1 ứng dụng muốn gửi và truyền xuống tầng giao vận.
- Segment: được tạo ra bởi tầng giao vận bằng cách đóng gói message của tầng ứng dụng với header của tầng giao vận.
- Datagram: đóng gói segment của tầng giao vận với header của tầng mạng.
- Frame: đóng gói datagram của tầng mạng với header của tầng liên kết.

**Câu 25: Một router xử lý ở tầng nào? Một switch xử lý ở tầng nào? Và một host xử lý ở tầng nào?**

Các router xử lý ở tầng 1, 2, 3 tức là tầng vật lý, tầng liên kết và tầng network. Các switch xử lý ở tầng 1 và 2 tức là tầng vật lý và tầng liên kết. Các host xử lý ở cả 5 tầng.

## Chapter1: : Introduction- Problem

**P5: Giả sử có 2 host A và B kết nối với nhau bởi 1 đường có tốc độ R bps. 2 host cách nhau m mét và tốc độ lan tỏa (propagation speed) là s (m/s). Host A gửi 1 packet kích thước L đến host B.**

- a. Biểu diễn độ trễ lan tỏa ( $d_{\text{prop}}$ ) thông qua m và s.
- b. Xác định thời gian truyền của gói tin  $d_{\text{trans}}$  thông qua L và R.
- c. Bỏ qua độ trễ xử lý và hàng đợi. Viết biểu thức tính độ trễ giữa 2 điểm kết thúc.
- d. Giả sử Host A bắt đầu truyền gói tin tại thời điểm  $t = 0$ . Tại thời điểm  $t = d_{\text{trans}}$ , bit cuối cùng của gói tin ở đâu?
- e. Giả sử  $d_{\text{prop}} > d_{\text{trans}}$ . Tại thời điểm  $t = d_{\text{trans}}$ , bit đầu tiên của gói tin ở đâu?
- f. Giả sử  $d_{\text{prop}} < d_{\text{trans}}$ . Tại thời điểm  $t = d_{\text{trans}}$ , bit đầu tiên của gói tin ở đâu?
- g. Giả sử  $s = 2,5 \cdot 10^8$ ,  $L = 120$  bits và  $R = 56$  kbps. Tính khoảng cách m để  $d_{\text{prop}} = d_{\text{trans}}$

TL:

- a.  $d_{\text{prop}} = m/s$  (s) – là độ trễ lan tỏa (thời gian gói tin đi hết khoảng cách m)
- b.  $d_{\text{trans}} = L/R$  (s) – là độ trễ truyền (thời gian từ lúc bit đầu tiên của gói tin ra khỏi host đến khi bit cuối cùng của gói tin ra khỏi host).
- c.  $d_{A-B} = d_{\text{prop}} + d_{\text{trans}} = m/s + L/R$  (s)
- d. Tại  $t = d_{\text{trans}}$ , bit cuối cùng của gói tin vừa rời khỏi host A.
- e. Tại  $t = d_{\text{trans}}$ , bit đầu tiên của gói tin vẫn đang ở trên đường truyền, chưa đến được host B.
- f. Tại  $t = d_{\text{trans}}$ , bit đầu tiên của gói tin đã đến host B.

$$g. \quad d_{\text{prop}} = d_{\text{trans}} \Leftrightarrow m/s = L/R \rightarrow m = Ls/R = 523158 \text{ (m)}$$

**P6:** Xét việc gửi giọng nói thời gian thực từ host A đến host B qua 1 mạng chuyển mạch gói (VoIP). Host A chuyển giọng nói thành luồng bit tín hiệu số 64 kbps. Sau đó host A nhóm các bit thành từng gói tin 56 bytes. Chỉ có 1 đường duy nhất từ A đến B tốc độ truyền là 2 Mbps và độ trễ lan tỏa là 10 ms. Ngay khi host A tạo được 1 gói tin nó gửi gói tin đó cho B. Ngay khi host B nhận được gói toàn bộ gói tin nó chuyển các bit của gói tin thành tín hiệu tương tự. Tính thời gian kể từ khi bit đầu tiên được tạo (từ tín hiệu gốc ở host A) đến khi bit cuối cùng được giải mã (ở host B)?

Xét bit đầu tiên trong 1 gói tin. Trước khi bit này được truyền đi, tất cả các bit trong gói tin phải được tạo

$$\text{ra. Vì vậy cần: } \frac{56.8}{64.10^3} \text{ (s)} = 7 \text{ (ms).}$$

$$\text{Thời gian cần để truyền gói tin là: } \frac{56.8}{2.10^6} \text{ (s)} = 224 \text{ } \mu\text{s}$$

Độ trễ lan tỏa là 10 ms.

$$\text{Độ trễ cho đến khi giải mã xong là: } 7 \text{ ms} + 224 \text{ } \mu\text{s} + 10 \text{ ms} = 17,224 \text{ ms}$$

**P12:** Giả sử N gói tin cùng đến 1 đường truyền và hiện tại không có gói tin nào được truyền đi hoặc phải xếp hàng. Mỗi gói tin có độ dài L và tốc độ đường truyền là R. Tính độ trễ hàng đợi trung bình của N gói tin.

Độ trễ hàng đợi là 0 khi gói tin đầu tiên đã truyền đi, là L/R khi gói tin thứ 2 đã truyền đi và tổng quát, là (n – 1) L/R khi gói tin thứ n đã truyền đi.

Vì vậy, độ trễ trung bình của N gói tin là:

$$\begin{aligned} & (L/R + 2L/R + \dots + (N-1).L/R) / N \\ &= L/(RN) * (1 + 2 + \dots + (N-1)) \\ &= L/(RN) * N(N-1)/2 \\ &= LN(N-1)/(2RN) \\ &= (N-1)L/(2R) \end{aligned}$$

**P24:** Giả sử có 2 host A và B cách nhau 20.000 km, được nối trực tiếp vs nhau bằng 1 đường tốc độ R = 2 Mbps. Tốc độ lan tỏa trên đường truyền là  $2,5.10^8$  m/s.

- Tính tích của băng thông và độ trễ ( $R.d_{\text{prop}}$ ).
- Xét việc gửi 1 file độ dài 800.000 bits từ A đến B. Giả sử file được truyền liên tục. Số bit tối đa được truyền đi trên đường truyền tại bất kì thời điểm nào là bao nhiêu?
- Giải thích ý nghĩa tích  $R.d_{\text{prop}}$ .
- Độ rộng của 1 bit (tính theo mét) trên đường truyền bằng bao nhiêu?
- Viết công thức tổng quát tính độ rộng của 1 bit trên đường truyền. Cho biết tốc độ lan tỏa là s, tốc độ truyền là R và độ dài đường truyền là m.

TL:

- a.  $R.d_{prop} = R \cdot m/s = 2 \cdot 20 \cdot 10^6 / (2,5 \cdot 10^8) = 160.000 \text{ bits.}$
- b. Số bit tối đa được truyền đi trên đường truyền tại bất kì thời điểm  $= R.d_{prop} = 160.000 \text{ bits.}$
- c. Tích số giữa băng thông( $R$ ) và độ trễ( $d_{prop}$ ) của đường truyền là số bit tối đa có thể ở trên đường truyền
- d. Độ rộng của 1 bit  $= \text{độ dài đường truyền} / (R.d_{prop}) = 20 \cdot 10^6 / 160.000 = 125 \text{ m}$
- e. Độ rộng của 1 bit  $= m / (R.d_{prop}) = s/R.$

## Chapter2: Application Layer -Review Question

**Câu 1: Liệt kê 5 ứng dụng internet không độc quyền và giao thức mà chúng sử dụng tại tầng ứng dụng**

- The Web - HTTP;
- File transfer (bittorrent) - FTP;
- Remote login - Telnet;
- Network News - NNTP;
- E-mail - SMTP;

**Câu 2: Sự khác biệt giữa Network architecture và Application architecture?**

- *Network architecture* là một hệ thống các tiến trình giao tiếp với nhau thông qua các tầng
- *Application architecture* được hiểu theo nghĩa khác, nó được thiết kế bởi nhà phát triển ứng dụng và nó điều khiển hoạt động của các application.

**Câu 4: Cho 1 ứng dụng chia sẻ file P2P. Bạn có đồng ý với nhận xét “không có định nghĩa bên client and server trong 1 giao tiếp giữa chúng? Tại sao?**

Không, vì trong mọi giao dịch đều có 1 bên là client và 1 bên là server. Trong ứng dụng chia sẻ file P2P thì bên nhận được file là client còn bên gửi file là server.

**Câu 6: Giả sử bạn muốn làm một giao dịch giữa 1 remote client và 1 server với tốc độ cao, bạn nên chọn UDP hay TCP?**

Nên chọn UDP, vì sử dụng UDP không cần thiết lập đường chuyển nên nhanh hơn bạn chỉ mất 1 RTT đó là client gửi yêu cầu tới UDP socket. Còn nếu sử dụng TCP bạn mất tối thiểu 2 RTT: 1 cho việc thiết lập kết nối TCP, còn lại cho client gửi yêu cầu và server trả phản hồi.

**Câu 15: Vì sao nói FTP gửi thông tin điều khiển là out-of-band?**

FTP sử dụng 2 kết nối TCP song song. Một kết nối để kiểm soát thông tin (chẳng hạn như 1 yêu cầu chuyển giao 1 tập tin) và một kết nối khác dùng để chuyển giao các tập tin. Bởi vì các thông tin kiểm soát không được gửi qua cùng 1 kết nối với tập tin nên có thể nói rằng FTP gửi thông tin điều khiển "out-of-band".

**Câu 18 : Nêu sự khác biệt giữa download-and-delete mode and the download-and-keep mode in POP3?**



- Với chế độ **download-and-delete** sau khi người dùng lấy tin nhắn từ một máy chủ POP, các tin nhắn sẽ bị xóa. Điều này đặt ra một vấn đề cần giải quyết đó là khi nhiều người cùng truy cập các tin nhắn từ nhiều máy khác nhau (văn phòng máy tính, máy tính gia đình, vv.)
- Trong chế độ **download-and-keep**, tin nhắn sẽ không bị xóa sau khi người sử dụng lấy các tin nhắn. Điều này cũng có thể là bất tiện là mỗi lần người sử dụng lấy các tin nhắn, tất cả các tin nhắn không bị xóa sẽ được chuyển vào máy tính (bao gồm các thông báo rất cũ).

**Câu 19 : Có hay không một tổ chức Mail server and Web server nào có cùng 1 bí danh(alias) cho 1 hostname không? Cái gì sẽ lưu trữ host name of the mail server?**

Có. The MX record được sử dụng để ánh xạ tên của mail server với địa chỉ IP.

**Câu 22: Overlay network (mạng bao phủ) là gì? Nó có chứa router không? The edges(rìa) của nó là gì? Nó được tạo ra và duy trì như thế nào?**

- Overlay network là một hệ thống gồm các nút tham gia vào chia sẻ tập tin và các liên kết giữa chúng.
- Nó không chứa router.
- The edges của nó là logical link (liên kết logic).
- Cách tạo ra nó là: khi 1 node mới muốn tham gia vào hệ thống nó cần biết địa chỉ IP của 1 hay nhiều node của hệ thống, sau đó nó sẽ gửi thông điệp cho các node này, các nút này nhận và xác nhận, và nó sẽ trở thành 1 phần của hệ thống.

**Câu 25: Skype công nghệ P2P cho 2 giao thức quan trọng nào?**

User location và NAT traversal.

**Câu 26: 4 ứng dụng quan trọng phù hợp với kiến trúc P2P?**

- a) File Distribution
- b) Instant Messaging
- c) Video Streaming
- d) Distributed Computing

## Chapter2: Application Layer -Problem

**Câu 1: Đúng / Sai**

- a. Một người dùng yêu cầu từ 1 trang web 1 text và 3 hình ảnh, client sẽ gửi 1 tin nhắn yêu cầu và nhận 4 tin nhắn phản hồi? - **Sai: gửi 4 nhận 4**
- b. 2 trang web khác biệt có thể gửi qua cùng 1 kết nối kiên trì. - **Đúng**
- c. Với một kết nối không kiên trì giữa trình duyệt và máy chủ có thể cho 1 gói tin TCP thực hiện 2 request HTTP khác nhau. - **Sai**
- d. Thông điệp yêu cầu của giao thức HTTP không thể rỗng.

**Câu 3:** Giả sử một HTTP client muốn lấy 1 tài liệu của trang web mà đã biết URL, địa chỉ IP chưa biết. Thì tầng ứng dụng và tầng giao vận cần giao thức nào

- Application layer protocols: DNS and HTTP
- Transport layer protocols: UDP for DNS; TCP for HTTP

**Câu 7:** Giả sử trong trình duyệt web bạn click vào một link chứa 1 trang web. Bạn cần lấy 1 địa chỉ IP của trang web đó chưa có trong cache. Bạn phải đi qua n DNS sau đó mới có được nó. Mỗi lần thăm 1 DNS bạn mất  $RTT_1, RTT_2, \dots, RTT_n$ . Trong trang web đó có 1 đối tượng text. Bạn mất  $RTT_0$  để đi từ host tới server chứa đối tượng đó. Tính thời gian từ khi click đến khi nhận được đối tượng.

Tổng thời gian để lấy được địa chỉ IP là:  $RTT_1 + \dots + RTT_n$ ;

Sau khi đã biết địa chỉ IP bạn mất  $RTT_0$  để kết nối TCP và  $RTT_0$  nữa để gửi thông điệp và nhận đối tượng.

Nên tổng thời gian cần thiết là:  $2.RTT_0 + RTT_1 + \dots + RTT_n$ ;

**Câu 8:** Với đề bài câu 7, giả sử HTML có thêm 8 đối tượng nữa trên cùng server. Mất bao lâu thời gian với:

a. **Không kiên trì và không có kết nối song song:**

Khi trang web thêm 8 đối tượng ta cần 8 lần thiết lập và 8 lần gửi thông điệp và nhận đối tượng nên mất thời gian là:  $8.2.RTT_0 = 16 RTT_0$

Nên tổng thời gian là:  $2.RTT_0 + RTT_1 + \dots + RTT_n + 16 RTT_0$

b. **Không kiên trì có 5 kết nối song song:**

1 lần kết nối gửi nhận được 5 đối tượng nên 8 đối tượng cần 2 lần kết nối nên mất thời gian là:  $4RTT_0$

Nên tổng thời gian là:  $2.RTT_0 + RTT_1 + \dots + RTT_n + 4 RTT_0$

c. **Kết nối kiên trì**

Do đã khởi tạo kết nối để lấy đối tượng text nên 8 đối tượng này không cần khởi tạo kết nối nữa.

Sử dụng kết nối kiên trì nên cần 1  $RTT_0$  để gửi yêu cầu và nhận đối tượng

Nên tổng thời gian là:  $2.RTT_0 + RTT_1 + \dots + RTT_n + RTT_0 = 3 RTT_0 + RTT_1 + \dots + RTT_n$

**Câu 14:** SMTP kết thúc thân mail như thế nào? HTTP thì sao? HTTP có thể sử dụng phương thức giống SMTP để kết thúc mail được không?

- SMTP kết thúc thư bằng một dòng chỉ chứa dấu chấm.
- HTTP quản lý thư bằng trường độ dài trong header.
- HTTP không thể sử dụng phương thức giống SMTP được. vì HTTP message có thể để ở dạng mã nhị phân còn SMTP thì phải để ở dạng ASCII

**Câu 17:** Giả sử truy cập mail của bạn bằng POP3



- a. Giả sử bạn định dạng cho thư của bạn to làm việc ở chế độ down xong xóa. Hoàn thành giao dịch bên dưới:
- C: dele 1  
C: retr 2  
S: (blah blah ...  
S: .....blah)  
S: .  
C: dele 2  
C: quit  
S: +OK POP3 server signing of
- b. Chế độ down xong giữ
- C: retr 2  
S: blah blah ...  
S: .....blah  
S: .  
C: quit  
S: +OK POP3 server signing off
- c. Ban đầu ở chế độ down xong giữ đọc thư 1 xong tắt đi sau 5 phút đọc thư 2. Đưa ra bản ghi cho trường hợp này?
- C: list  
S: 1 498  
S: 2 912  
S: .  
C: retr 1  
S: blah .....  
S: ....blah  
S: .  
C: retr 2  
S: blah blah ...  
S: .....blah  
S: .  
C: quit  
S: +OK POP3 server signing off

**Câu 20: Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers (outside your department) that are most popular among the users in your department?**

Chúng ta có thể định kì lưu nhanh các bản của DNS caches trong những máy chủ DNS địa phương đó. Web server xuất hiện thường xuyên nhất trong DNS caches chính là server phổ biến nhất. Điều này là bởi nếu nhiều người dùng quan tâm đến 1 Web server thì DNS requests đến Web server đó được gửi thường xuyên hơn. Do đó, Web server đó sẽ dc xuất hiện trong DNS caches nhiều hơn local DNS cache, vì vậy thời gian truy vấn sẽ là 0 msec. Các trường hợp khác, thời gian truy vấn lớn hơn.

**Câu 22:** Consider distributing a file of  $F = 15$  Gbits to  $N$  peers. The server has an upload rate of  $u_s = 30$  Mbps, and each peer has a download rate of  $d_i = 2$  Mbps and an upload rate of  $u_i$ . For  $N = 10, 100$ , and  $1,000$  and  $u_i = 300$  Kbps,  $700$  Kbps, and  $2$  Mbps, prepare a chart giving the minimum distribution time for each of the combinations of  $N$  and  $u_i$  for both client-server distribution and P2P distribution.

Để tính toán thời gian điều phối tối thiểu (minimum distribution time) cho điều phối client-server, ta sử dụng công thức:

$$D_{cs} = \max \{ NF/u_s, F/d_{\min} \}$$

Tương tự, để tính toán thời gian điều phối tối thiểu cho điều phối P2P, ta sử dụng công thức:

$$D_{P2P} = \max \{ F/u_s, F/d_{\min}, NF/(u_s + \sum_{i=1}^N u_i) \}$$

Ở đây,  $F = 15$  Gbits  $= 15 * 1024$  Mbits

$u_s = 30$  Mbps

$d_{\min} = d_i = 2$  Mbps

Note,  $300\text{Kbps} = 300/1024$  Mbps

#### Client Server

		N		
		10	100	1000
u	300 Kbps	7680	51200	512000
	700 Kbps	7680	51200	512000
	2 Mbps	7680	51200	512000

#### Peer to Peer

		N		
		10	100	1000
u	300 Kbps	7680	25904	47559
	700 Kbps	7680	15616	21525
	2 Mbps	7680	7680	7680

**Câu 23.** Consider distributing a file of  $F$  bits to  $N$  peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each peer at different rates, as long as the combined rate does not exceed  $u_s$

- Suppose that  $u_s/N \leq d_{\min}$ . Specify a distribution scheme that has a distribution time of  $NF/u_s$ .
- Suppose that  $u_s/N \sim d_{\min}$ . Specify a distribution scheme that has a distribution time of  $F/d_{\min}$ .
- Conclude that the minimum distribution time is in general given by  $\max\{NF/u_s, F/d_{\min}\}$ .

Xét một chương trình điều phối trong đó server gửi file tới từng client, song song, v một tốc độ  $u_s/N$ . Lưu ý tốc độ này nhỏ hơn tốc độ download của từng client, vì theo giả thiết  $u_s/N \leq d_{\min}$ . Do đó, mỗi client cũng có thể nhận được file với một tốc độ là  $u_s/N$ . Từ lúc mỗi client nhận tốc độ  $u_s/N$ , thời gian để mỗi client

nhận file hoàn chỉnh là  $F/(us/N) = NF/us$ . Tất cả các client đều nhận hoàn chỉnh file trong thời gian  $NF/us$ , nên tổng thời gian điều phối cũng là  $NF/us$ .

#### Câu 34

- Giả sử bạn mở FTPclient trước khi mở FTPserver. Có vấn đề gì xảy ra không?  
Khi đó client sẽ cố gắng kết nối với server trong khi server chưa mở=> kết nối lỗi.
- Bạn chạy UDP client trước khi chạy UDP server?  
Không có vấn đề gì. Vì client không cần khởi tạo kết nối tới server.
- Sử dụng port khác nhau giữa client và server.  
Khi đó client sẽ cố gắng kết nối TCP với 1 tiến trình không đúng. Nên sẽ có lỗi

### Chapter3: Transport Layer -Review Question

**Câu 1. Xem xét một kết nối TCP từ host A đến B. Giả sử TCP segment đi từ A -> B có port nguồn là x và port đích là y. Vậy port nguồn và port đích của segment đi từ B -> A là j?**

Source Port: y, Dest Port: x

#### Question 2

**Giải thích tại sao nhà phát triển ứng dụng lại chọn UDP hơn TCP.**

Solution :

Nhà phát triển ứng dụng chọn UDP vì họ không muốn ứng dụng của họ phải sử dụng cơ chế kiểm soát tắc nghẽn (cơ chế điều tiết tốc độ truyền dữ liệu của ứng dụng lúc bị tắc nghẽn). Cơ chế này có thể làm giảm tốc độ truyền, điều này có thể ảnh hưởng tới ứng dụng, nhất là những ứng dụng chat voice hay hội thảo trực tuyến). Những ứng dụng đó không cần độ tin cậy của dữ liệu được truyền, mà quan trọng là thời gian.

**Question 3. Liệu có ứng dụng nào có cả khả năng truyền dữ liệu đáng tin cậy ngay cả khi nó chạy trên giao thức UDP?**

Solution :

Có. Nhà phát triển ứng dụng có thể thêm giao thức truyền dữ liệu đáng tin cậy vào giao thức của lớp ứng dụng. Tất nhiên, nó sẽ đòi hỏi một lượng công việc đáng kể cho việc phát hiện lỗi.

#### Question 4 True or False

- Giả sử host A truyền một tập tin lớn tới host B trên giao thức TCP, và host B không có dữ liệu j để gửi tới host A. Host B sẽ không gửi ACK tới host A vì nó không thể đóng gói ACK vào dữ liệu dc. (☹ unlogical )
- Kích thước của cửa sổ bên nhận (TCP Rcv window) không bao giờ thay đổi trong suốt quá trình kết nối.
- Giả sử host A truyền một tập tin lớn tới host B trên giao thức TCP, số byte dữ liệu NAK không thể vượt quá bộ nhớ đệm của phía nhận.

- d. Giả sử host A truyền một tập tin lớn tới host B trên giao thức TCP, Số thứ tự của segment cho lần gửi này là m, thì số thứ tự cho segment tiếp theo phải là m+1. ( Sai vì nếu quá trình gửi gói trc mà bị lỗi thì phải gửi lại gói m).
- e. TCP segment có 1 trường trong tiêu đề cho cửa sổ bên nhận.
- f. Giả sử thời gian RTT gần nhất trong kết nối TCP là 1 s. Vậy phải cài đặt thời gian timeout  $\geq 1$  cho lần truyền này. (Sai vì cần sd nhiều RTT trong quá khứ chứ k sd RTT gần nhất).
- g. Giả sử host A gửi 1 segment tới B vs STT là 38 và 4-bytes dữ liệu. Vậy ACK cho segment này phải này 42. (sai. ACK là 38)

Solution : a. **F**   b. **F**   c. **T**   d. **F**   e. **T**   f. **F**   g. **F**

### Question 5

Giả sử A gửi 2 gói TCP liên nhau tới B. Segment thứ nhất có STT là 90, segment thứ 2 có STT là 110.

- a. Segment đầu tiên có dung lượng là bao nhiêu.
- b. Giả sử segment đầu tiên bị mất, nhưng segment thứ 2 vẫn đến đc B. Vậy stt của ACK đc gửi từ B về A là j?

Solution :

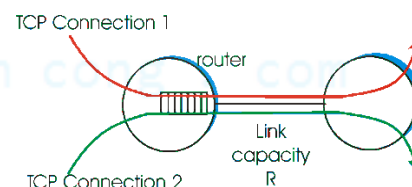
- a.  $110 - 90 = 20$  bytes.
- b. Vì gói 1 mất nên ACK vẫn là 90

**Question 7:** Hiện tại có 2 kết nối TCP với 1 nút cổ chai có tốc độ là R bps. Cả 2 kết nối đều có 1 file lớn cần gửi (đến cùng 1 hướng qua nút cổ chai đó). Việc truyền dữ liệu bắt đầu ở cùng một thời điểm. Vậy tốc độ truyền mà TCP sẽ phân cho mỗi kết nối là bao nhiêu?

**Chú ý :** Qua tính công bằng của giao thức TCP ta biết đc, nếu N kết nối cùng nhau chia sẻ một kênh truyền tắc nghẽn thì mỗi kết nối sẽ nhận đc  $1/N$  băng thông. (Tức là nhận đc băng thông bằng nhau).

Solution :

Qua chú ý thì đáp án là :  $R/2$



### Question 8

Xem xét cơ chế kiểm soát tắc nghẽn trong TCP. Nếu xảy ra timeout ở bên gửi thì threshold sẽ được đặt bằng một nửa của giá trị threshold trc đó.

⇒ Sai vì nếu xảy ra mất gói tin, ngưỡng threshold sẽ được đặt bằng một nửa của Congwin (Congestion window)

Note : Tốc độ truyền bị giới hạn bởi cửa sổ tắc nghẽn Congwin. Threshold là ngưỡng giữa 2 pha

Sự kiện	Trạng thái	Hành động bên gửi TCP	Giải thích
Nhận ACK cho dữ liệu không được ack trước đó	Slow Start (SS)	$CongWin = CongWin + MSS$ , If ( $CongWin > Threshold$ ) Đặt trạng thái thành "Congestion Avoidance"	Kết quả bởi gấp đôi CongWin mỗi RTT
Nhận ACK cho dữ liệu không được ack trước đó	Congestion Avoidance (CA)	$CongWin = CongWin + MSS * (MSS / CongWin)$	Tăng theo cấp số cộng, tăng CongWin lên 1 MSS mỗi RTT
Sự kiện mất gói phát hiện bởi 3 ACK lặp	SS hoặc CA	$Threshold = CongWin / 2$ , $CongWin = Threshold$ , Đặt trạng thái thành "Congestion Avoidance"	Nhanh chóng phục hồi, thực hiện tăng cấp số nhân. CongWin sẽ không giảm dưới 1 MSS.
Timeout	SS hoặc CA	$Threshold = CongWin / 2$ , $CongWin = 1 MSS$ , Đặt trạng thái thành "Slow Start"	Vào slow start
ACK lặp	SS hoặc CA	Tăng bộ đếm ACK lặp cho segment được ack	CongWin và Threshold không thay đổi

**Question 9 (P1 trong sách)** Giả sử Network Layer cung cấp dịch vụ sau. Network layer ở host nguồn chấp nhận gửi 1 segment có max size là 1200 bytes và địa chỉ đích từ transport layer. Network layer đảm bảo gửi thành công segment tới transport layer tại host đích.

Giả sử có nhiều tiến trình cùng chạy tại host đích.

a. Thiết kế một giao thức đơn giản cho transport layer để nhận được dữ liệu phù hợp cho mỗi tiến trình tại host đích. Giả sử HĐH ở host đích có thể cấp 4 bytes port number cho mỗi tiến trình đang chạy.

b. Điều chỉnh giao thức đó để nó có thể trả về địa chỉ ("return address") cho host đích.

c. Trong giao thức của bạn liệu transport layer có phải làm j trong lõi của mạng máy tính?

Solution :

- Gọi giao thức này là STP (Simple transfer protocol). Bên phía gửi, STP chấp nhận cho tiến trình gửi 1 chunk kích thước k vượt quá 1196 bytes dữ liệu, một địa chỉ host đích và một cổng nhận. STP thêm 4 bytes vào trg header của mỗi chunk, 4 bytes đó để lưu port number của tiến trình bên nhận. Và network layer sẽ chuyển gói tin tới transport layer của bên nhận. Sau đó STP sẽ giải nén segment và kiểm tra segment nhận đc có port number ứng với tiến trình nào rồi gửi segment đó tới đúng tiến trình cần nhận.
- Bây h mỗi segment sẽ có 2 trg tiêu đề (header). Một trg cho port nguồn và 1 trg cho port đích. Bên phía gửi chấp nhận gửi 1 chunk kích thước k vượt quá 1192 bytes, một địa chỉ host đích, một port nguồn và một port đích.  
STP tạo một segment bao gồm dữ liệu của application, port nguồn và port đích. Sau đó nó sẽ chuyển segment và địa chỉ host đích cho tầng network để gửi sang bên nhận... Sau khi bên đích nhận đc segment, STP sẽ chuyển tới ứng dụng dữ liệu cần nhận và cả port nguồn.
- Nó k phải làm j. Nó chỉ nằm trên thiết bị đầu cuối.

**Question 10. (P2)** Giả sử có 1 hành tinh có dịch vụ gửi thư. Mỗi gia đình có 1 địa chỉ (1 hòm thư riêng). Mỗi thành viên trong gia đình đều có một tên riêng. Dịch vụ gửi thư có thể gửi từ nhà này sang nhà khác. Nó yêu cầu phải cho bức thư và địa chỉ của nhà cần gửi vào trong phong bì. Mỗi gia đình đều có 1 ng đại diện, ng này sẽ nhận thư và phân phát thư cho các thành viên khác trong gia đình.

**a.Sử dụng giao thức ở Question 9 để gửi thư.Mô tả cách ng đại diện nhận và phân phát thư.**

**b.Trong giao thức này,liệu mail server có phải mở phong bì để kiểm tra bức thư bên trong hay không?**

Solution :

- a) Ng gửi sẽ đưa cho ng đại diện của gia đình họ bức thư cùng với tên ng nhận và địa chỉ của ng đó.Sau đó ng đại diện sẽ viết tên ng nhận vào đầu bức thư,nhét nó vào phong bì.Bên ngoài phong bì ghi địa chỉ của ngôi nhà phía ng nhận.Sau đó đưa cho bưu điện(chính là dịch vụ gửi thư của hành tinh đó) Bên kia,ng đại diện nhận đc phong bì sẽ lấy bức thư ra,xem tên ng nhận,và gửi cho đúng ng đó.  
(Giả sử đó là thư tình mà ng đại diện là bố or mẹ thì e k bik “What will happen?” lol)
- b) Mail server k cần mở phong bì,nó chỉ kiểm tra địa chỉ trên phong bì .

**Question 11(P5) Tại sao bây h các ứng dụng chat voice hay video lại dùng TCP hơn UDP,trong khi TCP có cơ chế kiểm soát tắc nghẽn gây trễ đường truyền?**

Sol : vì bây giờ firewall hay chặn các luồng truyền tải data của UDP.Trong khi TCP có thể vượt qua đc.

**Question 12.(P7) Giả sử một tiến trình ở host C có một UDP socket với port là 6789.Giả sử cả 2 host A và B đều có thể gửi segment tới C với cùng một port đích là 6789.Liệu cả 2 segment đều cùng hướng đến cùng một socket đích ở host C hay không?Nếu có thì làm sao để host C có thể biết đc sự khác nhau giữa 2 segment đó(nó có nguồn gốc từ đâu)?**

Solution: Có.Mỗi segment nhận đc đều đc thì hệ điều hành sẽ cung cấp 1 tiến trình với địa chỉ IP để xác định nguồn gốc của segment đó.

**Question 13.(P8) Giả sử Web server chạy trên host C tại port 80 và dùng kết nối kiên trì.Nó nhận đc 2 yêu cầu từ host A và B.Liệu cả hai yêu cầu đc gửi trên cùng một socket của C.Nếu 2 yêu cầu đc trả lời qua 2 socket khác nhau thì liệu cả 2 socket đều có port là 80?**

Solution : Trong kết nối kiên trì, Web server tạo ra từng kết nối socket riêng biệt . Mỗi socket kết nối được xác định với một bộ bốn : (địa chỉ IP nguồn, port nguồn , địa chỉ IP đích ,port đích ) .Khi host C nhận được IP datagram/segment , nó kiểm tra 4 trng này để xác định socket mà nó phải truyền segment qua . Vì vậy , các yêu cầu từ A và B đc truyền thông qua các socket khác nhau .Cả 2 đều có port number là 80 , tuy nhiên , việc nhận diện các socket này dựa vào địa chỉ IP nguồn.Không giống như UDP, khi lớp transport gửi một segment TCP lên tiến trình ở tầng ứng dụng , nó không chỉ ra địa chỉ IP nguồn ,mà nó ngầm quy định bởi các trình định danh socket.

**Question 14(P9) Trong giao thức rdt,tại sao chúng ta cần sử dụng STT(sequence number)?**

Solution : STT đc sử dụng cho bên nhận,giúp phân biệt được khi nào dữ liệu nhận đc là bị trùng.

**Question 15(P10) Trong rdt protocol,tại sao phải sử dụng bộ đếm thời gian(timer)?**



Solution : Nếu như thời gian bên gửi chờ ACK (or NAK) vượt quá thời gian quy định của bộ đếm, thì bên gửi kết luận gói tin có thể bị mất. Do đó, nó sẽ truyền lại gói tin đó.

**Question 16 (P11)** Giả sử roundtrip delay (độ trễ phản hồi) giữa bên nhận và bên gửi là hằng số, và bên gửi bị độ trễ này. Vậy ta có cần timer cho rdt 3.0 nữa hay không? Giả sử gói tin có thể bị mất.

Solution : Timer vẫn cần thiết cho rdt 3.0 vì nếu bên gửi bị độ trễ RTT (RTT là thời gian từ lúc gửi đến lúc nhận được ACK) thì có 1 ưu điểm duy nhất là bên gửi bị độ trễ packet hoặc ACK (NAK) có thể bị mất. Hãy đặt vào tình huống cụ thể, nếu ACK không mất mà chỉ đến trễ, lúc đó bên gửi vẫn cần timer để biết được gói tin đã mất hay chưa.

### Chapter3: Transport Layer - Problem

- Giả sử tại cùng 1 thời điểm, client A và client B đều muốn khởi tạo 1 phiên Telnet tới server S. Cung cấp số hiệu cổng nguồn và đích có thể có cho:
  - Các segments gửi từ A đến S
  - Các segments gửi từ B đến S
  - Các segments gửi từ S đến A
  - Các segments gửi từ S đến B
  - Nếu A, B là 2 host khác nhau, thì số hiệu cổng nguồn trong segments từ A tới S có được giống như là từ B tới S không?
  - Nếu chúng là cùng 1 host thì sao?

**Answer:**

	source port numbers	destination port numbers
a) A→S	467	23
b) B→S	513	23
c) S→A	23	467
d) S→B	23	513
e) Yes		
f) No		

- UDP và TCP sử dụng bù 1 cho trường checksum.

Giả sử ban đầu bạn có 3 byte 8 bit: 01010011, 01010100, 01110100. Bù 1 của tổng của các byte 8 bit là gì? Tại sao UDP lại lấy bù 1 của tổng? Tại sao không chỉ sử dụng tổng? Với việc sử dụng bù 1, làm thế nào để người nhận phát hiện lỗi? Có thể nào lỗi 1 bit sẽ đi không bị phát hiện? Lỗi 2 bit thì sao?

**Answer:**

$$\begin{array}{r}
 01010011 \\
 + 01010100 \\
 \hline
 10100111 \\
 10100111 \\
 + 01110100 \\
 \hline
 00011100
 \end{array}$$

⇒ Bù 1 là: 11100011

- Để phát hiện lỗi, người nhận sẽ tính tổng của 4 byte (3 byte gốc + 1 byte checksum)
- Nếu tổng có chứa 1 số 0  $\Rightarrow$  có lỗi
- Tất cả các lỗi 1 bit đều bị phát hiện, nhưng 2 bit lỗi thì có thể sẽ không phát hiện được (ví dụ chữ số cuối cùng của byte đầu tiên chuyển thành 0 còn chữ số cuối cùng của byte thứ 2 chuyển thành 1)

3. Giả sử người nhận UDP tính checksum cho gói tin UDP segment và nhận thấy rằng nó trùng khớp giá trị truyền tải trong trường checksum của gói tin. Người nhận có thể hoàn toàn chắc chắn rằng lỗi bit không xảy ra hay không? Giải thích.

**Answer:**

**Không**, người nhận không thể hoàn toàn chắc chắn lỗi bit không đã xảy ra

Bởi: nếu các bit tương ứng của 2 byte trong gói tin là 0 và 1 thì khi chúng đổi giá trị cho nhau (0 $\rightarrow$ 1, 1 $\rightarrow$ 0) thì tổng của chúng vẫn giống nhau  $\rightarrow$  bù 1 của tổng cũng sẽ giống nhau

4. So sánh phía gửi và phía nhận của giao thức rdt2.2 và rdt3.0

**Answer:**

Phía gửi dùng giao thức rdt3.0 khác phía gửi dùng giao thức rdt2.2, trong đó có thêm thời gian *timeouts*. Chúng ta đã thấy rằng việc thêm thời gian *timeouts* làm tăng khả năng trùng lặp các gói tin. Tuy nhiên, phía nhận dùng giao thức rdt2.2 đã có thể xử lý các gói dữ liệu trùng lặp. (Gói tin trùng lặp ở phía nhận trong rdt 2.2 sẽ phát sinh nếu ACK bị lỗi, và người gửi sau đó truyền lại dữ liệu cũ). Do đó phía nhận trong giao thức rdt2.2 làm việc như là phía nhận trong giao thức rdt3.0

5. Hãy xem xét một giao thức truyền dữ liệu đáng tin cậy mà chỉ sử dụng NAK.

Giả sử phía gửi sẽ gửi dữ liệu không thường xuyên. Giao thức chỉ dùng NAK có thích hợp hơn giao thức sử dụng ACKs hay không? Tại sao?

Bây giờ giả sử phía gửi đã có rất nhiều dữ liệu để gửi và kết nối end-to-end có rất ít mất mát. Trong trường hợp này, giao thức chỉ dùng NAK có thích hợp hơn giao thức sử dụng ACKs không? Tại sao?

**Answer:**

Trong giao thức chỉ dùng NAK, mất gói x chỉ có thể phát hiện được bởi phía nhận khi phía nhận nhận được gói x-1 và ngay sau đó là x+1.

Nếu phía gửi gửi không thường xuyên, và có 1 khoảng thời gian delay dài giữa 2 gói tin x và x+1 thì chỉ đến khi nào gói x+1 được gửi đi và đến được phía nhận thì phía nhận mới phát hiện được là gói x bị mất  $\Rightarrow$  sẽ mất 1 khoảng thời gian dài để khôi phục gói x theo giao thức chỉ dùng NAK

Mặt khác, nếu dữ liệu được gửi thường xuyên thì việc khôi phục theo giao thức chỉ dùng NAK sẽ xảy ra rất nhanh chóng. Hơn nữa, nếu lỗi không thường xuyên xảy ra thì NAK chỉ thỉnh thoảng mới được gửi (khi cần thiết) còn ACK thì không cần gửi. Do đó làm giảm đáng kể trong thông tin phản hồi

⇒ Giao thức chỉ dùng NAK tối ưu hơn giao thức sử dụng ACK

6. Xem xét giao thức GBN với kích thước windows phía gửi là 3 và phạm vi của seq num là 1024.

Giả sử tại thời điểm  $t$ , thứ tự tiếp theo trong gói tin mà phía nhận đang mong đợi là 1 sequence number của

k. Cho rằng, môi trường không sắp xếp lại thứ tự của các msg. Trả lời các câu hỏi

- Các tập seq num có thể có bên trong windows phía gửi tại thời điểm  $t$  là gì ?
- Tất cả các giá trị có thể có của trường ACK trong tất cả các msg có thể hiện đang truyền quay trở lại phía người gửi tại thời điểm  $t$  là gì ?

**Answer:**

- Ở đây chúng ta có 1 windows với kích thước  $N=3$ .

Giả sử phía nhận đã nhận được gói tin  $k-1$ , và đã gửi ACK cho nó và tất cả các gói tin khác. Nếu tất cả ACK đều được nhận ở phía gửi thì sau đó windows phía người gửi sẽ là  $[k, k+N-1]$ .

Giả sử thứ 2 là không có ACK nào được nhận tại phía gửi, khi đó windows phía gửi sẽ chứa  $N$  gói tin và phải bao gồm gói  $k-1$ , nên windows phải gửi sẽ là  $[k-N, k-1]$

Với những lập luận trên thì windows phía gửi sẽ có kích thước là 3 và sẽ bắt đầu tại 1 nơi nào đó trong phạm vi  $[k-N, k]$

- Nếu phía nhận đang chờ gói tin thứ  $k$ , và sau đó nó nhận được gói tin thứ  $k-1$  và  $N-1$  gói tin trước đó => .....P19

7. Trả lời đúng hoặc sai cho các câu hỏi sau đây và giải thích cho câu trả lời của bạn:

- Với giao thức *selective repeat*, nó có thể cho phía gửi nhận được một ACK cho một gói tin nằm bên ngoài windows hiện tại của nó.
- Với GBN, nó có thể cho phía gửi nhận được một ACK cho một gói tin nằm bên ngoài windows hiện tại của nó.

**Answer:**

- Đúng.

Giả sử phía gửi có kích thước windows là 3 và gửi các gói tin 1,2,3 tại  $t_0$ .

Tại  $t_1$  ( $t_1 > t_0$ ) phía nhận gửi ACK 1,2,3.

Tại  $t_2$  ( $t_2 > t_1$ ), phía gửi *timeouts* và gửi lại gói tin 1,2,3.

Tại  $t_3$ , phía nhận nhận các gói tin trùng lặp, và gửi lại ACK 1,2,3.

Tại  $t_4$  phía gửi nhận ACK 1,2,3 mà phía nhận gửi tại thời điểm  $t_1$ , và chuyển windows lên 4,5,6.

Tại  $t_5$  phía gửi nhận ACK 1,2,3 mà phía nhận gửi tại  $t_3$ . Khi này ACK ngoài khoảng của windows

- Đúng. <giống ý a>

8. Xem xét việc chuyển một tập tin rất lớn của L byte từ Host A đến Host B. Giả sử 1 MSS = 536 byte.
- Giá trị tối đa của L để TCP seq num không bị cạn kiệt là gì? Trường TCP seq num có 4 byte.
  - Với giá trị L có được trong câu a), phải mất bao lâu để truyền tải các tập tin.
- Giả sử rằng có tổng số 66 byte trường tiêu đề của tầng giao vận, mạng, liên kết được thêm vào segment trước khi gói dữ liệu được gửi qua một liên kết 155 Mbps. Bỏ qua kiểm soát lưu lượng và kiểm soát tắc nghẽn vì thế có thể gửi ra các segment liên tục.

**Answer:**

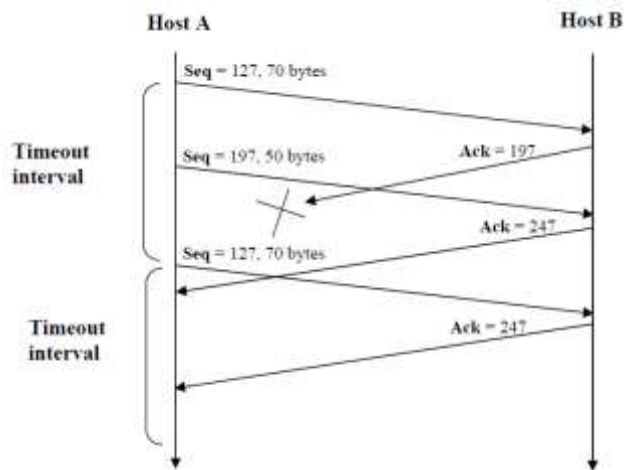
Có tất cả  $2^{32} = 4294967296$  sequence number

- Sequence number không tăng lên 1 mỗi segment. Thay vào đó, nó gia tăng số lượng các byte dữ liệu được gửi. Vì vậy kích cỡ của MSS là không liên quan – kích thước tập tin tối đa có thể được gửi từ A đến B chỉ đơn giản là số lượng byte biểu diễn bởi  $2^{32} = 4.19$  Gbytes
  - Số lượng segment là  $\frac{2^{32}}{536} = 8012999$  (lấy nguyên, làm tròn lên)  
66byte tiêu đề được thêm vào cho mỗi segment  
 $\Rightarrow$  tổng có  $\frac{2^{32}}{536} * 66 = 66 * 8012999 = 528,857,934$  bytes của tiêu đề  
 $\Rightarrow$  tổng độ lớn phải truyền đi là  $s = 2^{32} + \frac{2^{32}}{536} * 66 = 4,824 * 10^9$  bytes  
 $\Rightarrow$  thời gian để truyền tải là  $(s * 8) / (155 * 1000) = 249$  (s)
9. Host A và B giao tiếp qua kết nối TCP và host B đã nhận được từ A tất cả các byte thông qua byte 126. Giả sử rằng host A sau đó gửi thêm 2 segment tới host B **back-to-back**. Segment đầu tiên và thứ 2 chứa tương ứng 70 và 50 byte dữ liệu. Trong segment đầu tiên sequence number là 127, source port là 302, destination port là 80. Host B gửi ACK ngay khi nó nhận được segment từ A
- Trong segment thứ 2 được gửi từ host A đến host B thì sequence number, source port, destination port là bao nhiêu?
  - Nếu segment đầu tiên đến trước segment thứ 2, trong acknowledgement của segment đầu tiên đến thì acknowledgement number, source port, destination port là bao nhiêu?
  - Nếu segment thứ 2 đến trước segment đầu tiên, trong acknowledgement của segment đầu tiên đến thì acknowledgement number là bao nhiêu?
  - Giả sử 2 segment được gửi bởi A đến theo thứ tự tại B, acknowledgement đầu tiên là bị mất, acknowledgement thứ 2 đến sau khoảng thời gian time out. Vẽ sơ đồ thời gian hiển thị các segment và acknowledgements đã gửi.

**Answer:**

- Trong segment thứ 2:
  - sequence number là 197
  - source port number là 302
  - destination port number là 80

- b. Nếu segment đầu tiên đến trước segment thứ 2:
  - acknowledgement number là 197
  - source port number là 80
  - destination port number là 302.
- c. Nếu segment thứ 2 đến trước segment đầu tiên:
  - acknowledgement number là 127 – chỉ ra rằng nó vẫn còn đợi các bytes 127 trở đi
- d.



10. Host A và B được kết nối trực tiếp với liên kết 100 Mbps. Có một kết nối TCP giữa hai host, và host A gửi đến host B một tập tin có kích thước lớn qua kết nối này. Host A có thể gửi dữ liệu ứng dụng của nó vào socket TCP của nó với tốc độ cao là 120 Mbps nhưng Host B có thể đọc nhận được bộ đệm với tốc độ tối đa là 60 Mbps. Mô tả ảnh hưởng của kiểm soát lưu lượng TCP.

Answer:

Host A có thể gửi với tốc độ 100Mbps. Tuy nhiên host A gửi vào bộ đệm nhanh hơn so với host B có thể đọc. Bộ đệm đầy lên với tốc độ 40Mbps. Khi bộ đệm đầy, host B báo hiệu tới host A dừng gửi dữ liệu bằng cách thiết lập **RcvWindow = 0**. Host A sẽ dừng việc gửi dữ liệu cho đến tận khi nhận được segment với **RcvWindow > 0**. Do đó host A sẽ liên tục dừng lại và bắt đầu gửi. Trung bình trong thời gian dài, host A gửi dữ liệu đến host B như là 1 phần của kết nối này và không lớn hơn 60Mbps

11. Hãy xem xét các thủ tục TCP để ước lượng RTT. Giả sử  $\alpha = 0,1$ .

SampleRTT1 là SampleRTT gần đây nhất. SampleRTT2 là RTT gần đây nhất tiếp theo, và như vậy.

- a. Đối với một kết nối TCP nhất định, giả sử có bốn acknowledgments đã được trả lại với SampleRTTs tương ứng: SampleRTT4, SampleRTT3, SampleRTT2, và SampleRTT1, Thể hiện EstimatedRTT với bốn SampleRTTs.
- b. Khái quát công thức cho n SampleRTTs
- c. Đối với công thức trong phần (b) áp dụng cho  $n \rightarrow \infty$ .

Answer:

- a. Ký hiệu  $\text{EstimatedRTT}^{(n)}$  là ước lượng sau n sample

$$\text{EstimatedRTT}^{(1)} = \text{SampleRTT}_1$$

$$\text{EstimatedRTT}^{(2)} = \alpha \cdot \text{SampleRTT}_1 + (1-\alpha) \cdot \text{SampleRTT}_2$$

$$\begin{aligned} \text{EstimatedRTT}^{(3)} &= \alpha \cdot \text{SampleRTT}_1 + (1-\alpha) [\alpha \cdot \text{SampleRTT}_2 + (1-\alpha) \cdot \text{SampleRTT}_3] \\ &= \alpha \cdot \text{SampleRTT}_1 + (1-\alpha) \cdot \alpha \cdot \text{SampleRTT}_2 + (1-\alpha)^2 \cdot \text{SampleRTT}_3 \end{aligned}$$

$$\text{EstimatedRTT}^{(4)} = \alpha \cdot \text{SampleRTT}_1 + (1-\alpha) \cdot \alpha \cdot \text{SampleRTT}_2 + (1-\alpha)^2 \cdot \alpha \cdot \text{SampleRTT}_3 + (1-\alpha)^3 \cdot \text{SampleRTT}_3$$

- b.  $\text{EstimatedRTT}^{(n)} = \alpha \cdot \sum_{j=1}^{n-1} (1-\alpha)^j \cdot \text{SampleRTT}_j + (1-\alpha)^n \text{SampleRTT}_n$

c.

$$\begin{aligned} \text{EstimatedRTT}^{(\infty)} &= \frac{\alpha}{1-\alpha} \sum_{j=1}^{\infty} (1-\alpha)^j \text{SampleRTT}_j \\ &= \frac{1}{9} \sum_{j=1}^{\infty} 9^j \text{SampleRTT}_j \end{aligned}$$

12. Host A gửi một tập tin rất lớn đến host B qua một kết nối TCP. Qua kết nối này là không bao giờ có bất kỳ mất mát gói dữ liệu và không có timeout. Ký hiệu tốc độ truyền dẫn của liên kết kết nối host A đến Internet là R bps. Giả sử rằng quá trình host A có khả năng gửi dữ liệu vào TCP socket với tốc độ S bps, trong đó  $S = 10 \cdot R$ . Hơn nữa giả sử rằng bộ đệm nhận là đủ lớn để chứa toàn bộ tập tin, và bộ đệm gửi chỉ có thể giữ một phần trăm của tập tin. Điều gì sẽ ngăn chặn quá trình trong máy chủ liên tục truyền dữ liệu đến socket TCP của nó với tốc độ S bps ?

**Answer:**

Trong vấn đề này, không có nguy hiểm trong tràn bộ nhớ phía nhận nhận từ khi vùng đệm nhận phía nhận có thể giữ toàn bộ tập tin. Hơn nữa, vì không có mất mát và acknowledgements được gửi trả trước khi timeout, TCP kiểm soát nghẽn không điều tiết phía gửi. Tuy nhiên, quá trình trên host A sẽ không liên tục chuyển dữ liệu cho socket vì vùng đệm gửi sẽ nhanh chóng được lấp đầy. Một khi vùng đệm gửi trở thành đầy, quá trình sẽ chuyển dữ liệu ở tỉ lệ trung bình hoặc  $R \ll S$ .

13. Xem xét việc gửi một tập tin lớn từ một host khác qua kết nối TCP mà không có mất mát.
- Giả sử TCP sử dụng AIMD kiểm soát tắc nghẽn mà không có khởi đầu chậm chạp. Giả sử tăng cwnd lên 1 MSS mỗi khi tất cả ACKs nhận được và giả định RTT là hằng số, mất bao lâu để cwnd tăng từ 5 MSS lên 11 MSS (giả sử không có sự kiện mất)?
  - Tính thông lượng trung bình cho kết nối này trong khoảng thời gian = 6 RTT?

**Answer:**

- a. Phải mất:

1 RTT để tăng cwnd lên 6 MSS;

2 RTT để tăng cwnd lên 7 MSS;

3 RTT để tăng cwnd lên 8 MSS;

4 RTT để tăng cwnd lên 9 MSS;

5 RTT để tăng cwnd lên 10 MSS;

6 RTT để tăng cwnd lên 11 MSS



b. Trong RTT đầu tiên: 5 MSS đã được gửi;

Trong RTT thứ 2: 6 MSS đã được gửi;

Trong RTT thứ 3: 7 MSS đã được gửi;

Trong RTT thứ 4: 8 MSS đã được gửi;

Trong RTT thứ 5: 9 MSS đã được gửi;

Trong RTT thứ 6: 10 MSS đã được gửi.

Vì thế trong khoảng 6 RTT,  $5+6+7+8+9+10 = 45$  MSS đã được gửi (và acknowledged).

Do đó, chúng ta có thể nói rằng thông lượng trung bình trong khoảng 6RTT là:

$$\frac{45 \text{ MSS}}{6 \text{ RTT}} = 7,5 \text{ MSS/RTT}$$

14. Trong giao thức rdt3.0, gói tin ACK từ phía nhận đến phía gửi không phải đánh số, mặc dù chúng có trường trong ACK mà chứa stt của packet mà chúng đã biết. Tại sao các packet ACK không đòi hỏi đánh số (sequence number)

**Answer**

Chúng ta thấy rằng, phía gửi cần seq num để phía nhận có thể biết packet đó có trùng với 1 packet đã nhận từ trước không. Đối với ACKs, thì bên gửi không cần thông tin đó (VD: stt của ACK) để biết 1 ACK trùng lặp. Một ACK trùng lặp là rõ ràng đối với phía nhận của rdt3.0, kể từ khi nó nhận được ACK gốc thì nó chuyển sang trạng thái mới. ACK trùng lặp không phải là cái ACK người gửi cần nên nó bị bỏ qua bởi rdt3.0

15. Suppose an application uses rdt3.0 as its transport layer protocol. As the stop-and-wait protocol has very low channel utilization (shown in the cross-country example), the designers of this application let the receiver keep sending back a number (more than two) of alternating ACK 0 and ACK 1 even if the corresponding data have not arrived at the receiver. Would this application design increase the channel utilization? Why? Are there any potential problems with this approach? Explain

**Answer**

Yes. This actually causes the sender to send a number of pipelined data into the channel.

Yes. Here is one potential problem. If data segments are lost in the channel, then the sender of rdt 3.0 won't re-send those segments, unless there are some additional mechanism in the application to recover from loss

## Chapter4: Network Layer – Review Question

**R1: Đưa ra một vài thuật ngữ được sử dụng trong cuốn sách. Tên gọi packet của tầng transport là segment và tên gọi của tầng link là frame. Tên packet của tầng network? Routers và link – layer switches được gọi là packet switches. Nguyên tắc khác nhau cơ bản của routers và link – layer switches?**

Trả lời:

Packet tầng network được gọi là datagram. Một router muốn chuyển packet phải dựa trên địa chỉ IP của packet (tầng 3). A link-layer switch muốn chuyển packet phải dựa trên địa chỉ MAC (tầng 2).

**R2: Hai chức năng quan trọng của tầng network trong chuyển mạch gói là gì? Ba chức năng quan trọng của tầng network trong một chuyển mạch ảo?**

Trả lời:

Hai chức năng quan trọng của tầng network trong datagram – network: chuyển và định tuyến.

Ba chức năng quan trọng của tầng network trong virtual – circuit: chuyển, định tuyến, và thiết lập tuyến đường.

**R3: Sự khác nhau giữa routing và forwarding?**

Trả lời:

Forwarding là chuyển một packet từ một liên kết đầu vào của một router, ra một liên kết đầu ra của router đó sao cho thích hợp.

Routing là xác định các router trên đường truyền từ nguồn tới đích.

**R4: Routers trong chuyển mạch gói và chuyển mạch điện có được sử dụng trong bảng định tuyến? Nếu vậy, hãy mô tả bảng định tuyến cho hai mô hình trên?**

Trả lời:

Được sử dụng cho cả hai.

**R7: Tại sao mỗi cổng vào trong vùng nhớ của router có tốc độ cao (Discuss why each input port in a high-speed router stores a shadow copy of the forwarding table)**

With the shadow copy, the forwarding decision is made locally, at each input port, without invoking the centralized routing processor. Such decentralized forwarding avoids creating a forwarding processing bottleneck at a single point within the router

**R9: Miêu tả việc mất mát gói tin có thể xảy ra tại cổng vào. Miêu tả việc mất mát gói tin tại cổng vào do có thể bị loại bỏ. (trừ trường hợp sử dụng bộ nhớ đệm)**

Trả lời:

Mất mát gói tin xảy ra nếu kích cỡ hàng đợi tại cổng vào tăng lên cao bởi vì tốc độ forwarding chậm do đó bộ nhớ router sẽ bị đầy. Điều đó có thể bị loại trừ nếu tốc độ forwarding tối thiểu là  $n$  lần tốc độ của cổng vào,  $n$  chính là số cổng vào.

**R10: Miêu tả việc mất mát gói tin có thể xảy ra tại cổng ra.**

Trả lời:

Việc mất mát gói tin có thể xảy ra nếu kích cỡ hàng đợi tại cổng ra (số gói tin cần chuyển đi ra khỏi buffer của router) tăng cao trong khi tốc độ ra của chậm

**R11: HOL blocking là gì? Nó có thể xảy ra trong cổng vào hoặc cổng ra không?**

Trả lời:

HOL blocking – các datagram xếp hàng tại trước hàng đợi ngăn không cho datagram khác chuyển tiếp. Nó có thể xảy ra tại cổng vào.

**R12: Router có địa chỉ IP không? Nếu có, độ dài ?**

Trả lời:

Có. Nó có 1 địa chỉ IP cho mỗi interface.

**R13: Chuyển sang hệ có số nhị phân dải địa chỉ 223.1.3.27**

Trả lời:

11011111 00000001 00000011 00011100

**R15: Giả sử có 3 router giữa host nguồn và đích. Bỏ qua phân mảnh, một IP datagram gửi từ host nguồn đến đích sẽ đi qua bao nhiêu interfaces? Có bao nhiêu bảng chuyển tiếp để chuyển gói từ nguồn đến đích?**

Trả lời:

8 interfaces; 3 forwarding tables

**R16: Giả sử một ứng dụng tạo ra 40bytes dữ liệu trong khoảng thời gian 20msec và mỗi dữ liệu được đóng gói trong gói TCP segment và sau đó thành gói IP datagram. Tỷ lệ của mỗi datagram sẽ overhead là gì? Tỷ lệ dữ liệu ứng dụng là bao nhiêu?**

Trả lời:

Overhead: 20 byte của TCP header, 20 byte của IP header = 40 byte

⇒ 50% overhead

**R17: Giả sử host A gửi host B một gói TCP segment được đóng gói trong IP datagram. Khi host B nhận được gói datagram. Tầng network trong host B sẽ làm như thế nào để biết nó nên truyền bằng TCP tốt hơn UDP hay một cái gì khác?**

Trả lời:

8 bit trong trường giao thức của IP datagram bao gồm thông tin về giao thức tầng transport mà máy đích nên gửi segment lên lớp transport

**R20: Có nhận định rằng: “Khi IPv6 có đường ống qua IPv4, IPv6 nằm trong đường ống IPv4 như một giao thức tại tầng liên kết”. Bạn có đồng ý với phát biểu trên. Tại sao và tại sao không?**

Trả lời:

Có. Bởi vì toàn bộ gói tin IPv6 được đóng gói trong gói tin IPv4.

**R21. So sánh sự khác nhau giữa hai thuật toán linkstate và distance vector.**

Thuật toán linkstate: tính toán con đường đi có chi phí ngắn nhất từ nguồn tới đích bằng cách sử dụng những kiến thức hoàn chỉnh, tổng quát về mạng.

Định tuyến theo distance vector: : Việc tính toán chi phí đường đi ít nhất được thực hiện lặp đi lặp lại, phân phối theo cách thức. Một nút chỉ biết hàng xóm mà nó phải chuyển tiếp một gói tin đi qua để đạt được đích đến theo con đường có chi phí là ít nhất, và chi phí của con đường đó từ nó đến đích.

**R22, Thảo luận làm thế nào mà internet có thể tổ chức sắp xếp để có thể phân chia cho hàng triệu người dung?**

Router được tổng hợp vào các hệ thống tự điều khiển (các AS). Trong một hệ thống AS, tất cả các router chạy cùng giao thức định tuyến trong nội bộ AS. Đặc biệt gateway router trong các AS khác nhau chạy các giao thức định tuyến liên hệ thống tự điều khiển xác định đường dẫn định tuyến giữa các AS.

Vấn đề của scale(phân chia) được giải quyết khi một router trong AS chỉ cần biết về các router và gateway router trong AS của nó

**R23, Có cần thiết việc mỗi hệ thống từ trị (AS) sử dụng cùng thuật toán định tuyến intra- AS không?**

Không. Mỗi AS có quyền tự chủ trong quản lý định tuyến trong AS.

**R26, Điền vào ô trống sau: RIP advertisements typically announce the number of hops to various destinations. BGP updates, on the other hand, announce the \_\_\_\_\_ to the various destinations.**

“sequence of ASs on the routes” - Trình tự các Ass trên các tuyến đường.

**R29, Định nghĩa và so sánh các thuật ngữ sau: Subnet,prefix và BGP route.**

Trả lời

Subnet là một phần của mạng lưới lớn hơn, một subnet không chứa 1 router. Ranh giới của nó được xác định bởi giao diện của router và host.

Prefix là một phần của địa chỉ CDIRIzed được viết dưới dạng abcd/x. Một prefix có thể gồm 1 hay nhiều subnet.

Khi một router quảng bá một prefix qua một phiên BGP, nó bao gồm một số thuộc tính BGP.

Trong thuật ngữ BGP , một prefix cùng với các thuộc tính của nó là một tuyến đường BGP (hoặc chỉ đơn giản là mộttuyến đường).

## Chapter 4: Network Layer – Problem

**P1:** Trong câu hỏi này, chúng ta xét 1 vài ưu nhược điểm của virtual-circuit và datagram network.

- Giả sử routers là đối tượng chịu điều kiện khá thường xuyên bị lỗi. điều này có gây tranh cãi trong việc nó có lợi cho kiến trúc Vc hay kiến trúc datagram? Tại sao
- Giả sử rằng có 1 node nguồn và 1 node đích, yêu cầu cố định công suất luôn có sẵn tại các router trên đường dẫn từ nguồn đến đích, cho việc sử dụng độc quyền của các traffic giữa nguồn và đích. Sử dụng VC hay kiến trúc datagram? Tại sao?
- Giả sử các links và router trong mạng lưới không bị lỗi và định tuyến giữa tất cả các router nguồn/ đích vẫn không thay đổi. trong tình huống này, VC hay kiến trúc datagram có nhiều điều khiến luồng giao thông overhead hơn? Tại sao

Trả lời

- Với 1 mạng lưới kết nối mạng, mỗi router bị lỗi sẽ liên quan đến phần định tuyến của kết nối đó. ở mức tối thiểu, nó sẽ yêu cầu router đó “lùi ngược dòng” từ router thất bại đến xây dựng 1 đường dẫn mới đến node đích, với tất cả các dấu hiệu xây dựng liên quan đến set up đường dẫn mới. Hơn nữa, tất cả các router trong đường dẫn ban đầu từ node sai phải được đi xuống kết nối sai, với tất cả dấu hiệu xây dựng liên quan đến nó.

Với 1 mạng ko kết nối datagram, không dấu hiệu thiết lập set up cho cả đường dẫn mới hay đi xuống đường dẫn cũ, chúng ta có thể thấy, tuy nhiên, bảng định tuyến đó sẽ cần update lại( ví dụ dùng thuật toán distance vector hoặc link state) để đưa các router không thành công vào tài khoản. chúng ta có thể thấy vs distance vector, bảng định tuyến này có thể thay đổi thì thoả bằng đại phương hóa từng vùng gần router thất bại. do đó, 1 datagram network sẽ thích hợp hơn. Thật thú vị, các tiêu chuẩn thiết kế trong arpanet ban đầu có thể sử dụng dưới điều kiện căng thẳng là 1 trong những lý do kiến trúc datagram được lựa chọn cho tổ tiên Internet

- để cho 1 router có thể duy trì số lượng công suất cố định trên đường dẫn giữa node nguồn và node đích, nó cần biết các đặc tính của các truy cập từ tất cả các phiên truyền thông qua các liên kết trong router. Điều này có thể có trong mạng lưới liên kết định hướng, nhưng ko có trong mạng lưới ko kết nối. vì vậy 1 kết nối mạng định hướng VC sẽ thích hợp hơn.
- trong tình huống này, kiến trúc datagram có kiểm soát luồng trên không hơn. Điều này là do các tiêu đề gói tin khác nhau cần thiết để định tuyến các gói tin thông qua mạng. Nhưng kiến trúc VC, 11 khi tất cả các circuit đã được thiết lập. nó sẽ không bao h thay đổi. như vậy overhead là không đáng kể về lâu dài

**P2:** Xét 1 virtual circuit network, giả sử số VC là 1 trường 8 bits.

- Số tối đa VC có thể mang trong 1 link?
- Giả sử node giữa xác định đường dẫn và VC number ở thiết lập kết nối. giả sử có 1 cái VC number giống vậy được sử dụng trong mỗi link cùng VC's path. Miêu tả làm thế nào node trung tâm có thể xác định VC number ở thiết lập liên kết. có thể có ít VC trong tiến trình hơn số tối đa đã xác định?
- Giả sử sự khác nhau được cho phép trong mỗi link dọc theo đường dẫn của VC. Trong quá trình thiết lập kết nối, sau khi 1 đường dẫn end – to – end được xác định. Miêu tả làm thế nào links có thể chọn VC number của nó và cấu hình nó theo cách phân cấp, mà không có sự phụ thuộc vào 1 node trung tâm?

Trả lời

- Số tối đa VC có thể mang trong 1 link =  $2^8 = 256$

- b. các node trung tâm có thể chọn bất kỳ số lượng VC đó free từ  $\{0, 1, \dots, 2^8 - 1\}$ . Theo cách này, nó không thể có ít VC hơn 256 trong tiến trình mà không có bất kỳ VC phổ biến free
- c. mỗi cấp của các liên kết độc lập có thể phân bố VC number từ tập  $\{0, 1, \dots, 2^8 - 1\}$  do đó 1 số VC có khả năng sẽ có 1 số VC khác nhau cho mỗi liên kết dọc theo con đường của VC phải thay thế VVC number của mỗi gói tin đến VV number liên kết vs các lk ra bên ngoài.

**P3: Bảng chuyển tiếp bare-bones trong VC network có 4 cột. Ý nghĩa của từng giá trị trong mỗi cột? Bảng chuyển tiếp bare-bones trong datagram network có 2 cột, ý nghĩa?**

Trả lời:

Đối vs bảng chuyển tiếp VC, các cột là: Incoming Interface, Incoming VC Number, Outgoing Interface, Outgoing VC Number

Đối vs bảng chuyển tiếp datagram, các cột là: Destination Address, Outgoing Interface

**P5: Xem như 1 VC network có 2 bit cho trường VC number. Giả sử network muốn thiết lập 1 đường truyền chuyển ảo qua 4 links A B C D. Giả sử mỗi link đó có cần thực hiện 2 đường chuyển ảo khác nhau. Với số VC như hình vẽ:**

Link A	Link B	Link C	Link D
00	01	10	11
01	10	11	00

Hãy nhớ rằng 1 VC chỉ truyền trên 1 trong 4 link.

- a. Nếu 1 VC muốn dùng 1 number VC giống nhau cho cả 4 link. Thì VC number có thể gán cho new VC.
- b. Nếu mỗi VC được phép có nhiều VC number trong các link khác nhau thì có bao nhiêu tổ hợp cho 4 VC number có thể sử dụng?

Trả lời:

- a. Không có VC number nào được gán cho new VC. Nên cũng không có VC nào được thiết lập.
- b.  $2^4$  ( TRR)

**P9: xem như 1 datagram sử dụng 32 bit để đánh địa chỉ của host. Giả sử 1 router có 4 links từ 0 đến 3, và gói tin có thể chuyển qua link interface như hình sau:**

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3



- a. Cung cấp 1 bảng chuyển sử dụng prefix matching dài nhất và chuyển gói tin qua link interface đúng.  
b. Bạn xác định bảng chuyển như thế nào với các địa chỉ đích như dưới đây

```
11001000 10010001 01010001 01010101
11100001 01000000 11000011 00111100
11100001 10000000 00010001 01110111
```

Giải:

a.

Prefix Match	Link Interface
11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

- b. Prefix match for first address is 5<sup>th</sup> entry: link interface 3  
Prefix match for second address is 3<sup>rd</sup> entry: link interface 2  
Prefix match for third address is 4<sup>th</sup> entry: link interface 3

**P12:** giả sử 1 router có kết nối tới 3 subnet 1,2,3. Mỗi subnet có dạng 223.1.17/24. Giả sử rằng sub1 có thể hỗ trợ cho 63 interface, sub 2 hỗ trợ cho 95 interface, sub 3 hỗ trợ cho 16 interface. Cung cấp 3 địa chỉ mạng thỏa mãn các yêu cầu trên.

Trả lời:

223.1.17.0/26      vì  $63 < 2^6$   
223.1.17.128/25    vì  $95 < 2^7$   
223.1.17.192/28    vì  $16 = 2^4$

**P15:** Hãy xem xét 1 subnet 128.119.40.128/26. Đưa ra 1 ví dụ về 1 địa chỉ IP dạng xxx.xxx.xxx.xxx có thể gán cho mạng này. Giả sử 1 ISP sở hữu khối địa chỉ có dạng 128.119.40.64/25. Giả sử muốn tạo ra 4 mạng con từ khối này, với mỗi khối có cùng số địa chỉ IP. Subnet của 4 mạng con là gì?

Trả lời:

- Địa chỉ IP nằm trong mạng có subnet 128.119.40.128/26:  
Ta có 10000000 01110111 00101000 10000000  
Các IP từ 128.119.40.128  $\Leftrightarrow$  10000000 01110111 00101000 10000000  
đến 128.119.40.191  $\Leftrightarrow$  10000000 01110111 00101000 10111111
- Với khối địa chỉ 128.119.40.64/25  $\Rightarrow$  10000000 01110111 00101000 01000000  
25 bit đầu là cố định  
Vì bit thứ 26 = 1 nên ta còn 6 bit nữa để phân biệt địa chỉ IP (không tự tin cho câu nói này)  
 $\Rightarrow$  có tất cả  $2^6 = 64$   
Mỗi mạng con có số IP bằng nhau nên mỗi mạng có  $64/4 = 16 = 2^4$   
 $\Rightarrow$  Ta dùng 4 bit cuối để phân biệt địa chỉ IP, 2 bit tiếp theo ta để phân biệt 4 mạng con với nhau  
 $\Rightarrow$  4 mạng con có subnet là:  
128.119.40.64/28 - 10000000 01110111 00101000 01000000  
128.119.40.80/28 - 10000000 01110111 00101000 01010000  
128.119.40.96/28 - 10000000 01110111 00101000 01100000  
128.119.40.112/28 - 10000000 01110111 00101000 01110000

**P17:** Giả sử gửi 2400 byte datagram vào trong một link, link này có MTU của 700bytes. Cho rằng datagram gốc được đánh dấu với định danh số 422. Có bao nhiêu cò được tạo ra? Giá trị của mỗi trường khác nhau trong datagram được tạo ra có liên quan gì đến phân mảnh?

Trả lời:

Kích cỡ tối đa của trường dữ liệu trong mỗi phân đoạn = 680 byte (700-20 byte IP header)

Do đó số phân đoạn cần đến =  $(2400-20)/680=4$

Mỗi phân đoạn sẽ có một định danh số 422 và loại trừ ra cái cuối cùng có kích cỡ 700byte (bao gồm IP header). Gói cuối cùng có kích cỡ 360byte(bao gồm IP header). Vị trí của 4 phân đoạn sẽ là 0, 85, 170, 255. Ba cái phân đoạn đầu tiên có cờ bằng 1, phân đoạn cuối cùng có cờ = 0.

**P18:** Giả sử datagram có giới hạn là 1500byte (bao gồm header) giữa host nguồn A và host đích B. Thừa nhận rằng có 20byte IP header. Hỏi có bao nhiêu datagram cần đến để gửi 1 MP3 bao gồm 5 triệu byte ?. giải thích cho tính toán của bạn.

Trả lời:

1 file MP3 = 5000000byte

Thừa nhận dữ liệu mang theo TCP segment, với mỗi TCP segment có 20byte của header. Mỗi datagram có thể mang  $1500-40=1460$  byte của file MP3

Số datagram cần đến =  $5.10^6/1460 = 3425$ .

Datagram cuối cùng  $960 + 40 = 1000$ byte. Chú ý rằng, tại đó không có phân mảnh – host nguồn không thể tạo một datagram lớn hơn 1500 và cái datagram sẽ nhỏ hơn MTUs ở the links.

**P20:** Giả sử bạn vừa khám phá ra số của host đằng sau một NAT. Bạn theo dõi thấy rằng lớp IP được đánh dấu với một số là định danh, các số đó liên tiếp trên mỗi IP packet. Số định danh của IP packet đầu tiên được tạo ra bởi host là một số ngẫu nhiên, và số định danh của những IP packet tiếp theo là tiếp tục. Thừa nhận tất cả IP packet tạo ra bởi hosts đằng sau NAT đã được gửi ra bên ngoài.

- Dựa vào những theo dõi và thừa nhận, bạn có thể phát hiện ra tất packet được gửi bởi NAT ra bên ngoài. Bạn có thể đưa ra một vài kĩ thuật đơn giản- kĩ thuật phát hiện ra số duy nhất của host đằng sau NAT ? Bào chữa cho câu trả lời của bạn.
- Nếu số định danh không được chỉ ra liên tục nhưng ngẫu nhiên, kĩ thuật của bạn nên có là gì?

Trả lời:

- Kể từ khi tất cả IP packet được gửi ra ngoài, vậy chúng ta có thể sử dụng một packet phát hiện ra bản ghi tất cả IP packet tạo ra bởi host đằng sau một NAT. Khi mỗi host tạo ra một chuỗi của IP packet với số liên tiếp và một định danh khác hẳn nhau lúc ban đầu, chúng có thể tạo nhóm IP packet với IDs liên tục trong một khối (đám).Số của khối là số của hosts đằng sau NAT.

Tuy nhiên, nếu số định danh không chỉ ra liên tiếp mà được chỉ ra ngẫu nhiên, kĩ thuật được gợi ý ở phần (a) sẽ không làm được gì, khi đó sẽ không thể tạo nhóm cho dữ liệu được phát hiện ra

## Chapter5: Link Layer – Review Question

**R2. Nếu tất cả các đường kết nối trên Internet đều cung cấp dịch vụ truyền thông tin cậy thì giao thức TCP có cần thiết nữa hay không ??? Tại sao có và tại sao không?**

Mặc dù mỗi liên kết đảm bảo rằng một gói tin IP được gửi qua liên kết sẽ được nhận được ở đầu kia của liên kết mà không có lỗi, nó không được bảo đảm rằng các gói tin IP đến điểm đến cuối cùng trong thứ tự hợp lý. Với IP, các gói trong cùng một kết nối TCP có thể sử dụng các tuyến khác nhau trong mạng, và do đó đến không đúng thứ tự. TCP vẫn còn cần thiết để cung cấp kết thúc nhận của các ứng dụng byte dòng trong thứ tự đúng. Ngoài ra, IP có thể bị mất các gói tin do vòng lặp định tuyến.

**R3. Những dịch vụ nào tầng liên kết dữ liệu có thể cung cấp cho tầng mạng ? Những dịch vụ nào trong đó tương tự với IP, với TCP ???**

Framing ( IP and TCP), link access, reliable delivery (TCP), flow control (TCP), error detection ( IP and TCP), error correction, full duplex( TCP)

**R4. Giả sử 2 nút bắt đầu truyền dữ liệu cùng lúc với kích thước packet là L bit(broadcast không phải p2p) với tốc độ R. Ký hiệu sự chậm trễ phiên truyền giữa 2 nút là d. Hỏi có xung đột hay không nếu  $d < L/R$ . giải thích??**

Có thể có xung đột khi một nút đang truyền và gặp packet trả về cho nút kia

**R5. Trong Phần 5.3, chúng ta đã liệt kê bốn đặc tính mong muốn của một broadcast channel. Giao thức Slotted ALOHA có những đặc tính nào, giao thức token passing có những đặc tính nào?**

4 đặc tính:

- When only one node has data to send, that node has a throughput of R bps.
- Khi M nút dữ liệu gửi đi, mỗi nút được gửi vs thông lượng  $R/M$  bps. Đây chỉ là tốc độ trung bình trong một khoảng thời gian
- giao thức được triển khai phân tán, nghĩa là ko có nút giữ vai trò điều phối(khi nút đó hỏng, mạng sẽ sụp đổ)
- giao thức đơn giản để cài đặt không cao

Slotted Aloha: 1, 2 and 4

Token ring: 1, 2, 3, 4.

**R7. Tại sao giao thức token-ring sẽ là không hiệu quả nếu một mạng LAN có chu vi lớn ?**

Vì khi một nút gửi đi 1 frame, nó phải đợi frame đi hết 1 vòng trước khi nút bắt đầu 1 phiên truyền mới. vì vậy nếu  $L/R$  nhỏ như  $t_{prop}$  thì giao thức sẽ không hiệu quả

**R8. Độ lớn trường địa chỉ MAC ? Trường Ipv4 ? Trường Ipv6?**

$2^{48}$  MAC addresses;

$2^{32}$  IPv4 addresses;

$2^{128}$  IPv6 addresses.

**R10. Tại sao gói tin truy vấn ARP lại được gửi broadcast ? tại sao gói tin trả lời ARP lại chỉ gửi đến 1 địa chỉ MAC duy nhất ?**

*Vì khi gửi ARP query thiết bị chưa biết được địa chỉ MAC của interface nó gửi đến. Khi gửi lại gói tin trả lời thì địa chỉ MAC của thiết bị gửi đã có trong gói ARP query nên không cần phải gửi broadcast nữa.*

**R11. Với mạng ở Figure 5.19, router có 2 ARP modules, mỗi modules có bảng ARP của nó. Có thể hay không 1 địa chỉ MAC xuất hiện ở cùng 1 bảng ARP**

*Không thể vì mỗi mạng Lan có 1 tập adapter gắn vs nó, với mỗi adapter thì có 1 địa chỉ MAC duy nhất.*

**R14. Trong CSMA/CD sau lần xung đột thứ 5. Xác suất một nút chọn  $K=4$  là bao nhiêu? Tương ứng với  $K=4$  thì độ trễ là bao nhiêu với mạng 10Mbps.**

*Sau lần thứ 5 xung đột nút sẽ chọn  $K$  trong  $\{0,1,2, \dots, 31\}$  nên xác suất là  $1/32$ .*

*Thời gian chờ:  $4 \cdot 512 \cdot 0,1 = 204,8$  microseconds*

**R16. Số VLANs maximum có thể được cấu hình trên switch được cung cấp giao thức 802.1 . Why??**

*vì có 12 bit VLANs được định danh nên  $2^{12} = 4096$  VLANs*

cuu duong than cong . com

cuu duong than cong . com

**Chapter5: Link Layer – Problem**

**P1:** Giả sử nội dung thông tin của một packet là 1 mẫu bit 1100 1011 1001 1101 và tính chẵn lẻ cũng được sử dụng. Giá trị của trường chứa các bit chẵn lẻ đối với trường hợp sử dụng bit chẵn lẻ 2 chiều là gì? Câu trả lời của bạn nên là trường checksum có độ dài nhỏ nhất được sử dụng.

**Trả lời:**

1 1 1 0 1

1 0 1 1 1

1 0 0 1 0

1 1 0 1 1

0 0 0 1 1

**P2:** Cho một ví dụ cho thấy kiểm tra bit chẵn lẻ 2 chiều có thể sửa và phát hiện được 1 lỗi bit. Cho một ví dụ cho thấy 2 lỗi bit có thể được phát hiện nhưng không được sửa.

**Trả lời:**

Giả sử chúng ta bắt đầu với ma trận chẵn lẻ 2 chiều nhị phân:

0 0 0 0

1 1 1 1

0 1 0 1

1 0 1 0

Với 1 lỗi bit ở hàng 2, cột 3, tính chẵn lẻ của hàng 2 và cột 3 giờ bị sai, thể hiện trong ma trận sau:

0 0 0 0

1 1 0 1

0 1 0 1

1 0 1 0

Bây giờ giả sử có 1 lỗi bit ở hàng 2, cột 2 và cột 3. Tính chẵn lẻ của hàng 2 giờ đã đúng. Tính chẵn lẻ của cột 2 và 3 sai, nhưng ta không thể phát hiện ra lỗi sai xảy ra ở hàng nào.

0 0 0 0

1 0 0 1

0 1 0 1

1 0 1 0

Ví dụ trên cho thấy là 2 lỗi bit có thể được phát hiện (nếu không được sửa).

**P3:** Giả sử phần thông tin của 1 packet (D trong phần 5.4) chứa 10 bytes gồm 8 bit nhị phân không dấu ASCII miêu tả xâu “Link Layer”. Tính Internet Checksum cho dữ liệu này.

**Trả lời:**

01001100 01101001

+ 01101110 01101011

-----

10111010 11010100

+ 00100000 01001100

-----

11011011 00100000

+ 01100001 01111001

-----  
 00111100 10011010 (tràn, sau đó wrap around)  
 + 01100101 01110010

-----  
 10100010 00001100  
 Bù của tổng này là 01011101 11110011

**P4:** Xét vấn đề trước, nhưng thay vì chứa mã nhị phân của các số từ 0 tới 9, giả sử 10 bytes này chứa:

- Mã nhị phân của các số từ 0 tới 10
- Mã ASCII của các ký tự từ A tới J (viết hoa)
- Mã ASCII của các ký tự từ a tới j (viết thường)

Tính Internet Checksum cho các dữ liệu trên.

**Trả lời:**

a) Để tính Internet Checksum, chúng ta cộng thêm các giá trị độ dài 16 bit.

00000001 00000010  
 00000011 00000100  
 00000101 00000110  
 00000111 00001000  
 00001001 00001010

-----  
 00011001 00011110

Bù của tổng này là 11100110 11100001.

c) Để tính Internet Checksum, chúng ta cộng thêm các giá trị độ dài 16 bit:

01100001 01100010  
 01100011 01100100  
 01100101 01100110  
 01100111 01100111  
 01101000 01101001

-----  
 11111001 11111101

Bù của tổng này là 00000110 00000010.

**P5:** Xét bộ sinh 7 bit,  $G = 10011$ , giả sử  $D$  có giá trị 1010101010. Giá trị của  $R$  là gì?

**Trả lời:**

If we divide 10011 into 1010101010 0000, we get 1011011100, with a remainder of

$R=0100$ . Note that,  $G=10011$  is CRC-4-ITU standard.

Nếu ta chia 10011 cho 1010101010 0000, ta được 1011011100, với dư là 0100. Chú ý rằng,  $G = 10011$  là chuẩn CRC-4-ITU.

**P6:** Xét như bài trên, nhưng giả sử  $D$  có giá trị :

- 1001000101.
- 1010001111 .
- 0101010101.

**Trả lời :**



- a) ta được 1000100011, với số dư  $R=0101$ .
- b) ta được 1011111111, với số dư  $R=0001$ .
- c) ta được 0101101110, với số dư  $R=0010$ .

**P7:** Chúng ta tìm hiểu 1 số thuộc tính của CRC. Với bộ sinh  $G (= 1001)$  được cho trong Section 5.2.3, trả lời các câu hỏi sau:

- a. Tại sao nó có thể phát hiện bất kì lỗi đơn nào trong dữ liệu  $D$ ?
- b.  $G$  ở trên có thể phát hiện bất kì 1 số lẻ các lỗi bit không? tại sao?

**Trả lời:**

a. Không mất tính tổng quát, giả sử bit thứ  $i$  được “lật”, nơi  $0 \leq i \leq d+r-1$  và giả sử bit ít quan trọng nhất là bit thứ 0.

Một lỗi bit đơn nghĩa là dữ liệu nhận được là  $K=D*2^i \text{ XOR } R + 2^i$ . Rõ ràng là nếu chúng ta chia  $K$  cho  $G$ , thì số dư sẽ khác 0. Nhìn chung, nếu  $G$  chứa ít nhất hai 1's, thì một lỗi bit đơn luôn luôn có thể phát hiện.

b. Mấu chốt là  $G$  có thể chia cho 11 (số nhị phân), but any number of odd-number of 1's không thể chia cho 11. Vì thế, 1 dãy (không cần liên tục) của một số lẻ các bit lỗi không thể chia cho 11, vì thế nó không thể chia cho  $G$ .

**P8:** Ở Section 5.3, chúng ta đã rút ra được một phác thảo về hiệu suất của ALOHA chia khe, trong vấn đề này, chúng ta sẽ hoàn thiện nó.

a, Nhớ lại là khi có  $N$  node hoạt động, hiệu suất của ALOHA chia khe là  $Np(1-p)^{N-1}$ . Tìm giá trị của  $p$  cho hiệu suất đó lớn nhất.

b, Sử dụng giá trị tìm được ở câu a, tìm hiệu suất của ALOHA chia khe bằng cách cho  $N$  tiến tới vô cùng. Gợi ý:  $(1 - 1/N)^N$  tiến tới  $1/e$  khi  $N$  tiến tới vô cùng.

**Trả lời:**

a.

$$\begin{aligned} E(p) &= Np(1-p)^{N-1} \\ E'(p) &= N(1-p)^{N-1} - Np(N-1)(1-p)^{N-2} \\ &= N(1-p)^{N-2}((1-p) - p(N-1)) \end{aligned}$$

$$E'(p) = 0 \Rightarrow p^* = \frac{1}{N}$$

b.

$$E(p^*) = N \frac{1}{N} \left(1 - \frac{1}{N}\right)^{N-1} = \left(1 - \frac{1}{N}\right)^{N-1} = \frac{\left(1 - \frac{1}{N}\right)^N}{1 - \frac{1}{N}}$$

$$\lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right) = 1 \quad \lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right)^N = \frac{1}{e}$$

Thus

$$\lim_{N \rightarrow \infty} E(p^*) = \frac{1}{e}$$

**P9:** Chứng minh là hiệu suất lớn nhất của ALOHA thuần túy là  $1/(2e)$ . Chú ý: vấn đề này sẽ rất dễ nếu bạn đã trả lời xong câu hỏi ở trên.

**Trả lời:**

$$E(p) = Np(1-p)^{2(N-1)}$$

$$\begin{aligned} E'(p) &= N(1-p)^{2(N-2)} - Np2(N-1)(1-p)^{2(N-3)} \\ &= N(1-p)^{2(N-3)}((1-p) - p2(N-1)) \end{aligned}$$

$$E'(p) = 0 \Rightarrow p^* = \frac{1}{2N-1}$$

$$E(p^*) = \frac{N}{2N-1} \left(1 - \frac{1}{2N-1}\right)^{2(N-1)}$$

$$\lim_{N \rightarrow \infty} E(p^*) = \frac{1}{2} \cdot \frac{1}{e} = \frac{1}{2e}$$

cuu duong than cong . com

**P10:** Giả sử có 2 node, A và B, sử dụng giao thức ALOHA chia khe. Giả sử node A có nhiều dữ liệu để chuyển hơn node B, và xác suất truyền lại của A là  $p_A$ , lớn hơn của B là  $p_B$ .

- Tìm công thức thông lượng trung bình của node A. Tổng hiệu suất của giao thức với 2 node đó là gì?
- Nếu  $p_A = 2p_B$  thì thông lượng trung bình của A có gấp 2 lần của B không? Tại sao? Nếu không, có thể chọn  $p_A$  và  $p_B$  thế nào để điều đó xảy ra.
- Tổng quát, giả sử có N nodes, trong đó node A có xác suất truyền lại là  $2p$  và các node khác có xác suất truyền lại là  $p$ . Tính thông lượng trung bình của node A và của các node khác.

**Trả lời:**

- a) Thông lượng trung bình của A là  $p_A(1-p_B)$

Tổng hiệu suất  $p_A(1-p_B) + p_B(1-p_A)$

- b) Thông lượng của A là  $p_A(1-p_B) = 2p_B(1-p_B) = 2p_B - 2(p_B)^2$

Thông lượng của B là  $p_B(1-p_A) = p_B(1-2p_B) = p_B - 2(p_B)^2$

$\Rightarrow$  Thông lượng của A không gấp 2 lần của B

Để thông lượng của A gấp 2 lần của B thì

$$p_A(1-p_B) = 2p_B(1-p_A)$$

$$\Rightarrow p_A = 2 - (p_A/p_B)$$

- c) Thông lượng của A là  $2p(1-p)^{N-1}$ , và các node khác là  $p(1-p)^{N-2}(1-2p)$

**P11:** Giả sử có 4 node hoạt động là A,B,C,D đang cạnh tranh truy cập vào kênh sử dụng ALOHA chia khe. Giả sử mỗi node có vô hạn packet để gửi. Mỗi node cố gắng gửi vào mỗi slot với xác suất p. Slot đầu được đánh số 1, slot 2 đánh số 2, cứ thế.

- Xác suất để node A gửi thành công ngay lần đầu tiên tại slot 5 là bao nhiêu?
- Xác suất để vài node (có thể là A, B, C hoặc D) gửi thành công tại slot 4?
- Xác suất để lần gửi thành công đầu tiên xảy ra lại slot 3?
- Hiệu suất của hệ 4 node này là bao nhiêu?

**Trả lời:**

a)  $(1-p(A))^4 p(A)$

Trong đó:

$p(A)$  = xác suất A thành công tại 1 slot

$p(A) = p(A \text{ gửi và B,C,D không gửi})$

$= p(A \text{ gửi}).p(B \text{ không gửi}).p(C \text{ không gửi}).p(D \text{ không gửi})$

$= p(1-p)(1-p)(1-p) = p(1-p)^3$

Do đó:

$p(A \text{ thành công trong lần gửi đầu tại slot 5})$

$= (1-p(A))^4 . p(A)$

$= (1-p(1-p)^3)^4 . p(1-p)^3$

b)  $p(A \text{ thành công ở slot 4}) = p(1-p)^3$

$p(B \text{ thành công ở slot 4}) = p(1-p)^3$

$p(C \text{ thành công ở slot 4}) = p(1-p)^3$

$p(D \text{ thành công ở slot 4}) = p(1-p)^3$

$p(A,B,C \text{ hoặc } D \text{ thành công tại slot 4}) = 4 p(1-p)^3$

(bởi vì các sự kiện này loại trừ lẫn nhau)

c)  $p(\text{vài node thành công tại 1 slot}) = 4p(1-p)^3$

$p(\text{không có node nào thành công tại 1 slot}) = 1-4p(1-p)^3$

Do đó:

$p(\text{lần thành công đầu tiên xảy ra ở slot 3}) = p(p(\text{không có node nào thành công tại 2 slot đầu}).p(\text{vài node}$

$\text{thành công tại slot 3}) = (1 - 4 p(1-p)^3)^2 4p(1-p)^3$

d)  $\text{hiệu suất} = p(\text{thành công tại 1 slot}) = 4 p(1-p)^3$

**P13:** Giả sử một kênh broadcast(quảng bá) với N node và transmission rate là R bps. Giả sử kênh broadcast sử dụng giao thức hỏi vòng (polling) (với 1 node polling được thêm vào) cho đa truy cập. Giả sử khoảng thời gian từ khi một node hoàn thành việc truyền đến khi node tiếp theo được phép chuyển (trễ hỏi vòng – polling delay) là  $d_{poll}$ . Giả sử trong 1 vòng hỏi, 1 node được phép truyền tối đa Q bit. Thông lượng tối đa của kênh broadcast này là bao nhiêu?

**Trả lời:**

Thời gian 1 vòng hỏi là

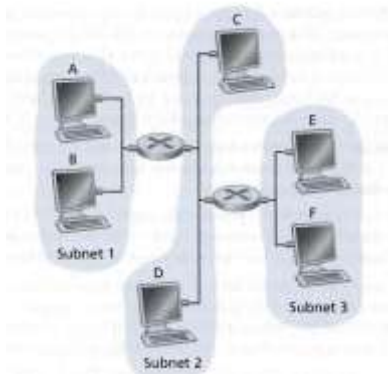
$$N(Q/R + d_{poll})$$

Số bit được truyền trong 1 vòng hỏi là NQ. Thông lượng tối đa vì thế sẽ là:

$$\frac{NQ}{N(\frac{Q}{R} + d_{poll})} = \frac{R}{1 + \frac{d_{poll}R}{Q}}$$

**P14:** giả sử 3 mạng LANs kết nối với nhau bởi 2 router, xem hình 5.8

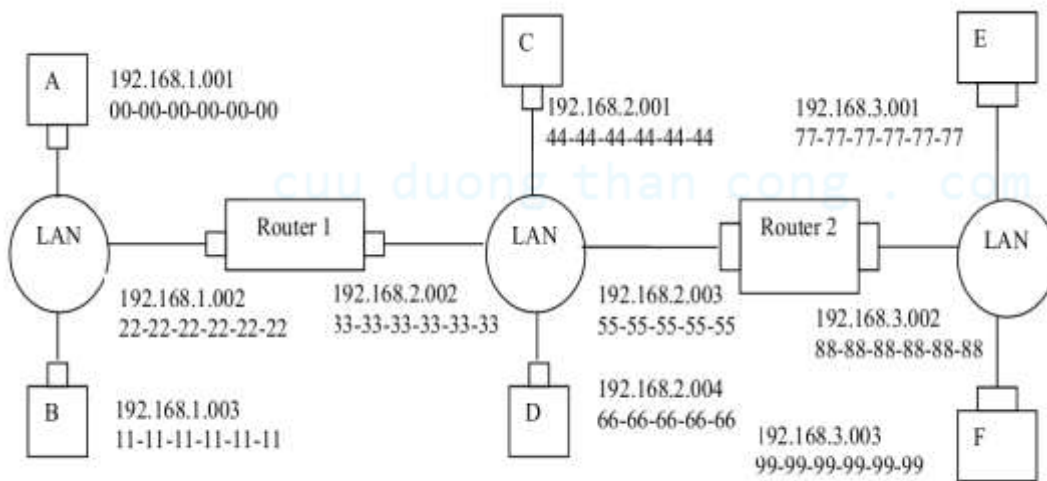
- a. Đặt địa chỉ IP cho tất cả các interface này. Với Subnet 1 thì ta dùng địa chỉ có dạng 192.168.1.xxx, Với Subnet 2 thì ta dùng địa chỉ có dạng 192.168.2.xxx, Với Subnet 3 thì ta dùng địa chỉ có dạng 192.168.3.xxx



- b. Đặt địa chỉ MAC cho các adapter.  
c. Gửi 1 IP datagram từ host E tới host B. Giả sử tất cả các bảng ARP đều được cập nhật. Liệt kê các bước, giống như ví dụ 1 router ở Section 5.4.2.  
d. Làm lại câu c, giờ giả sử bảng ARP ở host gửi chưa có dữ kiện gì, các bảng khác đều được cập nhật.

**Trả lời:**

a,b: xem hình:



- c)
1. Bảng Forwarding ở E cho thấy là datagram được định tuyến tới interface 192.168.3.002.

2. Adapter ở E tạo 1 Ethernet packet với địa chỉ đích là 88-88-88-88-88-88.

3. Router 2 nhận packet và giải nén datagram. Bảng forwarding ở router này chỉ ra là datagram này phải được định tuyến tới 192.168.2.002.

4. Router 2 gửi Ethernet packet với địa chỉ đích 33-33-33-33-33-33 và địa chỉ nguồn 55-55-55-55-55-55 thông qua interface của nó với địa chỉ IP là 192.168.2.003.

5. Tiến trình tiếp tục đến khi packet đến được host B.

d) ARP ở E lúc này phải xác định địa chỉ MAC của 192.168.3.002. Host E gửi 1 packet truy vấn ARP trong 1 frame Ethernet quảng bá @@ Router 2 nhận packet truy vấn và gửi tới Host E một packet ARP phản hồi. Packet ARP phản hồi này được mang đi bởi 1 frame Ethernet với địa chỉ đích Ethernet là 77-77-77-77-77-77.

**P15:** Xét hình 5.38, giờ chúng ta thay vị trí của router giữa subnet 1 và 2 bằng 1 switch S1, và đặt tên router giữa subnet 2 và 3 là R1.

- Gửi 1 IP datagram từ host E tới host F. Host E có phải hỏi router R1 giúp forward datagram đi không? Tại sao? Trong Ethernet frame chứa IP datagram, địa chỉ IP đích, nguồn và địa chỉ MAC là gì?
- Giả sử E muốn gửi 1 IP datagram tới B, và giả sử bộ nhớ cache ARP của E không chứa địa chỉ MAC của B. B có phải gửi 1 ARP truy vấn để tìm địa chỉ MAC của B không? Tại sao? Trong Ethernet frame (chứa IP datagram gửi tới B) được chuyển tới router 1, địa chỉ đích, nguồn và địa chỉ MAC là gì?
- Giả sử host A muốn gửi 1 IP datagram tới host B, bộ nhớ cache ARP của A không chứa địa chỉ MAC của B và bộ nhớ cache ARP của B không chứa địa chỉ MAC của A. Giả sử là bảng forwarding của switch S1 chỉ chứa entries cho host B và router 1. Vì thế, A sẽ broadcast một ARP request message. Switch S1 sẽ thực hiện hành động nào khi nó nhận được ARP message? Router R1 có cùng nhận được ARP request message đó không? Nếu nhận được, R1 có forward message tới Subnet 3 không? Ngay khi host B nhận được ARP request message, nó có gửi lại host A 1 ARP message phản hồi không? Tại sao? Switch S1 sẽ làm gì ngay khi nó nhận được ARP message phản hồi từ host B?

### Trả lời:

- Không. E có thể kiểm tra tiền tố subnet (subnet prefix) của địa chỉ IP của host F, sau đó nhận ra là F ở cùng 1 mạng LAN. Do đó, E sẽ không gửi packet tới router mặc định R1.

Ethernet frame từ E tới F:

IP nguồn = địa chỉ IP của E

IP đích = địa chỉ IP của F

Địa chỉ MAC nguồn = địa chỉ MAC của E

Địa chỉ MAC đích = địa chỉ MAC của F

- Không, vì chúng không ở trong cùng 1 mạng LAN. E có thể nhận ra điều này bằng cách kiểm tra địa chỉ IP của B.

Ethernet frame từ E tới R1:

IP nguồn = địa chỉ IP của E

IP đích = địa chỉ IP của B

Địa chỉ MAC nguồn = địa chỉ MAC của E

Địa chỉ MAC đích = địa chỉ MAC của interface của R1 kết nối tới Subnet 3

- Switch S1 sẽ broadcast Ethernet frame thông qua cả 2 interface của nó để địa chỉ đích của ARP frame nhận được là 1 địa chỉ broadcast. Và nó nhận ra là A nằm trong Subnet 1 đang kết nối tới S1 tại interface kết nối tới Subnet 1. Và S1 sẽ cập nhật bảng forwarding của nó, bao gồm cả entry cho host A.

Có, router R1 cũng nhận được ARP request message, nhưng R1 không forward message tới Subnet 3.

B không gửi ARP message truy vấn hỏi địa chỉ MAC của A, địa chỉ này có thể nhận được từ message truy vấn của A.

Khi Switch S1 nhận message phản hồi từ B, nó sẽ thêm 1 entry cho host B vào bảng forwarding, sau đó thả frame vừa nhận được với địa chỉ đích là host A trong cùng 1 interface như host B (ví dụ: A và B cùng nằm trong 1 mạng LAN).

**P16:** Vẫn vấn đề như trên, nhưng giả sử bây giờ router giữa subnet 2 và 3 được thay bằng 1 switch. Trả lời câu hỏi từ a đến c với giả thiết mới.

### Trả lời:

Gọi switch giữa subnet 2 và 3 là S2. Theo đề bài, router R1 giữa subnet 2 và 3 giờ được thay thế bởi switch S2.

a) Không. E có thể kiểm tra tiền tố subnet (subnet prefix) của địa chỉ IP của host F, và nhận ra là F ở trong cùng 1 mạng LAN. Vì thế, E sẽ không gửi packet tới S2.

Ethernet frame từ E tới F:

IP nguồn = địa chỉ IP của E

IP đích = địa chỉ IP của F

Địa chỉ MAC nguồn = địa chỉ MAC của E

Địa chỉ MAC đích = địa chỉ MAC của F

b) Có, bởi vì E muốn biết địa chỉ MAC của B. Vì thế, E sẽ gửi 1 packet ARP truy vấn với địa chỉ MAC đích là địa chỉ broadcast.

Packet truy vấn này sẽ được broadcast lại bởi switch 1, và cuối cùng được host B nhận.

Ethernet frame từ E tới S2:

IP nguồn = địa chỉ IP của E

IP đích = địa chỉ IP của B

MAC nguồn = địa chỉ MAC của E

MAC đích = địa chỉ MAC broadcast: FF-FF-FF-FF-FF-FF.

c)

Switch S1 sẽ broadcast frame Ethernet thông qua cả 2 interface của nó tại địa chỉ đích của ARP frame nhận được là 1 địa chỉ broadcast. Và nó nhận ra là A nằm trong Subnet 1 đang kết nối tới S1 tại interface kết nối tới Subnet 1. Và S1 sẽ cập nhật bảng forwarding của nó, bao gồm cả entry cho host A.

Có, router S2 cũng nhận được ARP request message này, và S2 sẽ broadcast packet truy vấn này tới tất cả các interface của nó.

B không gửi message truy vấn hỏi địa chỉ MAC của A, địa chỉ này có thể nhận được từ message truy vấn của A.

Khi S1 nhận được tin nhắn phản hồi của B, nó sẽ thêm 1 entry cho host B vào bảng forwarding, sau đó thả frame vừa nhận được với địa chỉ đích là host A trong cùng 1 interface như host B (ví dụ: A và B cùng nằm trong 1 mạng LAN).

**P17:** Nhớ lại về giao thức CSMA/CD, adapter đợi  $K \cdot 512$  bit times sau khi có xung đột, K được rút ngẫu nhiên. Với  $K = 100$ , adapter phải đợi bao lâu tới khi trở lại bước 2 với 10Mbps Ethernet? với 100Mbps Ethernet?

**Trả lời:**



Đợi 51200 bit times. Với 10Mbps, thời gian đợi là:

$$\frac{51.2 \times 10^3 \text{ bits}}{10 \times 10^6 \text{ bps}} = 5.12 \text{ msec}$$

Với 100Mbps, thời gian đợi là 512  $\mu$ sec.

**P18:** Giả sử node A và B ở trên cùng 1 bus 10Mbps Ethernet, và trễ lan tỏa giữa 2 node là 325 bit times. Giả sử node A bắt đầu gửi 1 frame và, trước khi kết thúc, node B bắt đầu gửi 1 frame. A có kết thúc việc gửi trước khi nó phát hiện là B đã gửi không? Tại sao? Nếu câu trả lời là có, thì A đã hiểu nhầm là frame của nó được chuyển thành công mà không có xung đột. Gợi ý: giả sử tại thời điểm  $t = 0$  bit times, A bắt đầu gửi 1 frame. Trong trường hợp xấu nhất, A gửi 1 frame có kích thước nhỏ nhất là  $512 + 64$  bit times. Nên A sẽ kết thúc việc gửi tại  $t = 512 + 64$  bit times. Thế nên, câu trả lời là không, nếu tín hiệu của B tới được A trước bit time  $t = 512 + 64$  bits. Trong trường hợp xấu nhất, khi nào thì dấu hiệu của B tới được A?

#### Trả lời:

Tại thời điểm  $t = 0$ , A bắt đầu gửi. Tại  $t = 576$ , A sẽ kết thúc việc gửi. Trong trường hợp xấu nhất, B bắt đầu gửi tại thời điểm  $t = 324$ , là thời điểm ngay trước khi bit đầu tiên của frame của A tới được B. Tại thời điểm  $t = 324 + 325 = 649$ , bit đầu tiên của B tới được A. Vì  $649 > 576$ , A kết thúc việc gửi trước khi phát hiện ra là B đã gửi. Nên A hiểu nhầm là frame của nó được gửi thành công mà không có xung đột.

**P19:** Giải thích tại sao lại đòi hỏi kích thước frame là nhỏ nhất với Ethernet. Ví dụ, 10Base Ethernet yêu cầu kích thước nhỏ nhất của frame bắt buộc là 64 bytes (nếu bạn đã làm vấn đề trước, bạn có thể hiểu được lý do). Giờ giả sử khoảng cách giữa 2 ends of Ethernet LAN là  $d$ . Bạn có thể xây dựng 1 công thức để tìm kích thước frame nhỏ nhất mà 1 Ethernet packet cần không? Dựa vào lý do của bạn, kích thước tối thiểu của frame cần cho 1 Ethernet kéo dài 2 km là bao nhiêu?

#### Trả lời:

Theo lý do ở vấn đề trước, chúng ta cần đảm bảo là that one end of Ethernet có thể phát hiện xung đột trước khi nó hoàn thành việc chuyển frame. Vì thế, đòi hỏi kích thước frame phải là nhỏ nhất.

Để BW biểu thị băng thông của Ethernet. Xét trường hợp xấu nhất cho việc phát hiện xung đột của Ethernet:

1. Tại  $t = 0$  (bit times): A gửi 1 frame
2. Tại  $t = d_{\text{prop}} - 1$  (bit times): B gửi 1 frame ngay trước khi nó cảm nhận dc bit đầu tiên của A.
3. Tại  $t = 2d_{\text{prop}} - 2$  (bit times): nếu A kết thúc việc truyền bit cuối cùng của nó ngay trước khi frame của B đến A, thì A sẽ không thể phát hiện xung đột trước khi kết thúc việc truyền frame của nó. Do đó, để A có thể phát hiện xung đột trước khi kết thúc việc truyền, đòi hỏi kích thước tối thiểu của frame  $\geq 2d_{\text{prop}} - 1$  (bit times).

Giả sử tốc độ lan tỏa tín hiệu trong 10BASE-T Ethernet là  $1.8 \times 10^8$  m/sec. Với  $d_{\text{prop}} = d / (1.8 \times 10^8) \times \text{BW}$  (ở đây, chúng ta cần convert trễ lan truyền từ giây sang bit times để phù hợp với đặc trưng của link Ethernet). Hoặc ta chọn  $2 \times d / (1.8 \times 10^8) \times \text{BW}$  (bits). Nếu  $d = 2$  km, thì cần kích thước tối thiểu của frame là 222 bits.

**P20 :** Giả sử bạn có thể tăng tốc độ link của cáp Ethernet của bạn, sự nâng cấp này ảnh hưởng thế nào tới kích thước yêu cầu tối thiểu của packet ? Nếu bạn nâng cấp cáp của bạn tới 1 tốc độ cao hơn và nhận ra là bạn không thể thay đổi kích thước packet, bạn sẽ làm gì để giữ cho quá trình hoạt động đúng ?

#### Trả lời :

Dựa vào giải pháp ở vấn đề trước, bạn biết là link có tốc độ cao hơn thì đòi hỏi kích thước yêu cầu tối thiểu của packet lớn hơn. Nếu bạn không thể thay đổi kích thước packet, thì bạn có thể thêm các switch hoặc router để phân vùng mạng LAN của bạn, để đảm bảo là kích thước mỗi vùng mạng LAN là đủ nhỏ so với kích thước nhỏ của frame.

**P21 :** Giả sử node A và B ở cùng 1 Ethernet bus 10Mbps, và trễ lan tỏa giữa 2 node là 245 bit times. Giả sử A và B gửi frame cùng lúc, frame xung đột, sau đó A và B chọn giá trị khác nhau của K trong thuật toán CSMA/CD. Giả sử không có node khác hoạt động, lần gửi lại của A và B có bị xung đột không ? Giả sử A và B bắt đầu gửi tại  $t = 0$  bit times. Chúng cùng phát hiện xung đột tại  $t = 245$  bit times. Chúng gửi xong tín hiệu tắc nghẽn tại  $t = 245 + 48 = 293$  bit times. Giả sử  $K_A = 0$  và  $K_B = 1$ . Tại thời điểm nào thì B sắp xếp gửi lại ? Tại thời điểm nào A bắt đầu gửi ? (Chú ý : node phải đợi 1 kênh rảnh rồi sau khi quay trở lại bước 2 – giao thức nhin.) Tại thời điểm nào thì tín hiệu của A tới được B ? B có trì hoãn việc gửi vào thời điểm đã được sắp xếp của nó ?

**Trả lời :**

Time, t	Event
0	A và B bắt đầu gửi
245	A và B phát hiện xung đột
293	A và B kết thúc việc gửi tín hiệu tắc nghẽn
$293 + 245 = 538$	Bit cuối cùng của B tới A; A phát hiện 1 kênh rảnh
$538 + 96 = 634$	A bắt đầu gửi
$293 + 512 = 805$	B quay lại bước 2
	B phải cảm nhận được kênh rảnh rồi for 96 bit times trước khi nó gửi
$634 + 246 = 879$	frame A gửi tới được B

Vì frame gửi lại của A tới B trước thời điểm B sắp xếp gửi lại ( $805 + 96$ ), B trì hoãn gửi trong khi A gửi lại. Vì thế A và B không xung đột. Thế nên nhân tố 512 xuất hiện trong thuật toán exponential backoff là đủ lớn.

**P22:** Xét 100BASE-T Ethernet 100 Mbps với tất cả các node kết nối trực tiếp tới 1 hub. Để có hiệu suất 0.50, khoảng cách lớn nhất giữa 1 node và hub nên là bao nhiêu? Giả sử chiều dài của 1 frame là 1000 bytes và không có lặp lại (repeater). Khoảng cách lớn nhất này có đảm bảo là node A đang gửi có thể phát hiện là các node khác đã gửi khi A đang gửi không? Tại sao? Khoảng cách lớn nhất đó so với chuẩn 100 Mbps thực tế thì thế nào? Giả sử tốc độ lan truyền tín hiệu trong 100BASE-T Ethernet là  $1.8 \cdot 10^8$  m/sec.

**Trả lời :**

Kích cỡ 1 frame là  $1000 \cdot 8 + 64 = 8064$  bit, vì phần preamble 64 bit được thêm vào frame.

Chúng ta cần  $1/(1+5a) = 0.5$  hoặc tương đương,  $a = 0.2 = t_{\text{prop}}/t_{\text{trans}}$ .  $t_{\text{prop}} = d/(1.8 \cdot 10^8)$  m/sec và  $t_{\text{trans}} = (8064 \text{ bit})/(10^8 \text{ bit/sec}) = 80.64 \mu\text{sec}$ . Tính ra ta được  $d = 2903$  mét.

Với nơi gửi là A, để phát hiện xem có nơi khác gửi trong thời gian A đang gửi,  $t_{\text{trans}}$  phải lớn hơn  $2t_{\text{prop}} = 2 \cdot 2903 \text{ m} / 1.8 \cdot 10^8 \text{ m/sec} = 32.36 \mu\text{sec}$ . Vì  $32.26 < 80.64$ , A sẽ phát hiện tín hiệu của B trước khi quá trình gửi của nó hoàn tất.

**P23 :** Giả sử 4 node, A, B, C và D cùng kết nối vào 1 hub thông qua cáp Ethernet 10Mbps. Khoảng cách giữa hub và 4 node lần lượt là 300m, 400m, 500m và 700m. Giao thức CSMA/CD được sử dụng cho Ethernet này. Giả sử tốc độ lan truyền tín hiệu là  $2 \cdot 10^8$  m/sec.

- Chiều dài tối thiểu yêu cầu của frame là bao nhiêu ? Tối đa là bao nhiêu ?
- Nếu tất cả frame đều có chiều dài 1500 bit, tìm hiệu suất của Ethernet này.

**Trả lời :**

- a) Chiều dài tối thiểu yêu cầu của frame là :

$$2 \cdot d_{\text{prop}} \cdot BW = 2 \cdot (500 + 700) / (2 \cdot 10^8) \cdot 10 \cdot 10^6 = 120 \text{ bit}$$

Không có chiều dài tối đa yêu cầu của frame.

- b) Hiệu suất:

$$1 / (1 + 5 \cdot d_{\text{prop}} / d_{\text{trans}}) = 1 / (1 + 5 \cdot 120 / 2 / 1500) = 0.83$$

**P25:** Giả sử có 2 node A, B ở hai đầu của 1 cable dài 800 m. và có 1 một frame 1500 bit cần chuyển qua cable đó. Cả 2 node đều truyền tại thời điểm  $t=0$ . Giả sử có 4 lần lặp giữa A và B. mỗi lần chèn có 20 bit trễ. Giả sử đường truyền là 100 Mbps và CSMA/CD with backoff intervals of multiples of 512 bits is used. Sau lần xung đột đầu tiên, A cho kết quả  $K=0$ , B cho kết quả  $K=1$  in the exponential backoff protocol. Không có tín hiệu nào bị nhiễu,  $t$  he 96-bit time delay.

- Độ trễ lan tỏa từ A tới B là? Giả sử tốc độ lan tỏa là  $2 \cdot 10^8$  m/s.
- Gói tin của A chuyển xong khi nào?
- Giả sử chỉ A cần gửi gói tin, và repeaters thay thành switch. Giả sử mỗi switch có 20-bit processing delay in addition to a store-and-forward delay.

Trả lời:

- a.

$$\begin{aligned} & \frac{800m}{2 \cdot 10^8 m/sec} + 4 \cdot \frac{20bits}{100 \times 10^6 bps} \\ &= (4 \times 10^{-6} + 0.8 \times 10^{-6}) sec \\ &= 4.8 \mu sec \end{aligned}$$

- First note, the transmission time of a single frame is give by  $1500 / (100Mbps) = 15$  micro sec, longer than the propagation delay of a bit.
  - At time  $t=0$ , both A and B transmit.
  - Vào thời điểm 4.8 micro giây A, B phát hiện xung đột, và kết thúc.
  - Vào thời điểm 9.6 micro giây bit cuối cùng của gói tin lỗi từ B tới A
  - Vào thời điểm 14,4 micro giây bit đầu tiên của frame được gửi lại đến B.
  - Vào thời điểm 14,4 + 15 gói tin của A đến B.

- c. Đường đi được chia làm 5 phần, độ trễ lan tỏa của mỗi phần là 0.8.

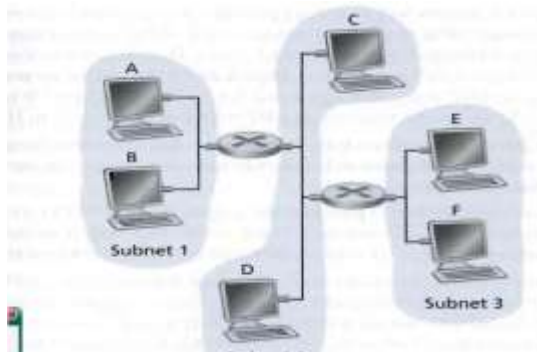
Độ trễ từ host A tới switch 1 là 15 = độ trễ truyền. switch 1 sẽ đợi 16 = 15 + 0.8 + 0.2 (độ trễ tiến trình) cho đến khi nó có gói tin để chuyển tới switch 2. Tương tự cho các switch còn lại. vậy tổng thời gian là:  $16 \cdot 4 + 15 + 0.8 = 79.8$  micro giây.

**P26:** Trong chuẩn Internet, người gửi dùng 96 lần bit giữa 2 lần gửi khung hình liên tiếp. Thời gian tạm dừng này được gọi là khoảng cách giữa các khung hình (inter-frame gap) và nó được sử dụng để cho phép 1 thiết bị tiếp nhận để hoàn thành tiến trình nhận và chuẩn bị cho lần nhận frame kế tiếp. Từ khi chuẩn Ethernet được quy định, đã có 1 cải tiến to lớn trong công nghệ bao gồm cả tốc độ của vi xử lý, bộ nhớ và tỷ lệ Ethernet. Nếu các tiêu chuẩn được viết lại, sự cải tiến này tác động như thế nào đến khoảng cách giữa các khung hình ?

**Solution:**

Nâng cao tốc độ của vi xử lý ám chỉ khoảng cách giữa các khung hình có thể được rút ngắn, bắt đầu từ lúc nó mất ít thời gian hơn để hoàn thành tiến trình nhận frame. Nhưng, nâng cao tốc độ cáp Ethernet ám chỉ khoảng cách giữa các khung hình phải được tăng lên.

**P 27.**



Cung cấp địa chỉ MAC và địa chỉ IP cho interface tại Host A , tất cả các router và Host F . Giả sử Host A gửi datagram đến Host F .Hãy chỉ ra địa chỉ MAC nguồn ( source ) và địa chỉ MAC đích (destination ) trong frame đóng gói IP datagram này như 1 frame đã được truyền .:

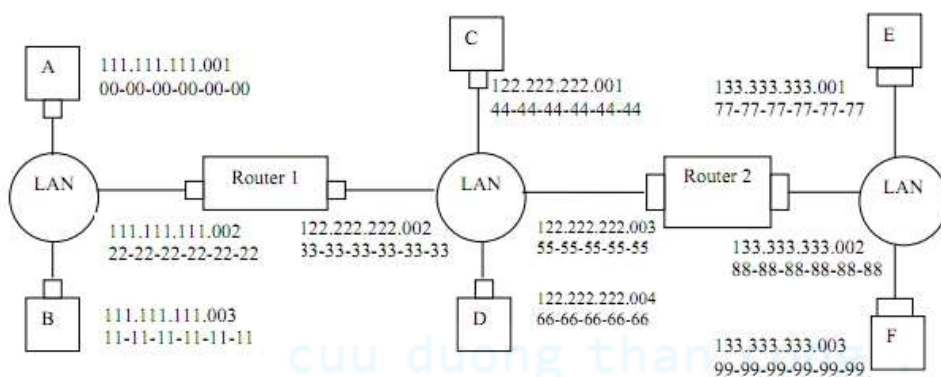
**I , Từ A đến left router .**

**II , Từ left router đến right router .**

**III , Từ right router đến F .**

Ngoài ra , cần chỉ ra địa chỉ IP nguồn và địa chỉ IP đích tại mỗi điểm .

**Giải :**



**I , A to left router :**

Source MAC : 00-00-00-00-00-00

Destination MAC : 22-22-22-22-22-22 .

Source IP : 111.111.111.001

Destination IP : 133.333.333.003

**II , Từ left đến right router :**

Source MAC : 33-33-33-33-33-33

Destination MAC : 55-55-55-55-55-55 .

Source IP : 111.111.111.001

Destination IP : 133.333.333.003

III , Từ right router đến F :

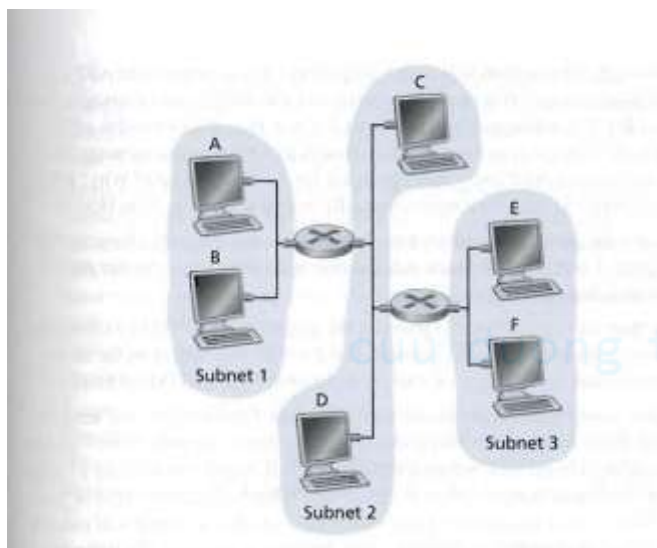
Source MAC : 88-88-88-88-88-88

Destination MAC : 99-99-99-99-99-99

Source IP : 111.111.111.001

Destination IP : 133.333.333.003

**P28:** Giả sử bây giờ router tận cùng bên trái trong hình 5.38 được thay thế bằng 1 switch. Host A,B,C và D và router bên phải là các sao kết nối vào switch này. Cung cấp cho địa chỉ MAC nguồn và đích trong frame đóng gói IP datagram như 1 frame được truyền ( I ) từ A đến switch, ( II ) từ switch đến router bên phải, ( III ) từ router bên phải đến F. Ngoài ra cung cấp địa chỉ IP nguồn và đích trong IP datagram được đóng gói trong frame tại những thời điểm này.



### Solution

I, Từ A đến switch:

Địa chỉ Mac nguồn: 00-00-00-00-00-00

Địa chỉ Mac đích: 55-55-55-55-55-55

IP nguồn : 111.111.111.001

IP đích: 133.333.333.003

II, Từ switch đến router bên phải

Địa chỉ Mac nguồn: 00-00-00-00-00-00

Địa chỉ Mac đích: 55-55-55-55-55-55

Địa chỉ IP nguồn: 111.111.111.001

Địa chỉ IP đích: 133.333.333.003

III, Từ router bên phải đến F:

Địa chỉ Mac nguồn: 88-88-88-88-88-88

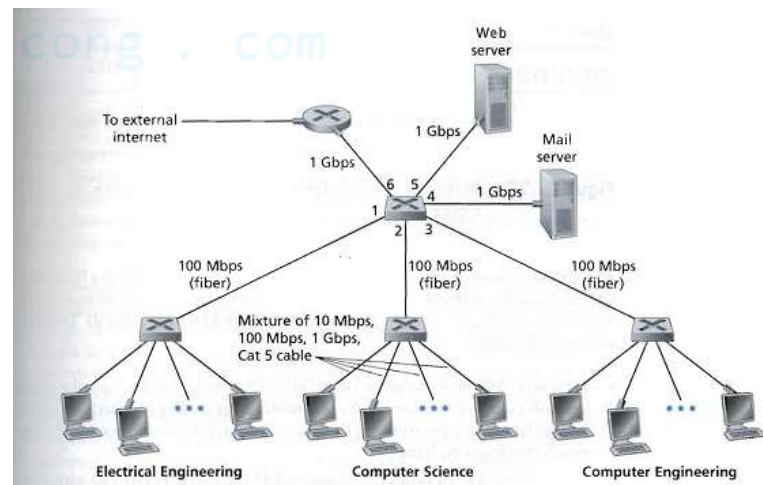
Địa chỉ Mac đích: 99-99-99-99-99-99

Địa chỉ IP nguồn: 111.111.111.001

Địa chỉ IP đích: 133.333.333.003

**P29:** Xem xét hình 5.26. Giả sử tất cả các các liên kết đều là 100 Mbps. Tổng thông lượng tối đa có thể đạt được trong 9 host và 2 server trong mạng này là bao nhiêu ? Bạn có thể giả định bất kỳ host hay server nào cũng có thể gửi đến bất kỳ 1 host hay server . Tại sao?

Nếu tất cả 11=9+2 nút gửi dữ liệu tốc độ tối đa có thể là 100 Mbps, tổng thông lượng đạt được là  $11 \times 100 = 1100$  Mbps





**P 30 .** Giả sử 3 departmental Switch được thay thế bởi các hub . Tất cả link vẫn là 100 Mbps . Tổng thông lượng đạt được giữa 9 host và 2 server trong mạng là bao nhiêu ? Host và server có thể gửi đến host và server khác được hay không ? Tại sao ?

Mỗi departmental hub là 1 tên miền duy nhất có thông lượng tối đa là 100Mbps . Các link kết nối vào web server hay mail server cũng có thông lượng tối đa là 100Mbps . Do đó , nếu 3hub và 2 server này gửi data ở mức tối đa là 100Mbps thì thông lượng tổng cộng là 500Mbps . Thông lượng tối đa này có thể đạt được giữa 11 end system .

Các host và server có thể send đến host và server khác .

**P31:** Giả sử tất cả các switch trong Figure 5.26 được thay bởi hub. Tất cả các link là 100 Mbps. Trả lời các câu hỏi trong P29.

**Trả lời:**

Tất cả các thiết bị đầu cuối đều nằm trong cùng 1 miền xung đột. Trong trường hợp này, tổng tối đa thông lượng 100 Mbps của tập hợp này là có thể xảy ra giữa 11 thiết bị đầu cuối. (dịch xong đêk hiểu gì )

**P32 :** Xét hoạt động học về switch (nghe ngu quá =))) ở Figure 5.24. Giả sử là (i) B gửi 1 frame tới E, (ii) E gửi trả lại 1 frame tới B, (iii) A gửi 1 frame tới B, (iv) B gửi trả lại 1 frame tới A. Bảng switch ban đầu trống không. Cho biết trạng thái của bảng switch trước và sau mỗi sự kiện. Với mỗi sự kiện, cho biết đường đi (s) mà mỗi frame sẽ được forward đi, và giải thích ngắn gọn câu trả lời.

**Trả lời :**

Hành động	Trạng thái bảng Switch	Đường đi mà packet sẽ được forward theo	Giải thích
B gửi frame tới E	Switch biết interface tương ứng với địa chỉ MAC của B	A,C,D,E và F	Từ lúc bảng switch trống, switch không biết interface tương ứng với địa chỉ MAC của E
E gửi lại 1 frame tới B	Switch biết interface tương ứng với địa chỉ MAC của E	B	Từ lúc switch đã biết interface tương ứng với địa chỉ MAC của B
A gửi 1 frame tới B	Switch biết interface tương ứng với địa chỉ MAC của A	B	Từ lúc switch đã biết interface tương ứng với địa chỉ MAC của B
B gửi lại 1 frame tới A	Trạng thái của bảng Switch giữ nguyên	A	Từ lúc switch đã biết interface tương ứng với địa chỉ MAC của A

**P33:** Trong vấn đề này, chúng ta khám phá việc sử dụng các gói nhỏ cho ứng dụng Voice-over IP. Một trong những hạn chế của gói dữ liệu nhỏ là một phần lớn băng thông liên kết tiêu thụ bởi các byte hao phí (overhead bytes). Giả sử gói bao gồm P bytes và 5 bytes tiêu đề.

- Xét việc gửi một nguồn âm thanh số mã hoá trực tiếp. Giả sử nguồn được mã hoá với tần số 128kbps. Giả sử mỗi gói tin đều đầy trước khi nguồn gửi gói tin vào mạng. Thời gian cần thiết để làm đầy 1 gói tin là packetization delay. Với điều kiện của L, xác định packetization delay trong miliseconds.
- Packetization delays lớn hơn 20 ms có thể gây ra phản hồi khó chịu và đáng chú ý. Xác định Packetization delays với  $L = 1,500$  bytes ( gần tương ứng với kích cỡ tối đa của 1 gói tin Ethernet ) và với  $L = 50$  ( ứng với 1 gói tin ATM).
- Tính độ trễ store-and-forward tại 1 chuyển đổi đơn, với tần số của liên kết là  $R = 622$  Mbps, với  $L = 1,500$  bytes và với  $L = 50$  bytes.
- Nhận xét về lợi thế của việc dùng gói tin nhỏ.



**Solution:**

Thời gian cần để làm đầy L-8 bits là :

$$\frac{L*8}{128*10^3} \text{ sec} = \frac{L}{16} \text{ msec}$$

B Với L=1500 Packetization delays là:

$$\frac{1500}{16} \text{ msec} = 93.75 \text{ msec}$$

Với L=50 Packetization delays là :

$$\frac{50}{16} \text{ msec} = 3.125 \text{ msec}$$

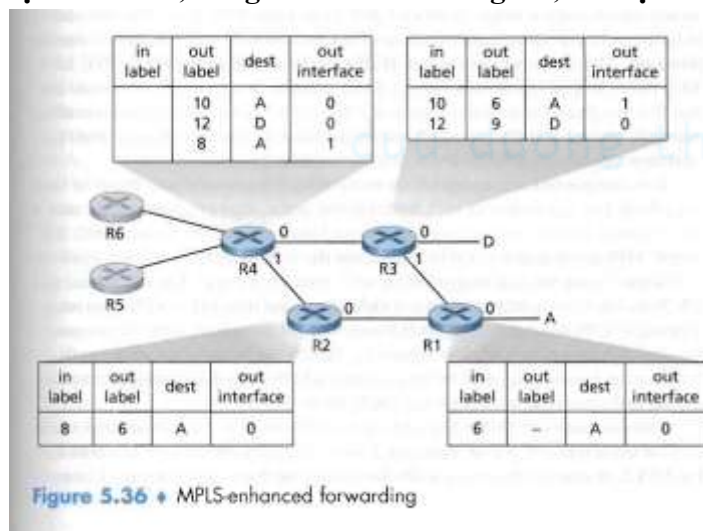
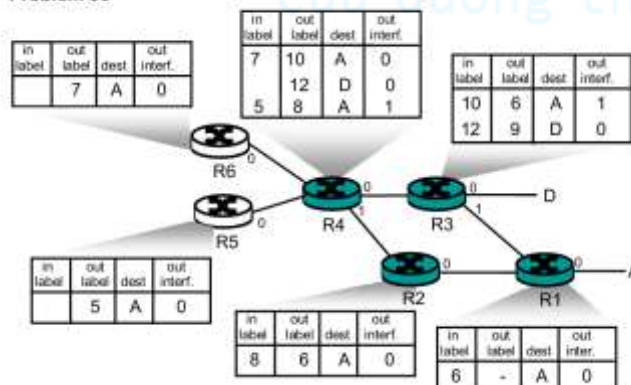
C Store-and-forward delay =  $\frac{L*8+40}{R}$

Với L=1500, độ trễ xấp xỉ 19.4  $\mu\text{sec}$

Với L=50, độ trễ nhỏ hơn 1  $\mu\text{sec}$

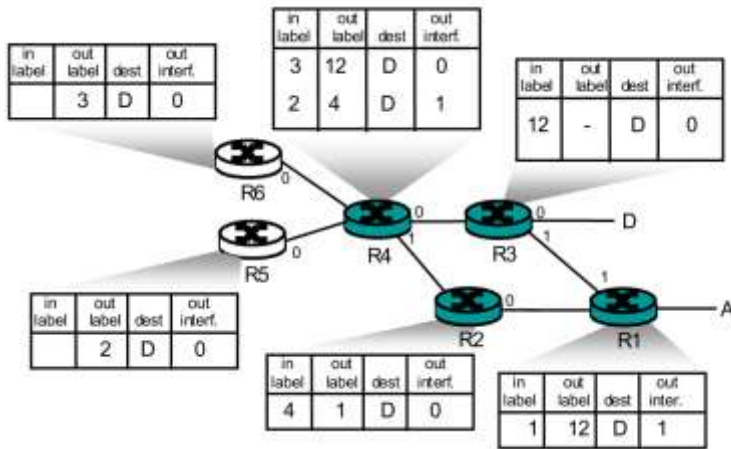
D Store-and-forward delay nhỏ trong trường hợp là tốc độ kết nối thông thường. Nhưng, Packetization delays quá lớn đối với các ứng dụng đối thoại thực.

**P35, Xem xét ạng MPLS được hiện thị trong hình 5.36 và giả sử rằng router R5 và R6 được MPLS cho phép. Giả sử chúng ta muốn thực hiện kỹ thuật vận chuyển (traffic engineering) do đó các packet từ R6 đến A được chuyển qua A qua R6-R4-R3-R1, từ R5 đến A được chuyển qua R5-R4-R2-R1. Hiện thị bảng MPLS tại R5 và R6, cũng như sửa đổi bảng R4, để mọi thứ xảy ra.**

**Solution:****Problem 35**

**P36, Xem xét lại tình huống giống như vấn đề trước, nhưng giả sử gói packet từ R6 đến D được chuyển qua R6-R4-R3, trong khi gói packet từ R5 đến D được chuyển qua R4-R2-R1-R3. Vẽ bảng MPLS của tất cả các router trong tình huống này.**

**Solution**



**P37, Trong vấn đề này, bạn sẽ cần sử dụng rất nhiều kiến thức mà bạn đã học về giao thức Internet. Giả sử bạn đi vào phòng, kết nối Internet, và muốn download 1 trang web. Những giao thức nào sẽ được sử dụng kể từ lúc bật máy tính đến khi nhận trang web. Giả sử không có trong DNS của bạn hoặc bộ nhớ của browser khi bật máy tính. (Gợi ý: Các bước bao gồm việc sử dụng các giao thức Ethernet, DHCP, ARP, DNS, TCP và HTTP). Chỉ ra rõ ràng trong các bước làm thế nào bạn có được IP và MAC của 1 router gateway.**

**Solution:**

Máy tính đầu tiên sử dụng DHCP để có được 1 địa chỉ IP. Máy tính của bạn đầu tiên sẽ tạo ra 1 IP datagram đặc biệt đến 255.255.255.255 ở bước phát hiện máy chủ DHCP và đặt nó trong 1 frame Ethernet và phát nó trong Ethernet. Sau đó, theo các bước trong giao thức DHCP, máy tính của bạn có thể lấy được 1 địa chỉ IP trong 1 khoảng thời gian.

Máy chủ DHCP trên Ethernet cũng cung cấp cho máy tính của bạn 1 danh sách các địa chỉ IP của first-hop router, subnetmask của subnet nơi bạn dùng máy và địa chỉ của DNS server địa phương (nếu có)

Kể từ khi bộ nhớ ARP của máy bạn rỗng, máy tính sử dụng giao thức ARP để lấy địa chỉ MAC của router first-hop và local DNS server.

Máy tính của bạn đầu tiên sẽ lấy địa chỉ IP của trang web bạn muốn tải về. Nếu local DNS server không có địa chỉ IP thì máy tính sẽ dùng giao thức DNS để tìm địa chỉ IP của trang web.

Khi máy tính đã có được địa chỉ IP của trang web, nó sẽ gửi các HTTP request thông qua các router first-hop nếu trang web không có trong một local web server. Tin nhắn HTTP request sẽ được phân đoạn và đóng gói thành các gói TCP packets, sau đó đóng gói thành các gói IP packets và cuối cùng là đóng gói vào Ethernet frame. Máy tính của bạn sẽ gửi Ethernet frame đến router first-hop. Khi router nhận được frames, nó đi vào lớp IP, kiểm tra bảng định tuyến của nó, sau đó gửi các gói tin đến giao diện bên phải trong số tất cả các giao diện của nó. Sau đó, các gói tin IP sẽ được định tuyến thông qua Internet cho đến khi chúng vươn tới được Web server.

Máy chủ lưu giữ trang web sẽ gửi lại trang web để máy tính của bạn thông qua HTTP response messages. Những tin nhắn sẽ được đóng gói trong các TCP packets và sau đó được đóng gói thành các IP packets. Những IP packets sẽ theo các tuyến IP và cuối cùng đến first-hop router, sau đó router sẽ chuyển tiếp những gói tin IP vào máy tính của bạn bằng cách đóng gói chúng vào Ethernet frames.

cuu duong than cong . com

cuu duong than cong . com