

Môn Mạng Máy Tính

(4 câu trong 60 phút, không sử dụng tài liệu)

1. Cho biết sự khác biệt cùng với ưu/nhược điểm của các mạng chuyển mạch và mạng chuyển mạch gói.

- Giống nhau: đều là phương pháp truyền thông tin dữ liệu qua mạng.
- Khác nhau:

Mạng chuyển mạch	Mạng chuyển gói
<p>-- Hoạt động: Để thực hiện được việc liên kết giữa 2 nút, một đường nối giữa điểm nút này và điểm nút kia được thiết lập trong mạng thể hiện dưới dạng cuộc gọi thông qua các thiết bị chuyển mạch.</p> <p>Với mô hình chuyển mạch mọi đường đều có thể từ một đường bất kỳ khác, thông qua những đường nối và các thiết bị chuyên dụng người ta có thể liên kết một đường tạm thời từ nơi gửi tới nơi nhận từ đường nối vật lý. Đường nối trên duy trì trong suốt phiên làm việc và chỉ giải phóng khi phiên làm việc kết thúc.</p> <p>-- Tài nguyên mạng (như băng thông) được chia thành các “Mảnh”:</p> <ul style="list-style-type: none"> + Mỗi mảnh được cấp phát cho một cuộc gọi. + Mảnh được cấp phát sẽ rảnh rỗi nếu không được sử dụng trong cuộc gọi (Không chia sẻ). + Phân chia băng thông (bandwidth) thành các “Mảnh”. <p>--Có 2 loại chuyển mạch:</p> <ul style="list-style-type: none"> + Chuyển mạch tương tự (Analog). 	<p>--Hoạt động: Khi một trạm trên mạng cần gửi dữ liệu, nó cần phải đóng dữ liệu thành từng gói tin, các gói tin đó được truyền đi trên mạng từ nút này đến nút khác tới khi đến đích.</p> <p>Do việc sử dụng kỹ thuật trên nên khi một trạm không gửi tin thì mọi tài nguyên của mạng sẽ dành cho trạm khác. Do vậy mà mạng tiết kiệm được các tài nguyên và có thể sử dụng chúng một cách tốt nhất.</p> <p>--Dữ liệu được chia thành các gói “chunck”:</p> <ul style="list-style-type: none"> + Các gói chia sẻ các tài nguyên mạng. + Mỗi gói dùng hết băng thông liên kết. + Các tài nguyên được dùng theo nhu cầu. <p>- Có 2 phương pháp chuyển mạng gói:</p> <ul style="list-style-type: none"> + Chuyển mạng gói theo sơ đồ rời rạc.

<p>+ Chuyển mạch số (Digital).</p> <p>--Ưu điểm:</p> <p>+ Hiệu suất đảm bảo.</p> <p>--Nhược điểm:</p> <p>+ Tài nguyên độc quyền, không share.</p> <p>+ Yêu cầu thiết lập cuộc gọi.</p> <p>--Phương pháp hỗ trợ: Đảm bảo băng thông cần thiết cho các ứng dụng audio/video.</p>	<p>+ Chuyển mạng gói theo đường đi xác định.</p> <p>--Ưu điểm:</p> <p>+ Cho phép nhiều người dùng chung một mạng.</p> <p>+ Tuyệt vời với việc bùng nổ dữ liệu (Chia sẻ tài nguyên, đơn giản nhất, không cần thiết lập cuộc gọi.). Thích hợp cho lưu thông dữ liệu lớn.</p> <p>--Nhược điểm:</p> <p>+ Tổng số yêu cầu tài nguyên có thể vượt quá lượng có sẵn. → Tắc nghẽn: hàng đợi đóng gói chờ truyền.</p> <p>+ Gói tin có thể bị mất mát do quá trình truyền.</p> <p>--Cần khắc phục tình trạng tắc nghẽn (gói tin bị trễ và mất mát): cần các giao thức cần thiết cho truyền dữ liệu tin cậy và giải quyết tắc nghẽn.</p>
--	--

→ Do tính chất khác nhau của nhu cầu thực tế mà người dùng có thể sử dụng mạng chuyển gói hay chuyển mạch.

2. Phân biệt dịch vụ truyền thông có liên kết (*connection oriented*) và dịch vụ truyền thông không liên kết (*connectionless*).

- Giống nhau: Đều là dịch vụ truyền thông dữ liệu.
- Khác:

Connection oriented	Connectionless
<p>--Mục tiêu: truyền thông dữ liệu giữa các host.</p> <p>-- Giao thức truyền: sử dụng giao thức TCP (Transmission Control Protocol).</p> <p>+ TCP service [RFC 793].</p>	<p>--Mục tiêu: Dữ liệu truyền giữa các hệ thống đầu cuối. -</p> <p>--Giao thức truyền: UDP (User Datagram Protocol).</p> <p>+ UDP [RFC 768].</p> <p>+ Không kết nối.</p>

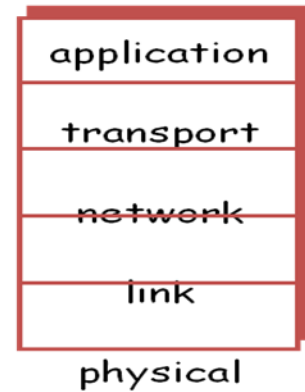
<ul style="list-style-type: none"> + Là giao thức tin cậy, an toàn dữ liệu theo dòng byte (nếu mất mát sẽ có thông báo và truyền lại). + Điều khiển luồng (bên gửi không lấn áp bên nhận). + Giải quyết tắc nghẽn (Bên gửi giảm tốc độ truyền khi mạng nghẽn). <p>-- Các ứng dụng dùng TCP:</p> <p>HTTP (Web), FTP (File Transfer Protocol), TelNet (Remote Login), SMTP (email).</p>	<ul style="list-style-type: none"> + Truyền thông tin cậy. + Không điều khiển luồng dữ liệu. + Không giải quyết tắc nghẽn. <p>--Các ứng dụng UDP: Stream media, hội thảo từ xa, DNS, Điện thoại internet.</p>
--	--

3. Cho biết vì sao cần phân lớp, tầng trong mô hình truyền thông. Giới thiệu mô hình TCP/IP.

- Phân lớp, tầng trong mô hình truyền thông nhằm xử lý tốt hơn với các hệ thống phức tạp.
 - Cấu trúc rõ nhằm xác định quan hệ giữa các mảnh của hệ thống đó.
 - Mô-đun hóa làm cho việc bảo trì và cập nhật hệ thống trở nên dễ dàng hơn.
 - Thay đổi việc thực hiện các dịch vụ của lớp là trong suốt với phần còn lại của hệ thống.

Ví dụ: Thay đổi thủ tục kiểm tra ở cổng không ảnh hưởng đến các thành phần còn lại của hệ thống.
- Trong thực tế chúng ta gặp khá nhiều ví dụ về việc phân tầng. Giả sử người A viết thư gửi cho người B. Sau khi viết thư xong, A cho thư vào phong bì, dán kín, ghi địa chỉ của B, dán tem và cho phong thư vào hộp thư ở bưu điện. Giữa A và B đơn vị dữ liệu trao đổi là các bức thư. Bức thư có thể xem là dữ liệu thực sự, trong khi phong bì có thể xem là một loại tiêu đề chứa các thông tin điều khiển. Hệ thống bưu điện (Bao gồm nhiều bưu cục – là một trong các trạm mà bức thư sẽ đi qua) chịu trách nhiệm chuyển bức thư tới địa chỉ của B. Ví dụ này tầng dưới (hệ thống bưu điện) sẽ cung cấp dịch vụ chuyển thư cho tầng trên (A và B). A và B chỉ qua tâm đến nội dung bức thư, khuôn dạng thư, ngôn ngữ, chữ viết trong thư... mà không cần quan tâm làm thế nào để thư có thể chuyển tới B. Đây chính là ưu điểm của việc phân tầng, Tầng trên sử dụng các dịch vụ của tầng dưới mà không cần quan tâm đến cách thức thực hiện của tầng dưới như thế nào.
- Giới thiệu mô hình TCP/IP:

- ❑ **application:** cung cấp các dịch vụ cho các ứng dụng mạng
 - ❖ FTP, SMTP, HTTP
- ❑ **transport:** xử lý dữ liệu truyền
 - ❖ TCP, UDP
- ❑ **network:** dẫn đường cho các gói tin từ nguồn đến đích
 - ❖ IP, các giao thức dẫn đường
- ❑ **link:** dữ liệu truyền giữa các lớp lân cận
 - ❖ PPP, Ethernet
- ❑ **physical:** các bit “trên đường dây”



4. Phân tích sự khác biệt giữa kiến trúc client – server và Peer-to-Peer của lớp ứng dụng mạng.

- Giống nhau: Đều là quá trình trao đổi thông tin giữa 2 máy tính trên mạng.
- Khác:

Client – Server (Khách hàng – người phục vụ)	Peer-to-Peer (P2P) (Mô hình Ngang hàng)
<p>--Trong mô hình này 1 máy tính sẽ đóng vai trò là client và máy tính kia đóng vai trò là server.</p> <p>-- Hoạt động:</p> <p>+ Server:</p> <ul style="list-style-type: none"> . Host luôn hoạt động. . Địa chỉ IP cố định . Nhóm các server để chia sẻ công việc <p>+ Client:</p> <ul style="list-style-type: none"> . Truyền thông với server. . Có thể kết nối không liên tục. . Có thể có địa chỉ IP thay đổi. 	<p>-- Trong mô hình này một máy tính vừa đóng vai trò là client, vừa đóng vai trò là server.</p> <p>-- Hoạt động:</p> <p>+ Không có server, luôn hoạt động.</p> <p>+ Truyền thông với hệ thống đầu cuối bất kỳ.</p> <p>+ Các điểm kết nối không liên tục và thay đổi địa chỉ IP.</p> <p>Ví dụ: Gnutella, KazaA.</p>

. Không truyền thông trực tiếp với client khác.	
-- Độ linh hoạt không cao nhưng dễ quản lý.	--Độ linh hoạt cao nhưng khó quản lý.

5. So sánh HTTP không bền vững và giao thức HTTP bền vững? (HTTP: HyperText Transfer Protocol).

- Giống: Đều sử dụng TCP, HTTP là không có trạng thái.
- Khác:

HTTP - nonpersistent	HTTP – persistent
<p>--HTTP 1.0 (Nonpersistent HTTP)</p> <p>-- Sau khi gửi đi một đối tượng thì server sẽ đóng kết nối TCP lại. Kết nối không được lấy để sử dụng đối tượng khác. Như vậy mỗi kết nối TCP chuyển duy nhất một thông điệp yêu cầu và 1 thông điệp trả lời.</p> <p>Ví dụ: Để gửi 1 đối tượng trang Web gồm 1 trang HTML và 10 file ảnh JPEG. Thì sẽ có tới 11 kết nối TCP được thiết lập.</p> <p>--Vấn đề: Khi liên kết mới tạo ra phía client và server phải tạo ra vùng đệm TCP (buffer) cũng như lưu giữ các biến TCP → Tạo gánh nặng cho server khi có nhiều client yêu cầu cùng lúc.</p> <p>- Yêu cầu 2 RTT mỗi đối tượng.</p> <p>- Hệ điều hành liên quan đến mỗi kết nối TCP.</p> <p>- Các trình duyệt thường mở // các kết nối TCP để đem về các tham chiếu đến các đối tượng.</p>	<p>--HTTP1.1 (Persistent HTTP).</p> <p>- Server không đóng kết nối TCP sau khi gửi thông điệp trả lời. Các thông điệp yêu cầu và trả lời sau đó (giữa cùng 1 client và server) được gửi liên tiếp nhau thông qua cùng 1 kết nối. Thông thường HTTP sẽ đóng liên kết khi liên kết không sử dụng trong một thời gian nào đó.</p> <p>Trong ví dụ bên: Toàn bộ đối tượng trang Web (gồm 1 trang HTML và 10 file ảnh JPEG) sẽ được truyền nối tiếp nhau qua 1 kết nối TCP.</p> <p>--HTTP client gửi yêu cầu khi nó nhận được 1 tham chiếu (1 siêu liên kết, hay 1 tham chiếu đến file ảnh) vì vậy client có thể gửi các yêu cầu liên tiếp. Khi server nhận được yêu cầu, nó sẽ gửi các đối tượng nối tiếp nhau.</p> <p>- Server bỏ kết nối sau khi mở để gửi đáp ứng leaves.</p> <p>- Các thông điệp HTTP của tiến trình con cùng mô hình client/server gửi thông qua kết nối mở.</p>

6. Phân tích giao thức Rdt2.0 (Principles of Reliable Data Transfer: Nguyên lý truyền dữ liệu tin cậy).

Rdt2.0 kênh với các lỗi (Channel with bit errors)

- Hãy xem xét, Khi chúng ta nói chuyện qua điện thoại, phía bên kia nói mà ta nghe rõ thì có thể trả lời là “OK”, nhưng nếu ta không nghe rõ người kia nói gì ta sẽ yêu cầu “ Xin Nhắc Lại”. Những thông điệp này nhằm thông báo cho bên gửi biết rằng dữ liệu nào nhận đúng, dữ liệu nào bị lỗi và yêu cầu truyền lại dữ liệu bị lỗi. Trong mạng máy tính, giao thức truyền dữ liệu tin cậy như vậy gọi là giao thức ARQ (Automatic Repeat Request).
- ARQ có 3 khả năng để xử lý trong trường hợp có lỗi bit: Phát hiện lỗi (error detection), Phản hồi từ phía nhận (receiver feedback), Truyền lại (Retransmission).
- Kênh ưu tiên có thể có thể bật lên một số bit trong gói (checksum để kiểm tra lỗi).
- Trong giao thức Rdt2.0, Phía bên gửi có 2 trạng thái. Ở trạng thái thứ nhất, Phía bên gửi đợi dữ liệu từ tầng trên. Trong trạng thái thứ 2, phía bên gửi phản hồi ACK (Acknowledgement) hoặc NAK (Negative Acknowledgement) từ phía nhận.
 - + Nếu nhận được ACK phía bên gửi biết được gói dữ liệu chuyển đến đích an toàn, vì vậy nó trở về trạng thái đợi dữ liệu từ tầng trên để chuyển tiếp.
 - + Nếu nhận được NAK phía gửi, gửi lại gói dữ liệu rồi quay lại trạng thái chờ phản hồi (ACK hoặc NAK), nó không thể nhận thêm dữ liệu từ tầng trên đưa xuống. Nó chỉ chấp nhận dữ liệu khi nhận được ACK và chuyển trạng thái.
 - + Phía gửi không gửi thêm dữ liệu cho đến khi biết chắc rằng phía nhận đã nhận đúng gói dữ liệu đã gửi. Giao thức Rdt2.0 do có những hành vi như vậy nên còn có tên gọi là Stop-and-Wait (dừng và chờ: tức là người gửi một gói sau đó dừng lại và chờ phản hồi từ người nhận).
- Rdt2.0 có một lỗi hồng nghiêm trọng là: Điều gì sẽ xảy ra khi ACK hoặc NAK bị hỏng.
 - + Người gửi không biết điều gì đã xảy ra tại bên nhận.
 - + Không thể đơn phương truyền lại vì sẽ có khả năng trùng lặp.
 - ➔ Cách khắc phục: Quản lý trùng lặp:
 - . Người gửi truyền lại gói hiện tại nếu ACK/NAK hỏng.
 - . Người gửi thêm số thứ tự vào mỗi gói.
 - . Người nhận hủy, không nhận gói trùng lặp.

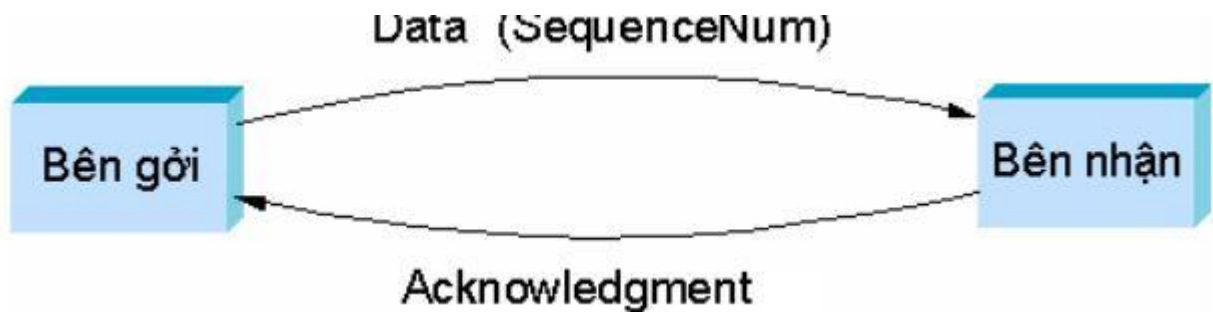
7. *Phân tích giao thức Rdt3.0 (Rrinciples of Reliable Data Tranfer)?*

Rdt3.0 Kênh với lỗi và mất mát (Channel with errors and loss).

- Dữ liệu trên kênh truyền không những bị lỗi mà còn có thể bị mất mát. Đây là tình huống không phải không phổ biến trong máy tính ngày nay. Bởi vì cơ chế checksum , đánh số thứ tự , các ACK, các việc truyền lại đưa ra ở các phiên bản Rdt trước sẽ hỗ trợ nhưng không đủ. Do đó, cần có một cơ chế mới để giải quyết vấn đề lỗi và mất mát trên kênh truyền.
 - Phía người gửi không biết gói dữ liệu bị mất, gói ACK bị mất hay chỉ đơn giản là chúng bị trễ... Lúc này, truyền lại là một giải pháp “Vạn Năng”. Trong tất cả các trường hợp hành động của nó là giống nhau: Truyền lại. Để thực hiện cơ chế truyền lại theo thời gian, một bộ định thời đếm ngược (coutdown timer) được sử dụng để nhắc phía người gửi, thời gian đợi đã hết (khoảng thời gian chấp nhận được đã hết). Do đó phía người gửi phải có khả năng:
 - + Khởi tạo timer mỗi khi gửi gói dữ liệu.
 - + Phản ứng với ngắt của timer (Đưa ra những hành động thích hợp).
 - + Dừng timer.
 - Nếu gói (hoặc ACK) chỉ trễ (không mất):
 - + Truyền lại sẽ gây trùng nhưng dùng số thứ tự sẽ giải quyết được.
 - + Người nhận phải xác định số thứ tự của gói vừa gửi ACK để kiểm soát việc nhận trùng.
- Rdt3.0 là một giao thức truyền dữ liệu tin cậy thực sự hoạt động được.

8. *Giải thích nhiệm vụ của trường: số thứ tự (Sequence numbers) và số tiếp nhận (Acknowledgement numbers) trong gói tin TCP (Transmission Control Protocol)?*

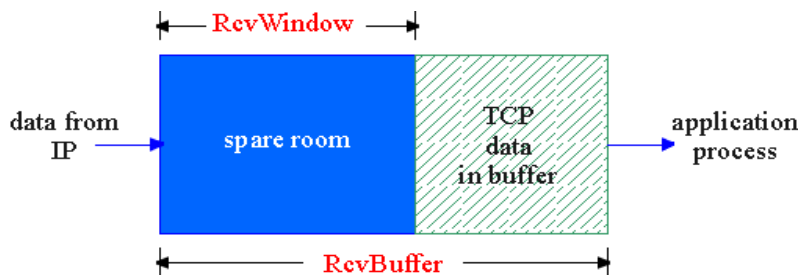
- TCP là một giao thức hướng byte, nên mỗi byte của dữ liệu sẽ có một số thứ tự, trường Sequence number chứa số thứ tự của byte đầu tiên của một dãy các bytes chứa trong một segment. Trường ACK number chứa số thứ tự của byte kế tiếp được chờ đợi từ phía bên kia để thông báo tiến độ nhận các bytes trong luồng dữ liệu và khả năng tiếp nhận chúng.
- Để đơn giản hóa vấn đề , ta bỏ qua sự thật là dữ liệu có thể truyền theo 2 chiều . Ở đây , ta đưa ra một sơ đồ như sau. Bên gửi sẽ gửi một segment dữ liệu, byte đầu tiên trong segment đó sẽ có số thứ tự là Sequence number, và phía bên nhận sẽ thông báo cho bên gửi trường ACK number về tiến độ nhận segment dữ liệu bằng ACK tích lũy.



Khi bên nhận nhận được n bytes trong một segment bắt đầu từ byte thứ Sequence number, nó sẽ báo nhận tốt n bytes này và chờ nhận tiếp từ byte thứ ACK number ($\text{ACK number} = \text{Sequence number} + n$).

9. Giải thích phương thức điều khiển phân luồng (Flow Control) trong giao thức TCP.

- Phương thức điều khiển phân luồng là phương thức mà bên gửi không là tràn bộ đệm vì truyền quá nhiều và quá nhanh, bên nhận cũng đảm bảo không tràn bộ đệm nhờ:
 - + Bên nhận của kết nối TCP có 1 bộ đệm nhận.
 - + Dịch vụ trùng tốc độ: so trùng tốc độ gửi và tốc độ nhận.
 - + Tiến trình ứng dụng có thể chậm lại lúc đọc bộ đệm.
- Do kích thước cửa sổ là động nên trong mỗi khung báo nhận của mình, bên nhận thông báo về khoảng dự trữ của nó (lượng buffer còn trống), đó chính là trị RcvWindow.



- Lần sau, bên gửi sẽ không được gửi lượng byte vượt quá RcvWindow (để đảm bảo bộ đệm nhận không bị tràn.).
- Ví dụ: Bên gửi muốn gửi 5 kb và bộ đệm của bên nhận là 4 kb (còn rỗng).
 - + Bên gửi đặt số thứ tự Sequence number = 0, sau đó truyền 2 kb, buffer bên nhận còn lại 2 kb rỗng do đó nó báo nhận “ $\text{ACK} = 2048, \text{RcvWindow} = 2048$ ”.
 - + Bên gửi gửi tiếp 2 kb, khi đó buffer bên nhận đầy nó báo nhận “ $\text{ACK} = 4096, \text{RcvWindow} = 0$ ”, không còn buffer nhận, nên bên gửi tạm thời tắt nghẽn.
 - + Sau khi bên nhận xử lý xong 2 kb nó liền báo “ $\text{ACK} = 4096, \text{RcvWindow} = 2048$ ”. Lúc này bên gửi có thể gửi tối đa là 2 kb nhưng nó chỉ còn 1 kb để gửi.

10. Chức năng và nhiệm vụ của Congestion Window (CongWin) trong điều khiển tắc nghẽn của giao thức TCP ?

- Cửa sổ tắc nghẽn (congestion window) ký hiệu là CongWin biểu thị số lượng dữ liệu tối đa mà người gửi có thể gửi qua kết nối. Như vậy, khối lượng dữ liệu được gửi không được vượt quá CongWin và RcvWindow (Lượng Buffer còn trống của bên nhận).

$$\text{LastByteSend} - \text{LastByteAcked} \leq \{ \text{CongWin}, \text{RcvWindow} \}.$$

(LastByteSend – LastByteAcked: là lượng dữ liệu chưa được biên nhận mà người gửi gửi qua kết nối.)
Ngưỡng ký hiệu là Threshold sẽ ảnh hưởng tới quá trình tăng của CongWin.

- CongWin có nhiệm vụ là tăng nhanh theo cấp số lũy thừa khi chưa vượt ngưỡng. Khi xảy ra hiện tượng tắc nghẽn (CongWin \geq Threshold) thì CongWin tăng theo cấp tuyến tính đến khi xảy ra mất mát dữ liệu. Điều này dẫn đến sự kiện Timeout ở phía gửi. Lúc này ngưỡng (Threshold) = $\frac{1}{2}$ CongWin, CongWin = 1 MSS. Bên gửi sẽ tiếp tục tăng nhanh giá trị CongWin theo hàm mũ cho đến khi vượt ngưỡng...

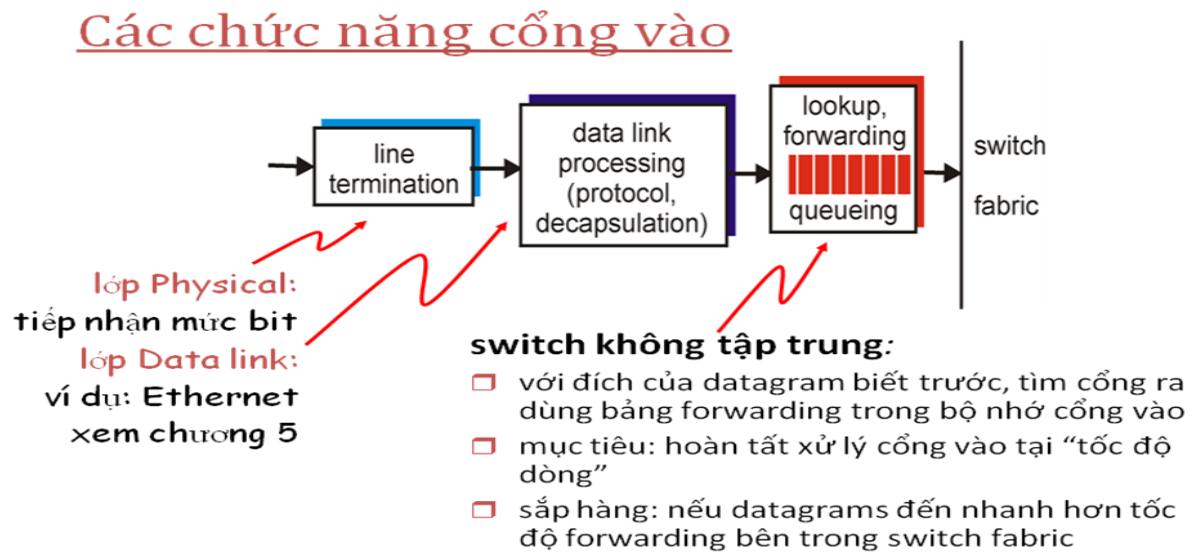
11. Cho biết sự khác biệt giữa 2 chức năng Forwarding và Routing của 1 Router trong lớp mạng.

Forwarding	Routing
<ul style="list-style-type: none"> - Chức năng: Di chuyển các gói từ đầu vào đến đầu ra thích hợp của Router - Nhiệm vụ: Tiến trình lập kế hoạch chuyển đi từ nguồn đến đích. - Thuộc “Transport Plane” 	<ul style="list-style-type: none"> - Chức năng: Xác định đường đi cho các gói từ nguồn đến đích bằng các giải thuật Routing (giải thuật tìm đường). - Nhiệm vụ: Tiến trình vận chuyển qua mỗi giao điểm. - Thuộc một phần của “control panel” của IP internet,

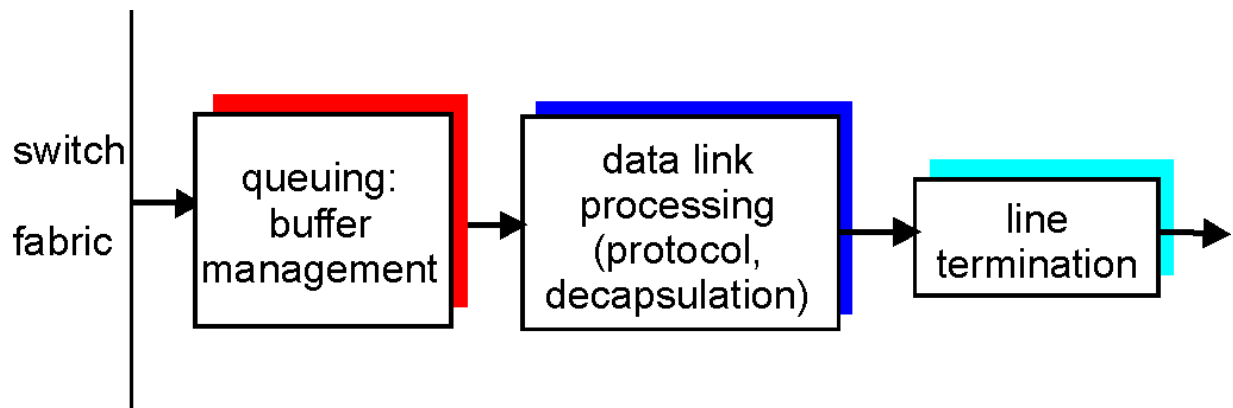
Forwarding và Routing hoạt động này được đồng nhất trong các IP router truyền thống, nhưng bây giờ chúng được tách biệt một cách luận lý như trong môi trường MPLS.

Việc tách biệt cho phép nhiều thành phần mạng hơn để thay thế các IP router để thực hiện các hoạt động control plane, cho phép việc định tuyến phân tán với hiệu suất được cải thiện. Thêm vào đó, việc tách biệt giữa các chức năng của routing và forwarding cho phép tách biệt hiệu suất giữa chúng, đồng thời cho phép chúng ta theo dõi rõ hơn sự cải tiến của các lý thuật ở cả kỹ thuật forwarding và các giải thuật định tuyến.

12. Vẽ và giải thích cổng vào và cổng ra của 1 Router ?

a. Cổng vào (Input Port):

Network Layer 4-22

b. Cổng ra (Output Port):

Quá trình xử lý ở cổng ra ở hình trên là lấy gói dữ liệu đã được lưu trữ trong bộ đệm (buffer) của cổng ra và truyền qua đường liên kết ra. Các chức năng xử lý giao thức liên kết dữ liệu là chức năng của tầng liên kết dữ liệu (data link processing) và tầng vật lý (line termination) để làm việc với đầu vào bên kia của đường truyền vật lý. Chức năng quản lý vùng đệm (buffer management) và hàng đợi được sử dụng khi tốc độ dữ liệu mà kết cấu chuyển chuyển tới nhanh hơn tốc độ gói đi của cổng ra.

The end.