**TCP Congestion  
Sự khác biệt giữa flow control và congestion control?**flowcontrol về cơ bản có nghĩa là TCP sẽ đảm bảo rằng sender không áp đảo reciver bằng cách gửi các gói nhanh hơn mức nó có thể xử lý. Ý tưởng là một node reciever sẽ gửi một số loại phản hồi đến node sender để cho nó biết về tình trạng hiện tại của nó. Và nó không giống với congestion flow. Mặc dù có một số chồng chéo giữa các cơ chế mà TCP sử dụng để cung cấp cả hai dịch vụ. chúng là các tính năng riêng biệt. congestion control là ngăn một node làm quá tải đường mạng (tức là các liên kết giữa hai nút), trong khi flow control kiểm soát về node reciever.  
**Dấu hiệu mất gói trong TCP?**nhận 3 gói ACK cùng lúc do retransmisson, timeout do retransmisson. **Cải tiến của TCP Reno so với Tahoe  
Tahoe:** Đây là thuật toán kiểm soát tắt nghẽn cơ bản và dễ hiểu nhất. Tahoe tăng congestion window (cwnd) theo hàm mũ trong giai đoạn Slow Start và tăng tuyến tính trong giai đoạn congestion avoidance. Tahoe dùng một timer canh giờ để xét xem liệu một gói tin có “timeout” hay chưa. Trong giai đoạn congestion advoidance, khi phát hiện tắt nghẽn (một gói tin bị time out) thì threshold sẽ bị giảm còn bằng ½ của cwnd lúc đó. Đồng thời cwnd sẽ bị giảm xuống còn 1 và bắt đầu lại từ giai đoạn slow start.  
khuyết điểm: Chính vì sử dụng timer để xét một gói tin bị timeout, nên vấn đề chính ở TCP Tahoe là thời gian chờ timeout để phát hiện gói tin bị mất là khá lâu. Các gói tin ACK không được gởi ngay sau mỗi gói tin mà đươc gởi cộng dồn. Vì TCP Tahoe sử dụng phương thức “go back N” nên mỗi khi có một gói tin bị mất thì đường truyền sẽ trống trong một khoảng thời gian. Gây lãng phí tài nguyên.Không sử dụng tối ưu băng thông.  
**Reno:** Tương tự như TCP Tahoe ở giai đoạn slow start nhưng có một số cải tiến. Giúp phát hiện gói tin bị mất sớm hơn trước khi một gói tin bị timeout. Tăng hiệu suất gởi nhận. Điểm khác biệt: Với mỗi gói tin được gởi thì các gói ACK trả về sẽ được trả ngay lập tức theo đúng thứ tự của gói tin đã nhận, chứ không cộng dồn các gói ACK như TCP Tahoe. TCP Reno có cài đặt thêm thuật toán “Fast-Retransmit” khi gặp trường hợp có 3 gói ACK bị ***lặp lại***. Reno sẽ giảm đặt giá trị của cwnd mới bằng ½ giá trị của cwnd hiện tại. Và đặt ngưỡng threshold mới bằng giá trị của cwnd mới. Đồng thời chuyển nhanh lại gói tin đã bị mất (Fast-Retransmit) và bước vào một pha gọi là Fast Recovery. Nếu sau pha này mà gói ACK của gói tin vừa được gởi lại bị time out thêm lần nữa thì cwnd sẽ bị hạ xuống thành 1 và trở lại giai đoạn Slow Start giống như Tahoe. Tahoe sử dụng timout để phát hiện tắt nghẽn còn Reno sử dụng cả timer và thuật toán Fast-Retransmit để nhận biết tắt nghẽn. Tahoe hạ cwnd xuống còn 1 sau khi mất gói tin, còn Reno hạ cwnd xuống ½ của cwnd hiện tại khi mất gói tin (khi phát hiện 3 duplicate ACK).  
Ưu điểm: Quá trình phục hồi truyền dữ liệu nhanh hơn so với Tahoe (Fast Recovery) vì thường thì thời gian chờ để nhận 3 gói ACK trùng lặp sẽ nhanh hơn là chờ timeout rồi mới send lại gói tin bị mất. Hiệu quả hơn Tahoe  
Khuyết điểm: Nếu cwnd size của Reno quá nhỏ (nhỏ hơn 4 gói) thì có thể sẽ không nhận đủ 3 gói ACK để chạy thuật toán. Mỗi lần Reno chỉ có thể phát hiện được một gói tin bị mất và không thể nhận biết được trường hợp có nhiều gói tin bị mất  
**Sự khác nhau giữa EGP và IGP**Giao thức IGP thức được sử dụng bên trong một AS cho việc chia sẻ thông tin định tuyến giữa các router. Phù hợp cho các mạng doanh nghiệp lớn. Trong khi Giao thức EGP được sử dụng cho việc giao tiếp chia sẻ thông tin giữa các AS khác nhau.  
**Sự khác biệt giữa Distance Vector và Link State**Distance Vector sử dụng thuật toán Bellman-Ford để thực hiện định tuyến. Trong khi LinkState sử dụng Dijkstra để thực hiện định tuyến. Trong định tuyến Distance Vector, các router nhận thông tin topology từ neighbour. Ngược lại, trong LinkState, các router nhận được thông tin đầy đủ về topology. Định tuyến Distance Vector tính toán tuyến đường tốt nhất dựa trên khoảng cách (số bước nhảy ít nhất). Ngược lại, định tuyến LinkState tính toán tuyến đường tốt nhất trên cơ sở chi phí ít nhất. Định tuyến LinkState chỉ cập nhật trạng thái liên kết, trong khi Định tuyến Distance Vector cập nhật toàn bộ bảng định tuyến. Distance Vector hội tụ chậm hơn LinkState. Distance Vector thiết lập bằng RIP và IGRP, LinkState thiết lập bằng OSPF

**Mô hình hoạt động của chuyển gói tin.**

Đầu tiên, dữ liệu được xử lý bởi tầng application. Tầng này có nhiệm vụ tổ chức dữ liệu theo khuôn dạng và trật tự nhất định để tầng application ở máy B có thể hiểu được. Điều này giống như khi bạn viết một chương trình thì các câu lệnh phải tuân theo thứ tự và cú pháp nhất định thì chương trình mới chạy được. Tầng application gửi dữ liệu xuống tầng dưới theo dòng byte nối byte. Cùng với dữ liệu, tầng application cũng gửi xuống các thông tin điều khiển khác giúp xác định địa chỉ đến, đi của dữ liệu.  
Khi xuống tới tầng TCP, dòng dữ liệu sẽ được đóng thành các gói có kích thước không nhất thiết bằng nhau nhưng phải nhỏ hơn 64 KB. Cấu trúc của gói dữ liệu TCP gồm một phần header chứa thông tin điều khiển và sau đó là dữ liệu. Sau khi đóng gói xong ở tầng TCP, dữ liệu được chuyển xuống cho tầng IP.  
Gói dữ liệu xuống tới tầng IP sẽ tiếp tục bị đóng gói lại thành các gói dữ liệu IP nhỏ hơn sao cho có kích thước phù hợp với mạng chuyển mạch gói mà nó dùng để truyền dữ liệu. Trong khi đóng gói, IP cũng chèn thêm phần header của nó vào gói dữ liệu rồi chuyển xuống cho tầng Datalink/Physical.  
Khi các gói dữ liệu IP tới tầng Datalink sẽ được gắn thêm một header khác và chuyển tới tầng physical đi vào mạng. Gói dữ liệu lúc này gọi là frame. Kích thước của một frame hoàn toàn phụ thuộc vào mạng mà máy A kết nối.  
Trong khi chu du trên mạng Internet, frame được các router chỉ dẫn để có thể tới đúng đích cần tới. Router thực ra là một module chỉ có hai tầng là Network và Datalink/Physical. Các frame tới router sẽ được tầng Datalink/Physical lọc bỏ header mà tầng này thêm vào và chuyển lên tầng Network (IP). Tầng IP dựa vào các thông tin điều khiển trong header mà nó thêm vào để quyết định đường đi tiếp theo cho gói IP. Sau đó gói IP này lại được chuyển xuống tầng Datalink/Physical để đi vào mạng. Quá trình cứ thế tiếp tục cho đến khi dữ liệu tới đích là máy B.

**TCP/IP hoạt động như thế nào?**

Trong ***giao thức TCP/IP***, **IP** có vai trò quan trọng. **IP** cho phép máy tính chuyển tiếp gói tin tới một máy tính khác. Thông qua một hoặc nhiều khoảng (chuyển tiếp) gần với người nhận gói tin. **TCP** sẽ giúp kiểm tra các gói dữ liệu xem có lỗi không? Sau đó gửi yêu cầu truyền lại nếu có lỗi được tìm thấy.

**Ưu điểm của TCP/IP là gì?**

Ưu điểm thứ nhất của **TCP/IP** chính là không chịu sự kiểm soát của bất kỳ tổ chức nào. Vì vậy, bạn có thể tự do trong việc sử dụng. Thứ hai, **TCP/IP** có khả năng tương thích cao với tất cả các hệ điều hành, phần cứng máy tính và mạng. Vì vậy, giao thức này hoạt động hiệu quả với nhiều hệ thống khác nhau.

**FTP**: [**FTP**](https://wiki.matbao.net/kb/ftp-la-gi-cach-chinh-sua-file-tren-hosting-cuc-nhanh-chong/) là phương thức trao đổi file được sử dụng giữa hai hoặc nhiều máy tính thông qua Internet. Nhờ FTP, các máy tính có thể gửi và nhận dữ liệu đến nhau một các trực tiếp.

***Tầng 1: Tầng vật lý (Physical)***: chịu trách nhiệm truyền dữ liệu giữa hai thiết bị trong cùng một mạng. Tại đây, các gói dữ liệu được đóng vào khung (gọi là **Frame**) và được định tuyến đi đến đích đã được chỉ định ban đầu.

***Tầng 2: Tầng mạng (Network)***: đảm nhận việc truyền tải dữ liệu một cách hợp lý. Các giao thức của tầng này bao gồm **IP** (**Internet Protocol**), **ICMP** (**Internet Control Message Protocol**), **IGMP** (**Internet Group Message Protocol**).

***Tầng 3: Tầng giao vận (Transport):*** **TCP** sẽ đảm bảo chất lượng truyền gửi gói tin, tuy nhiên lại mất thời gian khá lâu để thực hiện các thủ tục kiếm soát dữ liệu. Ngược lại, **UDP** lại cho tốc độ truyền tải nhanh nhưng lại không đảm bảo được chất lượng dữ liệu. Ở tầng này, **TCP** và **UDP** sẽ hỗ trợ nhau phân luồng dữ liệu.

***Tầng 4: Tầng ứng dụng (Application).*** đảm nhận vai trò giao tiếp dữ liệu giữa 2 máy khác nhau thông qua các dịch vụ mạng khác nhau (duyệt web, chay hay các giao thức trao đổi dữ liệu SMTP, SSH, FTP…).

**So sánh TCP/IP và mô hình OSI**

**Độ tin cậy chung**: **TCP/IP** được coi là một lựa chọn đáng tin cậy hơn so với mô hình **OSI**. Trong hầu hết các trường hợp, mô hình OSI được gọi là công cụ tham khảo.  
**Tính bảo mật**: **OSI** cũng được biết đến với **giao thức** và **ranh giới chặt chẽ**. **TCP/IP** cho phép “**nới lỏng**” **các quy tắc**, cung cấp các nguyên tắc chung được đáp ứng.  
**Về phương pháp tiếp cận**: **TCP/IP** thực hiện cách tiếp cận theo chiều ngang còn mô hình OSI thực hiện cách tiếp cận theo chiều dọc  
**Mô hình phân tầng**: **TCP/IP** kết hợp tầng phiên và tầng trình diễn trong tầng ứng dụng. Dường như OSI có một cách tiếp cận khác nhau, có các tầng khác nhau và mỗi tầng chỉ thực hiện một chức năng riêng.  
**Thiết kế mô hình**: Trong **TCP/IP**, các giao thức được thiết kế đầu tiên và sau đó mô hình được phát triển. Trong OSI, việc phát triển mô hình xảy ra trước và sau đó là phát triển giao thức.  
**Về truyền thông**: **TCP/IP** chỉ hỗ trợ truyền thông không kết nối phát ra từ tầng mạng. Ngược lại dường như OSI làm điều này khá tốt, hỗ trợ cả kết nối không dây và kết nối theo định tuyến trong tầng mạng.  
**Tính phụ thuộc**: **TCP/IP** là một mô hình phụ thuộc vào giao thức, còn OSI là một chuẩn giao thức độc lập.

**+ 2 điểm khác biệt giữa Go-Back-N và Selective Repeat**

Cả Giao thức Go-Back-N và selective repeate đều là các loại giao thức window sliding.Sự khác biệt chính giữa hai giao thức này là sau khi tìm thấy nghi ngờ hoặc thiệt hại trong các frame đã gửi, giao thức go-back-n sẽ truyền lại tất cả các frame trong khi **Selective Repeat** chỉ truyền lại frame bị hỏng.

**+ Cho biết 2 cách cấu hình bảng định tuyến của 1 router**

trả lời: định tuyến tĩnh: Việc xây dựng bảng định tuyến của router được thực hiện bằng tay bởi người quản trị.

định tuyến động:  Việc xây dựng và duy trì trạng thái của bảng định tuyến được thực hiện tự động bởi router.

Việc chọn đường đi được tuân thủ theo 2 thuật toán cơ bản:  
+ Distance vector: Chọn đường đi theo hướng và khoảng cách tới đích.  
+ Link State: Chọn đường đi ngắn nhất dựa vào cấu trúc của toàn bộ mạng theo trạng thái của các đường liên kết mạng.

**+ tại sao lại có sự ra đời IPv6, sự ra đời của IPv6 có cải tiến gì so với IPv4, xác định địa chỉ IPv6**

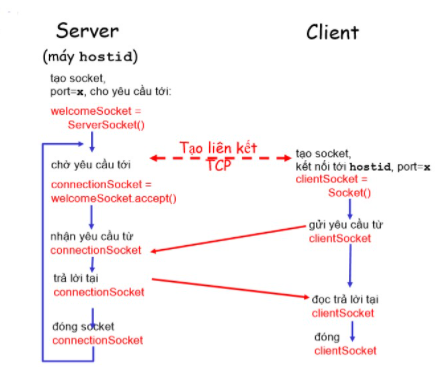
Trả lời: do sự phát triển của mạng internet ngày càng lớn nên cần mở rộng và nâng cấp không ngừng và trong đó vấn đề về địa chỉ IP ngày càng càng kiệt không thể đáp ứng nhu cầu phát triển này nên Ipv6 ra đời để cải tiến cho việc này.

Ipv6 có: Không gian gian địa chỉ IP lớn hơn vì Ipv6 hỗ trợ 128 bit địa chỉ nên số lượng địa chỉ host sẽ vô cùng lớn. Khuôn dạng header được đơn giản hóa để giảm chi phí đến mức tối thiểu bằng việc chuyển các trường không quan trọng sang các header mở rộng khác. Tự cấu hình địa chỉ: Ipv6 hỗ trợ việc tự cấu hình địa chỉ mà cần đến DHCP server. Khả năng xác thực và bảo mật: Ipv4 được thiết kế tại thời điểm chỉ có các mạng nhỏ nên biết rõ kết nối với nhau. Ipv6 được tích sắn khả năng bảo mật. Hỗ trợ tốt tính năng di động. Khả năng mở rộng: thiết kế Ipv6 có dự phòng cho sự phát triển trong tương lai nên dễ dàng mở rộng khi cần thiết.

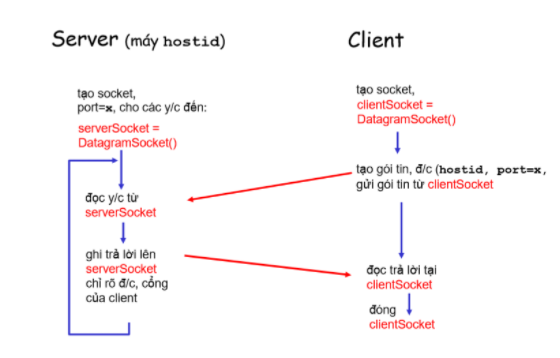
**+ định tuyến intradomain,**

**+ cách lập trình socket như thế nào, cách định danh socket trong TCP UDP, các client server trao đổi thông tin như thế nào.**

TCP: Ban đầu, phía server tạo Socket được ràng buộc với một cổng (port number) để chờ nhận yêu cầu từ phía client.Tiếp đến phía client yêu cầu server bằng cách tạo một Socket TCP trên máy kèm với địa chỉ IP và port number của tiến tình tương ứng trên máy server. Khi client tạo Socket, client TCP tạo liên kết với server TCP và chờ chấp nhận kết nối từ server. Khi được server accept thì client gửi request tới server và đợi phản hồi từ server.



UDP: Với UDP socket, chúng ta không cần thiết lập liên kết “handshaking” giữa client và server trước khi truyền thông điệp, thông điệp được gửi đi một cách độc lập. Bên gửi sẽ chỉ rõ địa chỉ IP và số hiệu cổng (port number) của bên nhận. Sau khi thông điệp được truyền đến bên nhận, bện nhận có thể dựa vào địa chỉ IP và số hiệu cổng tương ứng của bên gửi được gắn trên gói tin để gửi lại response.



**+ tại sao lại cần subnet mask trong IPv4? Cho địa chỉ IPv4 192.168.128/26**

**subnet mask tương ứng là gì, bao nhiêu bits làm net id, bao nhiêu bit làm host id?**

trả lời: subnet mask dùng để phân biệt phần netid và hostid, subnet mask tương ứng: 255.255.255.192, sử dụng 26 bit làm net id và 6 bit làm host id

**+ ứng dụng mạng có những đặc tính gì thì nên dùng giao thức TCP?**

trả lời: Đảm bảo việc truyền dữ liệu, Không cho phép mất gói tin, Độ tin cậy cao, Hướng kết nối

**+ trình bày cách phát hiện lỗi bằng checksum?**  
trả lời: Để kiểm tra tính toàn vẹn của dữ liệu, bên gửi dữ liệu tính toán giá trị checksum bằng cách lấy tổng dữ liệu nhị phân được truyền. Khi nhận dữ liệu, bên nhận có thể thực hiện phép tính tương tự trên dữ liệu và so sánh nó với giá trị checksum do người gửi cung cấp. Nếu hai giá trị khớp nhau, bên nhận có thể tin rằng dữ liệu đã được gừi và nhận một cách chính xác. (dùng phương pháp checksum, dữ liệu và checksum phải được cộng lại thành giá trị 0 khi không có lỗi)

**+ hãy trình bày kỹ thuật phát hiện và sửa lỗi trong tầng data link. Kể ra các phương pháp phát hiện và sửa lỗi trong tầng data link?**trả lời: sử dụng checksum để phát hiện lỗi nếu phát hiện lỗi thì yêu cầu bên gửi retranmiss lại frame mà bị lỗi đó. Các phương pháp phát hiện lỗi ch ecksum, bit parity. Các phương pháp sửa lỗi Automatic Repeat request là một phương pháp truyền dữ liệu sử dụng mã phát hiện lỗi, thông báo xác nhận và / hoặc xác nhận tiêu cực và timeout để nhận được dữ liệu đáng tin cậy Thông thường, khi bên gửi không nhận được thông báo trước khi timeout xảy ra nó sẽ gửi lại frame cho đến khi nhận được đúng hoặc gửi cùng 1 frame đó nhiều lần cho bên nhận.

**+ hãy cho biết địa chỉ IP host không hợp lệ với subnet mask là 255.255.255.224?**

**222.81.55.128, 222.81.22.104, 222.81.56.130, 222.88.65.135**

trả lời: ta thấy với địa chỉ subnet mask trên thì có suy ra là số ở octet cuối mượn 3 bit làm netid và còn 5 bit làm hostid 3 bit netid suy ra có 2^3 là 8 subnet, 5 bit hostid suy ra có 2^5 - 2 = 30 host và số bước nhảy của mỗi subnet là 2^5 = 32 suy ra địa chỉ 222.81.55.128 không thể thuộc địa chỉ host mà nó thuộc địa chỉ mạng của subnet 5.

**+ nếu yêu cầu phát triển một ứng dụng vừa nhanh vừa có độ tin cậy cao thì sử dụng TCP hay UDP? cần phát triển gì thêm.**

trả lời: nếu vừa nhanh vừa tin cậy cao thì sử dụng TCP

**+ ý nghĩa trường fragment offset trong header của IPv4? tại sao lại xảy ra việc chia nhỏ gói tin trong IPv4**

trả lời: fragment offset dùng để chỉ vị trí của đoạn fragment trong phần phần IP datagram.chia nhỏ gói để kích thước các gói tin phù hợp với mạch chuyển mạng mà nó dùng để truyền dữ liệu.

**+ giao thức nào dùng để gửi email? trình bày qui trình gửi 1 mail.**

trả lời: SMTP, qui trình tin nhắn của người gửi sẽ được gửi từ mail server của người gửi tới mail server của người nhận dựa trên địa chỉ email của người nhận. (kết nối TCP trực tiếp giữa 2 mail server) mailserver ở đây là hệ thống mail như gmail, yahoo

**Cho địa chỉ IP: 172.29.217.10/19**

**địa chỉ mạng: 172.29.224.0**

**địa chỉ broadcast: 172.29.224.255**

**số host IP: 2^13 - 2**

**dãy địa chỉ ip trong cùng network: 172.29.224.1 - 172.29.224.254**

**Ở đây, IP có vai trò quan trọng, nó cho phép các gói tin được gửi đến đích đã định sẵn,**

**bằng cách thêm các thông tin dẫn đường (chính là Header) vào các gói tin để các gói tin được đến đúng đích đã định sẵn ban đầu.**

**Giao thức TCP đóng vai trò kiểm tra và đảm bảo sự an toàn cho mỗi gói tin khi đi qua mỗi trạm.**

**Trong quá trình này, nếu giao thức TCP nhận thấy gói tin bị lỗi, một tín hiệu sẽ được truyền đi và yêu cầu hệ thống gửi lại một gói tin khác.**