**A. KHÁI NIỆM CƠ BẢN:**

**1. Mạng máy tính là:** hệ thống các máy tính được liên kết với nhau bằng phương tiện truyền dẫn để liên lạc, trao đổi thông tin (tin cậy, công bằng, hiệu quả)

+ tự phát hiện, sửa lỗi (hư data, mất data, data truyền đi bị trùng, data đến đích k đúng thứ tự)

+ tự tìm đường đi tối ưu đến đích

+ chia sẻ tài nguyên (tập tin thư mục, máy in máy fax).

**2. Internet:** mạng của các mạng, là mạng WAN lớn nhất, có khả năng truy cập toàn cầu.

**3. Cấu trúc Internet**: local → tier 3 → tier 2 → tier 1 (nhà cung cấp dịch vụ mạng).  
 **4. Phân loại:**

Địa hình: Cục bộ (LAN): kích thước nhỏ, thuộc 1 đơn vị or tổ chức, tốc độ cao ít lỗi, rẻ tiền.

Đô thị (MAN): nhiều mạng LAN kết hợp lại, có phạm vi 1 thành phố, thuộc 1 đơn vị hoặc tổ chức, chậm nhiều lỗi, chi phí cao hơn LAN.

Diện rộng (WAN): nhiều LAN, MAN kết hợp lại, phạm vi quốc gia châu lục quốc tế, thuộc nhiều đơn vị hoặc tổ chức, chậm nhiều lỗi, chi phí cao hơn LAN, MAN.

\* Chống vấn đề nghe lén tại mạng LAN: thay thế thiết bị tập trung Hub bằng Switch, sử dụng tính năng port security để hạn chế các thiết bị mạng kết nối trái phép, nên sử dụng địa chỉ IP tĩnh và bảng ARP tĩnh để hạn chế khả năng bị tấn công kiểu ARP spoofing thông qua giám sát chặt chẽ sự thay đổi địa chỉ MAC (Media Access Control) trên thiết bị switch, thay đổi password liên tục, mã hóa dữ liệu truyền dẫn bằng các cơ chế truyền thông dữ liệu an toàn SSL (Secure Sockets Layer), mạng riêng ảo VPN.

Phạm vi hoạt động: Intranet: nội bộ 1 đơn vị. Extranet: là Intranet nhưng accept bên ngoài truy cập vào thông qua chứng thực. Internet: accept bên ngoài truy cập.

Phương tiện truyền dẫn: Có dây. Không dây: infrastructure, Ad-hoc.

**5. Kiểu truyền:**

Unicast: từ 1 địa chỉ này đến 1 địa chỉ khác (chỉ 1 người gửi và 1 người nhận).

Broadcast: từ 1 địa chỉ đến tất cả các địa chỉ trong 1 vùng mạng.

Multicast: từ 1 địa chỉ đến 1 tập địa chỉ khác, (1 người gửi và nhiều người nhận).

Anycast: từ 1 địa chỉ đến 1 địa chỉ bất kỳ trong 1 nhóm.

**6. Giao thức:** hiểu như là một thống nhất (quy tắc, quy định) giữa các đối tượng trên mạng khi trao đổi thông tin, thống nhất về dạng dữ liệu trao đổi (syntax, semantic), thứ tự thông tin truyền nhận các thực thể. Do các tổ chức, hiệp hội xây dựng và quy định như: IEEE, ANSI, TIA, …

**7. Băng thông** (bandwidth): lượng thông tin trên lý thuyết có thể truyền đi trên 1 kết nối mạng trong 1 khoảng thời gian (Đơn vị tính: bps, Mbps, Gbps, …)

Thông lượng (throughput): lượng băng thông thực tế, nhỏ hơn nhiều so với băng thông lý thuyết; chịu ảnh hưởng: thiết bị liên mạng, topology, số lượng user, …

**8. Độ trễ**: thời gian trễ của 1 gói tin (dùng lệnh ping, tracert, pathping), nguyên nhân:

+ Trễ do tốc độ truyền: D(trans)=L/R(s) (dung lượng gói data (bit)/ băng thông sợi dây (bps))

+ Trễ trên đường truyền: thời gian gian truyền 1 bit nơi gửi → nơi nhận, D(prop)=d/c(chiều dài dây/tốc độ truyền)

+ Xử lý tại nút: thời gian xử lý header của 1 gói tin xem có lỗi hay không và quyết định chuyển hướng gói tin đi đâu D(proc) = khá nhỏ.

+ Hàng đợi: D(queue) thời gian đứng trong hàng đợi. (phụ thuộc vào số tập tin ở trước nó trong queue)  
**9. Firewall**: bảo vệ hệ thống, kiểm soát luồng dữ liệu: từ mạng bên trong đi ra bên ngoài và từ bên ngoài đi vào bên trong dựa vào Access Policy. Là một hệ thống bảo mật đặc biệt có khả năng quan sát và lọc các gói tin đi vào, đi ra. Nó thiết lập một rào cản ngăn cách mạng nội bộ và mạng bên ngoài.

Firewall không có chức năng quét virus, nhưng nếu AP được cập nhật có các thông tin về mối nguy là virus thì firewall sẽ đóng vai trò chặn mối nguy. Firewall không thể vô hiệu hóa phần mềm độc hại như quét virus.

**10. Proxy:** là một chương trình đặc biệt, dùng làm “thay thế kết nối” giữa client và server. Proxy giúp tạo điều kiện cho kết nối hai bên và có thể giúp tăng tốc độ trao đổi thông tin.

\* VPN: được gọi là mạng riêng ảo, cho phép những người dùng thiết lập được một mạng riêng ảo với một mạng khác trên hệ thống kết nối Internet, có thể được sử dụng dùng để truy cập trên các trang web bị hạn chế truy cập về mặt vị trí địa lý, bảo vệ hoạt động trình duyệt web của bạn khỏi những truy cập không cần thiết trên mạng Wifi chung; bằng cách thiết lập hệ thống mạng riêng ảo cho người sử dụng.

Ưu: Ẩn danh tính trực tuyến. VPN giúp bạn vượt qua khoảng cách địa lý. VPN bảo mật các kết nối trực tuyến. VPN có thể ngăn chặn việc điều chỉnh băng thông. VPN có thể vượt tường lửa

Nhược: VPN đôi khi làm chậm tốc độ truy cập internet. Sử dụng VPN sai có thể khiến quyền riêng tư của bạn gặp nguy hiểm. VPN chất lượng sẽ khá tốn tiền. Không phải tất cả các thiết bị đều hỗ trợ VPN.

| Firewall | Proxy |
| --- | --- |
| Dùng để giám sát gói tin, lọc gói tin dựa trên chính sách bảo mật đã thiết lập.  Ngăn chặn những truy cập bất hợp pháp từ bên ngoài vào và những gói tin nhạy cảm từ nội bộ ra bên ngoài.  Nằm giữa mạng nội bộ và mạng bên ngoài. Hoạt động ở cấp độ gói tin.  Bảo vệ mạng nội bộ trước những tấn công từ bên ngoài.  Chi phí cao hơn proxy. | Là trung gian cho kết nối giữa client và server. Tạo điều kiện cho kết nối client và server diễn ra thuận lợi.  Nằm giữa các mạng công cộng. Hoạt động ở cấp độ ứng dụng.  Bảo vệ mạng nội bộ trước những tấn công từ bên ngoài.  Chi phí thấp hơn firewall |

\*Ẩn danh trên Internet:

+ Trước khi kết nối wifi phải bảo đảm mọi tài khoản cá nhân đã ngắt kết nối, không dùng Google Chrome vì có liên kết tài khoản Google, không dùng ứng dụng nào kết nối dịch vụ thanh toán, dịch vụ onl, nhất là những app có liên quan thông tin cá nhân

+ Dùng hệ điều hành chuyên để hack: Linux ⇒ Không dùng Window vì có nhiều dịch vụ lấy thông tin gửi từ máy lên Microsoft còn Linux là mã mở tùy thuộc vào người dùng ⇒ Bảo đảm, người dùng quản lý các kết nối thông tin trên Internet

+ Dùng máy có hệ điều hành có tuổi đời ngắn (không dùng hệ điều hành chạy trên ổ cứng => Dùng hệ điều hành chạy trên USB ⇒ Không có cookie, file tạm,..

+ Thay đổi (giả mạo) MAC Address (đề phòng trường hợp đã bị theo dõi trên cùng đường mạng) => Cài phần mềm thay đổi

+ Giấu địa chỉ vật lý đang ngồi ⇒ Fake IP bằng mạng trung gian vd: Proxy, VPN,...

+ Sử dụng các browser tối ưu như Tor: sẽ dùng mạng riêng kết nối qua những server riêng (các kết nối đều được mã hoá) đi đến trang web đích để lấy thông tin ra.

**11. Phương thức truyền dữ liệu:**

+ Chuyển mạnh gói: host chia nhỏ dữ liệu cho tầng Application thành các gói (packets). Chuyển tiếp các gói từ 1 router này đến router tiếp theo thông qua các đường link trên đường đi từ nguồn → đích.

Cách thức hoạt động: nếu sóng điện từ là truyền link vô hướng, thì ở trong mạng lõi là link có hướng trên đường path. Để truyền từ source → destination gồm 2 thông tin: địa chỉ nơi đến (giúp tính toán được đường đi đúng trong mạng lõi) và thông tin muốn truyền. Dựa vào thuật toán định tuyến cùng bảng định tuyến có trong mỗi router. Mỗi gói tin khi đến router sẽ thực hiện chuyển hướng đến 1 router khác dựa vào gtri header của gói tin đó.

Ưu: mỗi packet được truyền tải với công suất max của đường link, nhiều người được sử dụng mạng vì các đường link không bị chiếm giữ liên tục, hiệu suất cao vì kích thước các gói tin được thiết kế sao cho node mạng có thể xử lý nhanh nhất mà ko cần lưu trữ tạm thời trên đĩa.

Nhược: các packet sẽ xếp hàng và đợi để được truyền tải trên đường link nên tốc độ truyền tại bị hạn chế, các packet có thể bị mất nếu bộ nhớ đệm bị đầy.

+ Chuyển mạch mạch: khi có 2 thực thể cần trao đổi thông tin thì giữa chúng sẽ thiết lập 1 kênh (circuit) cố định và duy trì cho đến khi 1 trong 2 bên ngắt liên lạc. Các dữ liệu chỉ được truyền theo con đường cố định đó. Mỗi “cuộc gọi” chiếm 1 tài nguyên nhất định.

Ưu: dữ liệu được truyền với độ trễ rất thấp, khó xảy ra mất dữ liệu.

Nhược: tiêu tốn thời gian để thiết lập kênh cố định, hiệu suất sử dụng đường truyền không cao vì khi 2 bên hết thông tin cần truyền, kênh bị bỏ không trong khi các thực thể khác cần thì không được phép sử dụng kênh

Khắc phục: FDM (ghép kênh theo tần số): truyền nhiều tần số trên 1 đường truyền, nên người nhận sẽ nhận được tất cả tín hiệu (tránh xung đột gây tắc nghẽn những lãng phí vì có những lúc không phải lúc nào mình muốn truyền hết tín hiệu thì không thể dùng lấn kênh truyền để tăng tốc độ).

TDM (ghép kênh theo thời gian): 1 đường truyền xài luân phiên theo chu kỳ, chờ đúng thời điểm mới truyền tín hiệu đi, chiếm hết dải tần (truyền nhanh hơn nhưng ngắt quãng, không đụng độ, lãng phí vì mỗi thằng chỉ được xài 1 thời gian, không lấn sang thời gian mà kênh khác không xài).

| Kiểu thiết kế | Ưu | Nhược |
| --- | --- | --- |
| Bus: nối trực tiếp vào 1 đường mạng chung | Tiết kiệm độ dài cáp, rẻ, dễ lắp đặt. Hệ thống đơn giản, tin cậy, dễ mở rộng | Gây ùn tắc nếu lượng dữ liệu lớn. Hư khó phát hiện, ảnh hưởng n người dùng |
| Ring: tạo thành vòng tròn | Mọi máy bình đẳng, hoạt động tốt nếu có n người dùng | 1 máy hư là đi. Hư khó phát hiện, không thể hoạt động khi đang cấu hình |
| Star: nối trực tiếp vào 1 thiết bị trung tâm | Sửa đổi hệ thống và thêm máy mới dễ dàng, hoạt động bình thường nếu có 1 máy hư | Trung tâm có sự cố là đi |
| Mesh: 2 thiết bị nối trực tiếp với nhau | Hệ thống cung cấp, khôi phục dữ liệu và độ tin cậy, dễ xử lý khi có sự cố | Tốn kém vì dùng n dây cáp |

**B. MÔ HÌNH OSI – TCP/IP:**

OSI là khung sườn biểu diễn cách thông tin di chuyển trên mạng.

TCP/IP: thể hiện sự phân tầng giao thức và đặc tả chồng giao thức Internet.

| OSI | Chức năng | TCP/IP | Chức năng |
| --- | --- | --- | --- |
| Application | Cung cấp dịch vụ mạng | Application | Cung cấp dịch vụ mạng, cung cấp các ứng dụng, trao đổi dữ liệu được mã hóa |
| Presentation | Cách biểu diễn dữ liệu |
| Session | Quản lý các phiên của ứng dụng |
| Transport | Truyền dữ liệu end to end | Transport | Cung cấp kết nối logic giữa các tiến trình, truyền dữ liệu giữa 2 app end to end |
| Network | Truyền dữ liệu host to host | Internet | Cung cấp kết nối logic giữa các host, truyền dữ liệu từ host to host |
| Datalink | Truyền dữ liệu link to link | Host to network (Network interface) | Điều khiển truy cập đường truyền, điều khiển liên kết |
| Physical | Truyền dữ liệu nhị phân |

Giống nhau: Đều phân lớp chức năng. Có lớp ứng dụng gồm một số các dịch vụ. Đều có lớp vận chuyển và lớp mạng. Hỗ trợ chuyển mạch gói. Đều giống nhau về mối liên hệ trên dưới và ngang hàng.

Khác nhau: TCP/IP gộp chức năng lớp trình bày và lớp phiên vào lớp ứng dụng. Gộp lớp vật lý và lớp liên kết dữ liệu vào thành một lớp. TCP/IP đơn giản vì có ít lớp hơn. OSI không có khái niệm chuyển phát thiếu tin cậy ở lớp 4 như UDP ở mô hình TCP/IP

\* Đóng gói dữ liệu:

Application: thông tin được chuyển thành dạng chung để mã hoá và nén dữ liệu tạo thành Message, rồi chuyển cho tầng Transport.

Transport: nhận Message và ghép thêm thông tin Header (thông tin cho phép Transport chuyển lên ứng dụng thích hợp và các bit phát hiện lỗi) tạo nên các Segment và chuyển cho tầng Network.

Network: thêm các thông tin như được nguồn và đích tạo ra Datagram rồi chuyển đến tầng Datalink.

Datalink: thêm các info Header (thông tin đk luồng, thông tin kiểm tra lỗi dữ liệu, FCS…) tạo thành Frame.

Physical: chuyển từng bit trong 1 frame từ nút này đến nút kế tiếp.

**C. ĐỊA CHỈ IP:**

Địa chỉ IP cung cấp danh tính của các thiết bị được kết nối mạng dùng để nhận diện và liên lạc, giao tiếp với nhau trên mạng máy tính bằng cách sử dụng giao thức Internet.

Ưu: IP là giao thức kết nối giao tiếp giữa các thiết bị mạng qua Internet. Giúp truy cập Internet dễ dàng hơn. Quản lý hệ thống mạng đơn giản và chặt chẽ. Đây là sự phát triển vượt bậc.

Nhược: thông tin cá nhân dễ dàng bị khai thác nếu bị ai đó xâm nhập và phá hoại. Hoạt động truy cập của user sẽ bị lưu lại địa chỉ IP

**1. Phân loại:**

Public IP: được mà nhà cung cấp dịch vụ Internet sd để chuyển đi các y/c Internet đến 1 gđ or tổ chức cụ thể. Đây là địa chỉ mà mạng gia đình hay tổ chức sử dụng để liên lạc với các thiết bị kết nối Internet khác, cho phép các thiết bị trong mạng truy cập mạng hay liên lạc trực tiếp với máy tính của user khác ⇒ Địa chỉ thật.

Private IP: được riêng sử dụng trong nội bộ mạng LAN như mạng gia đình, trường học, cty. Khác với IP Public, IP Private không thể kết nối với mạng Internet mà chỉ có các thiết bị trong mạng mới có thể giao tiếp với nhau thông qua bộ định tuyến router. Địa chỉ IP riêng được bộ định tuyến gán tự động or bạn tự thiết lập thủ công ⇒ Địa chỉ ảo.

Địa chỉ loopback: là 1 loại địa chỉ dành riêng chỉ có trên máy tính cục bộ. Nếu 1 máy tính cố gắng gửi 1 tin nhắn đến địa chỉ loopback thì tin nhắn đó sẽ không bao giờ được gửi đến mạng mà thay vào đó sẽ lặp lại thẳng vào máy tính. Điều này thường không hữu ích với user nhưng nó hữu ích khi truy cập các dịch vụ mạng như máy chủ web trên thiết bị. 172.0.0.0 – 172.255.255.255 (các được class A có range là 127).

**2. Địa chỉ đường mạng:** dùng để biết được địa chỉ đường mạng có máy đó, 2 địa chỉ IP có cùng địa chỉ đường mạng thì chúng ở cùng đường mạng. Quy tắc tìm: bit Net giữ nguyên, bit Host chuyển về 0.

**3. Địa chỉ Broadcast:** là địa chỉ đặc biệt của mạng dùng để gửi thông điệp cho all các máy trên đường mạng này (send to all). Quy tắc tìm: bit Net giữ nguyên, bit Host chuyển về 1. (bit cuối luôn lẻ)

**4. Số địa chỉ host**: số hostIP hợp lệ là: 2^(số bit trong bit host) – 2, trừ 2 vì cần lại Net Addr và Broadcast

**5. Phân lớp**: để tiện quản lý → phân mạng thành các lớp với đặc trưng nhất định.

Class A(0): range (1 – 126).

Class B(10): range (128 – 191).

Class C(110): range (192 – 223) .

**6. Địa chỉ MAC**: ở tầng Datalink, gồm 6 bytes.

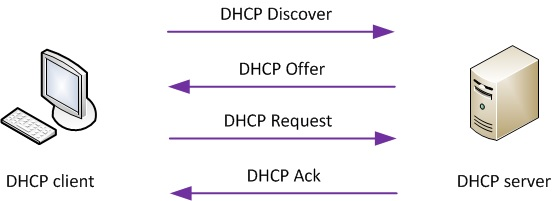
3 bytes đầu: OUI – định danh của NSX đã có đăng ký từ trước với IEEE.

3 bytes sau: NIC – thông tin NSX ấn định cho mỗi card mạng.

**D. DHCP:**

Là dịch vụ cung cấp tự động địa chỉ IP cho các thiết bị → admin ko cần can thiệp, giảm quản trị → mạng ổn định hơn (số lượng được lớp, khó trùng được khi phải cấp thủ công, giảm xung đột)

Là dịch vụ ở tầng Application, theo Client-Server. Sử dụng kiểu UDP để trao đổi ở tầng Transport, bên Client sẽ là bên khởi tạo kết nối trước dùng để cấp địa chỉ IP động.

Mô hình hoạt động: 

DHCP Discover: gói tin được gửi đến DHCP server khi có 1 thiết bị yêu cầu cung cấp được IP truyền kiểu Broadcast truy cập vào mạng.

DHCP Offer: được DHCP server gửi phản hồi cho Client sau khi nhận được DHCP Discover. Gói chứa được IP, cấu hình TCP/IP bổ sung.

DHCP Request: gói tin được DHCP Client phản hồi về server về sự chấp nhận được IP sau khi nhận DHCP Offer.

DHCP Acknowledge: gói tin mà DHCP server phản hồi lại với Client nhằm xác minh rằng đã chấp nhận được IP DHCP Request đồng thời định hướng các tham số để cho phép Client truy cập mạng TCP/IP. Nếu DHCP khác đường mạng: Router sẽ không nhận UDP Broadcast → cần cấu hình dịch vụ DHCP Relay Agent cho Router biết nào là gói tin DHCP thì Router hãy cho qua.

Tại sao lại sử dụng UDP: UDP không có cơ chế đảm bảo việc chuyển dữ liệu, không cung cấp tính năng kiểm tra lỗi và không kiểm soát luồng dữ liệu, và nó cũng không có yêu cầu bắt tay 3 bước như TCP => Tốc độ của UDP nhanh hơn TCP rất nhiều. Client không biết địa chỉ của DHCP server và ban đầu chúng thường chưa có một địa chỉ IP nào cả. Chính vì thế trao đổi giữa client và server phải sử dụng những gói tin Broadcast. Kết nối có hướng như TCP yêu cầu cần phải xác định rõ IP của người gửi và người nhận, điều này không phù hợp với những gói tin Broadcast. Chính vì thế mà DHCP sử dụng UDP (kết nối vô hướng). Ngoài ra, nếu TCP sử dụng trong kết nối Broadcast, thì nó phải thiết lập kết nối đến tất cả các máy trong một mạng. Điều này đồng nghĩa là client phải cung cấp tài nguyên cho tất cả các kết nối trên => lãng phí tài nguyên không cần thiết, đôi khi không đủ tài nguyên nếu mạng nội bộ quá lớn.

DHCP dùng UDP vì: TCP là hướng kết nối yêu cầu cả 2 điểm đầu cuối phải có được IP duy nhất. Tại thời điểm mà 1 máy lấy được yêu cầu sử dụng DHCP nó không có được IP mà nó có thể lấy các gói từ đó cũng như không có được IP của DHCP server. VÌ vậy nó sử dụng 0.0.0.0 làm được IP nguồn và 255.255.255.255 làm địa chỉ IP đích. Các địa chỉ IP này không phải là được IP máy chủ hợp lệ và có thể được sd bởi n` máy khác bất kỳ lúc nào. Do đó DHCP sử dụng UDP thay vì TCP.

Ưu: dùng cho máy kết nối không dây vì linh hoạt. Khó trùng được IP → hệ thống ổn định, đơn giản. Vừa quản lý được IP, tham số của TCP/IP → dễ dàng. Admin thay đổi cấu hình và thông số → update system hiệu quả.

Nhược: không phù hợp các thiết bị cố định (máy in, PC). Các máy in có IP thay đổi liên tục khiến máy in phải cập nhật lại để có thể kết nối được với máy tính.

**E. DNS (Domain Name System):** hệ thống chuyển đổi các tên miền sang 1 địa chỉ IP tương ứng và ngược lại. Mục đích: định vị và gán 1 được cụ thể cho các thông tin trên internet.

Dịch vụ DNS ở tầng Application, giúp người dùng dễ nhớ địa chỉ truy cập hơn, mô hình Server-Client. Cơ sở dữ liệu lưu dạng cây, mỗi nút là gốc của 1 cây con, mỗi cây con là 1 miền, mỗi miền có nhiều miền con.

Giao thức dùng ở tầng Transport: UDP để truy vấn dữ liệu (do cần tốc độ nhanh). TCP để cập nhật thông tin (do cần độ chính xác cao, vì nếu sai thì không thể sử dụng dịch vụ được)

Lưu trữ: sử dụng CSDL phân tán và phân cấp, lưu dưới dạng các resource record – RR (name, value, type, ttl)

SOA: thông tin cho toàn bộ 1 zone.

MX: thông tin các name server nhận mail của miền.

NS: thông tin các name server quản lý zone(Name: tên miền, Value: được NS của miền).

A: dùng để phân giải tên máy thành được IP (Name: hostname, Value: IP address).

CNAME: lưu tên phụ 1 máy (Name: tên alias, Value: tên thật).

PTR: dùng để phân giải được IP thành tên máy (Name: ID addr, Value: hostname).

Phân giải truy vấn:

+ Truy vấn tuần tự: máy host sẽ hỏi trực tiếp lần lượt tất cả các DNS server cho đến khi đúng được đường mạng ứng với tên miền. Nếu Local DNS server đã lưu tên miền trong bộ nhớ cache thì lần sau gửi lại cho host để đỡ tốn thời gian.

+ Truy vấn đệ quy: máy host sẽ hỏi gián tiếp qua các server đã hỏi bằng tính chất bắc cầu để đi đến được đường mạng ứng với tên miền. Quy trình hỏi như “câu tên miền” – cấp thấp hỏi cấp cao hơn tới khi tìm ra.

Authoritative DNS Server: server giữ tên miền cần phân giải → trả lời đáp án cuối cùng.

Non-authoritative DNS Server: không quản lý tên miền bạn cần tìm sẽ trả lời thông qua việc caching/forwarder – tức là chuyển tiếp trung gian đáp án cho bạn.

Caching: lưu tạm kết quả đã truy vấn trong cache → giảm thời gian truy vấn + giảm lưu lượng mạng.

Forwarder: gồm Standard và Conditional. Kỹ thuật cho phép Name Server nội bộ chuyển yêu cầu truy vấn cho các Name Server khác để phân giải các miền bên ngoài. Khi Internal DNS Servers nhận yêu cầu truy vấn của máy trạm nó kiểm tra xem có thể phân giải được yêu cầu này hay không. Nếu không thì nó sẽ chuyển yêu cầu này lên Forwarder DNS server (multihomed) để nhờ Name Server này phân giải dùm. Sau khi xem xét xong thì Forwarder DNS Server (multihomed) sẽ trả lời yêu cầu này cho Internal DNS Servers or nó sẽ tiếp tục forward lên các Name Server ngoài Internet  
**F. TẦNG ỨNG DỤNG:** tầng này chứa các dịch vụ sẽ được sử dụng bởi các ứng dụng mạng, các máy khách và các máy chủ, các tiến trình và các phương thức giao tiếp với tầng vận chuyển.

Một tiến trình [process] là một chương trình [program] đang chạy trên một máy tính. Các tiến trình đang chạy trên cùng một máy tính có thể truyền thông với nhau, sử dụng các quy tắc do hệ điều hành trên máy tính quy định.

Liên lạc giữa các tiến trình:

Trên cùng một máy: hệ điều hành, chia sẻ bộ nhớ, truyền thông điệp giữa các tiến trình.

Trên 2 máy khác nhau: truyền dữ liệu qua đường mạng, ví dụ: socket, name pipe,…

Ứng dụng mạng: Chương trình ứng dụng mạng:

Chạy trên các end – system kết nối thông qua mạng

Một số mô hình phổ biến: server – client, peer-to-peer,…

**1. Server – client:**

Server: luôn luôn hoạt động, chạy trên một địa chỉ cố định, thường trong các trung tâm dữ liệu để mở rộng quy mô, nhận và yêu cầu xử lý từ client .

Client: (ví dụ: HTTP, IMAP, FTP,…) liên hệ, giao tiếp với máy chủ, có thể được kết nối không liên tục, có thể dùng IP động, 2 client không thể liên lạc trực tiếp với nhau.

**2. Mô hình peer-to-peer:** không có máy chủ hoạt động, ứng dụng có cả 2 chức năng của server và client, khả năng tự mở rộng mang lại năng lực dịch vụ mới cũng như nhu cầu dịch vụ mới, các client có thể liên lạc trực tiếp, dùng địa chỉ “động”, quản lý khó. VD: P2P file sharing, Skype, Bittorrent,…

**3. Địa chỉ của tiến trình: địa chỉ IP + port**

0..1023: port chuẩn 1024..49151: port cố định đăng ký trước …: port linh động

**4. Giao thức tầng ứng dụng:**

Do người cài đặt ứng dụng xây dựng. Ví dụ: HTTP, FTP,…

Những yêu cầu dịch vụ của tầng ứng dụng: truyền dữ liệu tin cậy, thông lượng, mức độ kịp thời bảo mật.

**5. Dịch vụ tầng transport cung cấp:**

Dịch vụ TCP:

+ Dịch vụ hướng kết nối: có đặc tính là kết nối 2 chiều client – server

+ Dịch vụ truyền dữ liệu đáng tin cậy: như truyền các gói dữ liệu chính xác và theo đúng thứ tự, đảm bảo dữ liệu không bị sai khác.

+ Kiểm soát tắc nghẽn

+ Thời gian: đảm bảo thông lượng tối thiểu bảo mật

Dịch vụ UDP: là giao thức vận chuyển đơn giản, gọn lẹ, chỉ cung cấp dịch vụ tối cần thiết. UDP là giao thức phi kết nối.

+ Dịch vụ truyền dữ liệu không đáng tin cậy - nghĩa là khi một tiến trình gửi một thông điệp qua socket bên gửi, UDP không đảm bảo thông điệp sẽ đến được tiến trình nhận.

+ Không cung cấp: Độ tin cậy, kiểm soát luồng, kiểm soát tắc nghẽn, thời gian, đảm bảo thông lượng, bảo mật hoặc thiết lập kết nối.

Tại sao vẫn dùng UDP?

Giao thức UDP hoạt động tương tự như TCP, nhưng nó bỏ qua quá trình kiểm tra lỗi. Khi một ứng dụng sử dụng giao thức UDP, và các gói tin được gửi cho bên nhận và bên gửi không phải chờ để đảm bảo bên nhận đã nhận được gói tin, do đó nó lại tiếp tục gửi gói tin tiếp theo. Do đó thiết bị có thể giao tiếp nhanh hơn. Ví dụ: bạn đang xem phát video trực tiếp, thường được phát bằng UDP thay vì TCP. Máy chủ sẽ gửi một lần liên tục các gói tin UDP đến máy tính đang xem. Suy ra: UDP thường được sử dụng cho các chương trình phát sóng trực tiếp và game online.

**6. Lập trình Socket:**

Socket là giao diện giữa tiến trình ứng dụng và giao thức tầng vận chuyển (TCP, UDP).

Cung cấp interface để lập trình mạng ở tầng transport. 1 socket là một end-point of 1 liên kết giữa 2 app.

**7. Lập trình Socket với UDP:**

Không có kết nối giữa máy khách và máy chủ.

Người gửi đính kèm rõ ràng địa chỉ đích IP và số cổng vào mỗi gói.

Người nhận trích xuất địa chỉ IP của người gửi và số cổng từ gói tin đã nhận. ⇒ UDP: dữ liệu đã truyền có thể bị mất hoặc nhận không theo thứ tự.

Quan điểm ứng dụng: UDP cung cấp khả năng truyền các nhóm byte (“datagram”) không đáng tin cậy giữa máy khách và máy chủ.

Giai đoạn 1: server tạo socket tại port

Giai đoạn 2: client tạo socket

Giai đoạn 3: trao đổi thông tin giữa client và server

**8. Lập trình socket với TCP:**

Khách hàng kết nối với máy chủ:

Quy trình máy chủ chạy

Máy chủ tạo socket đón khách hàng

Máy chủ liên hệ với khách hàng:

Tạo TCP socket, chỉ định địa chỉ ip, số cổng của tiến trình máy chủ

Khi máy khách tạo socket: máy khách TCP thiết lập kết nối với máy chủ TCP

Khi được khách hàng liên hệ, máy chủ TCP sẽ tạo một socket mới cho quá trình giao tiếp.  
 Cho phép máy chủ kết nối với nhiều máy khách.  
 Giai đoạn 1: server tạo socket và lắng nghe yêu cầu kết nối tại port   
 Giai đoạn 2: client tạo socket, yêu cầu thiết lập một kết nối với server   
 Giai đoạn 3: trao đổi thông tin giữa client và server

Giai đoạn 4: kết thúc phiên làm việc.

9. Dịch vụ mail:

Hoạt động theo mô hình client – server, theo giao thức TCP (qua port 25) và client sẽ là bên khởi tạo kết nối

trước.

Thông điệp của mail được viết dưới dạng 7 bit theo mã ASCII.

Gồm 3 thành phần tham gia chính: user agent (UA), mail servers và protocol vận chuyển mail.

Quá trình gửi mail:

User Agent: thiết bị đầu cuối để biên soạn và đọc mail.

Khi bên gửi nhấn nút send, mail sẽ được gửi đến email server của người gửi => email server người nhận, 2 giai đoạn đều thông qua giao thức SMTP. Lưu ý, ở mỗi server sẽ có 1 queue để duyệt từng mail đi qua theo đúng trật tự.

Từ email server người nhận, cần sử dụng giao thức như IMAP, HTTP để có thể gửi để UA người nhận và bên nhận có thể truy cập được.

So sánh web mail với mail client:

Webmail: giao thức nhận mail IMAP. Ưu: lưu trữ email trên server của Google, tiện dụng do không phải cài đặt, khả năng tìm kiếm mạnh mẽ, tính phổ biến. Nhược: do lưu trữ trên server của Google nên dung lượng rất hạn chế, cần Internet để truy cập vào email, nếu mail bị thất lạc phải liên hệ nhà cung cấp dịch vụ, quyền riêng tư.

Mail client: giao thức nhận mail POP 3. Ưu: lưu trữ được email trong máy truy cập khi không có Internet, có khả năng phục hồi nếu bị mất giảm rủi ro mất mát do lưu cục bộ, tiết kiệm không gian lưu trữ trên server. Nhược: email được lưu trữ trong máy nên nếu có vấn đề lỗi phần cứng/mềm có khả năng mất dữ liệu, phải cài đặt.

**10. HTTP**: Phiên bản của giao thức luôn được đặt trong tất cả thông điệp, tất cả các tiêu đề đềudưới dạng (text)

| Tiêu chí | Stateless | Stateful |
| --- | --- | --- |
| Định nghĩa | Là trạng thái không lưu các dữ liệu của client trên server | Là trạng thái lưu dữ liệu của client trên server |
| Ví dụ | HTTP, DNS, UDP | FTP, Telnet |
| Hạn chế máy chủ | Máy chủ không cần lưu giữ thông tin máy chủ hoặc các chi tiết cho phiên của nó | Máy chủ được lưu giữ để yêu cầu duy trì trạng thái hiện tại & thông tin phiên làm việc |
| Sự phụ thuộc | Máy chủ và máy khách được kết hợp lỏng lẻo, có thể hoạt động độc lập | Máy chủ và máy khách có sự ràng buộc chặt chẽ |
| Thiết kế | Đơn giản | Phức tạp |
| Sự cố | Dễ khởi động lại khi có sự cố | Lưu trữ nhiều thông tin nên có sự cố là tạch |
| Tốc độ | Nhanh | Chậm |

Non persistent HTTP: kết nối TCP được mở. Sau đó tối đa 1 đối tượng được gửi qua kết nối TCP đó. Đóng kết nối TCP. Tải/ gửi nhiều đối tượng thì cần nhiều kết nối TCP.

Persistent HTTP: kết nối TCP được mở đến server. Nhiều đối tượng có thể gửi qua 1 kết nối giữa client và server đó.

Kết thúc thì đóng kết nối TCP.

Trạng thái HTTP:

200 OK

▪ Yêu cầu thành công, đối tượng được yêu cầu kèm sau trong thông điệp này

o 301 Moved Permanently

▪ Đối tượng được yêu cầu đã di chuyển, vị trí mới được xác định sau trong thông

điệp này (trong trường Location: )

o 400 Bad Request

▪ Máy chủ không hiểu thông điệp yêu cầu

o 404 Not Found

▪ Tài liệu được yêu cầu không tìm thấy trên máy chủ này

304 : Not Modified

o 505 HTTP Version Not Supported

502: Bad GateWay

**G. TẦNG TRANSPORT:**

Tầng transport có nhiệm vụ cung cấp một kết nối logic giữa các ứng dụng chạy trên các máy khác nhau. Vận chuyển được cài đặt tại các máy ở đầu cuối end-system, không phải tại các bộ định tuyến.

Phía bên gửi: thực hiện dồn kênh: Đóng gói các thông điệp nhận từ các ứng dụng nguồn (socket) thành các gói tin tại tầng vận chuyển, gọi là các phân đoạn (segment). Các segment được chuyển xuống tầng mạng (network layer).

Phía bên nhận: thực hiện phân kênh: Nhận các segment từ tầng mạng (được chuyển từ tầng dưới lên tầng trên). Chia các segment thành các tiến trình, sau đó truyền lên các ứng dụng (tới socket tương ứng).

**1. Truyền dữ liệu:**

Hỗ trợ 2 hình thức đó là truyền dữ liệu đáng tin cậy và truyền dữ liệu không đáng tin cậy. Các giao thức này được thực hiện ở máy, không phải ở router (router không đọc gói tin đến tầng transport).

Không hỗ trợ đảm bảo thời gian và trễ băng thông.

Truyền dữ liệu đáng tin cậy (đảm bảo dữ liệu không bị mất mát), bao gồm các đặc trưng: điều khiển luồng, Điều khiển tắc nghẽn, Thiết lập (truyền hướng kết nối) và duy trì kết nối.

Việc truyền dữ liệu luôn cố gắng để cho gói tin truyền đi không bị mất mát. Tuy nhiên trên thực tế là rất khó vì khi truyền qua các đường truyền không tin cậy sẽ luôn xảy ra sai sót, phổ biến trong đó là lỗi mất gói và lỗi bit. “Nghi thức truyền tin cậy” ở tầng transport được phát kiến ra để đảm bảo việc truyền tin cậy khi phải truyền trên các đường truyền không tin cậy.

Độ phức tạp của nghi thức truyền tin cậy được quyết định bằng đặc tính của đường truyền không tin cậy (đường truyền càng dễ mất gói tin thì càng phải đóng gói kỹ).

Hướng giải quyết của 2 lỗi được đề ra như sau:

Lỗi bit: bên gửi sẽ gửi các gói tin cùng thông tin kiểm tra lỗi (sử dụng các phương pháp như checksum, crc,…). Phía bên nhận sẽ kiểm tra và sửa lỗi, gửi tín hiệu về bệnh gửi nếu xảy ra lỗi.

Lỗi mất gói: bên gửi sẽ định nghĩa trường hợp mất gói, chờ nhận tín hiệu báo để hành động khi mất gói. Bên nhận gửi gói tin báo hiệu ACK, NAK.

**2. Giao thức RDT (Reliable Data Transfer) :**

Nguyên tắc: dùng và chờ phản hồi

| Phiên bản | RDT 1.0 | RDT 2.0 | RDT 2.1 | RDT 2.2 | RDT 3.0 |
| --- | --- | --- | --- | --- | --- |
| Mục đích | Kênh truyền tin cậy | Xử lý lỗi bit khi truyền | Xử lý lỗi gói tin ACK/NAK (gói tin xác nhận) | Giống RDT 2.1, nhưng chỉ xài ACK (không sử  dụng NAK) | Xử lý lỗi bit và mất gói |
| Đặc điểm | Truyền và nhận bình thường, không sử dụng gói tin xác nhận | Sử dụng ACK để báo nhận gói tin thành công, NAK để báo gói tin bị  lỗi | Gửi ACK, NAK có kèm checksum và đánh số thứ tự cho gói tin gửi/nhận. | Như 2.1, nhưng không xài NAK, mà đánh số thứ tự cho gói tin phản hồi ACK | Đánh số thứ tự cho gói tin gửi (gồm gói tin và ACK). Có đếm thời gian để nhận ACK (Sau khi time out là thoát ra không thao tác nữa ) |

**3. Nguyên lý Pipeline**: Cho phép gửi nhiều gói tin khi chưa nhận ACK.

Sử dụng Buffer để lưu các gói tin.

Bên gửi: lưu gói tin đã gửi nhưng chưa ACK.

Bên nhận: lưu gói tin đã nhận đúng nhưng chưa đúng thứ tự.

Giải quyết mất gói:

Go back N

Gửi lại có chọn (selective repeat)

**4. Go-back-N**

Nguyên lý hoạt động: Khi một khung bị lỗi, bên nhận bỏ qua khung vì không một báo nhận nào gửi về cho bên nhận nên sự kiện quá thời gian xảy ra, bên gửi phải gửi lại khung bị lỗi và toàn bộ các khung phía sau nó.

Số thứ tự: k-bit; “window” = N => số gói tin được gửi liên tục k ACK; ACK (seq#): nhận đúng đến seq#.

-Bên gửi:

Sử dụng buffer (“window”) để lưu các gói tin đã gửi nhưng chưa nhận được ACK. Gửi nếu gói tin có thể đưa vào “window”.

Thiết lập đồng hồ cho gói tin cũ nhất (gói tin ở đầu “window”).

Timeout: gửi lại tất cả các gói tin chưa ACK trong window.

-Bên nhận:

Chỉ gửi ACK cho gói tin đã nhận đúng với số thứ tự cao nhất (có thể phát sinh trùng ACK). Chỉ cần nhớ số thứ tự đang đợi.

Gói tin không theo thứ tự (loại bỏ: không có bộ đệm, gửi lại ACK với số thứ tự lớn nhất).

**5. Gửi lại có chọn**

Nguyên lý hoạt động: Khung bị lỗi bỏ đi, nhưng các khung nhận tốt sau đó đều được lưu lại tạm thời trong vùng nhớ đệm. Khi quá thời gian, bên gửi chỉ gửi lại khung cũ nhất chưa được báo nhận. Nếu khung này đến nơi chính xác, bên nhận có thể chuyển lên tầng mạng tất cả các khung đã được lưu vào bộ nhớ đệm theo đúng thứ tự.

-Bên nhận:

Báo nhận riêng lẻ từng gói tin nhận đúng (ACK(seq#): đã nhận đúng gói tin seq#).

Dùng bộ đệm để lưu các gói tin không đúng thứ tự.

Nhận 1 gói tin không đúng thứ tự (đưa vào bộ đệm nếu còn chỗ, hủy gói tin).

-Bên gửi:

Có đồng hồ cho mỗi gói tin chưa nhận được ACK

Timeout: chỉ gửi những gói tin không nhận được ACK

6. Giao thức TCP (transport control protocol)

Kiểu point-to-point (1 người gửi và 1 người nhận).

Full-duplex: song song, dữ liệu 2 chiều trên cùng 1 kết nối.

Gồm 1 dãy byte có thứ tự, nhưng không cấu trúc.

Hướng kết nối: handshaking trước khi gửi dữ liệu. Cung cấp kết nối theo kiểu dòng (stream-of-bytes), không có ranh giới giữa các gói tin, sử dụng buffer gửi và nhận. Tin cậy, theo thứ tự. Pipeline. Điều khiển luồng. Điều khiển tắc nghẽn.

Cấu trúc 1 gói tin:

Source & destination point: port của nơi gửi và nơi nhận

Sequence number: thứ tự của byte đầu tiên trong phần data của gói tin

Acknowledgment number: số thứ tự của byte đang mong chờ nhận tiếp theo

Window size: thông báo có thể nhận bao nhiêu byte sau byte cuối cùng được xác nhận đã nhận.

Checksum: Checksum TCP header.

Urgent pointer: chỉ đến dữ liệu khẩn trong trường dữ liệu.

Cờ: URG = trường urgent pointer valid. ACK = trường Acknowledgement number valid. PSH = dữ liệu cần phân phối ngay. RST = chỉ định nối kết cần thiết lập lại (reset). SYN = sử dụng để thiết lập kết nối. FIN = sử dụng để đóng kết nối

TCP là truyền dữ liệu đáng tin cậy:

Nguyên tắc: dùng pipeline

Bên gửi đính kèm thông tin kiểm tra lỗi trong mỗi gói tin.

Sử dụng ACK để báo nhận.

Thiết lập thời gian timeout khi cho gói tin ở đầu buffer.

Gửi lại toàn bộ dữ liệu trong buffer khi hết timeout.

Phía bên gửi:

Nhận dữ liệu từ tầng ứng dụng (tạo các segment, bật đồng hồ nếu chưa bật, thiết lập thời gian chờ, timeout).

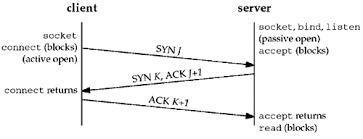
Nhận gói tin ACK (nếu trước đó chưa nhận: trượt window, thiết lập lại thời gian của đồng hồ).

Hết timeout (gửi lại dữ liệu còn trong buffer, reset đồng hồ).

Phía bên nhận:

Nhận gói tin đúng thứ tự (chấp nhận,gửi ACK về cho bên gửi).

Nhận gói tin không đúng thứ tự (phát hiện “khoảng trống dữ liệu (GAP)”, gửi ACK trùng).

Kết nối: 

Thiết lập kết nối: three-way handshake  
 Đóng kết nối: bắt tay 2 lần  
 Quản lý kết nối:

+ Điều khiển luồng:

Nguyên nhân: bên gửi làm tràn bộ đệm của bên nhận khi gửi quá nhiều dữ liệu hoặc gửi quá nhanh. Sử dụng trường “trường window size”: lượng dữ liệu có thể đưa vào buffer.

+ Kiểm soát tắc nghẽn:

Vấn đề: 1 node có thể nhận dữ liệu từ nhiều nguồn (buffer giới hạn, gói tin đến ồ ạt) => xử lý không kịp => tắc nghẽn. Hiện tượng mất gói, delay cao => sử dụng đường truyền không hiệu quả

Hướng giải quyết:

Bên gửi: thiết lập tốc độ gửi dựa trên phản hồi từ bên nhận (nhận ACK, mất gói, độ trễ gói tin).

Tốc độ gửi: có 2 pha slow – start và congestion avoidance

**7. Giao thức UDP:**

Mục tiêu: truyền nhanh, gọn cho nên dễ mất gói tin, các gói tin không đi theo đúng thứ tự ở cũng như không cần handshaking trước khi gửi.

Đơn giản, header nhỏ, chạy nhanh.

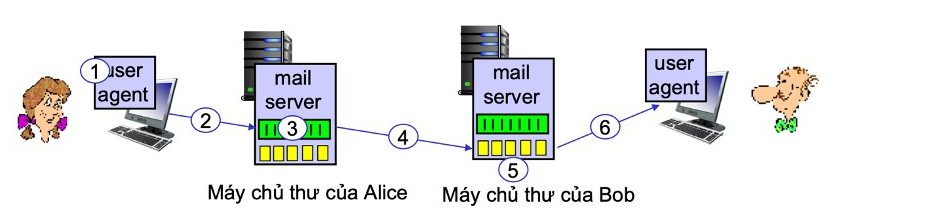
Một số giao thức có sử dụng UDP: DNS, SNMP, TFTP

**MỘT SỐ CÂU HỎI TRẮC NGHIỆM**

1. Giao thức RDT nào cung cấp dịch vụ giao hàng tin cậy?

- RDT 3.0

2.Ở bước 2 trong sơ đồ giao thức nào được sử dụng - SMTP

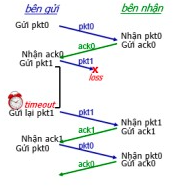


3 Hãy chọn phát biểu SAI: HTTP 1.0 sử dụng HTTP bền vững, HTTP 1.1 sử dụng HTTP không bền vững

1. Phát biểu nào sau đây là sai về kết nối HTTP không bền vững?

HTTP không bền vững không đòi hỏi hai chuyến đi-đến (2RTTs) cho mỗi đối tượng.

5 Xem hình và cho biết đây là trường hợp nào của rdt 3.0? - MẤT Gói



6. Mô tả nào sau đây SAI về giao thức IMAP: Là Internet Mail Access Protocol

7. Tong RDT 3.0, chuyện gì sẽ xảy ra khi bên gửi không nhận được ACK của bên nhận?

-Bên gửi tự phát hiện lỗi và gửi lại gói tin sau khi thời gian chờ hết hạn

8 Khi nào kết nối TCP của HTTP không bền vững sẽ được đóng?

-Ngay sau khi một đối tượng duy nhất đã được gửi qua kết nối.

1. Phát biểu nào sau đây đúng về giao thức HTTP?

-Phiên bản của giao thức luôn được đặt trong tất cả thông điệp, tất cả các tiêu đề đều dưới dạng text

10 Phát biểu nào dưới đây là SAI đối với kết nối HTTP Không bền vững -

Có thể gửi nhiều đối tượng trên một kết nối

11Cho một phiên làm việc SMTP ,hãy sắp xếp trình tự giao tiếp đúng ở phía cilent

HELO, MAIL FROM, RCPT TO, DATA, QUIT

12 Trong quá trình phân giải tên miền, việc đẩy trách nhiệm phân giải tên cho máy chủ tên miền được hỏi gọi là? - Truy vấn đệ quy

13 Để giảm lưu lượng trên đường liên kết truy cập ra Internet của một tổ chức, người ta thường dùng: Web caching

14 Mục đích của số port trong header của TCP và UDP là gì? - Xác định tiến trình đang gửi/nhận dữ liệu

15 Trong header của UDP, trường length là độ dài của thành phần nào sau đây? - Cả segment UDP

16 Ứng dụng nào sau đây không sử dụng giao thức UDP? Các ứng dụng truyền nhận tin cậy

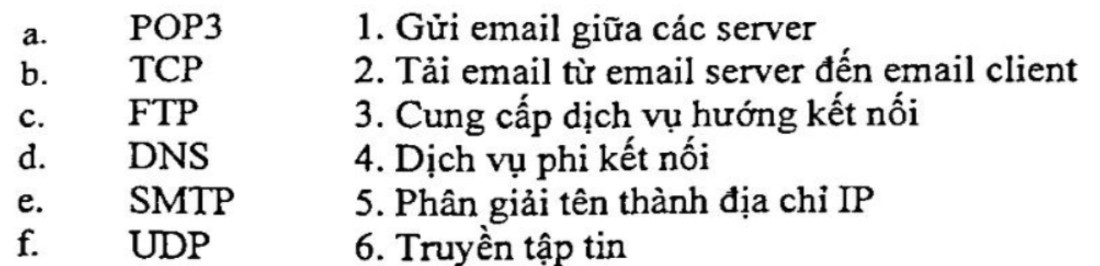
17 Trong giao thức truyền dữ liệu tin cậy (rdt), giao thức nào sau đây xử lý được trường hợp mất gói tin ACK? - rdt 3.0

18Trong quá trình truyền thông điệp thư điện tử với giao thức SMTP, thông điệp cần phải ở dạng: ASCII 7 bit

19 Phát biểu nào sau đây là sai khi nói về HTTP bền vững? HTTP bền vững không yêu cầu bất kỳ RTTs trong quá trình tải đối tượng.

20 Mã 404 trong thông điệp trả lời từ Web server về trình duyệt có ý nghĩa là gì? Đối tượng client yêu cầu không có

1. Resource record trong DNS với type là MX dùng để làm gì?Dùng cho dịch vụ chuyển mail



a2, b3, c6, d5, e1, f4

TCP 20 : FTP (Data),TCP 22 : SSH,TCP 23 : Telnet,TCP 25 : SMTP,UDP 53 : DNS,UDP 67 : DHCP（Server）,UDP 68 : DHCP（Client）,TCP 80 : HTTP,TCP 110 : POP3,UDP 123 : NTP,TCP 443 : HTTPS,TCP 21 : FTP (Control)