Link trang trắc nghiệm:
https://docs.google.com/document/d/1ZC5EbuMUpCCieb65WAnelyNfvulV4x2R9B1b8kvjv4Y/edit?usp=sharing

Công thức:
EstimatedRTT = (1 - a) * EstimatedRTT_bef + a * SampleRTT
= 0.875*EstimatedRTT_bef + 0.125*SampleRTT
DevRTT = (1 - b) * DevRTT + b * |SampleRTT - EstimatedRTT|
= 0.75 * DevRTT_bef + 7 * |SampleRTT - EstimatedRTT_bef| / 32
TimeOutInterval = EstimatedRTT + 4 * DevRTT

TCP Throughput = 1.22 * Max_Segment_Size / (RTT * căn(Tỉ lệ mất gói))

Độ trễ = K*512/Tốc độ
Tỉ lệ = 1/(2 ^ (số lần đụng độ - 1))

Độ hiệu quả của CSMA/CD:
- Tprop là thời gian xử lý giữa 2 node.
- Ttrans là thời gian chuyển 1 max - size frame.

efficiency = 1 / (1 + 5 * Tprop / Ttrans)

Trong một subnet, địa chỉ lớn nhất và nhỏ nhất không thể gán cho máy tính.
Cách rút gọn 1 mạng IPv6: Dùng dấu :: rút gọn các đoạn địa chỉ chỉ gồm các số 0, tuy nhiên, chỉ rút các số 0 trước số khác 0 trong 1 nhóm. 1 địa chỉ chỉ được dùng dấu :: để rút gọn 1 lần.
Cách gửi tin:
- Broadcast: địa chỉ đích là địa chỉ lớn nhất của subnet.
- Bit cuối bộ đầu tiên trong địa chỉ IP là 1: multicast.
- Bit cuối bộ đầu tiên trong địa chỉ IP là 0: unicast.

Các lớp địa chỉ IP:
- Lớp A: sẽ dành riêng cho địa chỉ của các tổ chức lớn trên thế giới. Lớp A có địa chỉ từ 1.0.0.1 đến 126.0.0.0.
- Lớp B: sẽ dành cho tổ chức hạng trung trên thế giới. Lớp B có địa chỉ từ 128.1.0.0 đến 191.254.0.0
- Lớp C: được sử dụng trong các tổ chức nhỏ. Trong đó có cả máy tính cá nhân. Lớp C có địa chỉ từ 192.0.1.0 đến 223.255.254.0
- Lớp D: có 4 bit đầu tiên luôn là 1110. Đặc biệt lớp D được dành cho phát các thông tin (multicast/broadcast). Lớp này sẽ có địa chỉ từ 224.0.0.0 đến 239.255.255.255
- Lớp E: có 4 bit đầu tiên luôn là 1111. Lớp E được dành riêng cho việc nghiên cứu. Nó sẽ có địa chỉ từ 240.0.0.0 đến 254.255.255.255

Chương 2:
Khi viết 1 ứng dụng là viết 1 chương trình chạy ở tầng ứng dụng, trao đổi dữ liệu qua internet bằng cách dùng các dịch vụ được cung cấp bởi tầng bên dưới.
Kiến trúc có thể dùng cho ứng dụng:

- Client – Server: Server luôn online, có địa chỉ IP cố định, có hệ thống để cân bằng tải. Client không cần địa chỉ IP cố định, client không thể trao đổi lẫn nhau mà phải thông qua server.
- Peer – to – peer (P2P): mọi thành viên có vai trò như nhau gọi là peer, không cần online toàn thời gian, không cần địa chỉ IP cố định, cần có cơ chế nào đó để đảm bảo tài nguyên. Số lượng peer càng nhiều thì hệ thống tải càng tốt (self scalability) nhưng việc quản lý phức tạp.

Processes Communicating: đơn vị thực sự trao đổi dữ liệu. Quan hệ client – server hay P2P là việc cái process trao đổi message giữa các process với nhau.
Socket: phương tiện để process trao đổi dữ liệu.
Addressing Processes: địa chỉ để xác định process gửi hay nhận dữ liệu. Dùng port number để làm địa chỉ cho process và địa chỉ IP để xác định vị trí máy tính trên mạng.
Một số quy tắc của giao thức tầng ứng dụng:
- Loại thông điệp trao đổi: request, response.
- Cú pháp: gồm các trường dữ liệu gì, kích cỡ.
- Ngữ nghĩa các trường.
- Quy tắc process cần làm khi nhận hay gửi dữ liệu.

Khi viết ứng dụng sẽ sử dụng dịch vụ của tầng vận chuyển cung cấp: cần xét các đặc điểm của ứng dụng.
- Chấp nhận mất dữ liệu không không.
- Throughput lớn hay nhỏ.
- Độ trễ truyền dữ liệu là bao nhiêu.
- An ninh mạng: cần mã hoá dữ liệu không.

Dịch vụ của tầng vận chuyển:
- TCP: gửi dữ liệu tin cậy, điều khiển dòng lưu, không có bảo mật, tốc độ truyền có thể lâu.
- UDP: dữ liệu có thể bị mất.

An ninh mạng của TCP: có nhu cầu bổ xung security. Sử dụng SSL để mã hoá dữ liệu từ tầng ứng dụng trước khi đưa đến tầng vận chuyển. Đã được tích hợp sắn trong hạ tầng mạng.
Web và HTTP:
Mỗi trang web có địa chỉ là URL gồm 2 phần: giao thức (http://) để chỉ cách truy cập trang web, địa chỉ tài nguyên năm trên mạng gồm địa chỉ máy tính (hostname) và đường dẫn đến tài nguyên (path name).
HTTP (hypertext transfer protocol): chịu trách nhiệm tải nội dung cho một ứng dụng web. Gồm có HTTP request để gửi yêu cầu đến server và HTTP response để đưa câu trả lời từ server về client.
Giao thức HTTP sử dụng giao thức TCP trên port 80. Web browser sẽ mở đường TCP ở port 80 để gửi yêu cầu đến web server và chờ câu trả lời từ server đến cổng đó. Mỗi yêu cầu HTTP là độc lập với các yêu cầu khác (không có trạng thái). HTTP chỉ có nhiệm vụ vận chuyển yêu cầu, không ghi nhớ trước đó đã làm gì. Nếu xuất hiện việc nhớ trạng thái, đó là do server nhớ, không phải do HTTP.
HTTP có 2 cách sử dụng đường TCP bên dưới:
- Non – persistent HTTP: mỗi lần sử dụng TCP, chỉ gửi 1 đối tượng đi và về. Sau đó, server xoá đường đó. Khi có đối tượng mới thì sẽ cần kết nối lại đường khác. Thời gian gửi n đối tượng đi và về là n * (2RTT + thời gian truyền file). Thực tế server sẽ mở nhiều đường song song để down về cùng lúc nên thời gian sẽ nhỏ hơn.

- Persistent HTTP: duy trì một đường TCP từ browser đến server để gửi các đối tượng đi. Khi không cần sử dụng nữa mới xoá. Thời gian gửi n đối tượng đi và về là (n + 1) * RTT + thời gian truyền file. Nhưng vì chỉ có 1 đường TCP nên việc truyền sẽ truyền tuần tự, nên đôi khi nó sẽ chậm hơn non – persistent.

HTTP request message: Cấu trúc gồm request line (chứa các dòng lệnh GET/POST/HEAD, URL, version), header lines (chứa danh sách các header và dữ liệu), body (chứa nội dung được gửi đi).

Có 2 cách gửi dữ liệu lên server:

- POST method: dữ liệu được gửi qua body.
- URL method: sử dụng GET method, dữ liệu được gửi lên theo phần mở rộng của URL. Ví dụ 'wwww.abc.com/website?babyshark', phần mở rộng là sau dấu '?', đó là 'babyshark'. Nó sẽ đơn giản nhưng nó sẽ không được mã hoá và chỉ nên gửi các dữ liệu đơn giản.

Các loại method:

- HTTP/1.0: GET, POST, HEAD (chỉ lấy phần header của file).
- HTTP/1.1: thêm lệnh PUT (upload dữ liệu lên server), DELETE (để xoá dữ liệu trên server).

HTTP response message: Dữ liệu server chuyển về. Gồm static line (chứa phiên bản HTTP sử dụng và mã trạng thái trả về), header lines (chứa danh sách các header và dữ liệu), request HTML file (file dữ liệu được trả về cho server).

Các mã trạng thái: chia các mã trạng thái theo nhóm:

- Nhóm 200: OK.
- Nhóm 300: liên quan đến file trên server.
- Nhóm 400: liên quan đến request.
- Nhóm 500: liên quan phần mềm trên server.

User – Server State: sử dụng Cookies: Là một mẫu dữ liệu server lưu trữ trong browser của client, để lưu trữ trạng thái của client. User sẽ được web browser hỏi để cho phép server ghi trước khi lưu trạng thái.

- Khi server gửi lệnh response, nếu muốn dùng Cookie sẽ gửi kèm theo trong header line.
- Trong HTTP request tiếp theo được client gửi tới server sẽ kèm theo Cookie đó.

Web Caches: Bộ nhớ phụ để lưu trữ thông tin, thường đi kèm trong Proxy Server. Thường nằm trong hệ thống mạng trường học hay ISP nào đấy.

- Khi client gửi yêu cầu, thay vì đến server sẽ đến proxy server.
- Proxy server gửi yêu cầu đến server và nhận về câu trả lời từ server.
- Sau đó, proxy server gửi câu trả lời cho client ban đầu và vẫn lưu trữ lại câu trả lời đó.
- Nếu 1 client khác yêu cầu tương tự, proxy server sẽ gửi câu trả lời đã có về, giúp tiết kiệm thời gian.
- Nhờ đó tiết kiệm được băng thông và thời gian sử dụng.

Cache vẫn sẽ kiểm tra xem phiên bản trên server ở thời điểm hiện tại có thay đổi so với phiên bản đã lưu không. Nếu có thay đổi thì sẽ load lại.

Electronic Mail: gồm 3 đối tượng chính:

- User agents: web hay app cho user sử dụng để xem email.
- Mail servers: nơi lưu trữ email. Gồm mailbox (các hộp thư của mỗi user trên server), message queue (hàng đợi để gửi email trong server), giao thức SMTP (dùng để chuyển email giữa các server với nhau).

Giao thức SMTP: dựa trên giao thức TCP với port là 25. Bảng mã sử dụng là ASCII.

Nội dung của mail: gồm có header (gồm to, from, subject), body (nội dung của email).
Để đọc email cần:

- Dùng web browser sẽ dùng giao thức HTTP.
- Nếu dùng app trên PC hay mobile sẽ dùng:

• POP3: cần nhập username và password để xác định hộp thư ở đâu, tuy nhiên không có mã hoá nên có thể bị bắt gói tin, sau này sẽ có phần mềm để mã hoá như SSL, nhưng do có chế độ download và xóa, khi dùng thiết bị khác vào, có thể gây mất email hoặc chế độ không xoá thì nó sẽ không đánh dấu đã đọc hay chưa.

• IMAP: dùng phổ biến hơn hiện nay, mặc định lưu trữ email trên server, phiên bản đọc được chỉ là bản sao, khi xem chỉ download header, khi chọn thì mới download toàn bộ, ngoài ra còn có lưu trữ trạng thái trên server, từ đó đánh dấu được mail nào đã đọc hay chưa và tránh mất.

DNS: Domain Name System: hạ tầng cho các ứng dụng khác dùng tên miền. Ta đặt tên cho các địa chỉ IP (32 bits) bằng một tên miền dễ nhớ. DNS giúp chuyển từ tên miền dễ nhớ sang địa chỉ IP. Tầng ứng dụng sẽ dùng tên miền, sau đó dùng DNS chuyển thành địa chỉ IP cho các tầng từ tầng vận chuyển trở xuống sử dụng.
Các dịch vụ:

- Chuyển đổi tên miền thành địa chỉ IP.
- Đặt các nickname ngắn gọn cho các tên miền khi dùng nội bộ cho dễ sử dụng.
- Mail server sẽ tách biệt với web server, DNS sẽ giúp xác định mail server cần tìm.
- Cung cấp hệ thống cân bằng tải: cho phép nhiều địa chỉ IP dùng chung 1 domain name, DNS sẽ phân tán để chia đều truy cập lên nhiều hệ thống.

Cấu trúc hệ thống cơ sở dữ liệu DNS: là hệ thống phân tán. Gốc (root name server) chia làm nhiều nhánh, nhánh gốc tương ứng một đuôi (com, org, edu), mỗi nhánh lại chia thêm theo tên miền (yahoo.com, amazon.com).
Root name server: hiện có 13 cái trên thế giới. Thông thường, người ta ít truy cập root mà sẽ truy cập ở tầng thứ 2 (top – level domain TLD) dự theo đuối của tên miền (com, org, edu, …).
Authoritative DNS Servers: server của các tổ chức, nơi trực tiếp quản lý địa chỉ IP của tổ chức đó.
Local DNS Name Server: hoạt động thường xuyên, số lượng nhiều nhất. Là địa điểm đầu tiên để máy tính tìm địa chỉ IP.

- Máy sẽ hỏi server này trước, nó có trả lời ngay từ dữ liệu trong cache, nếu không có sẽ tìm từ root và lưu câu trả lời vào cache, sau đó trả lời cho lời yêu cầu.
- Do nhớ để trả lời nên có thể sai.
- Có 2 cách hỏi từ Local đến root (tăng tải của root) là hỏi lặp và hỏi đệ quy (tăng tải của local).
- Dữ liệu lưu trong cache của local sẽ có TTL (time to life), khi hết thời gian sẽ tự động xoá.
- Cấu trúc sẽ gồm 4 phần (name, value, type, TTL).

Các type khi lưu trên DNS:

- A (mặc định): name là hostname, value là địa chỉ IP.
- NS: name là là domain name, value là hostname của server quản lý domain name.
- CNAME: name là tên khác của domain name nào đấy, value là tên gốc.
- MX: name là domain name, value là mail server của domain name.

Peer – to – peer application:
Ứng dụng phổ biến: BitTorrent, Streaming, VoIP, …

Khi có N máy cần file từ server, ở client – server, server cần upload N file nên thời gian sẽ tăng nếu N tăng, và tăng tuyến tính.
Nếu ở cấu trúc P2P, server chỉ cần upload 1 file, sau đó các peer chia sẻ nhau, càng nhiều peer tham gia thì thời gian vẫn không tăng mà sẽ dần tiến về 1 hằng số.
Ứng dụng BitTorrent: trong 1 thời điểm, 1 peer sẽ chọn 4 peer có tốc độ download nhanh nhất để down. Sau mỗi 30s, peer sẽ bắt đầu tìm loại 1 peer tốt nhất, nếu có cập nhật thì sẽ xoá đi 1 peer thấp nhất hiện có.

Video Streaming và mạng lưới CDN:

Dùng kiến trúc CDN để lưu trữ, tránh gây tắc nghẽn do quá nhiều truy cập.
Dùng nén những điểm giống nhau trong cùng 1 frame và nén các frame liên tiếp giống nhau để giảm băng thông.
Tải dữ liệu có thể theo tốc độ cố định (CBR – constant bit rate) hay tốc độ thay đổi (VBR – variable bit rate). Khi thay đổi tốc độ truyền dữ liệu thì chất lượng hình ảnh sẽ thay đổi theo.
Giao thức phổ biến cho streaming là DASH (Dynamic, Adaptive Streaming over HTTP): dùng HTTP để gửi dữ liệu.
Content Distribution Networks (CDN): hệ thống phân phối nội dung trên mạng. Nó sẽ rải rác phân tán dữ liệu trên các server. Lưu trữ càng gần người sử dụng càng tốt.
Vấn đề OTT (Over The Top): Tách rời hệ thống lưu trữ khỏi ISP.
Khi ta yêu cầu dữ liệu từ server, server sẽ xác định CDN server chứa dữ liệu yêu cầu mà gần client nhất và trả về địa chỉ server đấy.

Socket Programming:

Các ngôn ngữ lập trình đều có API để kiểm soát Socket.

Chương 3:

Tầng ứng dụng sẽ chia gói tin thành các segment để đưa xuống tầng vận chuyển. Hai giao thức dùng phổ biến ở tầng vận chuyển là TCP và UDP.
So sánh tầng mạng và tầng vận chuyển:

- Tầng mạng: tạo ra kênh để kết nối giữa các máy với nhau.
- Tầng vận chuyển: cung cấp kênh để kết nối giữa các process với nhau.

Multiplexing: quá trình tầng vận chuyển gom các gói tin từ các process ở tầng ứng dụng để gửi vào mạng.
Demultiplexing: quá trình tầng vận chuyển phân các gói tin nhận từ mạng về đúng process tương ứng. Cách phân gói tin dựa theo source port và dest port ở header của gói tin. Về phần địa chỉ IP sẽ xử lý ở tầng mạng.
Cơ chế tương tác giữa các socket:

- Đối với UDP: để gửi nhận dữ liệu, chỉ cần biết địa chỉ IP và port là có thể gửi dữ liệu tới 1 socket.
- Đối với TCP: mỗi socket chỉ được nhận và gửi dữ liệu với một socket khác. Do đó, chúng ta cần biết source port, source IP, dest port, dest IP. Khi đủ 4 trường thông tin, tầng vận chuyển sẽ chuyển gói tin đến đúng nơi.
- Đối với web server, nó chỉ có 1 IP, 1 port. Nó sẽ chỉ chạy 1 process nhưng gán với nhiều socket bằng cơ chế đa luồng.

UDP: người ta sẽ dùng UDP và thêm cơ chế kiểm soát lỗi ở tầng ứng dụng.
Header bao gồm: source port, dest port, length, checksum.
Tại sao dùng UDP: Ít overhead hơn, giúp truyền nhanh hơn, dùng đơn giản. Không có kiểm soát tắc nghẽn nên cũng chạy nhanh hơn. Về hiệu suất UDP tốt hơn TCP nên sử dụng.

Checksum: dùng để kiểm tra lỗi. Bên gửi tính checksum từ dữ liệu và gửi đi, bên nhận nhận về xem kiểm tra lại. Cách tích: tích body thành các bộ 16 bits rồi cộng lại, sau đó lấy 16 bits cuối kết quả + 1 và đảo bit của nó để được checksum.

Cơ chế giải quyết mất gói:

- Bên nhận nhận gói tin, nếu checksum bị lỗi, bên nhận sẽ gửi về NAKs, nếu gói tin không lỗi, gửi về ACKs.
- Nếu ACKs, bên gửi gửi gói tin tiếp theo. Nếu NAKs, bên gửi gửi lại cho đến khi không còn lỗi.
- Bên nhận thì sẽ chờ cho đến khi nhận gói tin không lỗi thì tiếp tục.
- Nếu gói tin xác nhận NAK/ACK bị lỗi không phân biệt được, gói tin sẽ tự động xem là NAK và gửi lại.
- Nếu có 2 gói tin trùng nhau nhờ việc đánh số thứ tự, bên nhận sẽ huỷ 1 gói.
- Nếu bên gửi gửi tin đi mà sau một khoảng thời gian không thấy phản hồi thì sẽ xem như mất gói và gửi lại gói tin.
- Hiệu suất gửi gói tin đi = độ dài gói tin / (độ dài gói tin + tốc độ truyền * RTT)

Giao thức pipelined: gửi nhiều gói tin một lúc để tăng tốc độ gửi gói tin. Nó sẽ cho phép gửi trước vài gói tin theo 2 cơ chế:

- Go – back – N: gửi trước N gói tin. Bên nhận được gửi xác nhận gộp nhiều gói tin cùng lúc. Chỉ cần 1 timer để nhớ gói tin được gửi lâu nhất chưa được xác nhận. Khi timmer hết thì gói tin nào chưa được xác nhận sẽ được gửi lại.
- Selective Repeat: gửi trước N gói tin. Nhận được gói tin nào sẽ xác nhận từng gói tin riêng lẻ. Từng gói tin có từng timer riêng. Khi timeout thì chỉ gửi lại các gói chưa xác nhận.

TCP:

Gồm các thuộc tính:

- Point – to – point: 1 gửi, 1 nhận.
- Đảm bảo gói tin toàn vẹn.
- Sử dụng pipelined.
- Full duplex data: cho phép gửi nhận đồng thời.
- Cần thiết lập kết nối trước khi gửi.
- Có điều khiển luồng.

Sequence numbers: số thứ tự của segment = số thứ tự byte đầu tiên của segment.

Acknowledgements number: số thứ tự byte kế tiếp bên nhận sẽ nhận = sequence number kế tiếp.

Tính RTT và Timeout: SampleRTT (RTT thực tế), EstimatedRTT (RTT ước lượng).

RTT ước lượng: EstimatedRTT = (1 – a) * EstimatedRTT_before + a * SampleRTT (a thường lấy 0.125)

DevRTT = (1 - b) * DevRTT_before + b * | SampleRTT – EstimatedRTT | (b thường lấy 0.25)

TimeoutInterval = EstimatedRTT + 4 * DevRTT

Dấu hiệu nhận biết mất gói: Timeout, nhận nhiều ACKs giống nhau.

Flow Control: TCP sẽ có một buffer để kiểm soát tránh tắc nghẽn.

Cơ chế kiểm soát tắc nghẽn:

Lượng dữ liệu gửi đi sẽ lớn hơn hoặc bằng lượng dữ liệu nhận được.

Khi một router bị quá tải, các router liên quan cũng có thể bị quá tải theo.

Khi đạt đến giới hạn tắc nghẽn, việc gửi nhận dữ liệu gần như bị dừng.

Cách TCP kiểm soát tắc nghẽn:

Khi phát hiệu dấu hiệu tắc nghẽn, TCP sẽ giảm tốc độ gửi dữ liệu đi.

Cơ chế hoạt động:

- Tăng theo cấp số cộng (additive increase): sau mỗi RTT sẽ tăng cwnd (congestion windows – chịu trách nhiệm điều khiển dòng dữ liệu điều khiển việc tắc nghẽn) 1 đơn vị.
- Giảm theo cấp số nhân (multiplicative decrease): giảm cwnd đi 1 nữa nếu gặp mất gói.

Tốc độ truyền dữ liệu ~ cwnd / RTT (bytes/sec) = cwnd_max * ¾ (điều kiện các điểm tắc nghẽn như nhau).

Cơ chế Slow Start:

- Khi bắt đầu, tăng theo cấp số nhân: bắt đầu đặt cwnd bằng 1, sau mỗi RTT tăng gấp 2 lần.
- Khi có dấu hiệu tắc nghẽn (mất gói, timeout): đặt cwnd bằng 1, sau đó tăng gấp đôi mỗi RTT, đến khi gặp threshold (giá trị bằng 1 nửa cwnd lúc trước khi tắc nghẽn) thì mỗi RTT cộng 1 đơn vị.
- Đối với TCP RENO, khi cặp 3 ACKs giống nhau sẽ giảm cwnd xuống 1 nửa sau đó tăng theo cấp số cộng.
- Đối với TCP Tahoe, khi gặp mất gói sẽ đặt cwnd về 1 và tăng theo cấp số nhân đến khi gặp threshold thì tăng theo cấp số cộng.

TCP Throughput: nếu chỉ có 1 đường truyền và chỉ có 1 luồng TCP đi qua, các điểm tắc nghẽn sẽ luôn như nhau. Throughput của đường truyền sẽ là ¾ băng thông của đường truyền.

Ví dụ: Có gói tin 1500 bytes, 100ms RTT, kỳ vọng throughput 10Gbps. Giả sử windows size W = 83 333.

TCP Throughput = 1.22 * Max_Segment_Size / (RTT * căn(L)) L là tỉ lệ mất gói.

Để đạt 10Gbps throughput, cần L = 2 * 10-10 – rất nhỏ.

TCP Fairness: Nếu kênh truyền có K luồng TCP đi qua, không quan trọng TCP nào đi trước, khi hệ thống chuyển sang trạng thái ổn định, mỗi luồng sẽ chiếm R/K băng thông đường truyền.

Nếu 1 luồng vừa có UDP vừa có TCP, do UDP không kiểm soát tắc nghẽn, khi tắc nghẽn TCP sẽ giảm, UDP vẫn tăng, đến 1 lúc nào đó UDP sẽ chiếm hết đường truyền. Do đó ta có thể xem UDP có tốc độ truyền cao hơn.

Explicit Congestion Notification (ECN): cơ chế hỗ trợ từ hạ tầng mạng, gói tin TCP gửi đi sẽ ghi nhận thêm tình trạng hạ tầng mạng trên đường đi để tự động kiểm soát việc gửi nhận.

Chương 4:

Tầng mạng (Network Layer):

- Tầng ứng dụng và tầng vận chuyển chỉ xuất hiện trên các end system.
- Tầng mạng có cả ở các router thuộc hạ tầng mạng.
- Tầng vận chuyển sẽ vận chuyển các segment. Ở tầng mạng, các segment được đóng gói trong datagram, gọi là packet để vận chuyển đi.

Chức năng tầng mạng:

- forwarding: chịu trách nhiệm chuyển packet từ cổng vào đến cổng ra phù hợp.
- routing: tìm đường đi tối ưu cho gói tin trên mạng máy tính. Sử dụng routing algorithms (giải thuật định tuyến).

Trong router gồm 2 phần: data plane và control plane.

Data plane: nằm gọn trong router, chịu trách nhiệm chức năng forwarding.

Control plane: tập trung điều khiển, tính toán đường đi tối ưu nhất. Chạy các giải thuật định tuyến. Router thế hệ mới kèm theo software – defined networking (SDN) sẽ là các server để kiểm soát hệ thống router.

Về các dịch vụ của tầng mạng: Tầng mạng không đảm bảo truyền gói tin đảm bảo, đảm bảo độ trễ, truyền dữ liệu theo trật tự, đảm bảo băng thông tối thiểu. Ở mô hình ATM thì đảm bảo nhưng không dùng trong Internet.

Cấu trúc Router:

Cấu trúc cổng vào gồm bộ phần kết nối đường dây, bộ phân xử lý giao thức tầng liên kết dữ liệu, hàng đợi của các gói tin. Khi xử lý gói tin, router xác định đích đến của gói tin. Sau đó hệ thống chuyển mạch sẽ đưa gói tin đến đúng cổng ra.

Cách lưu địa chỉ trong router: Các địa chỉ có phần đầu trong 32 bits giống nhau sẽ gom thành từng nhóm, từ đó phân chia ra các cổng. Trong trường hợp có 2 cổng khớp, cổng nào có số bit giống nhiều hơn thì sẽ ra cổng đó.

Bộ chuyển mạch: chuyển 1 gói tin từ cổng vào đến cổng ra. Gồm 3 cơ chế phổ biến:

- Thông qua Memory: Gói tin từ cổng vào chuyển vào bộ nhớ chỉnh của router, bộ xử lý tính toán xác định cổng ra và đưa từ bộ nhớ ra cổng ra tương ứng. Tốn 2 bus nên tốn thời gian hơn.
- Thông qua Bus: Có một bus kết nối các cổng vào và cổng ra. Khi bus lên, gói tin vào, khi bus xuống, gói tin ra. Tốc độ chuyển dữ liệu phụ thuộc tốc độ bus. Hệ thống router của Cisco 5600 có thể lên tới 32 Gbps.
- Thông qua Interconnection Network: Có hệ thống phức tạo kết nối cổng vào với tất cả các cổng ra theo dạng như bàn cờ. Các gói tin có thể đi đồng thời nên tăng tốc độ đáng kể. Hệ thống Cisco 12000 có tốc độ 60 Gbps.

Kích cỡ buffer = RTT * khả năng của đường truyền / căn(số dòng dữ liệu đi qua)

Giao thức IP: dùng để định dạng gói tin đi lại trên mạng.

Ở tầng mạng ngoài giao thức IP còn có giao thức ICMP.

Cấu trúc gói tin IP:

Header: 32 bits IP nguồn và IP đích. Đối với TCP, header là 20 bytes. Đối với IP thêm 20 bytes header. Vậy đến tầng mạng độ dài gói tên ban đầu sẽ tăng thêm 40 bytes.

Header còn chứa Time To Life, là số lượng router tối đa cho phép gói tin đi qua. Khi TTL = 0, gói tin sẽ bị xoá để tránh tình trạng gói tin lạc đường tồn tại mãi trên mạng.

Fragmentation:

Khi gói tin lớn đi tới một hạ tầng mạng có MTU (max transfer size) nhỏ hơn, gói tin sẽ cần bị phân mảnh ra để đưa đi. Sau đó ghép lại gói tin khi cần thiết.

Các fragment có cùng 1 ID là của cùng 1 gói tin. Fragflag = 1 để đánh dấu còn gói tin sau nó, fragflag = 0 để đánh dấu không còn gói tin sau nó. Offset là số thứ tự byte đầu tiên fragment chứa chia cho 8, sẽ dùng sắp xếp thứ tự các gói tin.

Các fragment bị phân ra, với mỗi fragment tăng thêm sẽ có thêm 20 bytes header.

Địa chỉ IP:

IPv4 là 32 bits biểu diễn bằng 4 cụm 8 bits đại diện cho 4 giá trị từ 0 đến 255.

Subnet: bộ các thiết bị có địa chỉ phần đầu giống nhau. Các máy trong cùng 1 subnet có thể trao đổi dữ liệu trực tiếp với nhau, không cần đi xuyên qua router.

Ta cần gán địa chỉ subnet để hoạt động theo cơ chế a.b.c.d/x với x là để đánh dấu mask của subnet.

Mask là dãy địa chỉ gồm x số 1 ở bên trái, còn lại là số 0. Phần x cũng để đánh dấu phần địa chỉ của subnet.

Cách cấu hình subnet: dùng giao thức DHCP (Dynamic Host Configuration Protocol) từ DHCP server để tự động cung cấp các địa chỉ IP trong subnet cho các client.

ISPs sẽ có 1 dãy địa chỉ, sau đó ISPs để tăng subnet mask lên để tạo 1 vùng địa chỉ IP khác, cung cấp cho tổ chức nào đó để cấp phát địa chỉ IP.

Cách tính số địa chỉ IP trong 1 subnet: subnet có dạng a.b.c.d/x, số địa chỉ trong subnet = 2 ^ (32 - x).
Tổ chức ICANN (Internet Corporation for Assigned Names and Numbers) quản lý hệ thống địa chỉ IP, DNS.
Network Address Translation (NAT): do địa chỉ IP đã dùng hết, xuất hiện nhu cầu tạo các local IP address để dùng cục bộ, không công nhận trong địa chỉ IP công cộng.
Cơ chế NAT nằm trong các router, có nhiệm vụ đổi địa chỉ để gửi các gói tin trong mạng nội bộ ra mạng công cộng.
Khi gói tin đến router, nó sẽ thay địa chỉ nguồn bằng địa chỉ router, thay port bằng một số ID nào đó. Sau đó lưu thông tin địa chỉ nội bộ vào 1 bảng của NAT.
Khi gói tin trả lời gửi về, router sẽ dùng bảng tra trước đó từ số ID để xác định địa chỉ và port của máy đích.
Port number có 16 bits, vậy ta có thể hỗ trợ gần 216 thiết bị. Tuy nhiên, việc dùng NAT sẽ là tầng network mở gói tin của tầng transport ra để chỉnh sửa, vi phạm quy tắc "tầng nào xử lý dữ liệu của tầng đó". Do đó, nếu không đủ địa chỉ có thể sử dụng địa chỉ IPv6 để thay thế, tránh vi phạm quy tắc.
Nếu ta có server trong mạng cục bộ mà muốn server chạy liên tục, thay vì để NAT cấu hình tự động địa chỉ IP, ta có thể thiết lập sẵn bên trong bảng của hệ thống NAT.
IPv6: Làm đơn giản header để tăng tốc độ xử lý. Thêm các trường thông tin để đảm bảo chất lượng dịch vụ.
Cấu trúc: Header cố định chiều dài là 40 bytes, không có phần option như IPv4. Không cho phép gói tin bị phân mảnh.
Địa chỉ IPv6 dài 128 bits.
Header chứa địa chỉ nguồn và đích, flow label (dùng để phân loại gói tin, kiểm soát chất lượng dịch vụ tốt hơn), version, traffic class, payload length, hop limit (thay thế cho Time to life), next hdr (trường header thêm thông tin).
So với IPv4: bỏ checksum, dùng giao thức ICMPv6 thay cho ICMP để kiểm tra lỗi, xử lý các sự cố trên đường truyền.
Vận chuyển IPv6 trong IPv4: Dùng giải pháp tunnel link: Khi một gói tin IPv6 đi qua một vùng IPv4, toàn bộ gói tin IPv6 sẽ được bỏ trong một gói tin IPv4 để truyền đi, khi chuyển về mạng IPv6, gói tin sẽ được đem ra để truyền đi tiếp.
OpenFlow: thay vì forwarding thông thường, mở rộng thêm một số quy tắc để tìm đường đi.
src = 1.2.*.*, dest = 3.4.5.* drop (xoá gói tin)
src = *.*.*.*, dest = 3.4.*.* forward(2) (chuyển sang cổng số 2)
src = 10.1.2.3, dest = *.*.*.* send to controller.
Ta sẽ so sánh các trường trong header so với luật. Nếu khớp một trường nào đó sẽ thực hiện hành động tương ứng.
OpenFlow có thể có ở cả Router và Switch. Nếu ở Router sẽ tập trung vào phần đầu của địa chỉ IP đích. Nếu ở Switch sẽ tập trung vào địa chỉ MAC đích. Đối với Firewall và NAT sẽ tập trung vào địa chỉ IP và port.

Chương 5:
Có 2 hướng tiếp cận trong control plane:
- Per – router control (truyền thống): mọi quyết định đều do router.
- Logically centralized control (dựa theo SDN – software defined networking): có các centralized server có các controller tập trung. Controller quyết định và cập nhật dữ liệu xuống router.

Giải thuật định tuyến chia làm 2 loại dựa theo cách thức hoạt động giải thuật:

- Global: khi cần thông tin tổng thể của mạng. Có giải thuật “link state” cần bức tranh tổng thể của mạng máy tính.
- Decentralized: khi mỗi router tự tương tác với nhau. Có giải thuật “distance vector”, không cần bức tranh tổng thể của mạng máy tính.

Ngoài ra còn có cách chia:

- Static: ít thay đổi. Cấu hình trực tiếp.
- Dynamic: thường xuyên thay đổi. Các router cập nhật thường xuyên.

Giải thuật link – state: từ đồ thị mạng, áp dụng giải thuật Dijkstra. Các bước:

- Tìm đồ thị mạng bằng cách gửi thông tin broadcast để kết nối đến các router lân cận, từ đó có được thông tin vị trí các router.
- Khi có đồ thị thì chạy giải thuật tìm đường đi tối ưu.
- Sau khi có kết quả, cập nhật vào forwarding table để chạy.

Giải thuật Distance Vector: dùng công thức bellman – ford để tìm đường đi ngắn nhất.

$$dx(y) = \min \{c(x, v) + dv(y)\}$$

Khoảng cách từ x đến y = số nhỏ nhất trong tổng chi phí từ x đến v + khoảng cách từ v đến y.

Mỗi router sẽ trao đổi vector khoảng cách với các router xung quanh, từ đó tính được khoảng cách.

Mạng máy tính chia làm nhiều khu vực gọi là **administrative autonomy**, mỗi khu vực được một tổ chức nào đó quản lý. Tổ chức đó cũng sẽ quyết định giải thuật sử dụng để tìm đường đi trên mạng là link – state hay distance vector.

Các giải thuật sẽ được chạy trong một mạng nội bộ gọi là AS (autonomous systems).

Interconnected ASes:

- Khi tìm đường đi trong cùng 1 AS, ta dùng intra – AS routing.
- Khi tìm giữa các mạng AS khác nhau, ta dùng inter – AS (để xác định gateway) và intra – AS (xác định đường đi ngắn nhất đến gateway) routing.

Intra – AS routing: sử dụng các giải thuật khác nhau trong nhóm interior gateway protocols (IGP):

- RIP: Routing Information Protocol, dựa trên distance vector.
- OSPF: Open Shortest Path First, dựa trên link state.
- IGRP: Interior Gateway Routing Protocol, giải thuật riêng của Cisco.

OSPF: Miễn phí. Sử dụng link – state.

Tạo cái broadcast là gói tin IP, gửi đến toàn bộ các router trong AS. Mỗi router tự tính toán đường đi tối ưu.

Giải thuật cung cấp một số tính năng khác: security, cho phép có nhiều đường đi khác nhau nếu chi phí bằng nhau (RIP chỉ có 1 đường).

Có hỗ trợ cơ chế multicast (gửi 1 gói tin đến 1 nhóm các thiết bị).

OSPF có thể phân cấp ra chạy các nhóm nhỏ giúp giảm thời gian chờ.

Inter – AS:

- Xác định các vị trí đi đến các AS khác.
- Thông báo đến toàn bộ router trong hệ thống.

Internet inter – AS routing: dùng BGP (Border Gateway Protocol), giải thuật này kết nối internet lại với nhau. BGP gồm 2 nhóm:

- eBGP: tìm thông tin về đường đi ra ngoài để đến các AS khác.
- iBGP: truyền thông tin tìm được từ eBGP cho các router bên trong mạng.
- Xác định đường đi tốt nhất dựa trên chính sách (policy): nhiều khi đường đi ngắn nhất chưa phải tốt nhất.

BGP còn giúp báo ra internet về sự tồn tại của mạng AS này.

Các router trong BGP trao đổi với nhau thông qua các kết nối TCP.
Một router trong AS này kết nối với router trong AS khác. Sau đó, router thông báo về khả năng kết nối đến 1 router khác trong AS. Từ đó báo với router bên AS kia rằng router này hứa sẽ chuyển gói tin đến router đã thông báo trước đó.
Thông tin trao đổi bao gồm prefix (địa chỉ của subnet) và attribute (một số thuộc tính liên quan đường đi đó).
Các attribute:
- AS – PATH: mô tả đường đi của gói tin đến đích như thế nào, qua các AS nào.
- NEXT – HOP: router nội bộ AS nào đi đến được AS kế.

Nếu router nhận được nhiều đường đi khác nhau để đến địch từ các router khác. Việc quyết định dựa trên yếu tố:
- Chính sách: khách hàng thuê ISP để sử dụng mạng, ISP không được gửi tin thông qua khách hàng; ISP chỉ gửi tin đến các khách hàng của mình; …
- AS – PATH ngắn nhất, đường đi qua ít trung gian nhất.
- NEXT – HOP router gần nhất, dựa trên Hot Potato Routing.
- Một số tiêu chí khác.

Hot Potato Routing: Tìm quãng đường ngắn nhất để gói tin đi ra khỏi mạng, không quan tâm quãng đường ngắn nhất.
Ở Intra – AS, tập trung sao cho chuyển gói tin hiệu quả nhất. Đối với Inter – AS, tập trung vào chính sách.
Software Defined Networking (SDN):
Thay vì để các router tìm đường đi một cách độc lập, dùng phần mềm để điều khiển. Có một controller điều khiển toàn bộ router, sau khi tìm đường sẽ gửi thông tin về các router. Giúp:
- Đơn giản hoá việc quản lý, tránh sai sót, kiểm soát dòng truyền tốt hơn.
- Cho phép lập trình dễ dàng hơn.
- Giao thức mở, không thuộc tổ chức nào.

Một số vấn đề: mức độ tin cậy, quy mô tăng nên thì khả năng mở rộng ra sao, bảo mật hệ thống.
Giao thức ICMP: đi cùng với giao thức IP, có nhiệm vụ kiểm tra hiện trạng mạng. Nằm ở tầng mạng nhưng sử dụng gói tin IP.
Các gói tin IP gửi đi sẽ có ICMP type và code để xác định các mã lệnh sử dụng.
Traceroute hoạt động dựa trên ICMP: nó sẽ gửi các gói tin IP đến các router để tìm đường đi. Nó sẽ gửi các gói tin có TTL tăng dần từ 1, mỗi TTL chạy 3 lần. Từ đó xác định độ trễ đến các router để tìm đường đi.

Chương 6:
Cấu trúc tầng liên kết dữ liệu:
- Các host, router gọi là node.
- Các node kết nối, trao đổi với nhau thông qua đường gọi là link (liên kết dây, không dây, LANs).
- Các gói tin trao đổi giữa các node gọi là frame, chứa các datagram trong đó.

Nhiệm vụ của tầng liên kết dữ liệu: truyền dữ liệu giữa 2 node kế cận nhau.
Quá trình framing:
- Đóng gói datagram trong frame, thêm header và trailer.
- Nếu dùng chung kênh truyền, sẽ điều phối kênh truyền đấy.
- Cung cấp địa chỉ MAC để xác định vị trí của thiết bị trong tầng link.

Cung cấp thêm cơ chế kiểm soát lỗi, truyền dữ liệu đáng tin cậy.

Một số dịch vụ hỗ trợ: flow control, kiểm tra lỗi, sửa lỗi, half – duplex (dữ liệu đi được 2 chiều nhưng mỗi thời điểm chỉ đi được 1 chiều) và full – duplex (có thể đi 2 chiều trong 1 thời điểm).
Đối với 3 tầng trên, xử lý ở memory và CPU. Link layer một phần xử lý ở CPU, một phần xử lý ở network interface.
Phát hiện lỗi: Khi gửi đi dữ liệu D, kèm theo đó là EDC (Error Detection and Correction bits) để kiểm tra lỗi (thường dùng checksum).
Không có cơ chế nào đảm bảo đúng 100%. Các giải thuật chỉ để hạn chế tối đa các trường hợp lỗi.
Parity Checking: mỗi hàng, cột trong dữ liệu đánh dấu thêm bit 0 hoặc 1 cho chẳng hoặc lẻ, sau đó khi kiểm tra sẽ thấy được vị trí chạm để chỉnh sửa bit lại. Không khả thi khi có 2 – 4 bits cùng lúc.
Cyclic Redundancy Check: dùng nhiều. Có thể phát hiện được cả một cụm bit bị lỗi.
Multiple Access Links: giải quyết vấn đề một kênh truyền có nhiều thiết bị truy cập vào. Nếu nhiều thiết bị va chạm lẫn nhau, tính hiệu của tất cả sẽ bị lỗi.
Multiple access protocol sẽ điều phối lại.
Đặc điểm kênh truyền lý tưởng: giả sử có kênh truyền tốc độ R bps.
- Hiệu suất: 100%. Nếu chỉ có 1 node, node gửi tốc độ R. Khi có M node, mỗi node gửi tốc độ R/M.
- Kiến trúc: Ổn định. Giao thức phải chạy hoàn toàn là phi tập trung, tránh trường hợp một node chết gây chết node còn lại, giải quyết vấn đề có 2 node còn thì hệ thống còn ổn định.
- Giao thức: Đơn giản.

Trong thực tế chỉ được 2 trong 3.
Chia 3 nhóm:
- Chia tĩnh: đơn giản. Nếu có N user dùng 1 kênh truyền thì chia kênh lớn thành N kênh nhỏ. Hiệu suất thấp. Dùng nhiều trong hệ thống mạng di động, DSM.
- Random access: ai dùng cứ dùng, lúc có va chạm thì giải quyết sau. Phức tạp nhưng hiệu suất cao. Dùng trong mạng WiFi, mạng internet.
- Điều phối (Taking Turn): thiết bị nào đc gửi và nhận. Đơn giản, hiệu suất cao, có thể bị overhead. Dùng trong bluetooth, hệ thống cáp quang, …

Cách chia tĩnh (channel partitioning MAC protocols):
- TDMA: chia theo thời gian, mỗi user dùng trong 1 khoảng nhất định.
- FDMA: chia theo tần số.

Random Access Protocols: Khi một node có nhu cầu gửi dữ liệu thì cứ gửi. Khi 2 node gửi cùng lúc sẽ có va chạm.
Giải thuật tập trung việc phát hiện và giải quyết va chạm.
Gồm Slotted ALOHA và CSMA.
Slotted ALOHA: Tất cả frame cùng size, chia ra các khoảng thời gian vừa đủ gửi 1 frame. Một khoảng chỉ được gửi 1 frame. Các node đồng bộ với nhau. Nếu có 2 node truyền cùng lúc thì va chạm xảy ra.
Khi đến khoảng thời gian: Nếu không va chạm, gửi frame đi. Nếu va chạm, đợi đến ô kế và gửi với 1 xác suất p. Vào các khoảng thời gian trống, mỗi gói sẽ gieo xác suất, nếu p = 1 thì gửi gói đó đi.
Hiệu quả: Tính được xác xuất 1 gói được gửi thành công trong N gói là NP (1 - P) N - 1.
Hiệu suất chỉ 37%.

Carrier Sense Multiple Access – CSMA: node muốn gửi dữ liệu sẽ lắng nghe kênh truyền để xem có tín hiệu hoạt động thì không gửi, tránh gây nhiễu. Xác suất va chạm giảm những vẫn xảy ra.
CSMA/CD (Collision Detection): phát hiện va chạm trong lúc gửi và dừng gửi, tránh lãng phí dữ liệu.
Độ trễ = K*512/Tốc độ
Tỉ lệ = 1/(2 ^ (số lần đụng độ - 1))
Taking Turn MAC Protocols: gồm các dạng:
- Mô hình polling: có 1 master điều phối, ai muốn gửi sẽ gửi yêu cầu tới master. Mất thời gian để tương tác với master. Master chết thì cả hệ thống dừng.
- Token passing: mạng tổ chức theo vòng tròn, có 1 token trong mạng, node nào giữa token sẽ được gửi. Sau khi gửi thì chuyển token. Mất thời gian nếu một máy không cần gửi cũng phải nhận token. Độ trễ lớn. Máy giữ token chết thì cả hệ thống dừng (single point of failure).

MAC address và ARP: 48 bits biểu diễn bằng 6 cụm số hex, mỗi cụm 8 bits. Gắn trực tiếp vào ROM của card mạng.
Địa chỉ MAC quản lý bởi IEEE.
ARP (Address Resolution Protocol):
Tầng mạng yêu cầu hoạt động theo địa chỉ IP, tầng data link hoạt động theo địa chỉ MAC. Từ đó sinh yêu cầu tìm địa chỉ IP từ địa chỉ MAC.
ARP sẽ chịu trách nhiệm lưu địa chỉ IP, địa chỉ MAC tương ứng, TTL là thời gian sống của dòng thông tin này (thường là 20 phút).
A muốn tìm địa chỉ MAC của B. A sẽ tạo ARP query packet, gửi broadcast đến tất cả máy trong mạng LAN (địa chỉ MAC đích là FF – …– FF).
B nhận được ARP packet và gửi lại địa chỉ MAC cho A.
A sau khi nhận được dữ liệu sẽ lưu địa chỉ IP, địa chỉ MAC, TTL vào ARP table.
Để kiểm tra cả 2 có cùng LAN hay không, ta kiểm tra mask của subnet có giống nhau không.
Khi gửi qua LAN khác:
- A gửi broadcast.
- Router sẽ nhận, lấy IP địa chỉ đích và xác định cổng đến được B.
- Sau đó router tiếp tục chuyển dataframe từ cổng của LAN chứa A sang cổng đến được B.
- Tiếp đó truyền tiếp cho đến khi gặp B. Gói tin lúc này sẽ có MAC nguồn là địa chỉ router trung gian.
- B nhận được, lấy địa chỉ của A từ gói tin và gửi ngược lại thông tin cho A.

Ethernet: dùng cơ chế CSMA/CD.
Ngày xưa sử dụng mô hình bus: các máy nối chung vào một dây.
Hiện nay sử dụng mô hình star: một switch ở giữa mắt tới các máy. Nhờ có switch giải quyết va chạm nên tốc độ nhanh hơn đáng kể.
Cơ chế CSMA/CD của ethernet:
- Khi card mạng nhận gói tin từ tầng network, sẽ tạo frame.
- Network lắng nghe kênh truyền. Nếu kênh truyền trống sẽ gửi đi. Nếu kênh truyền bận sẽ đợi đến khi trống thì gửi đi.
- Nếu card mạng gửi frame mà không bị lỗi, va chạm thì thành công.
- Nếu phát hiện lỗi hay va chạm thì sẽ huỷ và gửi đi một jam signal để báo các thiết bị trong LAN đang có va chạm, nhằm dừng truyền tải tránh va chạm.

- Sau khi huỷ, card mạng chuyển sang trạng thái binary (exponential) backoff: Sau mỗi va chạm thứ m, card mạng chọn gái trị K ngẫu nhiên từ 0 đến 2n – 1. Card mạng chờ K * 512 bits times (thời gian gửi 1 bits dữ liệu), sau đó đưa về bước 2.

Độ hiệu quả của CSMA/CD:

Cho Tprop là thời gian xử lý giữa 2 node xa nhất trong LAN, ttrans là thời gian chuyển 1 max – size frame.

efficiency = 1 / (1 + 5 * Tprop / ttrans)

efficiency tiến về 1 khi Tprop tiến về 0 hoặc ttrans tiến về vô cùng.

Cấu trúc Ethernet Frame: gồm 7 bytes đầu có cấu trúc 10101010, theo sau là 1 bytes có cấu trúc 10101011; dest address; source address; type; data; CRC để kiểm tra lỗi.

Các địa chỉ là 6 bytes. Type là giao thức được sử dụng ở tầng network. Phổ biến là IP. CRC để kiểm tra lỗi. Nếu có lỗi, frame sẽ bị huỷ.

Ethernet không có thiết lập kết nối, muốn gửi thì gửi. Không xác nhận, nếu frame bị huỷ thì xem như mất gói. Tầng trên có yêu cầu gửi lại thì gửi lại.

Chuẩn Ethernet 802.3, cấu trúc 100BASE – TX có nghĩa là truyền tốc độ 100Mbps, loại dây dẫn sử dụng là TX.

Ethernet Switch: thiết bị ở tầng link.

Router hoạt động trên địa chỉ IP, switch hoạt động dựa trên địa chỉ MAC.

Switch sẽ nhận gói tin, bỏ trong buffer, chuyển sang đường kế tiếp bằng cách đọc header.

Không diễn ra va chạm trong switch.

Switch không có địa chỉ nên các thiết bị không biết sự tồn tại của switch.

Cơ chế chọn đường đi của switch: trong switch có switch table lưu địa chỉ MAC, interface tới host, timestamp để lưu thời gian tồn tại.

Switch sẽ tự học để nhớ mỗi cổng sẽ gắn địa chỉ MAC nào.

Khi một frame đến switch:

- Switch lưu địa chỉ MAC của host gửi.
- Kiểm tra switch table xem tồn tại chưa.
- Nếu có tồn tại nhánh cho địa chỉ đích trong bảng. Nếu địch chỉ đích cùng nhánh địa chỉ gửi đi, hủy frame. Nếu khác, ta chuyển frame đến cổng đích tương ứng.
- Trong trường hợp không tìm được cổng đích, switch gửi đến tất cả các cổng.

VLAN: Các cách chia:

Chia theo port: mỗi nhóm là 1 LAN độc lập, dù kết nối chung một switch, mỗi VLAN có subnet ID khác nhau.

Nếu có nhiều switch, ta có thể set mỗi cổng nào thuộc VLAN nào.

Sau đó, các switch có đường kết nối với nhau gọi là trunk port.

Các chuẩn cho việc cấu hình VLAN là 802.1, 802.1q.

Cấu trúc gói tin VLAN:

802.1: tương tự cấu trúc 802.3 của gói tin trên Ethernet.

802.1q: bổ sung thêm VLAN ID, Tag để xác định protocol.

Đối với WiFi, dùng chuẩn 802.11.

Multiprotocol Label Switching (MPLS): tìm đường đi dựa trên IP. Cung cấp thêm 1 số header (label, exp, S, TTL) giúp tìm đường đi chặt chẽ hơn.

Ý tưởng tượng theo Virtual Circuit (VC) – mạng hướng kết nối.

Dựa theo label thay vì địa chỉ IP để quyết định đường đi.

Data center networks: trung tâm dữ liệu, nơi chứa rất nhiều máy tính. Dùng để lưu trữ và xử lý dữ liệu.

Để giải quyết vấn đề backup lẫn như, mạng kết nối giữa các máy rất phức tạp.

Có nhiều đường kết nối ra ngoài để đảm bảo hệ thống hoạt động tốt.

Chương 7:

Wireless:

Đặc điểm:

- Cường độ tín hiệu giảm theo khoảng cách.
- Dễ bị nhiễu.
- Bị phản xạ bởi tường, các vật cản.

SNR: signal – to – noise ratio: càng cao thì gửi tín hiệu càng tốt.

Hidden terminal problem: 2 thiết bị không biết nhau, cùng gửi tính hiệu đến một thiết bị khác có thể dẫn đến va chạm.

CDMA: Không cần chia kênh truyền, khi người sử dụng có yêu cầu cứ dùng hết băng thông kênh truyền.

Mỗi người sử dụng sẽ có một mã. Khi bên gửi gửi dữ liệu đi, hệ thống sẽ lấy dữ liệu như với mã theo dạng nhân vector với một số.

Bên nhận từ vector nhận được và mã của user, ta tính được dữ liệu.

Nếu có 2 người sử dụng. Kết quả tính toán sẽ bị giao thoa với nhau dẫn đến nếu cùng chiều thì xung tăng lên, ngược chiều thị xung thành không có.

bên nhận từ đoạn sóng giao thoa và mã user 1, tính ra được dữ liệu của user 1.

Thời gian xử lý của CDMA chậm hơn FDMA và TDMA, nhưng trong 1 khoảng thời gian, CDMA có thể cho 8 user sử dụng được. Nên về tốc độ, CDMA tương đương FDMA và TDMA.

IEEE 802.11:

Các chuẩn 802.11b, 802.11a, 802.11g, 802.11n (dùng CSMA/CA), 802.11ac.

Access Point sẽ liên tục phát beacon frame chứa ID và địa chỉ MAC để các thiết bị phát hiện.

Giải quyết Multiple Access: WiFi dùng CSMA/CA, tránh va chạm.

Bên gửi:

- Hệ thống kiểm tra nếu kênh truyền rỗi mới gửi. Nếu bận sẽ random số để chờ gửi sau.
- Khi timer hết, gói tin vẫn chờ khi có xác nhận mới bắt đầu gửi lên.

Bên nhận: Khi bên gửi gửi xong, bên nhận chờ thêm một khoảng thời gian. Nếu không còn gì được gửi, sẽ gửi lại 1 xác nhận để báo bên gửi hết quyền gửi, các thiết bị khác có thể tham gia quá trình gửi nhận.

Để kiểm tra kênh truyền có đang rỗi không, bên gửi sẽ gửi gói tin RTS (request to send). Nếu kênh truyền rỗi, nó sẽ gửi gói tin CTS (clear – to – send) đến tất cả thiết bị đang kết nối.

Cấu trúc gói tin 802.11:

Gồm Frame control (2B), Duration (2B), Address bên nhận (6B), Address bên gửi (6B), Address router (6B), Seq control (2B), Address dùng cho chế độ hoc (6B), Payload (0 – 2312B), CRC (4B).

802.15: Personal Area Network (Bluetooth): dùng cơ chế Master/Slaves: Master nằm giữa điều phối hoạt động cho các yêu cầu của các Slave.

Chương 8:

Network Security:

Vấn đề:

- Kẻ xấu tấn công mạng máy tính như thế nào.
- Làm sao chống được tấn công.
- Làm sao thiết kế các kiến trúc để miễn nhiễm tấn công.

Cách tấn công vào Internet:

- Mã độc: Virus (cần kích hoạt file để kích hoạt), Worm (tự kích hoạt và nhân bản).
- Phần mềm gián điệp (spyware malware).
- Botnet để tấn công DDoS.
- Packet “sniffing”: đánh cắp gói tin trên đường gửi đi.
- IP spoofing: giả danh 1 địa chỉ IP khác.

Network Security: giải quyết các vấn đề:

- Bảo mật như thế nào để trong quá trình truyền thông tin trên mạng chỉ có người gửi và người nhận xem được thông tin.
- Xác thực: khi trao đổi thì biết chắc chắn người nhận là người mà người gửi muốn gửi đến.
- Tính toàn vẹn thông điệp: nội dung không bị chỉnh sửa.
- Tính sẵn sàng của dịch vụ: đảm bảo dịch vụ 24/7.

Khi kẻ tấn công chỉ có ciphertext (gói tin đã mã hoá), có thể dùng các cách thức để tìm văn bản gốc:

- Đoán một cách ngẫu nhiên hoặc theo xác suất dựa trên đặc tính của ngôn ngữ.
- Biết cả known – plaintext (quy luật mã hoá).

Các cách mã hoá:

- ***Symmetric Key Cryptography – khoá đối xứng:*** dùng đúng 1 chìa khoá để mã hoá và giải mã. Đơn giản nhưng cần có cách chia sẻ cách mã hoá.
  - ***Substitution cipher:*** chuyển từ ký tự này sang ký tự kia. Key là bảng ánh xạ. Nhược điểm dễ giải mã.
  - ***Data Encryption Standard – DES:*** dùng block cipher để chia tin nhắn thành các khối gọi là bit và xử lý trên từng khối. Khối có 64 bits, khoá có 56 bits.
    Không còn an toàn ở thời điểm hiện tại. Ưu điểm là cần tốn thời gian để giải mã hơn do phải dùng vét cạn để giải.
    Nâng cấp lên 3DES để tạo 3 khoá khác nhau khiến tăng độ khó.
  - ***Advanced Encryption Standard – AES:*** tăng lên 1 block 128 btis, khoá có thể lựa chọn 128, 192, 256 bits tuỳ vào độ phức tạp.
    Một hệ thống giải mã DES trong 1 giây thì cần 149 ngàn tỉ năm để giải AES.
- ***Public Key Crypto – khoá công khai:*** dùng 2 chìa khoá thay vì 1 chìa. 1 chìa ai cũng biết, 1 chìa là private, chỉ bản thân biết. Khi dùng 1 chìa để mã hoá thì chỉ có thể dùng chìa còn lại để giải mã.

  RSA:

  1. Chọn 2 số nguyên tố p, q (private) dài 1024 bits.
  2. Chọn n = p * q (public), z = (p - 1) * (q - 1).
  3. Chọn e < n sao cho e và z không có ước số chung.
  4. Chọn d sao cho (e*d – 1) % z == 0.
  5. (n, e) là public key, (n, d) là private key.

  Cách mã hoá:

  1. Giả sử có tin nhắn m (< n), ta tính c = m^e % n.
  2. Để giải mã, m = c^d % n.

  Để tính toán RSA, thời gian tính toán chậm hơn DES, nên ít khi dùng. Thường dùng DES hay AES để mã hoá, rồi dùng RSA mã hoá key để gửi đi.
  Không tái sử dụng key để bảo đảm tính an toàn.

Authentication: Quá trình xác thực tài khoản đúng người dùng hay không.

- ***Playback:*** Cách tránh bắt gói tin để lấy password. Tạo thêm 1 số ngẫu nhiên nonce (R) (tương tự mã OTP) có thời gian tồn tại nhất định và chỉ sử dụng được 1 lần.

- ***Man in the middle attack:*** Bên tấn công đứng ở giữa để nhận và gửi mọi thông tin giữa 2 máy. Khó phát hiện.

Giải pháp:

Digital Signatures – Chữ ký số: dựa trên public key.

Dùng private key mã hoá văn bản, chỉ có public key của bản thân user đó mới có thể giải mã. Đảm bảo văn bản không thể sửa do chỉ bản thân user đó có thể mã hoá văn bản.

Nếu văn bản quá lớn thì sẽ tăng chi phí tính toán, độ dài chữ ký.

Dùng Message Digest để giải quyết: dùng hàm Hash đổi văn bản lớn thành chuỗi bit nhỏ hơn để áp dụng chữ ký số.

Đặc điểm hàm Hash: kích cỡ output là cố định; 2 văn bản có thể cùng ra một kết quả sau khi hash nhưng tỉ lệ rất nhỏ. Phổ biến đang dùng là MD5 (tạo ra chuỗi hash 128 bits), SHA – 1 (chuẩn của Mỹ, tạo ra chuỗi 160 bits).

Certification Authorities – CA: tổ chức có nhiệm vụ xác nhận public key là của ai.

E (person, router) đăng ký public key với CA.

- E cung cấp ID cho CA.
- CA tạo 1 chứng chỉ số xác nhận E có public key như thế nào.
- Một bên khác khi muốn lấy public key của user E sẽ liên hệ CA để lấy.

Trên thực tế có nhiều CA, CA muốn hoạt động cần đăng ký public key của CA đó đến root CA – cơ quan cao cấp nhất quản lý tất cả các CA.

Bảo mật Email: cần bảo vệ nội dung email để khỏi xem trộm. Dùng khóa đối xứng mã hoá dữ liệu. Sau đó dùng khóa công khai mã hoá khoá.

Để đảm bảo không bị chỉnh sửa nội dung, ta dùng chữ ký số. Từ nội dung tiến hành tạo chữ ký số và gửi nội dung sau hash kèm chữ ký đi. Nhưng như thế chưa mã hoá nội dung.

Do đó, đầu tiên ta sẽ tạo chữ ký số cho nội dung email. Sau đó, dùng khoá đối xứng và khóa công khai để mã hoá.

SSL – Secure Socket Layer: web dùng https là SSL. Dùng để mã hoá dữ liệu từ webbrowser đến web server. Được sử dụng giao dịch điểm tử.

Hỗ trợ giao thức TCP. Lớp SSL sẽ nằm giữa tầng ứng dụng và tầng vận chuyển.

Việc mã hoá sẽ tự động thực hiện thông qua việc sử dụng API của SSL.

Quá trình truyền gửi:

- Khi bắt đầu kết nối, server gửi về client public key của server.
- Client tạo khoá đối xứng.
- Client dùng public key để mã hoá khoá đối xứng và chuyển cho server.
- Từ sau đó, dữ liệu đi qua client và server sẽ mã hoá qua khoá đối xứng.

IP security – Network layer security: dùng Virtual Private Network (VPN).

IPsec Transport Mode: tạo 1 kênh trực tiếp từ máy này đến máy kia thông qua mạng.

Ipsec – Transport Mode: thông qua các router biên để hỗ trợ.

Firewall: kiểm soát gói tin nào đi vào và không được đi vào, được đi ra và không được đi ra.

- Dùng chống DoS.
- Ngăn chặn việc tiếp cận bất hợp pháp vào mạng nội bộ.
- Chỉ cho phép những ai được đi vào và ra khỏi mạng.

Các cơ chế chặn của Firewall:

- Stateless packet filters: không theo dõi trạng thái gói tin. Dựa trên header để quyết định cho phép hay không cho phép gói tin đi qua. Có Access Control Lists là danh sách các quy tắc cài sẵn để firewall sử dụng. Nhanh nhưng không xử lý được vấn đề phức tạp.
- Stateful packet filters: kiểm soát trạng thái gói tin. Theo dõi một số gói tin liên tiếp để ra kết luận.

- Application gateways: kiểm soát đến từng ứng dụng. Ngoài header, còn theo dõi nội dung dữ liệu bên trong gói tin.

GIới hạn của firewall:

- Không phát hiện được IP spoofing.
- Nếu kiểm soát quá kỹ, việc trao đổi sẽ khó khăn. Nếu thả lỏng thì sẽ có rủi ro về an ninh.

Intrusion Detection Systems: ngoài duyệt header, còn đọc nội dung gói tin. Nếu có dấu hiệu mã độc sẽ chặn và báo cho user để tìm hướng xử lý.