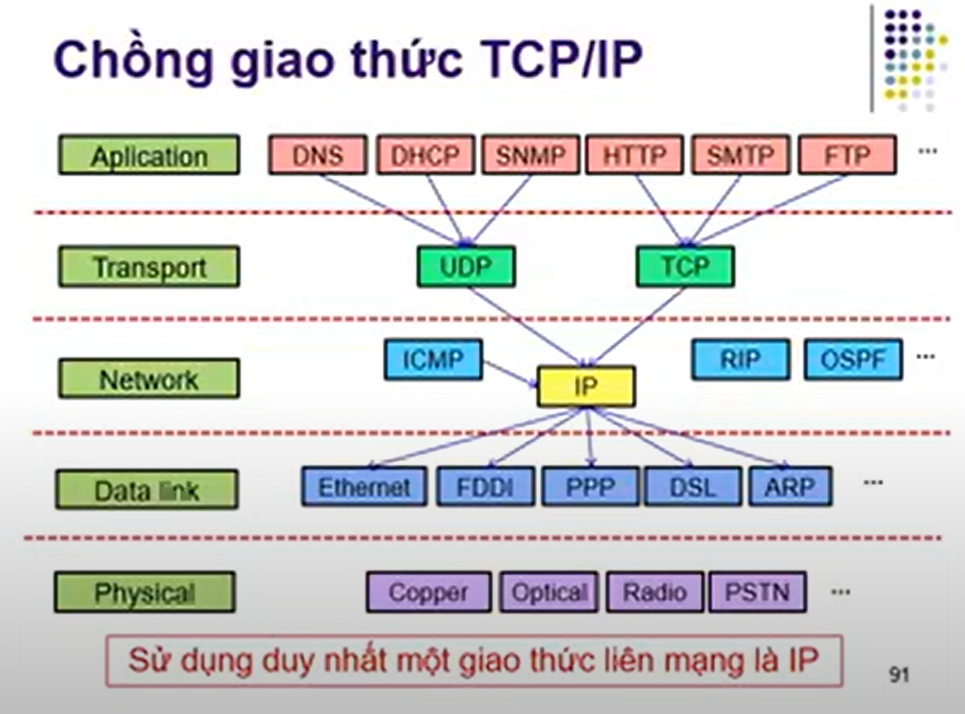
**Chương 1:**

* Giá trị BER(Bit Eror Rate/Ratio = Số bit lỗi/Tổng số bit truyền) phản ánh độ tin cậy
* Băng thông: Tốc độ truyền tin tối đa (Bps)
* Thông lượng: Tốc độ truyền dữ liệu qua 1 điểm nó trong mạng tại 1 thời điểm (Bps)
* Độ suy hao tin hiếu: mức độ suy giảm năng lượng
* Độ trễ: Thời gian truyền dữ liệu từ nguồn đến đích (s)
  + Trễ truyền tin: thời gian truyền hết tin
  + Trễ truyền dẫn: thời gian lan truyền tin hiệu từ nguồn đến đích
    - Kích thước băng thông và chiều dài gói tin
  + Trễ xử lý tại nút chuyển tiếp (nút trung gian)
    - Kiểm tra lỗi và xác định liên kết ra
  + Trễ hàng đợi: Thời gian chờ trong hàng đợi
* Thông số RTT(Round Trip Time): trễ 2 chiều giữa nút nguồn và nút đích
  + Thời gian từ khi nút nguồn gửi dữ liệu đi tới khi nhận được dữ liệu trả lời của nút đích
* Ví dụ 3: <https://www.youtube.com/watch?v=pnD7IesJ98g&list=PLqOOvs1KdYaMPcSrPBD4BIctgRkjeRz1b&index=5>
* Cơ chế truyền“best-effort”: Chỉ truyền dữ liệu 1 lần, không phát lại
  + Truyền với khả năng tối đa, nếu lỗi sẽ không truyền lại
* Đánh giá độ tin cậy của đường truyền :
  + Tỉ lệ lỗi bit BER
  + Tỉ lệ mất gói tin
* Giao thức: Các quy ước về địch dạng dữ liệu, thủ tục trao đổi dữ liệu
  + Trồng giao thức: Giao thức tầng trên dùng của tầng dưới...
* **Kiến trúc phân tầng** 
  + Tầng dưới cung cấp dịch vụ cho tầng trên qua điểm truy cập dịch vụ (SAP)
    - Tính trong suốt: tầng trên sử dụng dịch vụ của tầng dưới mà không cần quan tâm cách thức tầng dưới thực hiện
  + Một số tầng không cần triển khai trên tất cả các nút mạng
  + Tại mỗi tầng, hai bên tham gia quá trình truyền tin phải sử dụng giao thức giống nhau
  + Quá trình đóng gói dữ liệu (encapsulation) tại bên gửi đc thực hiện từ tầng trên xuống tầng dưới (được thêm tiêu đề mới,...)
  + PDU (Protocol data unit): đơn vị dữ liệu được đóng gói theo giao thức của mỗi tầng trong kiến trúc phân tầng
* TCP/IP:

1. Kênh

* Đường truyền vật lý
* **Chuyển mạch kênh:**
  + Tài nguyên của mỗi kênh được xác định trong giai đoạn thiết lập kênh và không đổi suốt quá trình truyền dữ liệu
  + Băng thông kênh phụ thuộc vào tài nguyên truyền
  + Kênh sẽ giải phóng nếu 1 trong 2 bên ngắt kết nối
* **Chuyển mạch gói:**
  + Hiệu suất sử dụng đường truyền cao hơn
  + Có thể có mức ưu tiên tài nguyên trên hàng đợi



* POP3,IMAP – email : TCP, lấy email từ sever về máy tính cá nhân
  + IMAP lấy tiêu đề trước thân
  + POP lấy đồng thời tiêu đề, thân email
* nslookup: phân giải tên miền sang IP
* Các tên miền là phân cấp
  + Tìm kiếm bắt đầu từ nút cấp 1
  + Không tìm thấy sẽ đệ quy về node gốc

**Tầng vật lí và liên kết dữ liệu**

1. **Chia kênh – ngẫu nhiên:**

* CSMA: điều khiển truy nhập ngẫu nhiên – đa sóng mang
* Họ Aloha: điều khiển truy nhập ngẫu nhiên
* TDMA: điều khiển truy cập theo thời gian
* FDMA: điều khiển truy cập theo tần số
* CDMA: điều khiển truy cập theo mã
* Token Passing: điều khiển truy cập theo thẻ bài

1. **Đụng độ:**

* Chia kênh :Tài nguyên đường truyền sẽ được chia nhỏ và gán cho 1 nút, các nút ko dùng chung đường truyền
* Không có đụng độ
* Thẻ bài (lần lượt): thẻ bài điều khiển các nút lần lượt sử dụng đường truyền
* Không có đụng độ
* Ngẫu nhiên: Thứ tự xác xuất đụng độ giảm dần Pure Aloha>Slotted Aloha>CSMA

1. **Pure Aloha hay Aloha thuần túy**

* Ngẫu nhiên
* Nút mạng sẽ truyền ngay (không kiểm tra)
* Không đồng bộ thời gian
* Truyền 1 Khung tin / 1 frame time

1. **Solotted Aloha**

* Ngẫu nhiên
* Đồng bộ thời gian
* Hiệu quả truyền gấp đôi pure aloha
* Nút mạng sẽ truyền ngay
* Truyền 1 khung tin/ 1 frame tim

1. **CSMA/CD**

* Cảm nhận năng lượng sóng mang trên đường truyền trước khi truyền (Nghe trước khi nói – CSMA)
* Cảm nhận năng lượng sóng mang khi truyền khung tin đầu tiên để phát hiện đụng độ (Nghe trong khi nói – CD)
* Sau khi phát hiện đụng độ: Duy trì việc phát tín hiệu đụng độ trên đường truyền trong một khoảng thời gian để tất cả nút mạng cảm nhận được – tín hiệu JAM
* Dùng cho token bus, ring hoặc cả 2

1. **Token Ring**

* Chỉ tồn tại duy nhất một thẻ bài để xác định quyền đưa dữ liệu lên đường truyền
* Mỗi nút mạng được phép sử dụng thẻ bài trong một khe thời gian xác định
* Nút mạng được sử dụng thẻ bài trong thời gian tùy ý
* Cho phép thiết lập mức độ ưu tiên truyền dữ liếu
* Là gói tin có khuôn dạng và kích thước xác định

1. **Chuyển mạch tầng 2 (Switch,Bridge)**

* Không cần giao thức xác định trước đường đi -> Tự học
  + Nếu chưa biết địa chỉ nguồn của khung tin, thêm vào bảng MAC table
  + Không bổ sung địa chỉ đích
* Cần bảng chuyển tiếp
* Chuyển mạch giữa các nút mạng trong cùng 1 mạng
* Chuyển tiếp dữ liệu quảng bá khung tin
  + FF-FF-FF-FF-FF-FF là 1 địa chỉ quảng bá -> Chuyển khung tin ra tất cả các cổng trừ cổng gửi
  + Chế độ “store and forward”: kiểm tra và gửi
  + Route thì không
* Nếu gói tin không có địa chỉ MAC của nút đích thì gửi thông điệp ARP Request để tìm kiếm khi có địa chỉ IP
  + Nhận được ARP trả lời + địa chỉ MAC
* Kiểm tra lỗi trên khung tin

1. **Giao thức ARP**

* Tìm kiếm địa chỉ MAC khi biết địa chỉ IP
  + Inverse ARP: Tìm kiếm địa chỉ IP khi biết MAC

1. **MAC**

* Địa chỉ MAC là địa chỉ vật lí 48 bit -> Không thay đổi
* Được gán khi thiết bị sản xuất
* Định danh các nút trong 1 mạng

1. **Ethernet 1000-BaseT**

* Tốc độ: Mbps
* T: kí hiệu cáp xoắn đôi
* Thuộc IEEE 802.3(nhóm đều dùng CSMA/CD)
  + IEEE 802.11 là mạng Lan không dây (WLAN)
* Chuẩn cho mạng hình sao
* Sử dụng đầu nối RJ-45

1. **Các chuẩn FAST Ethernet**

* Tốc độ truyền tối đa là 100Mbps

1. **Các mã phát hiện lỗi**

* Paritity: Mã chẵn lẻ - không cho phép phát hiện vị trí lỗi
* Checksum: Mã tổng – không cho phép phát hiện vị trí lỗi
* CRC – 16(32): Mã vòng 16bit(32bit) – cho phép phát hiện vị trí lỗi

1. **Các mã hóa điều chế dữ liệu số - tín hiệu:**

* Mã NRZ
* Mã Manchester
  + Tầng vật lí, để biểu diễn dữ liệu thành tín hiệu
* Mã Bipolar:
  + Bit 1 được biểu diễn luân phiên bởi các xung âm và xung dương
* Mã RZ:
  + Chuyển về mức điện áp 0 ở giữa xung
* Mã HDBn:
  + Thay thế chuỗi các bit 0 liên tiếp bằng mẫu bit đặc biệt tránh mất đồng bộ

1. **Cáp**

* Cáp xoắn đôi: T
* Cáp đồng trục mỏng: 2
* Cáp đồng trục chuẩn: 5
* Cáp quang: F

**Tầng Mạng**

1. **Địa chỉ IP**

* 2 loại là phân lớp (A,B,C,D) và không phân lớp (/xx)
  + Nếu phân lớp thì k có mặt nạ hay submark, ngược lại
* **/**xx -> NetWorkID có xx bit, HostID có 32-xx bit
  + HostID có cả bit 0,1 -> địa chỉ unicast -> gán cho máy trạm
  + Địa chỉ mạng LAN – private:
    - 10.0.0.0/8: lớp A
    - 172.16.0.0/16 -> 172.31.0.0/16: lớp B
    - 192.168.0.0/24 -> 192.168.255.0/24: lớp C
  + Loopback address : 127.0.0.0 /8 – 127.0.0.1/8
  + Địa chỉ liên kết nội bộ: 169.254.0.0/16 (tự động cấu hình khi các nút mạng không được cấu hình mạng tự động bởi dịch vụ DHCP)
* Cách tìm địa chỉ dải mạng:
  + AND với mặt nạ /xx với địa chỉ IP -> địa chỉ mạng
  + Cách 2: trick lỏ bí mật siêu cấp tính nhanh =))) đọc đc thì ib
* Chia mạng con:
  + Ví dụ : /22 chia cho mạng 160.12.64.0/19
  + Các mạng con có subnetID với số bit là k = Mặt nạ mạng con – Mặt nạ mạng gốc = 22-19 = 3
  + -> Có 2^3 = 8 mạng con có subnetID khác nhau
  + Số địa chỉ máy trạm có thể cấp phát: 2^(32-22)-2 (trừ địa chỉ mạng – full 0 và địa chỉ quảng phát – full 1)

1. **Hoạt đọng của giao thức IP**

* TTL: 8bit – Thời gian sống, cho biết độ dài còn lại gói tin có thể đi qua được hay số chặng tối đa gói tin có thể đi qua được
  + Phát báo nhận cho nút gửi
  + Đi qua 1 trạm thì -1
  + = 0 thì gói tin được hủy
* Ip là giao thức hướng không liên kết (không tin cậy)
* Định tuyến trước khi gửi gói tin
* Nguyên tắc đóng gói và truyền thông:
  + Chuyển gói tin cho tầng liên kết dữ liệu xử lý
  + Đặt dữ liệu nhận được từ tầng giao vận vào gói tin và thêm thông tin điều khiển (IP header có TTL,upper protocol,checksum)
  + Kiểm tra checksum để phát hiện lỗi – không sửa lỗi
  + Xác định giao thức tầng trên nào sẽ xử lí dữ liệu (upper protocol)
    - ICMP, IGMP, TCP, UDP, OSPF
* Hợp mảnh các gói tin nếu cần (quá to > MTU)
  + MTU: kích thước tối đa của gói tin
  + Trường Identification: ID được sử dụng để tìm các phần của gói tin
  + Flags – cờ : 3bit
    - Bit 1: dự phòng
    - Bit 2: Không được phép ghép mảnh
    - Bit 3: Còn phân mảnh
  + Độ lệch – Offset:
    - Vị trí gói tin phân mảnh trong gói tin ban đầu
    - Theo đơn vị 8 bytes
* **Ví dụ:** Một gói tin IP có kích thước phần dữ liệu (payload) là 1200 byte bị phân thành 3 mảnh có giá trị Fragment Offset lần lượt là 0,69,138. Phần dữ liệu trong các mảnh này có kích thước lần lượt là bao nhiêu byte

|  |  |  |
| --- | --- | --- |
| 0 x 8 551 | 69 x 8 = 552 1103 | 138 x 8 = 1104 1199 |

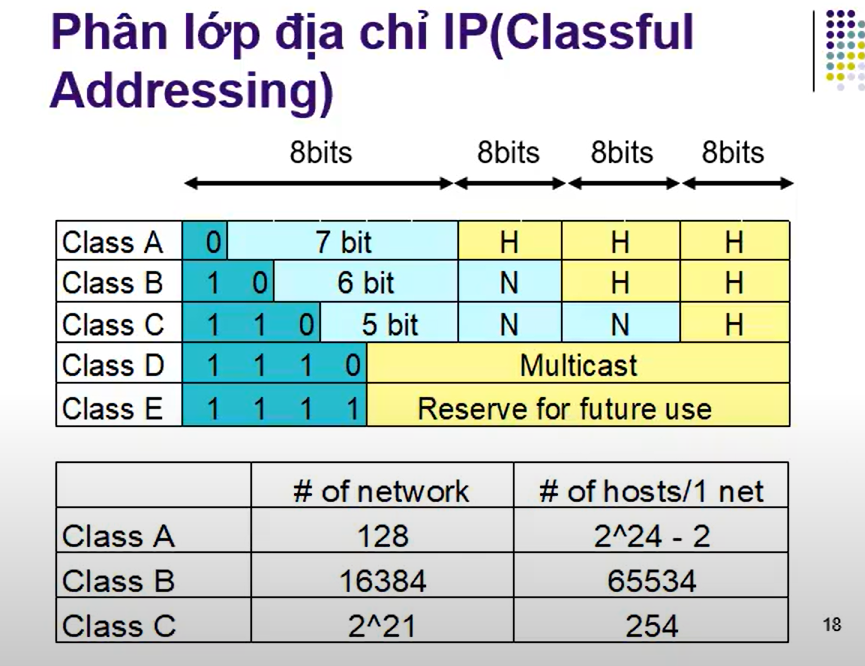
* + 552, 552, 96

1. **Địa chỉ Ipv4**

* Độ dài : 32 bit
* Biểu diễn : O1.O2.O3.O4 (0 <= Oi <= 255)
* Định dạng

|  |  |
| --- | --- |
| NetworkID (n bit) | HostID (32-n bit) |

* Các dạng địa chỉ:
  + Địa chỉ mạng (Network Address): Tất cả các bit của HostID là 0
  + Địa chỉ quảng bá (Broadcast Address):
    - Địa chỉ quảng bá cục bộ: 255.255.255.255
    - Địa chỉ quảng bá của mạng: tất cả các bit của HostID là 1
  + Địa chỉ nhóm (Multicast Address): Địa chỉ phân lớp D
  + Địa chỉ máy trạm (Unicast Address): Còn lại, không thuộc lớp D-E
* **Phân lớp: A, B, C, D:**

****

* **Không phân lớp:** O1.O2.O3.O4 /n:
  + **/**n : mặt nạ mạng = NetworkID có n bit
  + Prefix: /n
  + Nhị phân: 32 bit – n bit 1 đầu tiên, 32 – n bit 0
* Không gian địa chỉ đặc biệt: (phần đầu)
* Ví dụ:

1. **Mạng con**

|  |  |  |
| --- | --- | --- |
| Network ID n bit | SubnetID k bit | HostID 32 – n - k |

* Mặt nạ mạng con: /(n+k)
* SubnetID: định danh mạng con = Mặt nạ mạng con – mặt nạ gốc = k
* Số mạng con chia được là 2^k

1. **Router**

* Router báo lỗi bằng thông điệp ICMP
* Router luôn kiểm tra TTL trước khi chuyển tiếp một gói tin
* Router tìm kiếm lối ra dựa trên địa chỉ đích
* Phân mảnh gói tin nếu cần
* Gói tin bị loại bỏ khi:
  + Phát hiện lỗi thông qua checksum
  + Không tìm thấy cổng ra trên bảng chuyển tiếp
  + Hàng đợi router bị đầy
  + Giá trị TTL = 1 (hoặc = 0)

1. **Các giao thức IP**

* DNS: phân giải tên miền sang IP
* DHCP: Cấp phát địa chỉ tự động (Dynamic Host Configuration Protocol)
* ARP: Tìm kiếm địa chỉ MAC
* NAT (NetWork Address Tranlation): Chuyển đổi địa chỉ IP từ mạng cục bộ sang mạng công cộng

1. **Định tuyến**

* Định tuyến theo vector khoảng cách:
  + Bellman-ford - Cho phép tìm đường đi ngắn nhất giữa mọi cặp nút
  + Mỗi 1 nút khi nhận được 1 vector khoảng cách từ láng giềng sẽ tính toán lại vector khoảng cách của nó và gửi lại cho láng giềng
    - Note: không phải chuyển tiếp(lan chuyền) mà là trao đổi, thu thập thông tin
* Định tuyến theo trạng thái liên kết:
  + Thông tin trạng thái liên kết được lan truyền cho tất cả các nút trong mạng
  + Djistra – Đường đi ngắn nhất
  + Các nút sẽ thu thập các thông tin của các nút khác trong mạng và xây dựng hình trạng (topology) của mạng – quảng bá thông tin

1. **Giao thức định tuyến**

* **BGP**: giao thức định tuyến ngoại vùng - tìm đường đi giữa các vùng cực trị AS
  + Chỉ cài đặt trên router biên của AS chứ không phải tất cả định tuyến trong AS
  + Truyền thông tin qua liên kết TCP
  + Sử dụng thuật toán vector đường đi
  + Phiên eBGP: thực hiện giữa các router khác AS
  + Phiên iBGB: thực hiện giữa các router cùng AS
* Giao thức nội vùng:
  + RIP, OSPF, IGPF, EIGRP
* **OSPF:** Giao thức định tuyến theo trạng thái liên kết
  + Có cơ chế định tuyến phân cấp
  + Có thể cài đặt cho các router trong vùng tự trị > 30 router
* **RIP:** 
  + V1: dùng trong mạng có phân lớp (A, B, C, D) có tối đa 15 router
  + V2: dùng trong mạng không phân lớp (có mặt nạ /txx) có tối đa 30 router
  + Chi phí đường đi dựa trên số chặng (hop)
  + Đơn giản hơn OSPF

**Tầng ứng dụng**

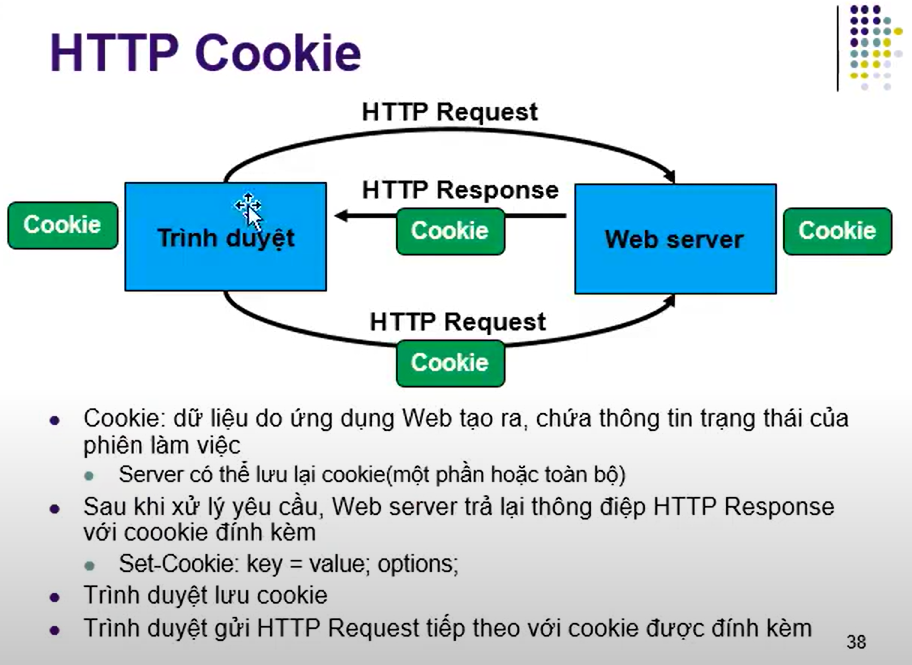
1. **1 số giao thức**

**Sử dụng TCP – giao vận**

* HTTP: giao thức web (Web mail,...)
  + HTTP Request 1.0 và 1.1:
    - Get-Post: yêu cầu
    - Pust: tải file
    - Head: lấy tiêu đề
  + HTML: mã nguồn của web (Có thể chứa nhiều văn bản)
  + Cần bao nhiêu thông điệp HTTP Request để truy cập vào một trang web chứa 20 bức ảnh ?
    - 1 cho một trang web để truy cập HTML
    - 20 cho yêu cầu 20 bức ảnh
    - -> Cần 21 thông điệp
    - 1 Request -> 1 Response
  + Cần bao nhiêu kết nối TCP nếu sử dụng giao thức HTTP 1.1 để truy cập 1 trang web có một đoạn văn bản và 10 ảnh (2 máy chủ web khác nhau)
    - 1 kết nối liên kết với máy chủ web để lấy mã nguồn HTML có đoạn văn bản và liên kết đến các ảnh
    - 1 kết nối liên kết (do 1.1 là liên kết liên tục) để lấy 10 file ảnh
    - -> 2 liên kết
* FTP : giao thức điều khiển dịch vụ truyền file
  + Cổng 20 – truyền dữ liệu,21 – truyền lệnh
* SMTP: giao thức điều khiển dịch vụ thư điện tử email
  + Có thể chuyển email từ máy người dùng -> email máy chủ
  + Có thể từ email máy chủ -> email máy chủ khác
    - Ví dụ từ user@gmail.com -> [user@yahoo.com](mailto:user@yahoo.com)
  + Note: gửi mail chứ không lấy mail
* POP, IMAP: Giao thức giúp người dùng truy cập vào hòm thư của họ
  + POP cho phép client lấy đồng thời tiêu đề và thân mail từ server
  + IMAP cho phép client lấy tiêu đề trước
* Web caches: là P2P

**Sử dụng UDP – giao vận**

* DNS: giao thức miền
  + Cổng 53
  + Không gian tên miền có cấu trúc phân cấp:
    - Ví dụ: andepzai.siunhan.edu.vn.
    - Ví dụ miền gốc -> miền cấp 1: .vn -> miền cấp 2: .edu -> ...
    - Note: Nút gốc không lưu trữ thông tin tên miền
  + Cơ chế phân giải đệ quy, máy chủ tên miền luôn chuyển truy vấn cho máy chủ gốc
    - Thường dùng cache để lưu máy chủ gốc, cấp 1 -> rút ngắn
  + Cơ chế phân giải tương tác
  + Mỗi tên miền có thể ánh xạ đến nhiều địa chỉ IP và ngược lại (many to many)
* DHCP: giao thức cấp phát IP
* SNMP: giao thức giám sát và quản lý thiết bị mạng – quản trị mạng
  + Ngôn ngữ chung cho thiết bị mạng
* NFS: tệp tin máy chủ
* Cookies:



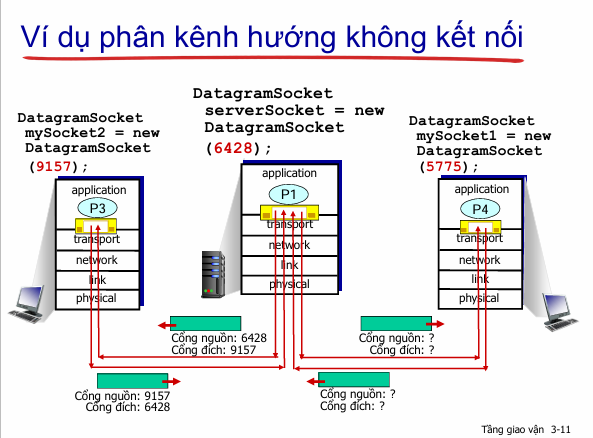
**Tầng giao vận**

1. **Số hiệu cổng**

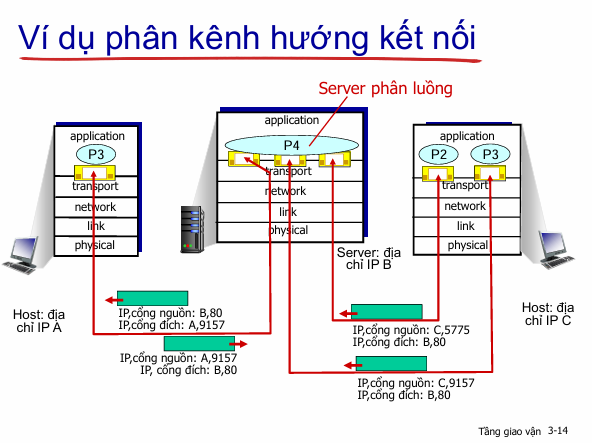
* Chuyển dữ liệu cho tiến trình tầng ứng dụng dựa vào số hiệu cổng đích
* Số hiệu cổng là định danh nội bộ trên cùng 1 máy tính
  + Các host có thể sử dụng số hiệu cổng giống nhau được
* 1 socket - 1 tiến trình

1. **UDP**

* Mỗi tiến trình trên nút A sử dụng 1 socket để trao đổi thông tin với tiến trình trên nút B

****

* + TCP: 1 tiến trình bên A có thể sử dụng nhiều socket (phân luồng) để trao đổi với socket bên B

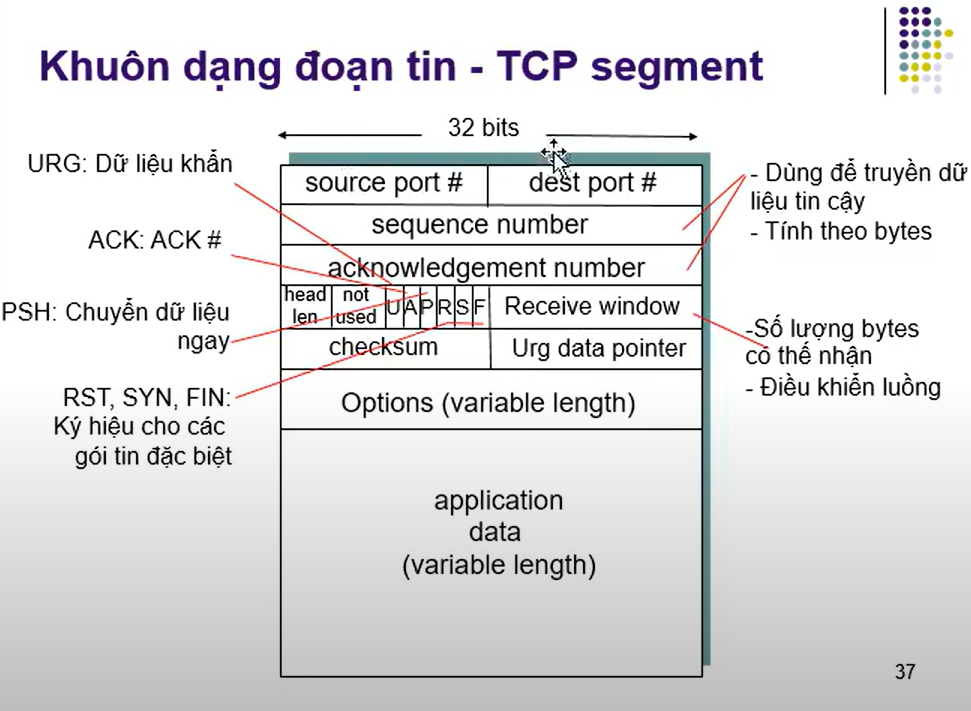
****

* + Mỗi socket có 1 số cổng hiệu khác nhau
    - Số cổng hiệu là định danh cho các tiến trình ứng dụng
* Nút B sẽ không mở sẵn socket do chưa biết nút A trao đổi bao nhiêu tiến trình
  + Khác với TCP
* Các gói tin từ nút A tới tiến trình trên nút B có cùng số hiệu cổng đích
* Gửi dữ liệu dạng “best-effort” – tốc độ cao nhất
* Hướng không liên kết – không thiết lập liên kết trước khi truyền
* Sử dụng checksum để kiểm tra lỗi
  + Nếu lỗi thì hủy (mất) gói tin – không sửa
* Truyền dữ liệu theo datagram
  + Gửi liên tục các datagram mà không cần chờ báo nhận – ko quan tâm kết quả gửi
* Không tin cậy
  + Đặc điểm :
    - Note: không phải do “không thiết lập liên kết trước khi truyền”
      * Nó có thể thiết lập cơ chế báo nhận -> có tin cậy
    - Không sử dụng báo nhận
    - Không có bộ đếm time-out
  + Dùng trong trường hợp:
    - Kiểm tra trạng thái hoạt động giữa các nút mạng - SNMP
    - Truyền dữ liệu video trong hội nghị trực tuyến

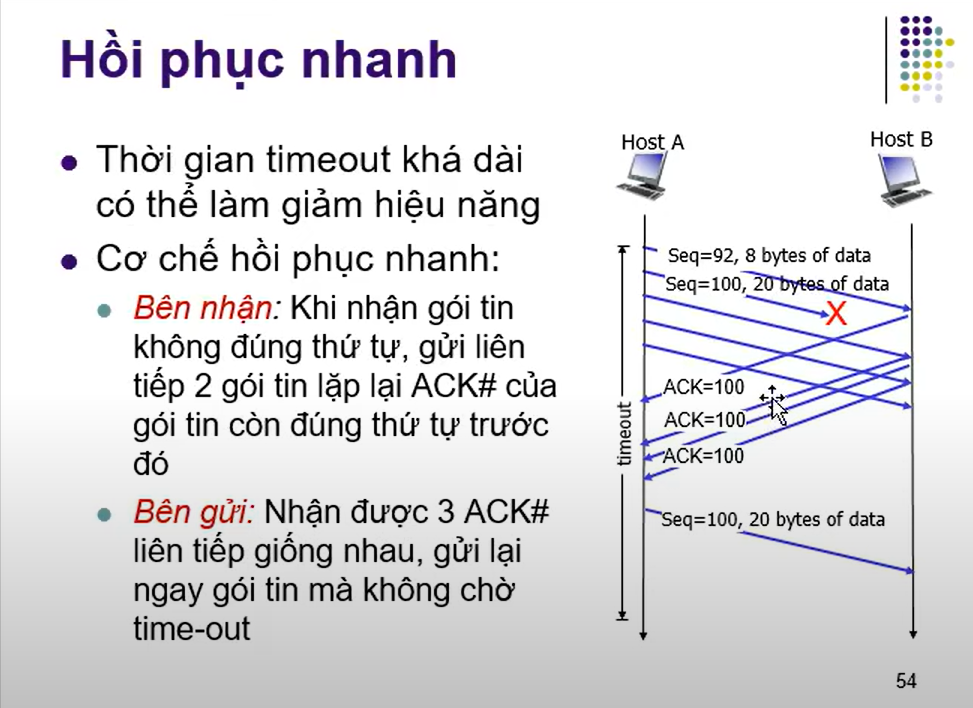
**Lợi thế so với TCP:**

* Kích thước phần tiêu đề nhỏ hơn
* Hoạt động đơn giản hơn, nhanh hơn

1. **TCP**

****

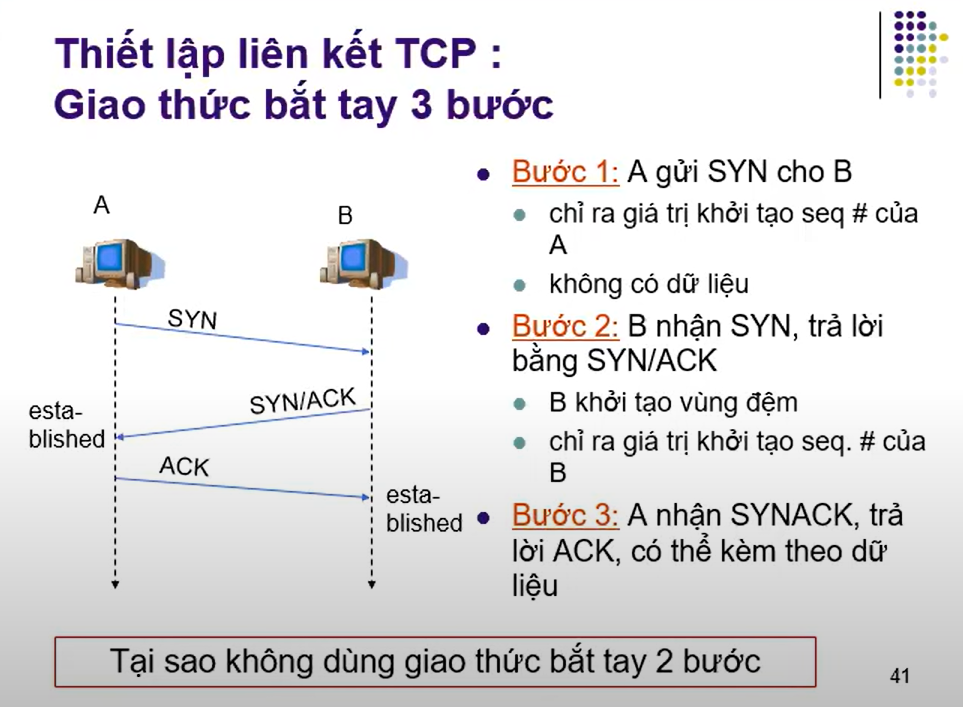
* Cờ SYN: cờ để thiết lập liên kết
  + 1: thiết lập
* Cờ ACK: thiết lập phản hồi
  + 1: thiết lập
* Cờ FIN: cờ báo kết thúc dữ liệu
* 1 tiến trình bên A có thể sử dụng nhiều socket để trao đổi với socket bên B
* Đặc điểm của truyền thông Tin cậy
  + Sử dụng checksum 16bit để kiểm soát lỗi
  + Sử dụng ACK báo nhận dữ liệu thành công
  + Phát lại dữ liệu khi xảy ra time-out
  + Note: không phải “kiểm soát luồng – tắc nghẽn”
    - Do nó có thể chạy theo cơ chế stop and wait – dừng và chờ
* Nếu bộ đệm bên nhận đầy -> loại bỏ gói tin -> không gửi lại ACK gói tin đó -> gửi ACK xác nhận gói tin trc đó với giá trị Rwnd = 0
* Cơ chế hồi phục nhanh (trong TH timeout dài):
  + Phát hiện sớm tắc nghẽn

****

* + Tóm lại là nhận 3 gói tin báo nhận có ACK number giống nhau -> phát lại
* Windows size trong phần tiêu đề: xác định lượng dữ liệu tối đa bên nhận có thể nhận
  + - Swnd: kích thước dữ liệu tối đa bên gửi có thể gửi
  + Swnd <= min{Rwnd, Cwnd}
    - Rwnd: Kích thước cửa sổ nhận( kích thước dữ liệu tối đa bên nhận có thể nhận)
    - Cwnd: Kích thước cửa số kiểm soát tắc nghẽn
  + Rwnd < Cwnd thì Swnd = Rwnd
  + Rwnd > Cwnd thì Swnd = Cwnd

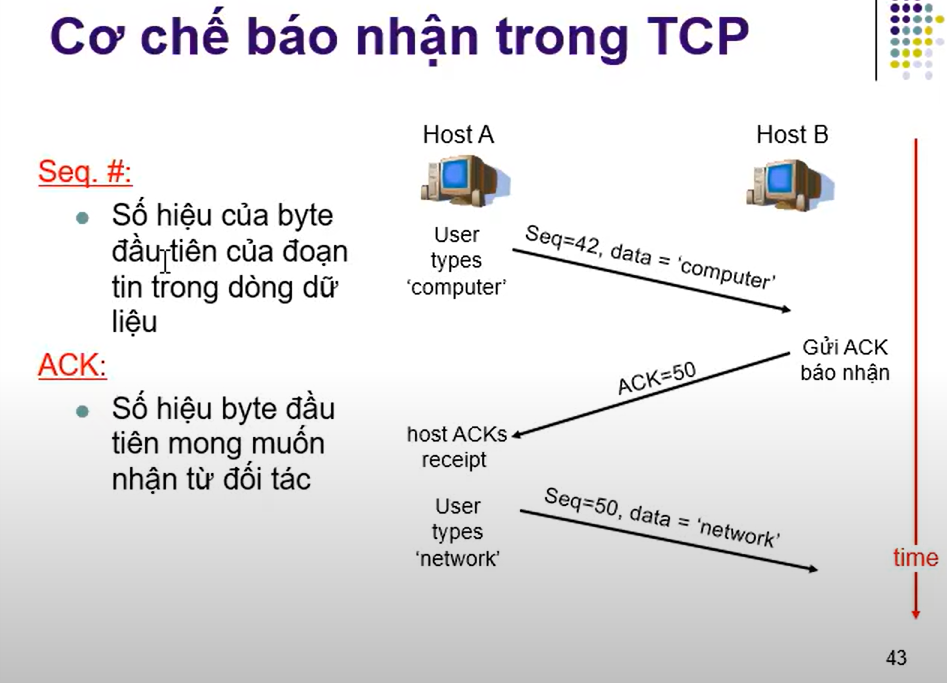
**Thiết lập liên kết: Giao thức bắt tay 3 bước:**

* Bước 1: SYN = 1, ACK = 0
* Bước 2: SYN = 1, ACK = 1
* Bước 3: SYN = 0, ACK = 1

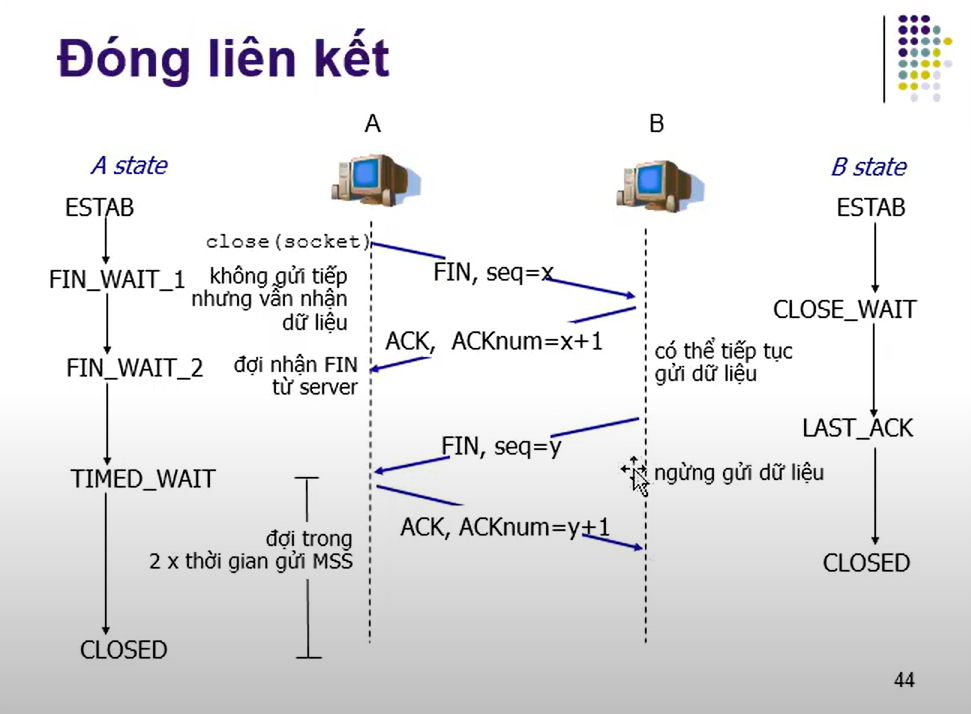
****

**Cơ chế báo nhận**

* Seq #: số hiệu byte đầu tiên của đoạn tin trong dòng dữ liệu
* ACK: số hiệu byte đầu tiên mong muốn nhận được từ đối tác
  + Gửi dữ liệu gói tin bắt đầu seq = 42, ACK = 1
  + Bên nhận trả lại thông báo muốn seq = 1, ACK = 90
  + Gửi dữ liệu gói tin bắt đầu seq = 90, ACK = 1

****

**Cơ chế đóng liên kết**

****

* + FIN = 1: đóng không gửi dữ liệu nhưng vẫn nhận tín hiệu

**Nguyên lý kiểm soát tắc nghẽn**

|  |  |
| --- | --- |
|  |  |

* Khởi tạo cửa sổ kiểm soát tắc nghẽn cwnd = 1 MSS
* 2 cách xử lí:
  + Time-out: cwnd bắt đầu lại quá trình slow- start
    - Giảm kích thước ngưỡng (threadhold) = 1 nửa so với time-out
  + Hồi phục nhanh: cwnd tụt xuống ngưỡng (tránh tắc nghẽn)
    - Giảm cwnd = 1 nửa so với mốc 3 ACK (hp nhanh) và biến thiên tuyến tính

**Pipeline:**

**Lợi thế so với UDP:**

* Tin cậy hơn
* Không quá tải nút nhận do có cơ chế kiểm soát tắc nghẽn
* Nút mạng nhận gói tin TCP -> kiểm tra 16 bit cuối(số hiệu cổng đích)
* Nút mạng gửi gói tin TCP -> kiểm tra 16 bit đầu(số hiệu cổng nguồn)