Lý Thuyết Mạng Máy Tính

**Mục Lục:**

[**Chương 2: Các thành phần của mạng máy tính.** 3](#_Toc491289773)

[**2.1 Phân loại mạng máy tính theo kỹ thuật truyền tin.** 3](#_Toc491289774)

[**2.2 phân loại mạng máy tính theo phạm vi địa lý.** 3](#_Toc491289775)

[1. Mạng cục bộ: 3](#_Toc491289776)

[2. Mạng hình sao: 4](#_Toc491289777)

[3. Mạng hình vòng: 4](#_Toc491289778)

[4. Mạng đô thị: mạng MAN dk sử dụng để nối tất cả các máy tính trong phạm vi toàn thành phố. 4](#_Toc491289779)

[**Chương 3: Tầng ứng dụng.** 4](#_Toc491289780)

[3.1: Các nguyên tắc của ứng dụng. 4](#_Toc491289781)

[3.2: WEB và HTTP 7](#_Toc491289782)

[3.3 FTP server 13](#_Toc491289783)

[**3.4: Giao thức mail:** 14](#_Toc491289784)

[3.5: DNS(Domain Name System) 17](#_Toc491289785)

[A. Tầng ứng dụng Transport(TCP và UDP) 22](#_Toc491289786)

[B\_ Tầng netword( tầng mạng). 26](#_Toc491289787)

**Giao thức: là cách thức giao tiếp.**

**Phần mềm hỗ trợ:Filezila- serve và client**

* **Các giao thức gôm:**

1. Host
2. Router
3. Server

* **Mô hình Các tầng giao thức của internet: gồm có 5 tầng.**

1. **Application: Hỗ trợ các ứng dụng mạng.(Data)**
2. **Transport: truyền dữ liệu host-host(port+Data)**
3. **Netword: dẫn đường các datagram từ nguồn tới đich.(IP|port+data)**
4. **Link: truyền dữ kiệu giữa các thành phần mạng láng giềng.(Mac|IP+Port+Data)**
5. **Physical**

* **Mô hình Giao thức OSI: Gồm có 7 tầng.**

1. **Application:tầng ứng dụng: cung cấp các ứng dụng truy suất dịch vụ mạng.**
2. **Presentation: tầng trình bày: đảm bảo cho các máy tính có kiểu dữ liệu định dạng dữ kiệu khác nhau vẫn có thể trao đổi thông tin cho nhau.**
3. **Session:tầng giao dịch: cho phép các ứng dụng thiết lập, sdung và xóa kênh giao tiếp giữa chúng.**
4. **Transport:tầng vận chuyển: đảm bảo việc truyền tải dữ liệu giữa các quá trinh, dữ liệu chuyển đi được đảm bảo ko có lỗi.**
5. **Netword:tầng mạng:đảm bảo gói tin có thể chuyền từ máy này tới máy kia dù ko có đường truyền vật lý trực tiếp giữa chúng.**
6. **Datalink: tầng liên kết: đảm bảo cho các khung dữ liệu giữa hai máy có đường truyền vật lý nối trực tiếp với nhau.**
7. **Physical: tầng vật lý: truyền tải các bit trên đường truyền vật lý.**

**Chương 2: Các thành phần của mạng máy tính.**

**2.1 Phân loại mạng máy tính theo kỹ thuật truyền tin.**

**-** Mạng quảng bá:

+ trong hệ thống mạng quảng bá chỉ tồn tại 1 lênh truyền đk chia sẻ cho tất cả các máy tính.

+ khi 1 máy tính gởi tin, tất cả các máy tính còn lại sẽ nhận tin đó

+ tại 1 thời điểm chhir cho phép 1 máy tính dk phép sdung đường truyền.

* Mạng điểm nối điểm:

+ Trọng hệ thống mạng này, các máy tính đk nối kaij với nhau thành từng cặp.

+ Thông tin được gởi đi sẽ dk truyền trực tiếp từ máy gởi đến máy tính nhận.

+ Hoặc đươc chuyển tiếp qua nhiều máy trung gian trước khi đến máy tính nhận.

**2.2 phân loại mạng máy tính theo phạm vi địa lý.**

- gồm có 3 loại: LAN – MAN – WAN

+ LAN: là mạng cục bộ. 10m -> 1km

+ MAN: mạng thành phố. Phạm vi: 10km

+ WAN : mạng diện rộng. Trên 100km.

1. Mạng cục bộ:

* Mạng hình bus: tất cả các máy tính dk nối lại bằng 1 dây dẫn

1. Mạng hình sao:

* Các máy tính đl nối vào 1 bộ tập trung nối kết, gọi là Hub,switch.

1. Mạng hình vòng:

* Khi truyền tin 1 máy tính có 1 thẻ bài, gửi gói tin đi 1 vòng trong qua các máy tính trên đường tròn.
* Chuyền theo chiều hình tròn (ngược chiều kim đồng hồ).

1. Mạng đô thị: mạng MAN dk sử dụng để nối tất cả các máy tính trong phạm vi toàn thành phố.

* Phần mềm mạng:
* Được xây dựng trên 3 khái niệm: giao thức, dịch vụ và giao diện.
* Giao thức: mô tả cách thức hai thành phần giao tiếp trao đổi thông tin với nhau.
* Dịch vụ: tô tả những jr mà 1 mạng máy tính cung cấp cho các tphan muốn giao tiếp với nó.
* Giao diện: mô tả cách thức mà 1 khách hàng có thể sdung dk các dịch vụ mạng và cách thức các dịch vụ có thể truy cập đến.

# **Chương 3: Tầng ứng dụng.**

## 3.1: Các nguyên tắc của ứng dụng.

* Một số ứng dụng mạng: E-mail, web, instant messaging, Remorte login, chia sẻ file 2 P2P, trò chơi nhiều người sử dụng, điện thoại qua internet, hội nghị truyền hình thời gian thực, tinh toán song song.
* Tạo 1 ứng dụng mạng:
* VIết chương trình: chạy trên các end systems khác nhau và giao tiếp qua mạng, vd web: phần mềm web server giao tiếp với phần mềm trình duyệt.
* KO có phần mềm viết cho các thiết bị trong netword Core: các thiết bị trong netword Core ko thực hiện chức năng tại lớp ứng dụng,thiết kế này để sự phát triển ứng dụng nhanh.
* Kiến trúc ứng dụng:
* Client-server:

+ server: Host ở trạng thái luôn chạy, địa chỉ cố định, server farm(cụm máy chủ) để tăng khả năng phục vụ.

+ Client: truyền thong với server, có thể ko kết nối liên tục, có thể có địa chỉ ip động, ko giao tieps trực tiếp với nhau.

* Kiến trúc P2P:

+ server ko ở trạng thái luôn chạy.

+ Hệ thống cuối tùy ý truyền thông trực tiếp,

+ các Peer kết nối ko liên tục và thay đổi địa chỉ ip.

+ ưu điểm: khả năng co giãn cao

+ nhược: khó quản lý

* Hybird:

+ Lai giữa P2P và client-server

+ Napster:

* Truyền file P2P
* Tìm kiếm file tập trung:
* Các Peer đăng ký nội dung tại server trung tâm
* Các peer gửi yêu cầu server trung tâm xác định vị trí của nội dung.

+ Intant Messaging

* Giao tiếp giữa 2 user là P2P
* Quản lý tập trung vị trí của user:
* Đăng ký địa chỉ IP với server trung tâm khi kết nối
* User thông qua server trung tâm để tìm địa chỉ IP của đối tượng cần giao tiếp.
* Truyền thông giữa các tiến trình

+ Tiến trình: chương trình chạy trong 1 Host. Có 2 loại tiến trình: tiến trình client( khởi đầu quá trình truyền thông ) và tiến trình server(tiến trình đợi kết quả).

* Trong cùng host, 2 tiền trình giao tiếp sử dụng

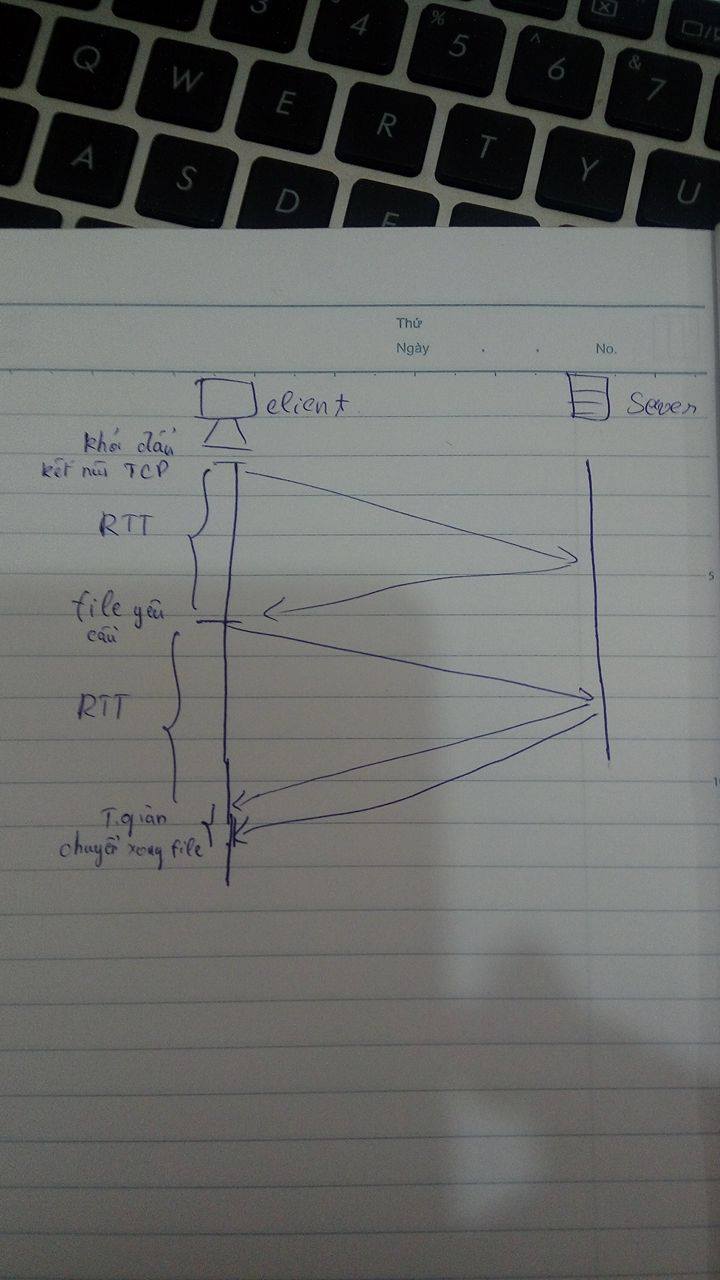
+ inter-process

+ communication( do OS định nghĩa)

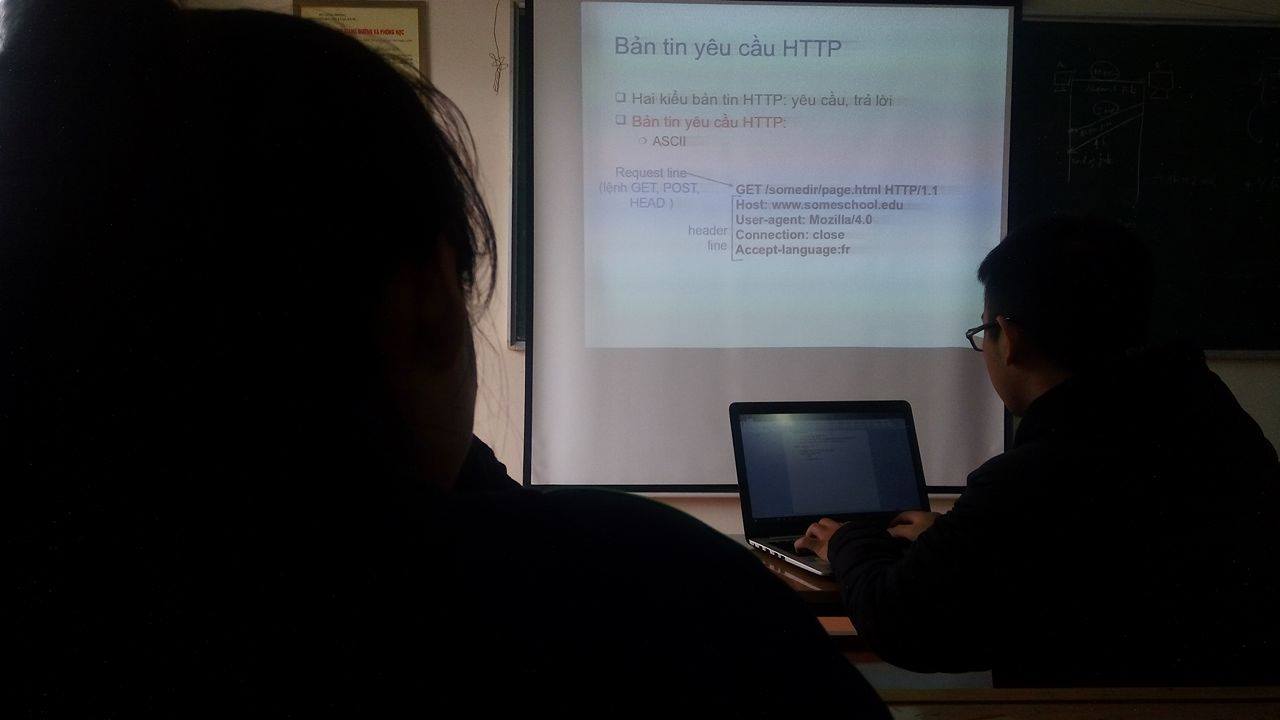
* Các tiến trình trong các host khác nhau giao tiếp bằng cách trao đổi các message
* Chú ý: các ứng dụng P2P có cả tiến trình client và tiến trình server
* Socket:
* Trình gửi nhận message tới/từ socket của nó.
* Socket tương tự như cửa ra vào
* Tiến trình gửi đẩy bản tin ra ngoài
* Dựa vào hạ tầng giao vận ở phía bên kia của cửa, nó mang message tới socket của tiến trình nhận.
* API: (1) Chọn giao thức giao vận, (2) : Gán gtri cho 1 vài tham số
* Các tiến trình đánh địa chỉ.
* Đối với tiến trình nhận message nó phải có 1 định danh
* 1 host có 1 địa chỉ IP duy nhất
* Địa chỉ IP của host ko định danh tiến trình
* Nhiều tiến trình có thể chạy trên cùng hos
* Định danh bao gồm cả địa chỉ IP và địa chỉ cổng gán cho ud trên host
* VD: gtri cổng của 1 số ứng dụng:
* HTTP server:80
* Mail server: 25
* Các dịch vụ giao vận mà ứng dụng cần
* Mất dữ liệu:
* 1 số ứng dụng (vd: audio) có thể chấp nhận 1 tỷ lệ mấy dữ liệu nào đó
* 1 ssos ứng dụng khác(vd: truyền file, telnet) đòi hỏi 100% dữ liệu truyền tin cậy.
* Thời gian:
* 1 số ứng dụng (vd: điện thoại internet, trò chơi tương tác) đòi hỏi độ trể thấp.
* Băng thông
* 1 số ứng dụng (vd: đa phương tiện) yêu cầu lượng băng thông tồi thiểu
* 1 số ứng dụng khác sử dụng băng thông chúng nhận đươc.

## 3.2: WEB và HTTP

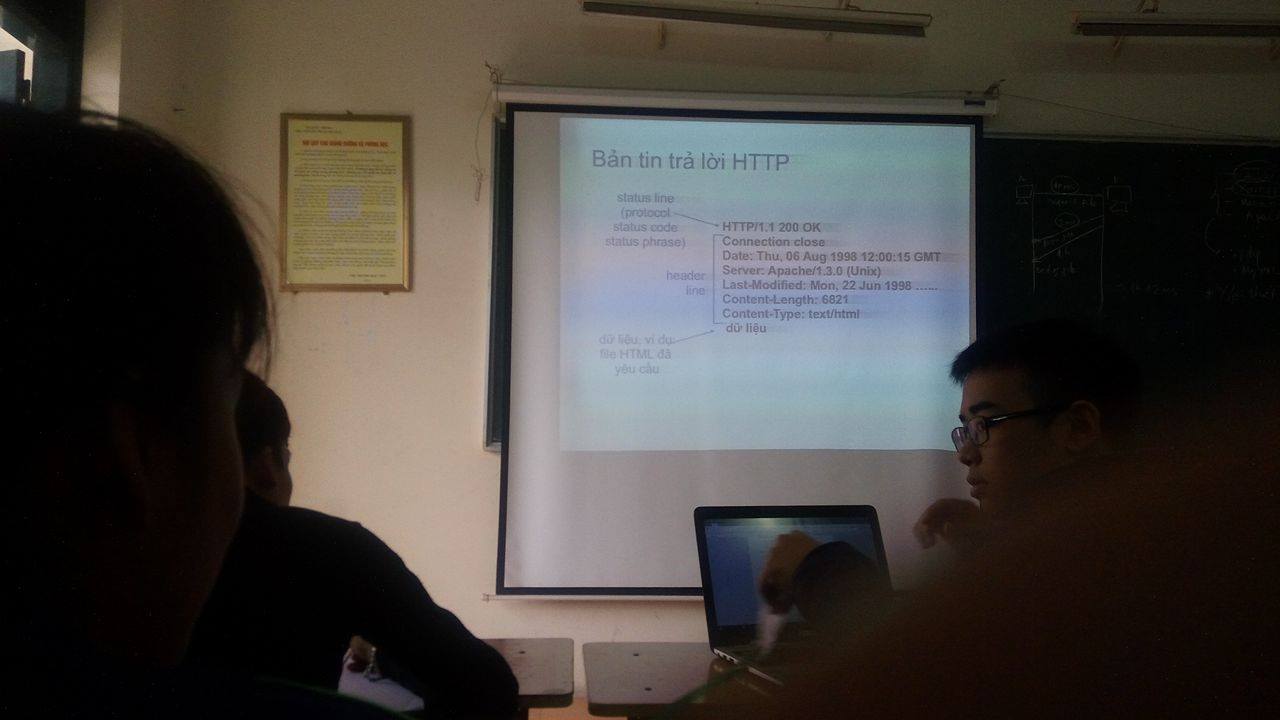
* Trang web chứa các đối tượng
* Đối tương có thể là fille HTML, ảnh LPEG, java applet, audio....
* Trang web chưa file HTML, chứa các đối tượng tham chiếu.
* HTTP: Hypertxt transfer protocol.
* Giao thức tầng ứng dụng của web
* Mô hình client/server
* Client: trình duyệt yêu cầu, nhận và hiển thi các đối tượng.
* Server: web server gửi các đối tượng trong trả lời.
* Sử dụng TCP:
* Client khởi đầu kết nối TCP(tạo socket) tới server, cổng 80
* Server chấp nhận kết nối TCP từ client
* Các bản tin HTTP(bản tin của giao thức tầng ứng dụng web) trao đổi giữa trình duyệt (HTTP client) và web server (HTTP server).
* HTTP là ko hướng trạng thái:
* Server ko duy trì thông tin về các yêu cầu của client trong quá khứ.
* Các giao thưc hướng trạng thái phức tập hơn:
* Lịch sử truy xuất phải được duy trì
* Nếu server/client lỗi , các trạng thái có thể ko thống nhất.
* Mô hình thời gian trả lời:
* RTT(Round- trip time): thời gian để gửi 1 gói tin từ client tới server và nhận gói dữ liệu đầu tiên trả về.
* Thời gian trả lời:
* 1 RTT để khởi tạo kết nối TCP
* 1 RTT cho yêu cầu HTTP và nhận byte đầu tiên của trả lời HTTP
* Thời gian truyền 2 file: total = 2RTT + transmit time
* Mô hình truyền file từ client đến server:



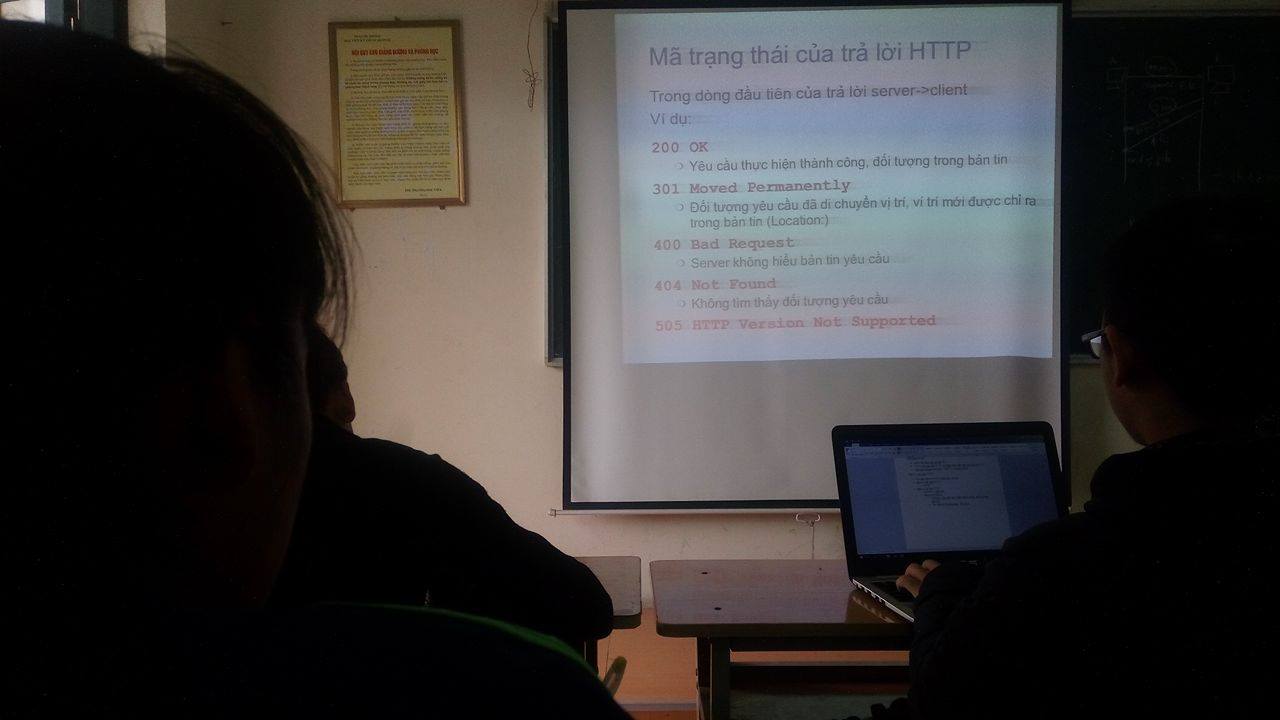
* Bản tin yêu cầu HTTP
* Hai kiểu bản tin của HTTP: yêu cầu, trả lời
* Bản tin yêu cầu HTTP
* ASCII
* Cấu trúc: ảnh bên dưới.....

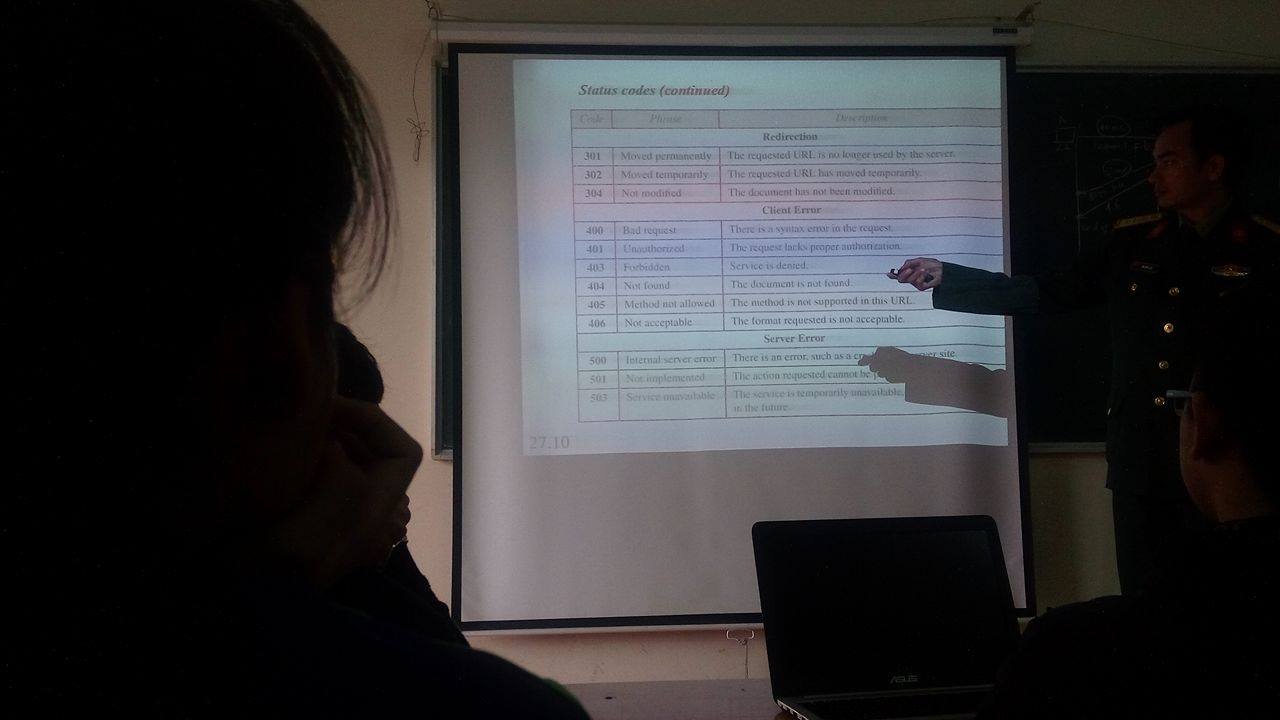


* Bản tin trả lời HTTP:

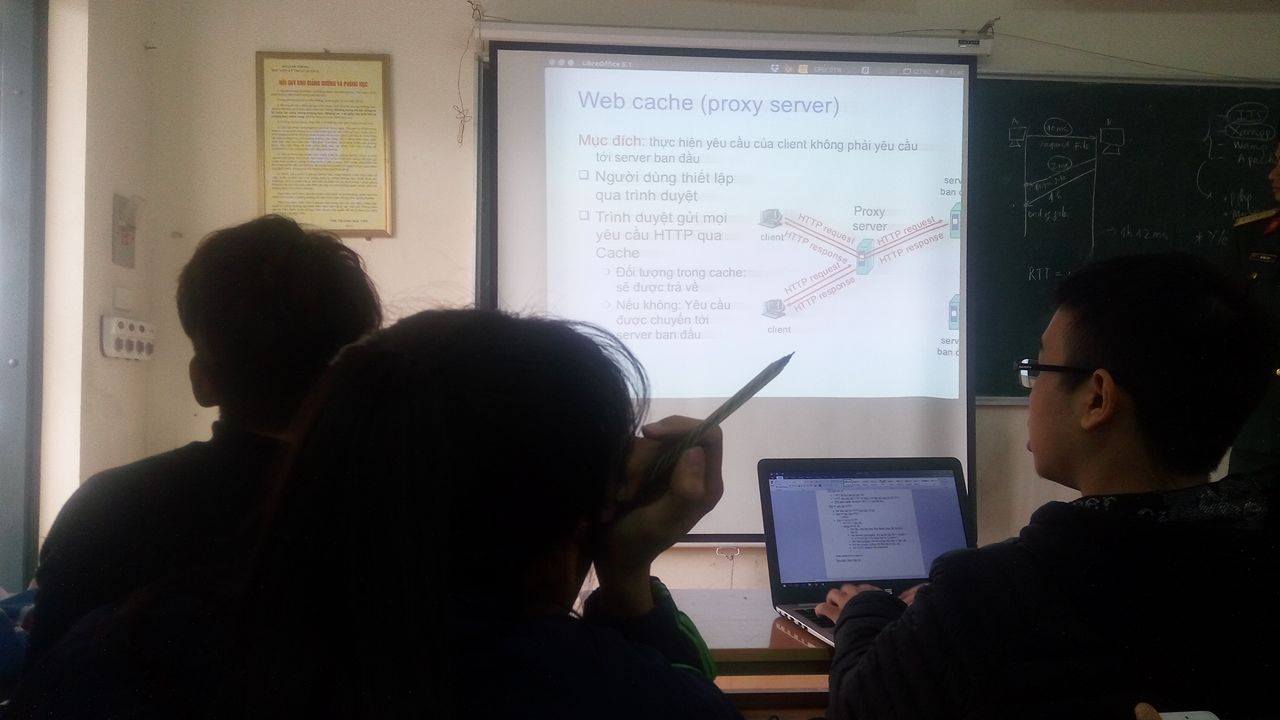


* Mã trạng thái của trả lời HTTP





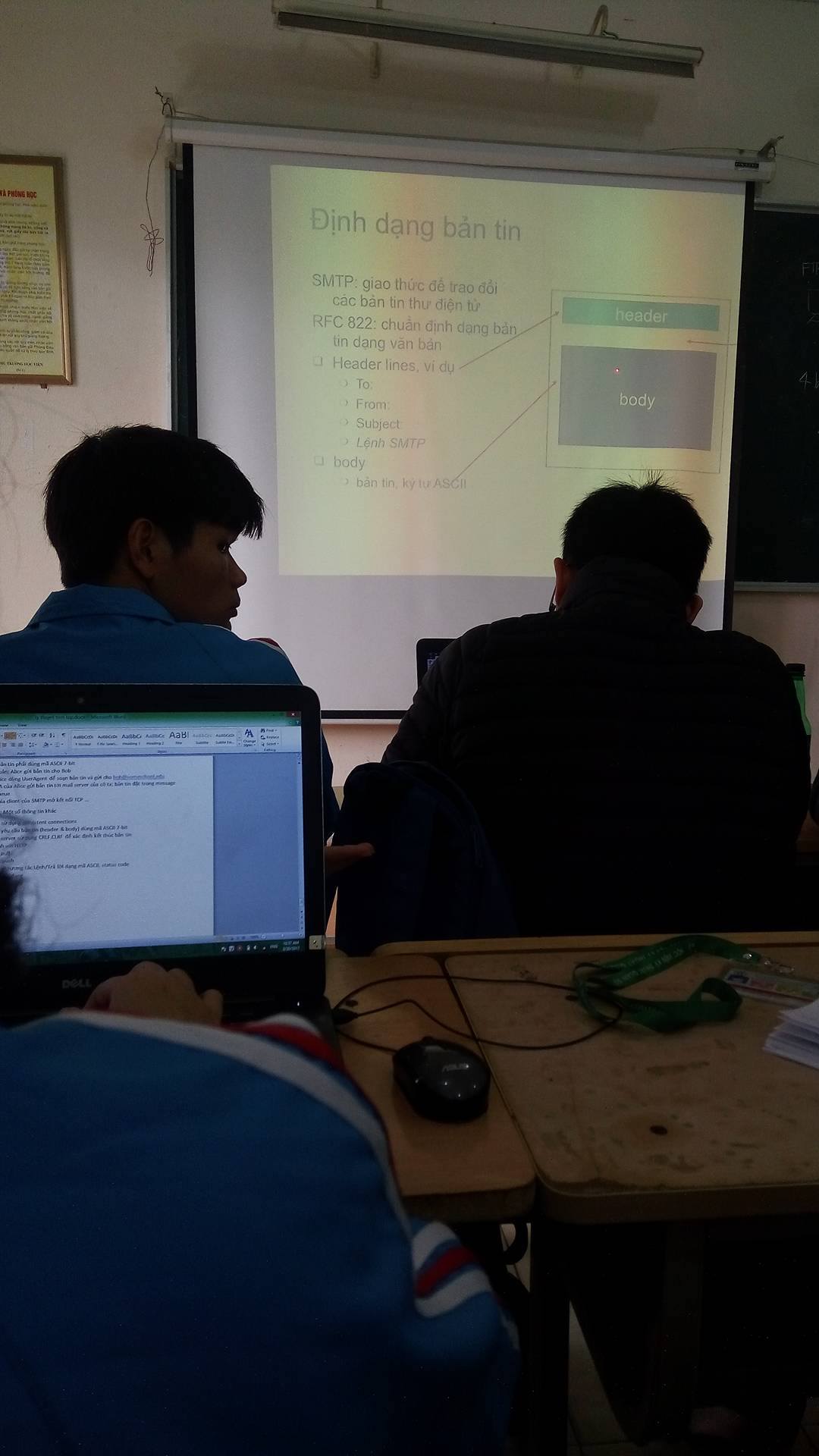
* Web cache



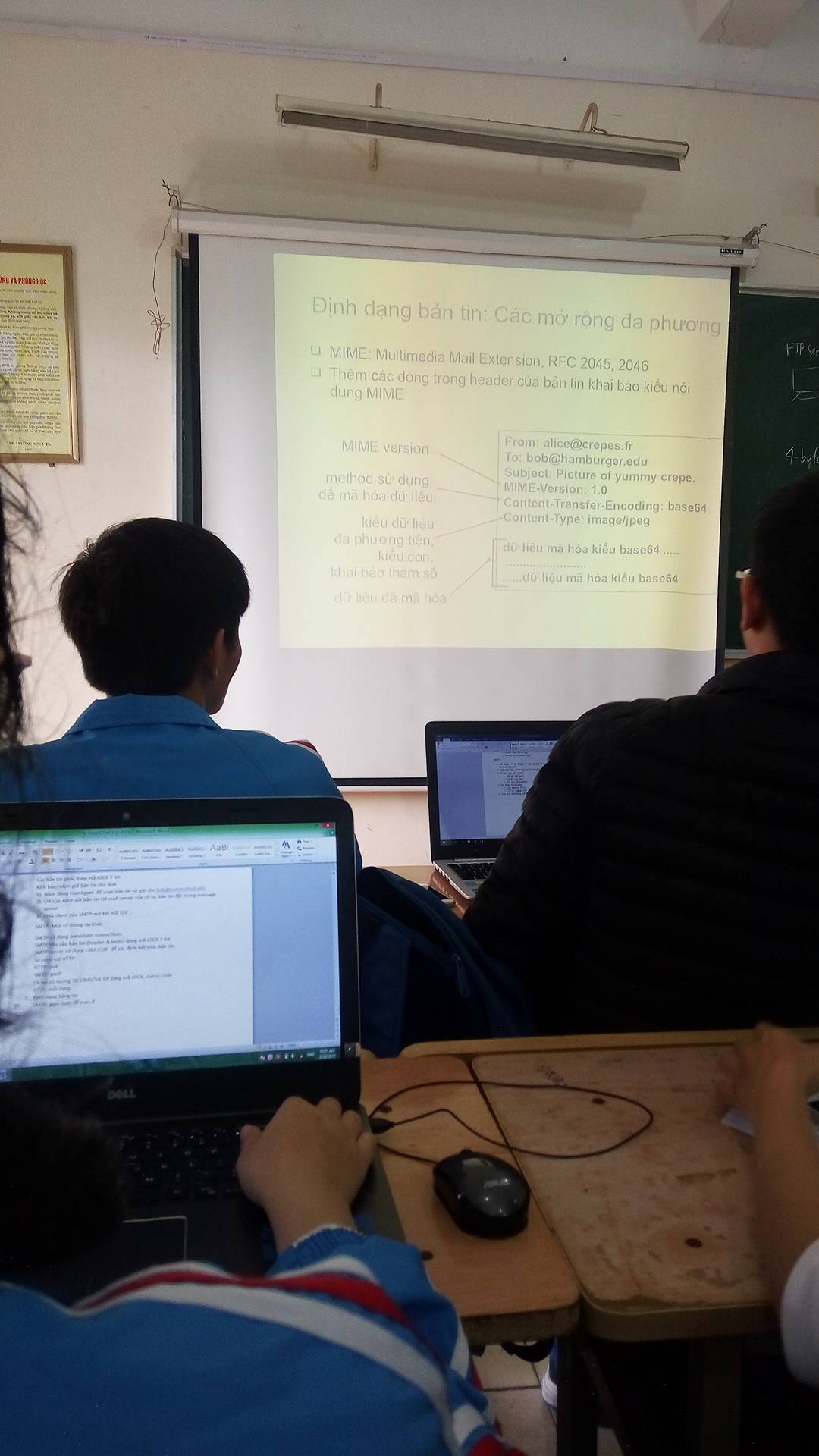
* Cache hoạt động như cả client và server
* Thông thường cache dk cài đặt bởi ISP (truongf đại học, công ty...)
* Lợi ích của web cache:
* Giảm thời gian trả lời cho yêu cầu của client.
* Giảm lưu lượng trên đường truy cập của tô chức.
* GET có điều kiện
* Mục đích: ko gửi đối tượng nếu như cache có phiên bản cập nhật
  1. FTP server
* FTP: file transfer Protocol
* Truyền file tới/từ host ở xa
* Mô hình clienr/server
* FTP: kết nối dữ liệu và điều kiện riêng biêt
* FTP client liên lạc với FTP server tại cổng 21, chỉ định dùng 010/. làm giao thức vận
* Client giành ủy quyền qua kết nối điều khiển
* Client xem thư mục từ xa bằng cách gửi lệnh qua kết nối điều khiển
* Khi server nhận 1 lệnh truyền file, server mở 1 kết nối dữ liệu TCP tới client
* Sau khi truyền file, server đóng kết nối
* Lệnh và trả lời của FTP
* Ví dụ 1 số lệnh:
* Gửi văn bản mã ASCII qua kênh điều khiển
* USER username
* PASS password
* LIST trả về 1 danh sách các file trong thư mục hiện tại
* RETR filename lấy file
* STOR filename đưa file lên remote host
* Ví dụ 1 số mã trả về:
* Status code và status phrase(như HTTP)
* 331 Username ok, đòi hỏi mật khẩu
* 125 kết nối dữ liệu đã mở, bắt đầu truyền
* 425 ko thể mở kết nối dữ liệu
* 452 lỗi ghi file.

## **3.4: Giao thức mail:**

* Electronic Mail
* Gồm 3 thành phần chính:
* User agent
* Mail server
* Simple mail transfer protocol: SMTP
* User Agent
* Còn gọi là Mail Reader
* Soạn , sửa, đọc bản tin mail
* Ví dụ: Eudora, Outlook,Netscape Messenger
* Các bản tin gửi tới và gửi đi
* Mail server
* Mailbox chứa các bản tin thư điện tử cho người sử dụng
* Message queue cua các bản tin thư điện tử gủi đi
* SMTP proticol giữa các mail server để gửi các bản tin thư điện tử
* Client: mail server gửi
* Server: mail server nhận.
* SMTP
* Sử dụng TCP để truyền tin cậy các bản tin thư điện tử từ client tới server, cổng 25.
* Gửi trực tiếp; server gửi tới server nhận
* Ba pha của việc truyền:bắt tay, truyền các bản tin, kết thúc
* Sự tương tác lệnh/ trả lời:
* Lệnh: văn bản mã ASCII
* Trả lời: status code và status phrase
* Các bản tin phải dungf mã ASCII 7-bit.
* So sánh SMTP và HTTP
* HTTP: pull(đẩy)
* SMTP:push(kéo)
* Cả hai có tương tác lệnh/trả lời dạng mã ASCII, status code
* HTTP: mỗi đối tượng được đóng gói trong bản tin trả lời
* SMTP: nhiều đối tượng được gửi tới bản tin có nhiều phần.
* Ví dụ bản tin: như trong ảnh



* Nội dung bản tin: chèn ảnh(các mở rộng đa phương)



* POP3 và IMAP
* Bob ko thể đọc lại các thư điện tử nếu bob chuyển sang client khác
* Chế độ “ tải và giữ” sao chép các bản tin trên các client khác nhau
* POP3 ko lưu trạng thái giữa các phiên bản
* IMAP:
* Giữ tất cả các bản tin tại 1 chỗ:server
* Cho phép người sử dụng tổ chức các bản tin vào các thư mục
* IMAP giữ trạng thái người sử dụng qua các phiên :
* Tên của các thư mục và ánh xạ giữa các định danh của bản tin và tên thư mục.

3.5: DNS(Domain Name System)

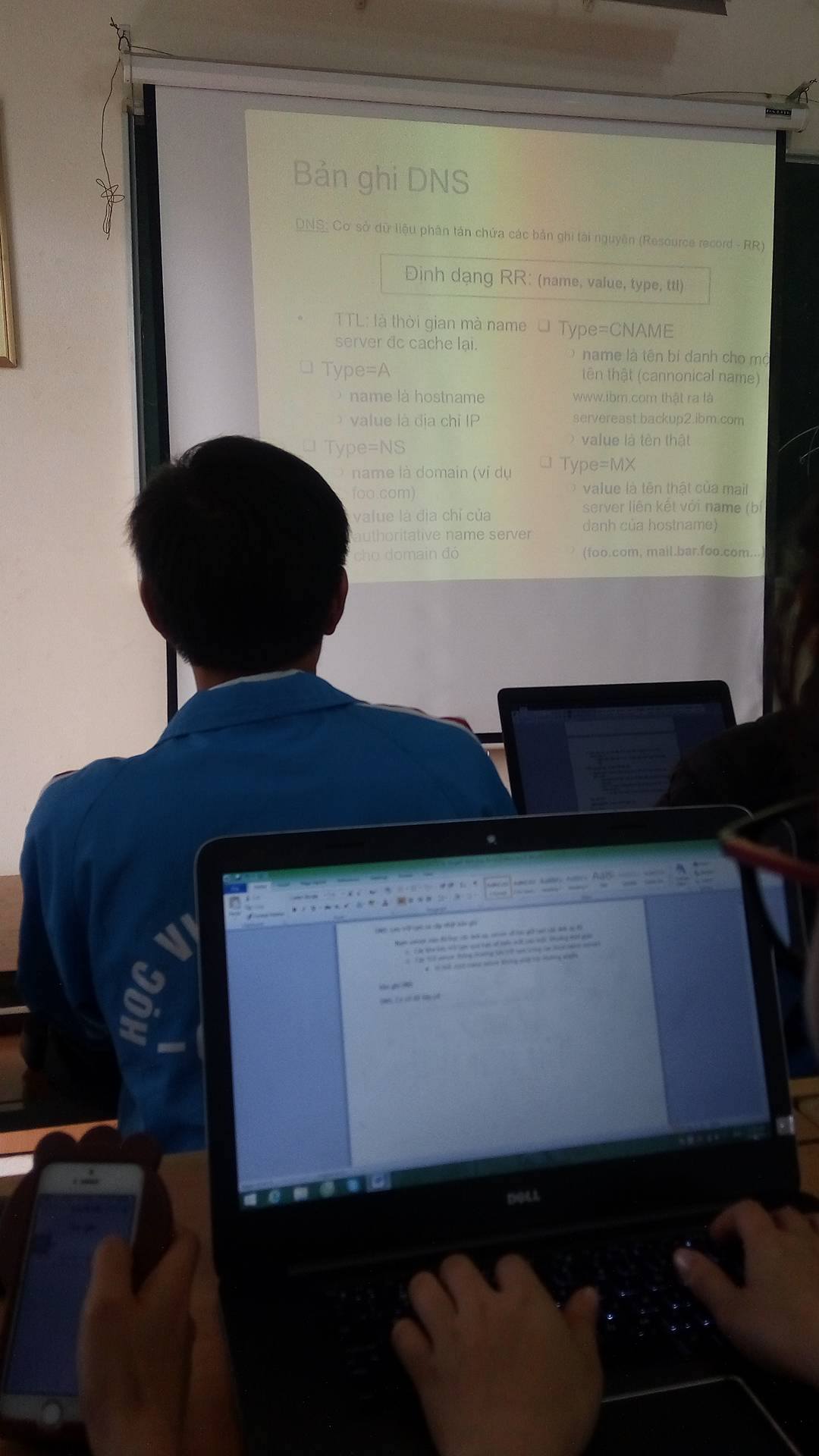
* Con người : có nhiều định danh: CMT, hộ chiếu, tên
* Host, router trên internet:
* Địa chỉ IP(32 bit) dể đánh địa chỉ các đơn vị dữ liệu
* Hệ thống tên miền(DNS):
* Cơ sở dữ liệu phân tán được thực hiện phân cấp bởi nhiều name server
* Giao thức tầng ứng dụng để host, router, name server giao tiếp dịch giữa địa chỉ và tên.
* Chú ý: chức năng cơ bản của internet, thực hiện bỏ giao thức ứng dụng
* Sự phức tạp tại netword edge
* DNS:
* Các dịch vụ DNS:
* Dịch tên host sang địa chỉ IP
* Bí danh cho Host
* Bí danh cho mail server
* Phân tải
* Tại sao ko dùng DNS tập trung:
* 1 điểm lỗi
* Khối lượng lưu lượng
* Cơ sở dữ liệu tập trung xa
* Bảo trì
* Ko co giãn.
* Cơ sở dữ liệu phân cấp và phân tán:



* DNS: Root name server:
* Local name server ko trả lời được thì sẽ liên lạc với Root name server.
* Liên lạc với authoritative server nếu có ko có ánh xạ tên
* Liên lạc với
* Lấy ánh xạ
* Tránh ánh xạ về cho local name server
* TLD và
* Top-lever domain(TLD) server: có vai trò đối với com, org, net, edu.... và tất cả các miền quốc gia mức trên cùng uk, fr, ca,jp,...
* Authoritative DNS server: DNS server của các tổ chức cung cấp ánh xạ au.. tive hostma thành địa chỉ IP cho server của tổ chức (vd: web, email)
* Có thể duy trigf bởi tổ chức hoặc nha cung cấp dịch vụ.
* Local Name Server
* Ko hoàn toàn thuộc vào phân cấp: còn gọi là “ default server”
* Khi 1 host tạo truy vấn DNS, truy vấn dduocj gửi tới local DNS server của nó
* Hoạt động như 1 proxy, chuyển tiếp query vào trong phân cấp.
* Ví dụ: ảnh ở dưới



* DNS: lưu giữ tạm và cập nhật bản ghi
* Name server nào đó học cách ánh xạ, server sẽ lưu giữ tạm các ánh xạ đó.
* Các khe lưu trưc tạm quá hạn sẽ biến mất sau 1 khảng thời gian
* Các TLD server thông thường lưu trữ trong các local name server
* Vì thế, root name server ko phải hỏi thường xuyên.
* Bản ghi DNS:



1. Tầng ứng dụng Transport(TCP và UDP)

- Mục đich: Giúp cho tất cả các ứng dụng trên 1 thiết bị có thể nhận đúng đk dữ liệu trên,chia các gói tin....

- Tính chất của TCP: bắt tay(3bước), segment(truyền dữ liệu theo tuần tự), window size(điều khiển lỗi, luồng) .

|  |  |  |
| --- | --- | --- |
| TCP | UDP | Both |
| FTP | VoIP | DNS |
| SMTP | IPTV |  |
| HTTP | TFTP |  |
| Telnet | DHCP |  |
|  | SNMP |  |

20

Bytes

|  |  |  |  |
| --- | --- | --- | --- |
| Bit(0) bit(15) | | | bit(16) bit(31) |
|  | | |  |
| Source Port(16) | | | Destination Port(16) |
| Sequence Numbr(32) | | |  |
| Acknowledgement | | | Number (32) |
| Header Length(4) | Reserved(6) | Control Bits(6) | Window(16) |
| Checksum(16) | | | Urgent(16) |
| Options (0 or 332 if any) | | |  |
| Application Layer Data (Sizevaries) | | |  |

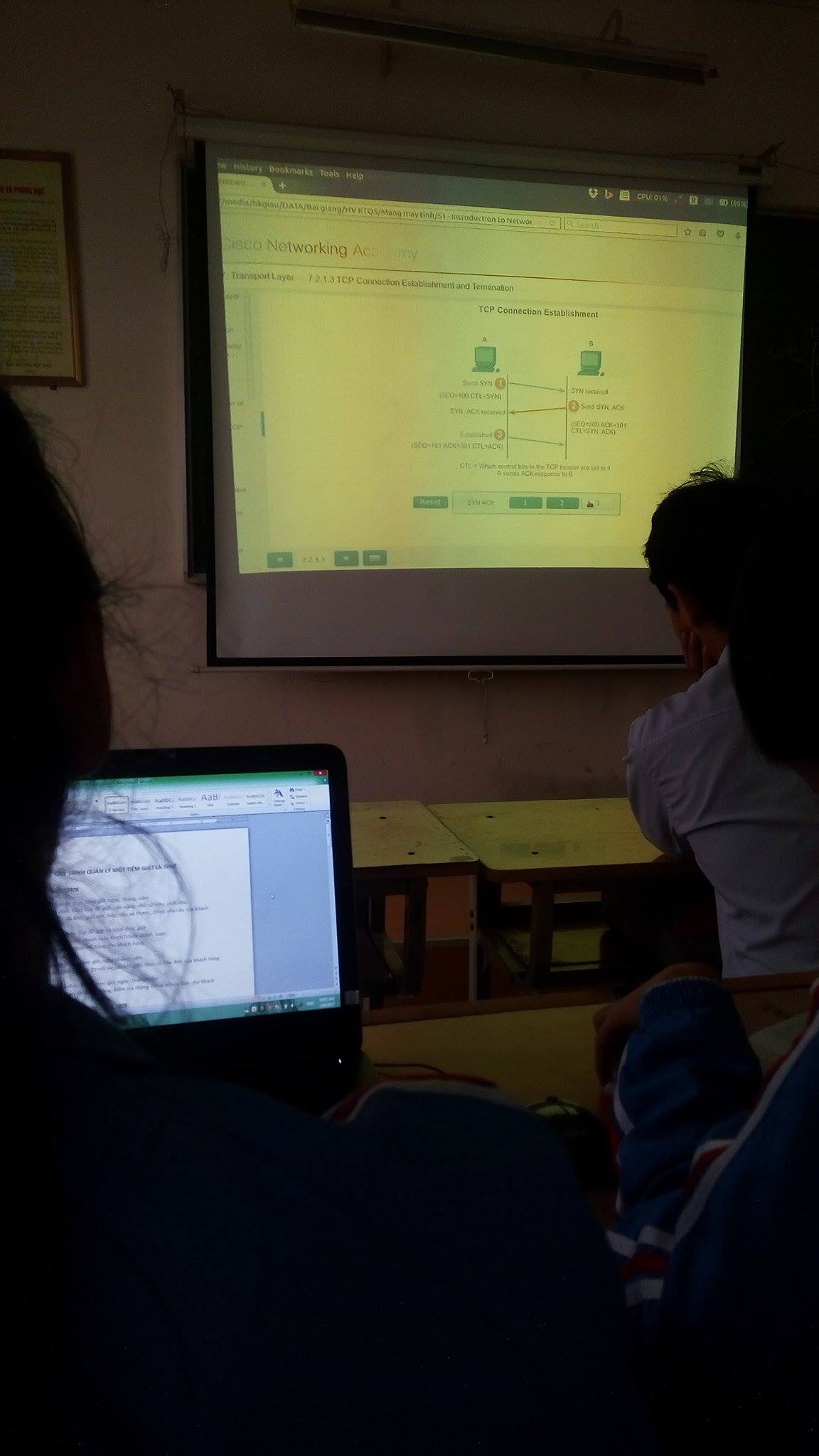
|  |  |
| --- | --- |
| Bit(0) Bit(15) bit(16) Bit(31) | |
| Source Port(16) | Destination Port(16) |
| Length(16) | Checksum(16) |
| Application Layer Data (Sizevaries) | |

|  |  |
| --- | --- |
| Port Number Range | Port group |
| 0 to 1023 | Wel-known Ports |
| 1024 to 49151 | Registered Port |
| 49152 to 65535 | Private and/or Dynamic Ports |

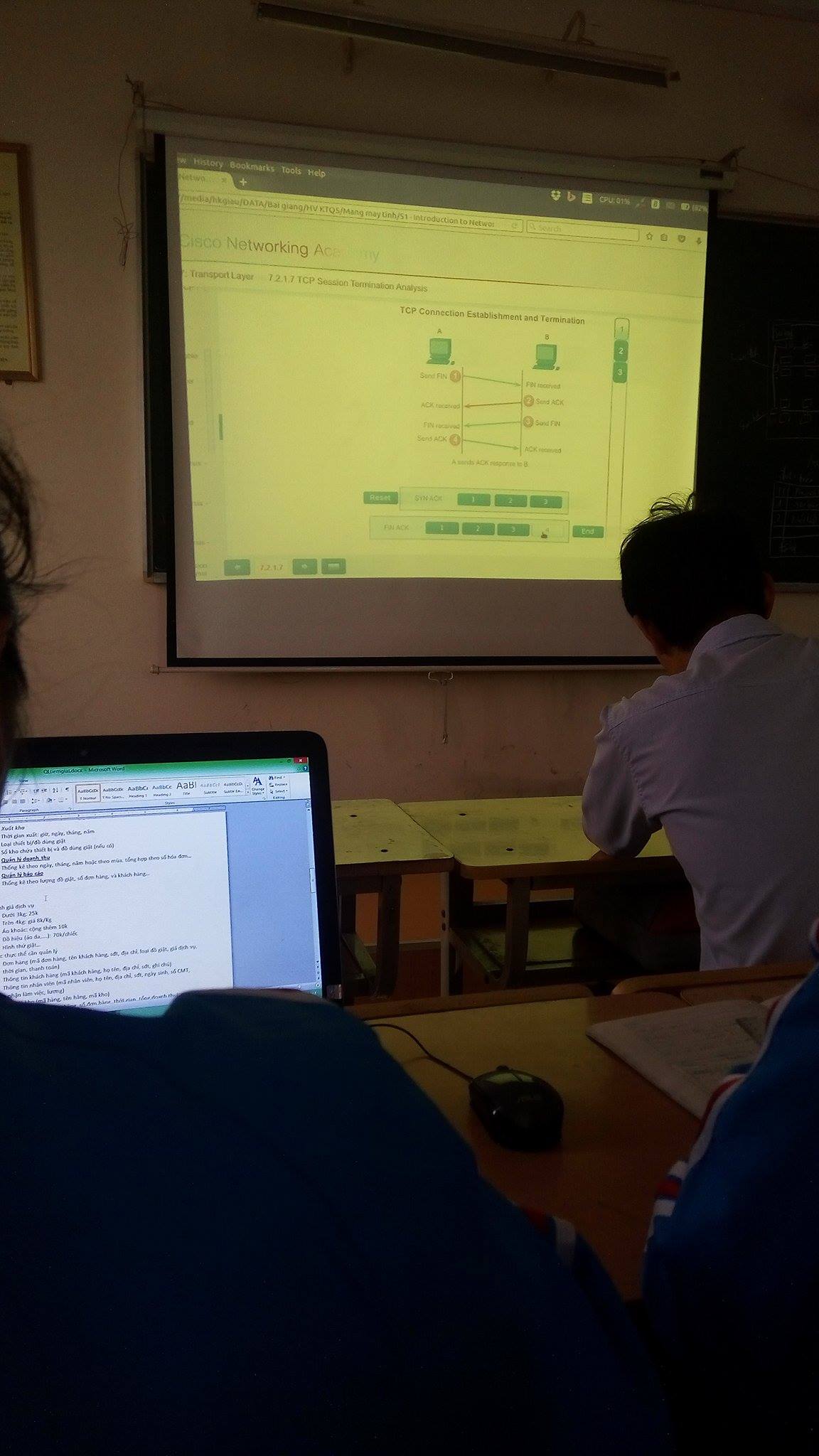
|  |  |
| --- | --- |
| Lengend | |
| Registered TCP Port | Well-know UDP Port |
| 1863 MSN Messenger | 21 FTP |
| 2000 Cisco SCCP (VoIP) | 23 Telnet |
| 8008 Altemate HTTP | 25 SMTP |
| 8080 Altemate HTTP | 80 HTTP |
|  | 143 IMAP |
|  | 194 |

|  |  |
| --- | --- |
| TCP(ko có hướng kết nối) | UDP( có hướng kết nối) |
| Session Estabishment | connectionless |
| Ordered Delivery | No Ordered Delivery |
| Flow Control | No Acknowledgement... |
| Guaranteed Delivery | Less Overhead |
| Sequenced Message Segments | Fast Transmission Requirements |

* Mô hình thực hiện bắt tay 3 bước của TCP:



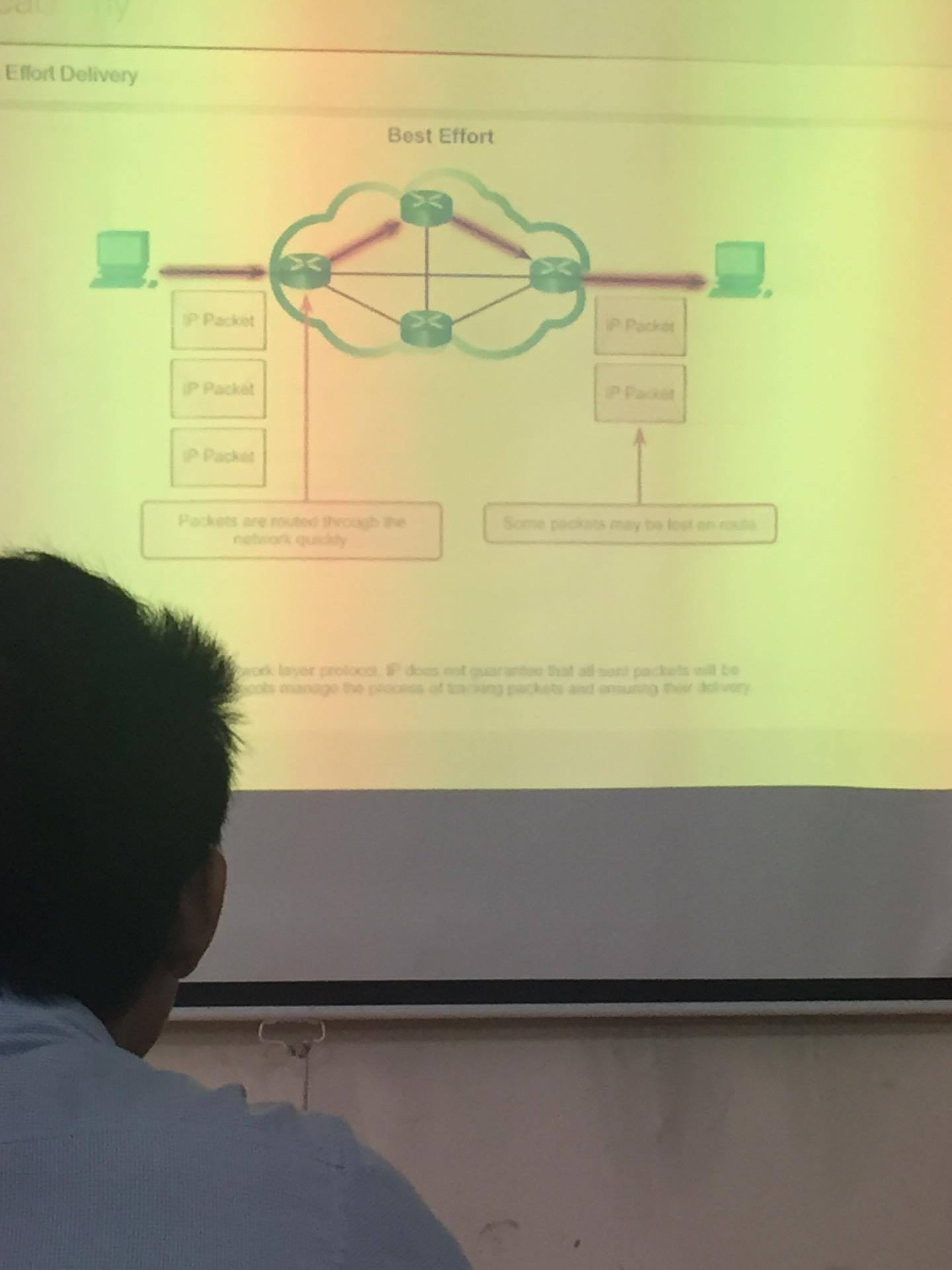
* Sau khi bắt tay xong rồi thì TCP sẽ truyền dữ liệu đi.
* Quá trình thực hiện ngắt kết nối: (chèn ảnh)



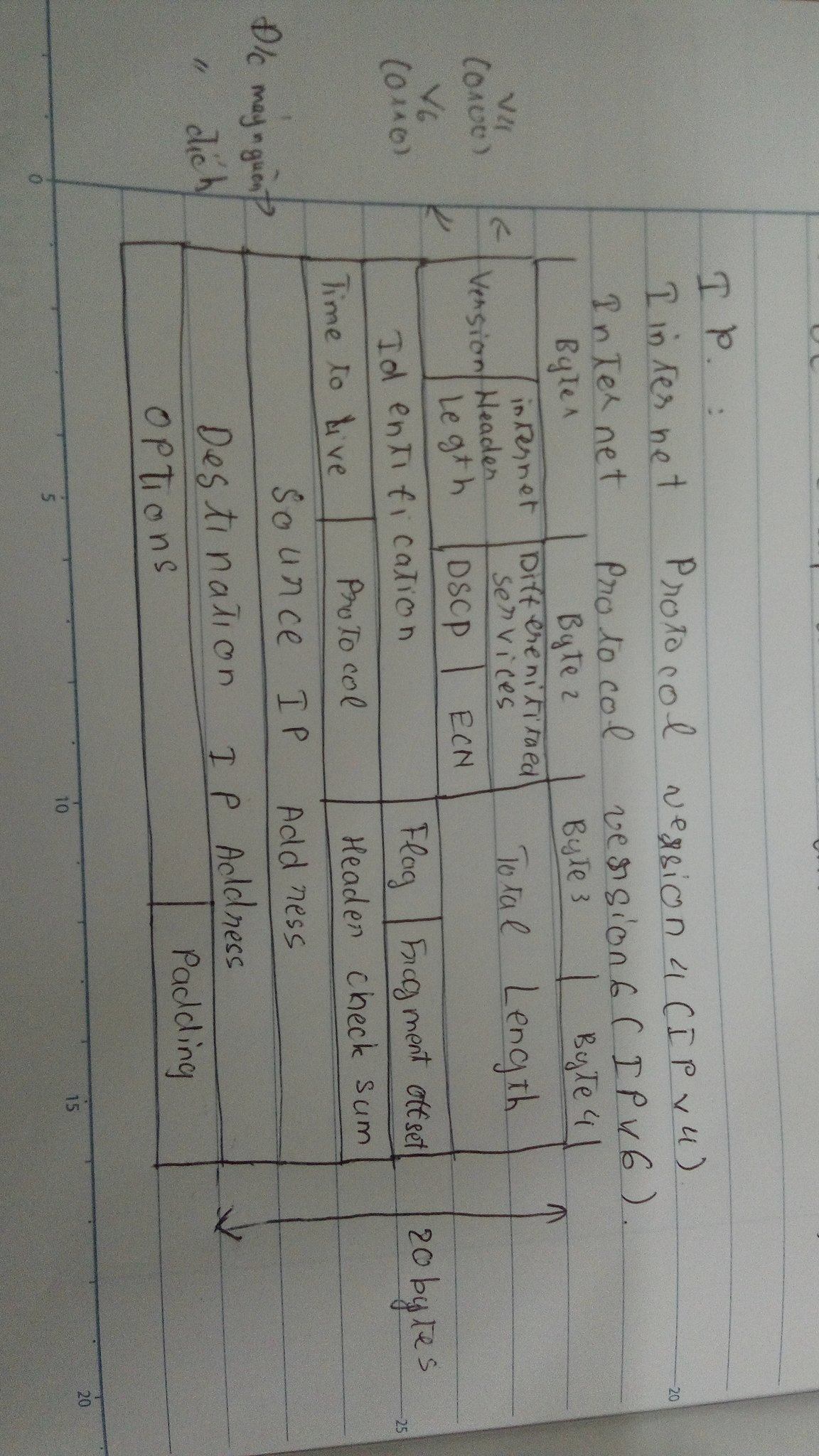
* SizeDữ liệu đk chia nhỏ thành các segment,các segment tung lên mạng, nhưng khi đó lại đk sắp xếp ko theo thứ tự. TCP có nhiệm vụ sắp xếp lại theo đúng thứ tự ban đầu.
* Cấu trúc của segment(chèn ảnh)
* 1 khoảng dữ liệu mà máy nguồn có thể chuyển đổi trc khi ACK nhận đk gọi là window Size(kích thước cửa sổ). Window size nằm trong TCP Header nó cho phép điều khiển luồng và điều khiển lỗi.
* Quản lý max dữ kiệu ko quá trình truyền:
* Để quản lý vc max dữ liệu và truyền lạ thì TCP nó tạo ra 1 bộ đếm thời gian, nếu hết thời gian mà bên nhận vẫn chưa nhận đk gói tin thì bên gửi sẽ gửi gói tin khác.
* UDP: truyền dữ liệu ko tin tưởng , ko tin tưởng ko có thiết lập bắt tay trc khi truyền dữ liệu. (chèn ảnh UDP:Connection...)
* Mô hình datagram là của UDP
* Tổng kết lại tầng ứng dung: (chèn ảnh)
* [data + port] 🡪 segment(TCP)/datagram(UDP).
* [data +port + IP] 🡪 packet
* Có 2 loại IP : version 4 và version 6.
* Nhiệm vụ của tầng ứng dụng:
* Đánh dấu địa chỉ cho các thiết bị đầu cuối
* Đóng gói thành các packet
* Định hướng các gói tin
* Giải nén ra để trả về cho tầng trên.
* 1 gói tin gồm: địa chỉ nguồn, địa chỉ đích, số thứ tự ngẫu nhiên.

# B\_ Tầng netword( tầng mạng).

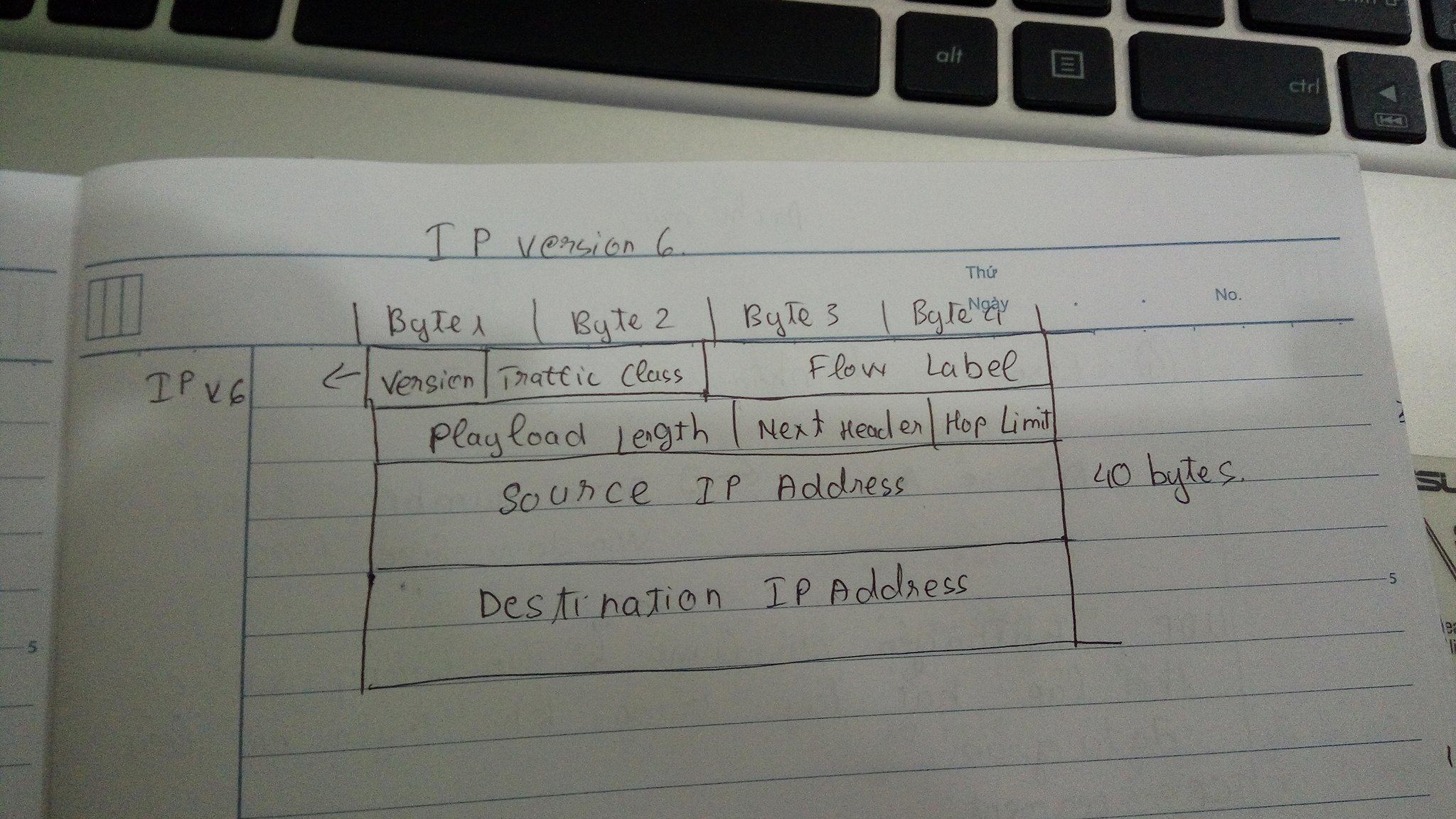
* Nỗ lực truyền gói tin tới mạng: làm sao để gói tin chuyển đi nhanh nhất.(chèn ảnh: best ef..)



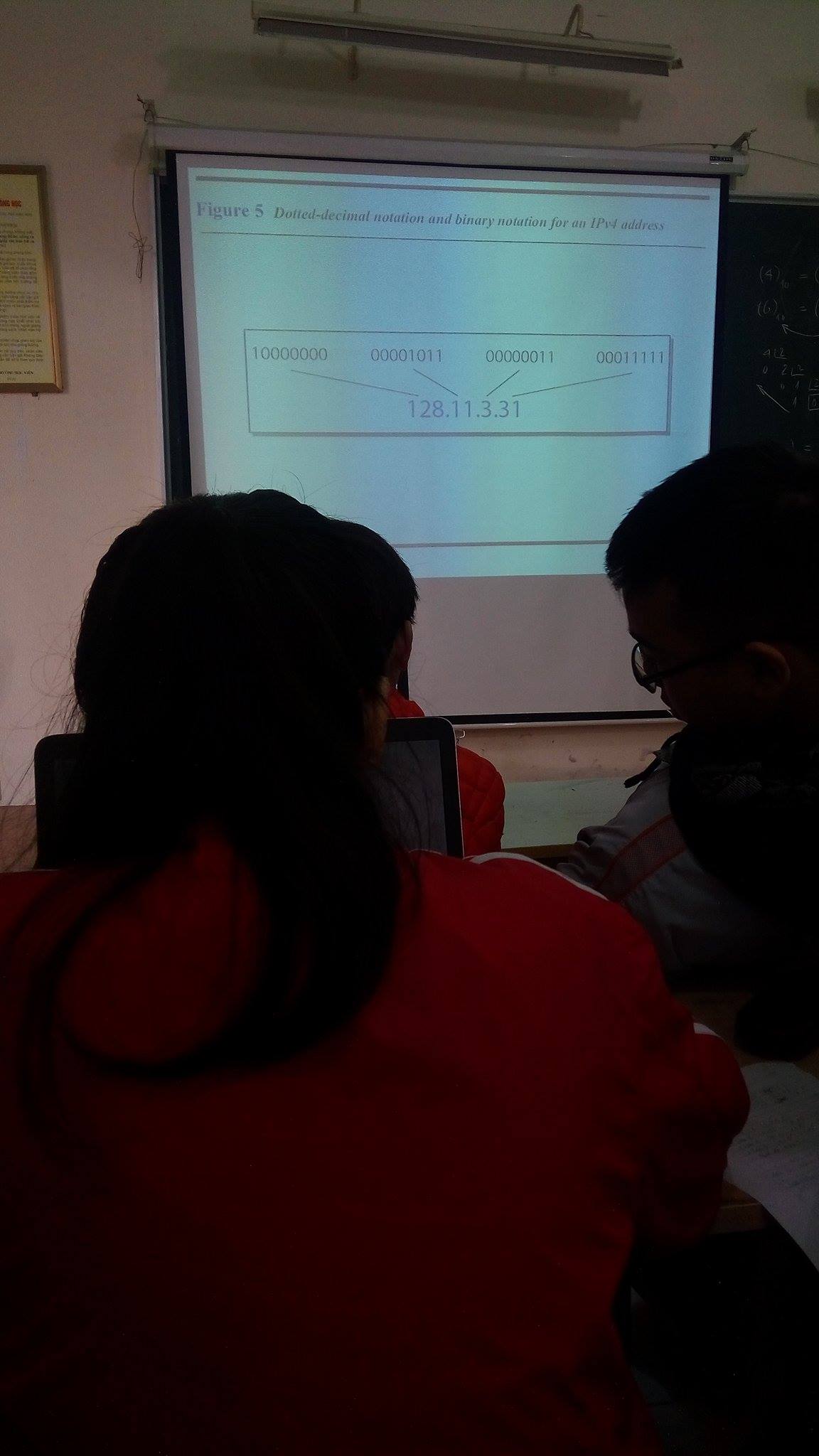
* Khi cài địa chỉ mạng IPv4 gồm:
* IP:192.168.1.15
* Subrd mask:255.255.255.0
* Default gateway: 192.168.1.1
* Ko phụ thuộc vào đường truyền.
* Cấu trúc header của Ipv4: (chèn ảnh : 6.1.3.1 Ipv4 packet header)



* Total lenght: tổng chiều dài(32 byte).
* Time- to- live: thời gian sống.
* Identification: định danh
* IPv6:



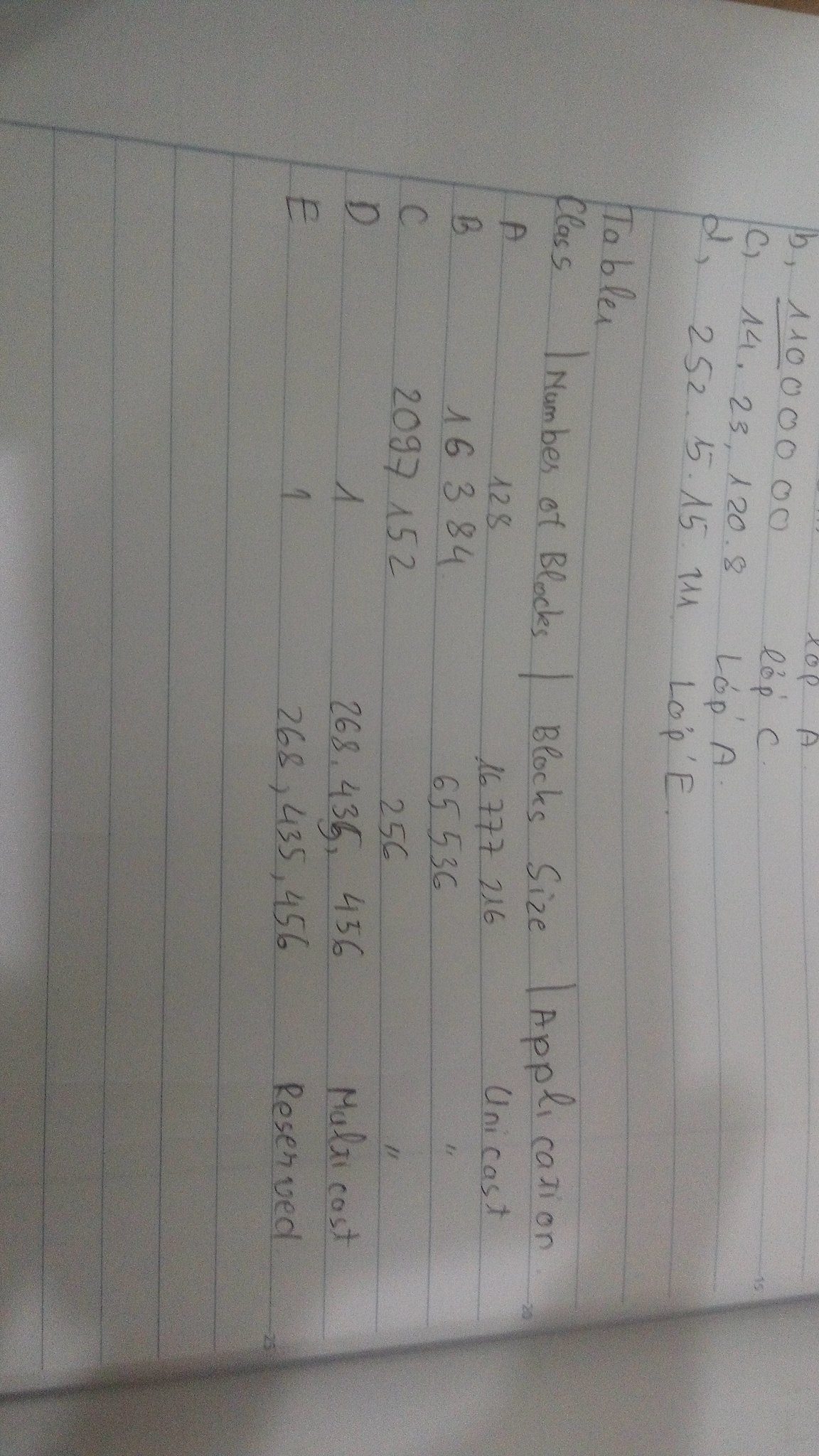
* IPv4(2^32) là: chuỗi chỉ 32 bit duy nhất cho định nghĩa vc kết nối giữa các máy tính router cho việc kết nối đến internet.
* Để định đanh đk các địa chỉ IP ta thêm dấu (.) vào nhóm 8 bit



* Phân lớp cho địa chỉ IP
* Đối với Ipv4 Phân thành 5 lớp A, B, C, D, E
* Địa chỉ IP gồm có 2 phần : Net ID(8 bit) và Host ID(24 bit)
* Net ID: để đinh danh cho mạng máy tính
* Host iD: máy tinh tham gia vào mạng đó
* (Chèn ảnh:)



* Màu vàng là next, màu trắng là host
* Ví dụ tính địa chỉ IP thuộc lớp nào:
* a. 00000001 00001011 00001011 11101111 🡪 thuộc về lớp A
* 11000001 10000011 00011011 11111111 🡪 thuộc về lớp C
* 14.23.120.8 🡪 thuộc lớp A
* 252.5.15.111 🡪 thuộc về lớp E
* Chèn ảnh(table1)



* ứng dụng:
* Unicast: dùng để giao tiếp giữa các máy(các mạng nối tiếp với nhau).
* Multicast : mạng quảng bá truyền cho các máy
* Reserved: mạng dự trữ
* Việc phân chia các địa chỉ IP rất lãng phí các địa chỉ IP 🡪 trong ipv4 1 block đk định danh bằng x.y.z.t/n (trong đó: n là mặt nạ)
* Phần nào mac =1 là net, phân nào mac id =0 chính là phần host id

Chương 3: Datalink layer

* Nhiệm vụ:
* Nhận và sửa lỗi
* Khung dữ liệu
* Điều khiển luông điều khiển lỗi
* Điều khiển đa truy cập
* Cấu trúc:
* Đóng gói và giải nén gói
* Gắn địa chỉ mac
* Điều khiển lỗi
* Điều khiển luồng
* Chèn ảnh(data link layer)
* Errors contro
* Phát hiện và sửa lỗi.
* Singgle- bit errors: 1 bit lỗi
* Burst: nhiều bit lỗi.
* Redundancy:
* Để phát hiện lỗi và sửa lỗi chúng ta thêm bit dư thừa.
* Coding:mã sửa lỗi
* Thêm dữ liệu dư thừa: gồm có mã khối:xem các dữ liệu thành các khối, mã vòng:xem dữ liệu như 1 dãy bit và xem 1 mã trên 1... .
* XOR Operation: (phép toán) ☹ ☺
* Hai bit giống nhau thì bằng 0,. Hai cái khác nhau thì bằng 1.
* Block coding: (độ dư thừa dữ liệu)
* Chia dữ liệu thành k bit block
* Sau đó cộng thêm r
* n = k +r.
* Notes:
* Phát hiện đk lỗi khi lỗi đó đã dk quy định sẵn
* Một số kỹ
* Kiểm tra lỗi kiểu cân bằng(chăn là 0, lẻ=1)
* Mã kiểm tra vòng
* Tính ra tổng để kiểm tra.
* Parity check:nếu số lượng bit 1 là số lẻ thì bit đk thêm vào là bit 1, nếu là số lẻ thì add vào là bit 0. Ví dụ: 1011🡪10111; 1010🡪10100
* Thuật toán: chèn ảnh: parity=-check: encoding/...
* Performance ò parity ..:
* Chỉ có thể kiểm tra đk 1 bit lỗi, nếu từ 2 trở lên thì ko thể.
* 2D parity check: (chèn ảnh)
* Nút mạng và liên kết:
* Chức năng:
* Đóng gói
* Điều khiển luồng
* Điều khiển lỗi...
* Phát hiện và sửa lỗi
* Mã chẵn lẻ:
* Mã đơn: chỉ phát hiện đk 1 bit lỗi
* Mã 2 chiều phát hiện đk 3 bit lỗi
* Mã CRC: (Cryclic Redundancy Check: mã vòng)
* Dữ kiệu đk xem như 1 số nhị phân: D
* Chọn 1 chuỗi r+1 bit, G
* Tìm 1 chuỗi R độ dài r bit, sao cho chuỗi ghép cảu D và R là một số nhị phân chia hết cho G
* Vi dụ : chèn ảnh ( CRC: cách tìm R)
* G càng dài thì mã CRC phát hiện lỗi càng hiệu quả
* CRC đk sử dụng rộng rãi trong thực tế...
* Các kỹ thuật kiểm tra lỗi:
* Đảm bảo truyền tin tincaapj trên môi trường truyền tin ko tin cậy
* Điều kiện
* Các loại lỗi:
* Mất khung dữ liệu
* Khung dữ liệu bị lỗi
* Thông báo lỗi bị mất
* Kiểm soát truy nhập đường truyền:
* Có 2 loại :
* Điểm nối điểm
* Quảng bá
* Các phuơng pháp chia kênh
* Chia theo tần số FDMA
* Theo thời gian TDMA
* Thep mạng CDMA
* Kiểm soát luồng dữ liệu:
* Đảm bảo trạm nguồn ko làm quá tải trạm dịch
* Trạm đích
* Lưu trữ các khung dữ liệu trong bộ nhớ đệm
* Thực hiện 1 số thao tác trc khi chuyển dữ liệu lên tầng trên
* Bộ nhớ đệm có thể bị đầy, dẫn đến mất khung dữ liệu
* Ko đặt vấn đề lỗi truyền tin
* Các khung dữ liệu luôn luôn đk truyền chính sác
* Giải pháp:
* Cơ chể dừng và chờ
* Nguyên tắc: nguồn gửi 1 khung dữ liệu; Đích nhận khung dữ liệu xử lý sau đó thông báo sẵn sàng nhận các khung dữ liệu tiếp theo bằng 1 thông báo nhận; Nguồn chờ đến khi nhận đk thông báo nhận truyền tiếp khung dữ liệu tiếp theo.
* Ưu điểm: đơn giản, đặc biệt thích hợp với các khung dữ liệu lớn
* Nhược điểm: với các khung dữ liệu nhỏ, thời gian sử dụng đường truyền bị lãng phí; ko thể sử dụng các khung dữ liệu lớn 1 cách phổ biến.
* Cơ chế cửa sổ trượt
* Nguyên tắc: gửi nhiều khung dữ liệu để giảm thời gian chờ; các khung đã gửi đi chưa báo nhận đk lưu trữ tạm thời trong bộ nhớ đệm; số khung đk truyền đi phụ thuộc vào bộ nhớ đệm; khi nhận đk báo nhận: giải phóng khung dữ liệu đã truyền thành công khỏi bộ nhớ đệm, truyền tiếp các khung bằng...
* Các khung dữ liệu đang đk gửi đi được đánh số: số thứ tự phải lớn hơn hoặc bằng kích thước của cửa sổ
* Các khung dữ liệu báo nhận bằng thông báo có đánh số
* Đk báo gộp. Nếu 1, 2, 3, 4 đk nhận thành công, chỉ gửi báo nhận4
* Khi đã nhận đk thông báo nhận đk khung k, có nghĩa là tất cả các khung k-1, k-2,.. đã nhận đk.
* Nguồn quản lý:
* Các khung đã gửi đi thành công
* Các khung đã gửi đi chwaa báo nhận
* Các khung đã gửi đi ngay
* Các khung chưa thể gửi đi ngay
* Đích quản lý:
* Các khung đã nhận đk
* Các khung đang chờ nhận