

Tổng quan về giao thức mạng

Giảng viên: Nguyễn Hoài Sơn, PhD

Email: sonnh@vnu.edu.vn, hoaisonuet@gmail.com

Bộ môn Mạng và Truyền thông máy tính

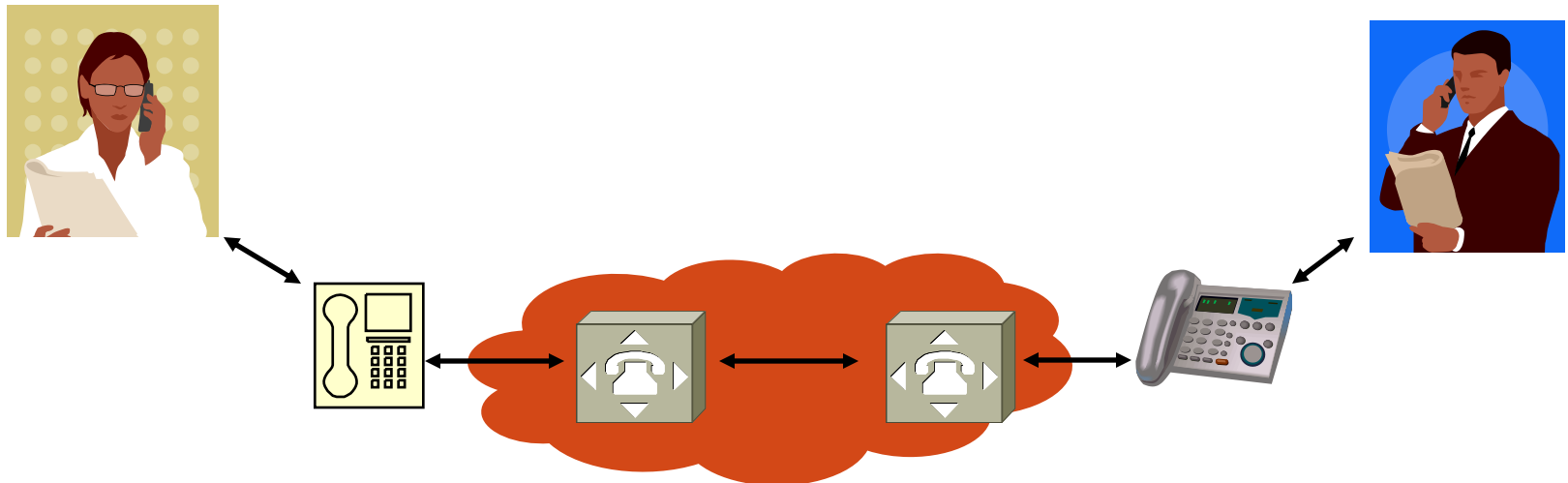
Khoa Công nghệ thông tin

Nội dung bài học

1. Khái niệm và các yêu cầu của giao thức mạng
2. Nguyên tắc thiết kế giao thức Internet
 - Nguyên tắc cuối cuối (End-to-end arguments)
 - Nguyên tắc phân tầng
 - Mô hình mạng OSI
 - Mô hình mạng TCP/IP
3. Giao thức tầng mạng
4. Giao thức tầng giao vận
5. Giao thức tầng ứng dụng
6. Mô hình client-server/ mô hình Peer-to-peer

Giao thức là gì?

- Giao thức là “quy ước giữa hai bên truyền tin về cách thức truyền tin”
 - Giao thức trong một cuộc gọi điện thoại: Người gọi – Điện thoại, Điện thoại – tổng đài, tổng đài – tổng đài, tổng đài – điện thoại, điện thoại - người nghe, người gọi – người nghe



Giao thức mạng là gì?

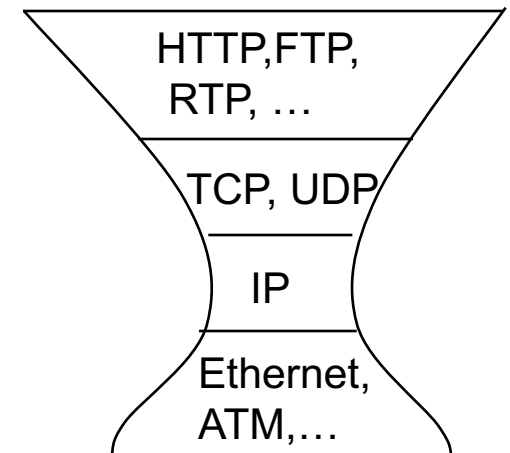
- Giao thức mạng là tập hợp các quy ước về định dạng và ý nghĩa của các thông báo được gửi giữa các máy tính thông qua mạng máy tính
- Có nhiều loại giao thức mạng
 - Giao thức nội mạng: Ethernet, AppleTalk, PPP, X.25, ...
 - Giao thức giữa các mạng: ATM, MPLS, TCP/IP, IPX, ...
 - Giao thức ứng dụng mạng: HTTP, FTP, SIP, ...

Yêu cầu với các giao thức mạng

- Phía người dùng mạng:
 - Chất lượng dịch vụ mà ứng dụng của họ cần
 - Đảm bảo các thông báo được gửi đến đúng địa chỉ không lỗi với độ trễ trong giới hạn cho phép
- Nhà cung cấp dịch vụ:
 - Sử dụng hiệu quả tài nguyên mạng
 - Đảm bảo tài nguyên mạng được sử dụng hiệu quả và công bằng với mọi người dùng
 - Hệ thống dễ điều hành và quản lý
 - Dễ dàng phát hiện và xử lý các lỗi hệ thống

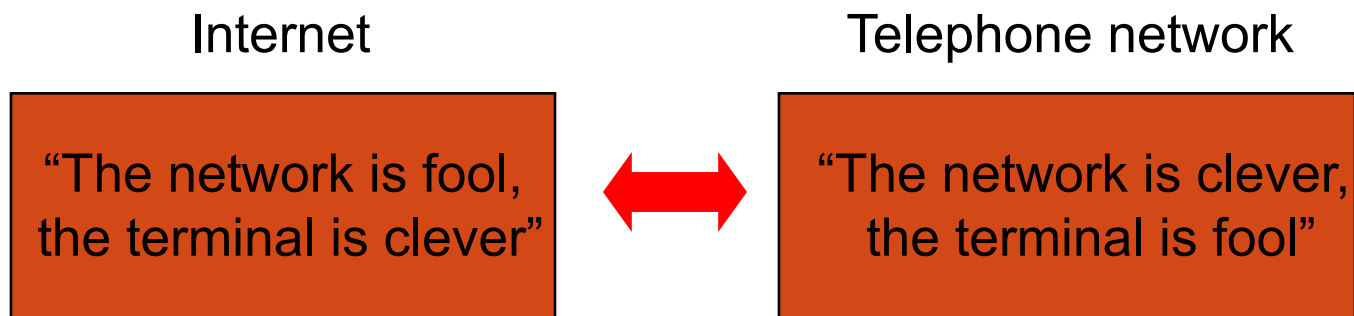
Internet

- Mạng công cộng kết nối các mạng máy tính
 - Quy mô toàn cầu
 - Mục đích chung, công cộng
 - Công nghệ đa dạng
- Giao thức trên Internet
 - Sử dụng giao thức TCP/IP
 - Chuẩn mở:
 - được chuẩn hoá bởi Internet Engineering Task Force (IETF)
<http://www.ietf.org>
 - Phát triển bởi cộng đồng nghiên cứu



Giao thức Internet được thiết kế và xây dựng như thế nào?

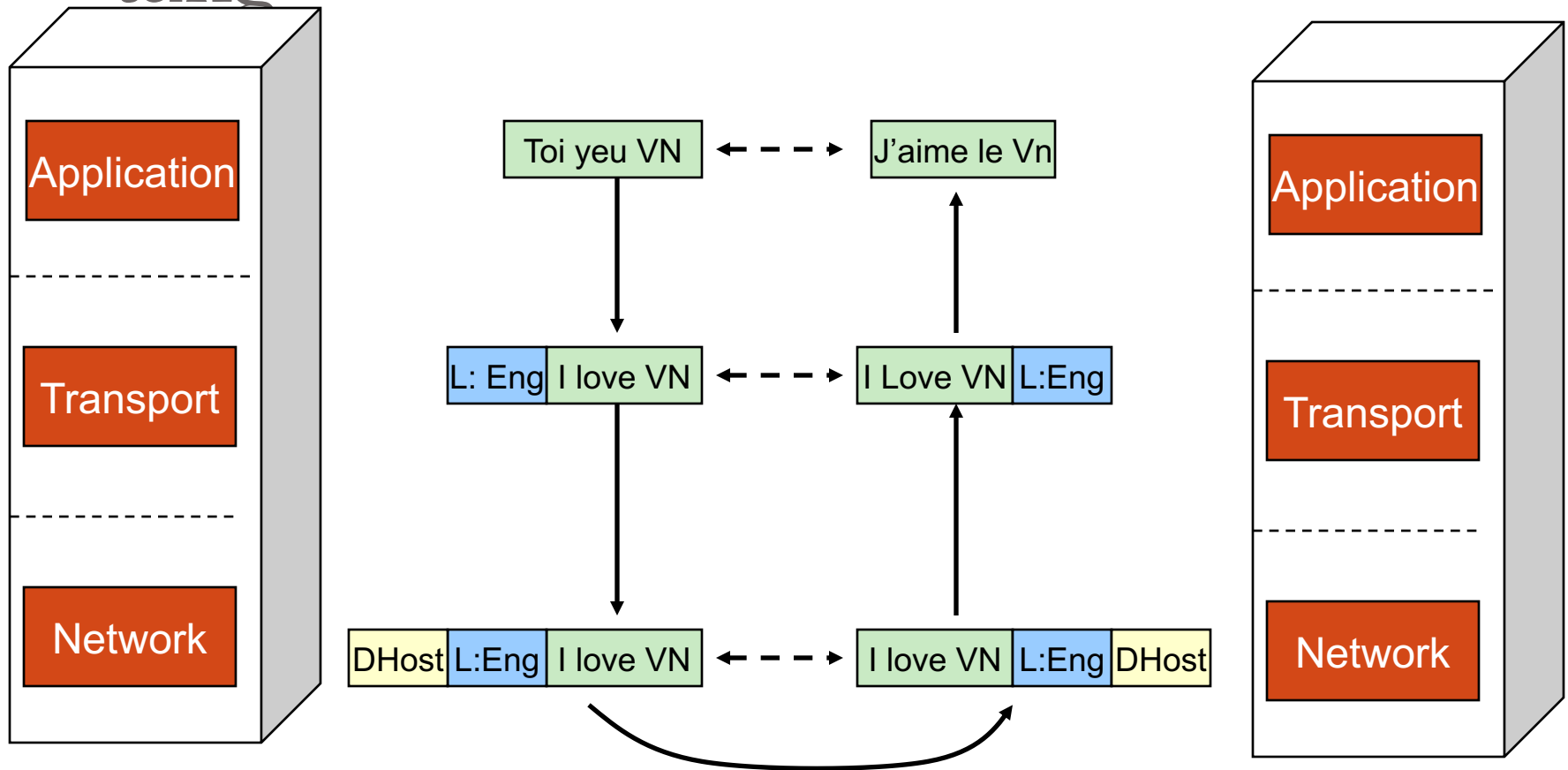
- **Nguyên tắc Điểm cuối-cuối**
 - *A function can only be completely and correctly implemented with the knowledge and help of the applications standing at the communication endpoints*



Mô hình mạng phân tầng

- Giao thức tầng cao sẽ tạo thông báo và gửi xuống giao thức tầng thấp thông qua giao diện giữa các tầng
- Giao thức tầng cao sẽ thêm thông tin điều khiển bao gồm các khóa giao thức vào tiêu đề của thông báo
- Thông báo của giao thức tầng trên sẽ được xử lý như dữ liệu của giao thức tầng dưới
- Giao thức tầng dưới sẽ sử dụng các khóa giao thức trong tiêu đề để xác định đúng giao thức tầng trên

Ví dụ về mô hình mạng phân tầng

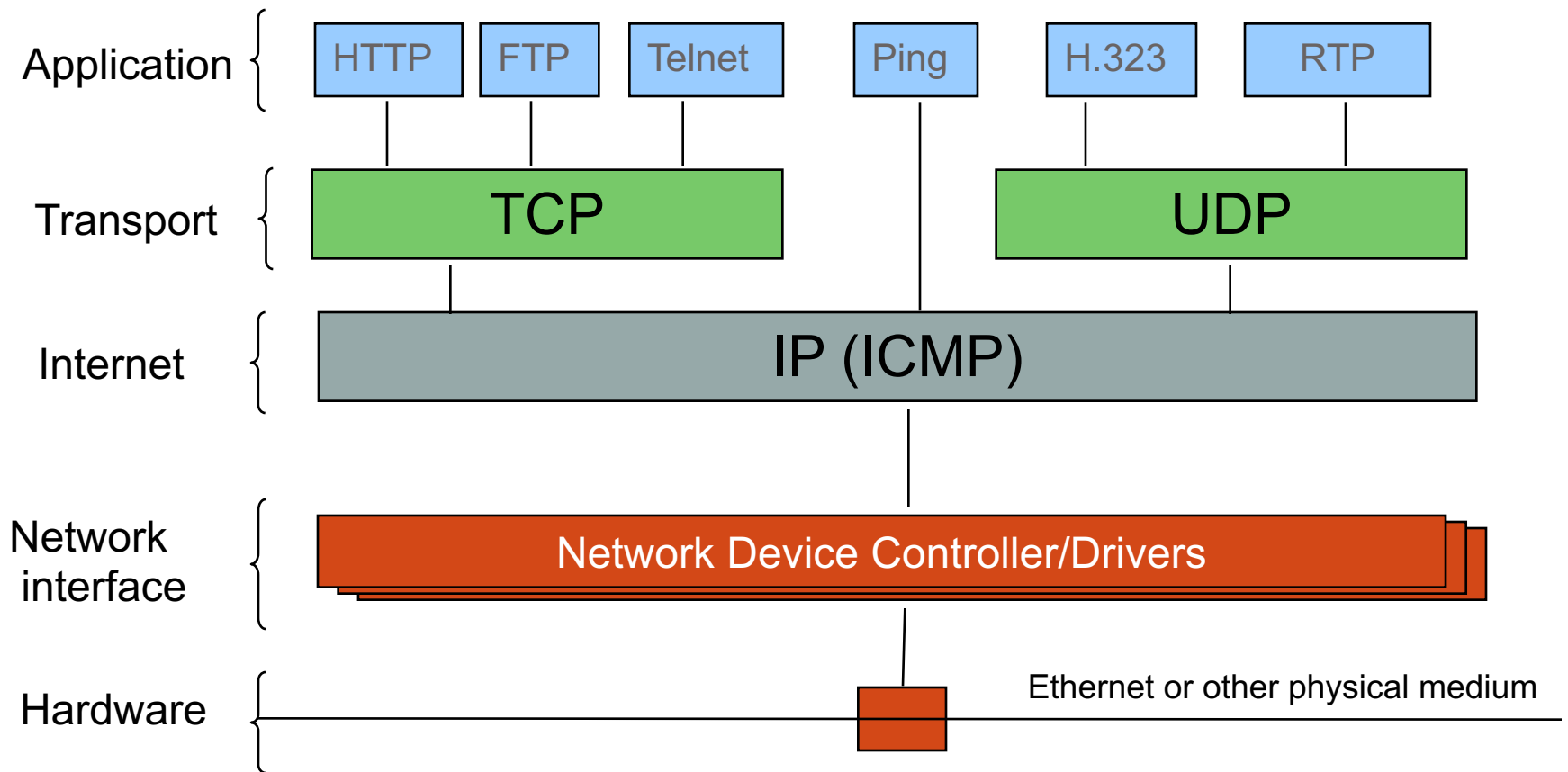


DHost = Destination service host

Giao thức Internet được thiết kế và xây dựng như thế nào?(2)

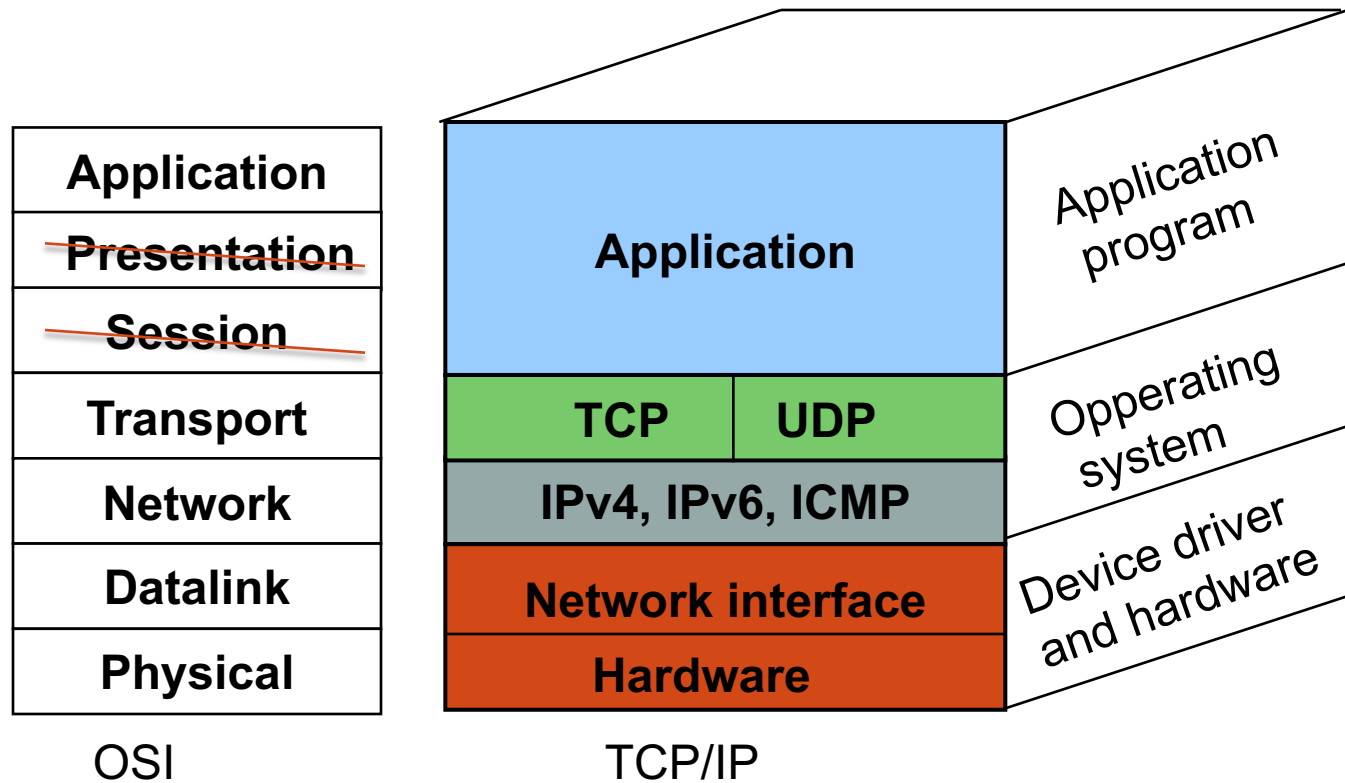
- Nguyên tắc phân tầng:
 - Chia giao thức mạng thành các tầng, mỗi tầng giao thức giải quyết một phần chức năng của truyền tin
- Ưu điểm của việc phân tầng
 - Cấu trúc rõ ràng – Cho phép định nghĩa rõ ràng các mối quan hệ giữa các giao thức mạng
 - Trừu tượng hóa chức năng – Có thể thay đổi một tầng giao thức mà không làm ảnh hưởng các tầng trên hoặc dưới
 - Sử dụng lại – Các tầng trên có thể sử dụng lại các chức năng được cung cấp bởi tầng dưới
- Nhược điểm của việc phân tầng
 - Che dấu thông tin – Giảm hiệu quả trong việc thực thi giao thức

Mô hình mạng TCP/IP



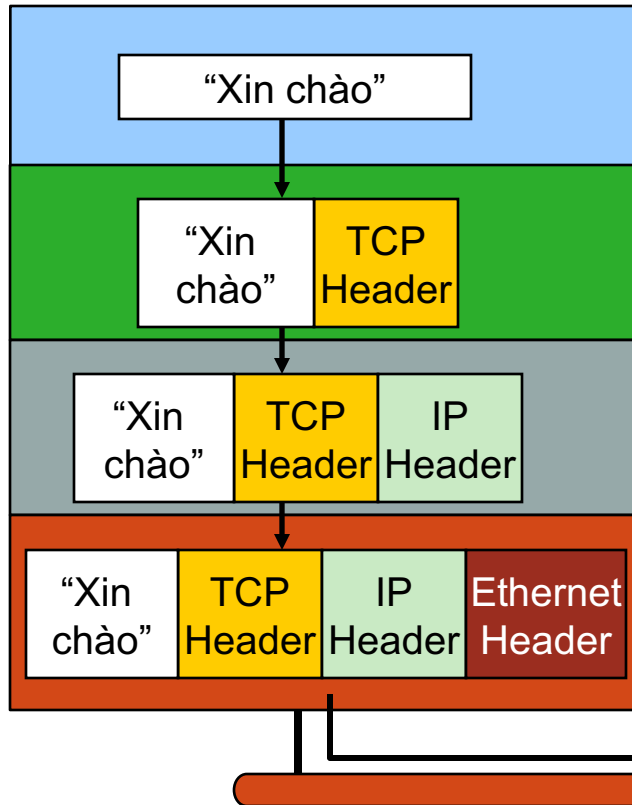
OSI vs. TCP/IP

- OSI: Định nghĩa khái niệm rõ ràng: dịch vụ, giao diện, giao thức
- TCP/IP: Được thực thi thành công

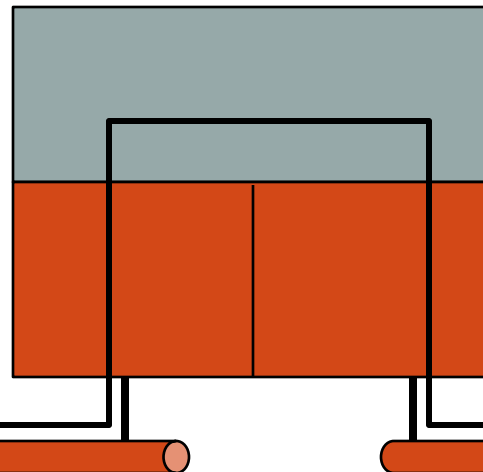


Ví dụ về giao thức TCP/IP

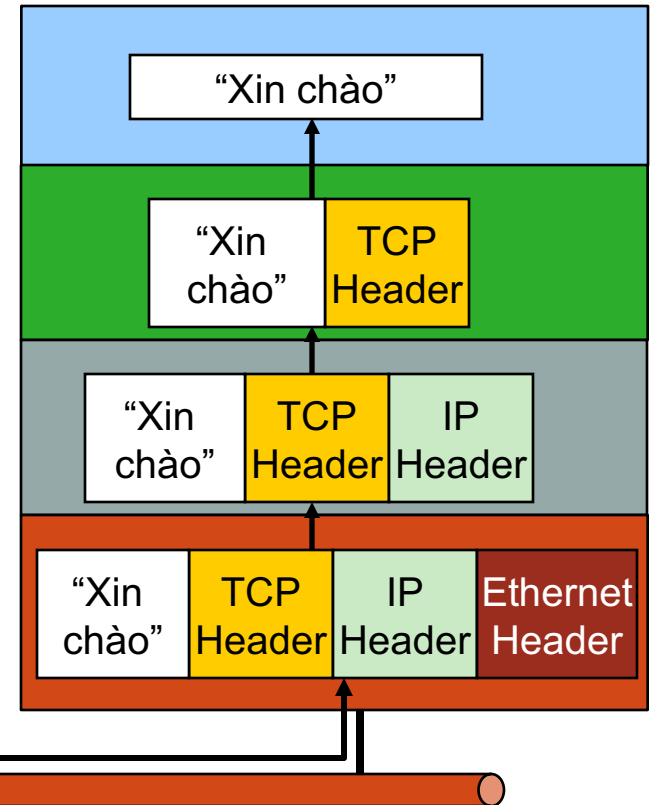
Sending host



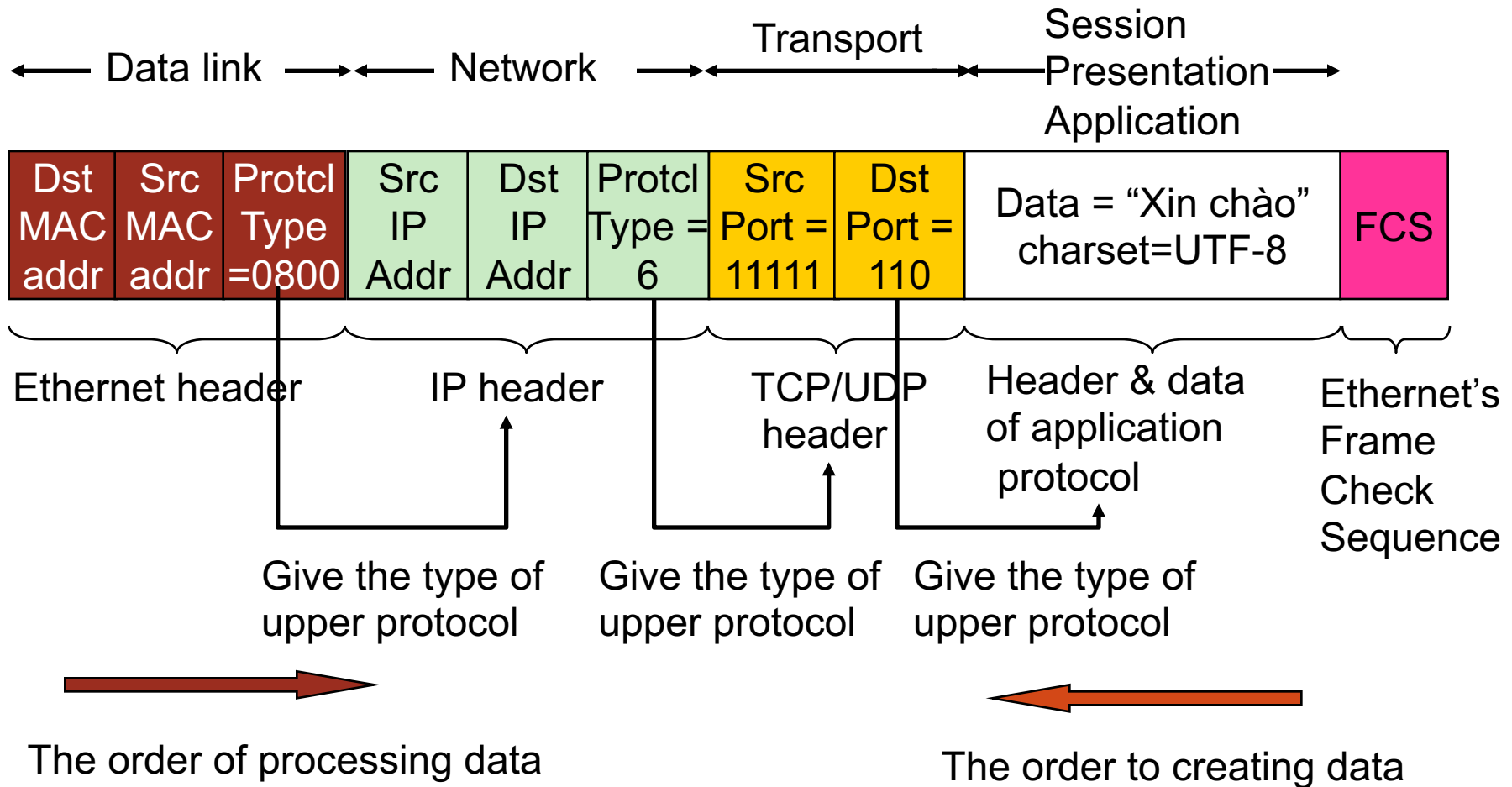
Router



Receiving host



Ví dụ về giao thức TCP/IP(2)



Giao thức tầng mạng

- Vận chuyển thông báo điểm cuối-cuối dựa trên địa chỉ IP
 - Không kết nối: Các gói tin được xử lý tách biệt
 - Không tin cậy: Việc vận chuyển gói tin không được đảm bảo
- Phân mảnh/ ghép mảnh
- Phát hiện lỗi

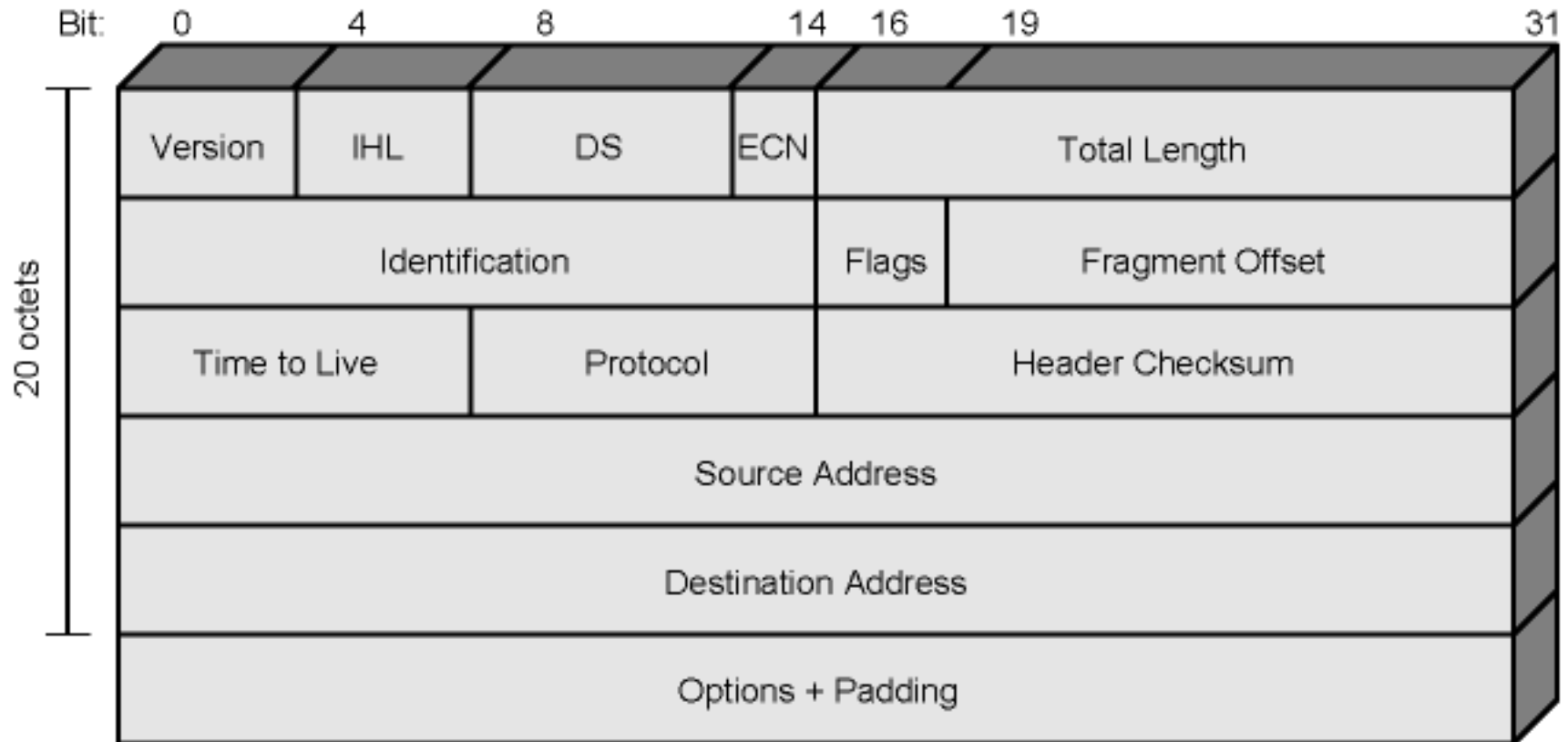
Địa chỉ IP

- Xác định một máy tính trên Internet
 - là duy nhất
 - độ dài 32 bit trong trường hợp của IPv4
 - Bao gồm ID mạng (network ID) và ID máy (host ID) được phân biệt bởi subnet mask (netmask)
 - E.g. 132.168.1.100/255.255.255.0
 - Trên thực tế địa chỉ IP gán cho một card mạng chứ không phải gán cho một máy
- Các lớp địa chỉ
 - A: 1.0.0.0 tới 127.255.255.255 (8 bit network address)
 - B: 128.0.0.0 tới 191.255.255.255 (16 bit network address)
 - C: 192.0.0.0 tới 223.255.255.255 (24 bit network address)
 - D,E: 224.0.0.0 and higher (multicast and reserved)
 - Địa chỉ quảng bá : 255.255.255.255
- Subnet mask
 - Chia một địa chỉ mạng thành nhiều phần mạng nhỏ hơn

Địa chỉ IP riêng

- Chỉ dùng cho các mạng riêng, không dùng cho mạng chung
 - Có thể truyền tin với các máy trên Internet thông qua Network Address Translator (NAT)
- Bao gồm 3 subnets
 - 10.0.0.0 ~ 10.255.255.255 (10/8)
 - 172.16.0.0 ~ 172.31.255.255 (172.16/12)
 - 192.168.0.0 ~ 192.168.255.255 (192.168/16)

Tiêu đề của gói tin IP



Khai báo tiêu đề của gói tin IP

```
struct ip{
#ifdef __BYTE_ORDER == __LITTLE_ENDIAN
    unsigned int ip_hl:4;          /* header length */
    unsigned int ip_v:4;          /* version */
#elseif
#ifdef __BYTE_ORDER == __BIG_ENDIAN
    unsigned int ip_v:4;          /* version */
    unsigned int ip_hl:4;          /* header length */
#elseif
    u_int8_t ip_tos;              /* type of service */
    u_short ip_len;               /* total length */
    u_short ip_id;                /* identification */
    u_short ip_off;              /* fragment offset field */
    u_int8_t ip_ttl;              /* time to live */
    u_int8_t ip_p;               /* protocol */
    u_short ip_sum;               /* checksum */
    struct in_addr ip_src, ip_dst; /* source and
                                    dest address */

};
```

```
#include <netinet/ip.h>
char packetBuffer[1500];
struct ip *ipPkt;
ipPkt = (struct ip *)packetBuffer;
```

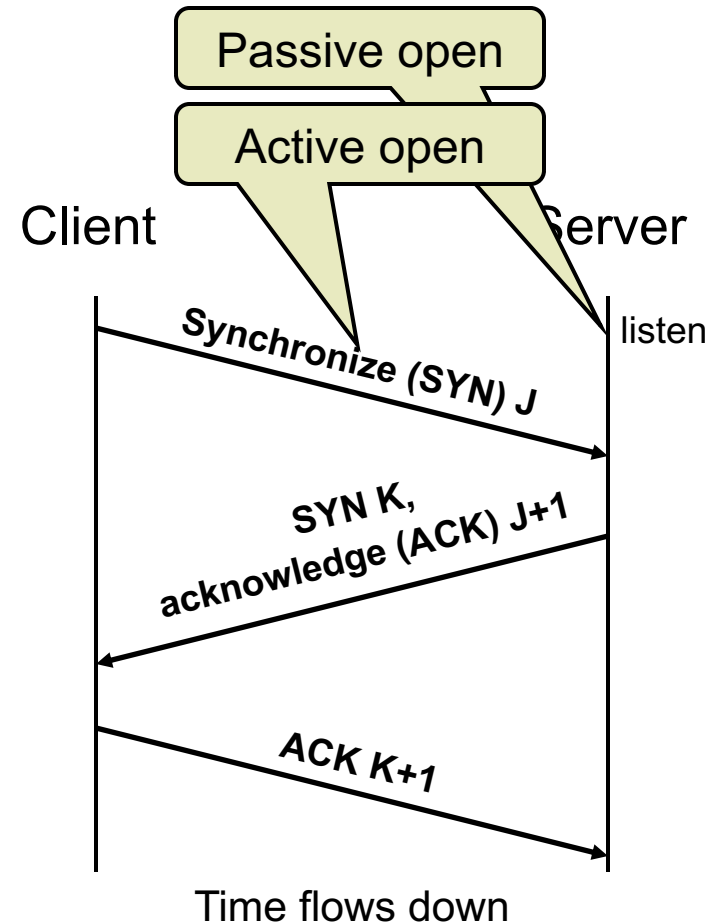
Giao thức tầng giao vận

- Giao thức TCP

- Hướng kết nối:
 - Thiết lập kết nối ảo bằng Bắt tay 3-bước trước khi dữ liệu được truyền đi
 - Một kết nối TCP được định danh bởi (srcIP, dstIP, src Port, dst Port)
- Tin cậy:
 - Bên nhận sẽ xác nhận việc nhận gói tin
 - Bên gửi sẽ gửi lại gói tin nếu không nhận được xác nhận của bên gửi
- Full-duplex:
 - Truyền và nhận gói tin cùng lúc
- Truyền theo dòng:
 - Đảm bảo việc truyền thông tin theo thứ tự của dòng bytes
- Điều khiển luồng:
 - Điều khiển chống tắc nghẽn

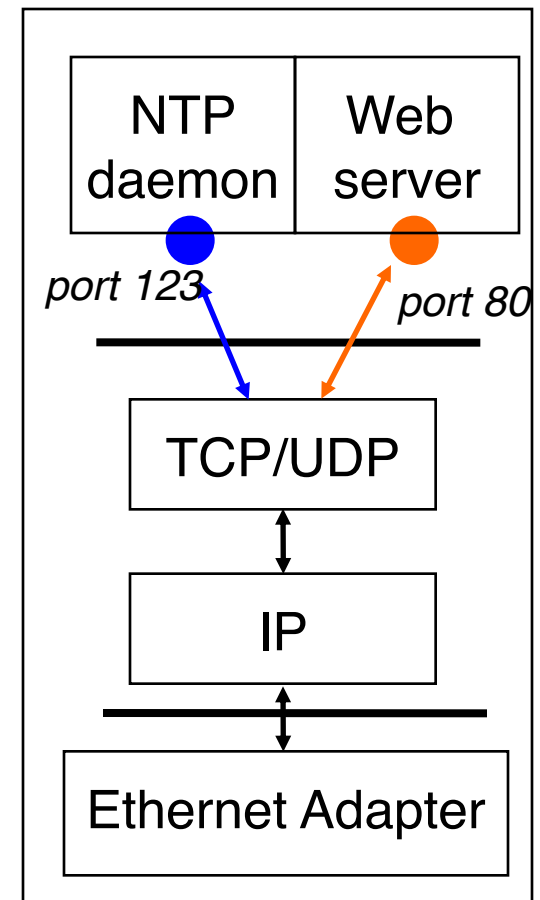
Thiết lập kết nối TCP

- **Bắt tay 3 bước**
 - Số thứ tự
 - J, K
 - Kiểu thông báo
 - Synchronize (SYN)
 - Acknowledge (ACK)
 - Mở thụ động
 - Máy chủ chờ kết nối đến từ máy khách
 - Mở chủ động
 - Máy khách bắt đầu một kết nối đến máy chủ



Khái niệm số hiệu cổng

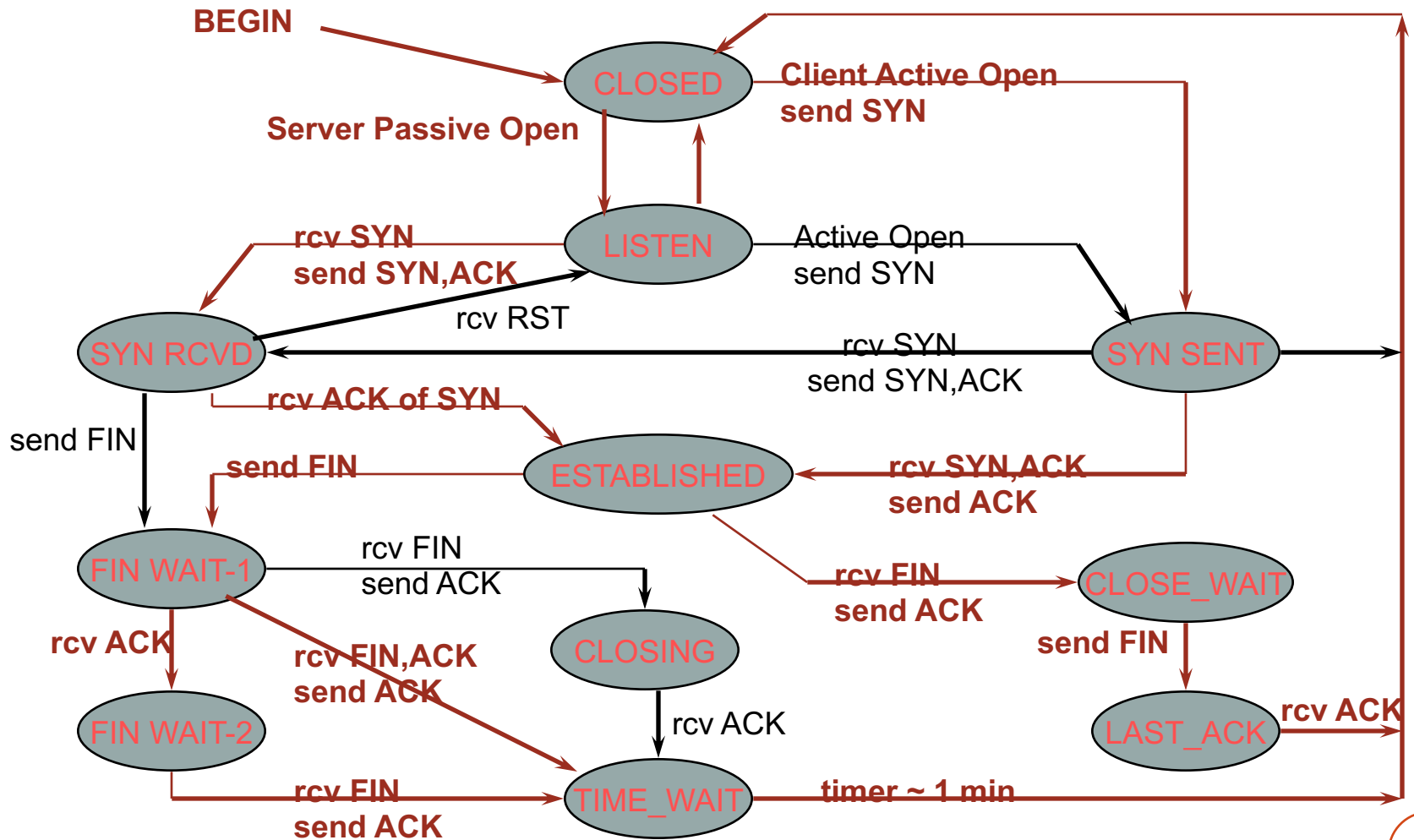
- Số hiệu cổng dùng để định danh tiến trình trên mỗi máy
- Số hiệu cổng có thể là
 - Well-known (port 0-1023)
 - Registered (port 1024-49151)
 - Dynamic or private (port 49152-65535)
- Máy chủ thường sử dụng well-known ports
 - Mọi máy khách có thể xác định được máy chủ/ dịch vụ cung cấp
 - HTTP = 80, FTP = 21, Telnet = 23, ...
 - */etc/service* định nghĩa well-known ports
- Máy khách thường sử dụng dynamic ports
 - được gán bởi hệ thống khi khởi tạo



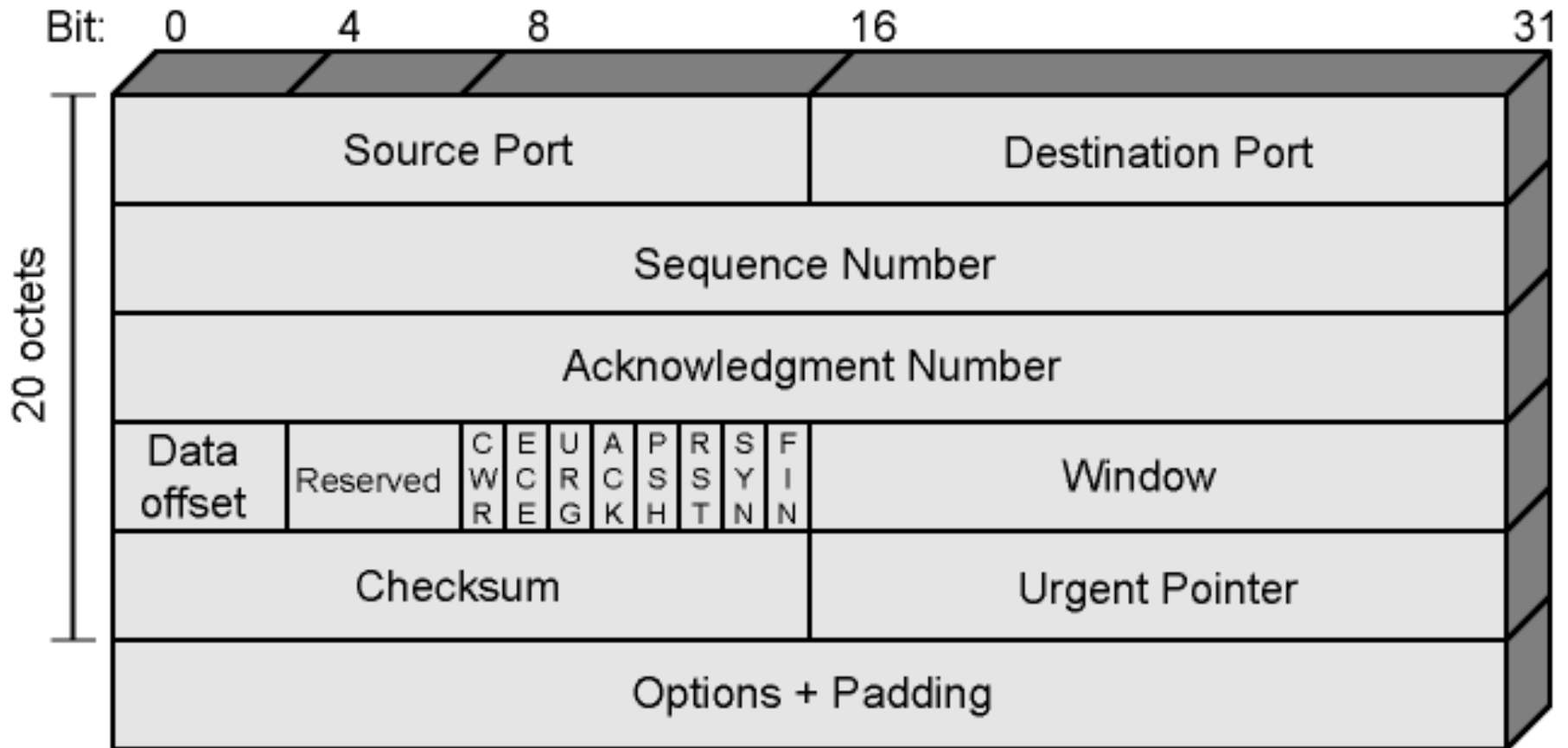
Well-known port numbers

Port number	Protocol	Keyword	Application
20	TCP	ftp-data	File transfer (Default data)
21	TCP	ftp	File transfer (Control)
22	TCP	ssh	SSH Remote Login Protocol
23	TCP	telnet	telnet
25	TCP	smtp	Simple Mail Transfer Protocol
53	TCP	domain	Domain Name Server
80	TCP	http	HTTP
110	TCP	pop3	Post Office Protocol – Ver 3
123	TCP	ntp	Network Time Protocol
443	TCP	https	HTTPS
7	UDP	echo	Echo
13	UDP	daytime	Daytime
53	UDP	domain	Domain Name Server
123	UDP	ntp	Network Time Protocol

Trạng thái kết nối TCP



Tiêu đề TCP



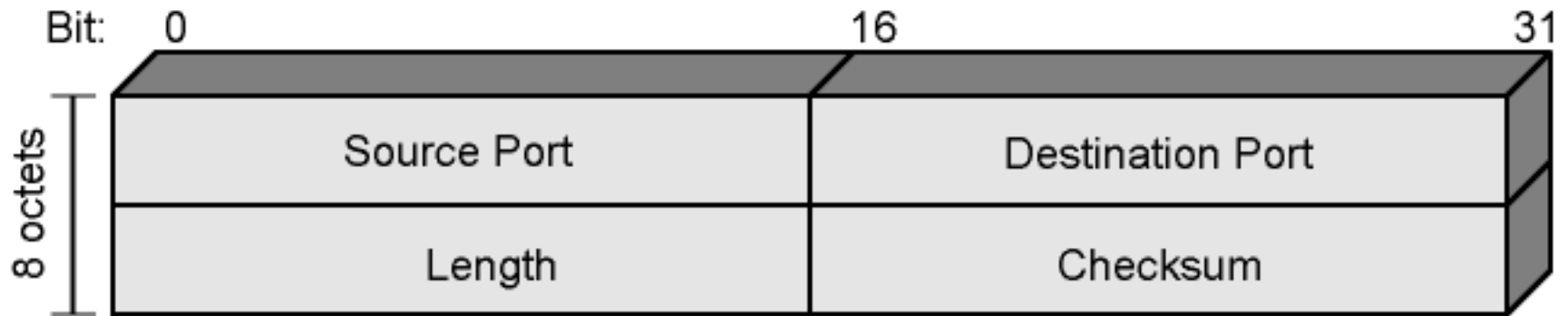
Khai báo tiêu đề TCP

```
struct tcphdr{
    u_int16_t th_sport;      /* source port */
    u_int16_t th_dport;      /* destination port */
    tcp_seq th_seq;          /* sequence number */
    tcp_seq th_ack;          /* acknowledgement number */
# if __BYTE_ORDER == __LITTLE_ENDIAN
    u_int8_t th_x2:4;        /* (unused) */
    u_int8_t th_off:4;       /* data offset */
# endif
# if __BYTE_ORDER == __BIG_ENDIAN
    u_int8_t th_off:4;       /* data offset */
    u_int8_t th_x2:4;        /* (unused) */
# endif
    u_int8_t th_flags;
    u_int16_t th_win;         /* window */
    u_int16_t th_sum;         /* checksum */
    u_int16_t th_urp;         /* urgent pointer */
};
```

Giao thức UDP

- Viết tắt của User Datagram Protocol
- Cung cấp dịch vụ truyền tin không kết nối cho các chương trình tầng ứng dụng
 - Truyền tin không tin cậy
 - Không đảm bảo việc truyền tin đến đúng địa chỉ và không bị trùng lặp
 - Truyền không giới hạn
 - Không có điều khiển luồng
- Yêu cầu xử lý nhỏ

Tiêu đề UDP

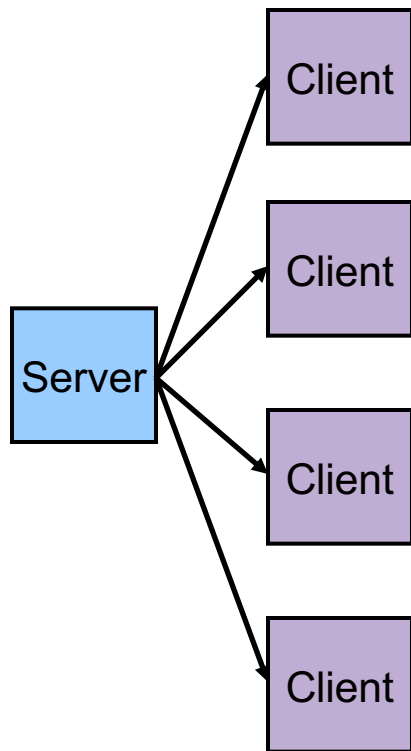


Khai báo tiêu đề UDP

```
struct udphdr
{
    u_int16_t uh_sport;        /* source port */
    u_int16_t uh_dport;        /* destination port */
    u_int16_t uh_ulen;          /* udp length */
    u_int16_t uh_sum;           /* udp checksum */
};
```

Giao thức tầng ứng dụng:

Mô hình truyền tin khách/chủ



- Truyền tin bất đối xứng
 - Máy khách gửi yêu cầu
 - Máy chủ gửi trả lời
- Máy chủ
 - Sử dụng định dạng đã biết (e.g., IP address + port)
 - Đợi kết nối đến
 - Xử lý yêu cầu, gửi trả lời
- Máy khách
 - Bắt đầu một kết nối
 - Đợi trả lời từ máy chủ

Giao thức tầng ứng dụng:

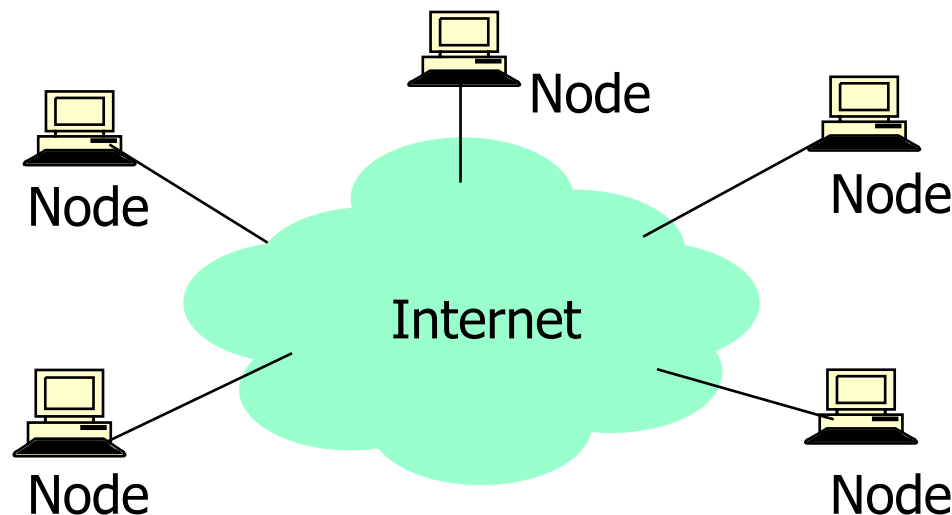
Mô hình truyền tin khách chủ(2)

- Mô hình dịch vụ
 - Xử lý tuần tự:
 - Máy chủ chỉ tạo một kết nối và xử lý yêu cầu của một máy khách tại mỗi thời điểm
 - E.g. Voice communication
 - Xử lý đồng thời:
 - Máy chủ tạo nhiều kết nối với các máy khách và xử lý các yêu cầu của các máy khách khác nhau cùng một lúc
 - E.g. Web server
 - Lai:
 - Máy chủ tạo ra nhiều kết nối với nhiều máy khách, nhưng xử lý yêu cầu của các máy khách một cách tuần tự
- Không có ranh giới rõ ràng giữa hai khái niệm máy khách và máy chủ
 - Một máy chủ có thể là máy khách của một máy chủ khác
 - Một máy chủ có thể là máy khách của chính máy khách của nó

Giao thức tầng ứng dụng:

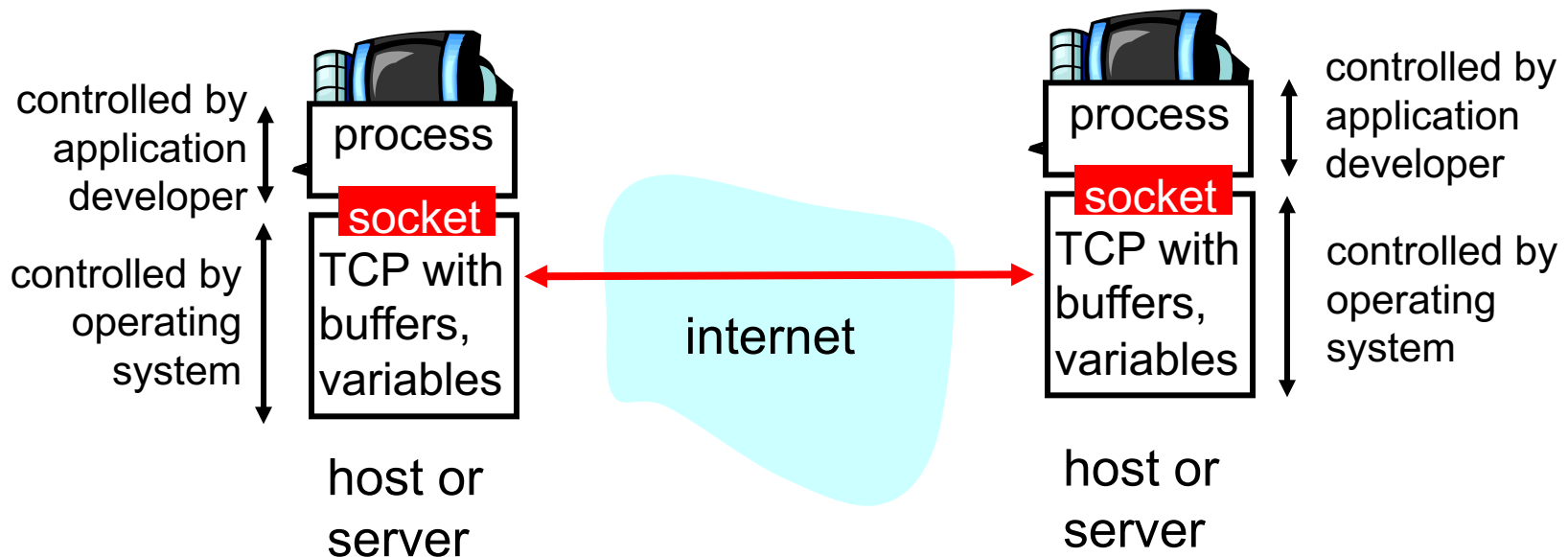
Mô hình truyền tin ngang hàng

- Các node trong mạng đóng cả hai vai trò máy chủ/máy khách
 - Cung cấp và sử dụng tài nguyên
 - Bất cứ node nào cũng có thể khởi tạo kết nối
- Không có máy chủ dữ liệu trung tâm
 - “The ultimate form of democracy on the Internet”
 - “The ultimate threat to copy-right protection on the Internet”



Giao thức tầng ứng dụng

- Sử dụng các dịch vụ truyền tin cung cấp bởi tầng dưới
 - Thông qua giao diện socket (= “bit pipe”)



Các yêu cầu về dịch vụ giao vận của các ứng dụng phổ biến

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	loss-tolerant	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kb-1Mb video: 10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps up	yes, 100's msec
financial apps	no loss	elastic	yes and no

Một số giao thức tầng ứng dụng

Application	Application layer protocol	Underlying transport protocol
e-mail	smtp [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
file transfer	ftp [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
remote file server	NSF	TCP or UDP
Internet telephony	proprietary (e.g., Vocaltec)	typically UDP