



MẠNG MÁY TÍNH

Chương 5

LỚP TRANSPORT (LỚP GIAO VẬN)

Nội dung chương 5

- I. Các vấn đề thiết kế lớp transport
- II. Lớp transport trên mạng TCP/IP
- III. Giới thiệu giao diện lập trình mạng socket

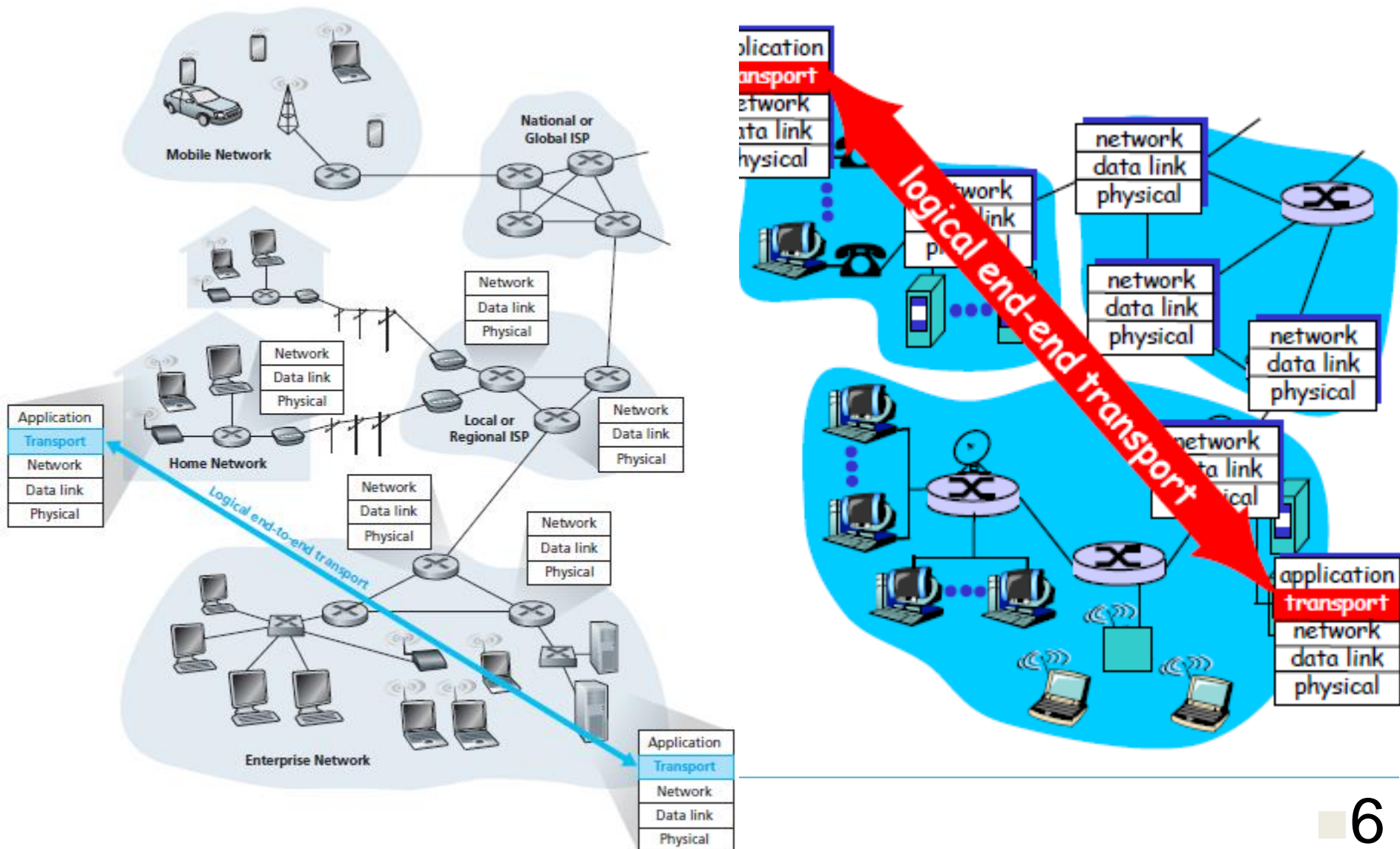
I. Các vấn đề thiết kế lớp transport

- Nhiệm vụ lớp transport
- Dịch vụ lớp transport

1. Nhiệm vụ lớp transport

- Cung cấp dịch vụ gửi nhận dữ liệu tin cậy giữa các chương trình trên hai máy bất kỳ
- Thực hiện:
 - Chia và ghép dữ liệu từ lớp application
 - Kiểm soát lỗi, kiểm soát lưu lượng
- Lớp transport có vai trò quan trọng trên kiến trúc mạng nhiều lớp

Nhiệm vụ lớp transport (tt)



2. Dịch vụ lớp transport

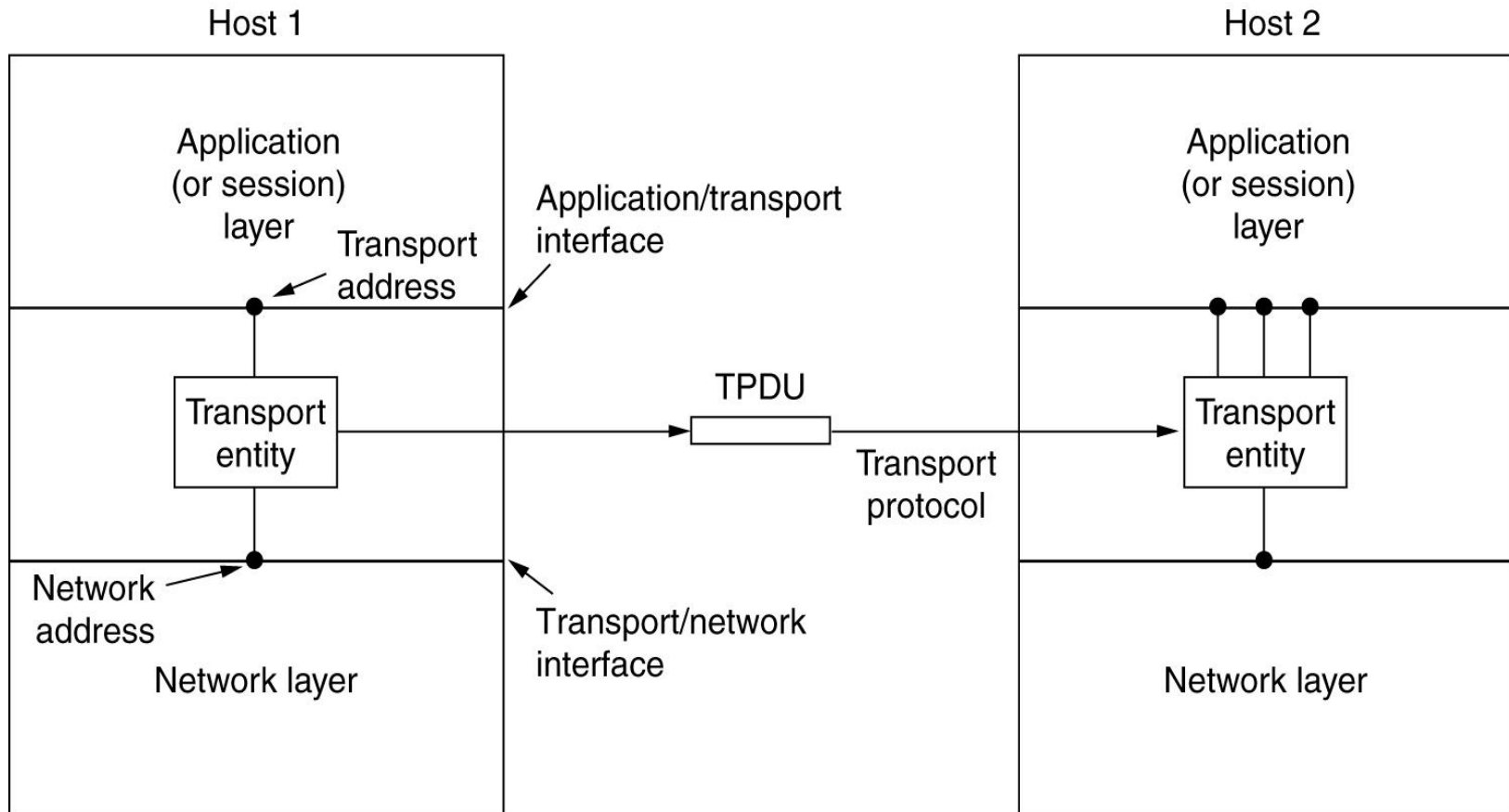
- Dịch vụ lớp transport
- Các thao tác cơ sở

a. Dịch vụ lớp transport

- Cung cấp *truyền thông logic* chạy trên các host khác nhau (truyền thông logic giữa các tiến trình)
- Các giao thức transport chạy trên các hệ thống đầu cuối
 - Phía gửi: cắt các thông điệp ứng dụng thành các segment (đoạn), chuyển cho lớp network
 - Phía nhận: ghép các đoạn thành các thông điệp, chuyển cho lớp application

a. Dịch vụ lớp transport

- Quan hệ giữa các lớp

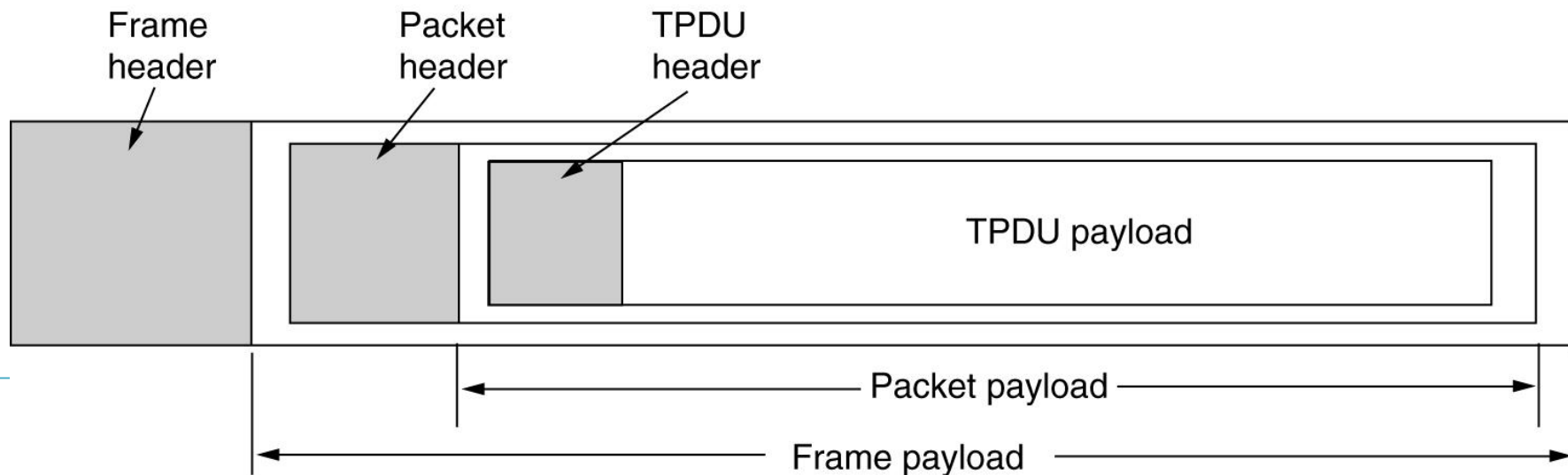


Dịch vụ lớp transport (tt)

- Các thuật ngữ:
 - Transport entity: thực thể lớp transport
 - TPDU (Transport Protocol Data Unit): đơn vị dữ liệu giao thức lớp transport
 - Transport address: địa chỉ lớp transport
 - Transport Service Access Point
 - Port (mạng TCP/IP)
 - Network address: địa chỉ lớp network
 - Địa chỉ IP (mạng TCP/IP)

Dịch vụ lớp transport (tt)

- Các dạng dịch vụ:
 - Có kết nối (connection-oriented service)
 - Không kết nối (connectionless service)
- Đơn vị dữ liệu giao thức lớp transport
 - TPDU trong packet và frame

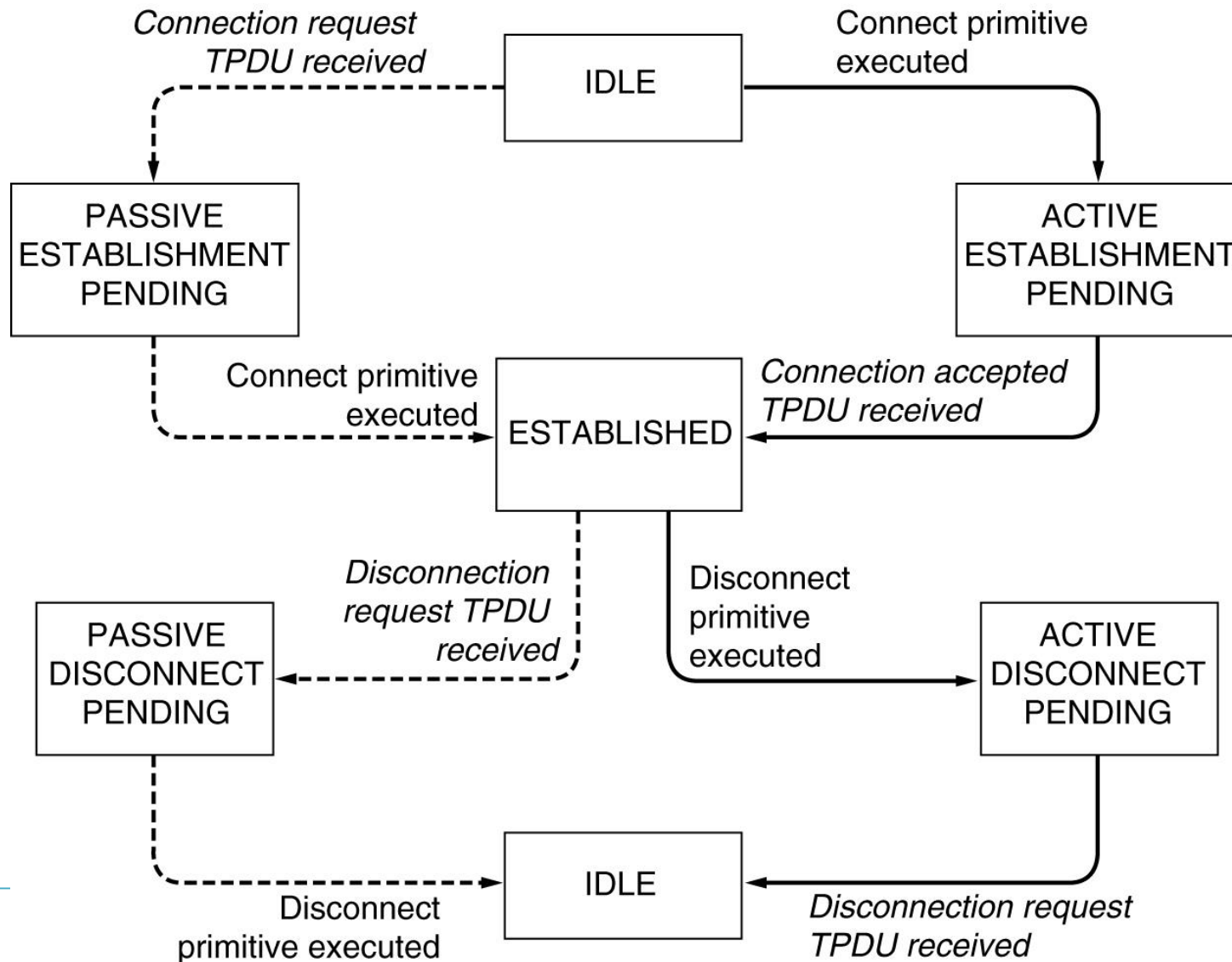


b. Các thao tác cơ sở (Transport service primitives)

- Các thao tác cơ sở của dịch vụ đơn giản
- Ví dụ: mô hình client-server dạng có kết nối

Primitive	Dữ liệu gửi	Ý nghĩa
LISTEN	Không có	Chờ process khác kết nối
CONNECT	CONNECTION REQUEST	Thiết lập kết nối
SEND	DATA	Gửi dữ liệu
RECEIVE	Không có	Chờ nhận dữ liệu
DISCONNECT	DISCONNECTION REQUEST	Yêu cầu hủy kết nối

Mô hình Client-Server dạng có kết nối



II. Lớp transport trên mạng TCP/IP

- Giao thức TCP (Transmission Control Protocol)
 - Vận chuyển hướng kết nối (connection-oriented)
 - Điều khiển tắc nghẽn
 - Điều khiển luồng
 - Thiết lập kết nối
- Giao thức UDP (User Datagram Protocol)
 - Vận chuyển không kết nối (connectionless)
 - Không tin cậy, truyền không theo thứ tự

1. Giao thức TCP

- Giới thiệu TCP
- Mô hình dịch vụ TCP
- Giao thức TCP
- TCP segment header
- Thiết lập kết nối TCP

a. Giới thiệu TCP

- Cung cấp dịch vụ gửi nhận chuỗi byte tin cậy giữa hai chương trình trên mạng có thể không tin cậy
- Thực thể TCP:
 - Thư viện
 - User process
 - Kernel
- Chia dữ liệu từ process ứng dụng, gửi trên các gói IP

b. Mô hình dịch vụ TCP

- Dịch vụ TCP thực hiện trên kết nối TCP (TCP connection)
- Kết nối TCP bao gồm hai đầu cuối (end-point), được gọi là socket
- Socket number (socket address):
 - Địa chỉ IP – 32 bit
 - Port – 16 bit

Port

- Khái niệm trừu tượng → nhiều ứng dụng TCP trên một máy
- Well-known ports: dùng cho các dịch vụ chuẩn, ví dụ:
 - Port 21: FTP - File Transfer Protocol
 - 25: SMTP - Email
 - 80: HTTP - Web

TCP port

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP Ports:

1863 MSN Messenger
8008 Alternate HTTP
8080 Alternate HTTP

Well Known TCP Ports

21 FTP
23 Telnet
25 SMTP
80 HTTP
110 POP3
194 Internet Relay Chat (IRC)
443 Secure HTTP (HTTPS)

Các tính chất của kết nối TCP

- Full-duplex
- Point-to-point
- Byte stream

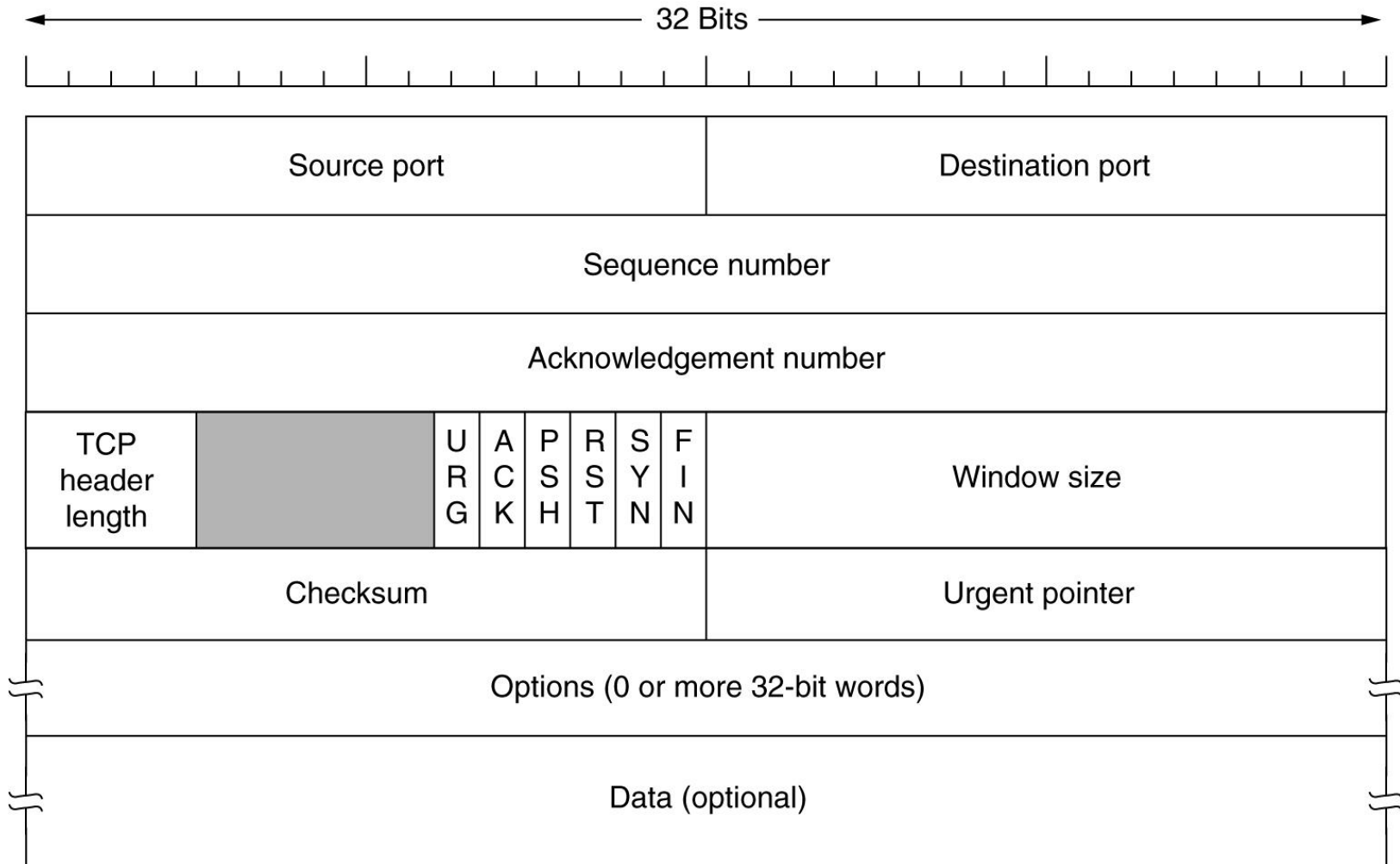
c. Giao thức TCP

- Đơn vị dữ liệu: TCP segment
 - TCP header ≥ 20 bytes
 - TCP data ≥ 0 bytes
- Kích thước TCP segment bị giới hạn bởi:
 - IP payload (65515 bytes)
 - MTU (Maximum Transfer Unit)
 - Ví dụ: MTU mạng Ethernet ~ 1500 bytes

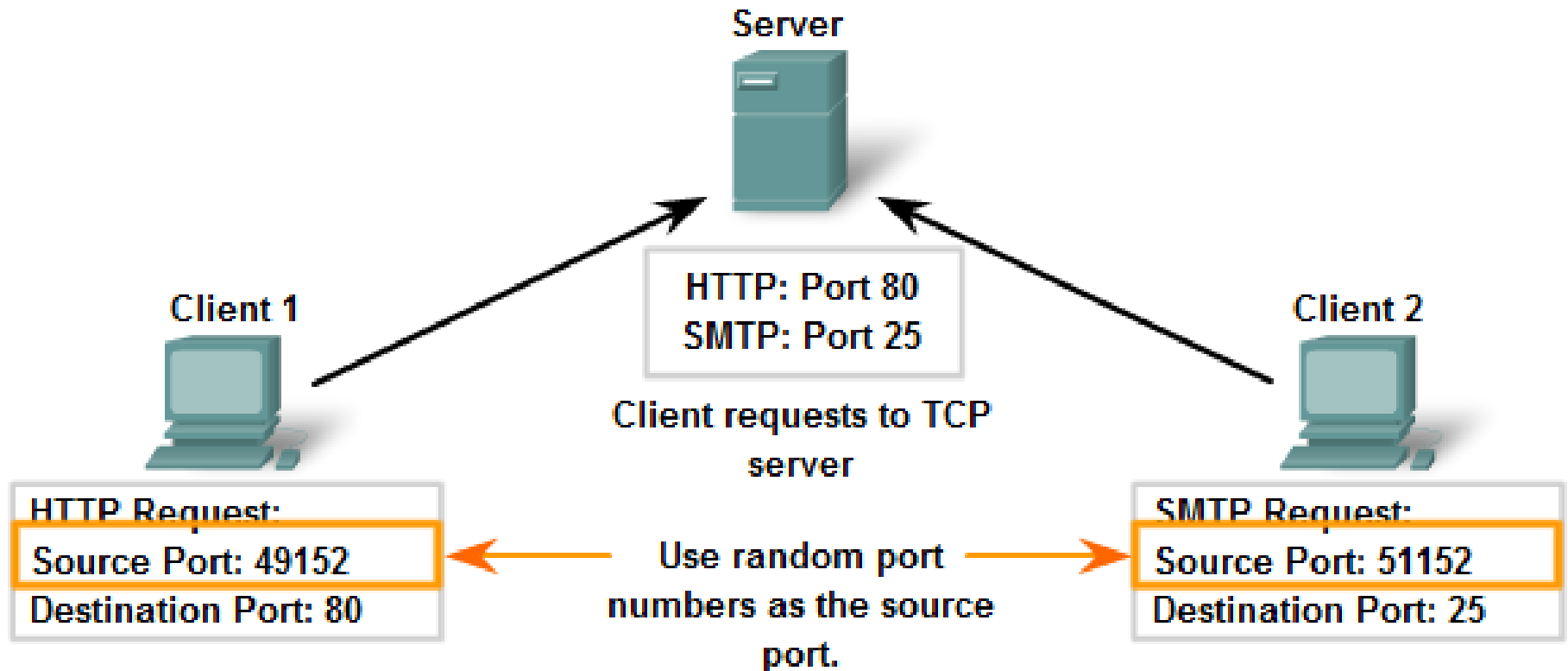
Giao thức TCP (tt)

- Mỗi byte truyền trên kết nối TCP có số thứ tự trình tự (sequence number) 32 bit
- Giao thức cơ bản: sliding window
 - Sender gửi segment, khởi động timer
 - Receiver gửi segment có kèm ACK number là số thứ tự byte chờ nhận tiếp theo
 - Sender sẽ gửi lại nếu không có ACK khi hết thời gian

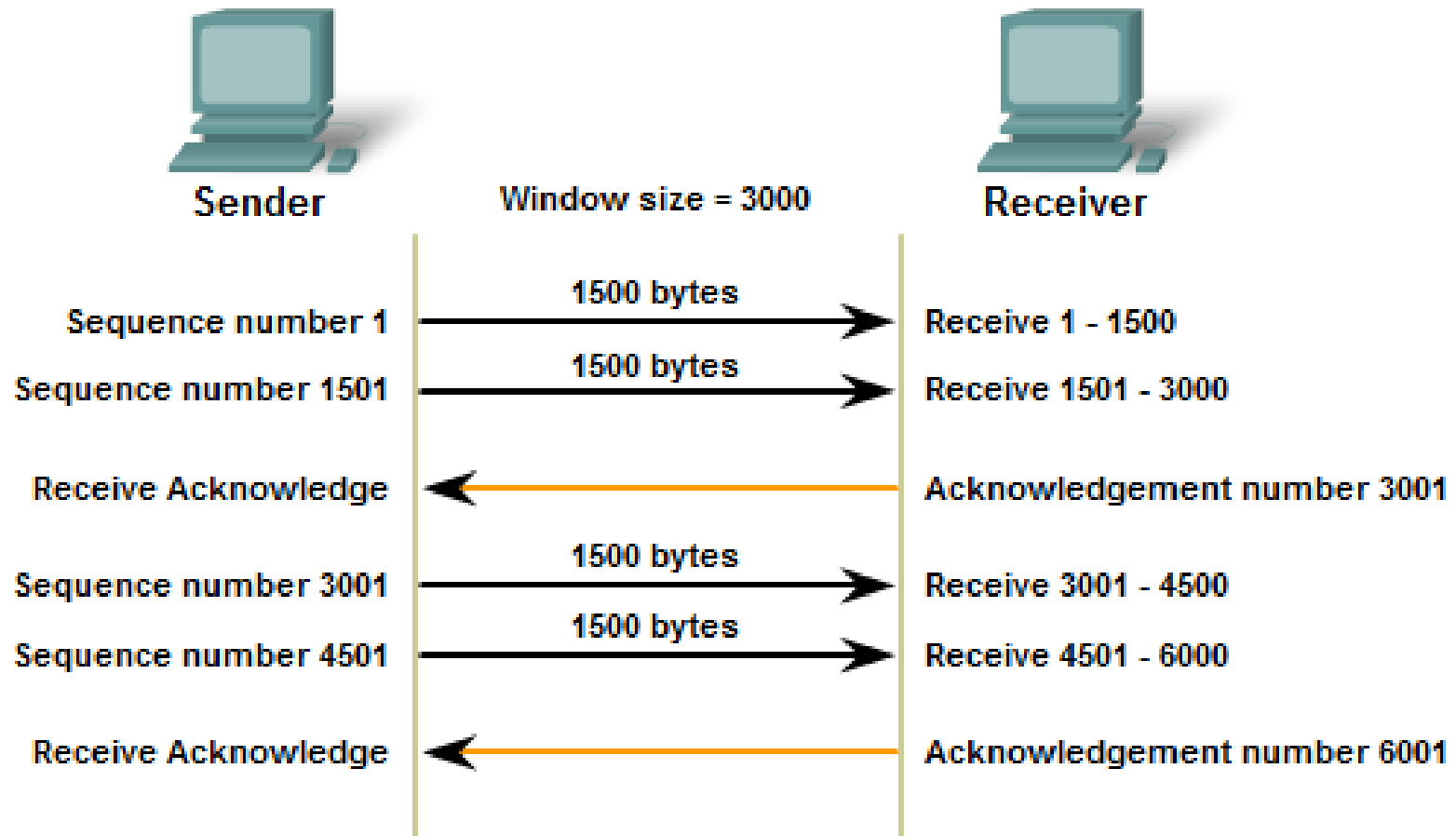
d. TCP segment header



Ví dụ: port trên TCP segment



Ví dụ: windows size, ack trên TCP segment

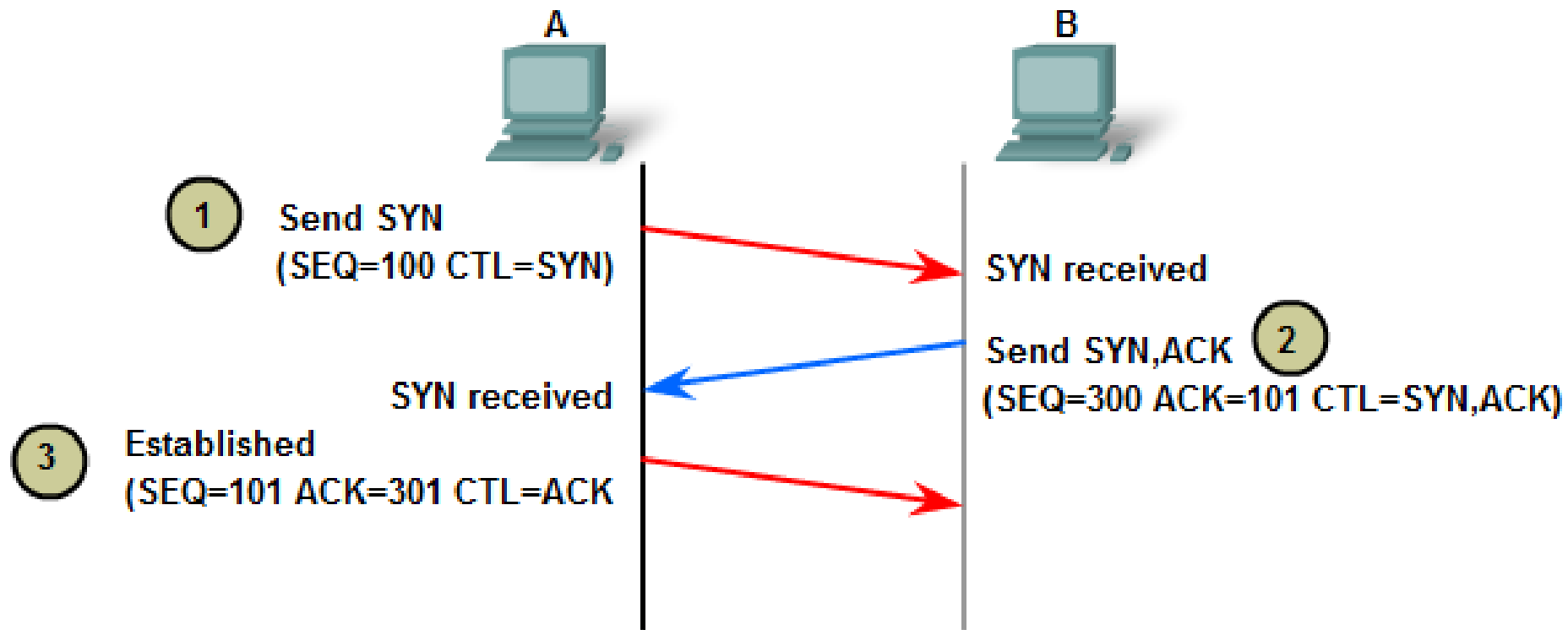


e. Thiết lập kết nối TCP

- Thiết lập kết nối: Three-way handshake
- VD: Host 1 (Client) kết nối đến Host 2 (Server)
 - Host 1 \rightarrow Host 2:
seq=x, ack=0, SYN=1, ACK=0
 - Host 2 \rightarrow Host 1:
seq=y, ack=x+1, SYN=1, ACK=1
 - Host 1 \rightarrow Host 2:
seq=x+1, ack=y+1, SYN=0, ACK=1

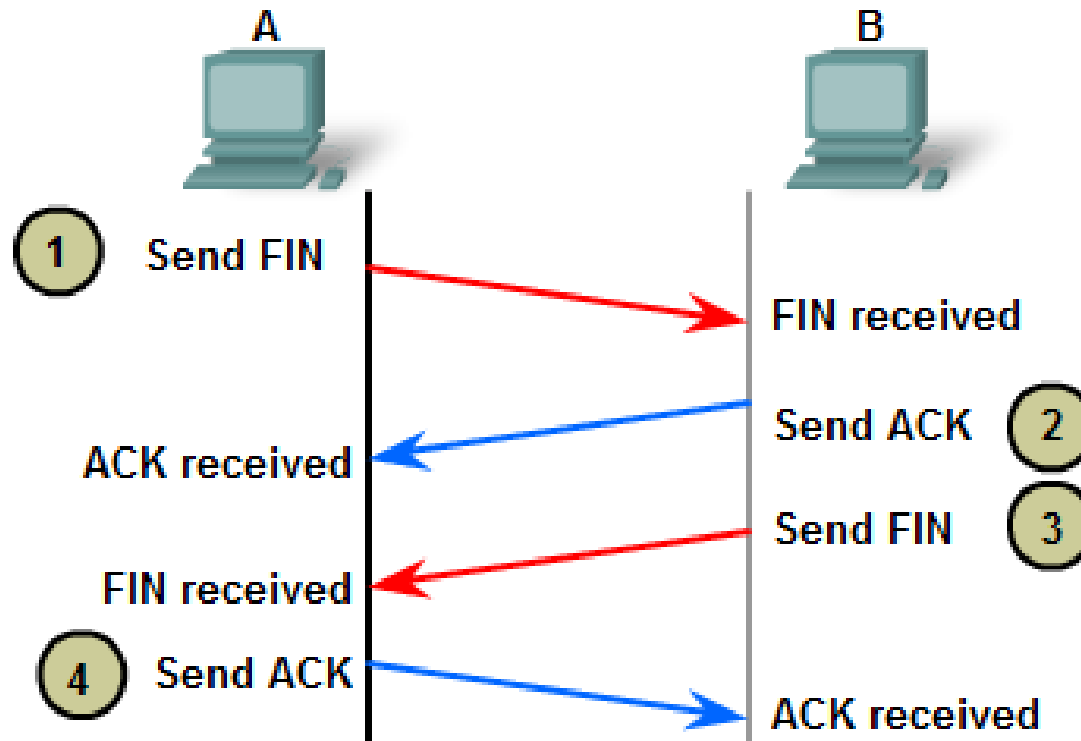
Thiết lập kết nối TCP (tt)

- Sơ đồ thiết lập kết nối TCP



Hủy bỏ kết nối TCP

- Gởi TCP segment với FIN=1
- Cần một FIN segment và một ACK segment cho một bên truyền thông



2. Giao thức UDP

- Giao thức dạng không kết nối
- Không có kiểm soát lỗi → nếu cần thì thực hiện trên lớp application
- Đơn vị dữ liệu: UDP datagram/segment
 - UDP header: 8 bytes
 - UDP data
- Sử dụng khái niệm port tương tự TCP

Giao thức UDP (tt)

- UDP port

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered UDP Ports:

1812 RADIUS Authentication Protocol
2000 Cisco SCCP (VoIP)
5004 RTP (Voice and Video Transport Protocol)
5060 SIP (VoIP)

Well Known UDP Ports:

69 TFTP
520 RIP

Giao thức UDP (tt)

- UDP header
 - Source port: địa chỉ port chương trình gửi
 - Destination port: địa chỉ port chương trình nhận
 - UDP length: kích thước header+data
 - UDP checksum: phát hiện lỗi cho header+data

32 Bits

Source port

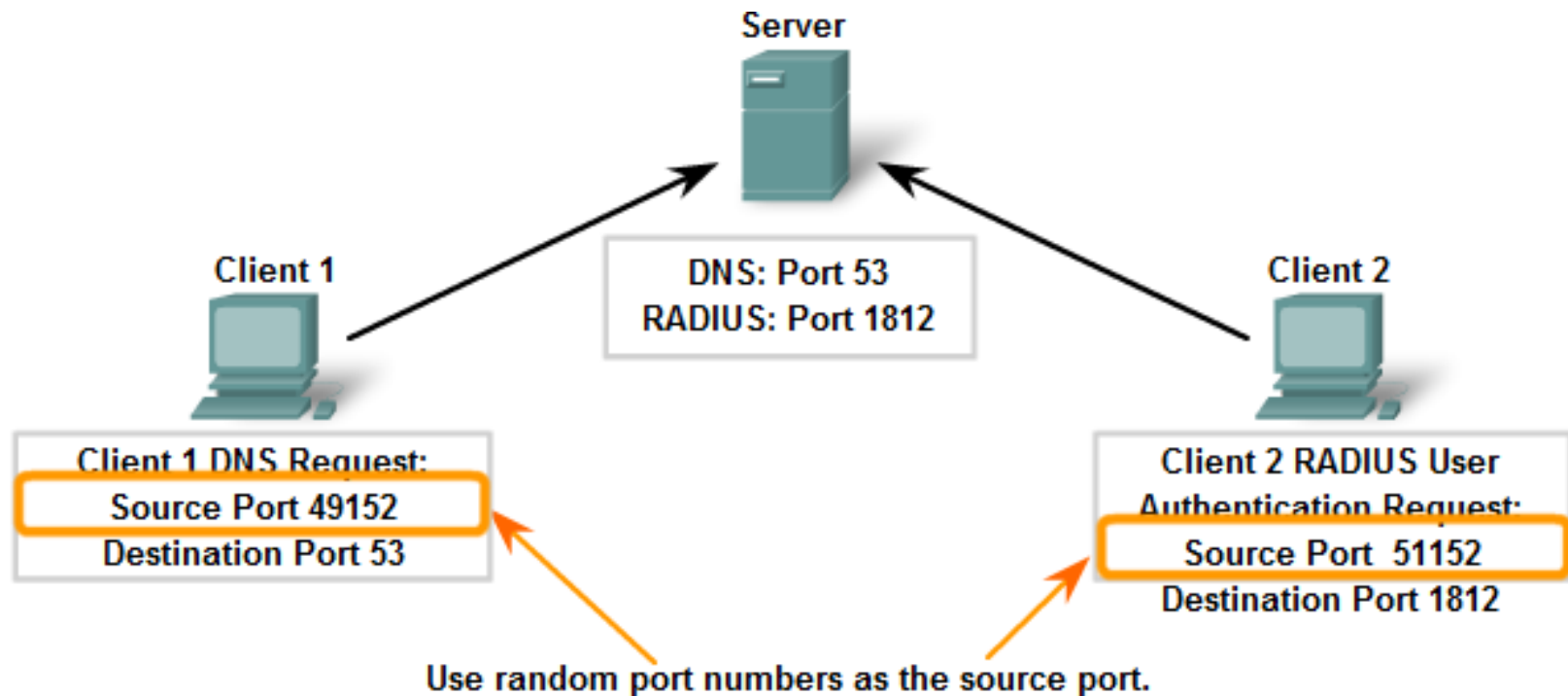
Destination port

UDP length

UDP checksum

Giao thức UDP (tt)

- Ví dụ: port trên UDP header



III. Giới thiệu giao diện lập trình mạng socket

- Khái niệm Socket API
- Giới thiệu Windows Sockets (WinSock)

1. Khái niệm Socket API

- API (Application Programming Interface)
 - Giao diện lập trình ứng dụng: tập hợp các hàm cung cấp cho chương trình ứng dụng
- Socket APIs trừu tượng hoá việc truyền thông dạng client/server trên bộ giao thức TCP/IP với mô hình socket
- Socket API có thể sử dụng cho các bộ giao thức khác như IPX/SPX, DECNet, ...)

Hai dạng Socket APIs

- Berkeley Sockets (BSD Sockets)
 - Cung cấp các thao tác cơ sở (primitives) dùng trên UNIX
- Windows Sockets (WinSock)
 - Có các mở rộng hỗ trợ cơ chế message-driven của Windows

Ví dụ

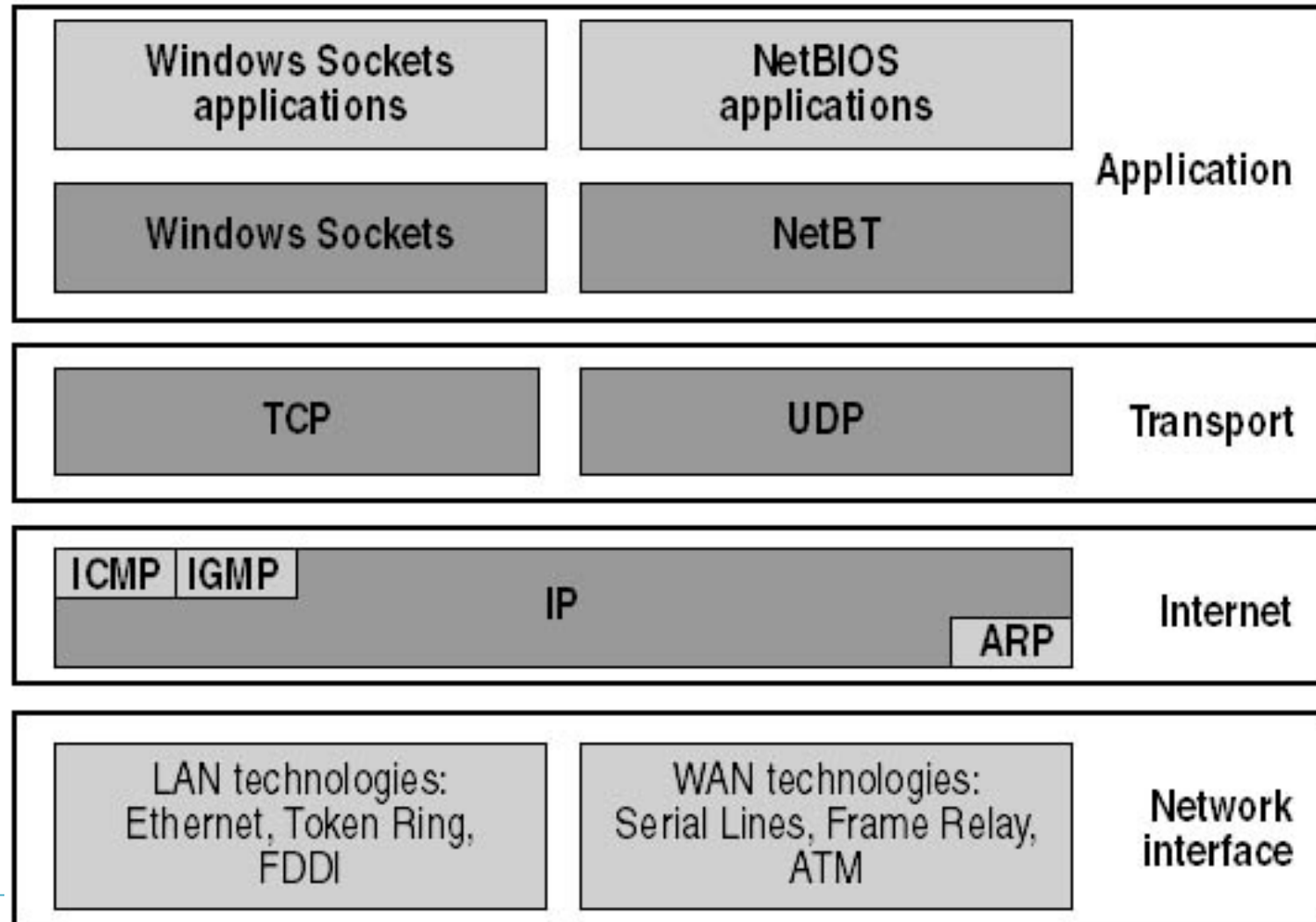
- Các thao tác cơ sở trên TCP của BSD sockets

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

2. Giới thiệu WinSock

- WinSock: giao diện lập trình mạng dùng trên hệ điều hành Windows trên mô hình socket
- Chương trình sử dụng WinSock API, liên kết với thư viện WinSock

Kiến trúc TCP/IP trên Microsoft Windows



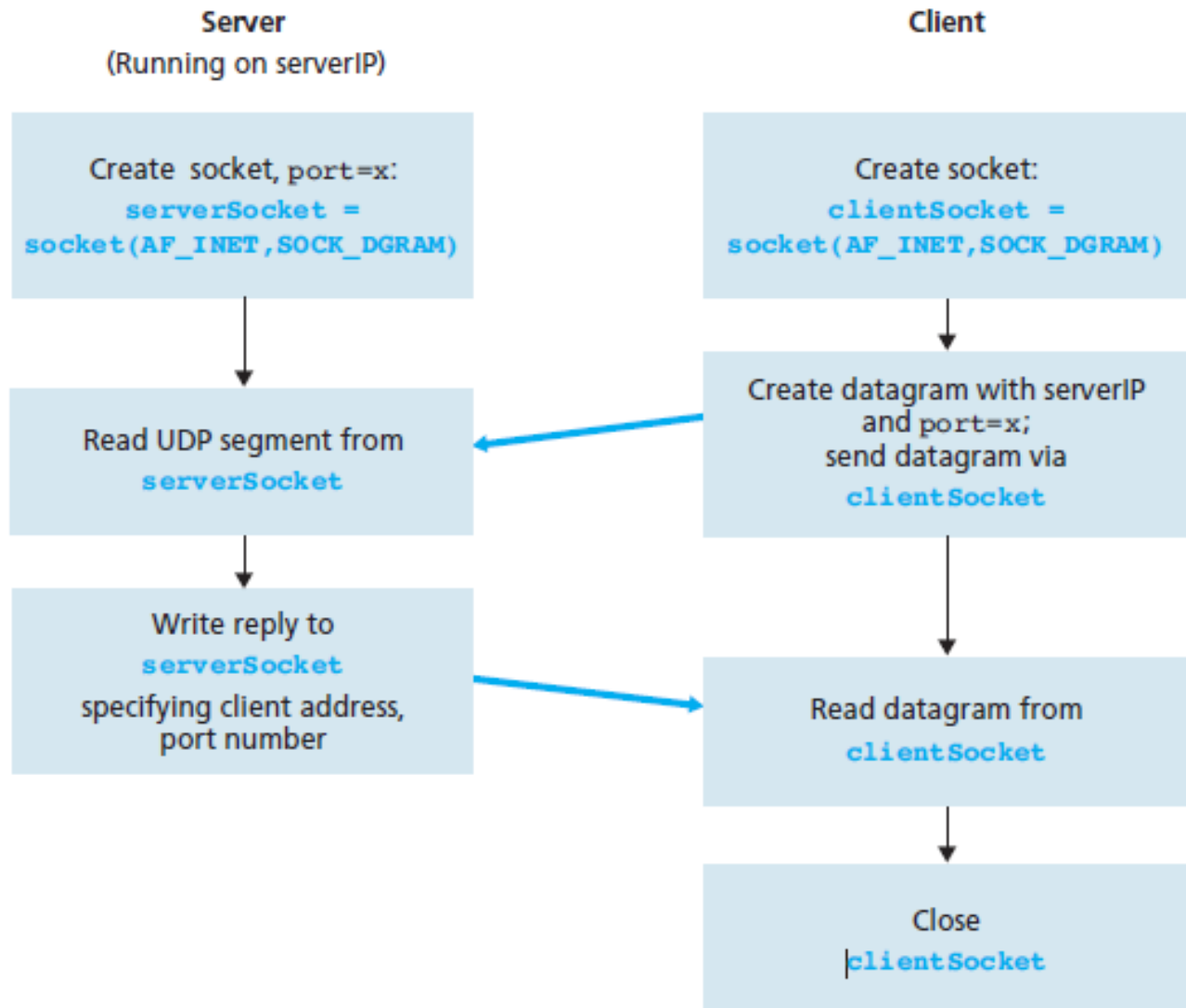
Dịch vụ WinSock

- Các thao tác cơ sở
 - Liên kết chương trình ứng dụng với socket
 - Khởi tạo, chấp nhận kết nối
 - Gởi nhận dữ liệu
 - Đóng kết nối
- Các hàm bất đồng bộ
- Các hàm chuyển đổi dữ liệu

Các dạng socket

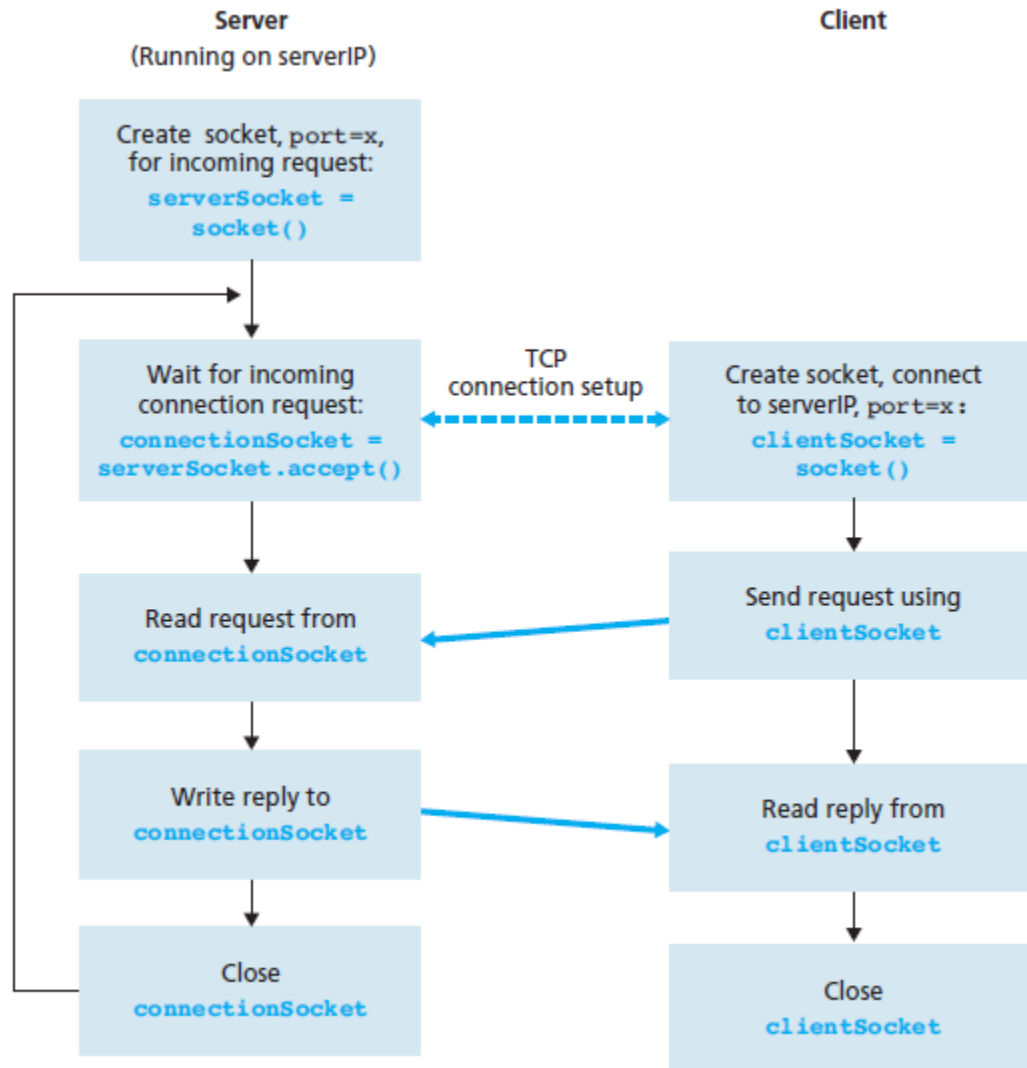
- Stream socket
 - Trao đổi dữ liệu tin cậy 2 chiều dùng TCP
- Datagram socket
 - Trao đổi dữ liệu 2 chiều dùng UDP
- Socket được định nghĩa theo:
 - Giao thức sử dụng
 - Địa chỉ

Ví dụ mô hình Client-Server dùng UDP



The client-server application using UDP

Ví dụ mô hình Client-Server dùng TCP



The client-server application using TCP