

# LẬP TRÌNH MẠNG

---

Phạm Huy Hoàng

Bộ môn Truyền thông và Mạng máy tính

Viện CNTT – TT, Đại học BKHN

# Thông tin học phần

- **Mã học phần:** IT4060
- **Khối lượng:** 2(2-1-0-4)
- **Học phần học trước:** IT3080 - Mạng máy tính
- **Mục tiêu:** Trang bị cho sinh viên các kiến thức và kỹ năng lập trình mạng TCP/IP.
- **Đánh giá:**
  - Quá trình: Chuyên cần, trao đổi thông tin, báo cáo bài tập lớn
  - Cuối kỳ: Trình bày & bảo vệ chương trình bài tập lớn

# BÀI 1. MỞ ĐẦU

---

# Nội dung

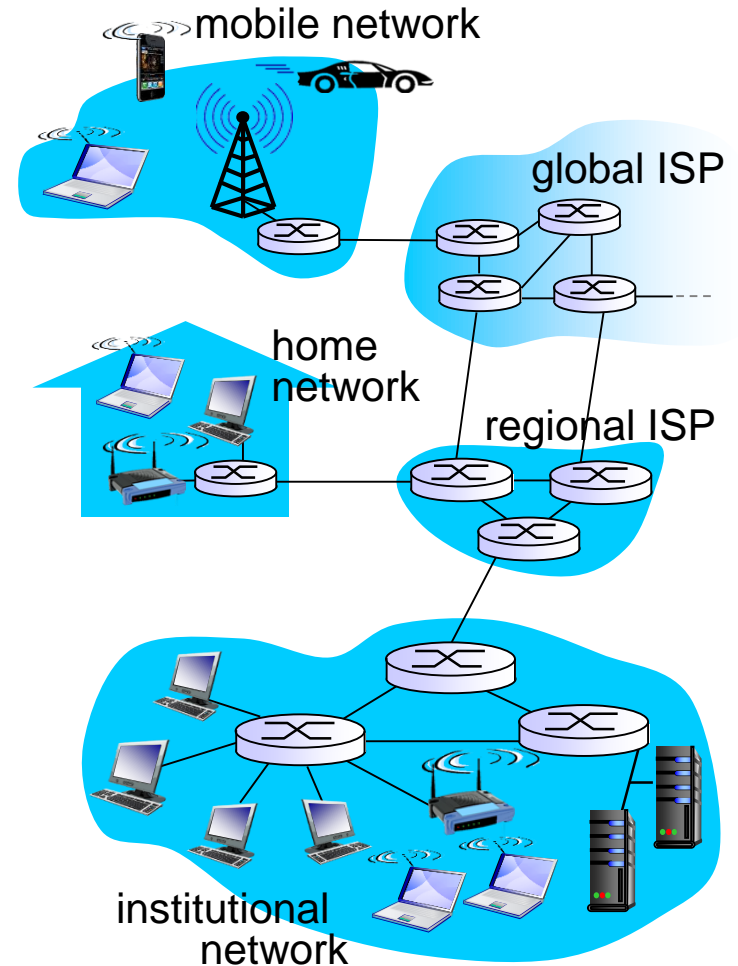
- Cơ bản về mạng máy tính
- Nguyên lý tầng ứng dụng
- Nguyên lý tầng giao vận
- Địa chỉ IPv4
- Thư viện lập trình

# 1. CƠ BẢN VỀ MẠNG MÁY TÍNH

---

# Mạng máy tính là gì?

- Tập hợp các máy tính kết nối với nhau dựa trên một kiến trúc nào đó để có thể trao đổi dữ liệu
  - Máy tính: máy trạm, máy chủ, bộ định tuyến
  - Kết nối bằng một phương tiện truyền
  - Theo một kiến trúc mạng



# Mạng máy tính là gì?

- Phương tiện truyền: đường truyền vật lý:
  - Hữu tuyến: cáp đồng, cáp quang
  - Vô tuyến: sóng hồng ngoại, sóng radio
- Kiến trúc mạng:
  - Hình trạng mạng: cách thức các máy tính kết nối bằng đường truyền vật lý với nhau
  - Giao thức mạng: cách thức các máy tính trao đổi dữ liệu với nhau như thế nào?
- Hoạt động cơ bản trên hệ thống mạng máy tính: truyền thông tin từ máy tính này sang máy tính khác
  - Tương tự như con người trao đổi thư tín qua hệ thống bưu điện
  - Máy nguồn: gửi dữ liệu
  - Máy đích: nhận dữ liệu

# Phân loại mạng máy tính

- Mạng cá nhân (PAN – Personal Area Network)
  - Phạm vi kết nối: vài chục mét
  - Số lượng người dùng: một vài người dùng
  - Thường phục vụ cho cá nhân
- Mạng cục bộ (LAN – Local Area Network):
  - Phạm vi kết nối: vài ki-lô-mét
  - Số lượng người dùng: một vài đến hàng trăm nghìn
  - Thường phục vụ cho cá nhân, hộ gia đình, tổ chức



# Phân loại mạng máy tính

- Mạng đô thị (MAN – Metropolitan Area Network)
  - Phạm vi kết nối: hàng trăm ki-lô-mét
  - Số lượng người dùng: hàng triệu
  - Phục vụ cho thành phố, khu vực
- Mạng diện rộng (WAN – Wide Area Network)
  - Phạm vi kết nối: vài nghìn ki-lô-mét
  - Số lượng người dùng: hàng tỉ
  - GAN – Global Area Network: phạm vi toàn cầu (Ví dụ: Internet)

# Trao đổi thông tin giữa các nút mạng

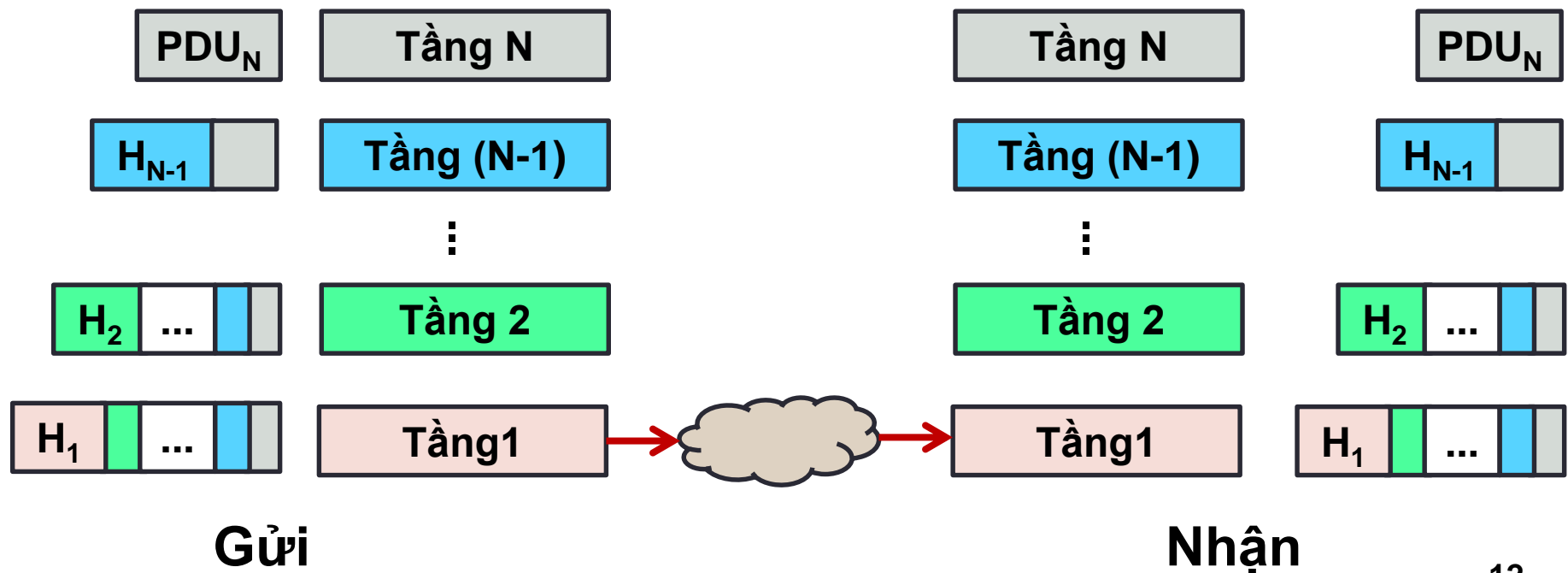
- Dữ liệu được tổ chức như thế nào?
  - Định danh – đánh địa chỉ: Phân biệt các máy với nhau trên mạng?
  - Tìm đường đi cho dữ liệu qua hệ thống mạng như thế nào?
  - Làm thế nào để phát hiện lỗi dữ liệu (và sửa)?
  - Làm thế nào để dữ liệu gửi đi không làm quá tải đường truyền, quá tải máy nhận?
  - Làm thế nào để chuyển dữ liệu thành tín hiệu?
  - Làm thế nào để biết dữ liệu đã tới đích?...
- Phân chia nhiệm vụ cho các thành phần, tổ chức các thành phần thành các tầng (layer)

# Phân tầng

- Mỗi tầng:
  - Có thể có một hoặc nhiều chức năng
  - Triển khai dịch vụ để thực hiện các chức năng
    - Cung cấp dịch vụ cho tầng trên
    - Sử dụng dịch vụ tầng dưới
    - Độc lập với các tầng còn lại
  - Mỗi dịch vụ có thể có một hoặc nhiều cách triển khai khác nhau, cho phép tầng trên lựa chọn dịch vụ phù hợp
- Lợi ích:
  - Dễ dàng thiết kế, triển khai
  - Dễ dàng tái sử dụng
  - Dễ dàng nâng cấp

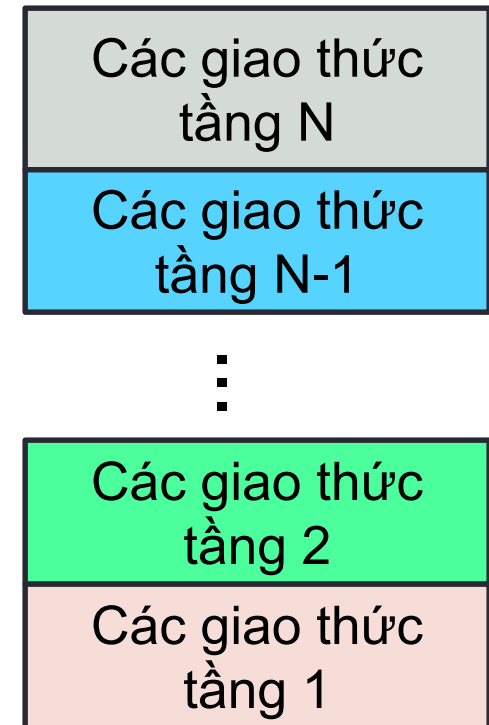
# Truyền thông trong kiến trúc phân tầng

- Bên gửi: thêm tiêu đề chứa thông tin phục vụ cho việc xử lý dữ liệu tại tầng tương ứng và chuyển cho tầng dưới (Đóng gói dữ liệu – Encapsulation)
- Bên nhận: xử lý dữ liệu theo thông tin trong phần tiêu đề, tách tiêu đề và chuyển dữ liệu cho tầng trên

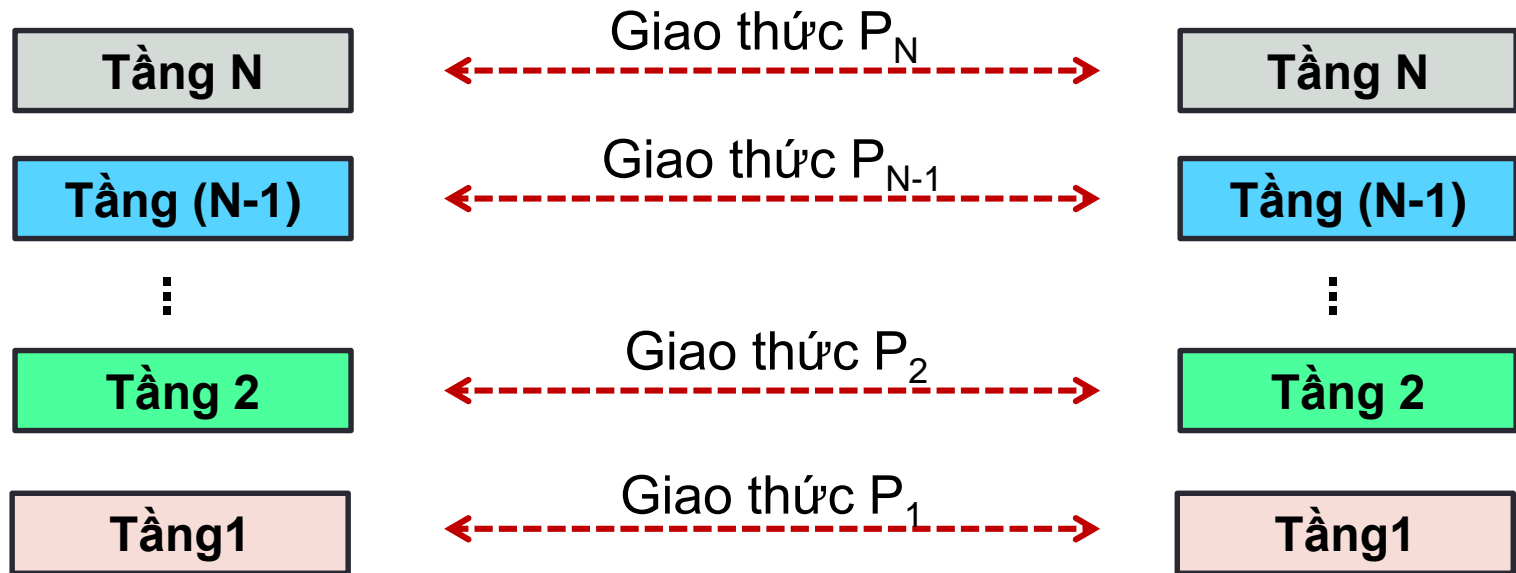


# Chồng giao thức (Protocol stack)

- Giao thức: Là tập hợp các quy tắc quy định *khuôn dạng, ngữ nghĩa, thứ tự* các *thông điệp được gửi và nhận giữa các nút mạng* và *các hành vi* khi trao đổi các thông điệp đó
  - Mỗi tầng có nhiều cách thức để thực hiện các chức năng → sinh ra các giao thức khác nhau
- chồng giao thức: ngăn xếp các giao thức truyền thông trên kiến trúc phân tầng



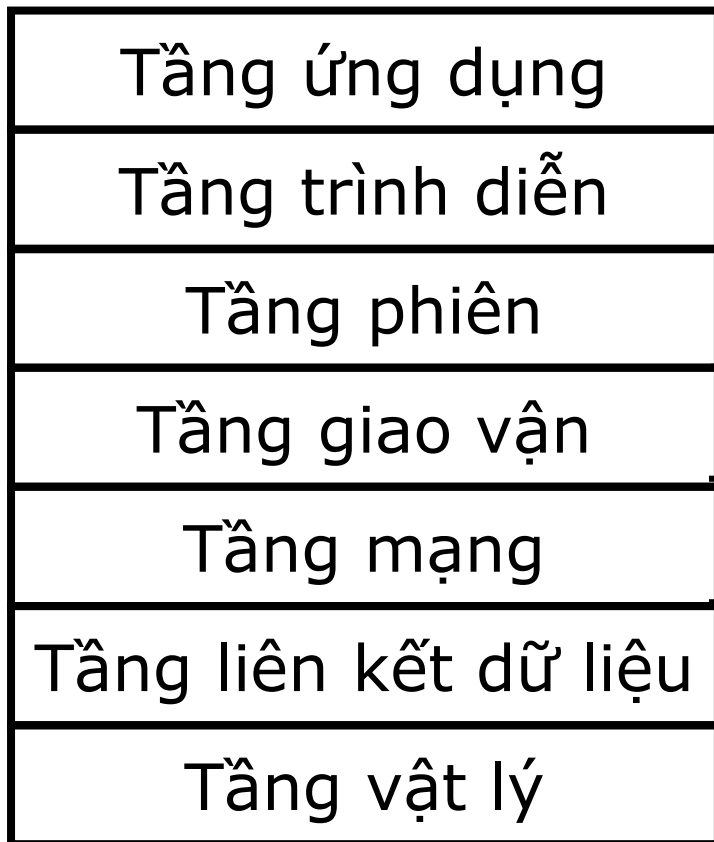
# Truyền thông trong kiến trúc phân tầng (tiếp)



- Các tầng đồng cấp ở mỗi bên sử dụng chung giao thức để điều khiển quá trình truyền thông logic giữa chúng
  - 2 cách thức để giao thức điều khiển truyền thông logic giữa các tầng đồng cấp: hướng liên kết hoặc hướng không liên kết

# Mô hình OSI và mô hình TCP/IP

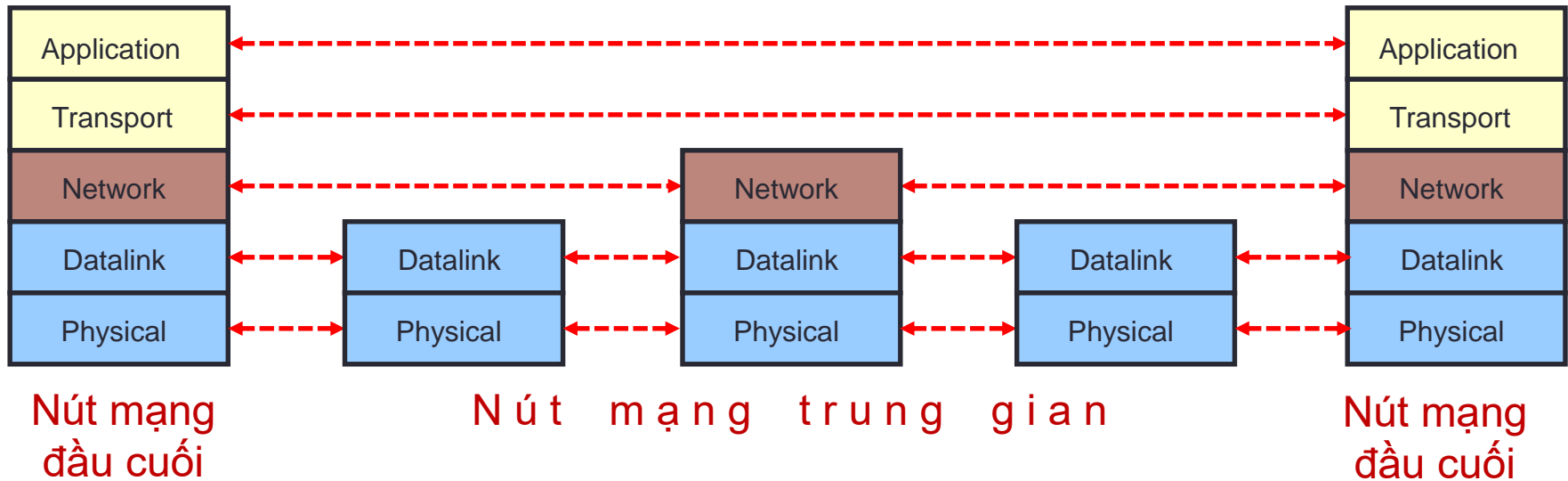
Mô hình OSI



Mô hình TCP/IP



# Triển khai kiến trúc phân tầng



- Nút mạng đầu cuối (end-system): PC, server, smartphone...
- Nút mạng trung gian: các thiết bị mạng chuyển tiếp dữ liệu



# Định danh trên kiến trúc phân tầng

- Tầng ứng dụng : **tên miền** định danh cho máy chủ cung cấp dịch vụ
  - Tên miền: chuỗi ký tự dễ nhớ với người dùng. Thiết bị mạng không dùng tên miền khi truyền tin
  - Ví dụ: mps.gov.vn (máy chủ Web của Bộ CA)
- Tầng giao vận: **số hiệu cổng** định danh cho các dịch vụ khác nhau
  - Số hiệu cổng: từ 0-65535
  - Ví dụ: Web-80, DNS-53, Email(SMTP-25, POP-110, IMAP-143)
- Tầng mạng: **địa chỉ IP** định danh cho các máy trạm, máy chủ, bộ định tuyến
  - Có thể dùng trong mạng nội bộ và mạng Internet
  - Địa chỉ IPv4: 4 số có giá trị từ 0-255, các nhau bởi 1 dấu ‘.’
  - Ví dụ: 123.30.9.222 (máy chủ Web của Bộ CA)
- Tầng liên kết dữ liệu: **địa chỉ MAC** định danh cho các máy trạm, máy chủ, thiết bị mạng
  - Chỉ dùng trong mạng nội bộ

## 2. NGUYÊN LÝ TẦNG ỨNG DỤNG

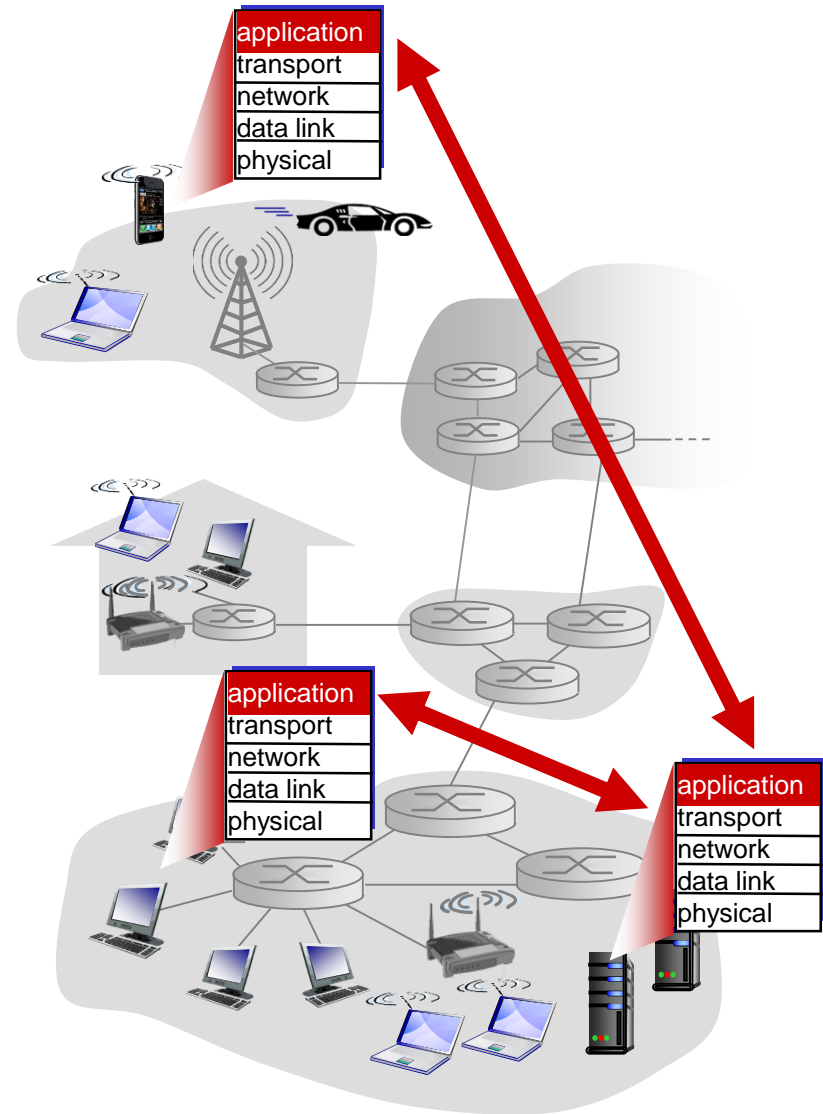
---

# Mô hình TCP/IP – Tầng ứng dụng

- Cung cấp dịch vụ mạng cho người dùng
- Phối hợp hoạt động của chương trình client và chương trình server
  - Client: cung cấp giao diện cho người dùng
  - Server: đáp ứng dịch vụ
- Một số dịch vụ tiêu biểu: Web, Email, Lưu trữ và chia sẻ file (FTP)...
- Mô hình cung cấp dịch vụ:
  - Client/Server
  - Ngang hàng
  - Mô hình lai
- Giao thức tầng ứng dụng

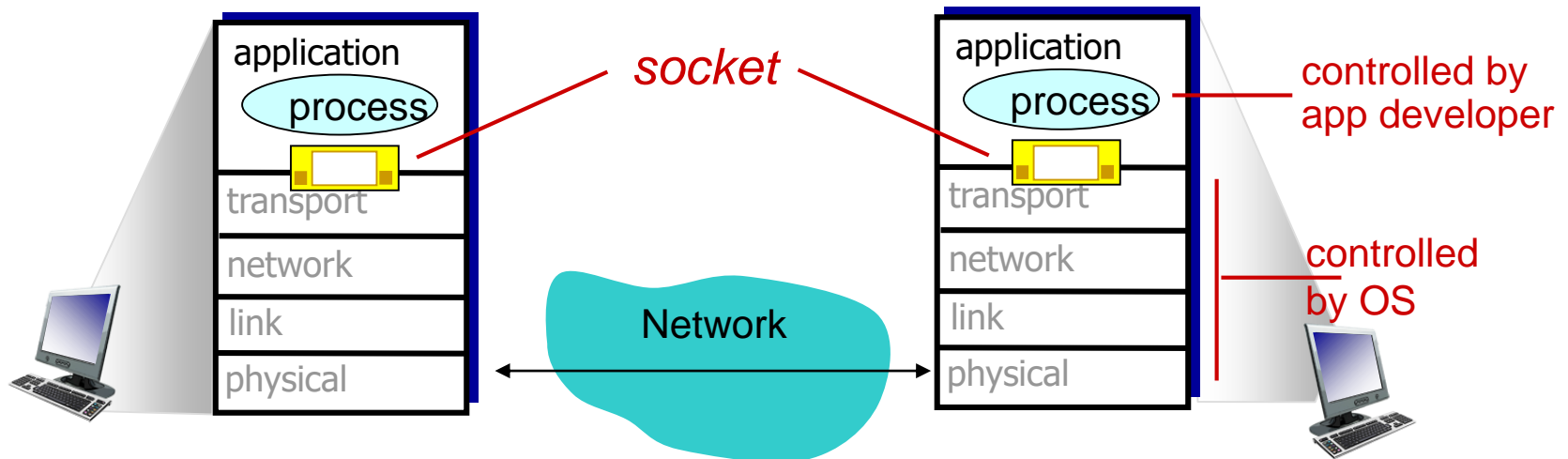
# Ứng dụng mạng

- Hoạt động trên các hệ thống đầu cuối (end system)
- Cài đặt giao thức ứng dụng để cung cấp dịch vụ
- Gồm có 2 tiến trình giao tiếp với nhau qua môi trường mạng:
  - Client: cung cấp giao diện NSD, gửi thông điệp yêu cầu dịch vụ
  - Server: cung cấp dịch vụ, trả thông điệp đáp ứng
- Ví dụ: Web
  - Web browser (trình duyệt Web): Chrome, Firefox...
  - Web server: Apache, Tomcat...

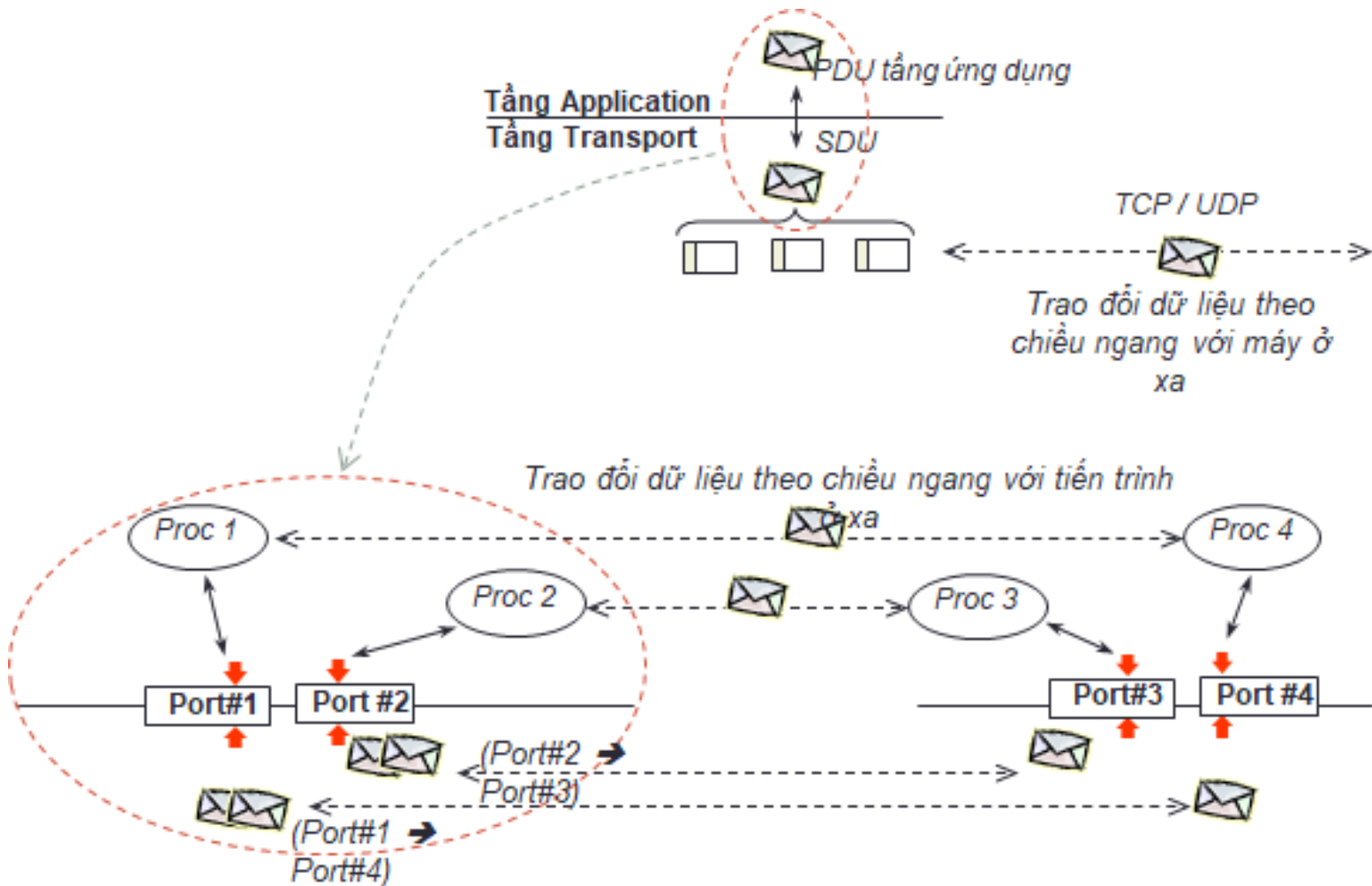


# Giao tiếp giữa các tiến trình ứng dụng

- Socket: SAP của tầng giao vận
  - Các tiến trình ứng dụng sử dụng socket gọi dịch vụ của tầng giao vận để trao đổi thông điệp
- Định danh cho tiến trình bởi: Địa chỉ IP, Số hiệu cổng
- Ví dụ: tiến trình web server trên máy chủ của SoICT có định danh 202.191.56.65:80

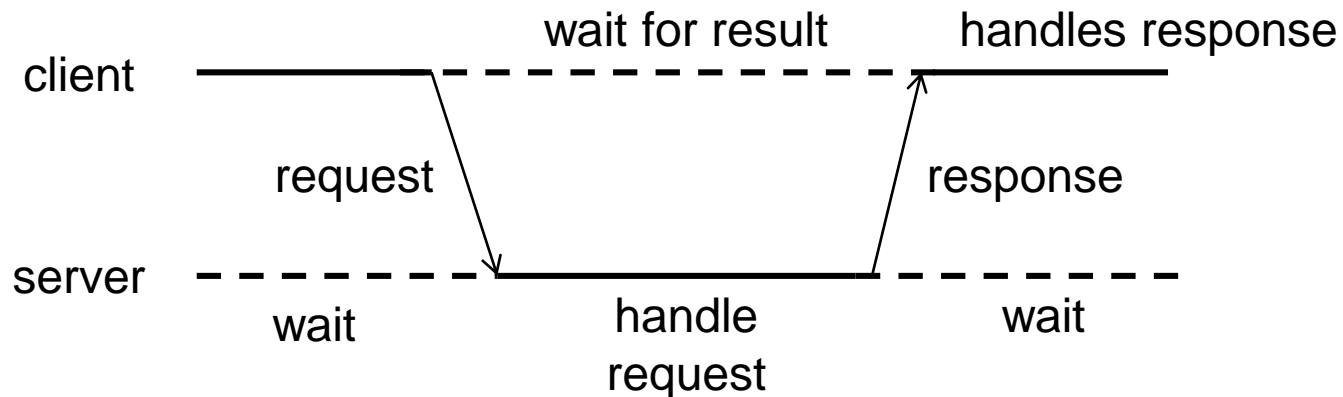


# Vai trò của Port trong giao tiếp các App

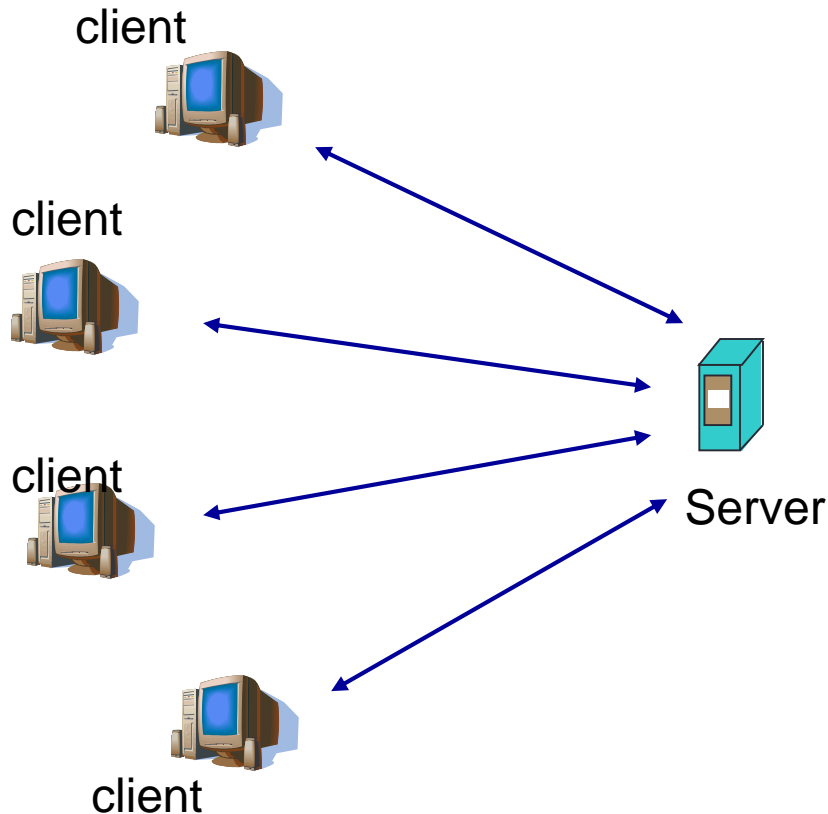


# Giao tiếp giữa các tiến trình

- Tiến trình client: gửi yêu cầu
- Tiến trình server: trả lời
- Mô hình điển hình: 1 server – nhiều client
- Client cần biết địa chỉ của server: địa chỉ IP, số hiệu cổng



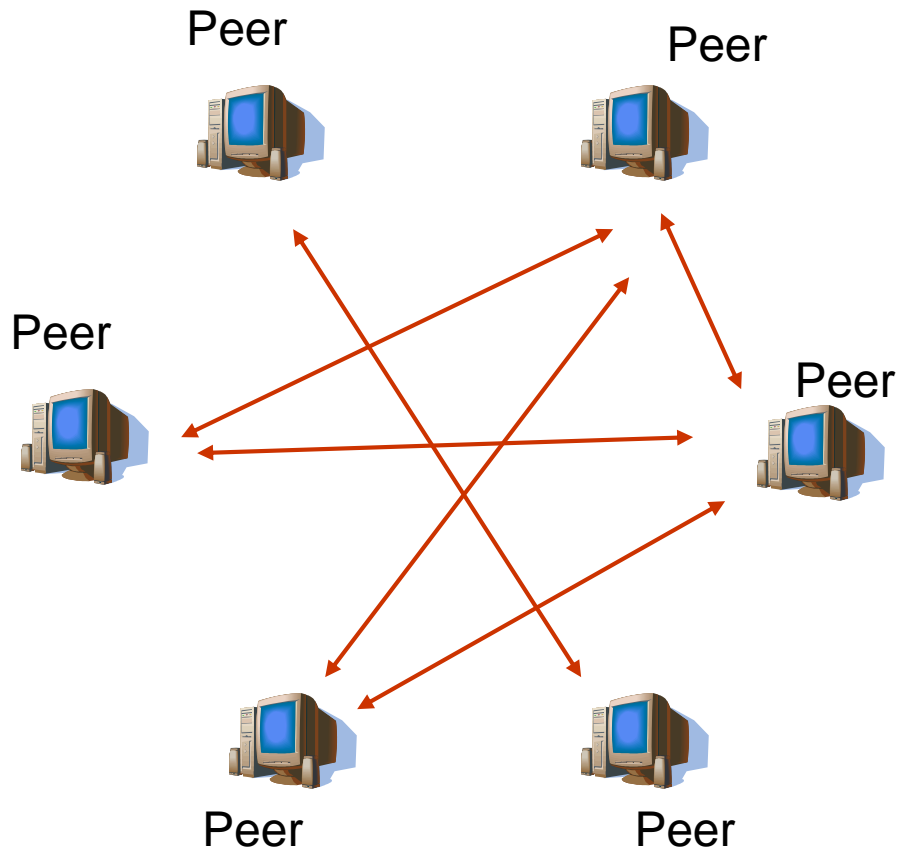
# Mô hình khách chủ



- Khách
  - Gửi yêu cầu truy cập dịch vụ đến máy chủ
  - Về nguyên tắc, không liên lạc trực tiếp với các máy khách khác
- Chủ
  - Thường xuyên online để chờ y/c đến từ máy trạm
  - Có thể có máy chủ dự phòng để nâng cao hiệu năng, phòng sự cố
- e.g. Web, Mail, ...

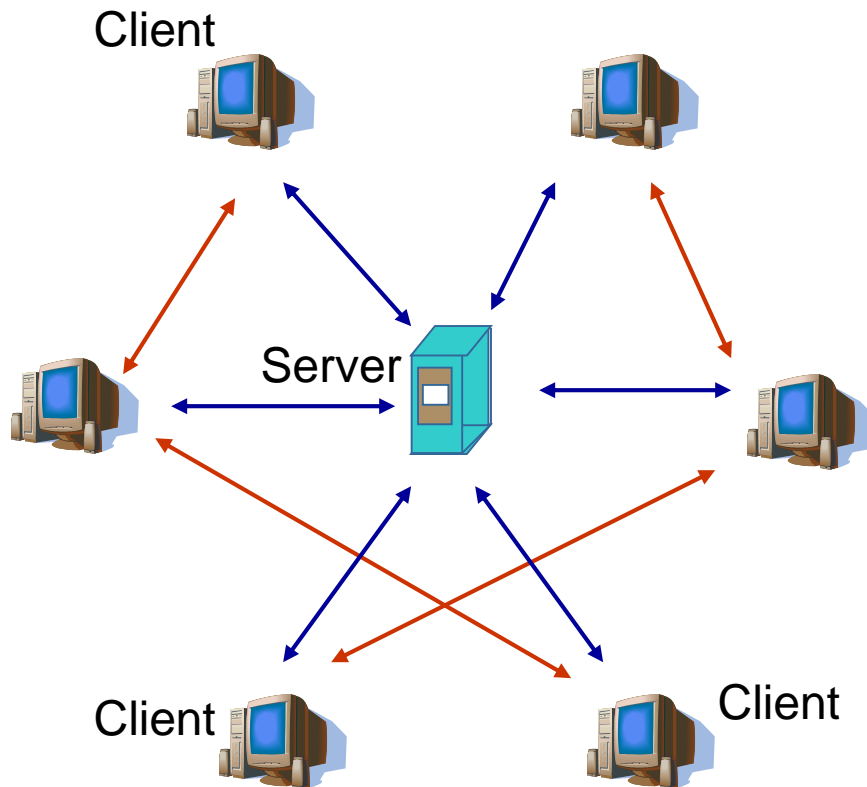


# Mô hình ngang hàng thuần túy



- Không có máy chủ trung tâm
- Các máy có vai trò ngang nhau
- Hai máy bất kỳ có thể liên lạc trực tiếp với nhau
- Không cần vào mạng thường xuyên
- E.g. Gnutella

# Mô hình lai



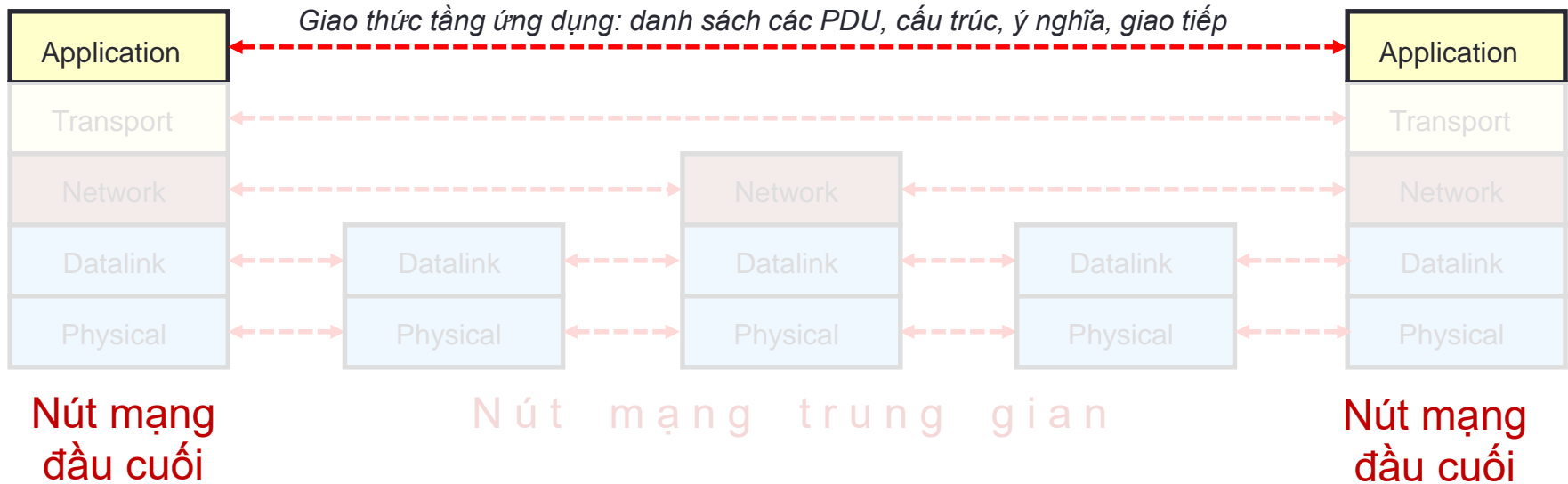
P2P Comm.



Client-Server Comm.

- Một máy chủ trung tâm để quản lý NSD, thông tin tìm kiếm...
- Các máy khách sẽ giao tiếp trực tiếp với nhau sau khi đăng nhập
- E.g. Skype
  - Máy chủ Skype quản lý các phiên đăng nhập, mật khẩu...
  - Sau khi kết nối, các máy sẽ gọi VoIP trực tiếp cho nhau

# Giao thức tầng ứng dụng



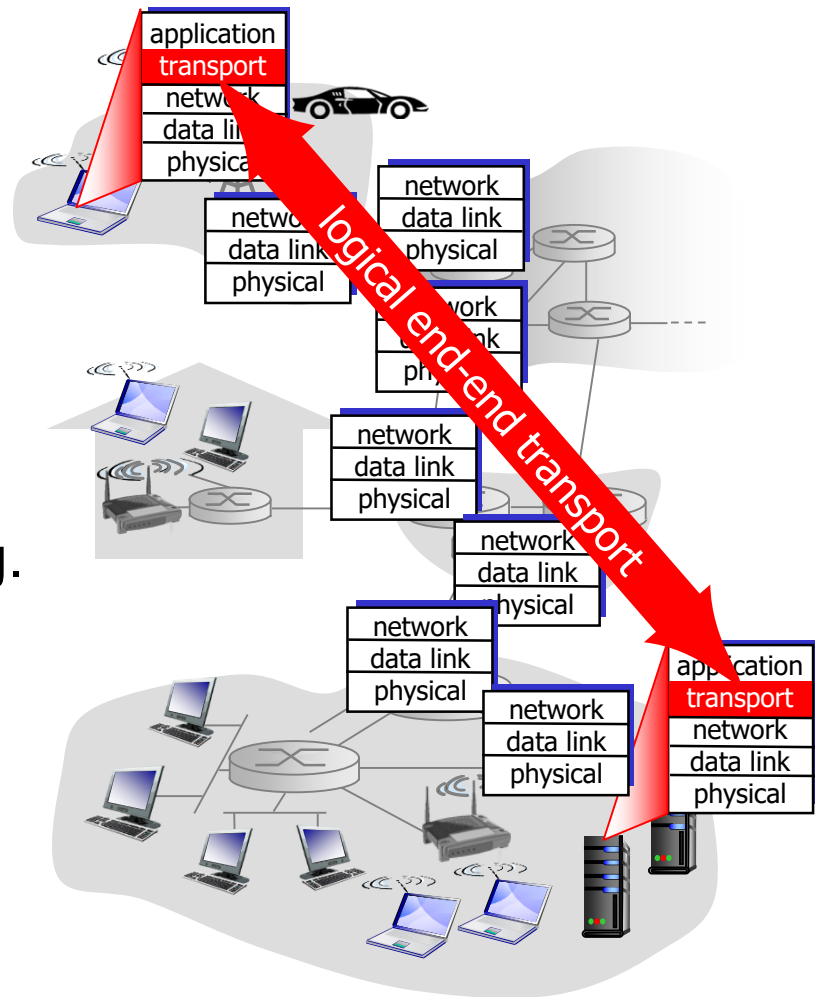
- Sử dụng free text để thiết kế PDU (header, data)
- Tập trung vào bài toán của ứng dụng, không quan tâm đến các vấn đề truyền dữ liệu tầng dưới
- Quan tâm đến các vấn đề đồng bộ client/server, client/client, bảo mật dữ liệu

## 2. NGUYÊN LÝ CHUNG TẦNG GIAO VẬN

---

# Tầng giao vận

- Được cài đặt trên các hệ thống cuối
- Cung cấp dịch vụ để các ứng dụng mạng trao đổi dữ liệu
- Hai dạng dịch vụ giao vận
  - Tin cậy, hướng liên kết, e.g TCP
  - Không tin cậy, không liên kết, e.g. UDP
- Đơn vị truyền: datagram (UDP), segment (TCP)



# Thông số của liên kết

- Mỗi một liên kết tạo ra trên tầng giao vận để vận chuyển dữ liệu cho tiến trình tầng ứng dụng của 2 nút mạng được xác định bởi bộ 5 thông số (5-tuple):

- Địa chỉ IP nguồn
  - Địa chỉ IP đích
  - Số hiệu cổng nguồn
  - Số hiệu cổng đích
  - Giao thức (TCP/UDP,...)
- } Tầng mạng
- } Tầng giao vận

# GIAO THỨC UDP

---

# Đặc điểm giao thức

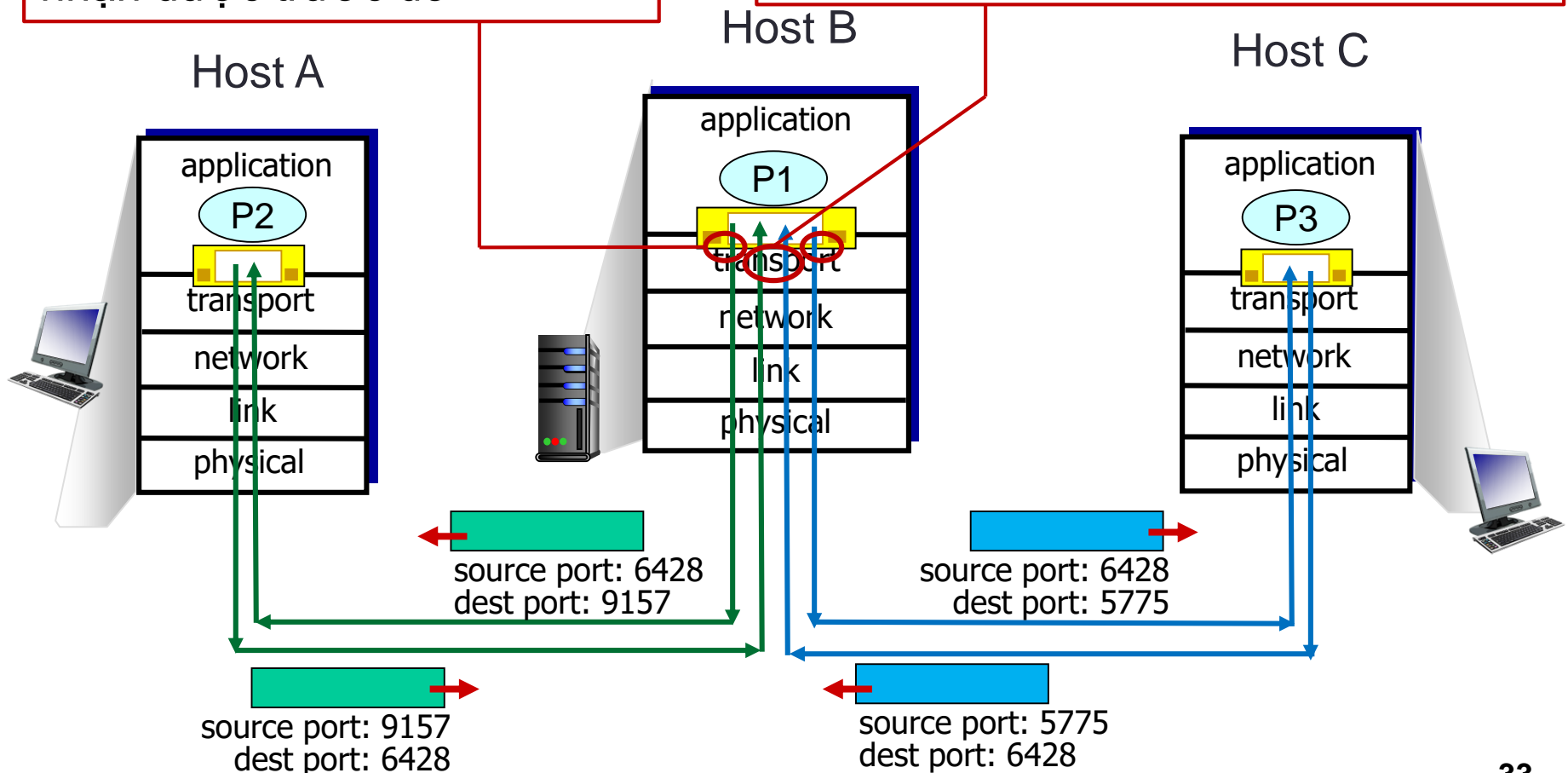
- Giao thức hướng không kết nối (connectionless)
- Truyền liệu theo datagram và “best-effort”
- Vì sao cần UDP?
  - Không cần thiết lập liên kết (giảm độ trễ)
  - Đơn giản: Không cần lưu lại trạng thái liên kết ở bên gửi và bên nhận
  - Phần đầu đoạn tin nhỏ
- Không có quản lý tắc nghẽn: UDP cứ gửi dữ liệu nhanh nhất, nhiều nhất nếu có thể
- Không bảo đảm được độ tin cậy
  - Nếu cần các ứng dụng phải cài đặt cơ chế tự kiểm soát độ tin cậy
  - Việc phát triển ứng dụng sẽ phức tạp hơn



# UDP socket trên ứng dụng mạng

Gửi dữ liệu tới đúng tiến trình đích bằng địa chỉ IP nguồn và cổng nguồn trên dữ liệu nhận được trước đó

Nhận dữ liệu: Dựa trên số hiệu cổng đích trên bức tin để đưa dữ liệu đến đúng socket



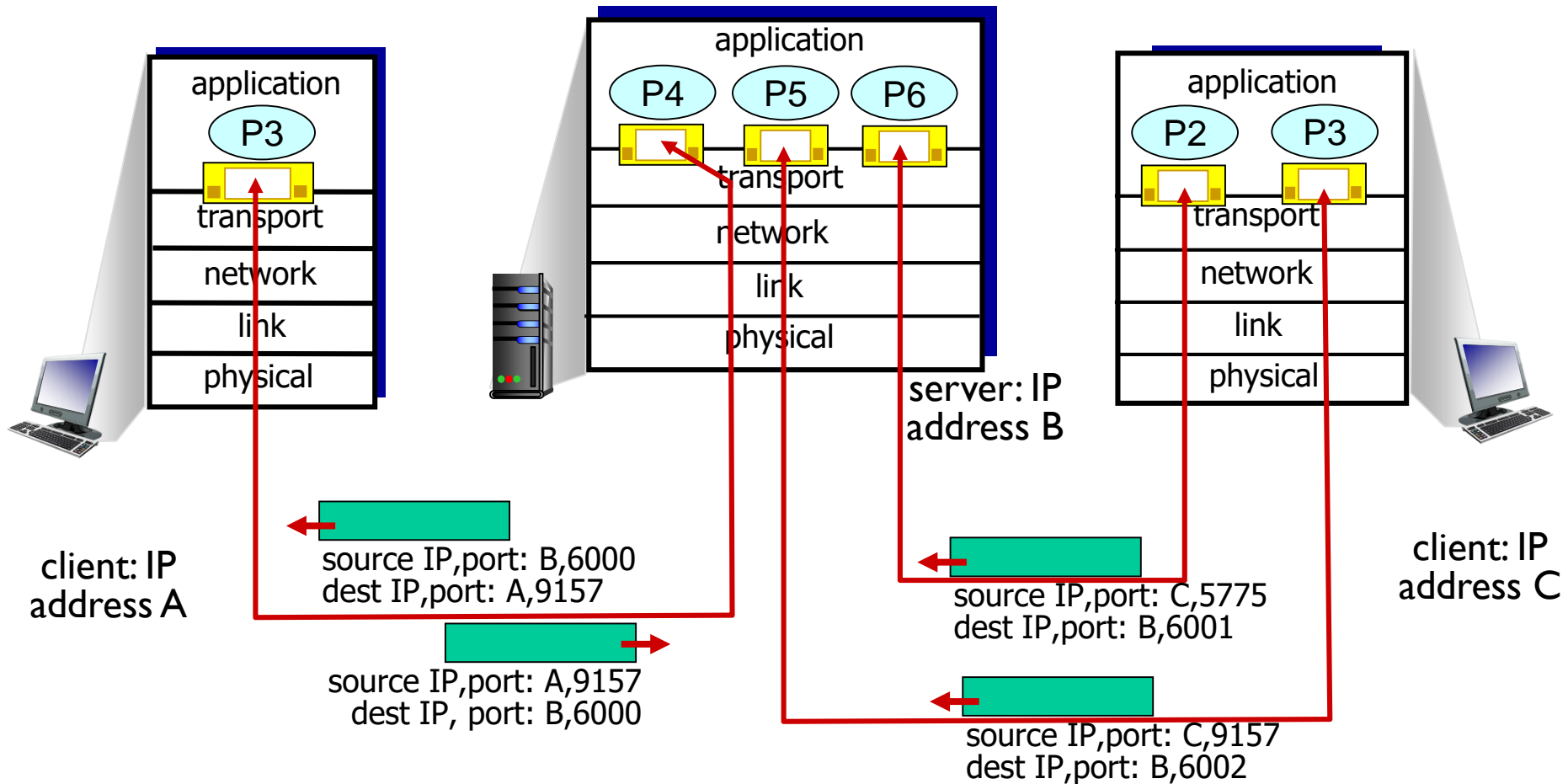
# GIAO THỨC TCP

---

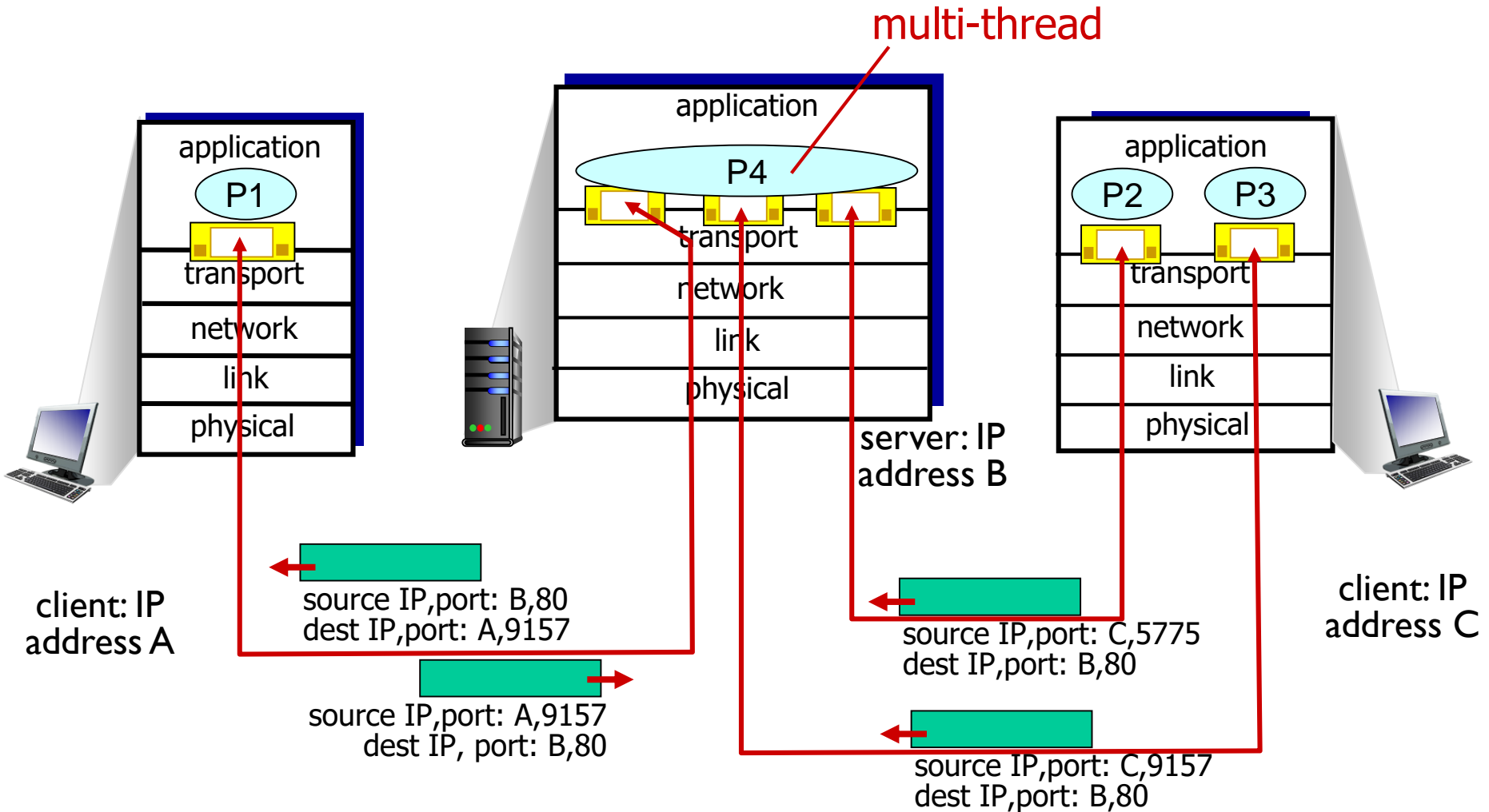
# Đặc điểm giao thức

- Giao thức hướng liên kết
  - Bắt tay ba bước
- Giao thức truyền dữ liệu theo dòng byte, tin cậy
  - Sử dụng vùng đệm
- Truyền theo kiểu pipeline
  - Tăng hiệu quả
- Kiểm soát luồng
  - Bên gửi không làm quá tải bên nhận (thực tế: quá tải)
- Kiểm soát tắc nghẽn
  - Việc truyền dữ liệu không nên làm tắc nghẽn mạng (thực tế: luôn có tắc nghẽn)

# TCP socket trên ứng dụng mạng



# TCP socket trên ứng dụng mạng

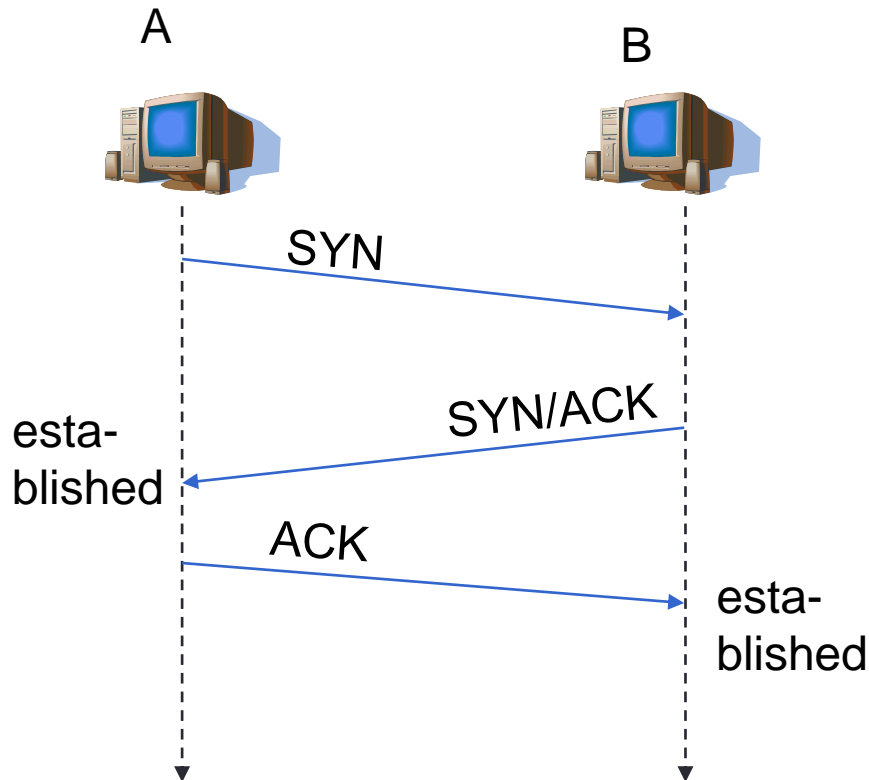


# TCP cung cấp dịch vụ tin cậy ntn?

- Kiểm soát lỗi dữ liệu: checksum
- Kiểm soát mất gói tin: phát lại khi có time-out
- Kiểm soát dữ liệu đã được nhận chưa:
  - Seq. #
  - Ack

} Cơ chế báo nhận
- Chu trình làm việc của TCP:
  - Thiết lập liên kết
    - Bắt tay ba bước
  - Truyền/nhận dữ liệu
  - Đóng liên kết

# Thiết lập liên kết TCP : Giao thức bắt tay 3 bước



- **Bước 1:** A gửi SYN cho B
  - chỉ ra giá trị khởi tạo seq # của A
  - không có dữ liệu
- **Bước 2:** B nhận SYN, trả lời bằng SYN/ACK
  - B khởi tạo vùng đệm
  - chỉ ra giá trị khởi tạo seq. # của B
- **Bước 3:** A nhận SYNACK, trả lời ACK, có thể kèm theo dữ liệu

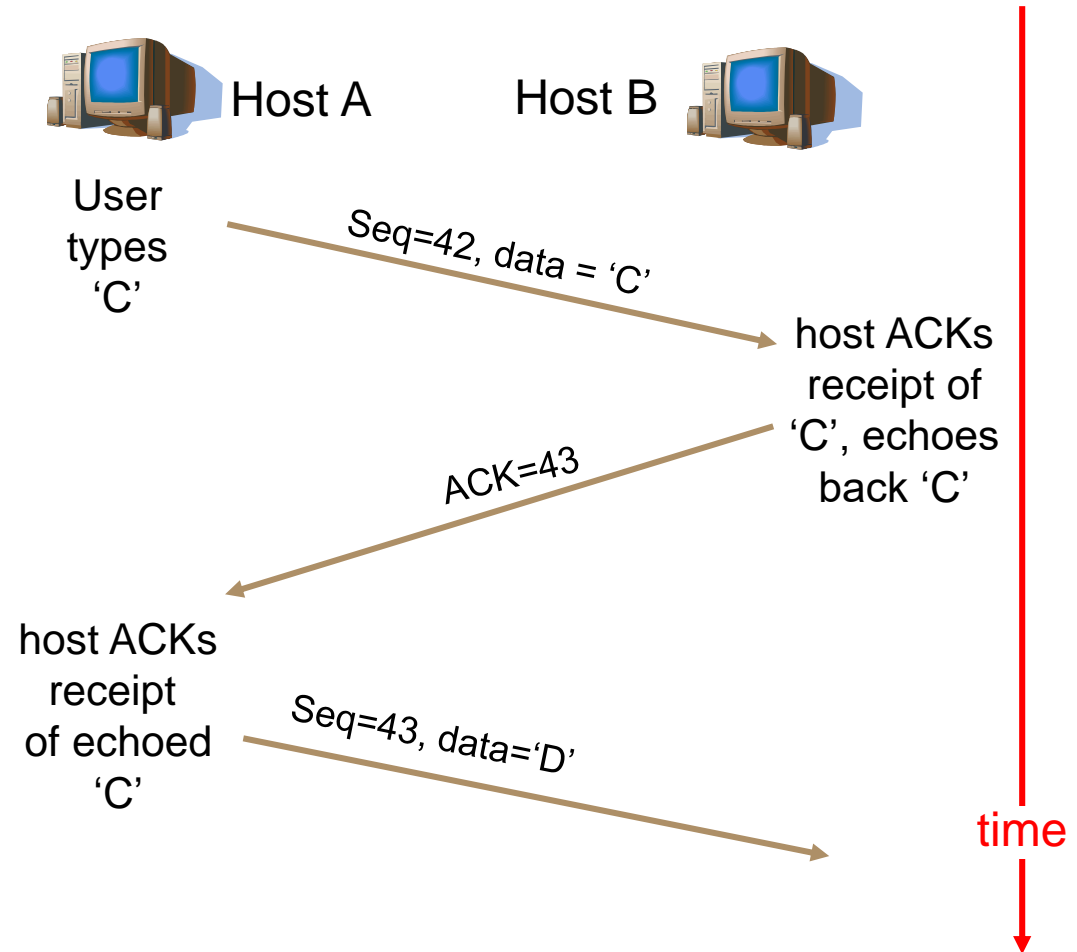
# Cơ chế báo nhận trong TCP

## Seq. #:

- Số hiệu của byte đầu tiên của đoạn tin trong dòng dữ liệu

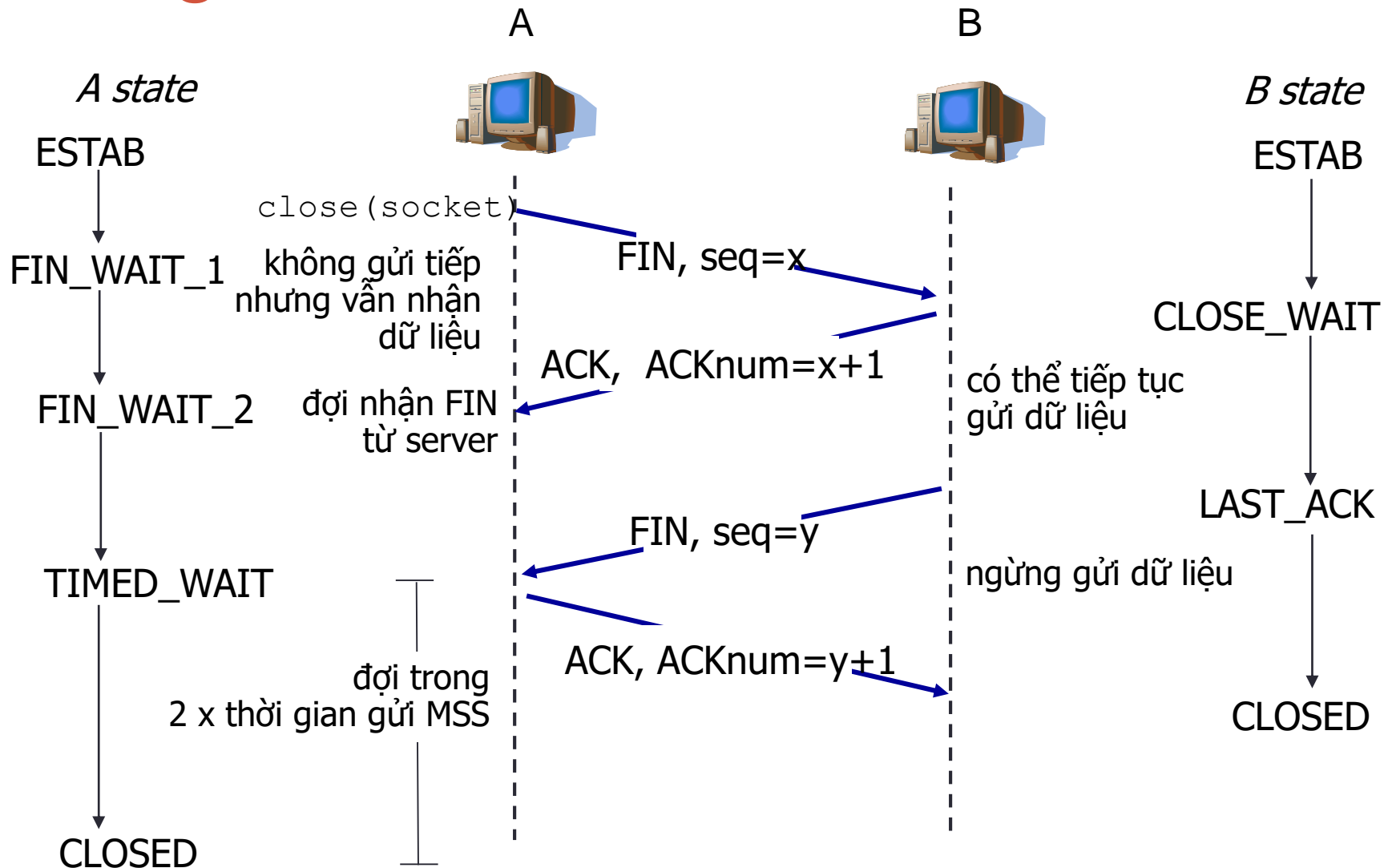
## ACK:

- Số hiệu byte đầu tiên mong muốn nhận từ đối tác





# Đóng liên kết



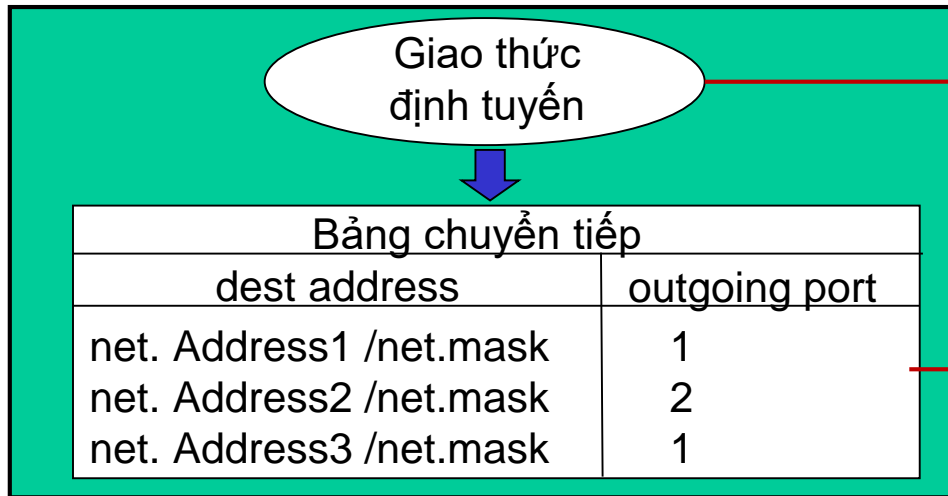
### 3. NGUYÊN LÝ CHUNG TẦNG MẠNG

---

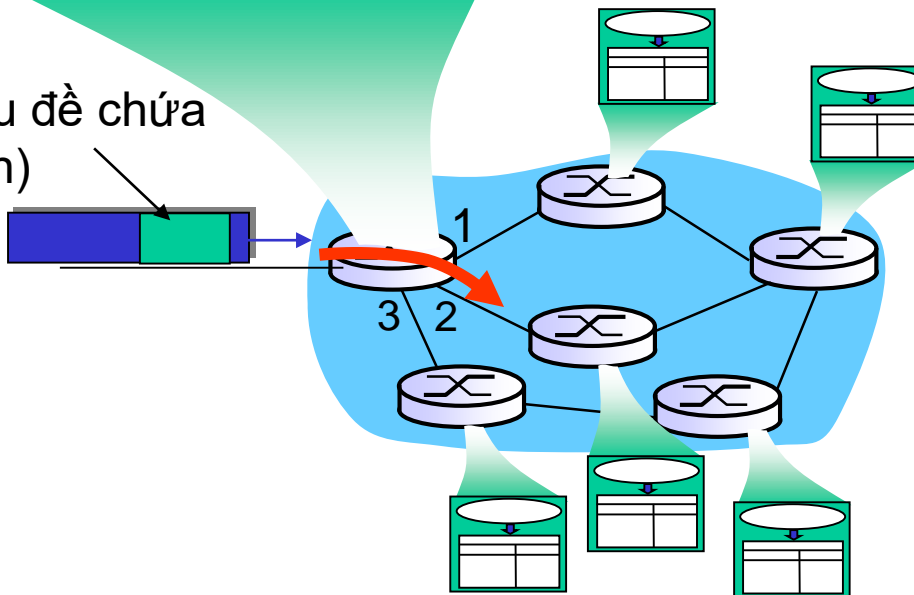
# Mô hình TCP/IP - Tầng mạng

- Cung cấp các cơ chế để kết nối các hệ thống mạng với nhau (internetworking)
  - Mạng của các mạng
- Giao thức IP : Internet Protocol
  - Định danh: sử dụng địa chỉ IP để gán cho các nút mạng (máy trạm, máy chủ, bộ định tuyến)
  - Khuôn dạng dữ liệu
- Định tuyến(chọn đường): tìm các tuyến đường tốt nhất qua hệ thống trung gian để gửi thông tin
- Chuyển tiếp: quyết định gửi dữ liệu qua tuyến đường nào

# Định tuyến và chuyển tiếp

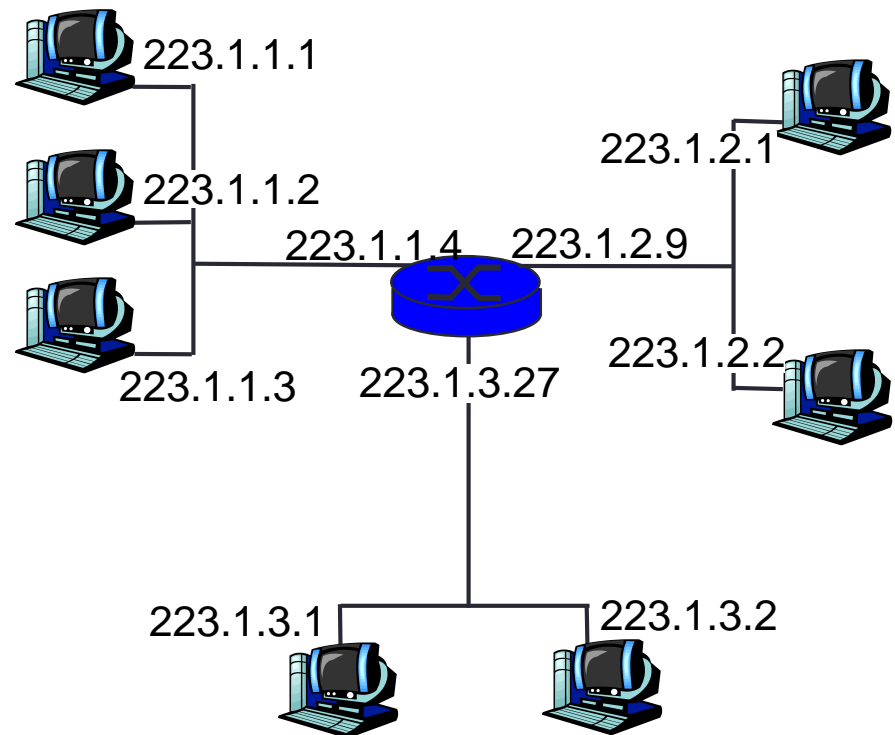


Gói tin (tiêu đề chứa địa chỉ đích)



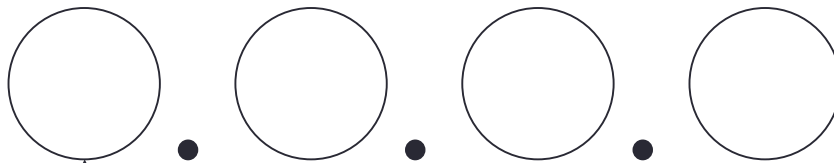
# Địa chỉ IP (IPv4)

- **Địa chỉ IP:** Một số 32-bit để định danh cổng giao tiếp mạng trên nút đầu cuối (PC, server, smart phone), bộ định tuyến
- Mỗi địa chỉ IP được gán cho một cổng duy nhất
- Địa chỉ IP có tính duy nhất trong mạng



$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_{15}$$

# Biểu diễn địa chỉ IPv4



8 bits

0 – 255 integer

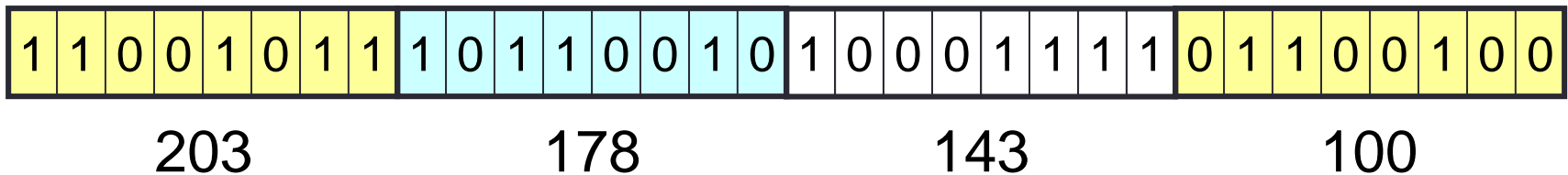
Ví dụ:

203.178.136.63      o

259.12.49.192      x

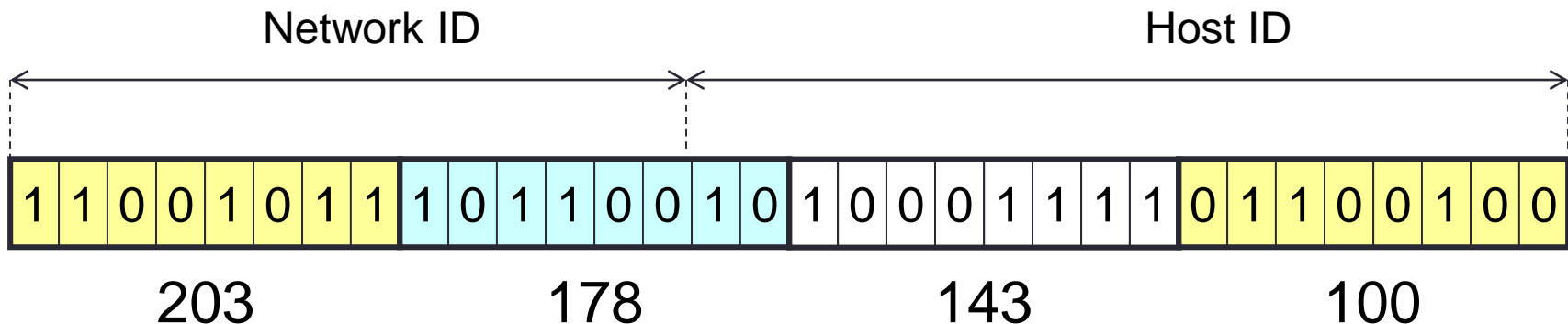
133.27.4.27      o

Sử dụng 4 phần 8 bits để miêu tả một địa chỉ 32 bits



# Địa chỉ IPv4

- Địa chỉ IP có hai phần
  - Host ID – phần địa chỉ máy trạm
  - Network ID – phần địa chỉ mạng



- Làm thế nào biết được phần nào là cho máy trạm, phần nào cho mạng?
  - Phân lớp địa chỉ
  - Không phân lớp – CIDR

# Phân lớp địa chỉ IP(Classful Addressing)



Class A	0	7 bit			H	H	H		
Class B	1	0	6 bit			N	H	H	
Class C	1	1	0	5 bit			N	N	H
Class D	1	1	1	0	Multicast				
Class E	1	1	1	1	Reserve for future use				

	# of network	# of hosts
Class A	128	$2^{24} - 2$
Class B	16384	65534
Class C	$2^{21}$	254



# Các dạng địa chỉ

- Địa chỉ mạng (Network Address):
  - Định danh cho một mạng
  - Tất cả các bit phần HostID là 0
- Địa chỉ quảng bá (Broadcast Address)
  - Địa chỉ dùng để gửi dữ liệu cho tất cả các máy trạm trong mạng
  - Tất cả các bit phần HostID là 1
- Địa chỉ máy trạm
  - Gán cho một cổng mạng
- Địa chỉ nhóm (Multicast address): định danh cho nhóm

# Không gian địa chỉ IPv4

- Theo lý thuyết
  - Có thể là 0.0.0.0 ~ 255.255.255.255
  - Một số địa chỉ đặc biệt
- Địa chỉ IP đặc biệt ([RFC1918](#))

Private address	10.0.0.0/8
	172.16.0.0/16 → 172.31.0.0/16
	192.168.0.0/24 → 192.168.255.0 /24
Loopback address	127.0.0.0 /8
Multicast address	224.0.0.0
	~239.255.255.255

- Địa chỉ liên kết nội bộ: 169.254.0.0/16

# Quản lý địa chỉ IP công cộng

- Internet Corporation for Assigned Names and Numbers (ICANN): quản lý toàn bộ tài nguyên địa chỉ IP
- Regional Internet Registries: quản lý địa chỉ IP theo vùng (châu Á-Thái Bình Dương, châu Âu và Trung Đông, châu Phi, Bắc Mỹ, Nam Mỹ)
- Cơ quan quản lý quốc gia
  - Việt Nam: VNNIC
- Nhà cung cấp dịch vụ (ISP)
- Cơ quan, tổ chức
- Ví dụ

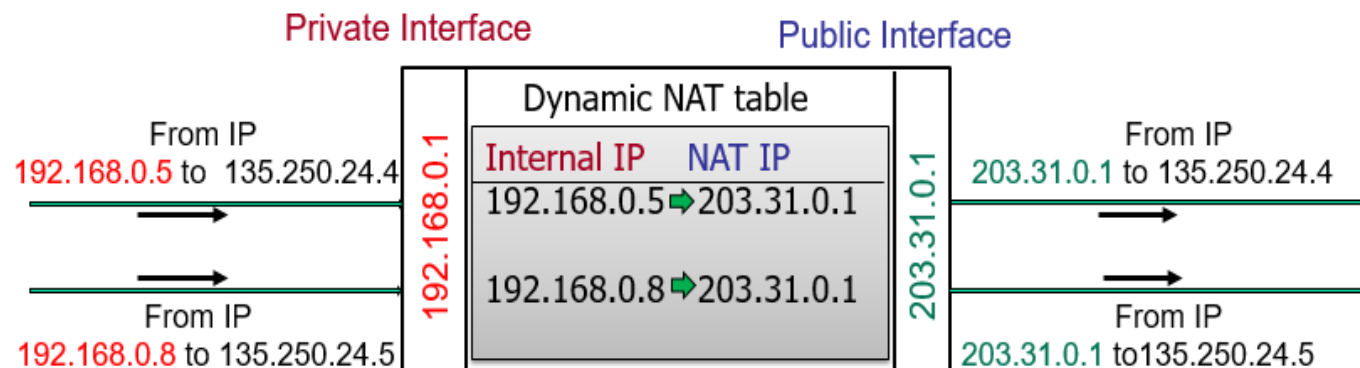
ICANN → APNIC → VNNIC → HUST

# Network Address Translation

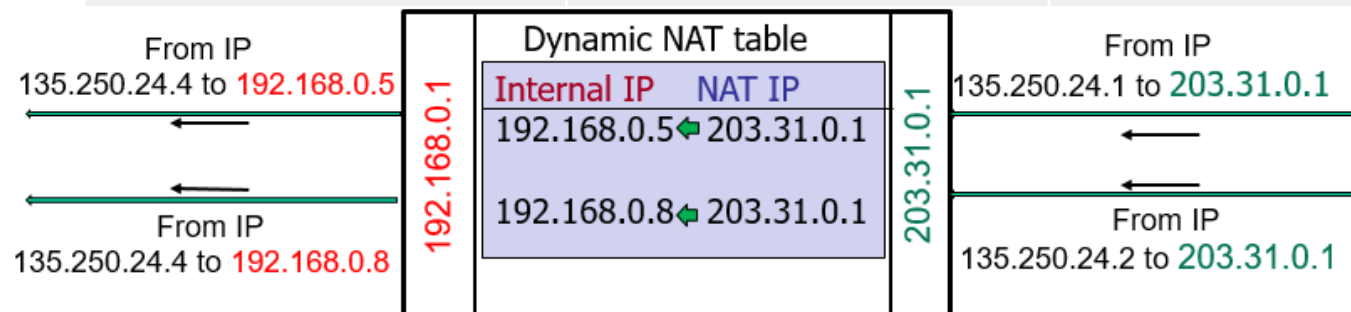
- NAT : Network Address Translation
  - Chuyển đổi địa chỉ trên gói tin từ IP cục bộ sang IP công cộng
  - Và ngược lại
- PAT : Port Address Translation
  - NAT with overloading sử dụng thêm số hiệu cổng ứng dụng trong quá trình chuyển đổi
- Lợi ích:
  - Tiết kiệm địa chỉ IP công cộng
  - Che giấu địa chỉ riêng
  - Giảm chi phí cấu hình khi thay đổi ISP
- Trên thực tế, có thể sử dụng NAT để chuyển đổi địa chỉ IP từ mạng LAN này sang mạng LAN khác

# Hoạt động của NAT

- Gói tin đi từ trong mạng ra ngoài

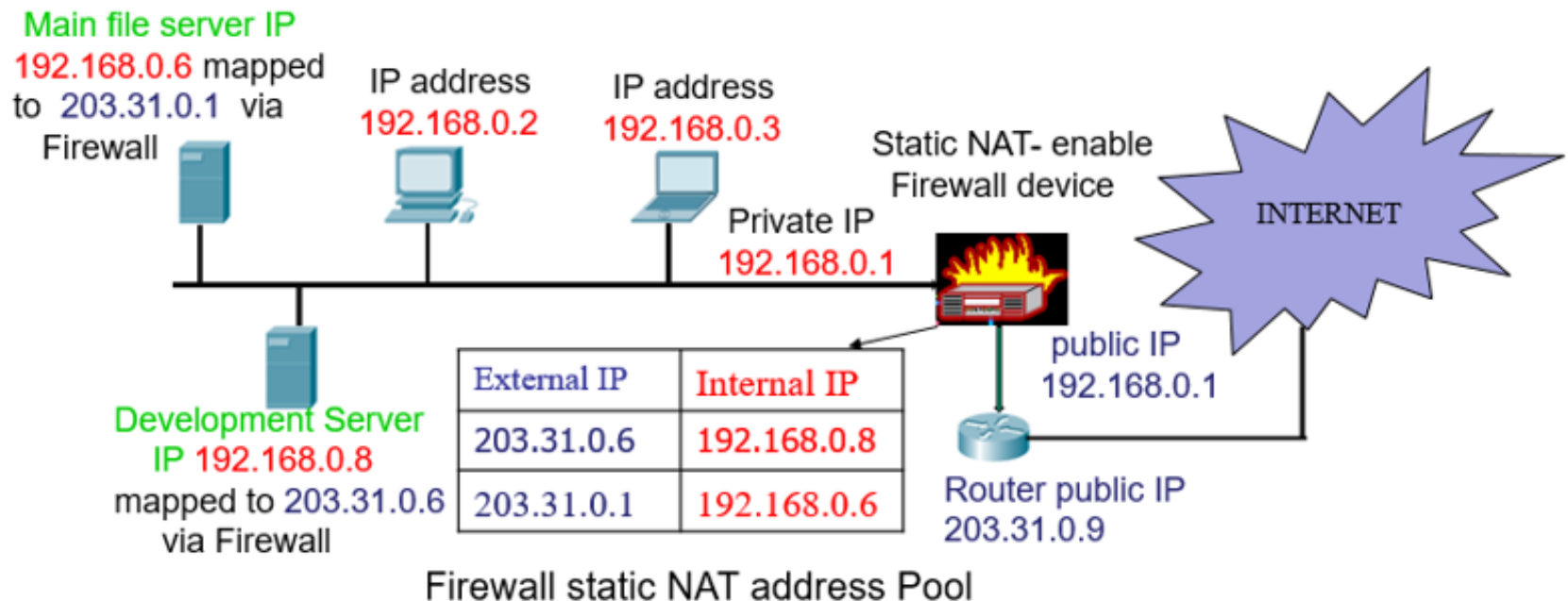


Private Addr	Public Addr	Destination
192.168.0.5	203.31.0.1	135.250.24.1
192.168.0.8	203.31.0.1	135.250.24.1

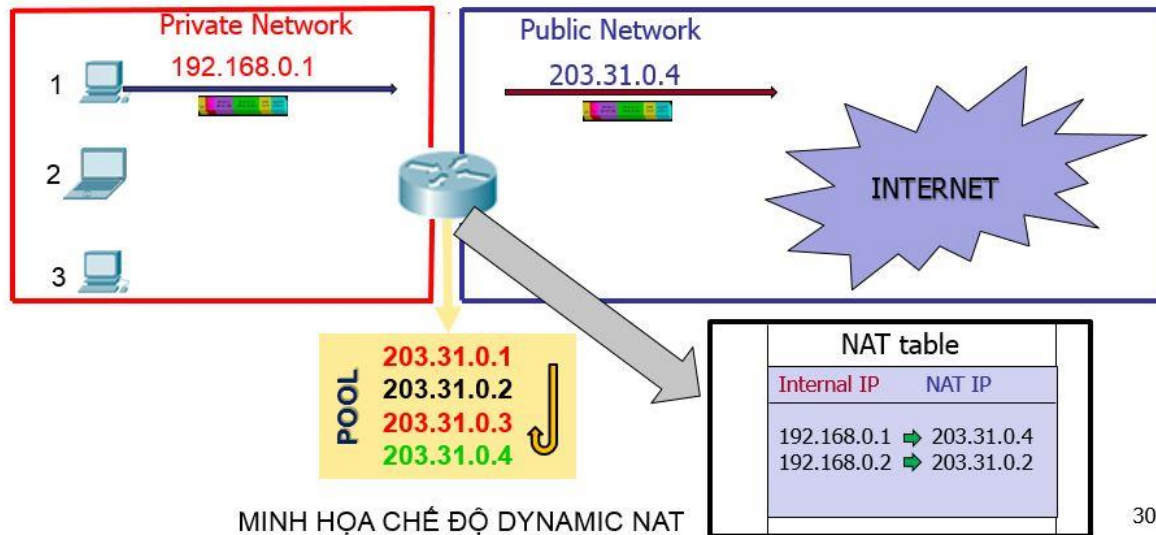


# Các chế độ hoạt động của NAT

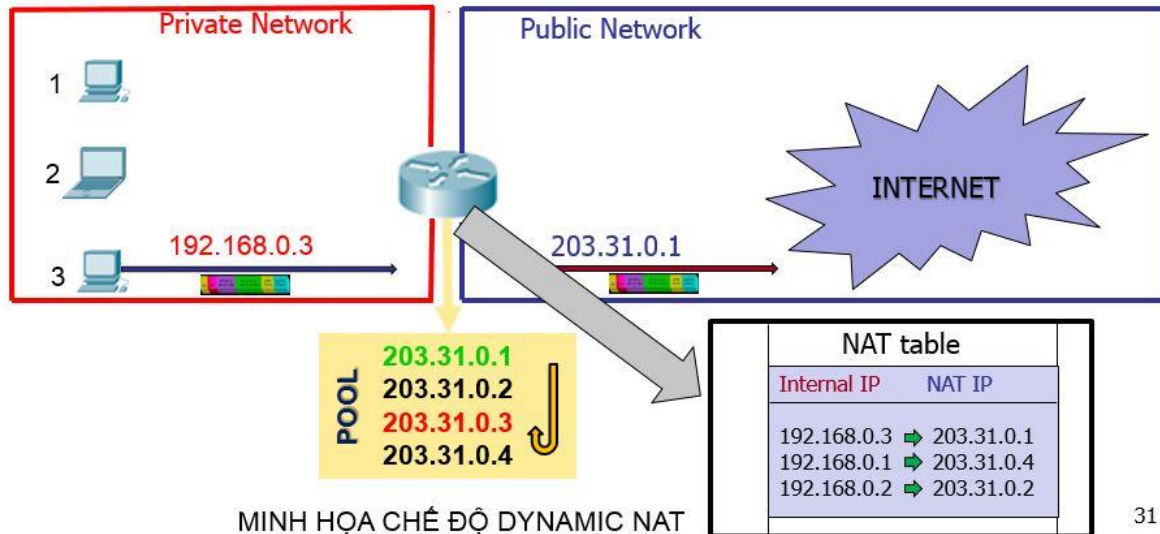
- Static NAT(NAT tĩnh): mỗi địa chỉ IP của mạng bên trong được ánh xạ tới một địa chỉ IP của mạng bên ngoài
  - Ánh xạ luôn được duy trì trong bảng NAT
  - Thường sử dụng cho các máy chủ cung cấp dịch vụ



# Dynamic NAT

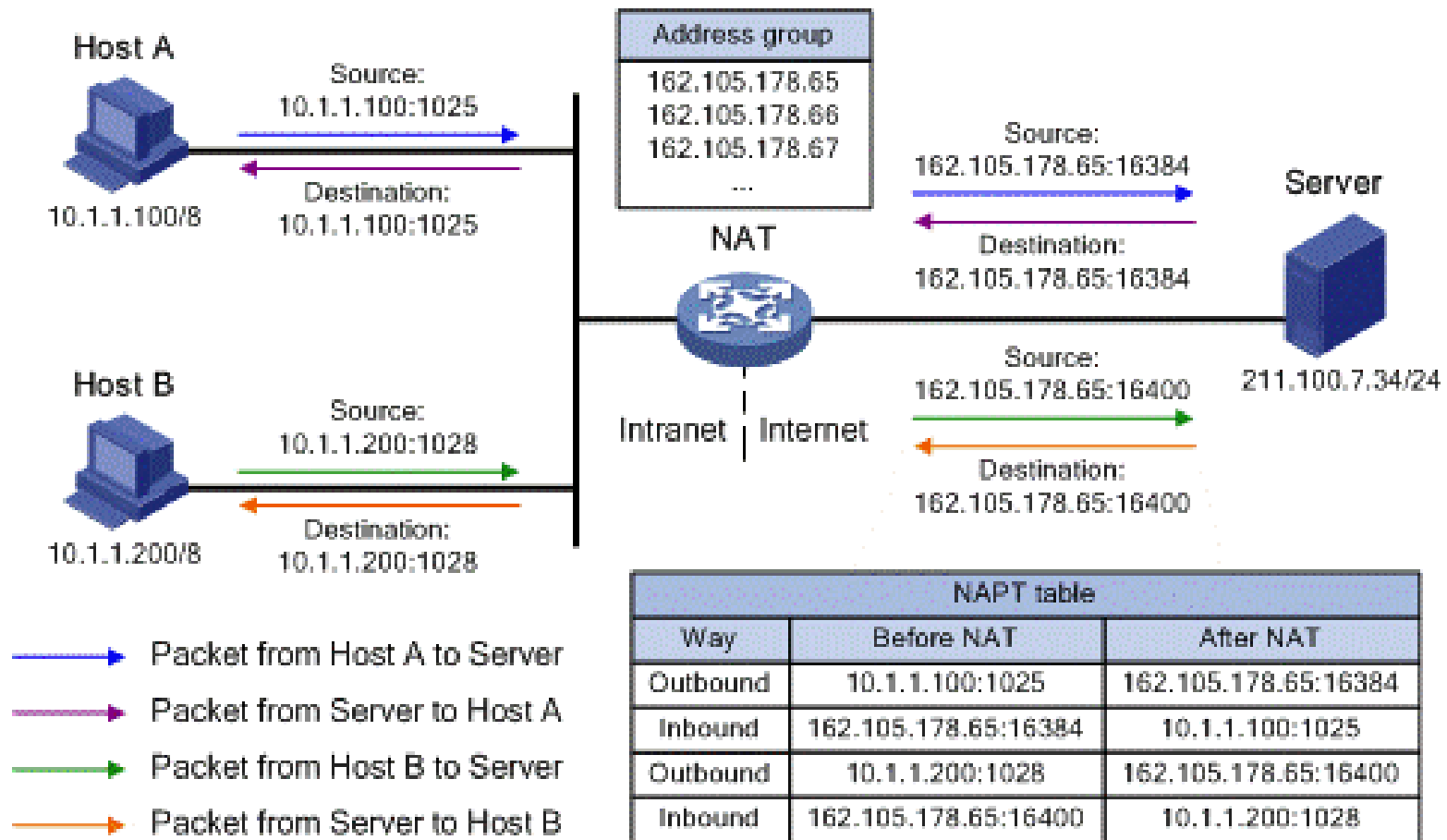


30

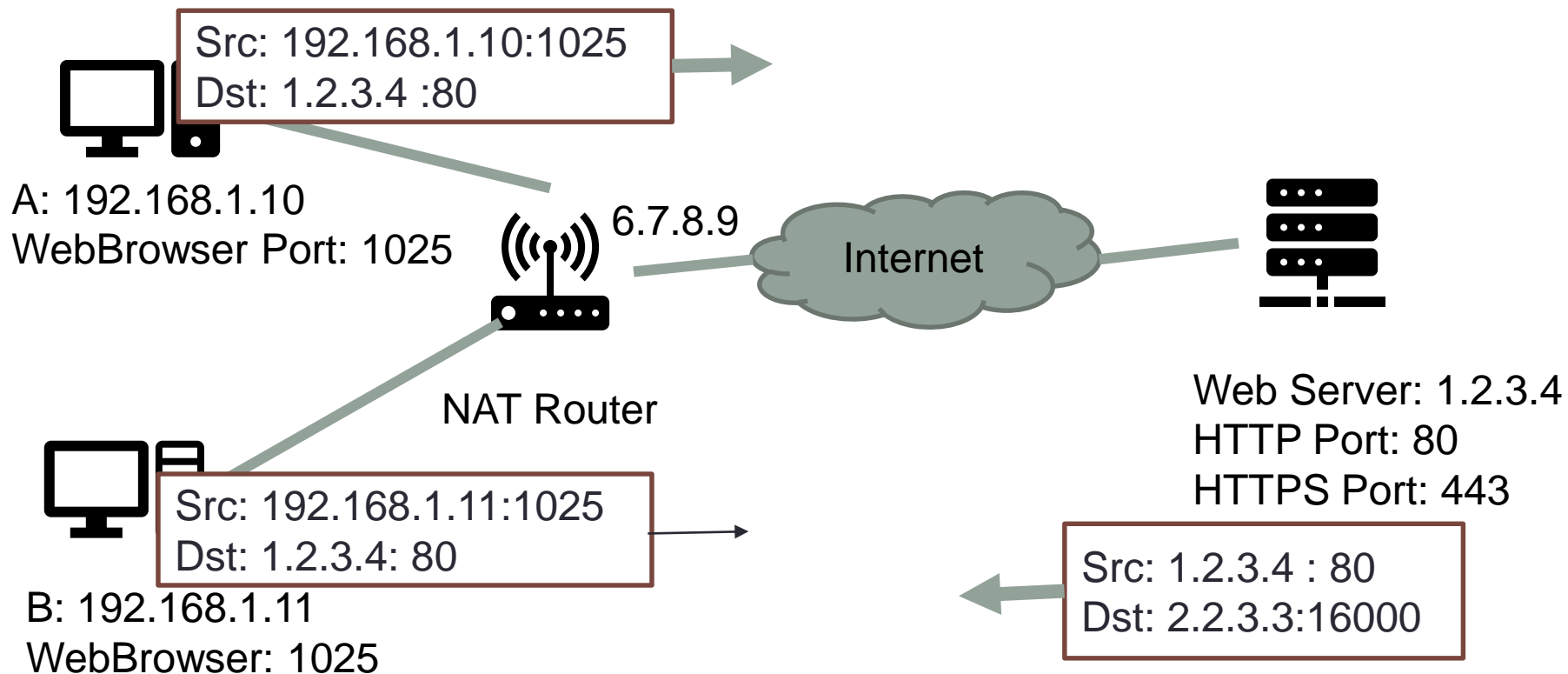


31

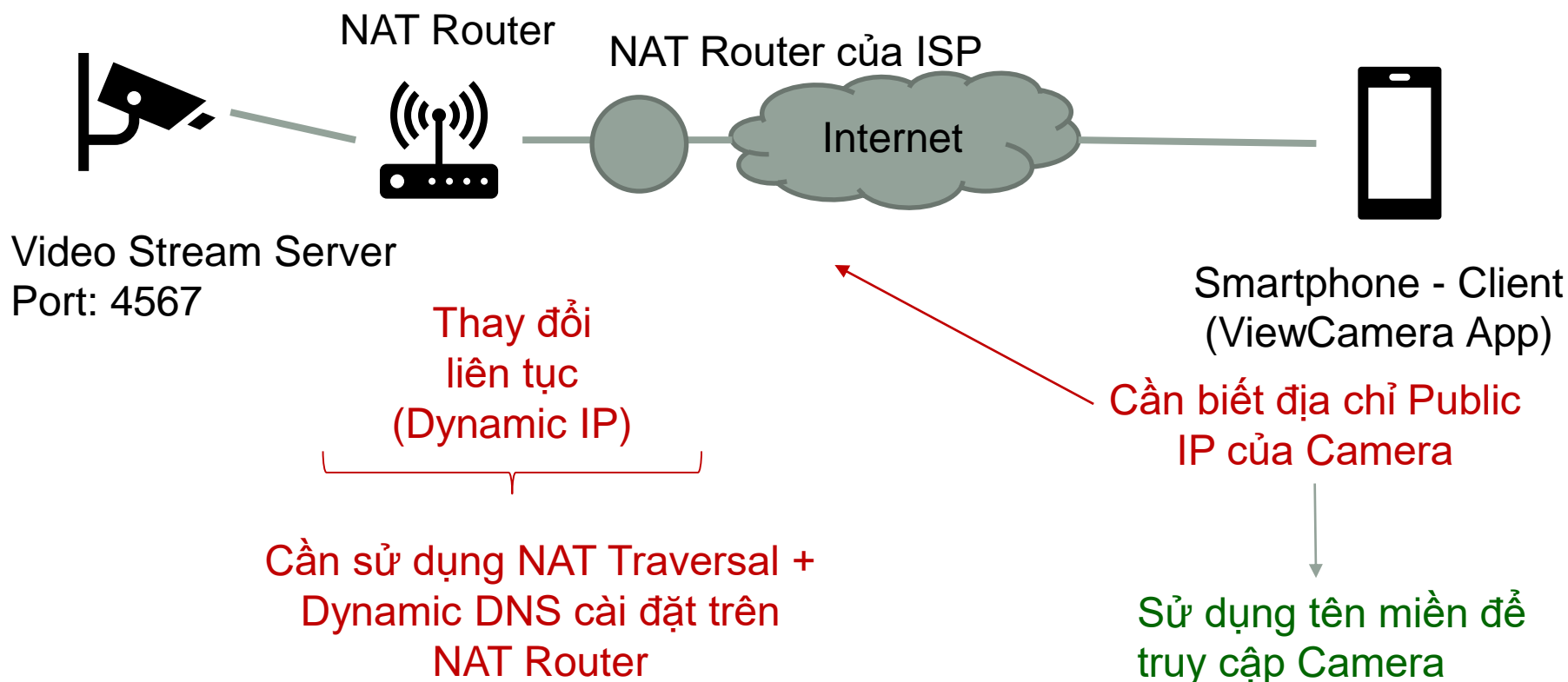
# PAT



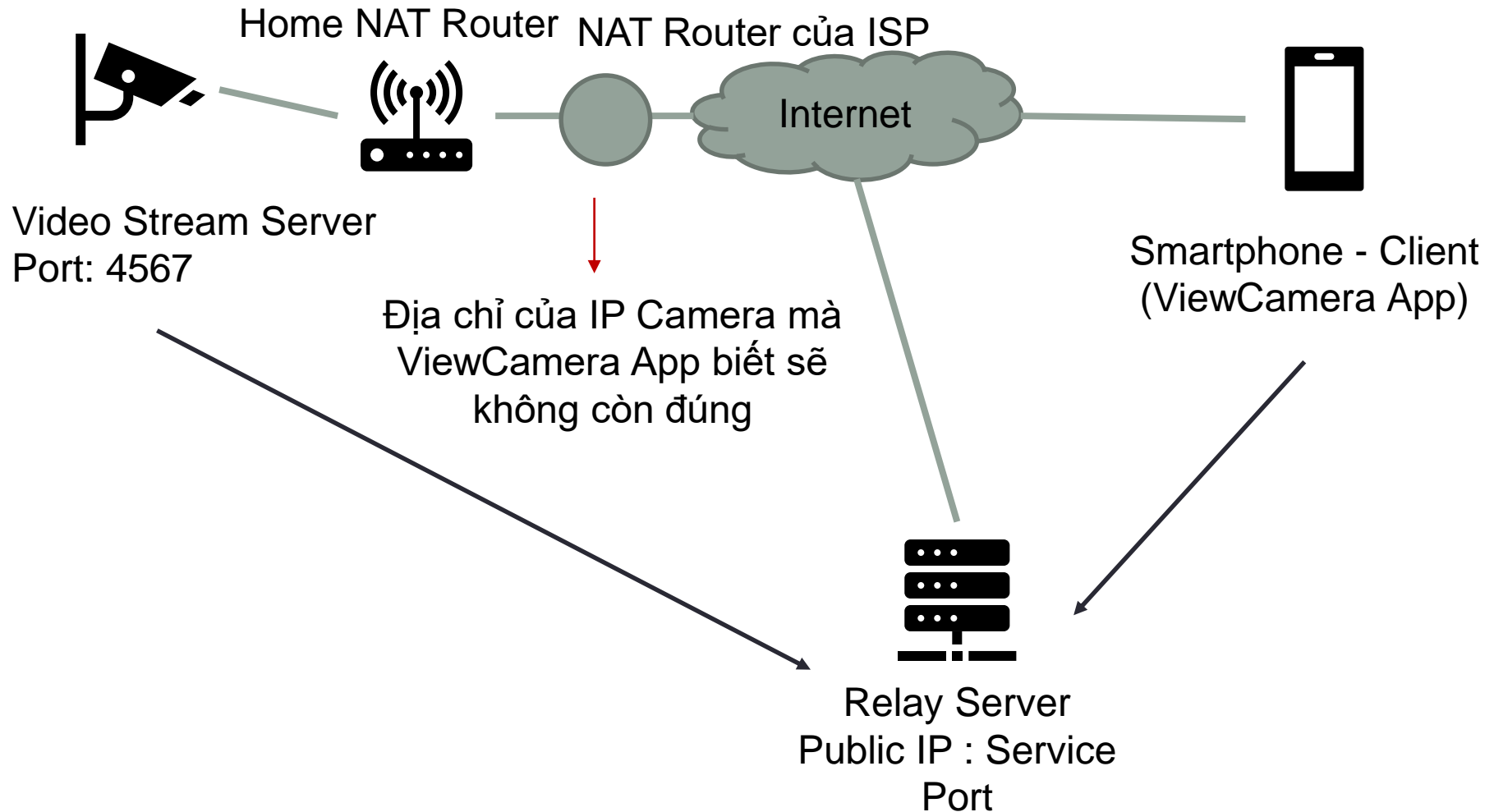




	Trước NAT	Sau NAT
Outbound	192.168.1.10 : 1025	6.7.8.9 : 16000
Inbound	2.2.3.3: 16000	192.168.1.10 : 1025
Outbound	192.168.1.11 : 1025	6.7.8.9 : 17000
Inbound	2.2.3.3 : 17000	192.168.1.11 : 1025



Dynamic DNS: phân giải tên miền động → Tự động cập nhật địa chỉ IP trong anh xạ phân giải tên miền



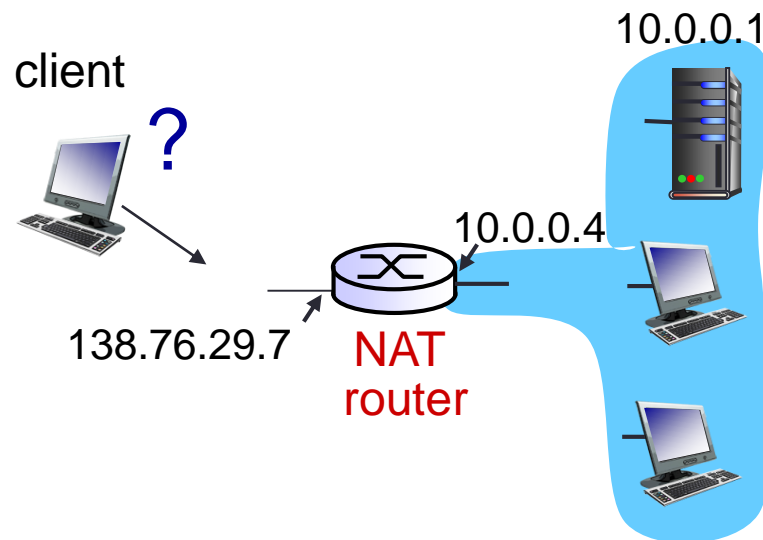
Giải pháp: Sử dụng Relay Server

# Triển khai

- Đăng ký tên miền: cam1.tungbt.dyndns.com
- Home Router:
  - Cài đặt Dynamic DNS Client: báo cáo sự thay đổi địa chỉ IP Public của Home Router cho Dynamic DNS Server → ánh xạ tên miền cam1.tungbt.dyndns.com sang địa chỉ IP sẽ được cập nhật liên tục
  - Hỗ trợ UPnP
- IP Camera: sử dụng UPnP đăng ký một cổng dịch vụ trên Home Router (Ví dụ: 10000)
  - Khi Home Router nhận được 1 gói tin tới cổng dịch vụ 10000 thì nó tự động chuyển tiếp tới IP Camera
- ViewCamera App trên smartphone đã biết địa chỉ của IP Camera:
  - Tên miền: cam1.tungbt.dyndns.com
  - Số hiệu cổng ứng dụng: 10000

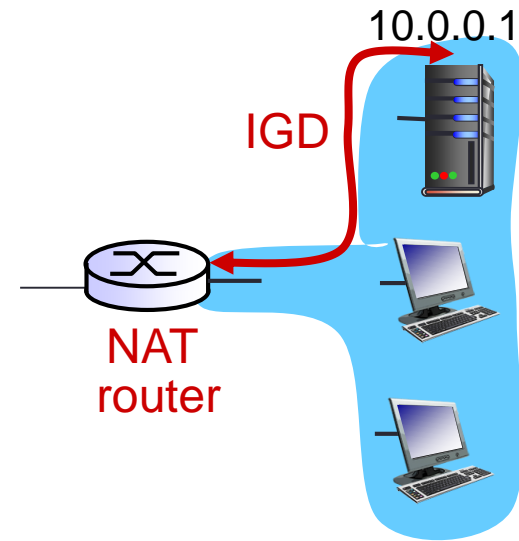
# Vấn đề NAT traversal

- Client muốn kết nối tới server có địa chỉ 10.0.0.1
  - Địa chỉ của server là địa chỉ LAN (client không thể sử dụng địa chỉ này như là địa chỉ đích)
  - Địa chỉ công cộng: 138.76.29.7
- **Giải pháp 1:** cấu hình chuyển tiếp yêu cầu thiết lập kết nối tới cổng trên server. Ví dụ:
  - Client gửi yêu cầu kết nối tới địa chỉ (123.76.29.7:2500)
  - Trên router, các gói tin gửi tới địa chỉ (123.76.29.7:2500) luôn được chuyển tiếp tới địa chỉ (10.0.0.1:5000)



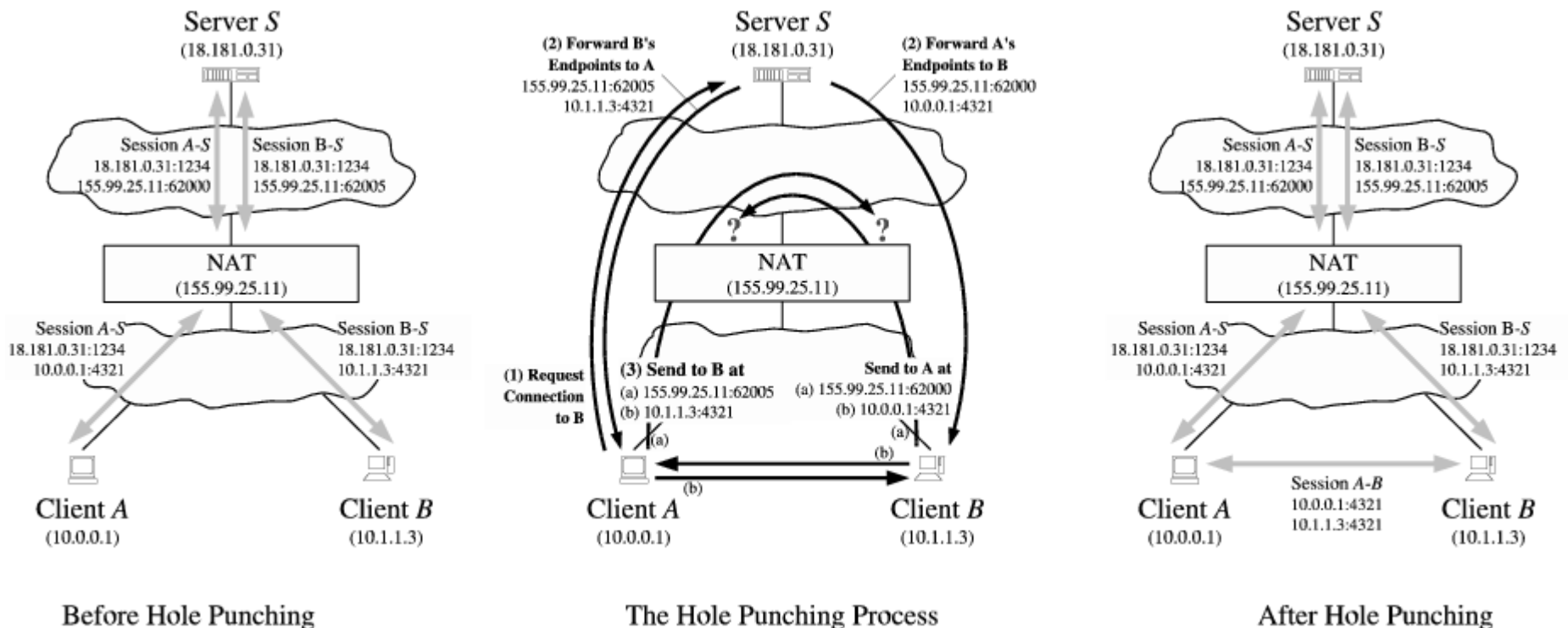
# Vấn đề NAT traversal

- *Giải pháp 2:* Sử dụng giao thức Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Cho phép các máy trạm được NAT:
  - ❖ Học địa chỉ IP công cộng (138.76.29.7)
  - ❖ Yêu cầu thêm và xóa ánh xạ trong bảng NAT



# Vấn đề NAT traversal

- Giải pháp 3:* NAT hole punching. VD: giao thức STUN



### 3. THƯ VIỆN LẬP TRÌNH

---

- Eclipse & Java Socket API (for all platforms)
- Eclipse & GCC (for Ubuntu Linux)
- MS Visual Studio & WinSock (for Windows)
- ??? (for MAC OS)