TẦNG VẬN CHUYỂN (Transport Layer)

Trình bày: Bùi Minh Quân

bmquan@ctu.edu.vn

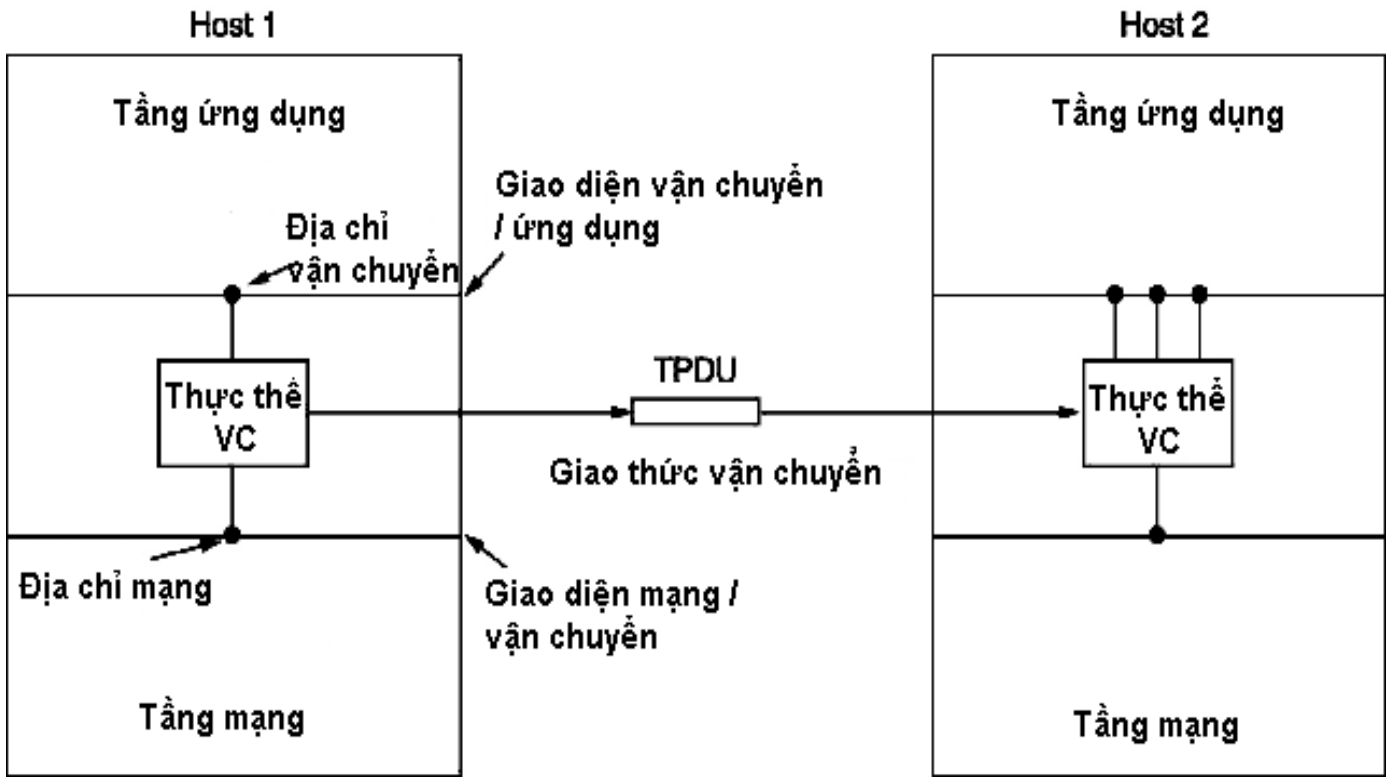
Nội dung

- Nhiệm vụ của tầng vận chuyển
- Dịch vụ tầng vận chuyển cung cấp tầng ứng dụng
- Cơ chế thiết lập và xóa nối kết của tầng vận chuyển
- TCP và UDP trong mạng Internet

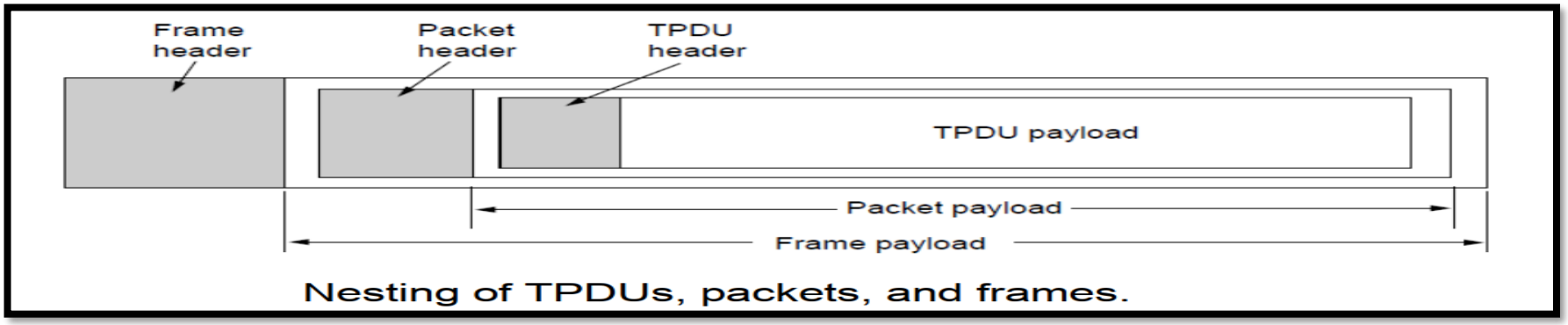
Nhiệm vụ của tầng vận chuyển

- Tầng mạng đảm bảo truyền tải kiểu Host – to – host.
- Tầng vận chuyển đảm bảo truyền tải kiểu End point –to – End point
- End point là các chương trình ứng dụng
- Cấp dịch vụ vận chuyển gói tin hiệu quả, tin cậy và tiết kiệm chi phí cho người dùng

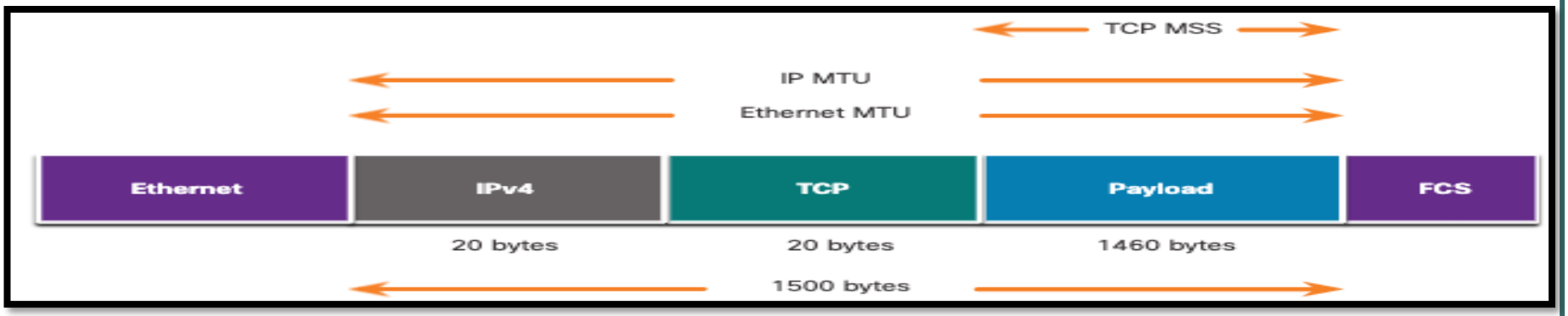
Vị trí của tầng vận chuyển

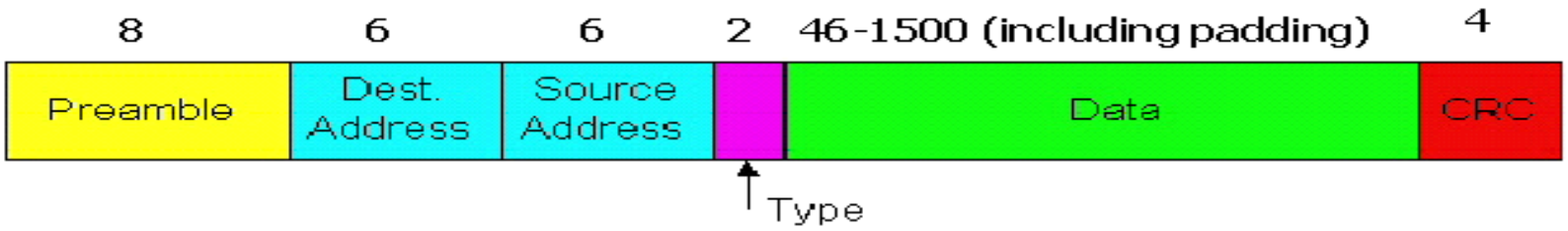


TPDU



Nesting of TPDUs, packets, and frames.





Dịch vụ cung cấp bởi tầng vận chuyển

- Cấp dịch vụ vận chuyển gói tin hiệu quả, tin cậy và tiết kiệm chi phí cho người dùng
  - Hai kiểu dịch vụ
    - Có nối kết : (hoạt động giống như có nối kết ở tầng mạng)
      - Thiết lập nối kết,
      - Truyền dữ liệu
      - Hủy nối kết
    - Không nối kết (hoạt động giống như có không nối kết ở tầng mạng)
  - Cải thiện chất lượng dịch vụ: các hàm dịch vụ cơ sở để triệu gọi các dịch vụ vận chuyển và các hàm này là đơn giản, duy nhất và độc lập với các hàm cơ sở ở tầng mạng

Các hàm dịch vụ cơ sở - Có nối kết

Hàm	Gói tin gởi đi	Ý nghĩa
LISTEN	Không có	Nghẽn cho đến khi tiến trình nào đó nối kết tới
CONNECT	Yêu cầu kết nối (Connection Request)	Chủ động yêu cầu thiết lập nối kết đến tiến trình khác
SEND	Dữ liệu (Data)	Gởi thông tin đi
RECEIVE	Không có	Nghẽn cho đến khi một gói tin đến và nhận nó
DISCONNECT	Yêu cầu hủy kết nối (Disconnection Request)	Muốn hủy kết nối với bên đối tác

Các hàm dịch vụ cơ sở - Không nối kết

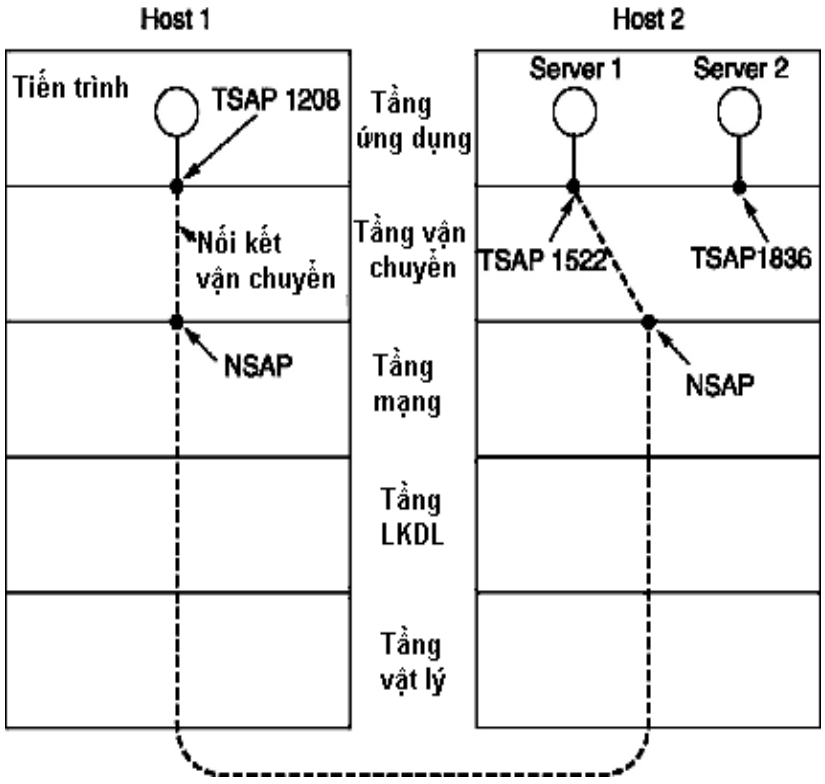
Hàm	Gói tin gởi đi	Ý nghĩa
SEND	Dữ liệu (Data)	Gởi thông tin đi
RECEIVE	Không có	Nghẽn cho đến khi một gói tin đến và nhận nó

Các yếu tố cấu thành giao thức vận chuyển

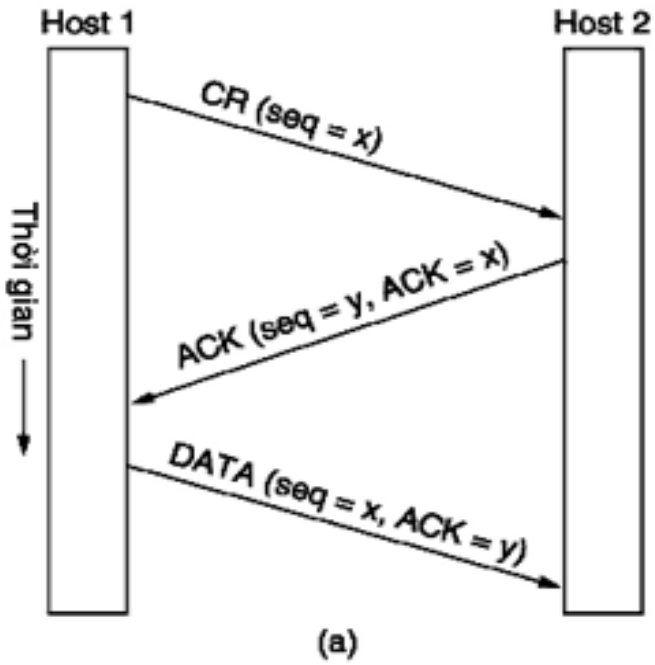
- Điều khiển lỗi, đánh số thứ tự gói tin và điều khiển luồng dữ liệu.
- Môi trường giao tiếp qua một tập các mạng trung gian
- Những vấn đề cần quan tâm:
  - Định địa chỉ các tiến trình trên các host
  - Xử lý những trường hợp mất gói tin, gói tin đi chậm dẫn đến mãn kỳ và gởi thêm một gói tin bị trùng lắp,
  - Đồng bộ hóa hai tiến trình đang trao đổi dữ liệu khi mà chúng đang ở rất xa nhau

Định địa chỉ

- Địa chỉ tiến trình là TSAP (Transport Service Access Point).
  - Mạng Internet là dùng số hiệu cổng (**port**),
  - Mạng ATM là AAL-SAP.
- Tầng mạng được gọi là NSAP: Internet là địa chỉ IP



Thiết lập nối kết

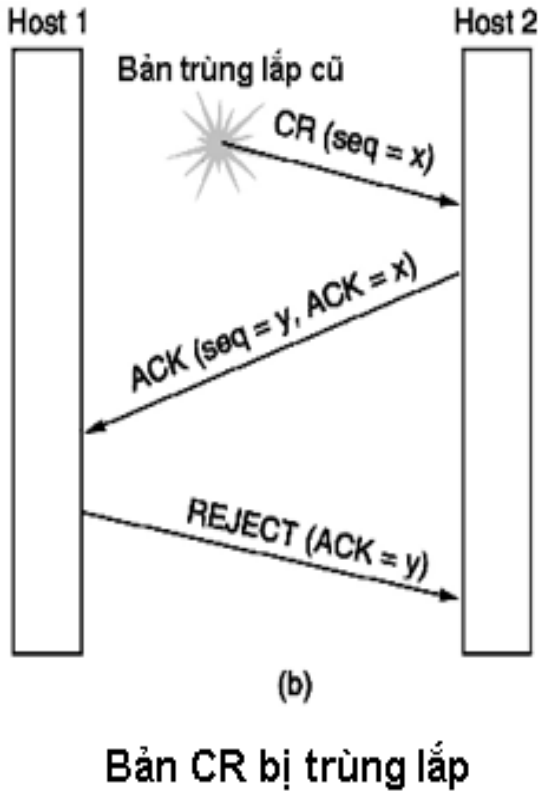


Three-way hand-shake
Hoạt động bình thường.

- Host1: chọn một số thứ tự x đính kèm số đó vào trong TPDU CR (CR(seq=x)).
- Host2: báo nhận ack (ack(seq=y, ack=x)) thông báo số khởi đầu là **y**.
- Host1: báo đã biết số khởi đầu của host2 là y bằng TPDU data(seq=x,ack=**y**)).

Thiết lập nối kết

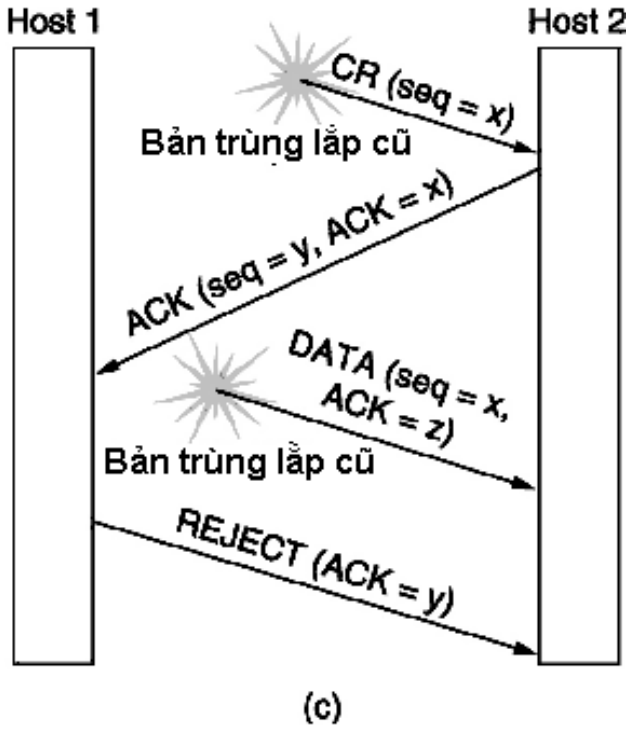
- **Tình huống TPDU CR bị trùng lắp**
  - Khi TPDU CR thứ hai đến host 2, host 2 liền trả lời ACK vì tưởng rằng host 1 muốn thiết lập nối kết khác.
  - **Khi host 1 từ chối** cố gắng thiết lập nối kết của host 2, host 2 hiểu rằng nó đã bị lừa bởi **CR bị trùng lắp** và sẽ từ bỏ nối kết đó.



Bản CR bị trùng lắp

Thiết lập nối kết

- **Khi TPDU CR và ACK host1 bị trùng lắp**
  - Như trường hợp (b) Khi TPDU CR (trễ) thứ hai đến host 2, host 2 liền trả lời với số thự khởi đầu là **y (giả sử trong trả lời CR trước khởi đầu là Z).**
  - Báo nhận ở chiều thứ 3 của host 1 lại trễ. Khi host 1 nhận được ACK(seg=y, ACK=x)
  - Host 1: nhận biết thông báo **Data(seg=x, ACK=Z)** bị trễ, do đó nó từ bỏ nối kết.

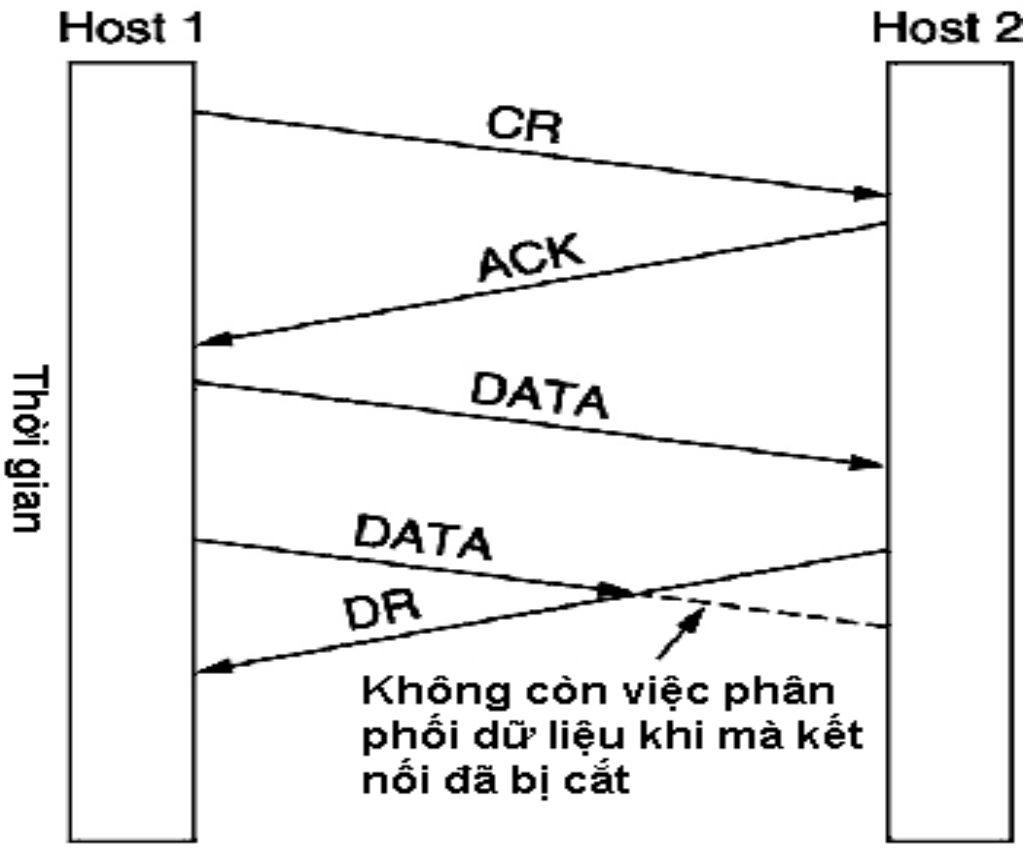


Cả CR và ACK đều bị trùng lắp

Giải phóng nối kết

- Hai kiểu giải phóng nối kết:
  - Kiểu dị bộ hoạt động như sau: khi một bên cắt nối kết, kết nối sẽ bị hủy bỏ (giống như trong hệ thống điện thoại).
  - Kiểu đồng bộ làm việc theo phương thức ngược lại: khi cả hai đồng ý hủy bỏ nối kết, nối kết mới thực sự được hủy

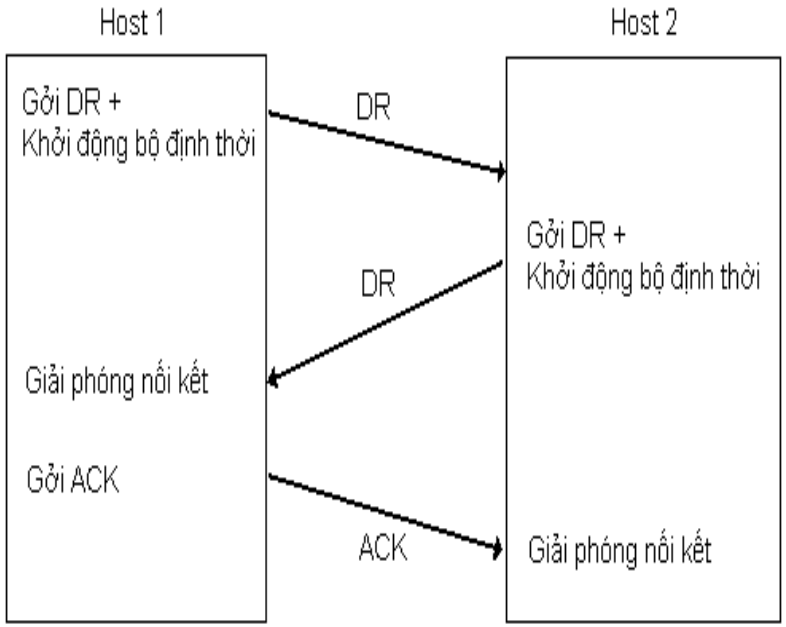
Giải phóng nối kết dị bộ



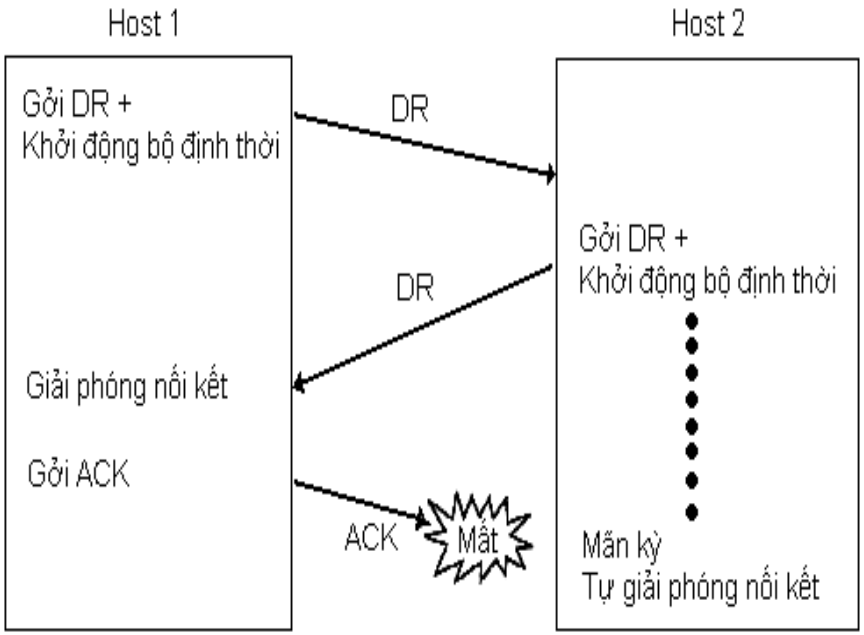
Giải phóng nối kết đồng bộ

- Một nút phải tiếp tục nhận dữ liệu sau khi đã gởi đi yêu cầu giải phóng nối kết (DISCONNECT REQUEST – DR), cho đến khi nhận được chấp thuận hủy bỏ nối kết của bên đối tác đó
- Sử dụng phương pháp **hủy** nối kết ba chiều cùng với bộ định thời

Giải phóng nối kết đồng bộ

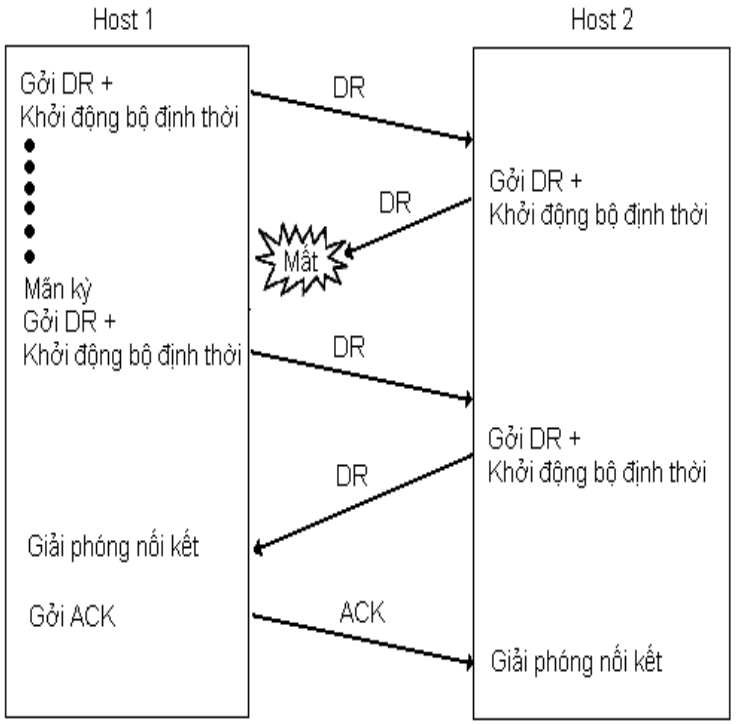


Bình thường

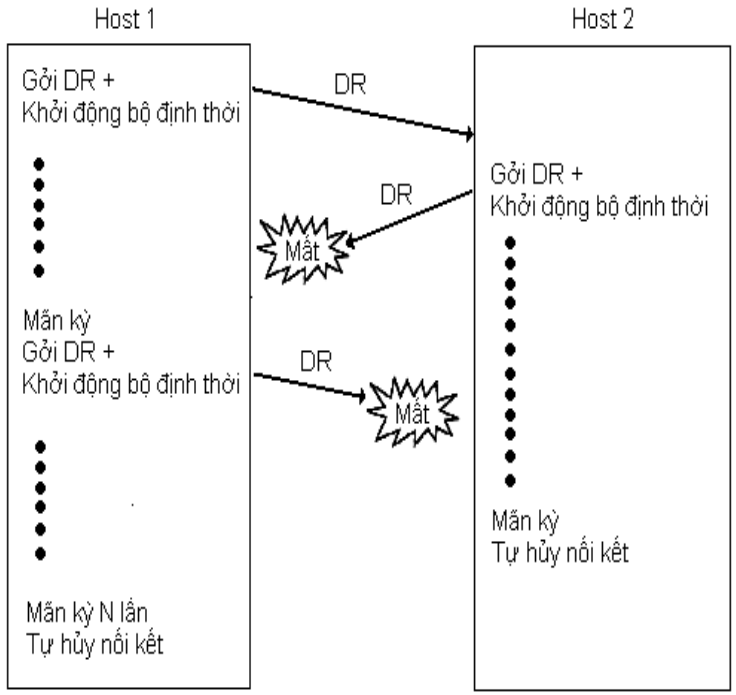


Khung ACK cuối cùng bị mất

Giải phóng nối kết đồng bộ



Trả lời bị mất



Trả lời mất và các gói tin DR theo sau cũng bị mất

Điều khiển thông lượng

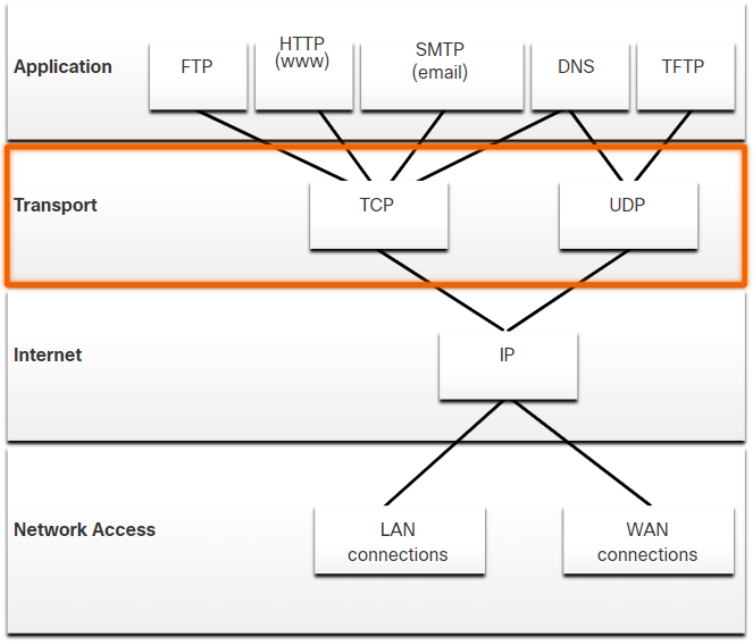
- Sử dụng giao thức cửa sổ trượt với kích thước cửa sổ của bên gởi và bên nhận là khác nhau
- Cần phải có sơ đồ cung cấp buffer động:
  - Trước tiên, bên gởi phải gởi đến bên nhận một yêu cầu dành riêng số lượng buffer để chứa các gói bên gởi gởi đến.
  - Bên nhận cũng phải trả lời cho bên gởi số lượng buffer tối đa mà nó có thể cung cấp.
  - Mỗi khi báo nhận ACK cho một gói tin có số thứ tự SEQ_NUM, bên nhận cũng phải gởi kèm theo thông báo cho bên gởi biết là lượng buffer còn lại là bao nhiêu để bên gởi không làm ngập bên nhận

Điều khiển thông lượng

	A	Message	B	Comments
1	→	< request 8 buffers>	→	A wants 8 buffers
2	←	<ack = 15, buf = 4>	←	B grants messages 0-3 only
3	→	<seq = 0, data = m0>	→	A has 3 buffers left now
4	→	<seq = 1, data = m1>	→	A has 2 buffers left now
5	→	<seq = 2, data = m2>	• • •	Message lost but A thinks it has 1 left
6	←	<ack = 1, buf = 3>	←	B acknowledges 0 and 1, permits 2-4
7	→	<seq = 3, data = m3>	→	A has 1 buffer left
8	→	<seq = 4, data = m4>	→	A has 0 buffers left, and must stop
9	→	<seq = 2, data = m2>	→	A times out and retransmits
10	←	<ack = 4, buf = 0>	←	Everything acknowledged, but A still blocked
11	←	<ack = 4, buf = 1>	←	A may now send 5
12	←	<ack = 4, buf = 2>	←	B found a new buffer somewhere
13	→	<seq = 5, data = m5>	→	A has 1 buffer left
14	→	<seq = 6, data = m6>	→	A is now blocked again
15	←	<ack = 6, buf = 0>	←	A is still blocked
16	• • •	<ack = 6, buf = 4>	←	Potential deadlock

Tầng vận chuyển trong mạng Internet

- IP không chỉ định cách thức phân phối hoặc vận chuyển các gói diễn ra
- Các giao thức lớp vận chuyển chỉ định cách truyền tin nhắn giữa các máy chủ và chịu trách nhiệm quản lý các yêu cầu về độ tin cậy của cuộc trò chuyện.
- Hỗ trợ hai phương thức hoạt động TCP và UDP.



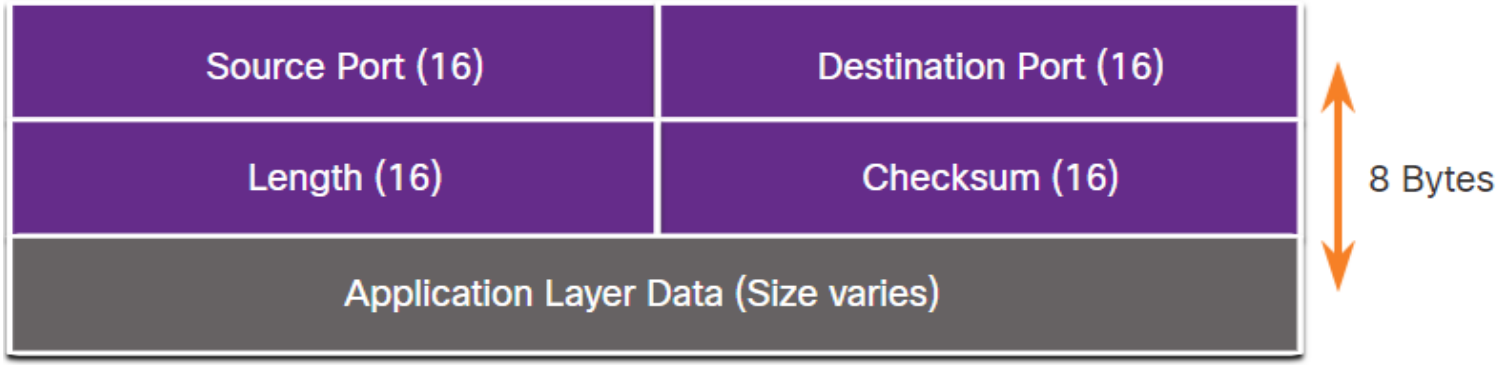
Tầng vận chuyển trong mạng Internet

- Nhiệm vụ
  - Đảm bảo việc phân phối thông điệp qua mạng.
  - Phân phối các thông điệp theo thứ tự mà chúng được gởi.
  - Không làm trùng lắp thông điệp.
  - Hỗ trợ những thông điệp có kích thước lớn.
  - Hỗ trợ cơ chế đồng bộ hóa.
  - Hỗ trợ việc liên lạc của nhiều tiến trình trên mỗi host

Giao thức UDP (User Datagram Protocol)

- UDP là dịch vụ truyền dữ liệu dạng không nối kết, không có thiết lập nối kết giữa hai bên truyền nhận.
- Gói tin UDP (segment) có thể xuất hiện tại nút đích bất kỳ lúc nào. Các segment UDP tự thân chứa mọi thông tin cần thiết để có thể tự đi đến đích.
- Không đảm bảo tính tin cậy khi truyền dữ liệu (không báo nhận) và không có cơ chế gửi lại gói tin bị mất (không biết gói tin có bị mất mát trên đường đi hay không)
- Không đảm bảo gói tin đến có thứ tự, không có cơ chế điều khiển luồng (do đó bên gửi sẽ làm ngập bên nhận).

UDP Header



- Checksum: là phần kiểm tra lỗi tổng hợp trên phần header, phần dữ liệu và cả phần header ảo.
- Phần header ảo chứa 3 trường trong IP header: địa chỉ IP nguồn, địa chỉ IP đích, và trường chiều dài của UDP.

Giao thức TCP (Transmission Control Protocol)

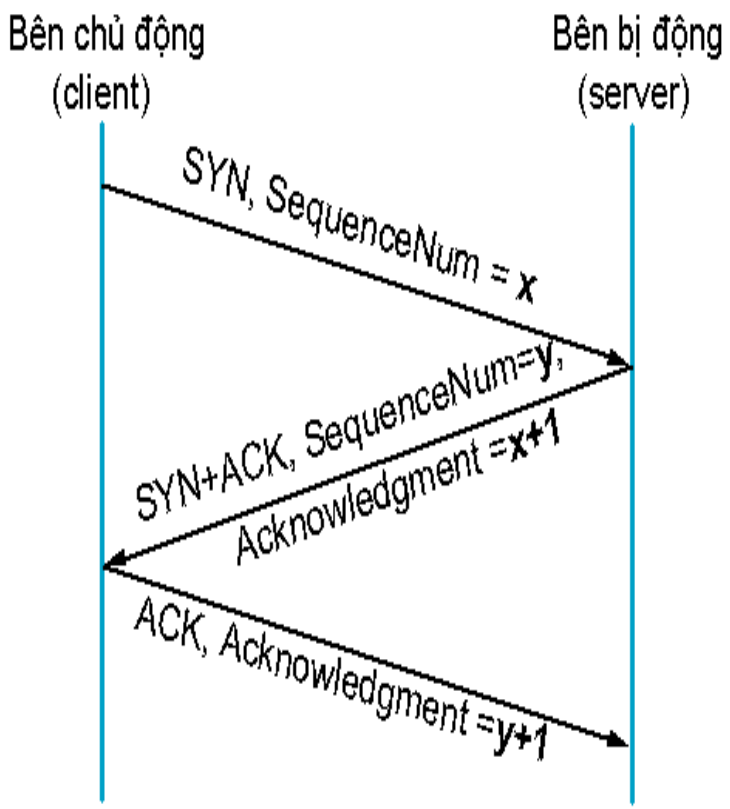
- TCP là giao thức cung cấp dịch vụ vận chuyển tin cậy và kiểm soát luồng
- Đánh số gói tin và có cơ chế báo nhận, truyền lại gói tin thất lạc
- TCP là giao thức truyền song công, hỗ trợ cơ chế đa hợp.

Giao thức TCP (Transmission Control Protocol)

Bước 1: Máy khách (client) khởi tạo yêu cầu phiên giao tiếp với máy chủ (server) Với số thứ tự khởi đầu **x**. (Flags=SYN, SequenceNum =x).

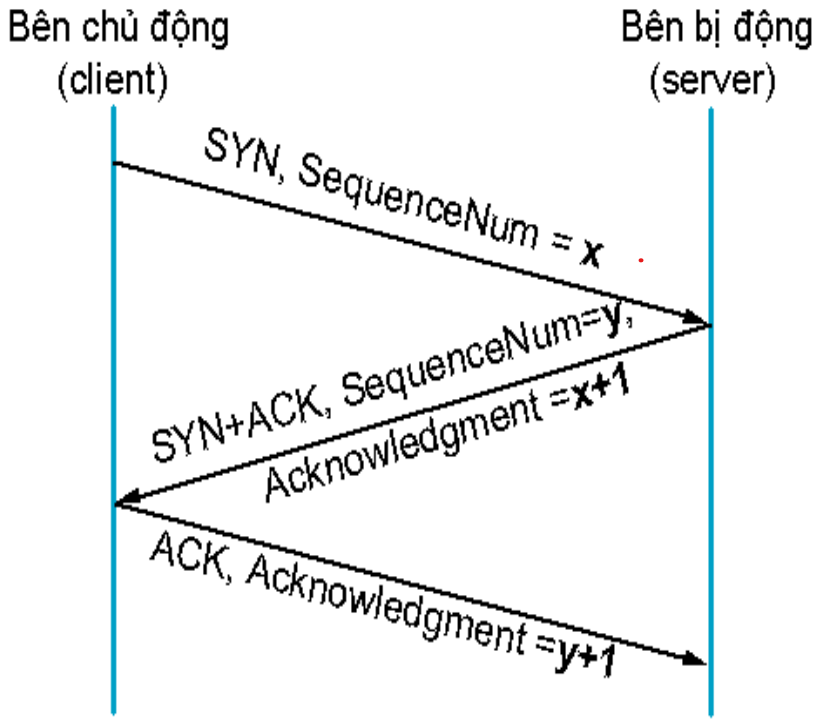
Bước 2: server trả lời cho client bằng một segment, báo nhận rằng nó sẵn sàng nhận các byte dữ liệu bắt đầu từ số thứ tự **x+1. (Flags = ACK, Ack = x+1) và cho biết số thứ tự khởi đầu (Flags = SYN, SequenceNum =y).**

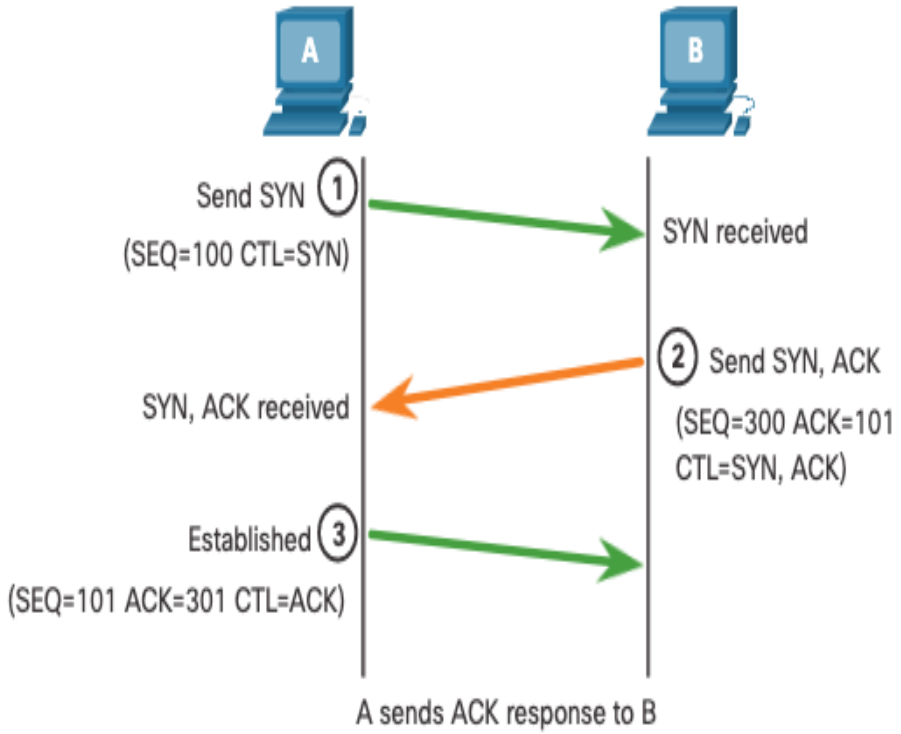
Bước 3: Client báo với máy chủ. Client biết số thứ tự của Server là y **(Flags = ACK, Ack = y+1).**



Bắt tay trong TCP

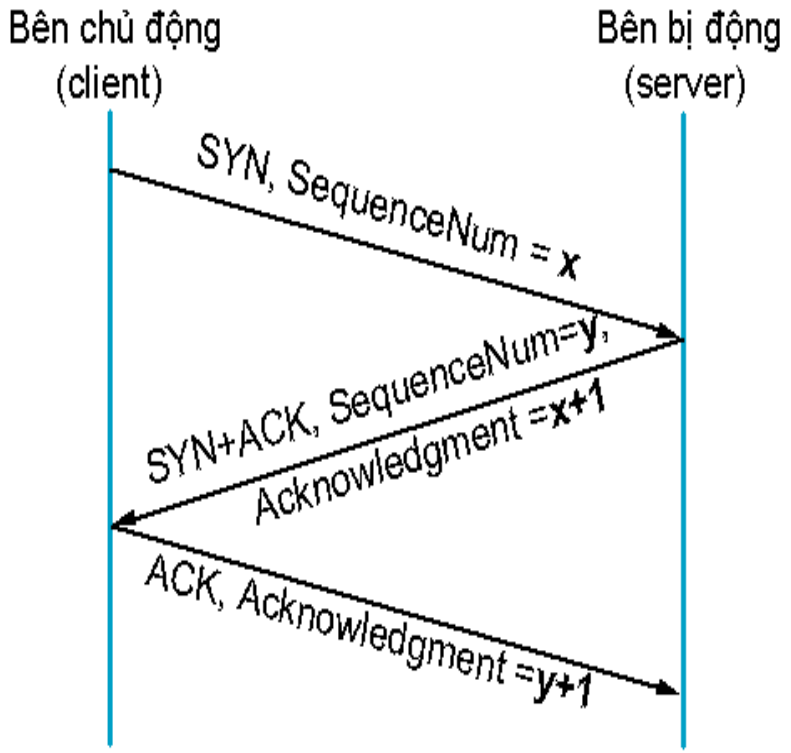
Giao thức TCP (Transmission Control Protocol)



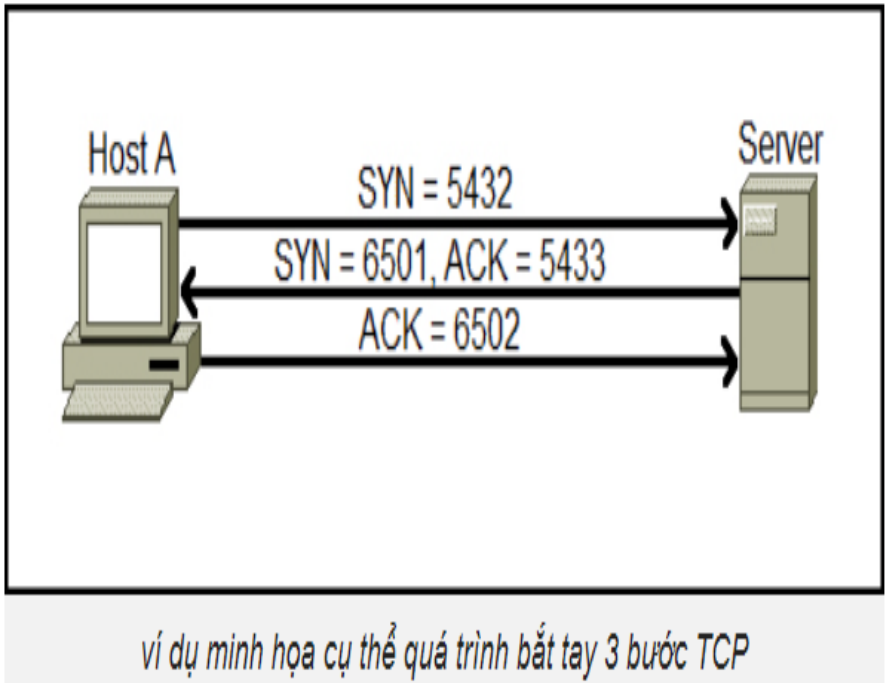


Bắt tay trong TCP

Giao thức TCP (Transmission Control Protocol)



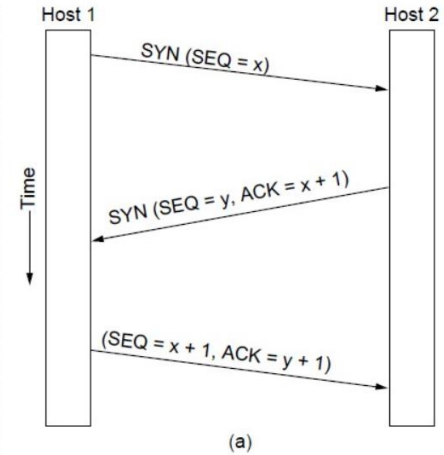
Bắt tay trong TCP

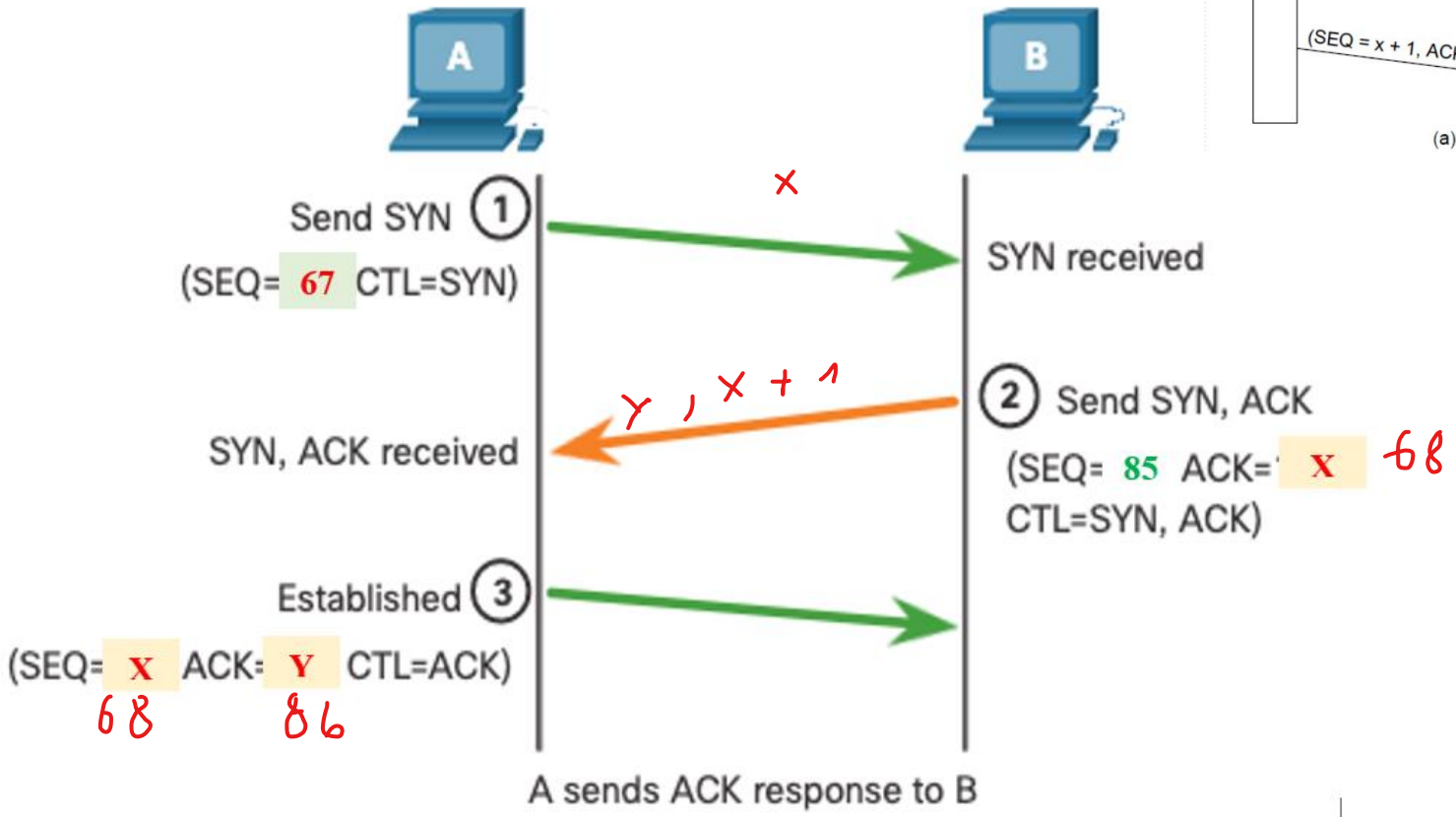


ví dụ minh họa cụ thể quá trình bắt tay 3 bước TCP

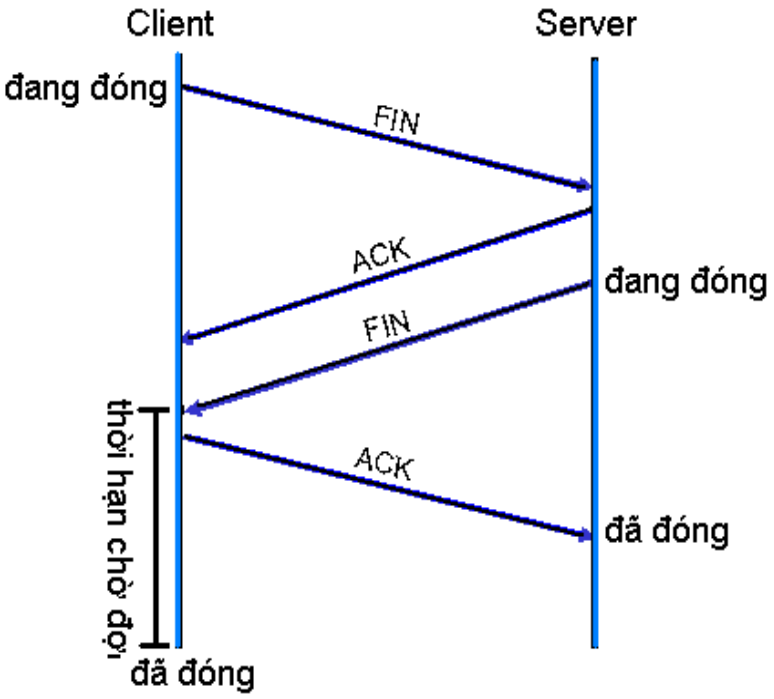
https://vienthongxanh.vn/quy-trinh-bat-tay-3-buoc-tcp/

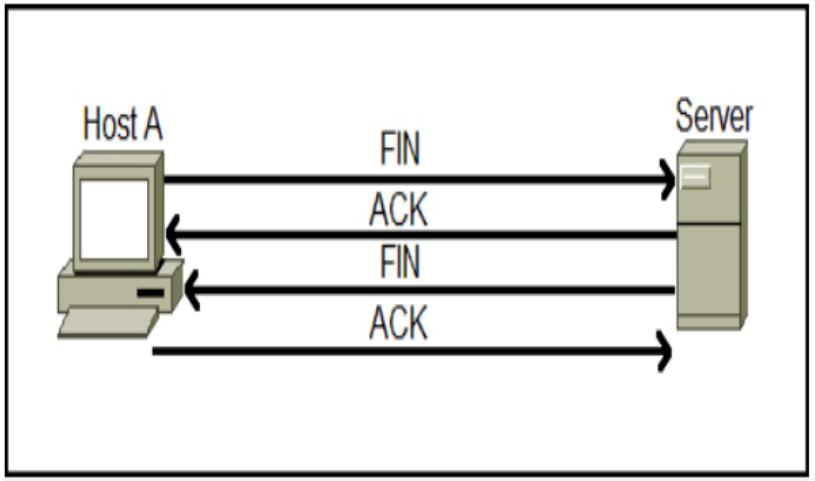
Bài tập: xác định X và Y?





Hủy bắt tay trong TCP





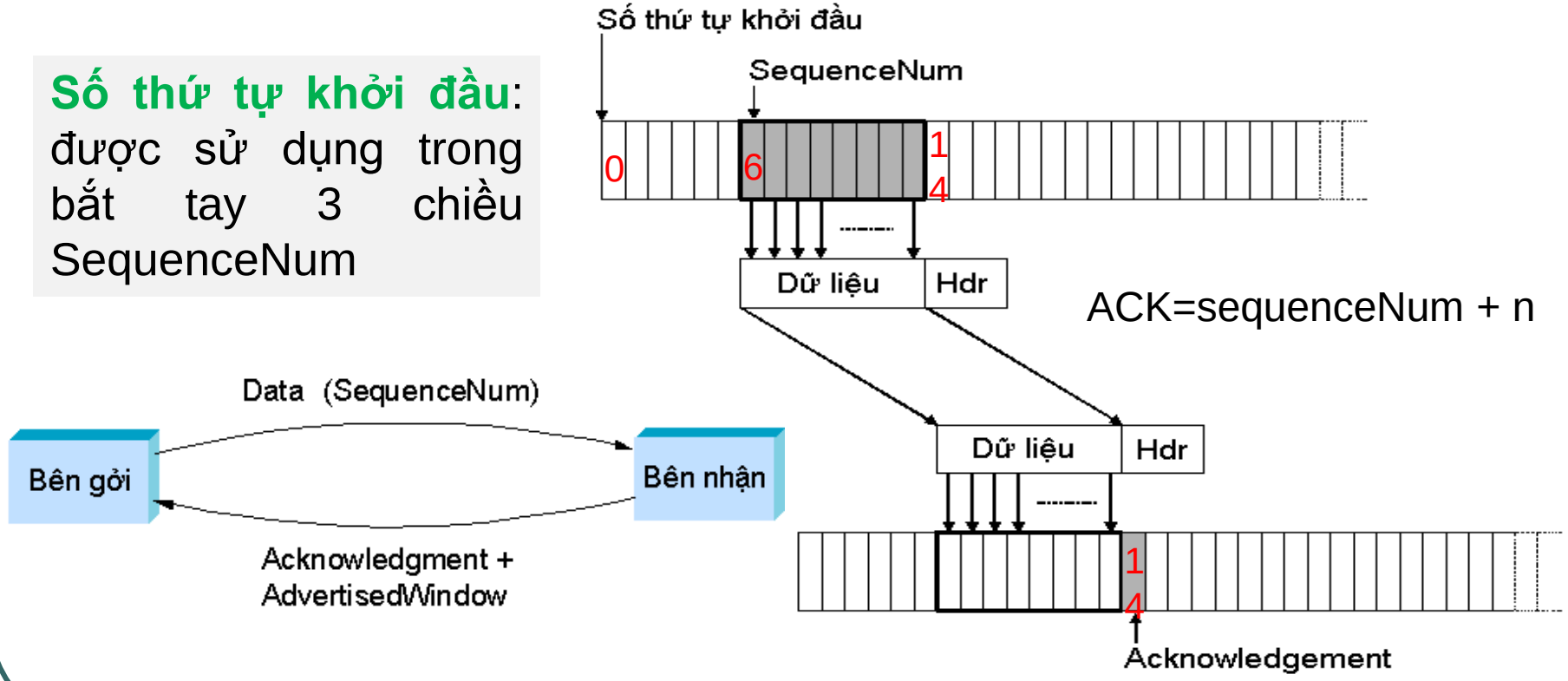
quy trình 4 bước của TCP

Hủy bắt tay trong TCP

Điều khiển thông lượng trong TCP

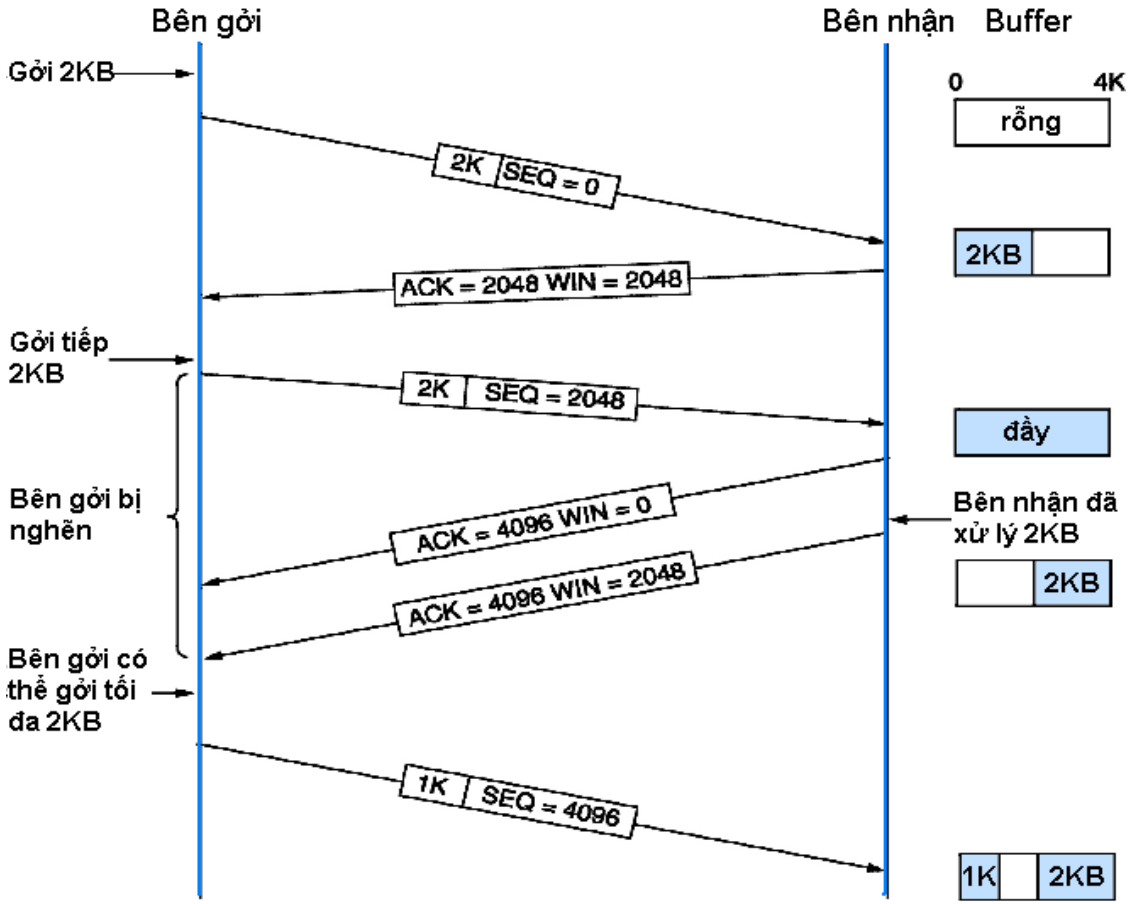
- Là giao thức truyền hướng bytes, mỗi lần truyền đi một Segment

Số thứ tự khởi đầu: được sử dụng trong bắt tay 3 chiều SequenceNum



Điều khiển thông lượng trong TCP

- Sử dụng giao thức cửa sổ trượt



Giao thức TCP
Data Transmission

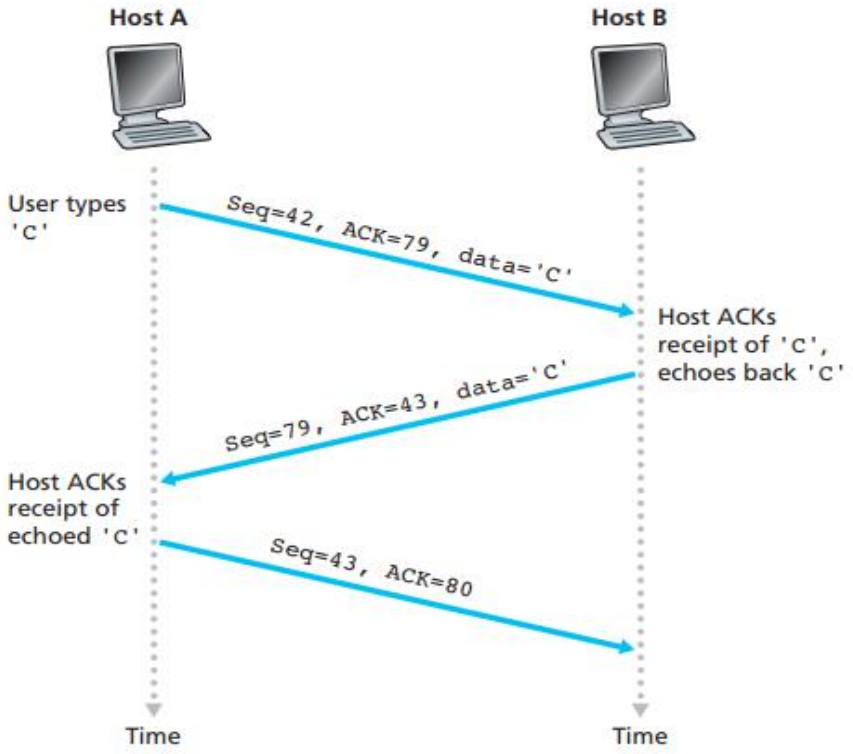
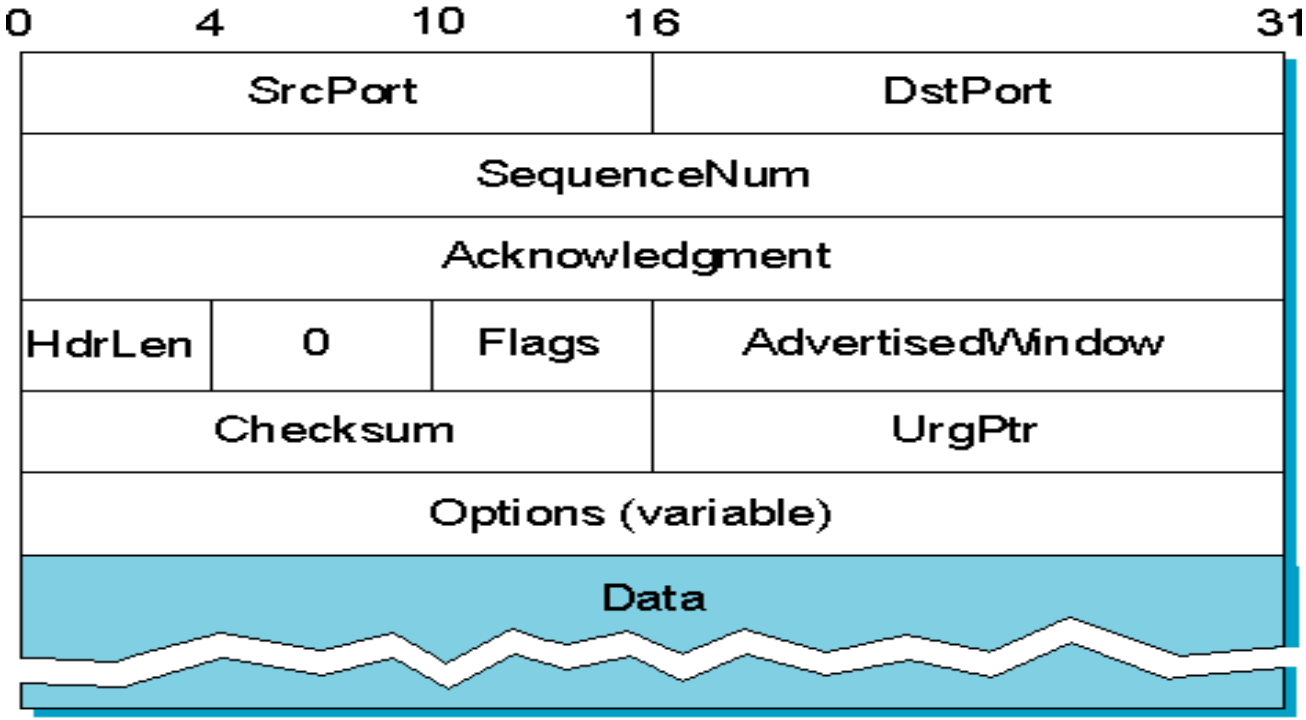


Figure 3.31 ♦ Sequence and acknowledgment numbers for a simple Telnet application over TCP

Giao thức TCP (Transmission Control Protocol)



Flags = [SYN, FIN, RESET, PUSH, URG, ACK]

Giao thức vận chuyển phù hợp ứng dụng

UDP

VoIP
(IP telephony)

DNS
(Domain Name Resolution)

Required protocol properties:

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

TCP

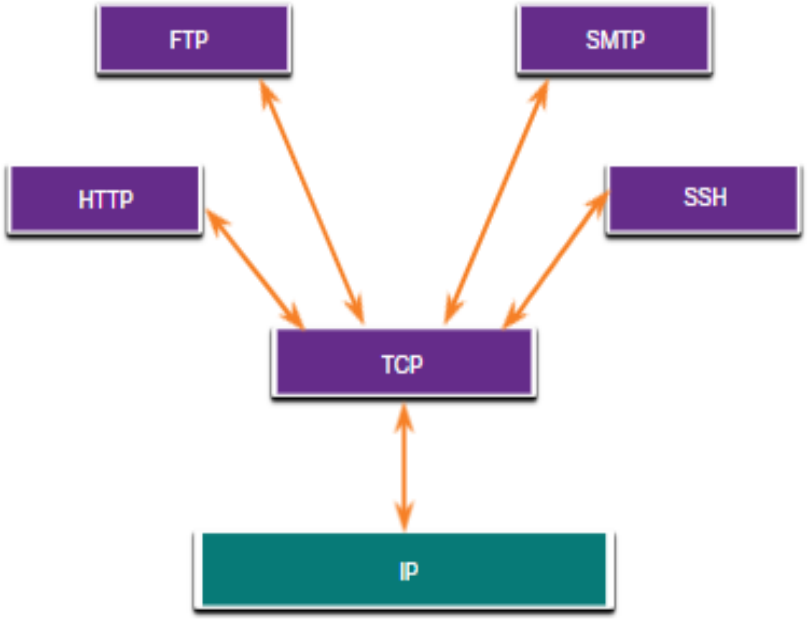
SMTP/IMAP
(Email)

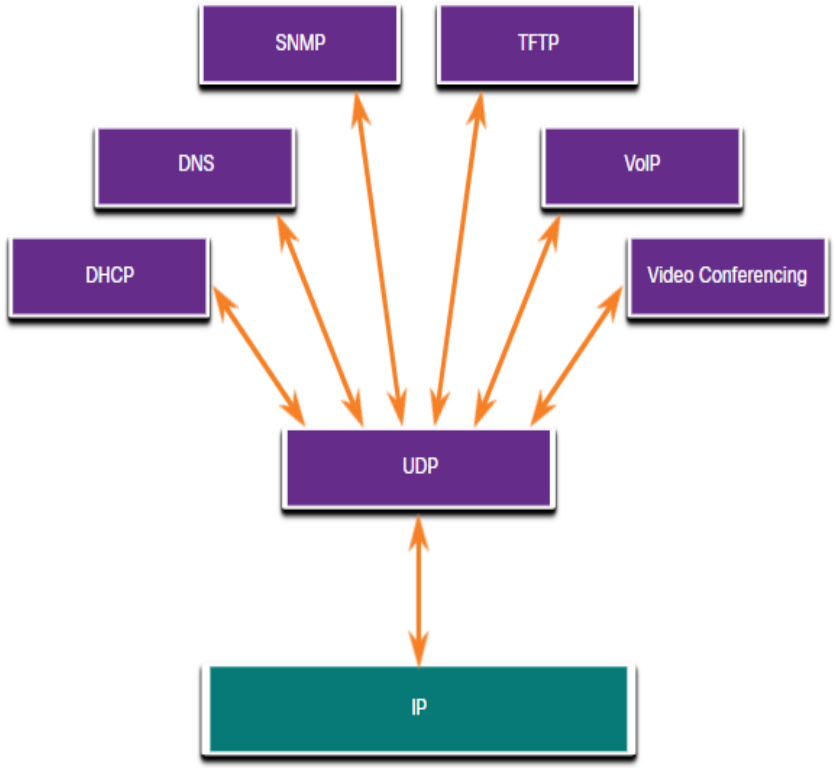
HTTP/HTTPS
(World Wide Web

Required protocol properties:

- Reliable
- Acknowledges data
- Resends lost data
- Delivers data in sequenced order

Ứng dụng sử dụng vận chuyển TCP/UDP





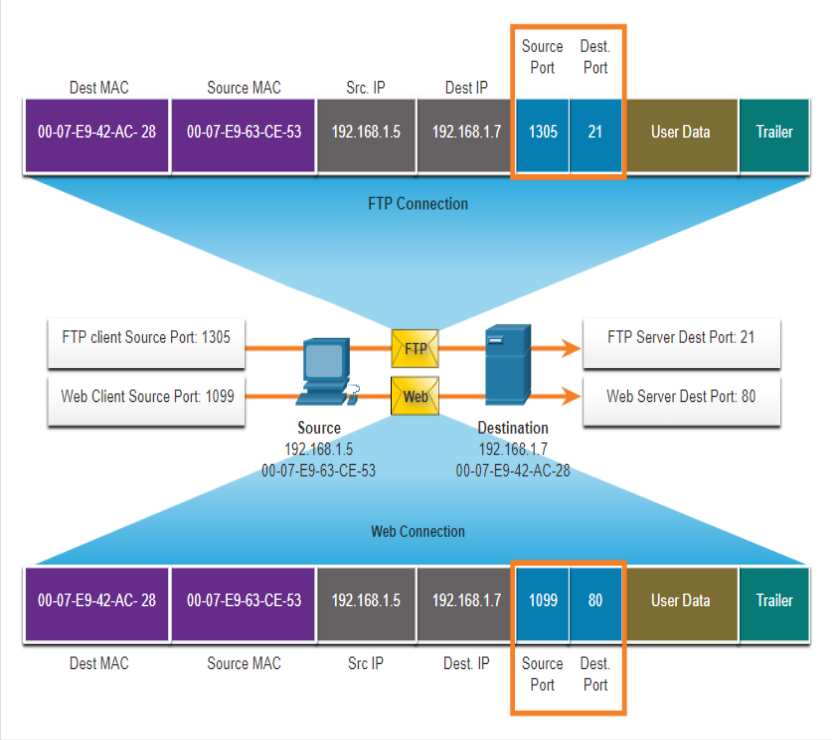
Port Numbers

- TCP và UDP sử dụng số cổng để quản lý nhiều cuộc hội thoại đồng thời.
- Số cổng nguồn (Source Port) được liên kết với ứng dụng gốc trên máy chủ cục bộ trong khi số cổng đích (Destination Port) được liên kết với ứng dụng đích trên máy chủ từ xa.

Port numbers

Socket Pairs

- The source and destination ports are placed within the **segment**.
- The segments are then encapsulated within an IP **packet**.
- The combination of the source IP address and source port number, or the destination IP address and destination port number is known as a **socket**.
- Sockets cho phép nhiều tiến trình, thực thi trên máy client có thể phân biệt với nhau, và nhiều nối kết đồng thời đến server cũng được phân biệt nhau.



Port Numbers
Port Number Groups

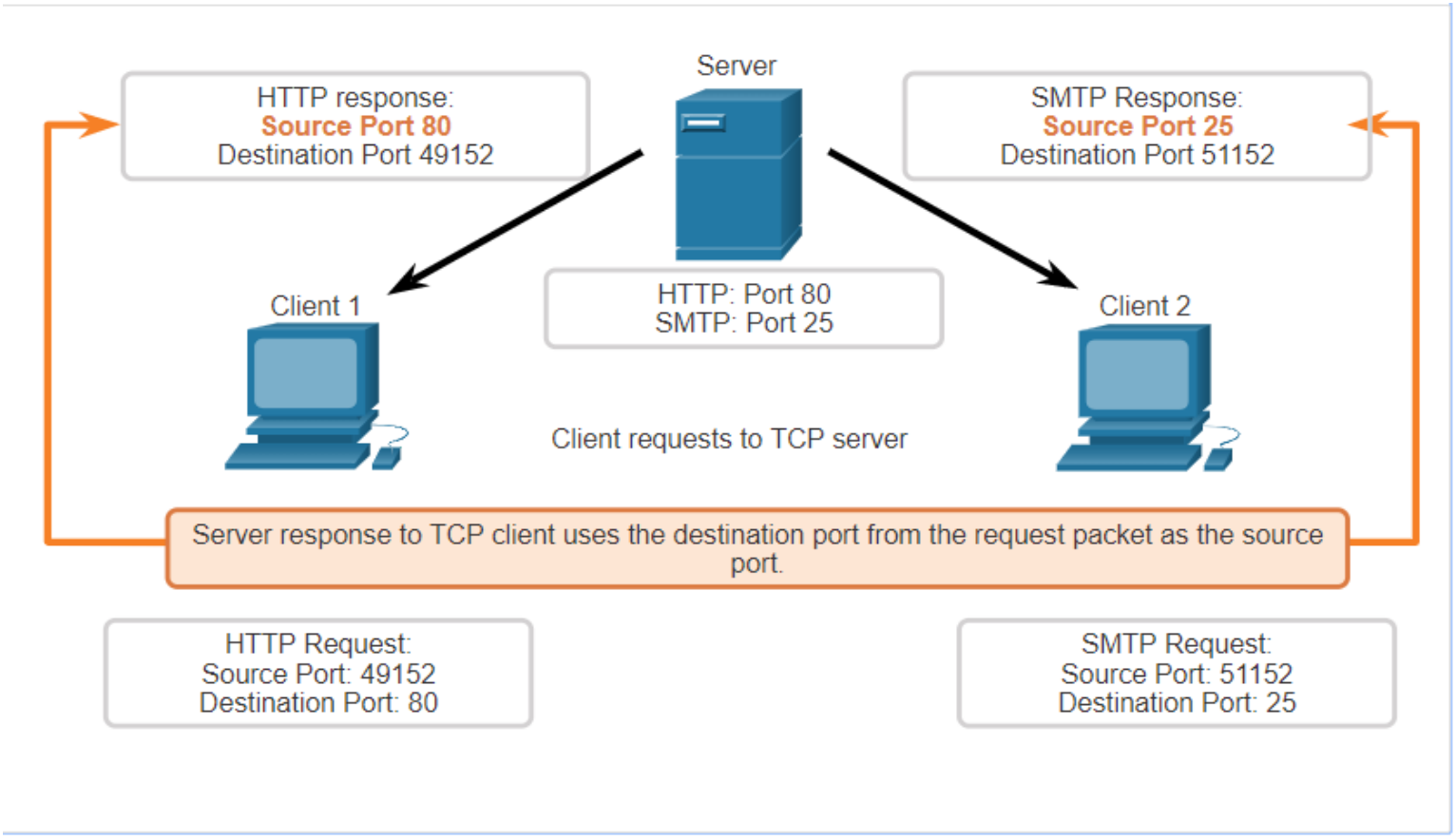
Port Group	Number Range	Description
Well-known Ports	**0 to 1,023**	•These port numbers are reserved for common or popular services and applications such as web browsers, email clients, and remote access clients. •Defined well-known ports for common server applications enables clients to easily identify the associated service required.
Registered Ports	**1,024 to 49,151**	•These port numbers are assigned by IANA to a requesting entity to use with specific processes or applications. •These processes are primarily individual applications that a user has chosen to install, rather than common applications that would receive a well-known port number. •For example, Cisco has registered port 1812 for its RADIUS server authentication process.
Private and/or **Dynamic Ports**	**49,152 to 65,535**	•These ports are also known as *ephemeral ports*. •The client's OS usually assign port numbers dynamically when a connection to a service is initiated. •The dynamic port is then used to identify the client application during communication.

Port Numbers
Port Number Groups (Cont.)

Well-Known Port Numbers

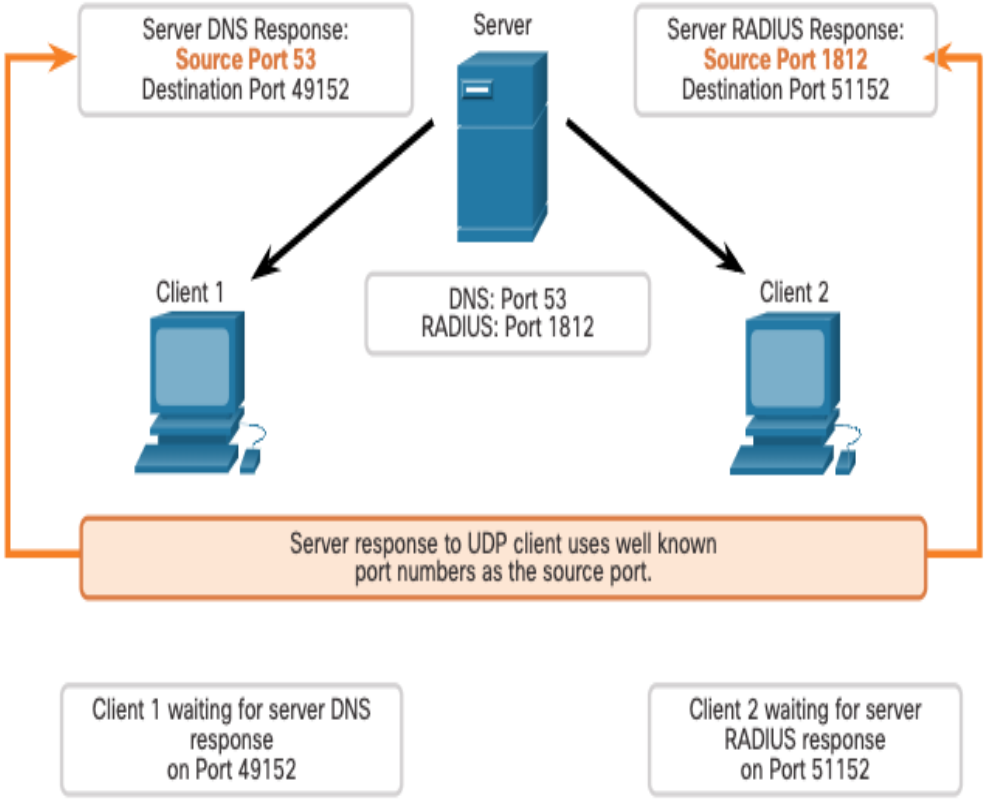
Port Number	Protocol	Application
20	TCP	File Transfer Protocol (FTP) - Data
21	TCP	File Transfer Protocol (FTP) - Control
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	UDP, TCP	Domain Name Service (DNS)
67	UDP	Dynamic Host Configuration Protocol (DHCP) - Server
68	UDP	Dynamic Host Configuration Protocol - Client
69	UDP	Trivial File Transfer Protocol (TFTP)
80	TCP	Hypertext Transfer Protocol (HTTP)
110	TCP	Post Office Protocol version 3 (POP3)
143	TCP	Internet Message Access Protocol (IMAP)
161	UDP	Simple Network Management Protocol (SNMP)
443	TCP	Hypertext Transfer Protocol Secure (HTTPS)

TCP Server/ Client Processes



UDP Server/ Client Processes

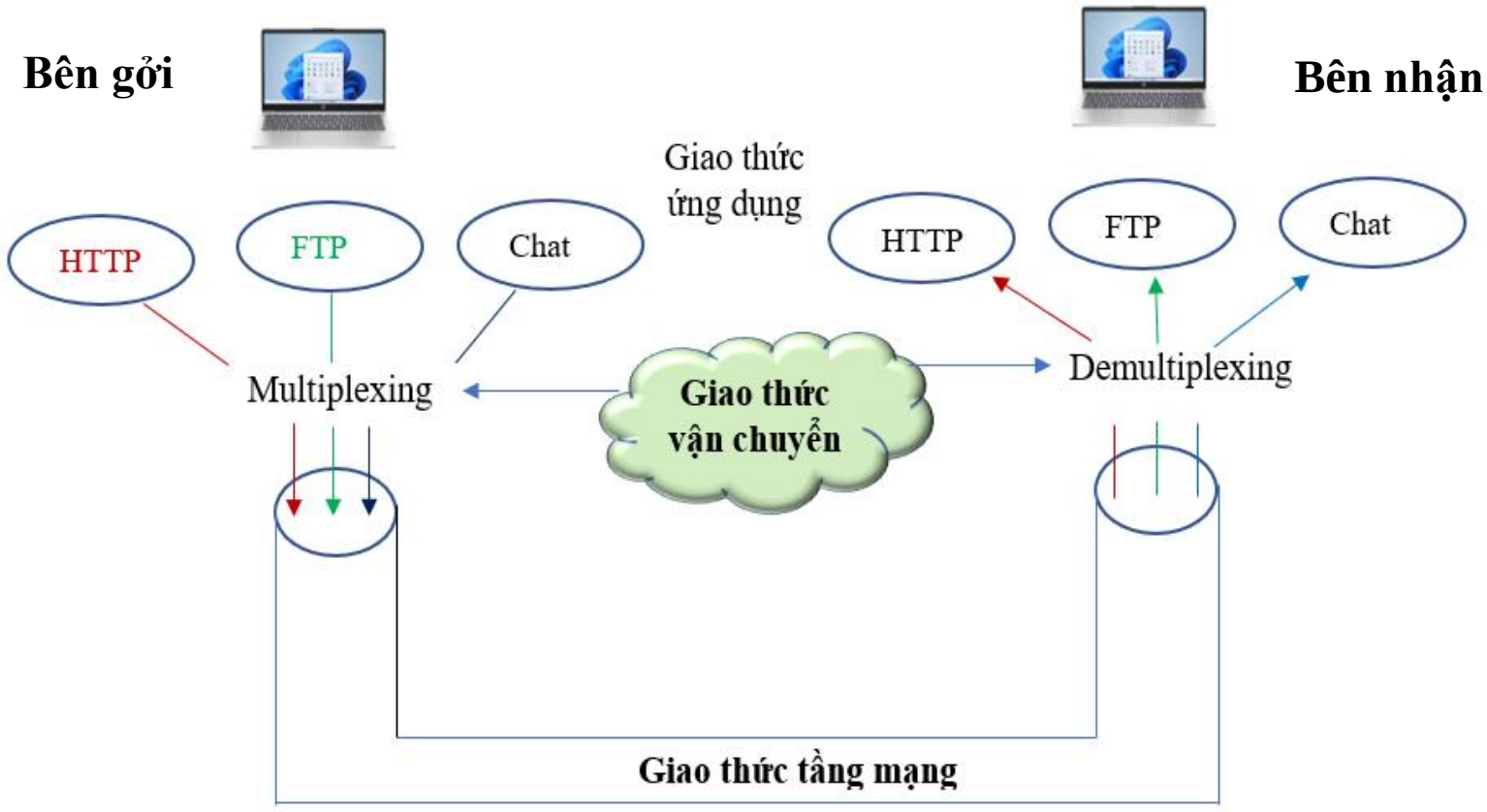
- Client tự động chọn số cổng (Port) từ dãy số cổng và sử dụng số cổng này làm cổng nguồn cho cuộc hội thoại.
- Cổng đích thường là số cổng đã đăng ký hoặc nổi tiếng được gán cho quy trình máy chủ.



Multiplexing/Demultiplexing

- **Bên gởi: thực hiện Dồn kênh**
  - Nhận dữ liệu từ tầng ứng dụng (từ các socket)
  - Phân đoạn thông điệp ở tầng ứng dụng thành các segment
  - Dán nhãn dữ liệu: đóng gói theo giao thức tại tầng Transport
  - Chuyển các segment xuống tầng mạng (network layer)
- **Bên nhận: thực hiện Phân kênh**
  - Nhận các segment từ tầng mạng
  - Phân rã các segment thành thông điệp tầng ứng dụng
  - Chuyển thông điệp lên tầng ứng dụng (đến socket tương ứng)

Multiplexing/Demultiplexing



Tham khảo

- https://www.geeksforgeeks.org/multiplexing-and-demultiplexing-in-transport-layer/
- https://www.youtube.com/watch?app=desktop&v=9e4vTcaEYCg