KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN MẠNG VÀ CÁC HỆ THỐNG THÔNG TIN

CHƯƠNG 4

Tầng Giao vận

MỤC TIÊU

- Mô tả mục đích của tầng trong việc quản lý quá trình vận chuyển dữ liệu
- Mô tả đặc tính của giao thức TCP và UDP
- Giải thích cách hoạt động của giao thức TCP và UDP

TỪ KHÓA

- Tầng Giao vận
- Giao thức UDP
- Giao thức TCP

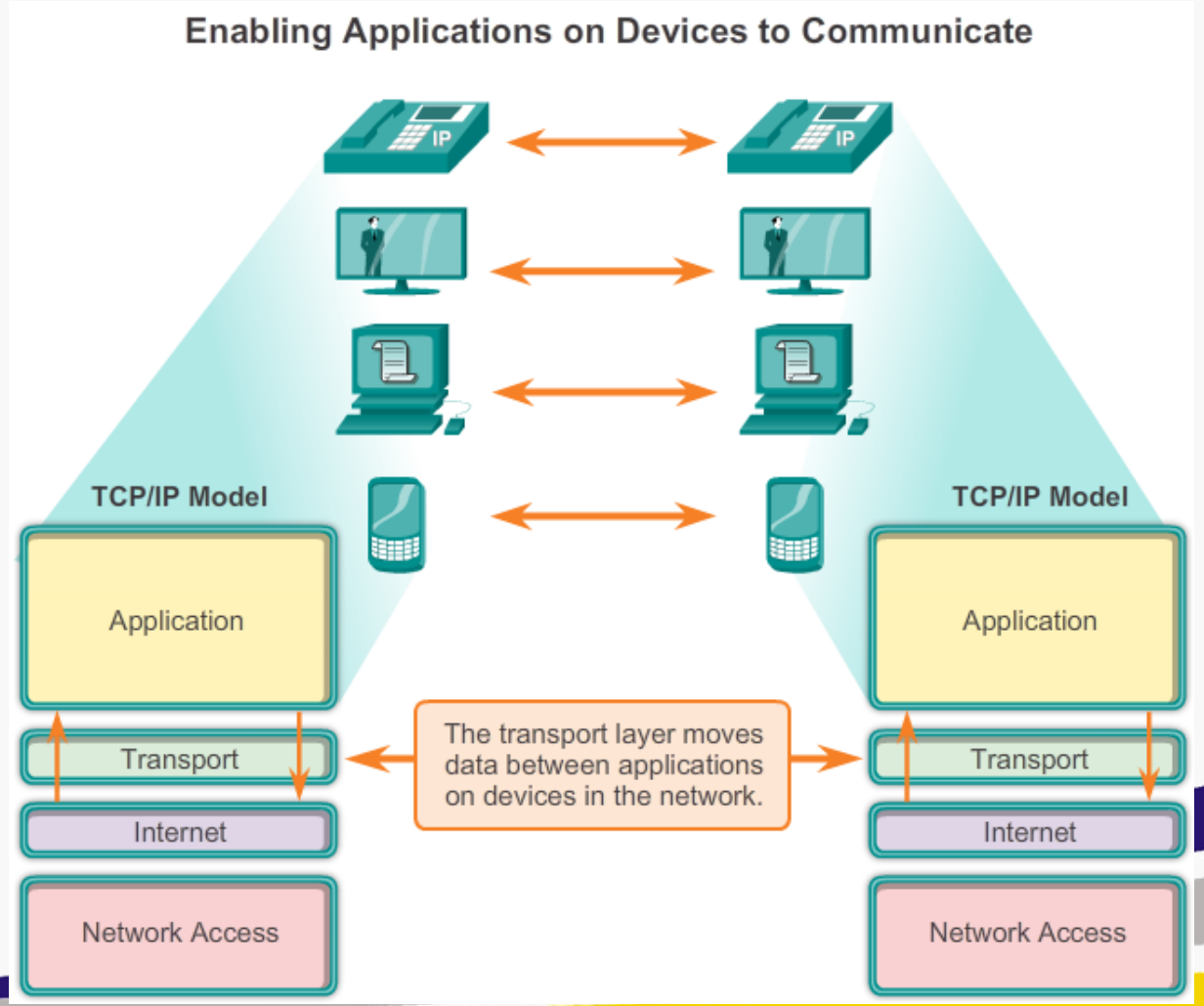
NỘI DUNG

- **Phần 1**: Tầng giao vận
- **Phần 2**: Giao thức UDP
- **Phần 3**: Giao thức TCP

Tầng Giao vận - Tổng quan

Tầng giao vận:

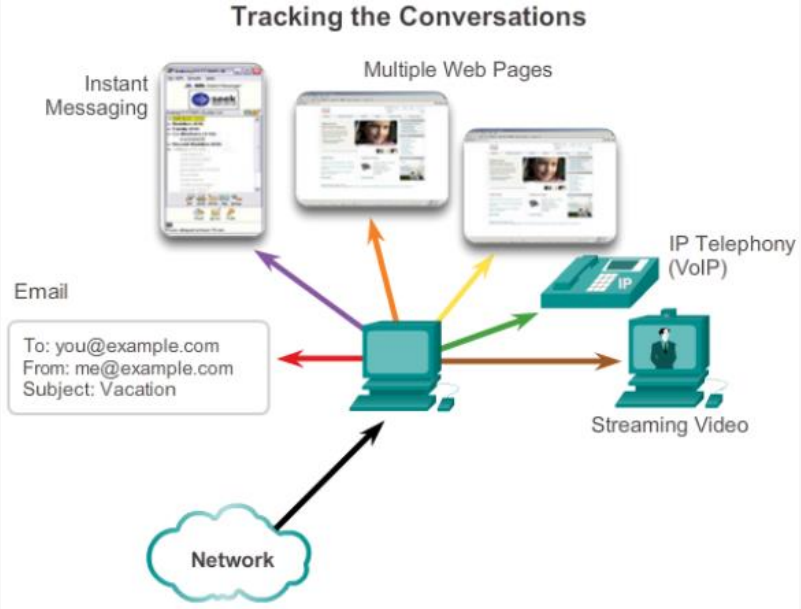
- Chịu trách nhiệm thiết lập phiên kết nối tạm thời và phân phối dữ liệu giữa hai ứng dụng
- Liên kết tầng ứng dụng và các tầng thấp hơn trong mô hình TCP/IP

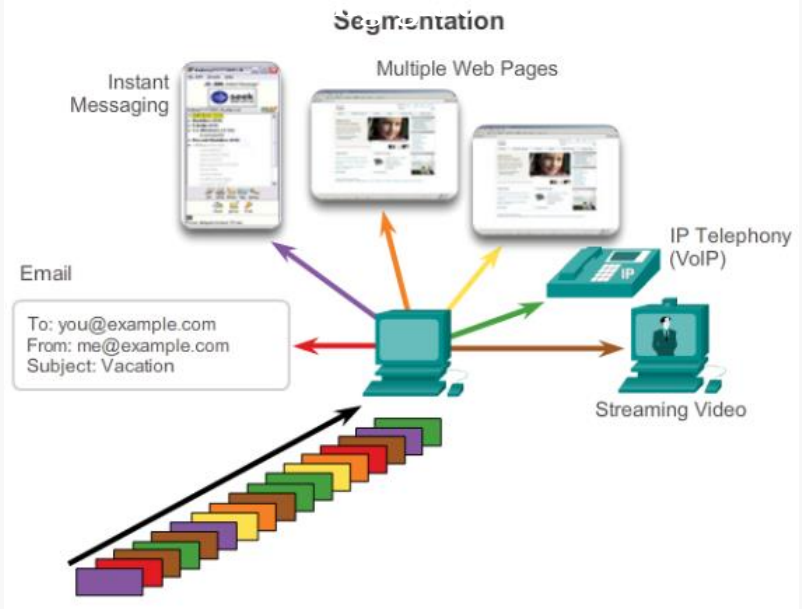


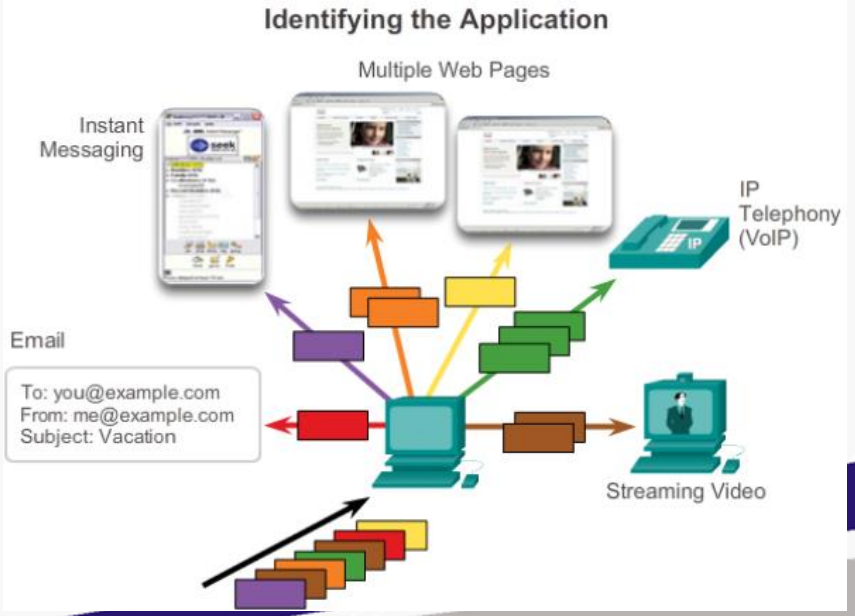
Tầng Giao vận - Tổng quan

Nhiệm vụ chính tầng giao vận:

- Tách biệt luồng dữ liệu được trao đổi giữa các ứng dụng trên các thiết bị với nhau
- Phân mảnh và tái hợp dữ liệu
- Định danh các ứng dụng







Tầng Giao vận - Tổng quan

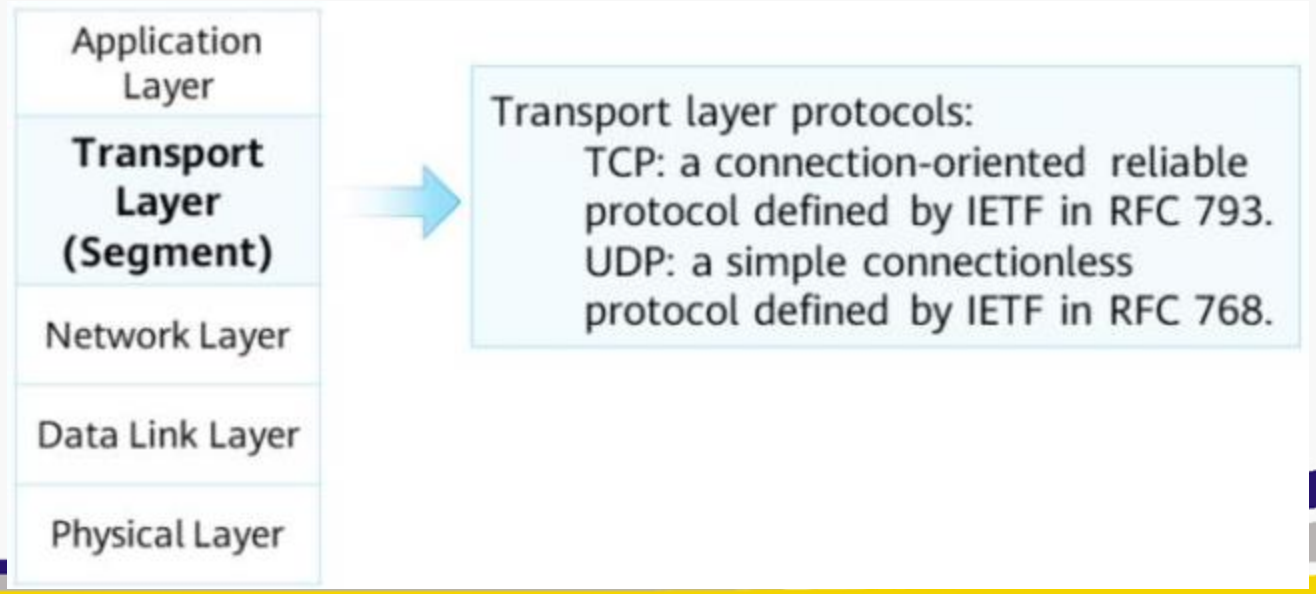
Nhiệm vụ chính tầng giao vận:

- Tách biệt luồng dữ liệu được trao đổi giữa các ứng dụng trên các thiết bị với nhau
  - ✓ Theo dõi việc trao đổi thông tin giữa các ứng dụng riêng biệt trên các thiết bị nguồn và đích
- Phân mảnh và tái hợp dữ liệu
  - ✓ Phân mảnh dữ liệu để quản lý và tái hợp lại dữ liệu đã bị phân mảnh thành các luồng dữ liệu gửi lên tầng ứng dụng ở thiết bị đích
- Định danh các ứng dụng
  - ✓ Định danh ứng dụng tương ứng với từng luồng dữ liệu

Tầng Giao vận - Tổng quan

Giao thức tầng giao vận:

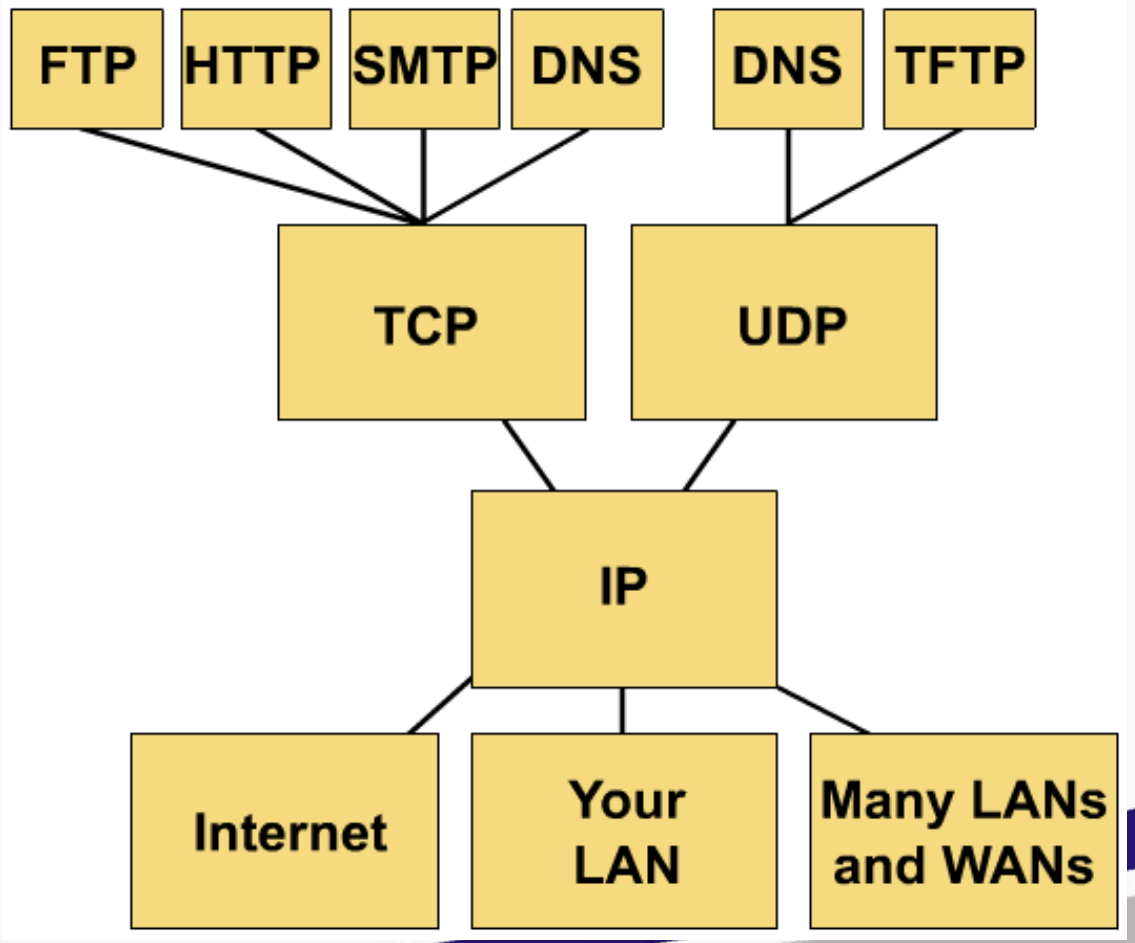
- Tầng giao vận đảm bảo dữ liệu có thể được tập hợp lại một cách chính xác ở bên nhận bằng cách phân mảnh và tái hợp.
- Trong TCP/IP, các tiến trình này có thể đạt được bằng cách sử dụng hai giao thức khác nhau:
  - ✓ Transmission Control Protocol (TCP): Định nghĩa bởi IETF trong RFC 793
  - ✓ User Datagram Protocol (UDP): Định nghĩa bởi IETF trong RFC 768



Tầng giao vận – Tổng quan

Giao thức tầng giao vận:

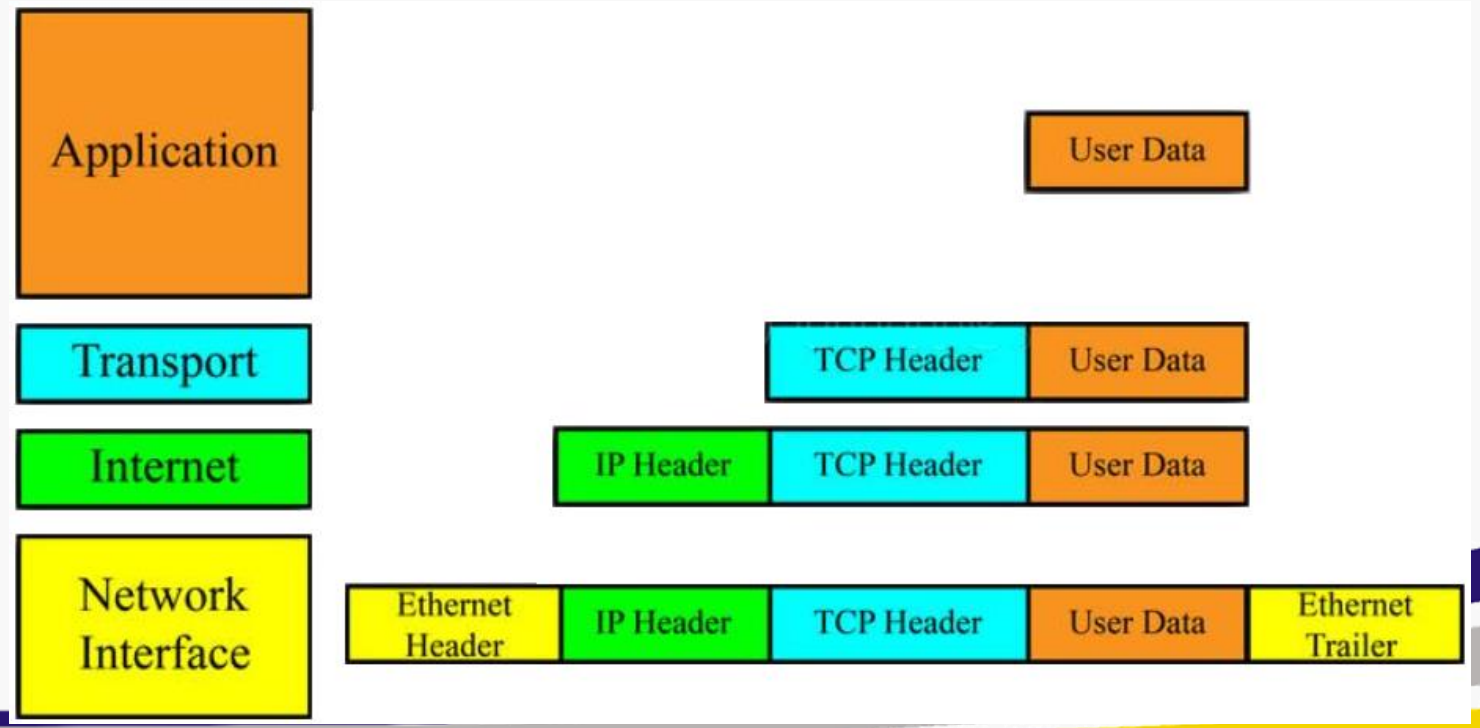
- TCP là một giao thức tin cậy, hướng kết nối (nghĩa là phải thực hiện thiết lập kết nối trước khi thực hiện truyền dữ liệu)
- UDP là một giao thức không tin cậy, không hướng không kết nối (nghĩa là không phải thực hiện thiết lập kết nối trước khi thực hiện truyền dữ liệu)



Tầng Giao vận - Đóng gói

Đóng gói ở tầng giao vận:

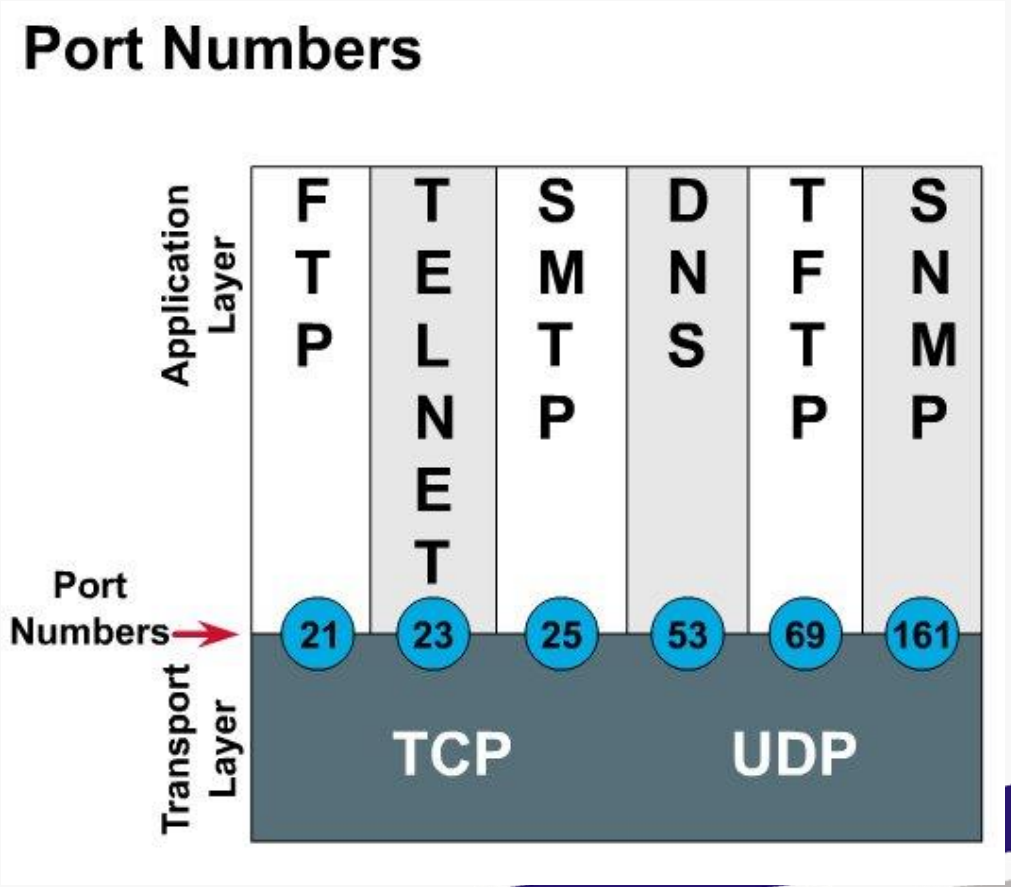
- Dữ liệu được đóng gói trong tầng Giao vận được gọi là Segment hoặc Datagram.
- Đối với mỗi giao thức:
  - ✓ TCP: Segment
  - ✓ UDP: Datagram



Tầng Giao vận - Số hiệu cổng và Socket

Số hiệu cổng:

- Là một phần trong tầng giao vận, được dùng trong quá trình truyền dữ liệu
- Mỗi cổng được hiển thị bằng một số nguyên không dấu 16-bit do IANA cấp.
- Số hiệu cổng được liên kết với giao thức tầng Giao vận để mô tả một ứng dụng.
- Ví dụ:
  - ✓ TCP/80 = HTTP; UDP/161 = SNMP
  - ✓ TCP/53 hoặc UDP/53 = DNS

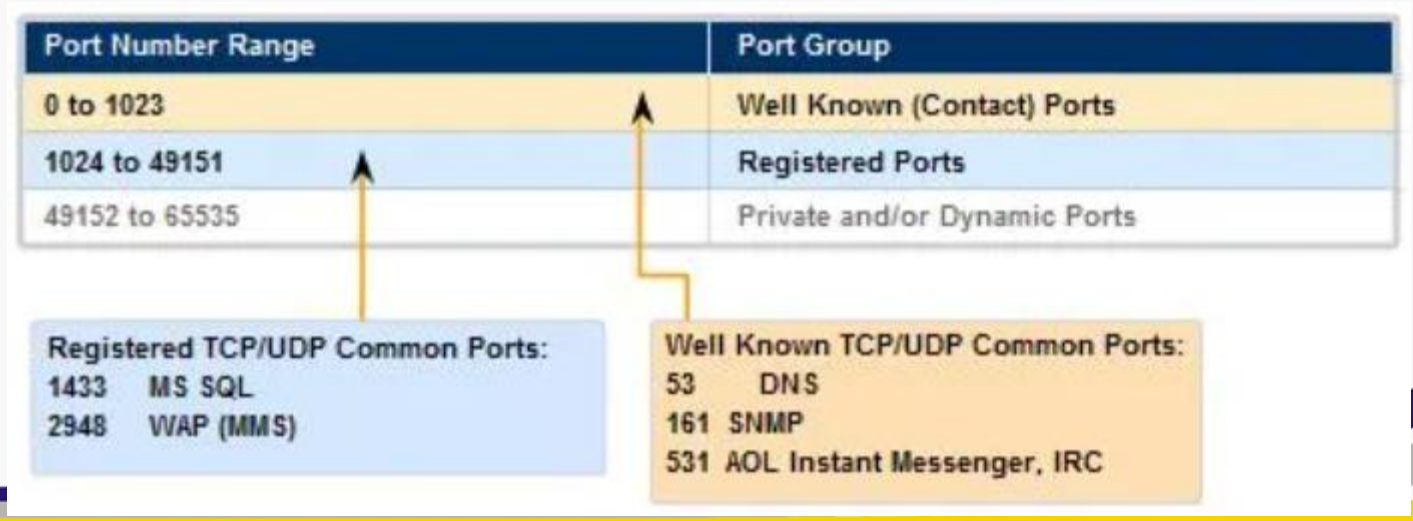


Tầng Giao vận - Số hiệu cổng và Socket

Các loại số hiệu cổng:

Phạm vi từ 0 tới 65535 và được chia thành 3 loại:

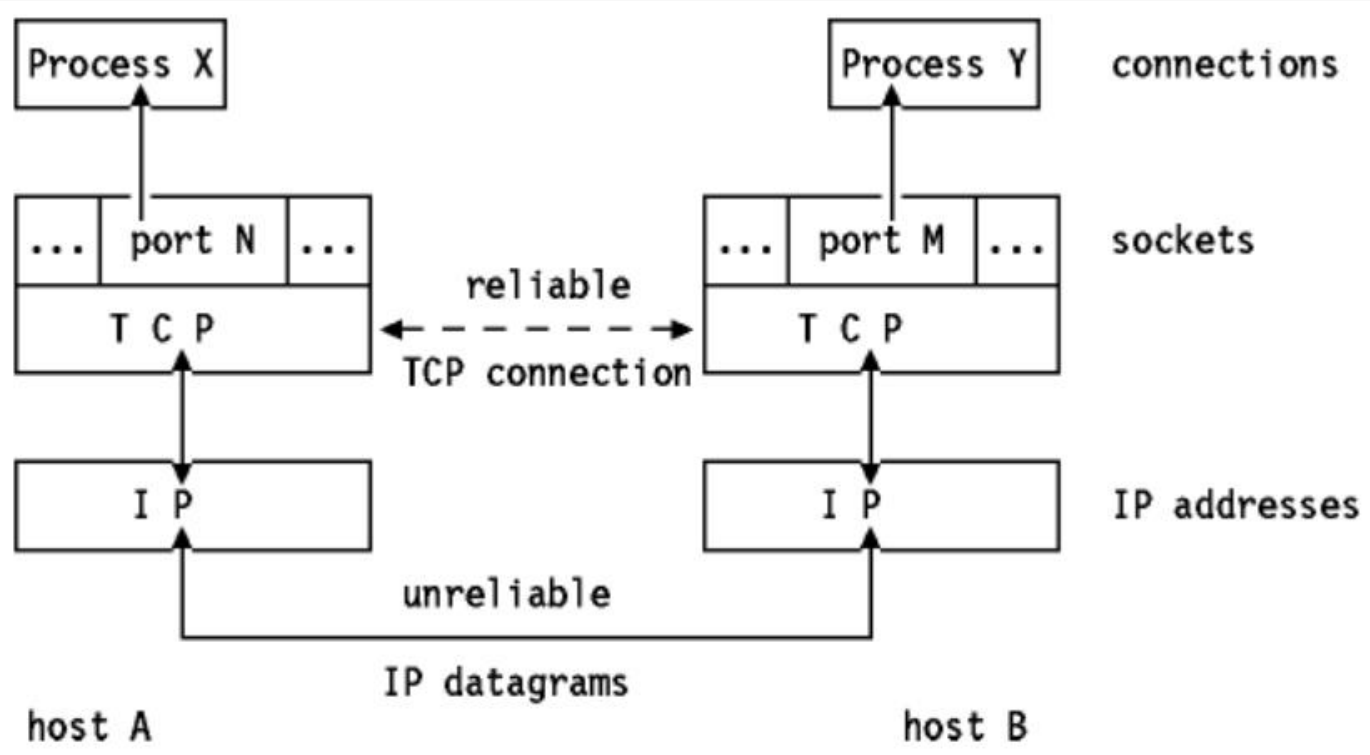
- Cổng thông dụng (Well-known): Dùng cho các dịch vụ và ứng dụng (Ví dụ DNS, HTTP, SNMP,...)
- Cổng đăng ký (Registered): Người dùng có thể tự đăng ký số hiệu cổng cho ứng dụng hoặc tiến trình
- Cổng động hoặc cổng tạm thời (Dynamic or Ephemeral): Được gán động cho các ứng dụng của máy khách khi nó khởi tạo kết nối tới một dịch vụ.



Tầng Giao vận - Số hiệu cổng và Socket

Socket:

- Số hiệu cổng là mã định danh duy nhất được sử dụng cùng với địa chỉ IP để tạo nên một socket
- Một socket định danh duy nhất điểm cuối của một luồng dữ liệu trao đổi giữa các ứng dụng.
- Một socket (hoặc địa chỉ socket) có ba phần: Địa chỉ IP của máy, giao thức truyền tải tầng giao vận, số hiệu cổng của dịch vụ trên máy.



UDP - Tổng quan

Các chức năng cơ bản:

UDP cung cấp các chức năng cơ bản để phân phối các datagram giữa các ứng dụng tương ứng, với rất ít chi phí và kiểm tra dữ liệu.

- UDP là một giao thức không kết nối.
- UDP được biết đến như một giao thức phân phối nỗ lực tối đa vì không có xác nhận rằng dữ liệu đã được nhận ở đích.

UDP - Tổng quan

Chức năng:

- Dữ liệu được xây dựng lại theo thứ tự nhận được
- Bất kỳ segment nào bị mất sẽ không được gửi lại.
- Không thiết lập phiên
- Việc gửi không được thông báo về tính sẵn có của tài nguyên.

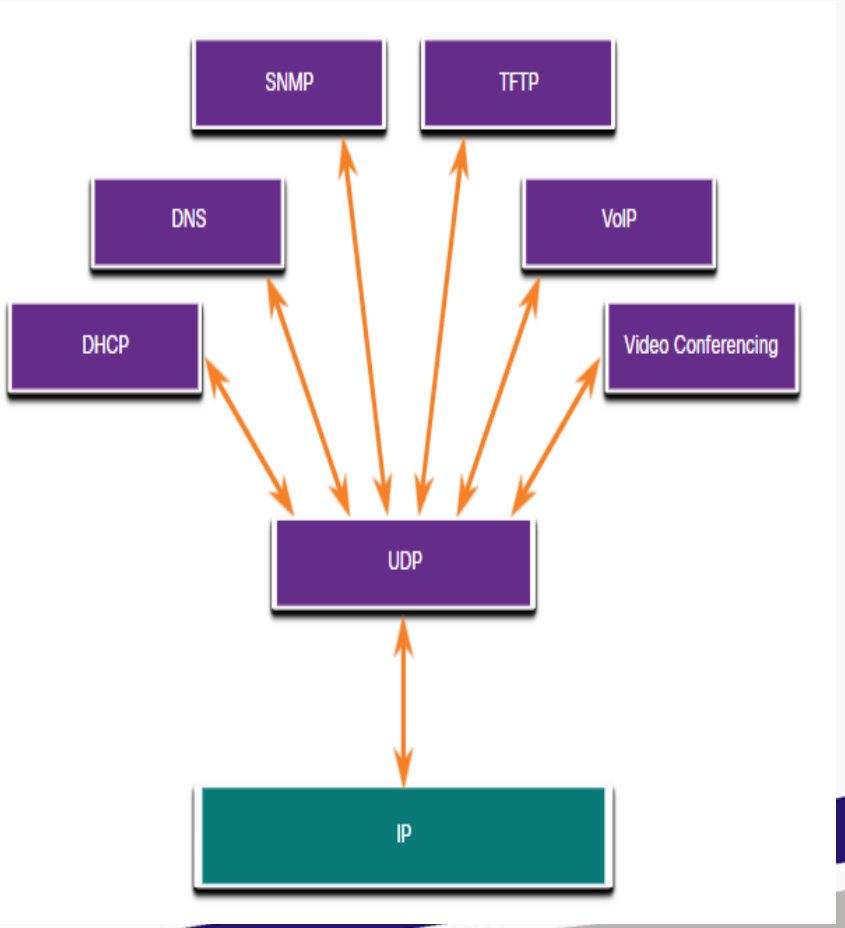
UDP - Tổng quan

Các ứng dụng sử dụng giao thức UDP

Ứng dụng truyền hình trực tiếp và đa phương tiện
- Các ứng dụng này có thể chịu được một số trường hợp mất dữ liệu nhưng yêu cầu ít hoặc không có độ trễ.
- Ví dụ bao gồm VoIP và truyền hình trực tiếp

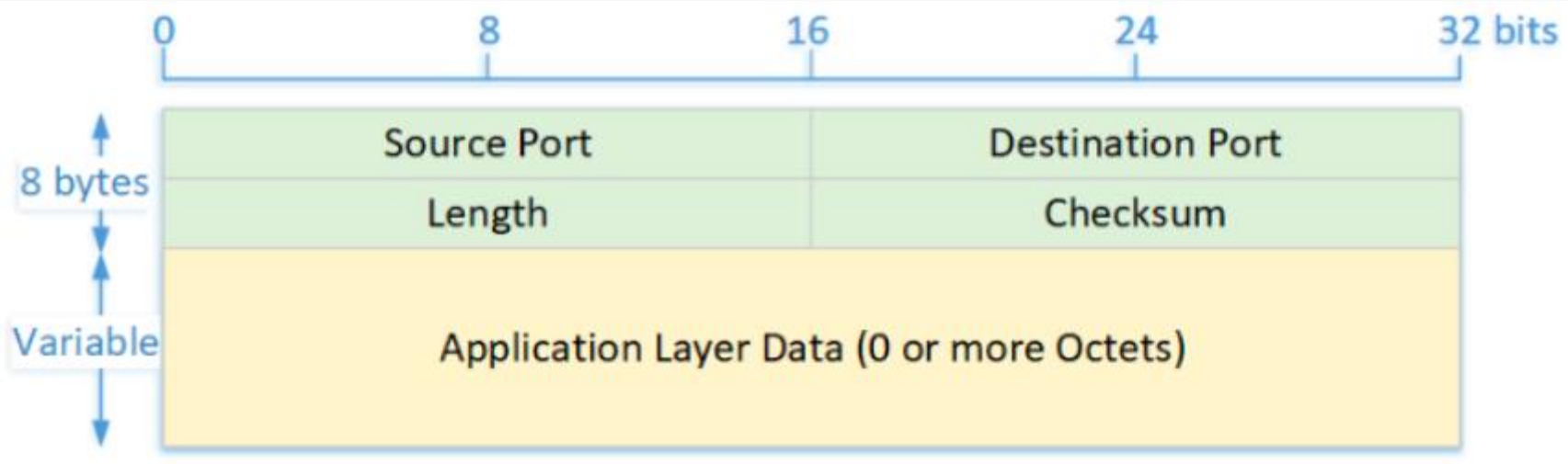
Ứng dụng yêu cầu và trả lời đơn giản
- Các ứng dụng có các giao dịch đơn giản trong đó máy chủ gửi yêu cầu và có thể nhận hoặc không nhận được phản hồi.
- Ví dụ bao gồm DNS và DHCP.

Các ứng dụng tự xử lý độ tin cậy
- Giao tiếp một chiều trong đó không yêu cầu kiểm soát luồng, phát hiện lỗi, xác nhận và khôi phục lỗi hoặc có thể được ứng dụng xử lý.
- Ví dụ SNMP và TFTP.



UDP - Định dạng phần tiêu đề

Tiêu đề UDP: 8 bytes



Field	Length	Description	Value
Source Port	16 bits	It identifies the port of the sending application	
Destination Port	16 bits	It identifies the port of the receiving application	
Length	16 bits	Specifies the total length of the UDP header and data (min = 8-byte header; max = 65535 bytes of 8-byte header and 65527-byte data)	
Checksum	16 bits	Checksum of the UDP header and UDP data	

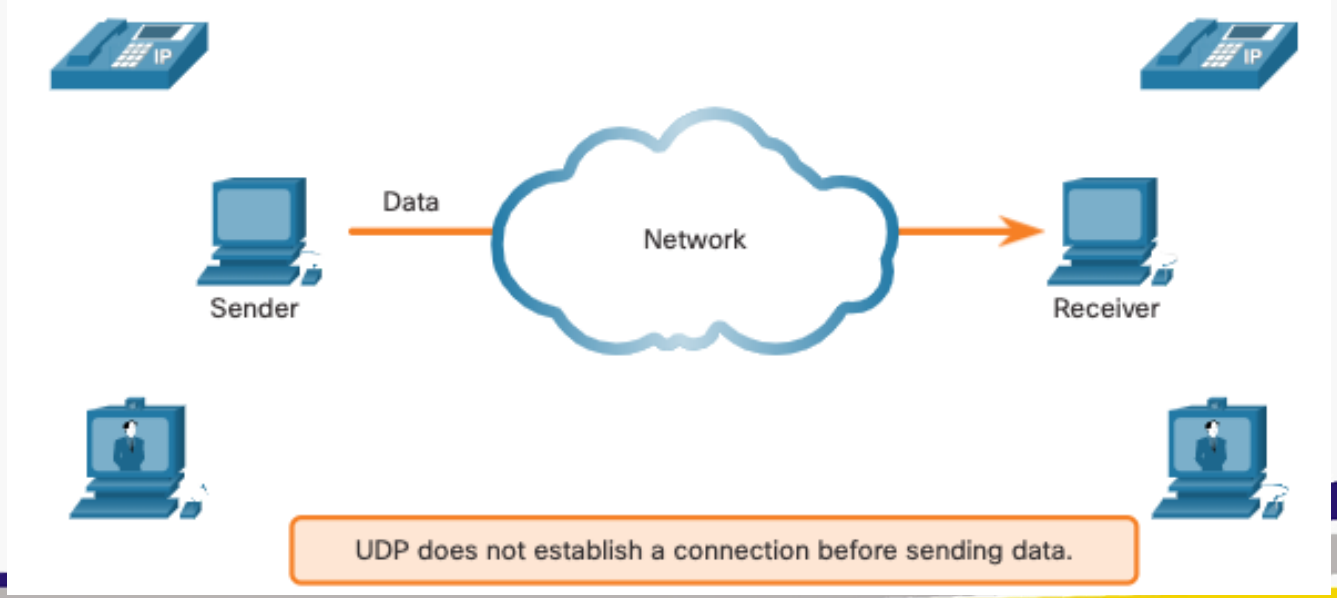
UDP - Số hiệu cổng thông dụng

PORT NUMBER	TRANSPORT PROTOCOL	SERVICE NAME	RFC
67, 68	UDP	Dynamic Host Configuration Protocol (DHCP)	RFC 2131
69	UDP	Trivial File Transfer Protocol (TFTP)	RFC 1350
123	UDP	Network Time Protocol (NTP)	RFC 5905
135-139	UDP and TCP	NetBIOS	RFC 1001-1002
500	UDP	Internet Security Association and Key Management Protocol (ISAKMP) / Internet Key Exchange (IKE)	RFC 2408 - 2409
53	UDP and TCP	Domain Name Server (DNS)	RFC 1034-1035
161, 162	UDP and TCP	Simple Network Management Protocol (SNMP)	RFC 1901-1908, 3411-3418
22	UDP and TCP	Secure Shell (SSH)	RFC 4250-4256
143	UDP and TCP	Internet Message Access Protocol (IMAP4)	RFC 3501
389	UDP and TCP	Lightweight Directory Access Protocol	RFC 4510
443	UDP and TCP	HTTP with Secure Sockets Layer (SSL)	RFC 2818
636	UDP and TCP	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)	RFC 4513

UDP - Truyền thông

Tiêu tốn ít tài nguyên và độ tin cậy thấp

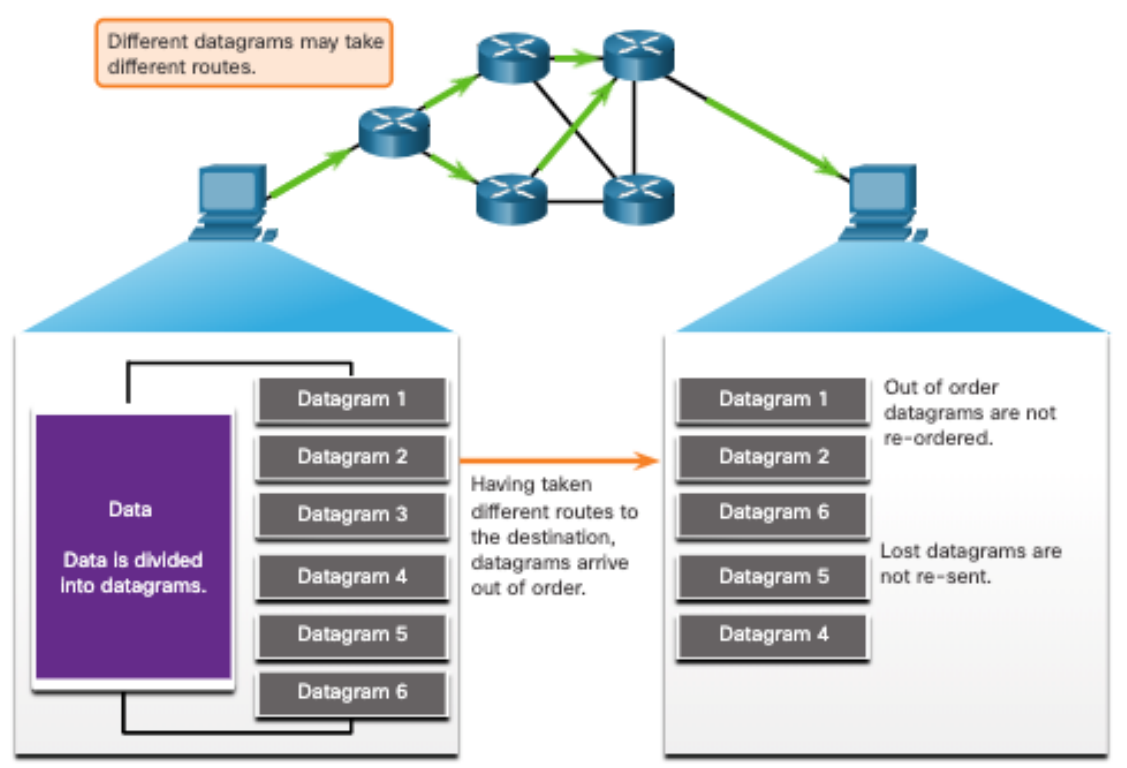
- UDP không thiết lập kết nối trước khi truyền
- UDP tiêu tốn ít tài nguyên khi truyền tải dữ liệu vì nó có tiêu đề nhỏ và không phát sinh lưu lượng quản lý mạng.



UDP - Truyền thông

Tái hợp Datagram trong UDP:

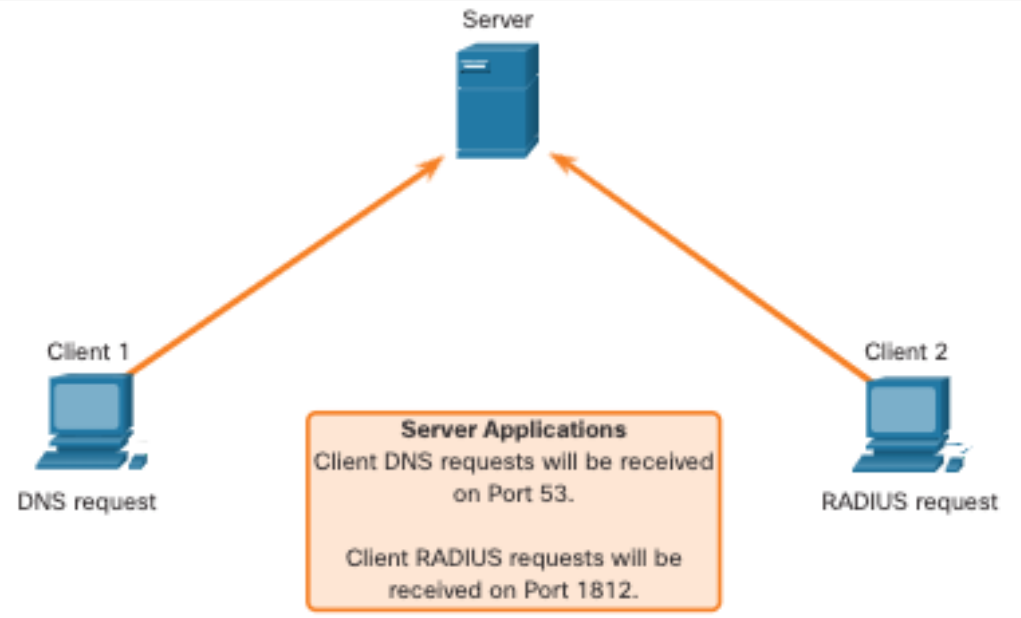
- UDP không theo dõi số thứ tự datagram.
- UDP không sắp xếp các datagram theo thứ tự truyền
- UDP chỉ tái hợp lại dữ liệu theo thứ tự nhận được và chuyển tiếp đến ứng dụng.



UDP - Truyền thông

Nhận và xử lý dữ liệu trên máy chủ/Server

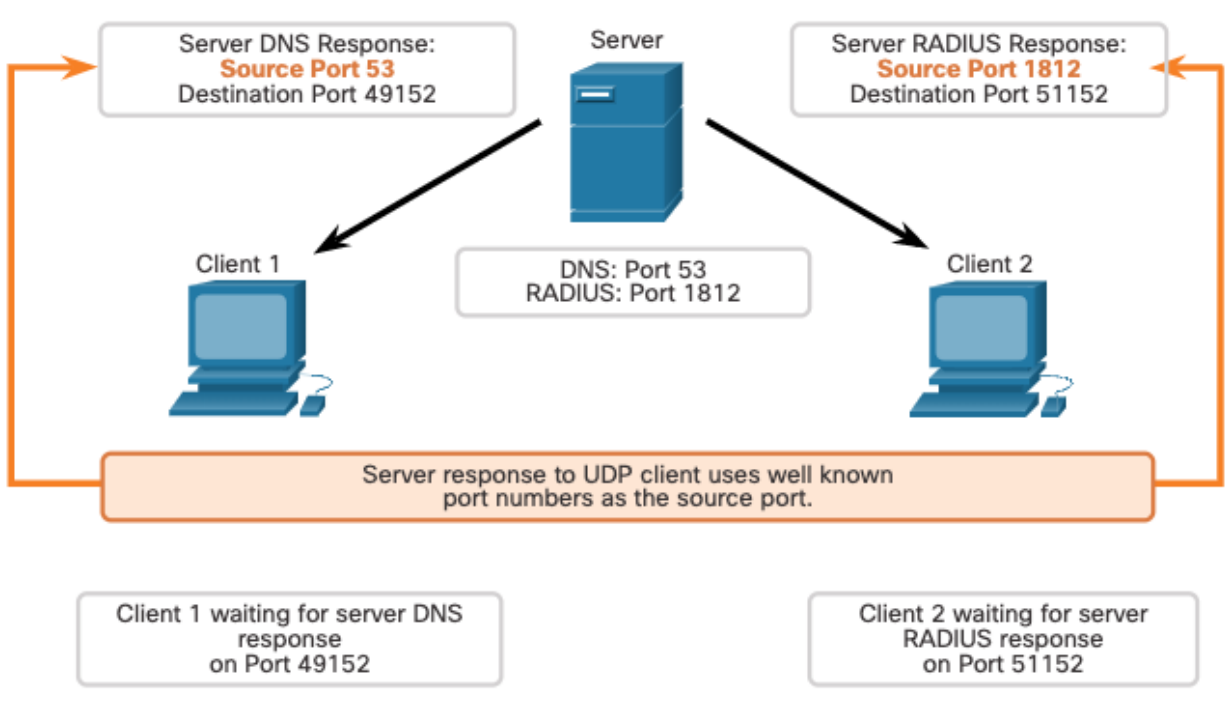
- Các ứng dụng trên máy chủ sử dụng dịch vụ UDP được gán số hiệu cổng thông dụng hoặc đăng ký.
- UDP nhận một datagram dành cho một trong các cổng này, tiếp theo nó chuyển tiếp dữ liệu đến ứng dụng thích hợp dựa trên số cổng của nó.



UDP - Truyền thông

Quá trình xử lý trên máy khách/Client

- Máy khách tự động chọn số hiệu cổng từ dãy số đã quy định và sử dụng số hiệu cổng này làm cổng nguồn để trao đổi dữ liệu.
- Cổng đích thường là cổng thông dụng hoặc đăng ký được gán cho dịch vụ trên máy chủ.
- Sau khi client đã chọn số hiệu cổng nguồn và cổng đích, cặp cổng tương tự sẽ được sử dụng trong tiêu đề của tất cả các gói datagram trao đổi.



TCP - Tổng quan

Các chức năng cơ bản

TCP cung cấp độ tin cậy và kiểm soát luồng. Các hoạt động cơ bản của TCP:

- Đánh số và theo dõi các phân mảnh dữ liệu được truyền đến một máy chủ cụ thể từ một ứng dụng xác định
- Gửi xác nhận về dữ liệu đã nhận
- Truyền lại bất kỳ dữ liệu nào chưa được xác nhận sau một khoảng thời gian nhất định
- Dữ liệu gửi đi theo thứ tự, nhưng có thể đến sai thứ tự
- Gửi dữ liệu với tốc độ tương thích với khả năng nhận của máy đích

TCP - Tổng quan

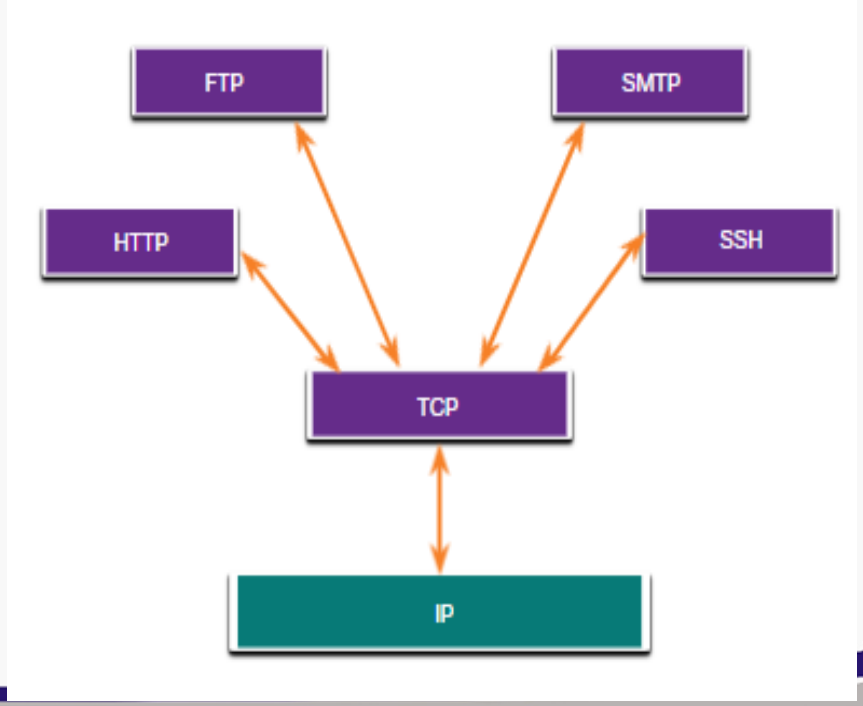
Các chức năng

- **Thiết lập phiên** - TCP là giao thức hướng kết nối, trao đổi và thiết lập kết nối (hoặc phiên) giữa các thiết bị nguồn và đích trước khi truyền dữ liệu.
- **Truyền tin tin cậy** - Vì nhiều lý do, một segment có thể bị hỏng hoặc bị mất khi truyền qua mạng. TCP đảm bảo rằng mỗi segment được gửi từ nguồn sẽ phải đến đích.
- **Điều khiển luồng** - Thiết bị mạng có tài nguyên hạn chế (vd. bộ nhớ và khả năng xử lý). Khi TCP biết rằng các tài nguyên này đã bị sử dụng quá mức, nó có thể yêu cầu ứng dụng phía gửi giảm tốc độ luồng dữ liệu.
- **Xắp xếp dữ liệu nhận theo thứ tự gửi** - Vì các mạng có thể cung cấp nhiều tuyến đường với tốc độ truyền khác nhau nên dữ liệu có thể đến sai thứ tự.

TCP - Tổng quan

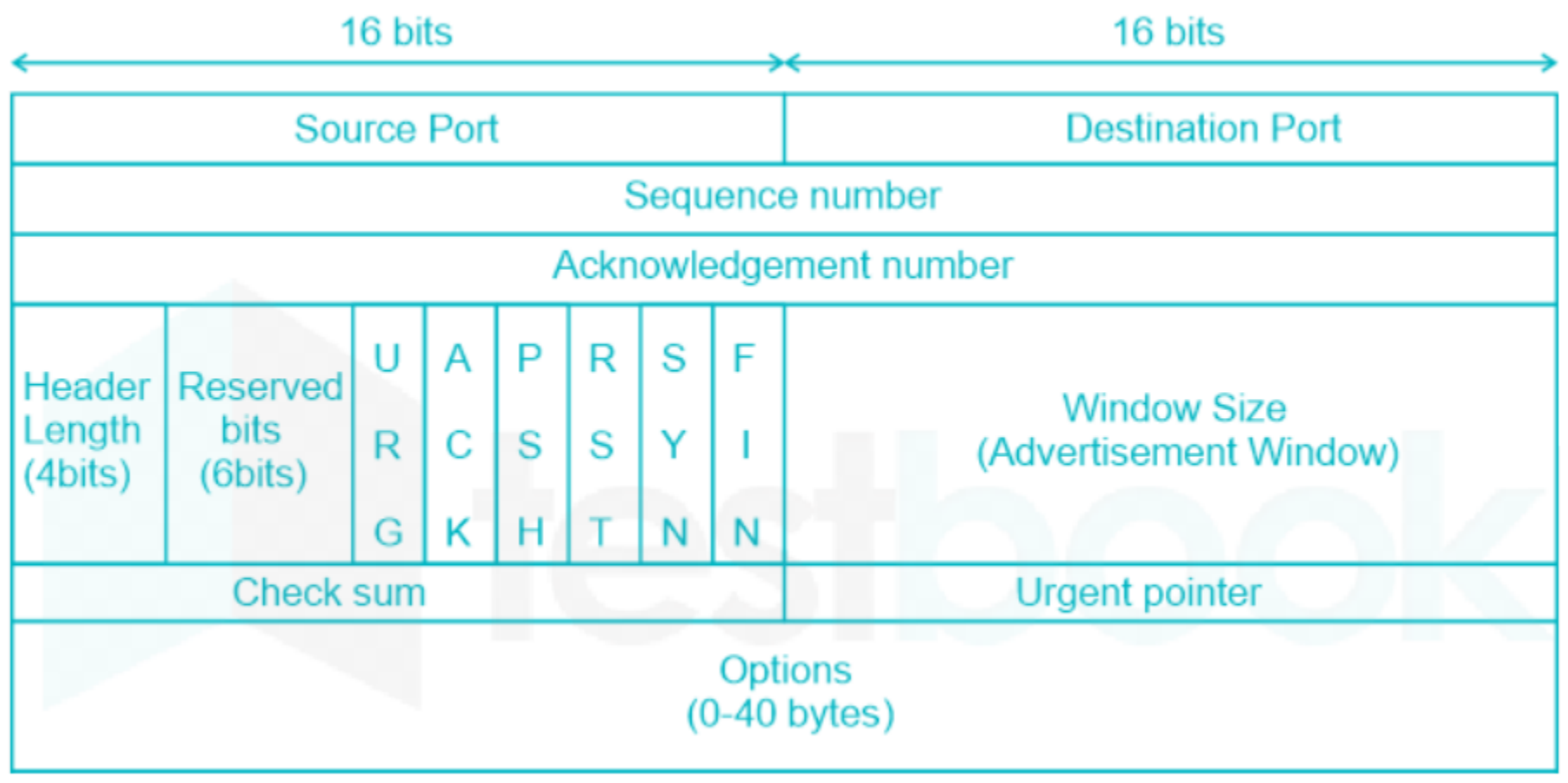
Các ứng dụng sử dụng TCP

- TCP xử lý tất cả các tác vụ liên quan đến việc chia luồng dữ liệu thành các segment, cung cấp độ tin cậy, kiểm soát luồng dữ liệu và sắp xếp lại các segment



TCP - Định dạng tiêu đề

Tiêu đề TCP: 20 – 60 bytes



TCP - Định dạng tiêu đề

Field	Length	Description	Value
Source Port	16 bits	It identifies the port of the sending application	
Destination Port	16 bits	It identifies the port of the receiving application	
Sequence Number (Seq)	32 bits	TCP assigns a unique Seq to each byte of data (in TCP segment). This field contains the Seq of the first data byte.	
Acknowledgement Number (ACK)	32 bits	It contains Seq of the data byte that receiver expects to receive next from the sender. It is always sequence number of the last received data byte incremented by 1	
Header Length	4 bits	It contains the length of TCP header. It helps in knowing from where the actual data begins.	from 5 to 15 (scaling factor is 4) (size of header: 20 - 60 bytes)
Reserved Bits	6 bits	They are reserved. These bits are not used	
URG Bit	1 bit	URG bit is used to treat certain data on an urgent basis If it's 1, It indicates the receiver that certain amount of data within the current segment is urgent	URG bit can be set to 0 or 1
ACK Bit	1 bit	It indicates whether ACK number field is valid or not. If it's 1, it indicates that ACK number contained in the TCP header is valid.	ACK bit can be set to 0 or 1
PSH Bit	1 bit	It is used to push the entire buffer immediately to the receiving application When it is 1, all the segments in the buffer are immediately pushed to the receiving application.	PSH bit can be set to 0 or 1
RST Bit	1 bit	RST bit is used to reset the TCP connection. When RST bit is set to 1, tt indicates the receiver to terminate the connection immediately.	RST bit can be set to 0 or 1
SYN Bit	1 bit	It is used to synchronize the Seq. If it's 1, it indicates the receiver that the Seq contained in the TCP header is the initial Seq.	SYN bit can be set to 0 or 1
FIN Bit	1 bit	FIN bit is used to terminate the TCP connection. If it's 1, it indicates the receiver that the sender wants to terminate the connection.	FIN bit can be set to 0 or 1
Window Size	16 bits	The window size that changes dynamically during data transmission is used for Flow Control It contains the size of the receiving window of the sender. It advertises how much data (in bytes) the sender can receive without ACK.	
Checksum	16 bits	It verifies the integrity of data in the TCP payload (for error control) Sender adds CRC checksum to the checksum field before sending the data. Receiver rejects the data that fails the CRC check.	
Urgent Pointer	16 bits	It indicates how much data in the current segment counting from the first data byte is urgent. Urgent pointer added to the SeqN indicates the end of urgent data byte. The field is considered valid and evaluated only if the URG bit is 1.	
Options	0-40 bytes	It is generally used for the following purposes: Time stamp + Window size extension + Parameter negotiation + Padding	

TCP – Số hiệu cổng thông dụng

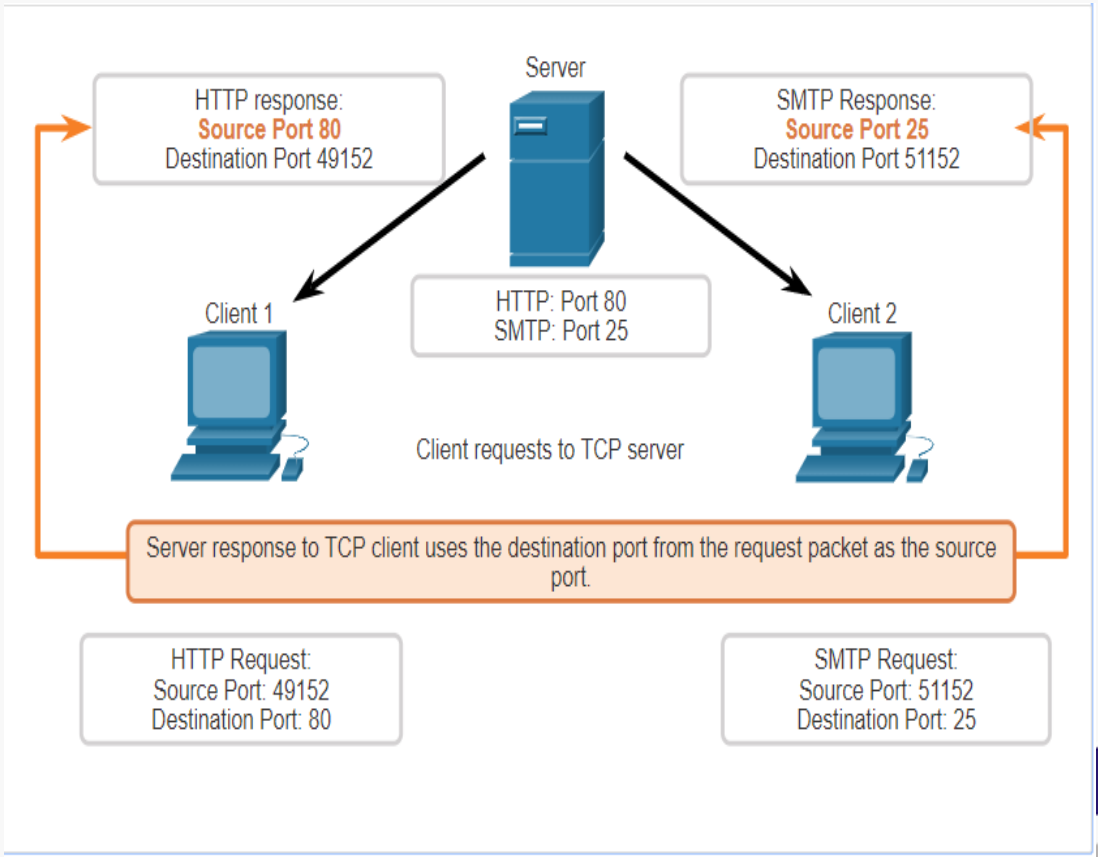
PORT NUMBER	TRANSPORT PROTOCOL	SERVICE NAME	RFC
20, 21	TCP	File Transfer Protocol (FTP)	RFC 959
22	TCP and UDP	Secure Shell (SSH)	RFC 4250-4256
23	TCP	Telnet	RFC 854
25	TCP	Simple Mail Transfer Protocol (SMTP)	RFC 5321
53	TCP and UDP	Domain Name Server (DNS)	RFC 1034-1035
80	TCP	HyperText Transfer Protocol (HTTP)	RFC 2616
110	TCP	Post Office Protocol (POP3)	RFC 1939
143	TCP and UDP	Internet Message Access Protocol (IMAP4)	RFC 3501
161, 162	TCP and UDP	Simple Network Management Protocol (SNMP)	RFC 1901-1908, 3411-3418
179	TCP	Border Gateway Protocol (BGP)	RFC 4271
389	TCP and UDP	Lightweight Directory Access Protocol	RFC 4510
443	TCP and UDP	HTTP with Secure Sockets Layer (SSL)	RFC 2818
636	TCP and UDP	Lightweight Directory Access Protocol over TLS/SSL (LDAPS)	RFC 4513
989/990	TCP	FTP over TLS/SSL	RFC 4217

TCP - Truyền thông

Tiến trình xử lý tại máy chủ/Server:

Mỗi tiến trình ứng dụng chạy trên máy chủ được cấu hình để sử dụng một số hiệu cổng.

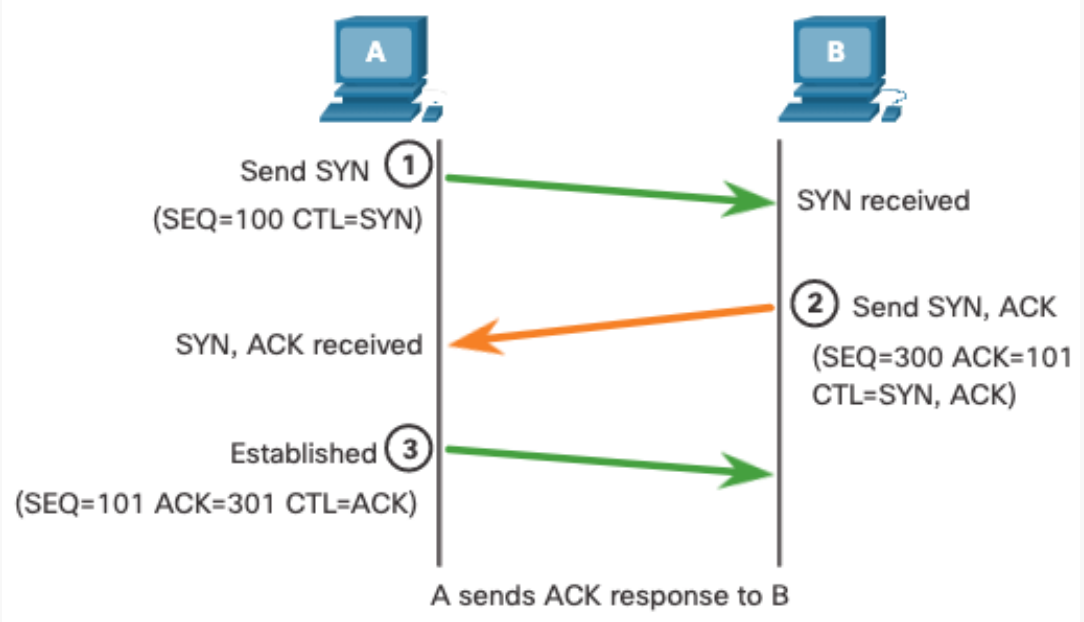
- Một máy chủ riêng lẻ không thể có hai dịch vụ được gán cho cùng một số hiệu cổng trong cùng một dịch vụ tầng giao vận.
- Một ứng dụng máy chủ đang hoạt động được gán cho một cổng cụ thể được coi là mở, điều đó có nghĩa là tầng giao vận chấp nhận và xử lý các segment được gửi đến cổng đó.
- Mọi yêu cầu từ máy khách gửi đến đúng socket đều được chấp nhận và dữ liệu được gửi đến ứng dụng máy chủ.



TCP - Truyền thông

Thiết lập (hay khởi tạo) phiên kết nối

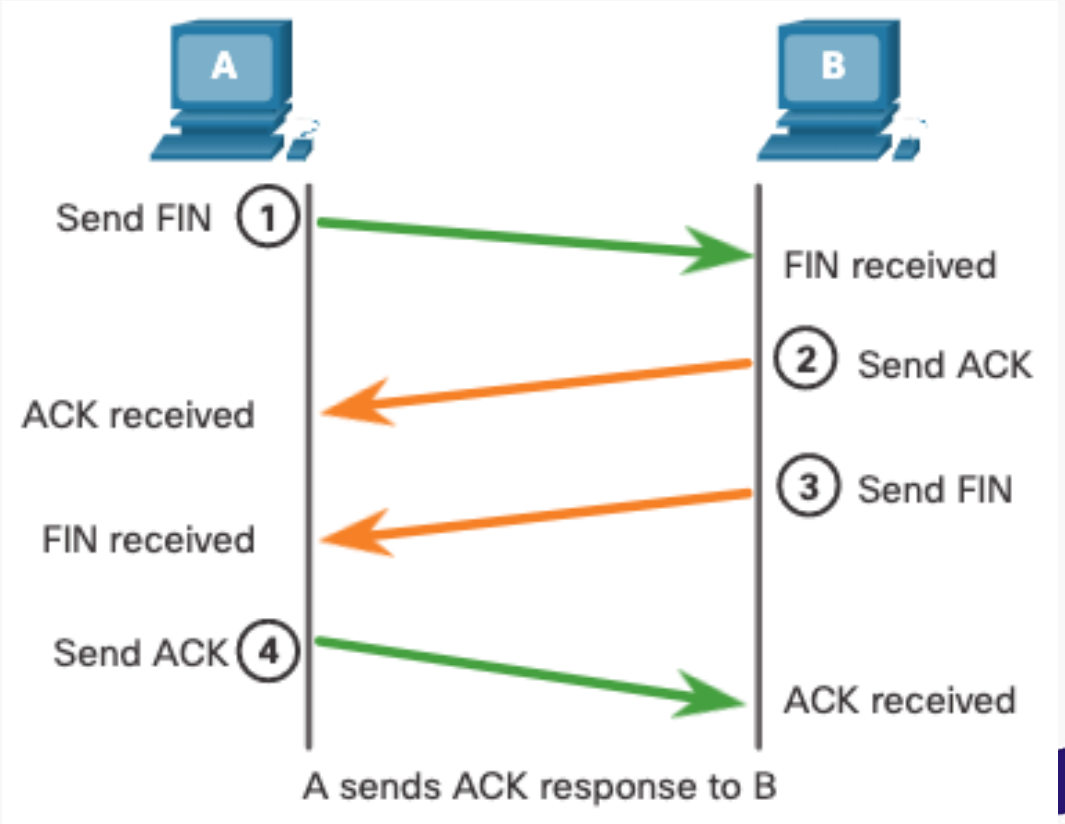
- Bước 1: Client khởi tạo phiên kết nối với Server (mô hình client-server)
- Bước 2: Server xác nhận phiên kết nối giữa Client với Server và phản hồi chấp nhận yêu cầu kết nối.
- Bước 3: Client bắt đầu phiên truyền thông với Server



TCP - Truyền thông

Kết thúc (hay đóng) phiên kết nối

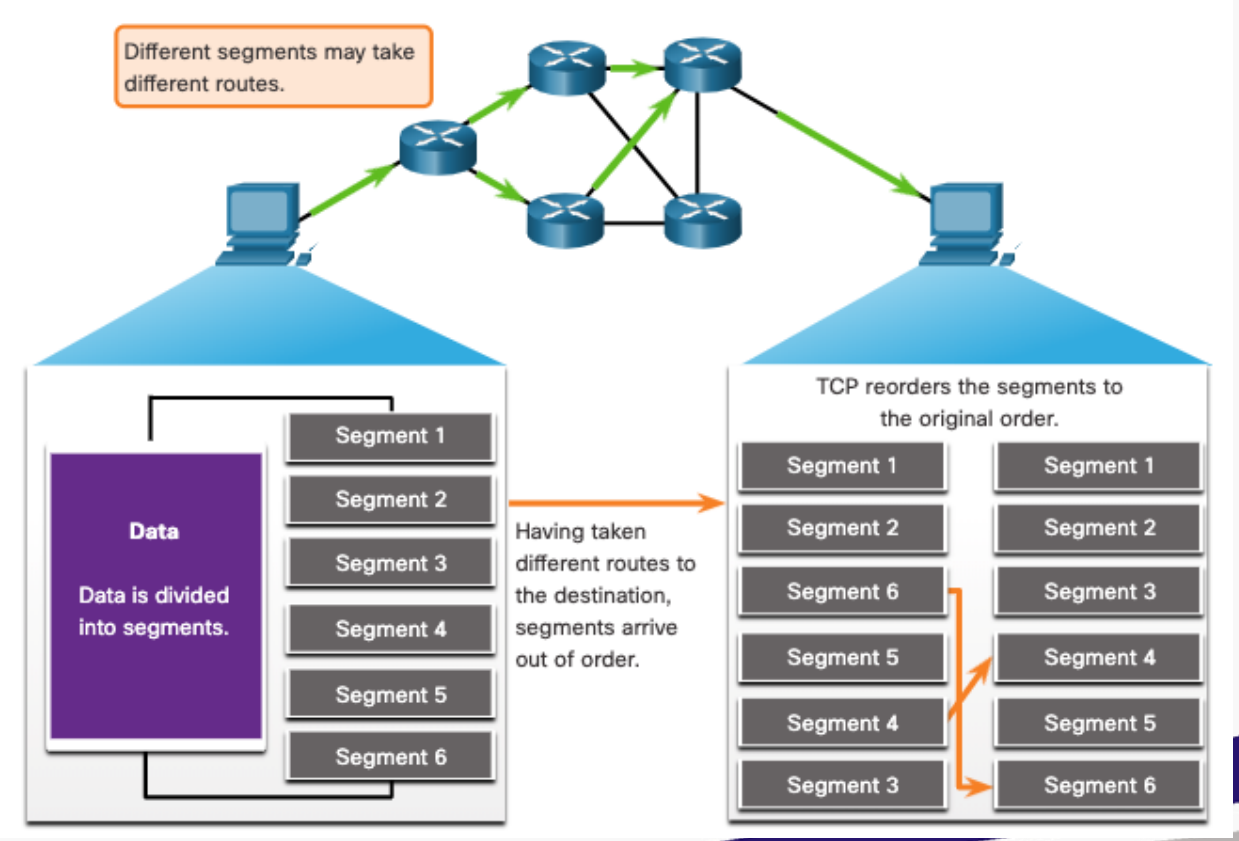
- Bước 1: Khi Client không còn dữ liệu để gửi, nó sẽ gửi một segment có gán cờ FIN
- Bước 2: Server gửi lại ACK để xác nhận đã nhận được FIN để chấm dứt phiên kết nối
- Bước 3: Server gửi FIN để đóng phiên kết nối này
- Bước 4: Client gửi lại ACK để xác nhận đã nhận được FIN từ Server



TCP - Điều khiển luồng và đảm bảo độ tin cậy

Truyền theo thứ tự và đảm bảo độ tin cậy:

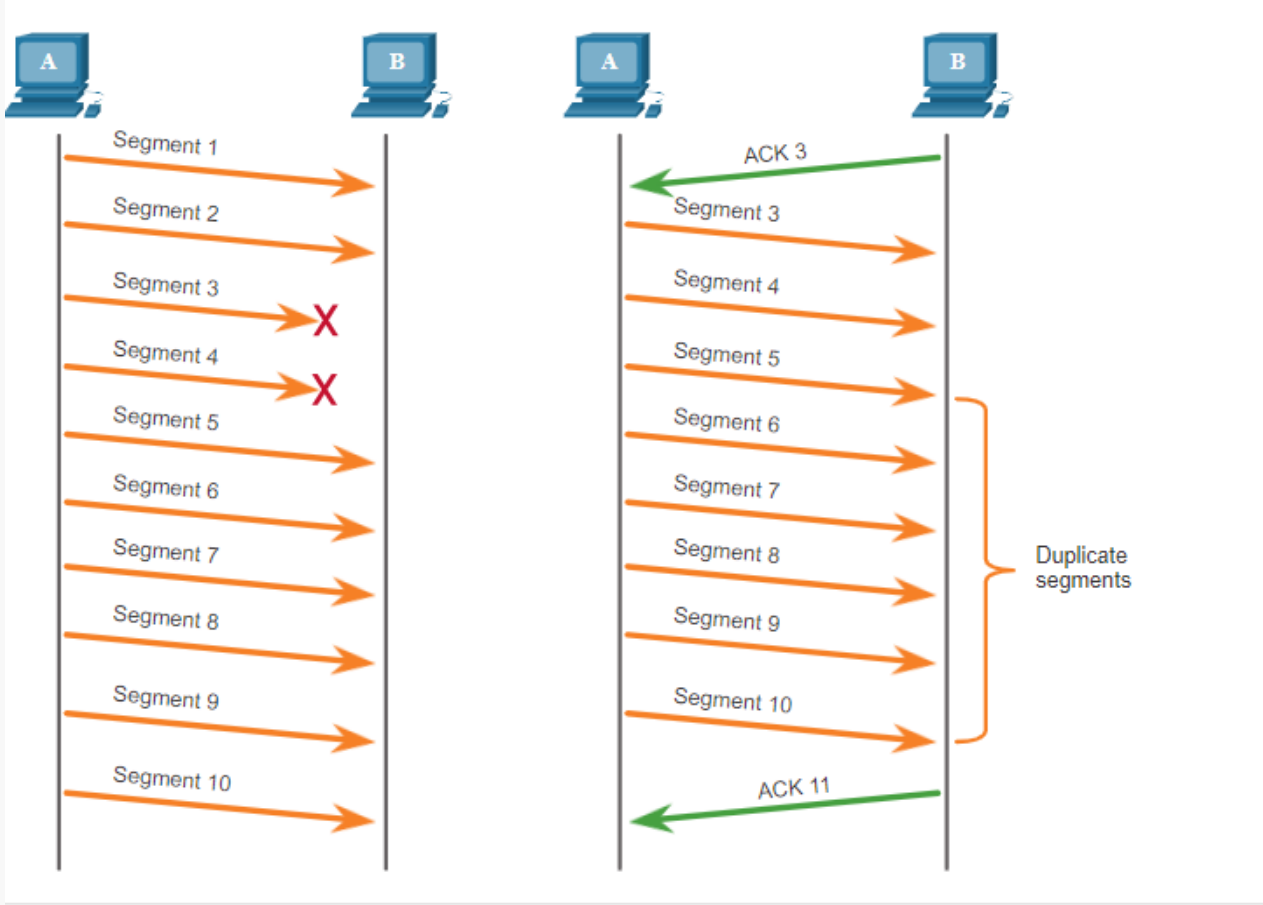
- TCP cũng giúp duy trì luồng để các thiết bị không bị quá tải.
- Có thể đôi khi các segment TCP không đến đích hoặc không đúng thứ tự.
- Tất cả dữ liệu phải được nhận và dữ liệu trong các segment phải được sắp xếp theo số thứ tự gốc
- Số thứ tự được gán trong mỗi tiêu đề của mỗi dữ liệu để kiểm soát vấn đề này



TCP - Điều khiển luồng và đảm bảo độ tin cậy

Đảm bảo độ tin cậy - Mất dữ liệu và truyền lại

- Cho dù mạng được thiết kế tốt đến đâu thì việc mất dữ liệu đôi khi vẫn xảy ra.
- TCP cung cấp các cách thức quản lý các segment bị mất. Một trong số các giải pháp là gửi lại các segments không được bên nhận gửi ACK xác nhận là đã nhận được.
- Gần đây, một tính năng tuỳ chọn được gọi là xác nhận có chọn lọc (SACK - selective acknowledgment) thường sử dụng trong quá trình thiết lập phiên TCP



TCP – Điều khiển luồng và đảm bảo độ tin cậy

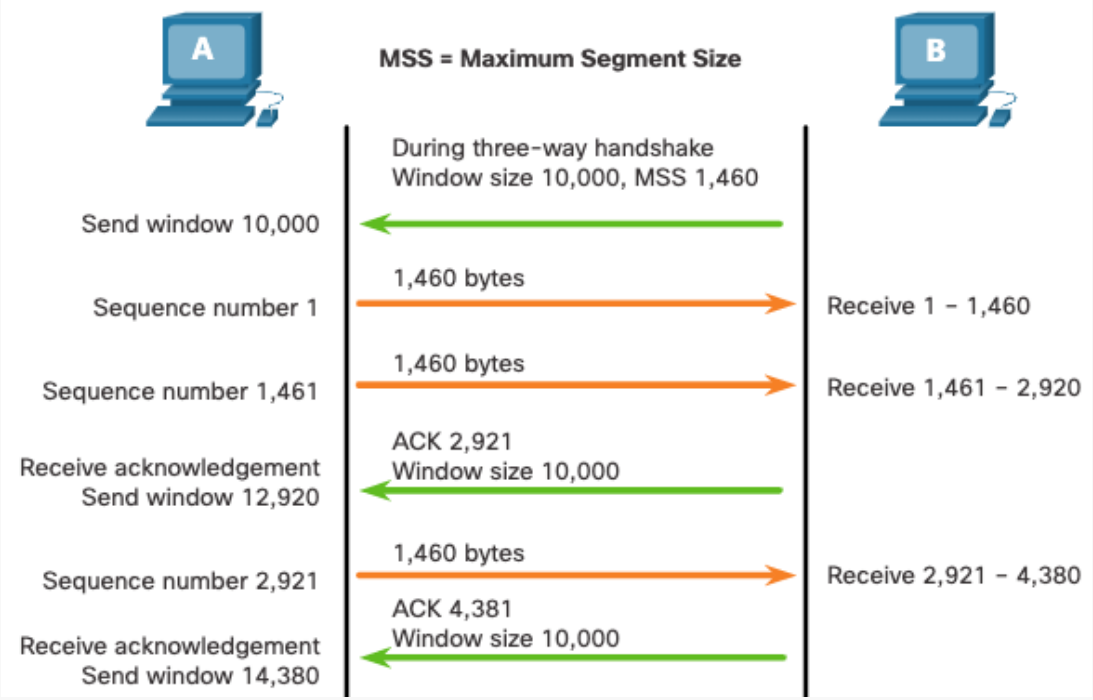
Điều khiển luồng – Kích thước cửa sổ và ACK

TCP cung cấp các cơ chế điều khiển luồng như sau:

- Điều khiển luồng là lượng dữ liệu mà máy đích có thể nhận và xử lý một cách đáng tin cậy.
- Điều khiển luồng giúp duy trì độ tin cậy của việc truyền TCP bằng cách điều chỉnh tốc độ luồng dữ liệu giữa nguồn và đích trong một phiên nhất định.

Ghi chú:

- TCP sử dụng kỹ thuật kích thước cửa sổ hay cửa sổ trượt để điều khiển luồng

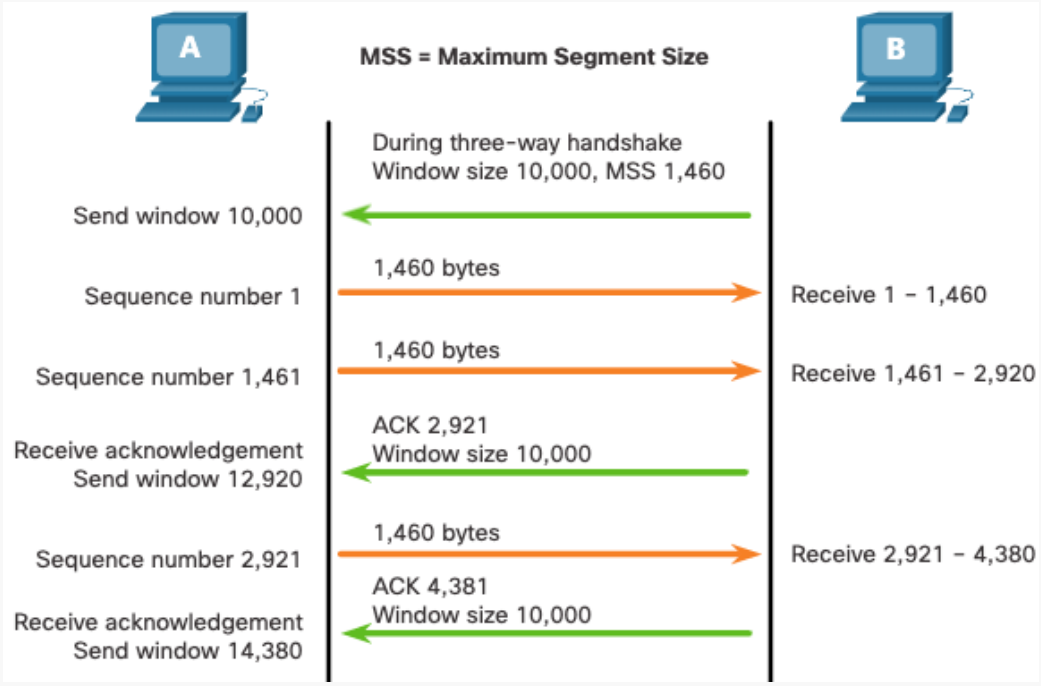


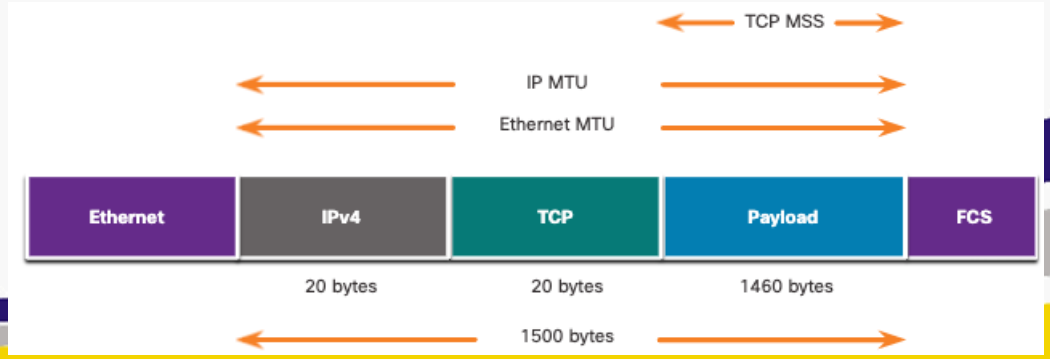
TCP – Điều khiển luồng và đảm bảo độ tin cậy

Điều khiển luồng - Kích thước Segment tối đa (MSS)

MSS (Maximum Segment Size) là lượng dữ liệu tối đa mà thiết bị đích có thể nhận được.

- Thường, MSS là 1.460 byte khi sử dụng IPv4.
- Thiết bị xác định giá trị của trường MSS bằng cách trừ các tiêu đề IP và TCP khỏi đơn vị truyền tối đa Ethernet (Maximum Segment Size), mặc định là 1500 byte (Ethernet MTU).
- 1500byte bao gồm 40byte (20 byte cho tiêu đề IPv4 và 20 byte cho tiêu đề TCP) còn lại 1460 byte.

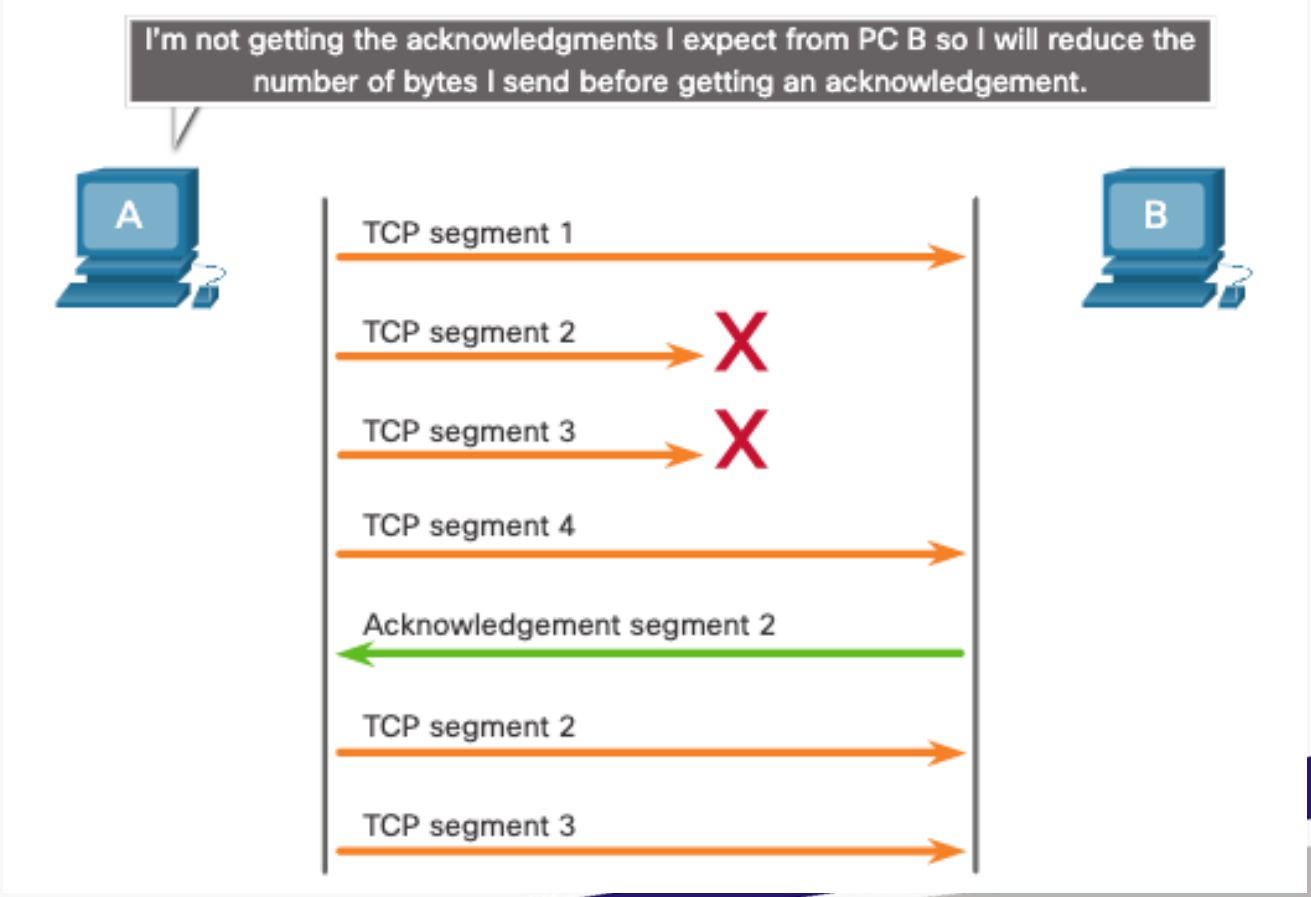




TCP – Điều khiển luồng và đảm bảo độ tin cậy

Điều khiển luồng - Tránh tắc nghẽn

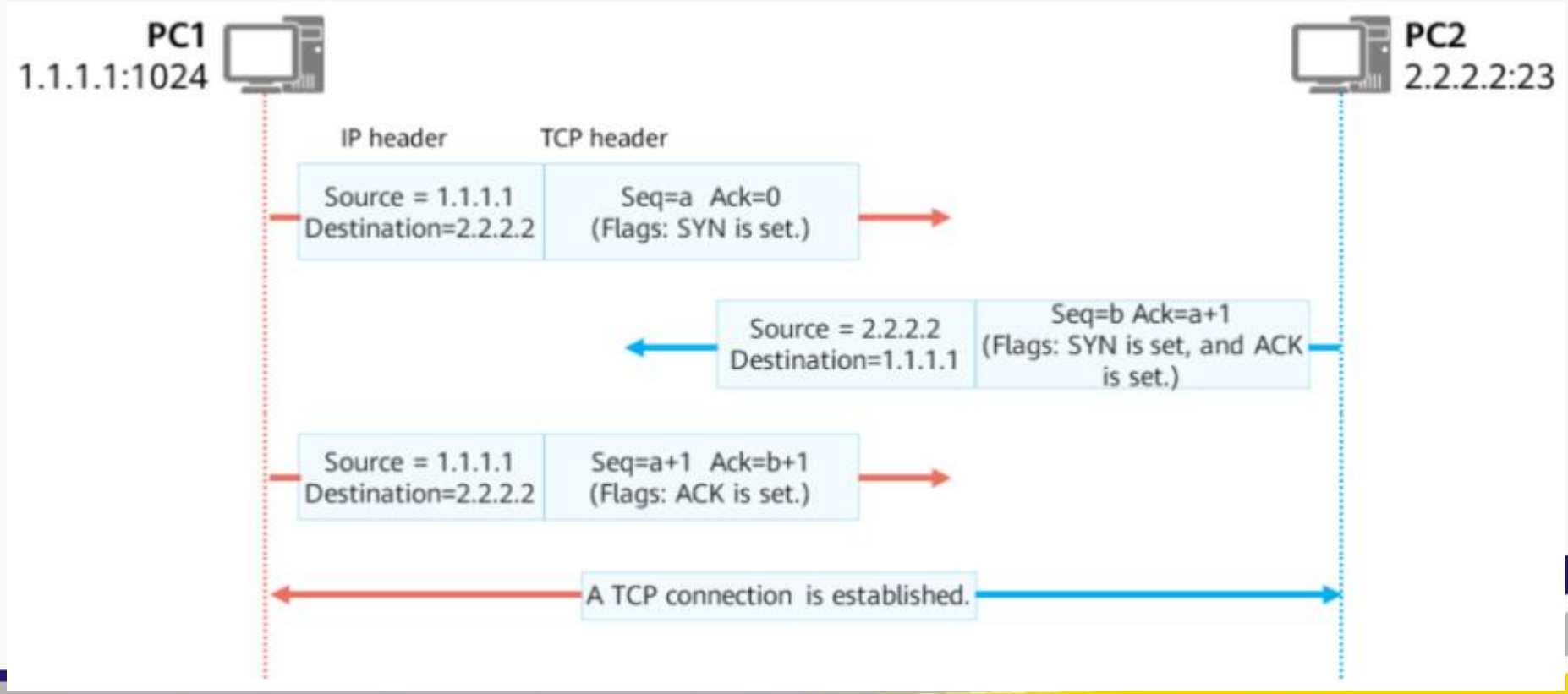
- Khi tắc nghẽn xảy ra, các gói tin sẽ bị huỷ bởi các bộ định tuyến bị quá tải
- Để tránh và kiểm soát tắc nghẽn, TCP sử dụng một số cơ chế, bộ định thời và thuật toán xử lý tắc nghẽn.



TCP - Ví dụ về thiết lập phiên

Ví dụ về quá trình thiết lập phiên kết nối TCP

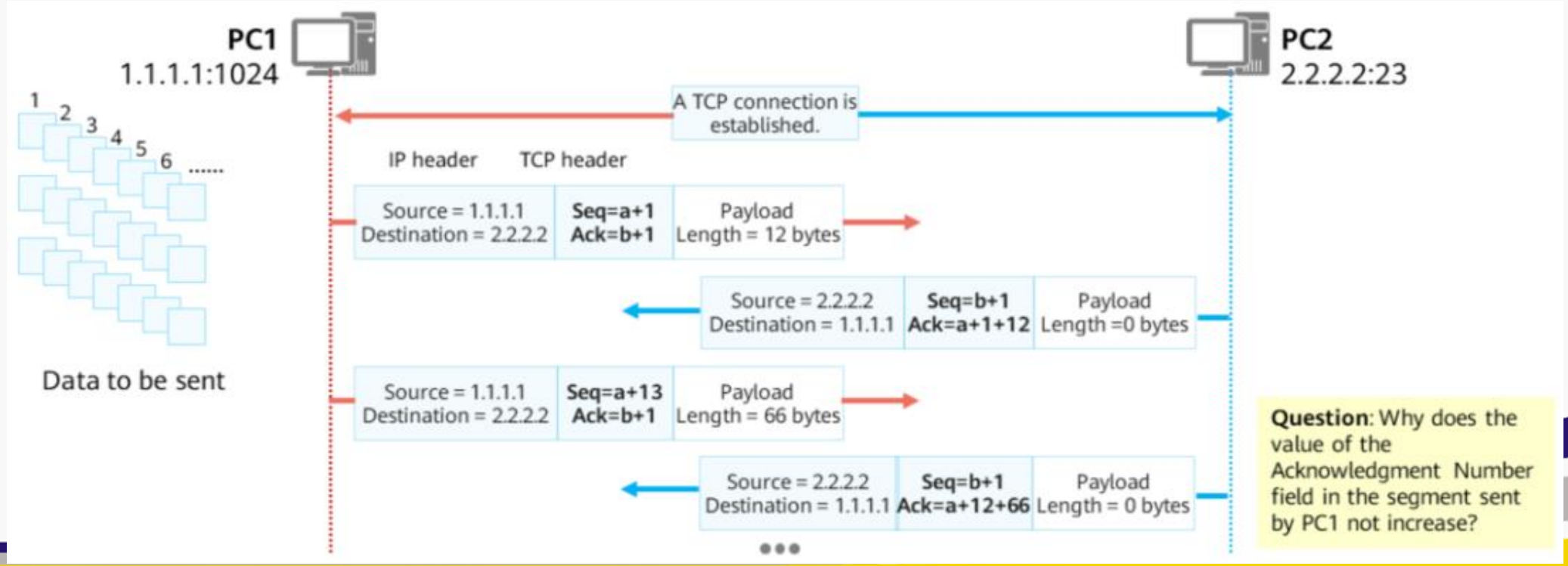
- Trước khi gửi dữ liệu, các thiết bị thiết lập kết nối thông qua quá trình bắt tay ba bước.



TCP - Ví dụ về truyền tin tin cậy

Ví dụ về truyền tin tin cậy

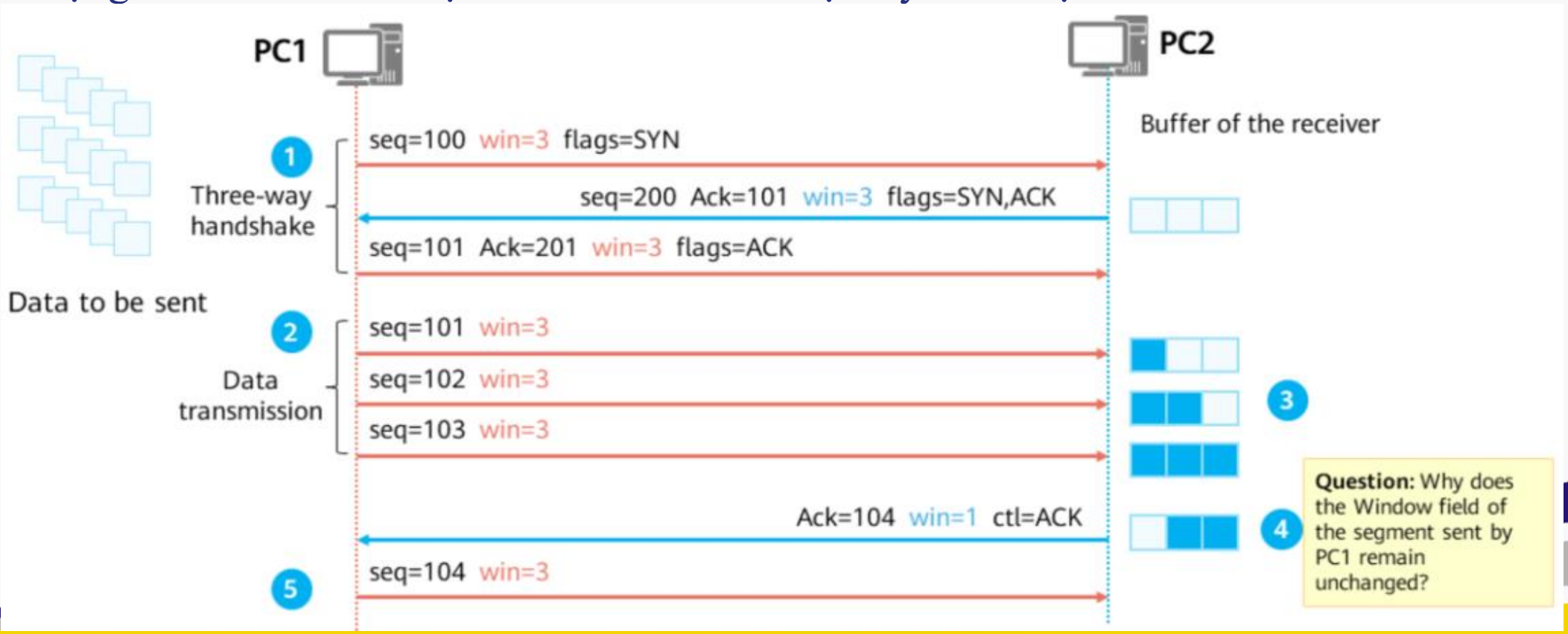
- Số thứ tự và số ACK được sử dụng để thực hiện truyền dữ liệu theo thứ tự và đảm bảo tin cậy trong TCP



TCP - Ví dụ về điều khiển luồng

Ví dụ về điều khiển luồng (Kỹ thuật cửa sổ trượt)

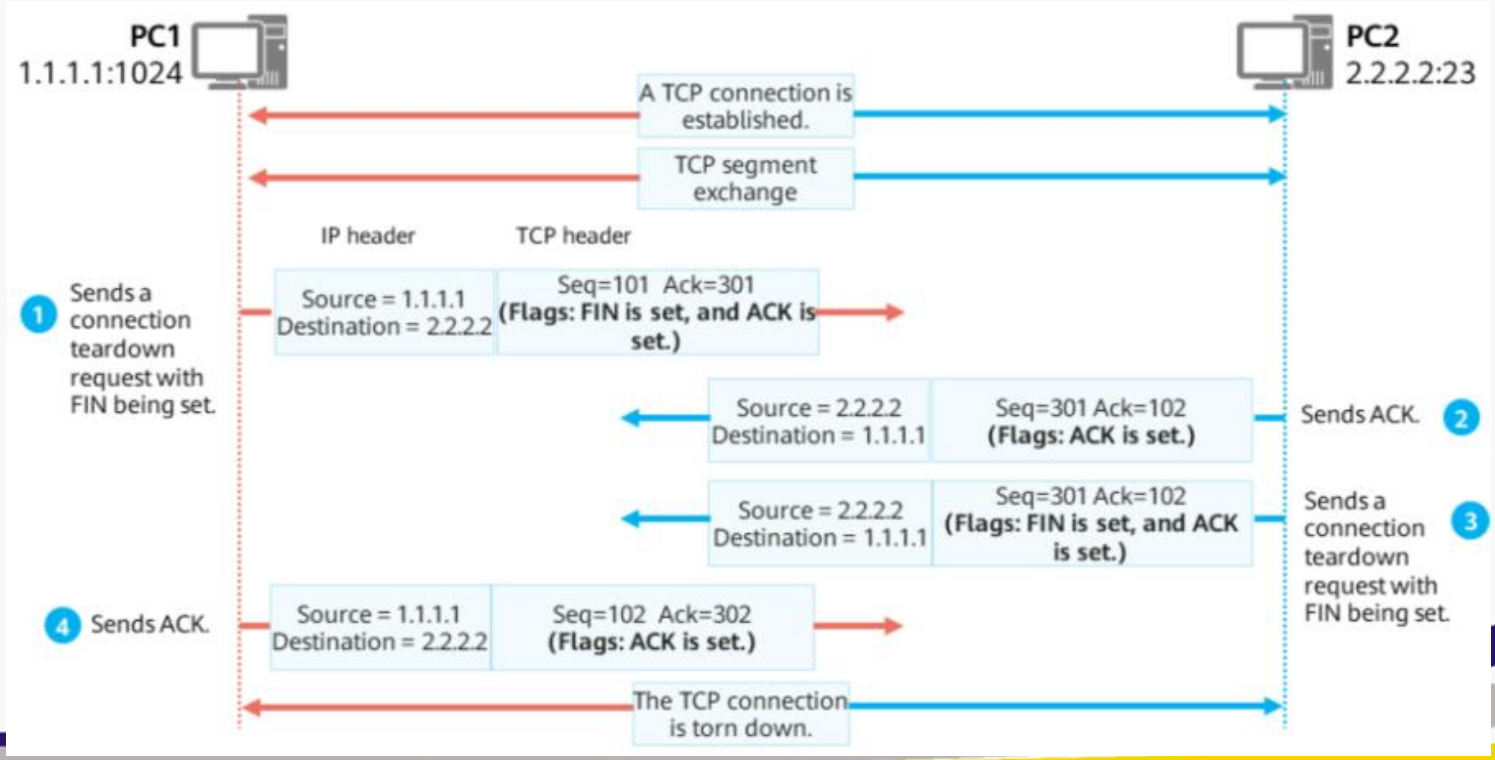
- TCP sử dụng cơ chế cửa sổ trượt để điều khiển tốc độ truyền dữ liệu



TCP - Ví dụ về đóng phiên kết nối

Ví dụ về quá trình đóng phiên kết nối (Bắt tay bốn bước)

- Sau khi truyền dữ liệu xong, TCP cần sử dụng cơ chế bắt tay bốn bước để ngắt kết nối TCP và giải phóng tài nguyên hệ thống



Trao đổi và Thảo luận