CMC UNIVERSITY

Computer networks and communications

Chap 5: Transport layer

Lecturer: Nguyen Duc An

CMC UNIVERSITY

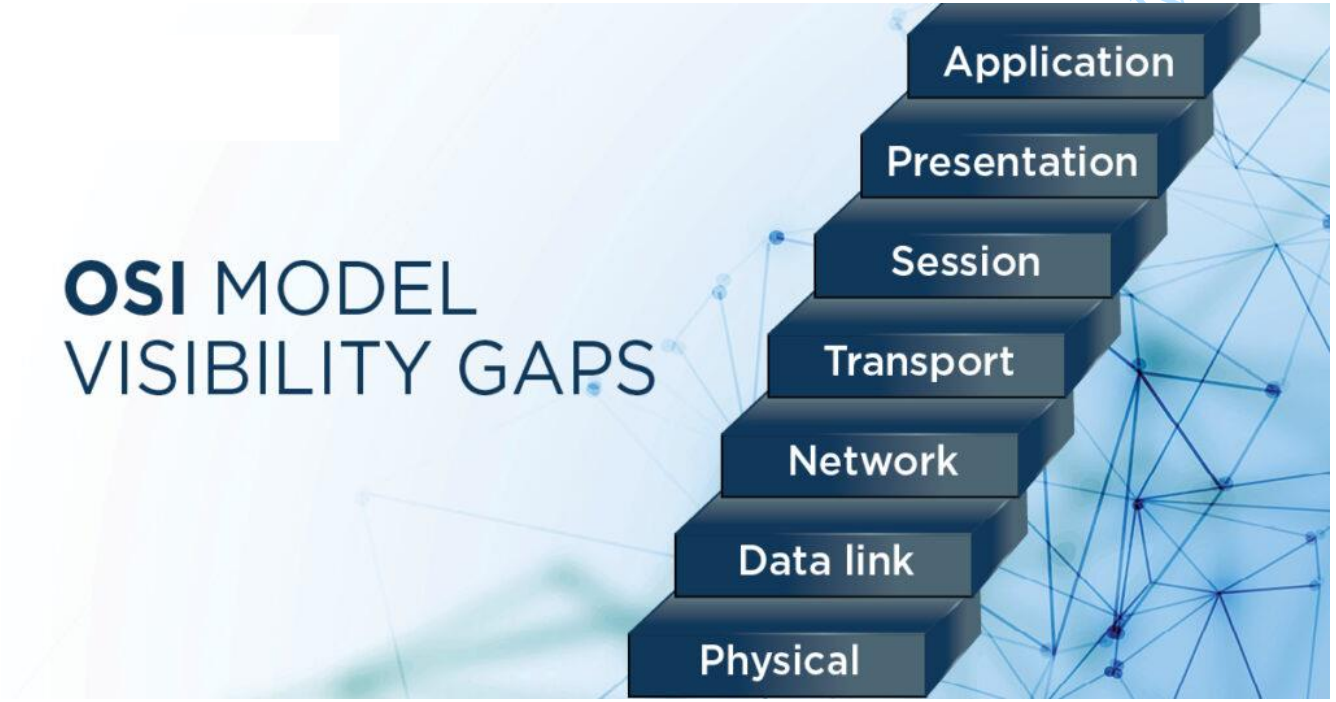
Nội dung

5.1 The transport services

5.2 Transport layer functions

5.3 UDP protocol

5.4 TCP protocol



5.3 Giao thức gói dữ liệu bên dùng-User Datagram Protocol

Giao thức user datagram (UDP) là một giao thức truyền tải không kết nối và không đáng tin cậy. Nó cung cấp phương tiện kết nối từ tiến trình (process)đến tiến trình thay vì kết nối từ host đến host.

- **Well-Known Ports cho UDP**
- **User Datagram**
- **Checksum**
- **Hoạt động của UDP**
- **Sử dụng UDP**

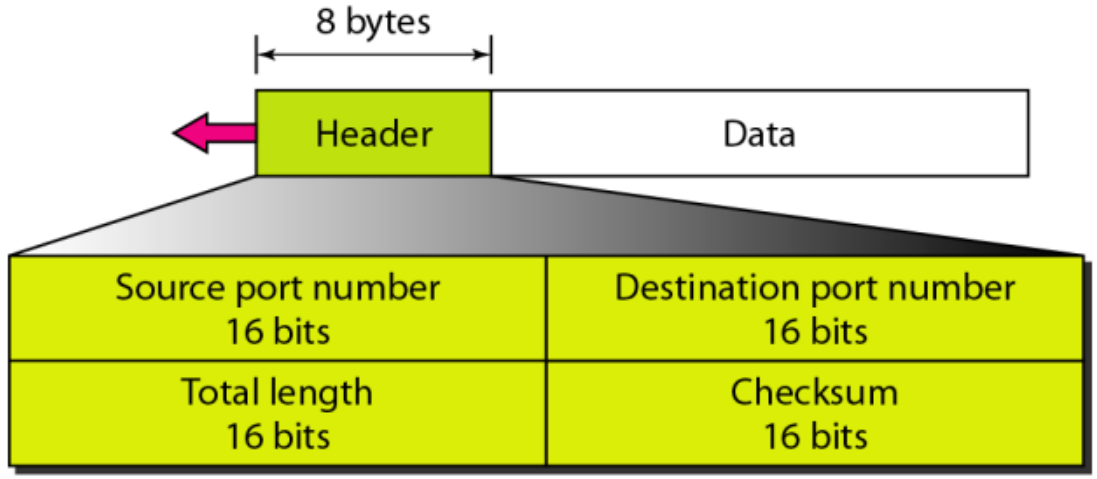
5.3 User Datagram Protocol

- **Well-Known Ports for UDP**

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

5.3 User Datagram Protocol

- **User Datagram**



5.3 User Datagram Protocol

- **Checksum**

TUỲ CHỌN, NẾU KHÔNG SỬ DỤNG, ĐẶT TẤT CẢ 1 MẶC ĐỊNH

Việc tính toán checksum của UDP khác với IP và ICMP. Ở đây, checksum bao gồm ba phần: một tiêu đề giả (pseudo-header), tiêu đề UDP, và dữ liệu đến từ lớp ứng dụng.

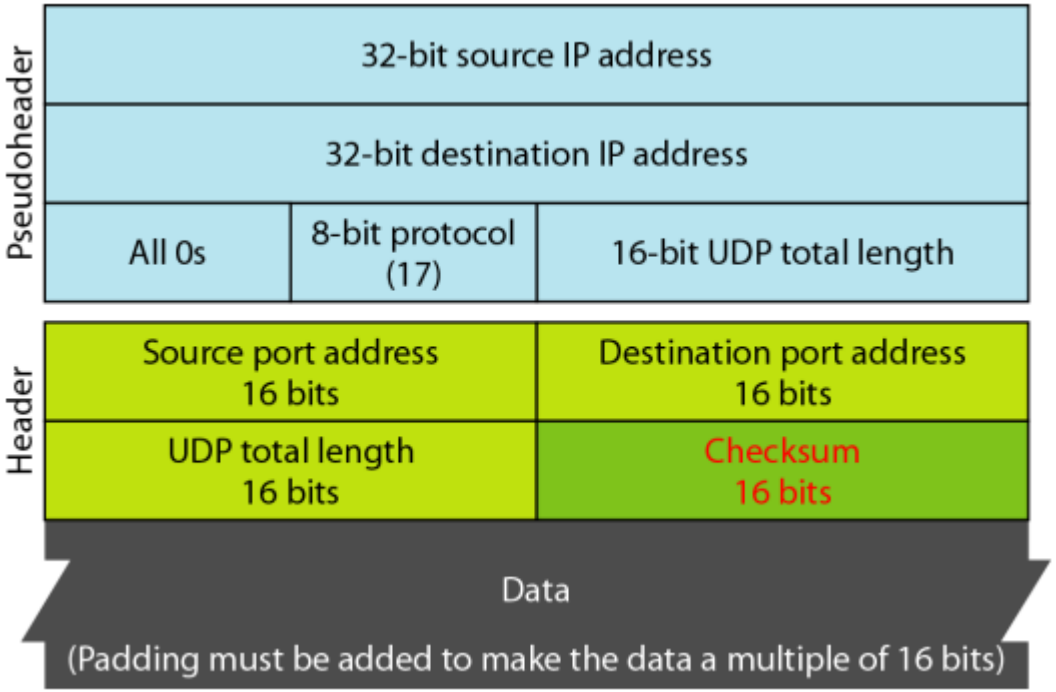
Tiêu đề giả là một phần của tiêu đề gói IP trong đó user datagram sẽ được đóng gói với một số trường được điền bằng các số 0.

Trường giao thức được thêm vào để đảm bảo rằng gói thuộc về UDP và không phải các giao thức lớp truyền tải khác.

5.3 User Datagram Protocol

- **Checksum**

OPTIONAL, IF NOT USED, SET ALL 1'S DEFAULT



5.3 User Datagram Protocol

- **Cơ chế hoạt động của UDP**

Dịch vụ không kết nối. UDP cung cấp dịch vụ không kết nối**.** Điều này có nghĩa là mỗi user datagram được gửi bởi UDP là một datagram độc lập. Không có mối quan hệ nào giữa các user datagram khác nhau, ngay cả khi chúng đến từ hoặc kết thúc cùng một tiến trình nguồn.

Các user datagram không được đánh số. Ngoài ra, không có thiết lập hoặc kết thúc kết nối, như trong trường hợp của TCP. Điều này có nghĩa là mỗi datagram của bên dùng có thể di chuyển trên một đường dẫn khác nhau.

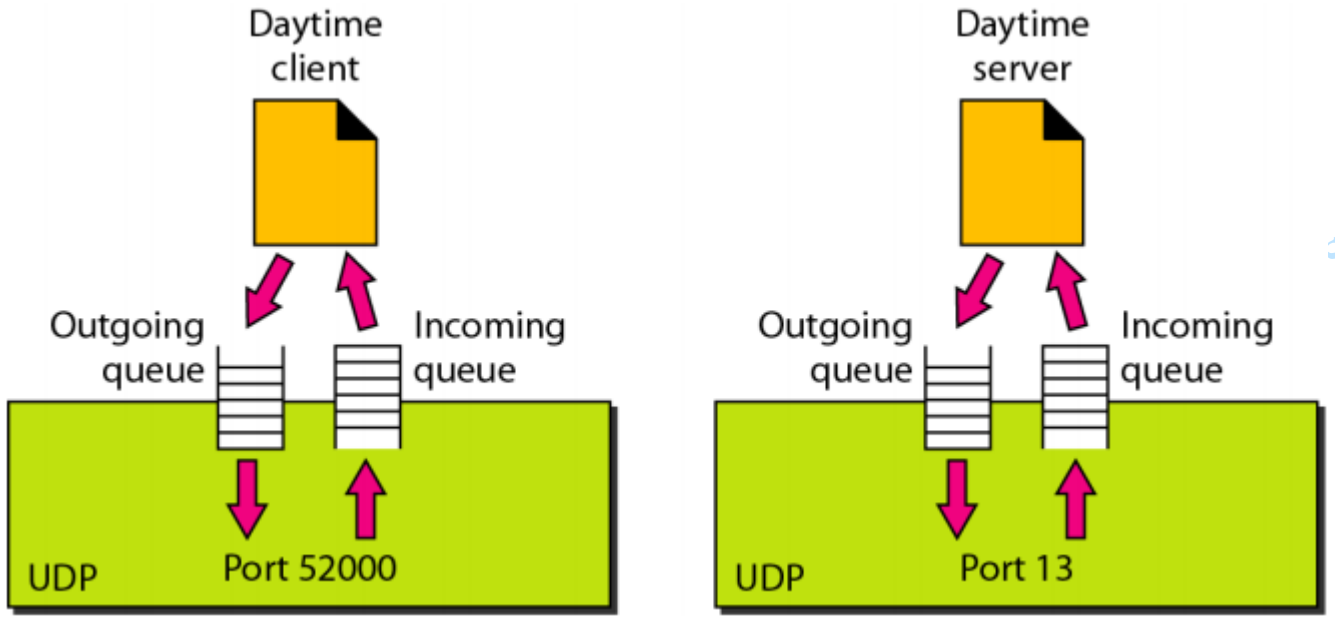
Kiểm soát lưu lượng và lỗi. UDP là một giao thức truyền tải rất đơn giản và không đáng tin cậy. Không có kiểm soát lưu lượng. Bộ nhận có thể bị tràn với các bản tin đến. Không có cơ chế kiểm soát lỗi trong UDP ngoại trừ kiểm tra tổng (checksum).

Đóng gói và mở gói. Giao thức UDP đóng gói và mở gói các bản tin trong một datagram IP để gửi một bản tin từ một tiến trình này đến tiến trình khác.

5.3 User Datagram Protocol

- **UDP Operation**

Hàng đợi trong UDP

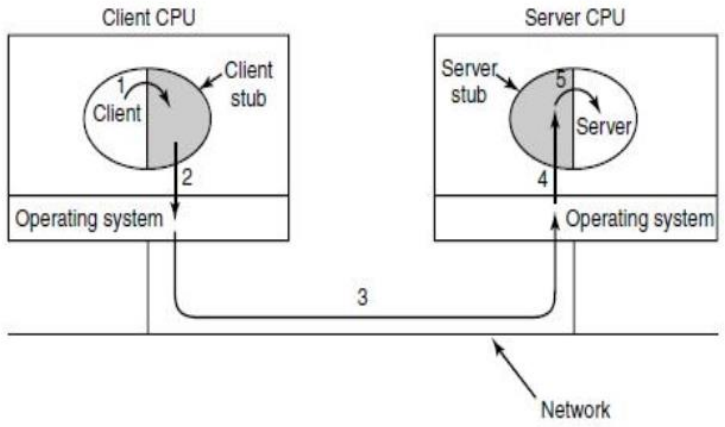


5.3 User Datagram Protocol

- **UDP Operation**

Cuộc gọi thủ tục từ xa-Remote Procedure Call

Remote Procedure Call (RPC) là một phương pháp cho phép một chương trình gọi một hàm hoặc thủ tục trên một máy chủ từ xa như thể nó đang gọi hàm đó trên máy của mình. RPC cho phép giao tiếp giữa các máy tính trong mạng mà không cần phải biết chi tiết về giao thức mạng.



5.3 User Datagram Protocol

- **UDP Operation**

Remote Procedure Call: Problems

Với RPC, việc truyền con trỏ là không khả thi vì client và server nằm trong các không gian địa chỉ khác nhau.

Về cơ bản, client stub không thể đóng gói các tham số: nó không có cách nào để xác định kích thước của chúng.

Một vấn đề thứ ba là không phải lúc nào cũng có thể suy ra kiểu của các tham số, ngay cả từ một đặc tả chính thức hoặc mã nguồn. (ví dụ: printf)

Vấn đề thứ tư liên quan đến việc sử dụng biến toàn cục. Thông thường, các thủ tục gọi và được gọi có thể giao tiếp bằng cách sử dụng biến toàn cục, bên cạnh việc giao tiếp qua các tham số. Nhưng nếu thủ tục được gọi được di chuyển đến một máy từ xa, mã nguồn sẽ gặp lỗi vì các biến toàn cục không còn được chia sẻ nữa.

5.4 Transmission Control Protocol

TCP là một giao thức định hướng kết nối; nó tạo ra một kết nối ảo giữa hai TCP để gửi dữ liệu. Ngoài ra, TCP sử dụng các cơ chế kiểm soát luồng và lỗi ở cấp độ truyền tải. Tóm lại, TCP được gọi là một giao thức truyền tải định hướng kết nối và đáng tin cậy. Nó bổ sung các tính năng định hướng kết nối và độ tin cậy vào các dịch vụ của IP.

- **TCP Services**
- **TCP Features**
- **Segment**
- **A TCP Connection**
- **Flow Control**
- **Error Control**

5.4 Transmission Control Protocol

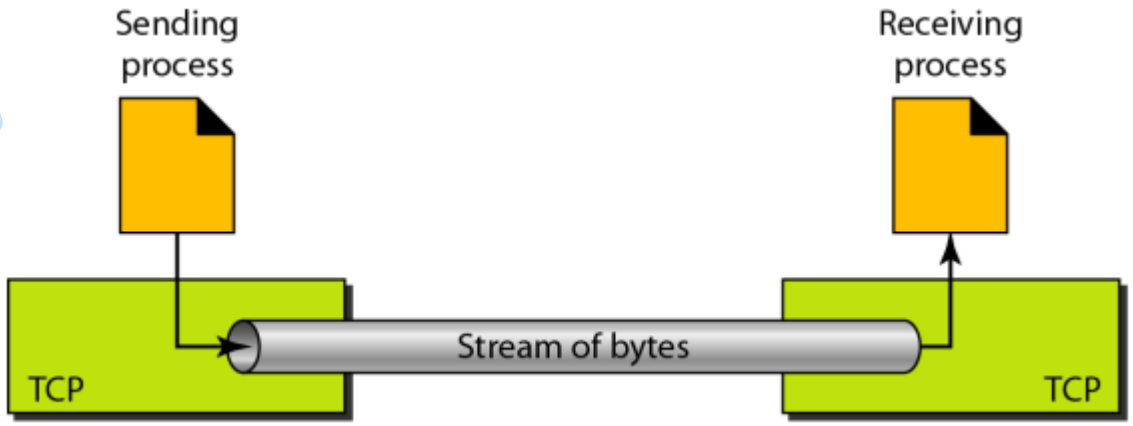
- **TCP Services: Process-to-Process Communication**

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

5.4 Transmission Control Protocol

- **TCP Services : Stream Delivery Service-Dịch vụ phân phối luồng**

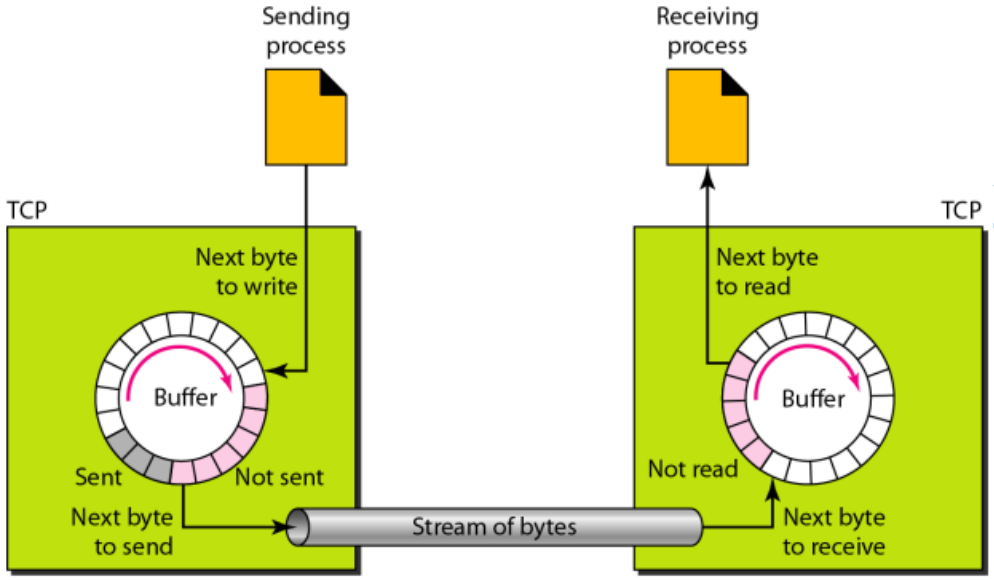
TCP tạo ra một môi trường trong đó hai tiến trình được kết nối bằng một "ống" ảo mang dữ liệu của chúng qua Internet. Môi trường ảo này được thể hiện trong hình dưới đây. Quá trình gửi tạo ra (ghi vào) dòng byte, và quá trình nhận tiêu thụ (đọc từ) chúng.



5.4 Transmission Control Protocol

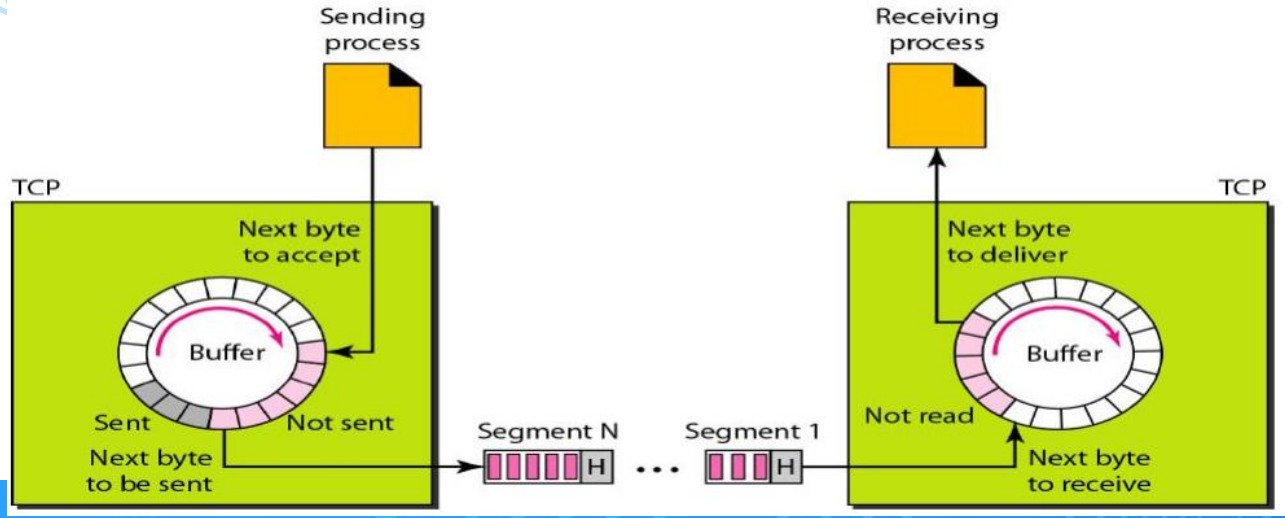
- **TCP Services : Sending and Receiving Buffers-Bộ đệm gửi và nhận**

Vì quá trình gửi và nhận có thể ghi hoặc đọc dữ liệu với tốc độ khác nhau, TCP cần các bộ đệm để lưu trữ. Có hai bộ đệm, bộ đệm gửi và bộ đệm nhận, mỗi bộ đệm dành cho một hướng. Một cách để triển khai bộ đệm là sử dụng một mảng vòng tròn các vị trí I-byte như được trình bày trong hình. Thông thường, các bộ đệm có kích thước hàng trăm hoặc hàng nghìn byte, tùy thuộc vào cách triển khai. Chúng tôi cũng cho thấy các bộ đệm có cùng kích thước, điều này không phải lúc nào cũng đúng.



5.4 Transmission Control Protocol

- **TCP Services: segments-segment**
- Tại lớp truyền tải, TCP nhóm một số byte lại với nhau thành một gói gọi là segment. TCP thêm một tiêu đề vào mỗi segment (để phục vụ mục đích điều khiển) và chuyển segment đó đến lớp IP để truyền tải. Các segment được bao bọc trong các datagram IP và được truyền đi.
- Các segment có thể được nhận không theo thứ tự, bị mất, hoặc bị hỏng và được gửi lại. Tất cả những điều này đều được TCP xử lý mà không cần quá trình nhận. Hình minh họa cách các segment được tạo ra từ các byte trong các bộ đệm.



5.4 Transmission Control Protocol

- **TCP Services: Full-Duplex Communication-Kết nối song công**
- TCP cung cấp dịch vụ hai chiều, trong đó dữ liệu có thể truyền theo cả hai hướng cùng một lúc. Mỗi TCP có bộ đệm gửi và một bộ đệm nhận, và các segment di chuyển theo cả hai hướng.
- **TCP Services: Connection-Oriented Service-Dịch vụ định hướng kết nối**
- TCP là một giao thức định hướng kết nối. Khi một tiến trình tại địa điểm A muốn gửi và nhận dữ liệu từ một tiến trình khác tại địa điểm B, các bước sau sẽ xảy ra:

  Hai TCP thiết lập một kết nối giữa chúng.

  Dữ liệu được trao đổi theo cả hai hướng.

  Kết nối được kết thúc.
- **TCP Services: Reliable Service-Dịch vụ đáng tin cậy**
- TCP là một giao thức truyền tải đáng tin cậy. Nó sử dụng cơ chế xác nhận để kiểm tra việc dữ liệu đã đến an toàn và nguyên vẹn.

5.4 Transmission Control Protocol

- **TCP Features: Numbering System-Hệ thống đánh số**

Có hai trường gọi là số hiệu tuần tự và số hiệu xác nhận. Hai trường này tham chiếu đến số byte chứ không phải số segment.

Số Byte: TCP đánh số các byte dữ liệu đang được truyền trong mỗi kết nối. Việc đánh số bắt đầu từ một số được tạo ngẫu nhiên. Ví dụ, nếu số ngẫu nhiên là 1057 và tổng số dữ liệu cần gửi là 6000 byte, các byte sẽ được đánh số từ 1057 đến 7056. Chúng ta sẽ thấy rằng việc đánh số byte được sử dụng để kiểm soát luồng và lỗi.

Số Hiệu Tuần Tự: Sau khi các byte đã được đánh số, TCP gán một số hiệu tuần tự cho mỗi segment đang được gửi. Số hiệu tuần tự của mỗi segment là số của byte đầu tiên được mang trong segment đó.

Số Hiệu Xác Nhận: Giá trị của trường xác nhận trong một segment định nghĩa số của byte tiếp theo mà bên nhận mong đợi nhận được. Số hiệu xác nhận là tổng hợp..

5.4 Transmission Control Protocol

TCP Features: Flow Control

TCP cung cấp kiểm soát luồng. Bên nhận dữ liệu sẽ kiểm soát lượng dữ liệu chuyển tới bên gửi. Điều này được thực hiện để ngăn bên nhận bị quá tải bởi dữ liệu. Hệ thống đánh số cho phép TCP sử dụng kiểm soát luồng dựa trên byte.

TCP Features: Error Control

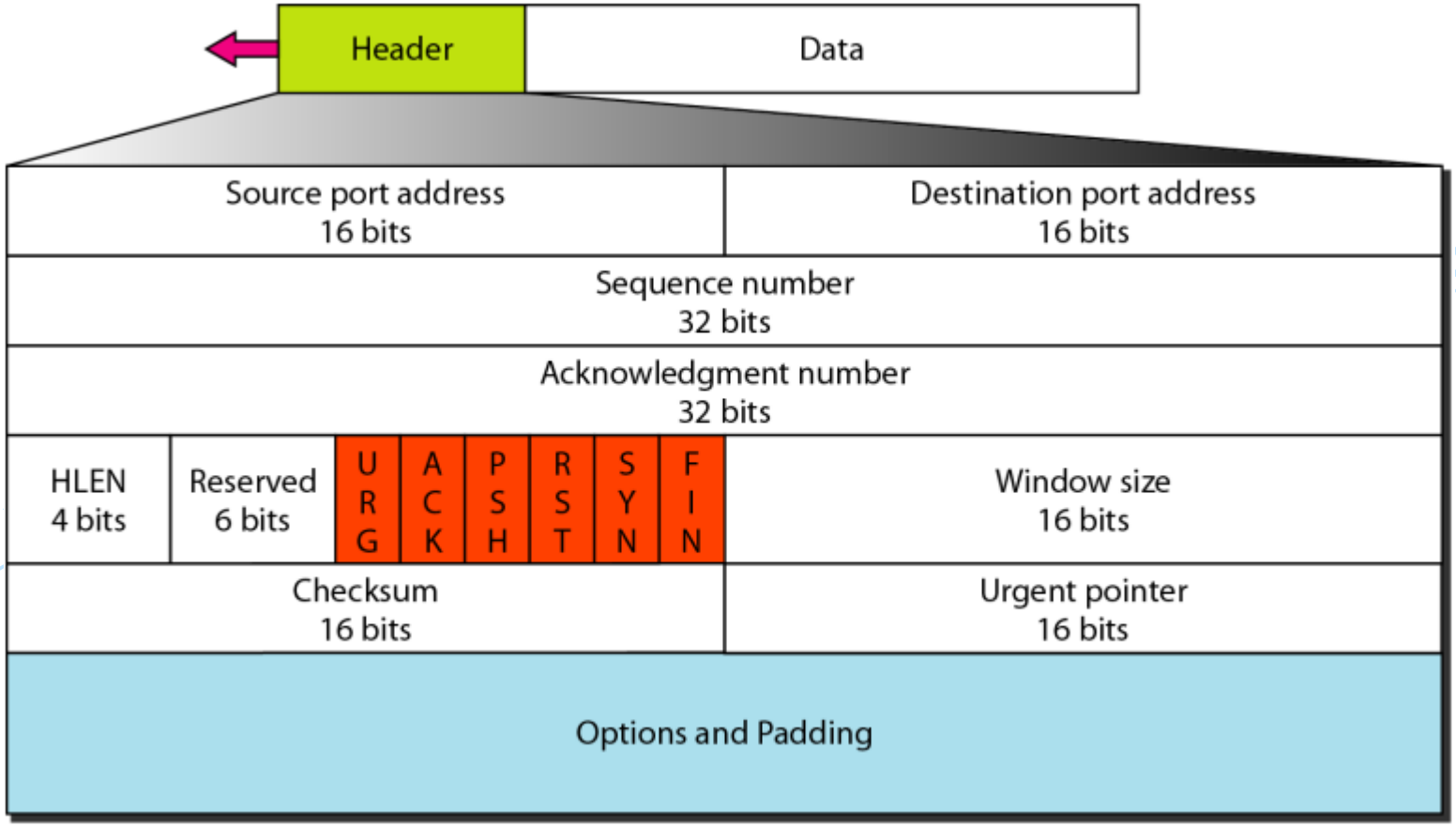
Để cung cấp dịch vụ tin cậy, TCP triển khai một cơ chế kiểm soát lỗi. Kiểm soát lỗi được định hướng theo byte. Kiểm soát lỗi xem một segment như là đơn vị dữ liệu để phát hiện lỗi (segment bị mất hoặc hỏng),

TCP Features: Congestion Control

TCP kiểm soát tình trạng tắc nghẽn trong mạng. Lượng dữ liệu gửi đi không chỉ được kiểm soát bởi bên nhận (kiểm soát luồng) mà còn được quyết định bởi mức độ tắc nghẽn trong mạng.

5.4 Transmission Control Protocol

TCP Segment format



5.4 Transmission Control Protocol

A TCP Connection

TCP là giao thức hướng kết nối. Một giao thức truyền tải hướng kết nối thiết lập một đường dẫn ảo giữa nguồn và đích. Tất cả các segment thuộc về một bản tin sẽ được gửi qua đường dẫn ảo này.

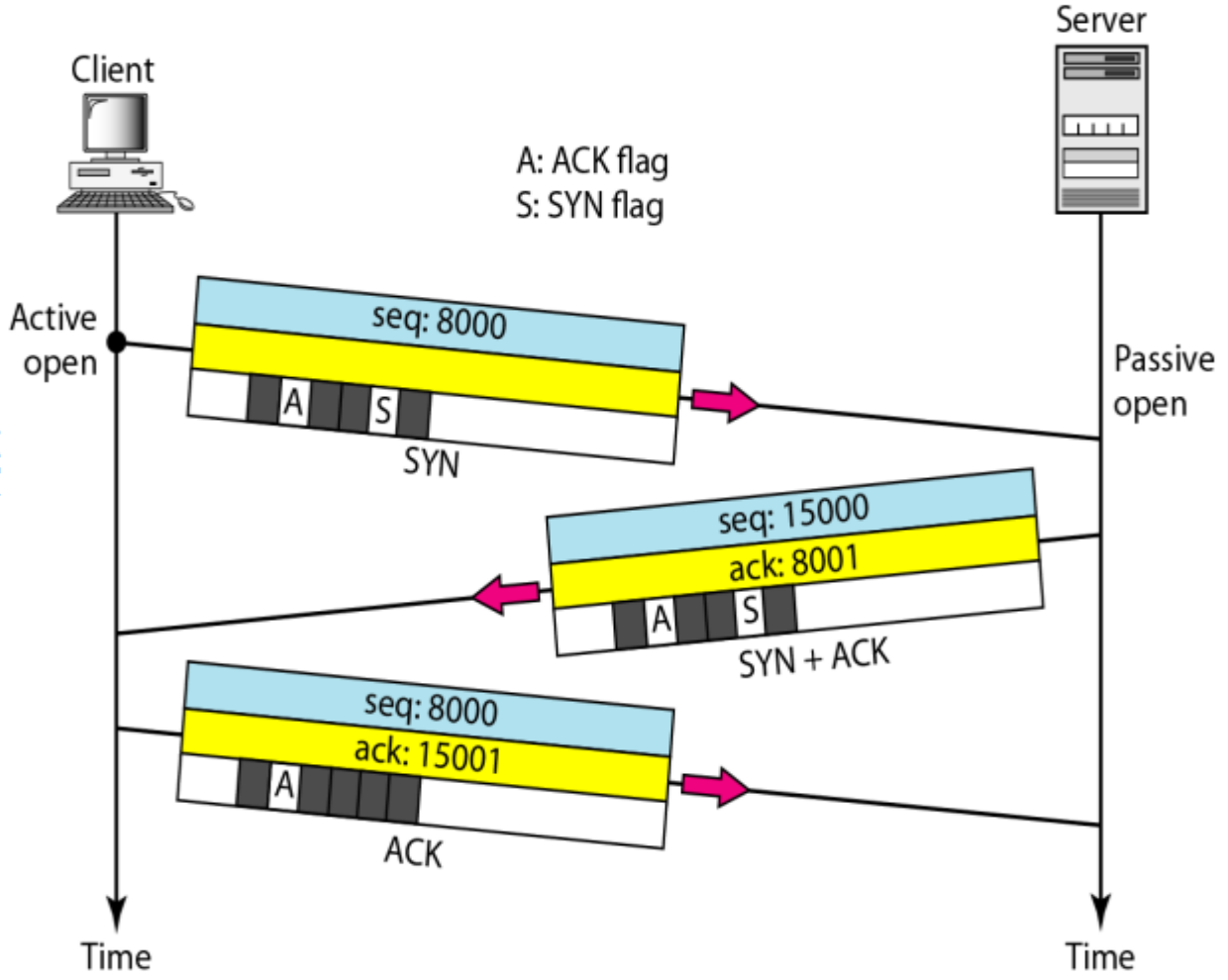
Việc sử dụng một đường dẫn ảo duy nhất cho toàn bộ bản tin giúp đơn giản hóa quá trình xác nhận cũng như việc truyền lại các khung bị hỏng hoặc mất.

Trong TCP, truyền tải hướng kết nối yêu cầu ba giai đoạn:

1. ***Thiết lập kết nối,***
2. ***Truyền dữ liệu,***
3. ***Kết thúc kết nối.***

5.4 Transmission Control Protocol

A TCP Connection



5.4 Transmission Control Protocol

A TCP Connection: SYN Flooding Attack

Điều này xảy ra khi một kẻ tấn công ác ý gửi nhiều segment SYN đến một máy chủ, mỗi segment đến từ một khách hàng khác nhau bằng cách giả mạo địa chỉ IP nguồn trong các gói dữ liệu. Giả sử rằng các khách hàng đang thực hiện một kết nối mở, máy chủ sẽ phân bổ các tài nguyên cần thiết, chẳng hạn như tạo các bảng giao tiếp và đặt bộ đếm thời gian. Máy chủ TCP sau đó sẽ gửi các segment SYN + ACK đến các khách hàng giả mạo, nhưng các segment này sẽ bị mất. Tuy nhiên, trong thời gian này, nhiều tài nguyên bị chiếm dụng mà không được sử dụng. Nếu trong thời gian ngắn này, số lượng segment SYN lớn, máy chủ cuối cùng sẽ cạn kiệt tài nguyên và có thể bị sập. Cuộc tấn công SYN flooding này thuộc loại tấn công bảo mật được gọi là tấn công từ chối dịch vụ (denial-of-service attack), trong đó kẻ tấn công độc quyền sử dụng hệ thống với quá nhiều yêu cầu dịch vụ đến mức hệ thống sụp đổ và từ chối dịch vụ cho mọi yêu cầu.

CÁC GIẢI PHÁP:

1. Một số hệ thống đã đặt giới hạn về số lượng yêu cầu kết nối trong một khoảng thời gian nhất định.
2. Một số khác lọc bỏ các gói dữ liệu đến từ các địa chỉ nguồn không mong muốn.
3. Chiến lược mới nhất là hoãn việc phân bổ tài nguyên cho đến khi toàn bộ kết nối được thiết lập.

5.4 Transmission Control Protocol

A TCP Connection: Data Transfer

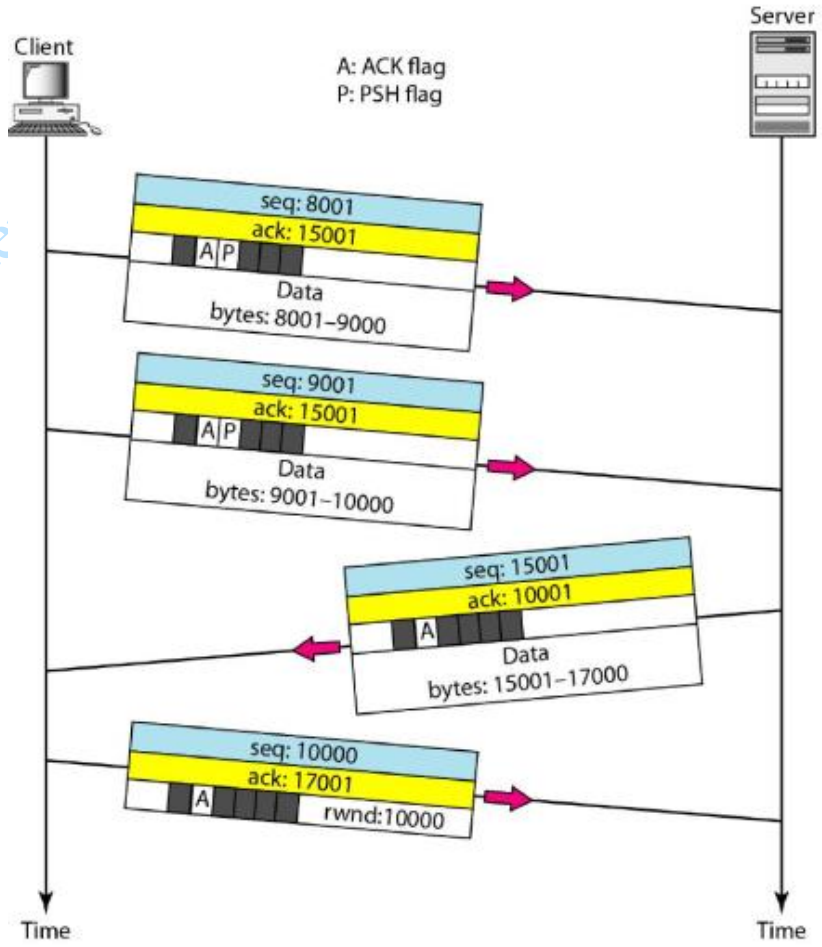
Sau khi kết nối được thiết lập, quá trình truyền dữ liệu hai chiều có thể diễn ra. Cả client và server đều có thể gửi dữ liệu và xác nhận. Xác nhận được "piggyback" (chèn kèm) với dữ liệu.

Ba segment đầu tiên mang cả dữ liệu và xác nhận, nhưng segment cuối cùng chỉ mang xác nhận vì không còn dữ liệu nào khác sẽ được gửi nữa.

Lưu ý các giá trị của số thứ tự (sequence number) và số xác nhận (acknowledgment number). Các segment dữ liệu được gửi bởi client có cờ PSH (push) được đặt, để TCP của máy chủ biết rằng cần chuyển dữ liệu đến tiến trình của server ngay khi chúng được nhận.

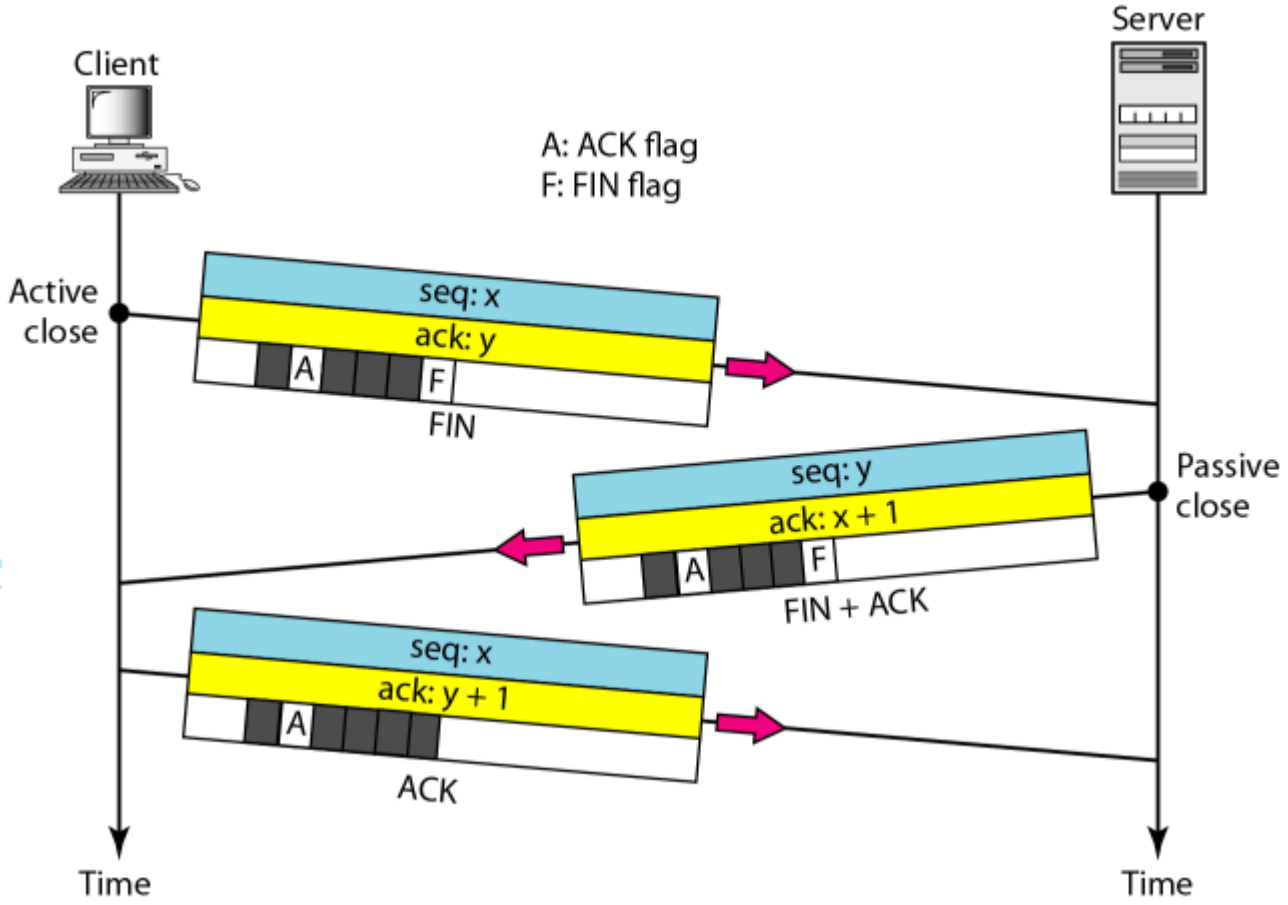
5.4 Transmission Control Protocol

A TCP Connection: Data Transfer



5.4 Transmission Control Protocol

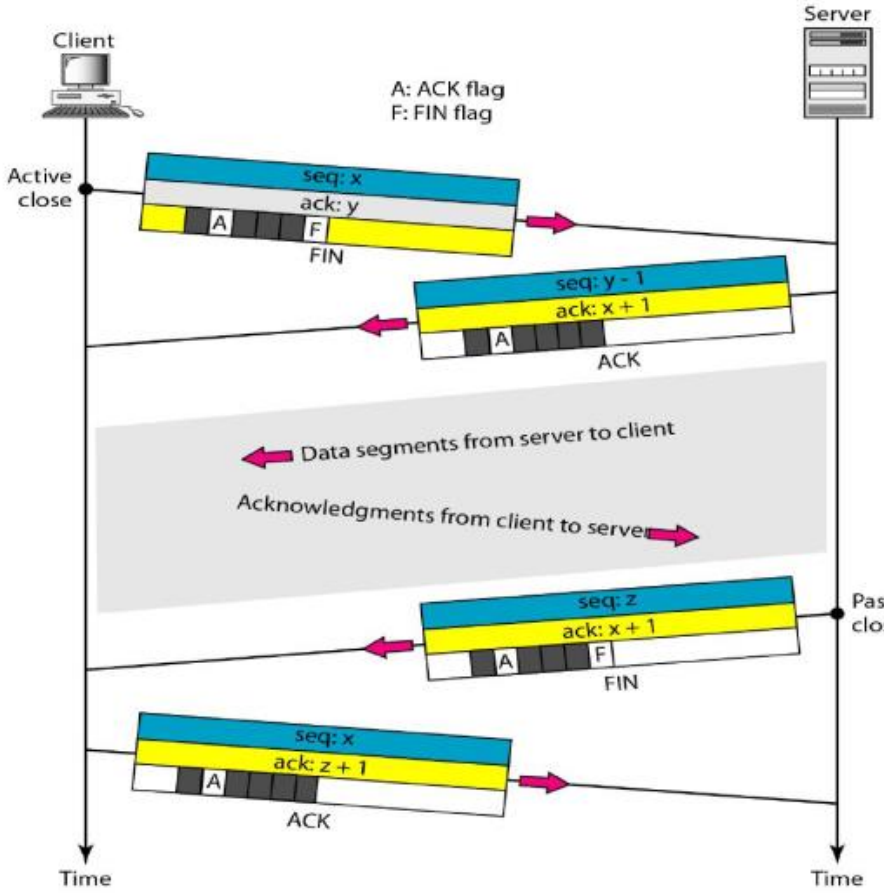
A TCP Connection: Chấm dứt kết nối



Connection termination using three-way handshaking

5.4 Transmission Control Protocol

A TCP Connection: Connection Termination



Half-close

5.4 Transmission Control Protocol

Kiểm soát luồng hoặc Cửa sổ trượt TCP

TCP sử dụng một cửa sổ trượt để xử lý kiểm soát luồng. Tuy nhiên, giao thức cửa sổ trượt mà TCP sử dụng nằm giữa giao thức Go-Back-N và Selective Repeat.

Giao thức cửa sổ trượt trong TCP giống với giao thức Go-Back-N vì nó không sử dụng NAKs (Negative Acknowledgments); nó giống với Selective Repeat vì phía nhận giữ các segment không theo thứ tự cho đến khi các segment bị thiếu đến.

Có hai sự khác biệt lớn giữa cửa sổ trượt này và cửa sổ trượt mà chúng ta đã sử dụng ở tầng liên kết dữ liệu:

1. Cửa sổ trượt của TCP là hướng byte; còn ở tầng liên kết dữ liệu là hướng khung (frame-oriented).
2. Cửa sổ trượt của TCP có kích thước thay đổi; còn ở tầng liên kết dữ liệu có kích thước cố định.

5.4 Transmission Control Protocol

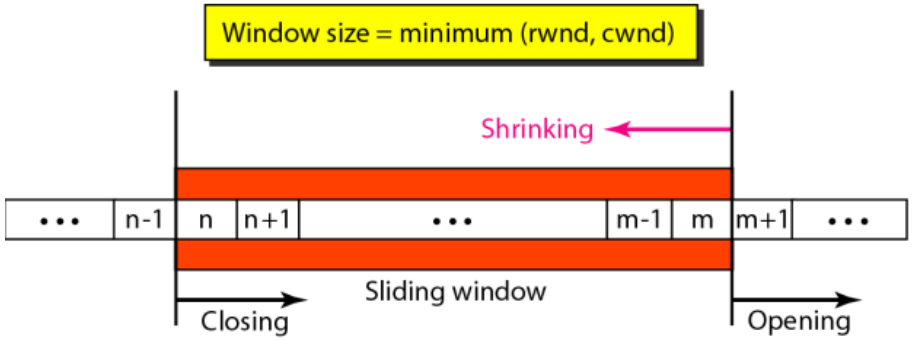
Flow Control or TCP Sliding Window

Cửa sổ có thể được mở, đóng, hoặc thu hẹp. Ba hoạt động này đều do bên nhận kiểm soát (và phụ thuộc vào tình trạng tắc nghẽn trong mạng), không phải bên gửi. bên gửi phải tuân theo các lệnh của bên nhận trong vấn đề này.

Mở cửa sổ có nghĩa là di chuyển bức tường bên phải sang phải. Điều này cho phép thêm nhiều byte mới trong bộ đệm có thể được gửi.

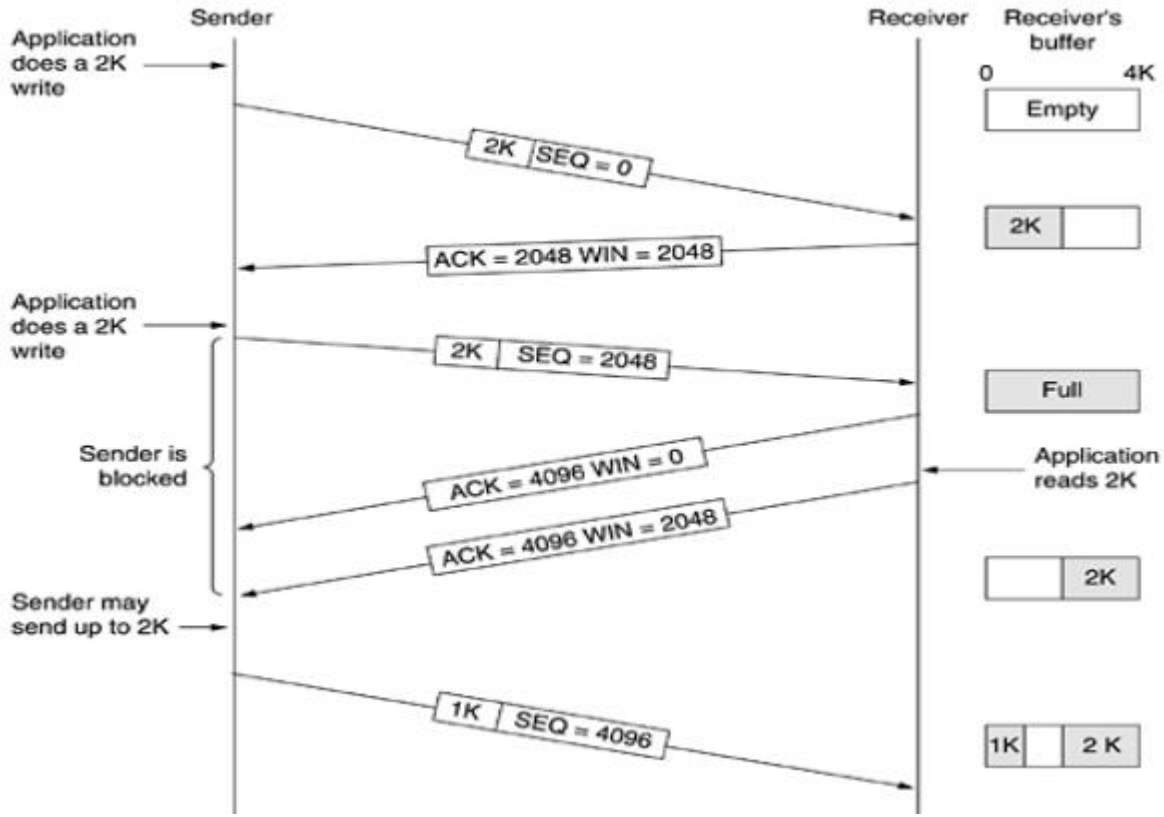
Đóng cửa sổ có nghĩa là di chuyển bức tường bên trái sang phải. Điều này có nghĩa là một số byte đã được xác nhận và bên gửi không cần lo lắng về chúng nữa.

Thu hẹp cửa sổ có nghĩa là di chuyển bức tường bên phải sang trái.



5.4 Transmission Control Protocol

Flow Control or TCP Sliding Window



Window management in TCP

5.4 Transmission Control Protocol

Error Control

TCP là một giao thức tầng truyền tải đáng tin cậy. Điều này có nghĩa là một chương trình ứng dụng khi gửi một luồng dữ liệu đến, sẽ dựa vào TCP để chuyển toàn bộ luồng dữ liệu đó đến chương trình ứng dụng ở đầu kia theo đúng thứ tự, không có lỗi và không bị mất hay trùng lặp phần nào.

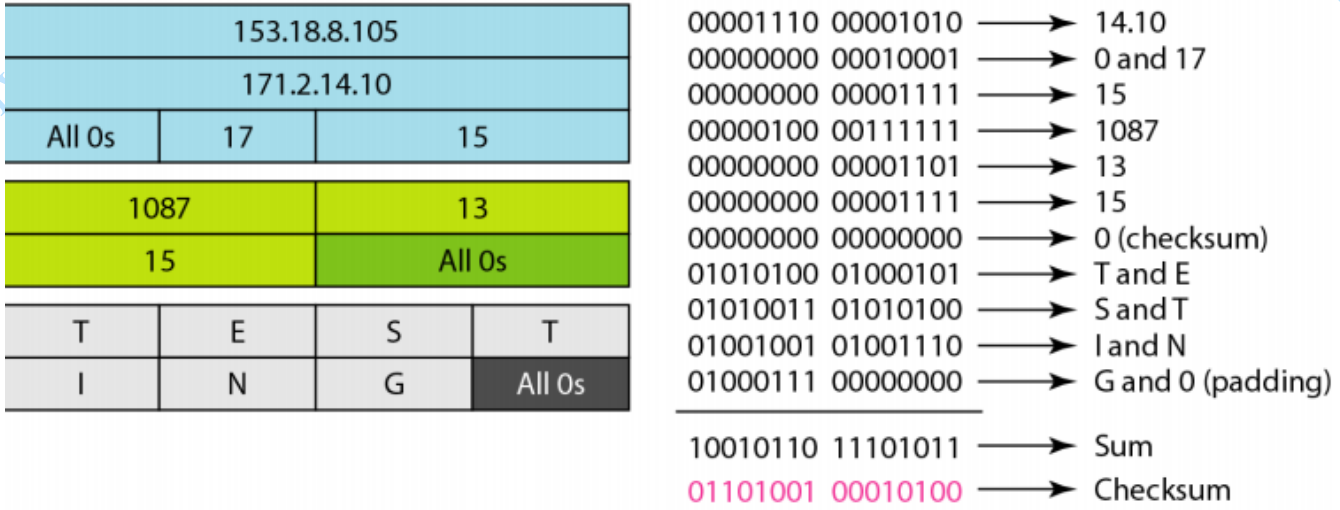
TCP cung cấp độ tin cậy thông qua việc kiểm soát lỗi. Kiểm soát lỗi bao gồm các cơ chế để phát hiện các phân đoạn bị hỏng, mất, sai thứ tự, và trùng lặp. Kiểm soát lỗi cũng bao gồm một cơ chế để sửa lỗi sau khi chúng được phát hiện. Việc phát hiện và sửa lỗi trong TCP được thực hiện bằng ba công cụ đơn giản: kiểm tra tổng (checksum), xác nhận (acknowledgment), và hết thời gian (time-out).

5.4 Transmission Control Protocol

Error Control

Checksum

Mỗi đoạn dữ liệu bao gồm một trường kiểm tra (checksum) được sử dụng để kiểm tra xem đoạn dữ liệu có bị hỏng hay không. Nếu đoạn dữ liệu bị hỏng, nó sẽ bị loại bỏ bởi TCP đích và được coi là đã mất. TCP sử dụng một trường kiểm tra 16-bit, là bắt buộc trong mỗi đoạn dữ liệu.



5.4 Transmission Control Protocol

Error Control

Acknowledgment

TCP sử dụng các thông báo xác nhận (acknowledgments) để xác nhận việc nhận các đoạn dữ liệu. Các đoạn điều khiển không chứa dữ liệu nhưng tiêu tốn một số thứ tự cũng sẽ được xác nhận.. Các đoạn ACK không tiêu tốn số thứ tự và không cần được xác nhận.

Retransmission

Trọng tâm của cơ chế kiểm soát lỗi là việc truyền lại các đoạn dữ liệu. Khi một đoạn dữ liệu bị hỏng, mất, hoặc bị trì hoãn, nó sẽ được truyền lại. Trong các cài đặt hiện đại, việc truyền lại xảy ra nếu bộ đếm thời gian truyền lại hết hạn hoặc ba đoạn ACK trùng lặp đã được nhận.

Truyền lại sau khi hết thời gian RTO (thời gian truyền lại hết hạn)

Truyền lại sau ba đoạn ACK trùng lặp (còn được gọi là truyền lại nhanh)

Out-of-Order Segments

Dữ liệu có thể đến không theo thứ tự và được lưu trữ tạm thời bởi TCP nhận, nhưng TCP vẫn đảm bảo rằng không có đoạn dữ liệu nào bị nhận ngoài thứ tự được gửi đến quá trình xử lý

5.4 Transmission Control Protocol

Kiểm soát tắc nghẽn TCP

Lớp mạng phát hiện tắc nghẽn khi các hàng đợi tại các bộ định tuyến trở nên quá lớn và cố gắng quản lý nó, dù chỉ bằng cách loại bỏ các gói tin. Lớp truyền tải chịu trách nhiệm nhận phản hồi về tắc nghẽn từ lớp mạng và làm chậm tốc độ truyền dữ liệu vào mạng.

Để kiểm soát tắc nghẽn, một giao thức truyền tải sử dụng quy luật điều khiển AIMD (Tăng thêm cộng, Giảm theo bội số). Kiểm soát tắc nghẽn TCP dựa trên việc triển khai phương pháp này bằng cách sử dụng một cửa sổ gọi là cửa sổ tắc nghẽn. TCP điều chỉnh kích thước của cửa sổ theo quy tắc AIMD. Kích thước cửa sổ tại máy gửi được thiết lập như sau:

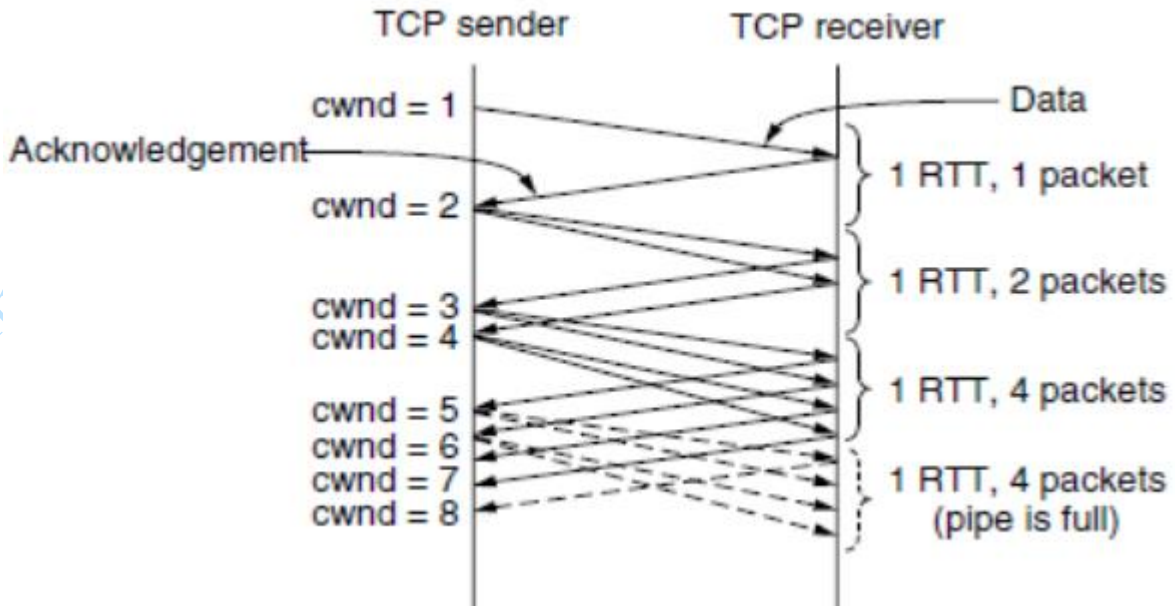
Cửa sổ gửi = MIN (cửa sổ điều khiển luồng, cửa sổ tắc nghẽn)

Trong đó: cửa sổ điều khiển luồng được quảng bá bởi máy nhận; cửa sổ tắc nghẽn được điều chỉnh dựa trên phản hồi.

5.4 Transmission Control Protocol

TCP Congestion Control

Giải pháp mà Jacobson chọn để xử lý cả hai vấn đề này là sự kết hợp giữa tăng tuyến tính và tăng theo bội số



SLOW-START

5.4 Transmission Control Protocol

TCP Congestion Control

Các phiên bản của kiểm soát tắc nghẽn TCP

- TCP Tahoe (1988)
  - Khởi động chậm (Slow Start)
  - Tránh tắc nghẽn (Congestion Avoidance)
  - Truyền lại nhanh (Fast Retransmit)
- TCP Reno (1990) (TCP Tahoe + FR)
  - Khôi phục nhanh (Fast Recovery)
- New Reno (1996)
- SACK (1996) (SACK (Xác nhận chọn lọc))
- RED (Floyd và Jacobson 1993)


CMC UNIVERSITY
THANK YOU