

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

3.5.1 Cấu trúc đoạn dữ liệu (segment)

3.5.2 Truyền dữ liệu tin cậy

3.5.3 Điều khiển luồng

3.5.4 Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-73

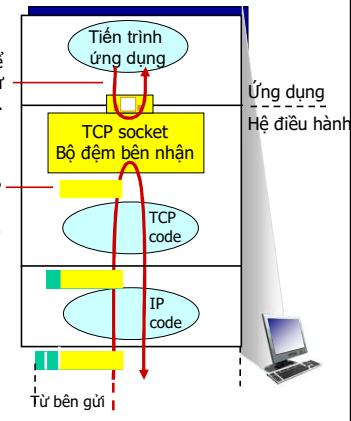
Điều khiển luồng trong TCP

Ứng dụng có thể di chuyển dữ liệu từ bộ đệm TCP socket ...

... chậm hơn so với bên nhận TCP (trong khi bên gửi vẫn đang gửi)

Điều khiển luồng

Bên nhận điều khiển bên gửi, để bên gửi sẽ không làm tràn bộ đệm của bên nhận do việc gửi quá nhanh và quá nhiều

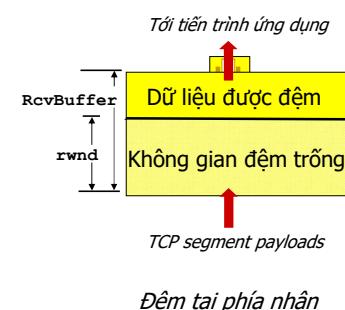


Tầng giao vận 3-74

Điều khiển luồng trong TCP

❖ Bên nhận “thông báo” không gian đệm còn trống bởi giá trị **rwnd** trong TCP header của các segment gửi-nhận

- Kích thước **RcvBuffer** được thiết lập qua tùy chọn (option) của socket (thường mặc định là 4096 byte)
- Nhiều hệ điều hành tự động điều chỉnh **RcvBuffer**



Tầng giao vận 3-75

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

3.5.1 Cấu trúc đoạn dữ liệu (segment)

3.5.2 Truyền dữ liệu tin cậy

3.5.3 Điều khiển luồng

3.5.4 Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

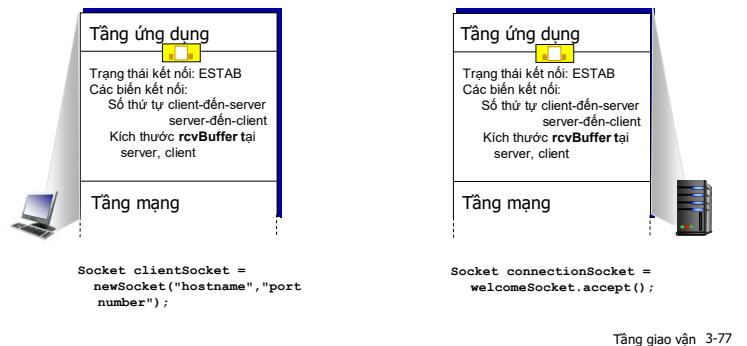
3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-76

Quản lý kết nối

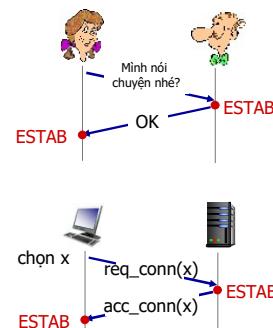
Trước khi trao đổi dữ liệu, bên gửi/bên nhận “bắt tay”:

- ❖ Đồng ý thiết lập kết nối
- ❖ Đồng ý các tham số kết nối



Đồng ý thiết lập kết nối

Bắt tay hai bước



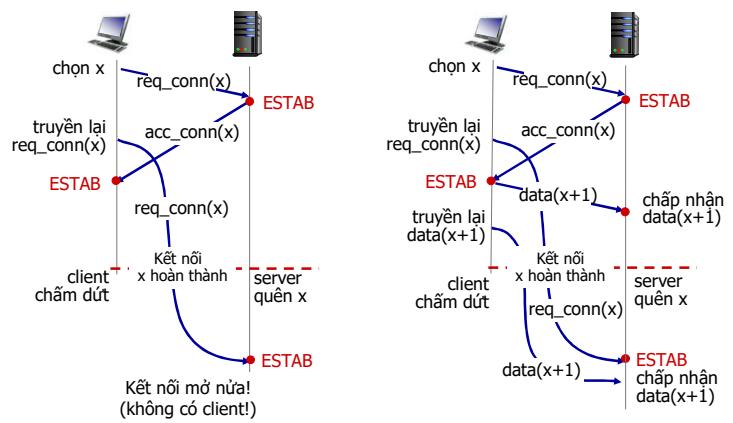
Hỏi: Bắt tay hai bước có luôn được thực hiện trên mạng?

- ❖ Chậm trễ liên quan tới các biến
- ❖ Các thông điệp được truyền lại (ví dụ, req_conn(x)) là do mất thông điệp
- ❖ Sắp xếp lại thông điệp
- ❖ Không thể “nhìn thấy” phía bên kia

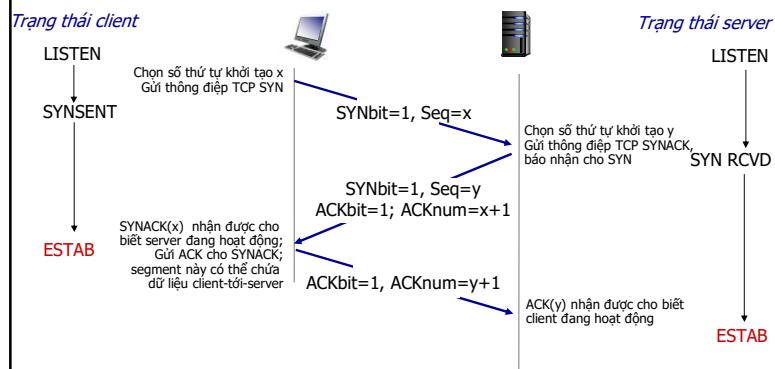
Tầng giao vận 3-78

Đồng ý thiết lập kết nối

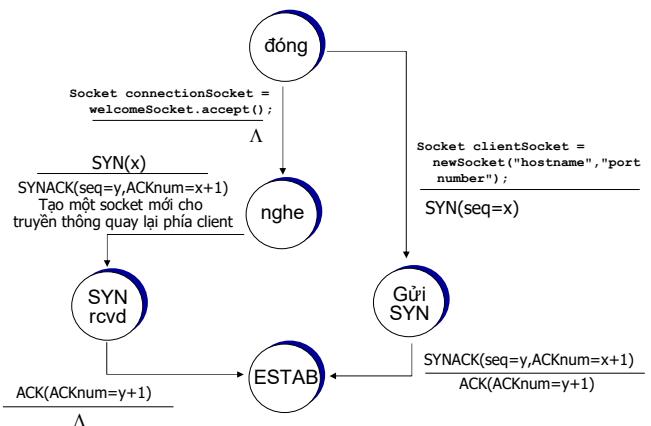
Các kịch bản có lỗi khi thực hiện bắt tay 2 bước:



Bắt tay 3 bước trong TCP



Bắt tay 3 bước trong TCP: FSM



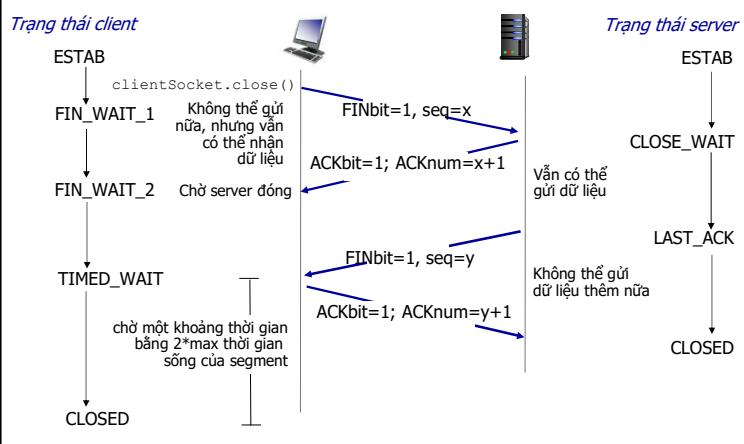
Tổng giao vận 3-81

TCP: đóng kết nối

- ❖ Mỗi bên client và server thực hiện đóng kết nối
 - Gửi TCP segment với bit FIN = 1
- ❖ Đáp ứng lại FIN nhận được bằng ACK
 - FIN, ACK đang nhận có thể được kết nối với FIN của nó
- ❖ Có thể thực hiện đồng bộ trao đổi FIN

Tổng giao vận 3-82

TCP: đóng kết nối



Tổng giao vận 3-83