

Họ và tên: Nguyễn Hải Đăng

Điểm

MSSV: 23127165

Lớp: 23CLC03

BÀI TẬP ÔN TẬP TẦNG TRANSPORT

Câu 1:

4 lý do tại sao UDP lại nhanh hơn TCP:

① Không thiết lập kết nối

+ UDP bắt đầu gửi dữ liệu ngay lập tức

+ TCP yêu cầu bắt tay 3 bước (three-way handshake) trước khi truyền dữ liệu.

② Không có cơ chế kiểm soát luồng và kiểm soát tắc nghẽn

+ TCP yêu cầu thực hiện cơ chế này để điều chỉnh tốc độ truyền, giảm rủi ro mất gói nhưng cũng làm giảm tốc độ.

③ Không yêu cầu xác nhận gói tin (~~ADK~~^{ACK})

+ TCP yêu cầu gói tin phải được xác nhận (ACK) để đảm bảo chúng đến đích.

④ Header đơn giản hơn

+ Header UDP chỉ 8 byte

+ Header TCP từ 20 - 60 byte với nhiều thông tin điều khiển hơn



ĐOÀN KẾT - TRÁCH NHIỆM - CHẤT LƯỢNG - TIÊN PHONG

Câu 2:

Stop & Wait	Pipeline
<ul style="list-style-type: none">Người gửi gửi một gói tin và chờ xác nhận (ACK)Chỉ một gói tin đang truyền tại một thời điểmSử dụng băng thông thấpCài đặt đơn giản hơn.	<ul style="list-style-type: none">Nhiều gói tin được gửi mà không cần chờ xác nhậnNhiều gói tin đang truyền đồng thờiSử dụng băng thông cao hơnPhức tạp hơn nhưng hiệu quả hơn

Câu 3:

Go-Back-N	Selective Repeat
<ul style="list-style-type: none">Nếu phát hiện lỗi, người gửi gửi lại tất cả các gói tin từ gói lỗi trở điNgười nhận chỉ chấp nhận các gói tin theo thứ tự.Bên nhận đơn giản hơn (chỉ theo dõi số thứ tự gói tin tiếp theo)Kém hiệu quả khi tỷ lệ lỗi caoLãng phí băng thông do phải gửi lại cả những gói đã nhận đúng.	<ul style="list-style-type: none">Chỉ truyền lại các gói tin cụ thể bị mất/lỗiNgười nhận chấp nhận các gói tin không theo thứ tựBên nhận phức tạp hơn (phải đếm các gói tin không theo thứ tự)Hiệu quả hơn khi tỷ lệ lỗi caoSử dụng băng thông tốt hơn



ĐOÀN KẾT - TRÁCH NHIỆM - CHẤT LƯỢNG - TIÊN PHONG

Gửi thích 2 hình:

Hình 1: Go-Back-N

Sender trong hình gửi 4 packets trong 1 lần gửi và Receiver đã nhận được pkt0, pkt1 nhưng pkt2 lại không nhận được. Slide minh họa khi đó dịch tới trước pkt2 (gửi pkt4, pkt5). Sau khi hết time của pkt2 thì Sender gửi pkt từ 2 → 5 cho Receiver và Receivers trả về các ACK theo thứ tự bình thường.

Hình 2: Selective Repeat

Cách nhận biết là khi pkt2 loss thì chỉ gửi lại gói tin pkt2 mà không ảnh hưởng đến các pkt khác. Tương tự trên Sender trong hình gửi 4 packets trong 1 lần gửi và Receivers nhận được pkt0, pkt1. Thế nhưng pkt2 bị loss, bỏ qua pkt2; pkt3, pkt4, pkt5 vẫn được gửi đi nhưng tạm thời Receivers chưa trả lại ACK. Cho tới khi pkt2 được gửi lại, ACK2, ACK3, ACK4, ACK5 được Receivers trả về bình thường.

Câu 4:

① Gói tin đầu tiên có sequence number là 127 và host B nhận đủ liên đến byte 126

Gói tin đầu tiên là 80 nên chỉ số ACK là: $80 + 127 = 207$



ĐOÀN KẾT - TRÁCH NHIỆM - CHẤT LƯỢNG - TIÊN PHONG

② Gói 2 đến trước gói 1 \Rightarrow 1 không gửi được nên sẽ yêu cầu gửi lại nên ACK của gói 1 là 127

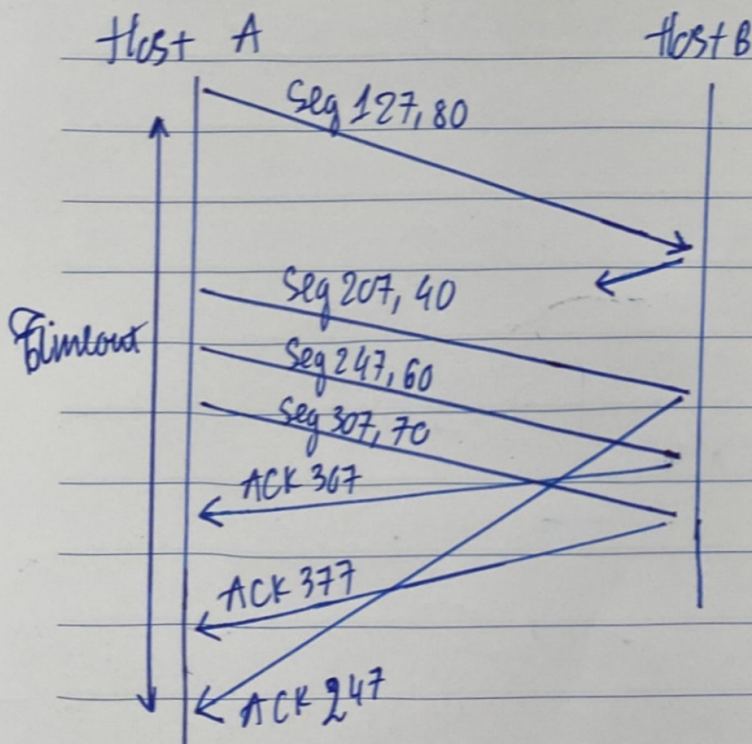
③ Trình tự xảy ra:

Gói 1: A \rightarrow B (127, 80)

Gói 2: A \rightarrow B (~~127~~, 207, 40)

Host B: nhận gói 1 và 2 nhưng ACK cho gói tin 1 mãi nên nhận ACK = 247

Host A: không nhận được ACK cho gói 1 \Rightarrow Timeout và retransmit gói 1.



④ Bỏ gói sau khi gửi ACK 127

trước khi hết time out

Fast Retransmit: lập tức bên A gửi lại các gói seg 127, 80



ĐOÀN KẾT - TRÁCH NHIỆM - CHẤT LƯỢNG - TIỀN PHONG