

Esercitazione

Livello di Trasporto
[Capitolo 3]

- È possibile che un'applicazione che gira su UDP ottenga un trasferimento dati affidabile?
- Sì. Lo sviluppatore dell'applicazione può inserire il trasferimento affidabile nel livello applicazione. Questo approccio può comunque risultare costoso in termini di quantità di lavoro e debugging

– 2.1 –

- Supponete che un processo su un host C abbia una socket Udp con un numero di porta 6789
- Supponete che entrambi gli host A e B mandino ciascuno un segmento UDP all'host C con numero di porta destinazione 6789
- Entrambi questi segmenti saranno diretti alla stessa socket dell'host C?
 - Sì

– 2.2 –

- Supponete che un processo in un host C abbia una socket Udp con un numero di porta 6789
- Supponete che entrambi gli host A e B mandino ciascuno un segmento UDP all'host C con numero di porta destinazione 6789
- Se sì, come il processo sull'host C saprà che questi due segmenti hanno origine da due host diversi?
 - Per ogni segmento ricevuto sulla socket, il SO fornirà al processo applicativo l'indirizzo IP del mittente presente a livello di rete

– 3 –

- Supponete che un server web sia in esecuzione sull'host C sulla porta 80, che usi connessioni persistenti, e che stia ricevendo richieste da due diversi host, A e B
- Tutte le richieste vengono inviate attraverso la stessa socket sull'host C?
 - In genere il server web crea un thread per ogni richiesta
 - A ogni thread viene assegnata una nuova socket su una differente porta, che si occupa di gestire la nuova richiesta

– 4.1 –

- Nei protocolli *rtd* esaminati, perché abbiamo bisogno di introdurre i numeri di sequenza?
 - Sono necessari al ricevente per distinguere la trasmissione di un nuovo messaggio dalla ritrasmissione di un messaggio già trasmesso
 - Rdt2.1, ACK o NAK corrotti

– 4.2 –

- Nei protocolli *rtd* esaminati, perché abbiamo bisogno di introdurre i timer?
- Cosa succede quando scade?
 - Per gestire le *perdite* dei messaggi
 - Quando scade il timer, il relativo messaggio viene inviato nuovamente
 - Rdt3.0

– 5.1 –

- Considerate il protocollo Go-back-N
- La sorgente invia 5 pacchetti e il primo si perde prima che qualche pacchetto riesca a raggiungere la destinazione
- Come reagisce il ricevente?
 - Non fa nulla
 - Normalmente, avendo rilevato un buco, invierebbe l'ACK dell'ultimo pacchetto ricevuto correttamente
 - In questo *caso speciale*, nessun pacchetto è stato mai ricevuto

– 5.2 –

- Considerate il protocollo Go-back-N
- La sorgente invia 5 pacchetti, il primo pacchetto raggiunge la destinazione, ma il suo riscontro viene perso
- Cosa accade?
 - Riscontro cumulativo
 - Tutti i pacchetti dallo 0 al 4 vengono correttamente riscontrati

– 5.3 –

- Considerate il protocollo Go-back-N
- La finestra è grande 5 ma il mittente prova a inviare 6 pacchetti
- Cosa accade?
 - Il sesto pacchetto non parte fino a quando uno dei precedenti 5 non viene riscontrato

– 6.1 –

- Considerate il protocollo Ripetizione Selettiva
- La sorgente invia 5 pacchetti e il primo si perde prima che qualche pacchetto riesca a raggiungere la destinazione
- Cosa accade?
 - I pacchetti 1-4 vengono bufferizzati e riscontrati
 - La finestra del mittente non cambia
 - Allo scadere del timer 0, il pacchetto 0 viene inviato nuovamente

– 6.2 –

- Considerate il protocollo Ripetizione Selettiva
- La sorgente invia 5 pacchetti, il primo pacchetto raggiunge la destinazione, ma il suo riscontro viene perso
- Cosa accade?
 - La stessa cosa del caso precedente
 - No riscontro cumulativo

– 6.3 –

- Considerate il protocollo Ripetizione Selettiva
- La finestra è grande 5 ma il mittente prova a inviare 6 pacchetti
- Cosa accade?
 - Il sesto pacchetto non parte fino a quando il pacchetto 0 non viene riscontrato
 - Non c'è riscontro cumulativo

– 7.1 –

- L'host A sta trasmettendo a B un file di grandi dimensioni su una connessione TCP
- B non ha dati da inviare ad A, quindi non manderà riscontri a quest'ultimo, dato che non ha suoi dati da inserire negli ACK
- **Vero o Falso?**
 - **Falso**

– 7.2 –

- La dimensione di `RcvWindow` in TCP non cambia mai per tutta la durata della connessione
- Vero o Falso?
 - Falso

– 7.3 –

- L'host A sta trasmettendo a B un file di grandi dimensioni su una connessione TCP
- Se il numero di sequenza di un segmento di questa connessione è m , allora il numero di sequenza del segmento successivo sarà necessariamente $m+1$
- Vero o Falso?
 - Falso

– 7.4 –

- Supponiamo che l'ultimo `SampleRTT` di una connessione TCP valga un secondo
- Il valore corrente di `TimeoutInterval` per la connessione sarà necessariamente maggiore o uguale a un secondo
- **Vero o Falso?**
 - **Falso**

– 7.5 –

- Su una connessione TCP l'host A sta trasmettendo a B un segmento con numero di sequenza 38 e 4 byte di dati
- In questo stesso segmento il numero di riscontro è necessariamente 42
- Vero o Falso?
 - Falso

– 8.1 –

- L'host A sta trasmettendo a B due segmenti su una connessione TCP
- Il primo segmento ha numero di sequenza 90 e il secondo 110
- Quanti dati si trovano nel primo segmento?
 - 20 byte

– 8.1 –

- L'host A sta trasmettendo a B due segmenti su una connessione TCP
- Supponiamo che il primo segmento vada perso, ma il secondo arrivi a B
- **Quale sarà il numero del riscontro che B manda ad A?**

– 90

- Consideriamo il controllo di congestione TCP
- Quando il timer del mittente scade, quale valore assume la soglia?
 - Metà del valore corrente della finestra di congestione

– 10 –

- I client A e B iniziano una sessione Telnet con il server S
- Quali sono i numeri di porta destinazione per i segmenti trasmessi da A,B verso S?

– 23

- Se A e B sono host diversi, è possibile che il numero di porta origine dei segmenti da A a S coincida con quello dei segmenti da B a S?

– Sì

- Se sono sullo stesso host?

– No

– 11.1 –

- UDP e TCP utilizzano il complemento a 1 per la checksum
- Supponiamo di avere i seguenti 3 byte
 - 01010101, 01110000 e 01001100
- Qual è il complemento a 1 della loro somma?
 - 11101110

– 11.2 –

- UDP e TCP utilizzano il complemento a 1 per la checksum
- Perché UDP considera il complemento a 1 della somma e non la semplice somma?
 - È più semplice il controllo da parte del destinatario
 - Se c'è qualche 0 → errore

– 11.3 –

- UDP e TCP utilizzano il complemento a 1 per la checksum
- È possibile che un errore su un bit non venga rilevato?
 - No

– 11.4 –

- UDP e TCP utilizzano il complemento a 1 per la checksum
- È possibile che un errore su due bit non venga rilevato?
 - Sì....fornire un esempio
 - Es.: l'ultimo bit del primo intero passa da 1 a 0 e l'ultimo bit del secondo intero da 0 a 1

– 11.5 –

- UDP e TCP utilizzano il complemento a 1 per la checksum
- Se il ricevente passa la verifica del checksum, può essere assolutamente sicuro che non vi siano errori sui bit?
 - No....fornire un esempio
 - Es.: tutti i bit delle parole arrivano “flippati”

– 12.1 –

- Nel protocollo Ripetizione Selettiva, il mittente può ricevere un ACK relativo a un pacchetto che ricade al di fuori della sua finestra corrente?
- Sì....fornite un esempio
 - Es.: mittente invia 0, 1 e 2 al tempo t_0 . Al tempo t_1 il ricevente invia gli ACK per 0, 1 e 2. Al tempo t_2 il mittente va in timeout, e invia nuovamente 0, 1 e 2. A t_3 il ricevente riceve i duplicati e invia nuovamente gli ACK per 0, 1 e 2. A t_4 il mittente riceve gli ACK e avanza la finestra (sia $N=3$). A t_5 arrivano gli ACK inviati a t_1 (ritardatari). Questi sono fuori dalla finestra

– 12.2 –

- Nel protocollo Go-Back-N, il mittente può ricevere un ACK relativo a un pacchetto che ricade al di fuori della sua finestra corrente?
- Sì....scenario simile al precedente

– 12.3 –

- In quale caso il protocollo stop-and-wait può essere considerato uguale al protocollo Ripetizione Selettiva?
- Quando la dimensione di finestra mittente e destinatario sono pari a 1

– 12.3 –

- Il protocollo stop-and-wait è uguale al protocollo Go-Back-N con dimensione di finestra mittente e destinatario pari a 1?
- Sì
 - In questo caso GBN e RS sono equivalenti
 - ACK cumulativo è un ACK ordinario

- Con quale protocollo di trasporto un'applicazione ha maggiore controllo su **quando** un segmento viene inviato?
- **UDP**
 - **Nessun controllo di flusso o di congestione**

– 14.1 –

- A e B comunicano su una connessione TCP, e B ha appena ricevuto da A tutti i byte fino al 248
- A manda 2 segmenti a B uno dietro l'altro
- I due segmenti contengono 40 e 60 byte di dati, rispettivamente
- Nel primo segmento il numero di sequenza è 249, il numero di porta sorgente è 503 e il numero di porta destinazione è 80
- B manda un riscontro non appena riceve un segmento da A
- Nel secondo segmento da A a B, quali sono il numero di sequenza, e i numeri di porta sorgente e destinazione?

– 289, 503, 80

– 14.2 –

- A e B comunicano su una connessione TCP, e B ha appena ricevuto da A tutti i byte fino al 248
- A manda 2 segmenti a B uno dietro l'altro
- I due segmenti contengono 40 e 60 byte di dati, rispettivamente
- Nel primo segmento il numero di sequenza è 249, il numero di porta sorgente è 503 e il numero di porta destinazione è 80
- B manda un riscontro non appena riceve un segmento da A
- Se il primo segmento arriva prima del secondo, nel riscontro del primo segmento arrivato, qual è il numero di riscontro, e i numeri di porta sorgente e destinazione?

– 289, 80, 503

– 14.3 –

- A e B comunicano su una connessione TCP, e B ha appena ricevuto da A tutti i byte fino al 248
- A manda 2 segmenti a B uno dietro l'altro
- I due segmenti contengono 40 e 60 byte di dati, rispettivamente
- Nel primo segmento il numero di sequenza è 249, il numero di porta sorgente è 503 e il numero di porta destinazione è 80
- B manda un riscontro non appena riceve un segmento da A
- Se il secondo segmento arriva prima del primo, nel riscontro del primo segmento arrivato, qual è il numero di riscontro, e i numeri di porta sorgente e destinazione?

– 249, 80, 503

– 15 –

- A e B sono direttamente connessi con un collegamento a 200 Mbps, e A sta inviando un file a B su TCP
- A invia dati a 100 Mbps, ma B può leggere dal suo buffer di ricezione TCP a una frequenza massima di 50 Mbps
- **Descrivete cosa accade**
- **Entra in gioco il controllo di flusso TCP**
 - **Quando il buffer di B è pieno, invia segmenti con `RcvWindow=0`**
 - **La frequenza di invio da A a B è non è più alta di 50 Mbps**

– 16.1 –

- Considerate l'invio di un grosso file da A a B su una connessione TCP senza perdite
- Supponete che TCP usi AIMD per il controllo della congestione
- Assumendo approssimativamente costanti i tempi di andata e ritorno, quanto ci metterà `CongWin` ad aumentare da 1 MSS a 6 MSS (assumendo che non vi siano eventi di perdita)?
 - 5 RTT

– 16.2 –

- Considerate l'invio di un grosso file da A a B su una connessione TCP senza perdite
- Supponete che TCP usi AIMD per il controllo della congestione
- Qual è il throughput medio (in termini di MSS e RTT) per questa connessione fino all'istante pari a 5 RTT?
 - 3 MSS/RTT
 - Nel primo RTT \rightarrow 1 MSS
 - Nel secondo RTT \rightarrow 2 MSS
 - Nel terzo RTT \rightarrow 3MSS
 - Nei primi 5 RTT $\rightarrow 1+2+\dots+5 = 15$ MSS
 - Throughput Medio = $15 \text{ MSS} / 5 \text{ RTT} = 3 \text{ MSS/RTT}$

Esercitazione

Livello di Rete
[Capitolo 4]

- Spiegate quando si può verificare la perdita di pacchetti presso le porte d'ingresso e sotto quale condizione non si verifica
- Si verifica quando la struttura d'interconnessione è più lenta della frequenza a cui arrivano dati dalle porte d'ingresso
- Non si verifica quando la struttura d'interconnessione è più veloce di $n * freq$ di arrivo dei dati, con n il numero di porte in ingresso

– 18 –

- I router hanno indirizzi IP? Se sì, quanti?
 - Sì, pari al numero di interfacce

– 19.1 –

- Ipotizzate che ci siano 3 router tra un host origine e uno destinazione
- Ignorando la frammentazione, quante interfacce attraversa un datagramma IP?

– 8

– 19.2 –

- Ipotizzate che ci siano 3 router tra un host origine e uno destinazione
- Quante tabelle d'inoltro saranno consultate per recapitare il datagramma?

– 3

- Supponete che un'applicazione generi blocchi di 40 byte di dati ogni 20ms e che i blocchi siano incapsulati in un segmento TCP e poi in un datagramma IP
- Quale percentuale di datagramma sarà costituita da *overhead* e quale da dati applicativi?

– 50%

- L'host A invia a B un segmento TCP incapsulato in un datagramma IP
- Come può sapere il livello di rete di B di dover passare il segmento (i dati) a TCP anziché a UDP o altro?
 - Dal campo “protocollo” della intestazione del datagramma IP (8 bit)

– 22 –

- Considerate una rete a circuito virtuale e supponete che il numero VC sia un campo a 16 bit
- Qual è il numero massimo di circuiti virtuali che possono essere trasportati su un collegamento?

$$- 2^{16} = 65536$$

– 23.1 –

- In una rete a circuito virtuale le tabelle d'inoltro hanno 4 colonne
- Quali sono i loro significati?
 - Interfaccia d'ingresso
 - Numero VC entrante
 - Interfaccia d'uscita
 - Numero VC uscente

– 23.2 –

- In una rete a datagramma le tabelle d'inoltro hanno 2 colonne
- Quali sono i loro significati?
 - Indirizzo di destinazione
 - Interfaccia d'uscita

– 24.1 –

A	B	C	D
00	01	10	11
01	10	11	00

- Sia data un rete a VC con un campo da 2 bit per il numero di VC
- La rete vuole impostare un circuito su 4 collegamenti (A, B, C, D)
 - Ciascun collegamento sta già trasportando altri 2 circuiti virtuali, numerati come riportato in tabella
- Se ogni circuito deve usare lo stesso numero di VC su tutti i collegamenti lungo il percorso, quale numero VC potrebbe essere assegnato al nuovo circuito virtuale?
 - Nessuno: il circuito non può essere stabilito

– 24.2 –

A	B	C	D
00	01	10	11
01	10	11	00

- Sia data un rete a VC con un campo da 2 bit per il numero di VC
- La rete vuole impostare un circuito su 4 collegamenti (A, B, C, D)
 - Ciascun collegamento sta già trasportando altri 2 circuiti virtuali, numerati come riportato in tabella
- Se ogni circuito può usare numeri di VC differenti, quale combinazione potrebbe essere usata?
 - 10, 00, 00, 01

– 25.1 –

- Sia dato un router con 2 porte d'ingresso (A e B), una struttura di commutazione, e due porte d'uscita (C e D)
- Supponete che la struttura di commutazione funzioni a 1,5 volte la velocità di linea
- Se tutti i pacchetti da A sono destinati a D e tutti i pacchetti da B a C, si può progettare una struttura di commutazione tale che non vi sia accodamento in ingresso?
 - No. La struttura sarebbe comunque troppo lenta
 - Quante volte dovrebbe essere più veloce della velocità di linea?

– 25.2 –

- Sia dato un router con 2 porte d'ingresso (A e B), una struttura di commutazione, e due porte d'uscita (C e D)
- Supponete che la struttura di commutazione funzioni a 1,5 volte la velocità di linea
- Se i pacchetti da A e B sono destinati in modo casuale a D e a C, si può progettare una struttura di commutazione tale che non vi sia accodamento in ingresso?
 - No. La struttura sarebbe comunque troppo lenta

11100000			0
11100001	00000000		1
11100001	00000000		1
11100001			2

- Supponete che un router abbia 4 collegamenti, e che i pacchetti debbano essere inoltrati come segue

					Interfaccia
da	11100000	00000000	00000000	00000000	0
a	11100000	11111111	11111111	11111111	
da	11100001	00000000	00000000	00000000	
a	11100001	00000000	11111111	11111111	
da	11100001	00000001	00000000	00000000	
a	11100001	11111111	11111111	11111111	
altrimenti					

- Presentare una tabella d'inoltro con 4 righe che utilizza il confronto a prefisso più lungo

– 27 –

- Considerate una rete a datagramma con indirizzi host a 8 bit
- Supponete che un router utilizzi il confronto a prefisso più lungo e abbia la seguente tabella d'inoltro
- Per ognuna delle 4 interfacce, fornite l'intervallo di indirizzi coperto

		Interfaccia
da	00000000	0
a	00111111	
da	01000000	1
a	01111111	
da	10000000	2
a	10111111	
da	11000000	3
a	11111111	

Confronto di prefisso	Interfaccia
00	0
01	1
10	2
11	3

– 28 –

- Considerate una rete a datagramma con indirizzi host a 8 bit
- Supponete che un router utilizzi il confronto a prefisso più lungo e abbia la seguente tabella d'inoltro
- Per ognuna delle 4 interfacce, fornite l'intervallo di indirizzi coperto

		Interfaccia
da	10000000	0
a	10111111	
da	11000000	1
a	11011111	
da	11100000	2
a	11111111	
da	00000000	3
a	01111111	

Confronto di prefisso	Interfaccia
1	0
11	1
111	2
altrimenti	3