

RETI DI CALCOLATORI – prova scritta del 12/07/2016

Per essere ammessi alla prova orale è necessario ottenere una valutazione sufficiente della prima parte

Prima parte (15 punti)

- Q1.** L'utente *mickey@disney.com* invia dal suo PC un email a *donald@disney.com*. Indicare la sequenza di comandi SMTP inviati e ricevuti dal PC di *mickey@disney.com* se: (a) il mailserver di *disney.com* non è raggiungibile, oppure (b) il mailserver di *disney.com* è raggiungibile ma *donald* non è un utente di *disney.com*.
- Q2.** Un sender Selective Repeat con dimensione della finestra uguale a $2N$ e con N segmenti "in volo" riceve un riscontro non duplicato R . Determinare – giustificando la risposta – quanti nuovi segmenti tale sender potrebbe inviare dopo avere ricevuto R .
- Q3.** (a) Un router IPv4 $R1$ deve inoltrare su un collegamento con MTU di 500 byte con un router IPv4 $R2$ un datagram IP che ha ricevuto, la cui intestazione non contiene opzioni e che contiene un segmento TCP di 990 byte. Indicare i valori dei campi *offset* ("scostamento") e *length* ("lunghezza totale") dei frammenti inviati da $R1$. (b) $R2$ deve inoltrare tutti i frammenti ricevuti da $R1$ su un collegamento con MTU di 260 byte con un router IPv4 $R3$. Indicare i valori dei campi *offset* e *length* dei frammenti inviati da $R2$.
- Q4.** Consideriamo una rete locale che utilizza il protocollo MAC Slotted Aloha p -persistente, con $p=0,5$, e supponiamo che un host A di tale rete trasmetta per la seconda volta un frame F durante lo slot X , rilevando una collisione. Indicare – giustificando la risposta – se la probabilità che A riesca a trasmettere F con successo nello slot $X+3$ senza rilevare altre collisioni è minore di $1/4$.
- Q5.** Consideriamo un rete wireless IEEE 802.11 che ha un raggio di 400 metri, velocità di trasmissione di 5 Mbps e velocità di propagazione di $2 \cdot 10^8$ m/sec; la durata di DIFS e SIFS è 50 e 10 microsecondi, rispettivamente, mentre le lunghezze di RTS, CTS e ACK sono 20, 14 e 18 byte, rispettivamente. La stazione A vuole inviare un frame F di 800 byte alla stazione B . Indicare – giustificando la risposta – quale è la durata minima in microsecondi a cui verrà posto il NAV dalle stazioni che ricevono il CTS inviato da B senza avere ricevuto il RTS inviato da A .

Seconda parte (15 punti)

- E1.** Al tempo t_0 il TCP di un host A ha una connessione già stabilita, per la quale ha 2 segmenti full sized in volo e 2 MSS di nuovi dati da spedire, si trova nello stato di congestion avoidance, con $S_r=Y$, $ssthresh=4$ MSS, $cwnd=17/4$ MSS, e $rwnd=2$ MSS. Al tempo t_1 riceve un riscontro duplicato per la terza volta (ovvero riceve per la quarta volta lo stesso riscontro), al tempo t_2 riceve di nuovo un riscontro duplicato, al tempo t_3 scatta un timeout, mentre ai tempi t_4 e t_5 vengono ricevuti due riscontri non duplicati. Indicare – giustificando la risposta – lo stato del TCP e i valori di $ssthresh$, $cwnd$ e $rwnd$ subito dopo i tempi t_1 , t_2 , t_3 e t_4 , supponendo che nell'intervallo $[t_0, t_5]$ il TCP invii solo un segmento S contenente nuovi dati, subito dopo t_2 e che nell'intervallo $[t_0, t_5]$ scatti un solo timeout (in t_3).
- E2.** Descrivere con uno pseudocodice il comportamento dell'algoritmo Link State per inviare, ricevere e aggiornare le informazioni contenute nel suo Link-State Database. Utilizzare i comandi:

<i>receive()</i>	che restituisce un pacchetto LS e l'interfaccia su cui è stato ricevuto
<i>send(i,p)</i>	che invia sull'interfaccia i un pacchetto LS p
<i>LSDBlookup(p)</i>	che restituisce la copia di p già presente nell'LSDB (e restituisce NULL se il router non ha ricevuto p precedentemente)
<i>LSDBupdate(p)</i>	che aggiorna l'LSDB con p

e spiegare il significato di eventuali altri comandi utilizzati.

Traccia della soluzione

Q1. (a) Il PC di *mickey@disney.com* non riesce a stabilire una connessione TCP con il mailserver di *disney.com*, quindi nessun messaggio SMTP viene inviato o ricevuto. (b) Il PC di *mickey@disney.com* stabilisce una connessione TCP con il mailserver di *disney.com* su cui scambia i seguenti comandi SMTP:

```

<- 220 service ready -
- HELO ... ->
<- 250 OK -
- MAIL FROM: mickey@disney.com ->
<- 250 OK -
- RCPT TO: donnald@disney.com ->
<- 550 user unknown -
- QUIT ->
<- 221 service closed -
    
```

Q2. Dopo avere ricevuto R il sender potrebbe inviare da 0 a N+1 nuovi segmenti, a seconda di quale segmento viene riscontrato da R e di come sono disposti nel buffer i segmenti in volo. Nel caso pessimo infatti il sender non potrà inviare alcun nuovo segmento – per esempio se il segmento meno vecchio in volo occupa l’ultima posizione della finestra e R non riscontra il segmento più vecchio in volo. Nel caso ottimo invece il sender potrà inviare N+1 nuovi segmenti – per esempio se i segmenti in volo occupano le prime N posizioni della finestra, le successive N posizioni possono essere utilizzate per spedire nuovi segmenti e R riscontra il segmento più vecchio in volo.

Q3. a) Per inoltrare il datagram di 1010 byte ricevuto (990 byte di dati e 20 byte di intestazione), R1 invierà a R2 3 frammenti: (b) R2 frammenterà il primo e il secondo frammento ricevuto in due frammenti ciascuno, non dovrà invece frammentare il terzo:

	(byte dati)	offset	length
f1	480	0	500
f2	480	60	500
f3	30	120	50

	(byte dati)	offset	length
f1.1	240	0	260
f1.2	240	30	260
f2.1	240	60	260
f2.2	240	90	260
f3	30	120	50

Q4. Indichiamo con K_M il numero di slot da attendere scelto da A dopo avere rilevato una collisione nello slot M e sia R_N il numero scelto da A all’inizio dello slot N per la p-persistenza. A può riuscire a trasmettere F nello slot X+3 senza rilevare collisioni negli slot X+1 e X+2 solo se $(K_X=0 \wedge R_{X+1}>0.5 \wedge R_{X+2}>0.5 \wedge R_{X+3}\leq 0.5) \vee (K_X=1 \wedge R_{X+2}>0.5 \wedge R_{X+3}\leq 0.5) \vee (K_X=2 \wedge R_{X+3}\leq 0.5)$. Quindi A può riuscire a trasmettere con successo nello slot X+3 con probabilità minore uguale di $(\frac{1}{4} \times \frac{1}{2^3}) + (\frac{1}{4} \times \frac{1}{2^2}) + (\frac{1}{4} \times \frac{1}{2}) = \frac{7}{32} < \frac{1}{4}$.

Q5. Le stazioni che ricevono il CTS inviato da B senza avere ricevuto il RTS inviato da A imposteranno il NAV a:

$$SIFS + d_{trasm}(F) + d_{prop} + SIFS + d_{trasm}(ACK) + d_{prop} = (10 + 1280 + 2 + 10 + 28,8 + 2) \mu s = 1332,8 \mu s.$$

E1. Subito dopo t1 il TCP di A passa nello stato di fast recovery, ponendo ssthresh a 17/8 MSS e cwnd a 41/8 MSS, mentre rwnd non può essere maggiore di 2MSS dato che non vengono spediti nuovi dati. Subito dopo t2 il TCP pone cwnd=49/8 MSS, mentre rwnd non può essere maggiore di 2MSS+dimDati(S) dato che viene spedito solo il segmento S contenente nuovi dati. Subito dopo t3 il TCP passa nello stato di slow start, ponendo ssthresh a 49/16 MSS e cwnd a 1 MSS, mentre rwnd non può essere maggiore di 2MSS+dimDati(S) dato che non vengono spediti nuovi dati. Infine subito dopo t4 il TCP pone cwnd a 2 MSS, mentre rwnd non può essere superiore a 1 MSS + dimDati(S) o a dimDati(S) o a 0, a seconda del riscontro ricevuto. In sintesi:

Subito dopo t1	Subito dopo t2	Subito dopo t3	Subito dopo t4
stato=FR	stato=FR	stato=SS	stato=SS
ssthresh =17/8 MSS	ssthresh =17/8 MSS	ssthresh=49/16 MSS	ssthresh =49/16 MSS
cwnd=41/8 MSS	cwnd=49/8 MSS	cwnd=1 MSS	cwnd=2 MSS
rwnd≤2MSS	rwnd=2MSS+dimDati(S)	rwnd≤2MSS+dimDati(S)	R.ack=Y+1MSS → rwnd≤1MSS+dimDati(S) R.ack=Y+2MSS → rwnd≤dimDati(S) R.ack=Y+2MSS+ dimDati(S) → rwnd=0

E2. Assumiamo che le interfacce del nodo siano rappresentate dagli interi [0,N-1] e indichiamo con *welcome* il messaggio di “benvenuto” inviato dal nodo a tutti i suoi vicini e con *p.seqNum* il numero di sequenza del pacchetto p.

```

for (j=0; j<N; j++)
    send (j, welcome);
while true {
    <p, i>=receive();
    x=LSDBlookup(p);
    if ((x==null) || (x.seqnum<p.seqnum)) {
        LSDBupdate(p);
        for (j=0; j<N; j++)
            if (j!=i) send(j, p);
    }
}
    
```