RETI DI CALCOLATORI – prova scritta del 06/02/2014

Per essere ammessi alla prova orale è necessario ottenere una valutazione sufficiente sia della prima parte che dell'intera prova scritta.

Prima parte (10 punti)

Q1. Siano A e B due computer direttamente collegati da una linea di comunicazione lunga 60 Km, in cui la velocità di propagazione del segnale è $2x10^8$ m/sec e la frequenza di trasmissione è 100 Mbps. Supponiamo che i due computer si scambino frame di 1500 byte, utilizzando Go-Back-N per mascherare l'inaffidabilità della rete. Indicare – giustificando la risposta – quale è la dimensione massima della pipeline per Go-Back-N.

Q2. Un server Web S invia a un client C una pagina formata da tre oggetti, aventi dimensione 3,5 MSS, 2,3 MSS, e 4 MSS, rispettivamente. Supponiamo che S e C utilizzino HTTP 1.1 con connessioni TCP persistenti, e che il TCP di S si trovi nello stato di slow start con threshold uguale a 10 MSS e con valore della finestra di congestione *cwnd* uguale a 1 MSS. Supponiamo inoltre che tutti i segmenti siano riscontrati positivamente da C, e che tali riscontri arrivino tutti correttamente, e nell'ordine con cui sono stati spediti, a S. Indicare – giustificando la risposta – la successione dei valori assunti da *cwnd*. Assumere per semplicità che non scada alcun timeout.

Q3. Consideriamo un nuovo protocollo applicativo non "data loss tolerant", denominato *twixter,* che può essere utilizzato solamente per inviare messaggi di al più 180 caratteri. Indicare – giustificando la risposta – quale è il migliore protocollo di livello trasporto da utilizzare per *twixter*.

Q4. Indicare – giustificando la risposta – se e come CSMA/CA risolve il problema del terminale nascosto.

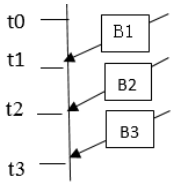
Seconda parte

E1 (5 punti). Considerare una versione "jitter-sensitive" di Go-Back-N, in cui il receiver attende di ricevere una copia non corrotta del segmento con numero di sequenza E entro un tempo uguale a L volte JJ istanti se l'ultimo segmento non corrotto ricevuto aveva numero di sequenza E-L. Ad esempio, se il primo segmento viene ricevuto non corrotto al tempo t0, il receiver attende di ricevere il secondo segmento entro il tempo t0+JJ.

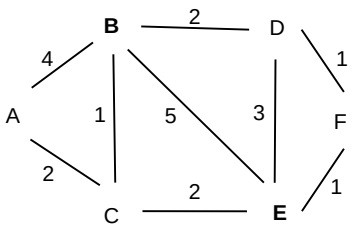
- Se il secondo viene ricevuto non corrotto al tempo t1<t0+JJ allora il receiver attende di ricevere il terzo segmento entro il tempo t1+JJ (ovvero L=1).
- Se il secondo *non* viene ricevuto non corrotto prima di t0+JJ allora il receiver passa ad attendere di ricevere il terzo segmento entro il tempo t0+2xJJ (ovvero L=2), e così via.

Inoltre, quando L assume valore maggiore o uguale di 3, il receiver invia al sender un NACK ogni volta che scade il timeout e passa ad attendere il successivo segmento, fino a quando non riceve il segmento atteso non corrotto. In tutti gli altri casi, il receiver si comporta normalmente. Descrivere – utilizzando un automa a stati finiti – il comportamento del receiver "jitter-sensitive" di Go-Back-N su descritto. (Assumere che il sender funzioni correttamente con un receiver come quello descritto sopra.)

E2 (5 punti). Supponiamo che al tempo t0 il TCP di un processo applicativo A riceva tre ACK non duplicati (B1, B2, B3) uno dopo l'altro. Supponiamo inoltre che, dopo aver ricevuto il terzo ACK B3, la finestra di congestione *cwnd* del TCP di A abbia valore $\frac{1405}{222}\ MSS$ e il TCP di A si trovi nello stato di congestion avoidance. Indicare – giustificando la risposta – i valori di *cwnd* e della soglia *ssthresh* ai tempi t0, t1 e t2, assumendo che il TCP di A non si trovi mai nello stato di fast recovery nell'intervallo [t0,t3].



E3 (5 punti). Consideriamo il sistema autonomo AS0 a lato, in cui B ed E sono gli unici due gateway e i cui nodi utilizzano OSPF come protocollo di routing intra-AS e non utilizzano alcuna preferenza locale per BGP. Supponiamo che al tempo t, sia B che E comunichino agli altri router di AS0 la raggiungibilità di due sistemi autonomi esterni AS1 e AS2. In particolare, B trasmette un advertisement di AS1 con |AS-PATH|=8 e uno di AS2 con |AS-PATH|=6, mentre E trasmette un advertisement di AS1 e uno di AS2 entrambi con |AS-PATH|=8. Indicare -giustificando la risposta- su quale collegamento A, C, D e F inoltrano i pacchetti destinati ad AS1 e ad AS2 dopo avere ricevuto tutti gli advertisement sopra menzionati.



E4 (5 punti). Una LAN utilizza un protocollo di tipo *slotted aloha* come protocollo MAC, in cui la banda di comunicazione non è utilizzata tutta per trasmettere dati, ma una parte viene utilizzata per inviare informazioni di controllo. In particolare, la comunicazione è formata da cicli di N slot: i primi M slot di ogni ciclo vengono usati per trasmettere solamente dati, mentre i restanti N-M vengono usati per trasmettere solamente informazioni di controllo. Descrivere – utilizzando un automa a stati finiti – il comportamento di un nodo che adotta tale protocollo, utilizzando l'evento *sendRequest(d)* per indicare la ricezione di una richiesta dal livello superiore e l'azione *send(f)* per spedire un frame f. Per semplicità supporre che il nodo descritto non abbia informazioni di controllo da spedire e non considerare il problema della gestione di eventuali collisioni.

Traccia della soluzione

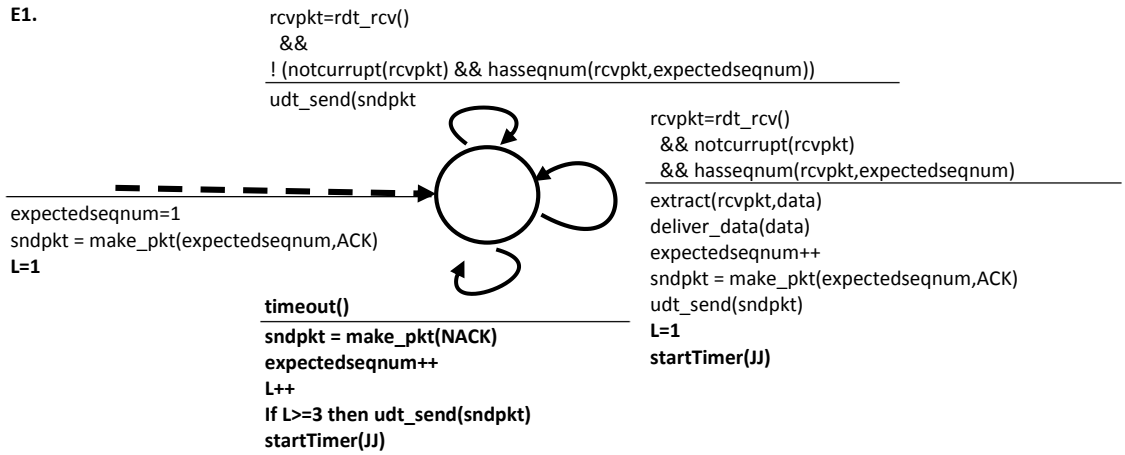
Q1. Il numero massimo n di bit che possono occupare simultaneamente il canale può essere determinato risolvendo la disequazione: $\frac{n}{100\times10^6} \leq \frac{1}{100\times10^6} + \frac{60\times10^3}{2\times10^8}$ ovvero n≤30001. Quindi, dato che A e B si scambiano frame di 1500 byte, la dimensione massima della pipeline (considerando solo segmenti "full-sized") per Go-Back-N è 2 dato che 2x1500x8<30001<3x1500x8.

Q2. C invierà a S i tre oggetti di dimensione 3,5 MSS, 2,3 MSS e 4 MSS in 4, 3 e 4 segmenti, rispettivamente. Il TCP di S incrementerà di 1 MSS la dimensione di *cwnd* ogni volta che riceverà un ACK di uno dei primi 9 segmenti, ottenendo così *cwnd*=10 MSS. Passato nello stato di congestion avoidance, il TCP di S aggiornerà quindi *cwnd* a (10+$\frac{1}{10}$) MSS = $\frac{101}{10}$ MSS quando riceverà l'ACK del decimo segmento e a ($\frac{101}{10}$+$\frac{10}{101}$) MSS = $\frac{10301}{1010}$ MSS quando riceverà l'ACK dell'undicesimo segmento.

Q3. Dipende da che cosa si intende con "migliore" protocollo di livello trasporto da utilizzare per Twixter:
- Scegliendo TCP, che garantisce l'affidabilità del trasferimento dati, lo sviluppo dell'applicazione è banale, a scapito dell'efficienza,
- Scegliendo UDP l'affidabilità del trasferimento dei dati deve essere garantita a livello applicazione (con riscontri, timeout e ritrasmissioni), a vantaggio dell'efficienza.

Q4. Mediante l'utilizzo di RTS e CTS. Se infatti B e C non si "vedono" tra loro ma entrambi "vedono" A, se B invia un RTS ad A e A risponde con un CTS, tale CTS raggiunge sia B che C.

E1.



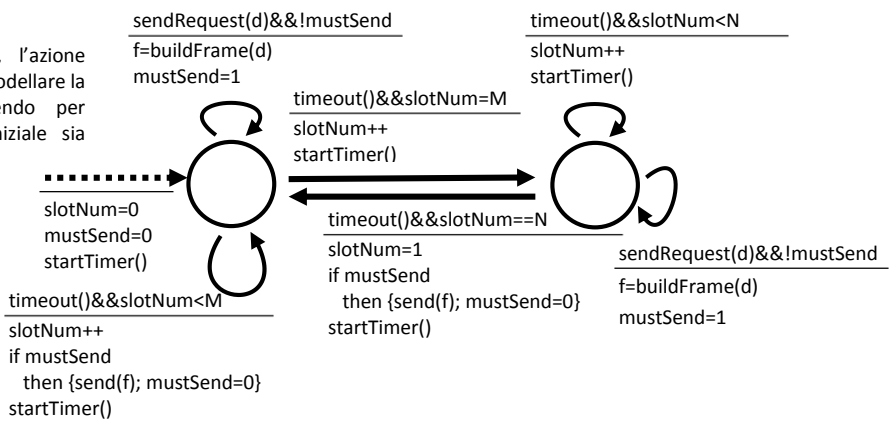
E2. Sappiamo[1] che $cwnd_{t2} + \frac{1}{cwnd_{t2}} = \frac{1405}{222}$ ovvero $cwnd_{t2}{}^2 - \frac{1405}{222} cwnd_{t2} + 1 = 0$ e quindi $cwnd_{t2} = \frac{37}{6}$. Analogamente $cwnd_{t1} + \frac{1}{cwnd_{t1}} = \frac{37}{6}$ e quindi $cwnd_{t1} = 6$. Quindi 5 < $ssthresh_{t1} \leq 6$ e $cwnd_{t0} = 5$ (e $ssthresh_{t0}$=$ssthresh_{t1}$=$ssthresh_{t2}$= $ssthresh_{t3}$).

E3. Dato che i nodi non utilizzano alcuna preferenza locale, per AS2 sceglieranno tutti il percorso pubblicizzato da B (che ha |AS-PATH| minore di quello pubblicizzato da |E|) e inoltreranno quindi i pacchetti destinati ad AS2 nel modo seguente:
- A sul collegamento AC,
- C sul collegamento CB,
- D sul collegamento DB,
- F sul collegamento FD.

Nel caso di AS1, dato che i percorsi pubblicizzati da B ed E hanno uguale |AS-PATH|, ogni nodo sceglierà il percorso pubblicizzato dal gateway meno distante:
- A sul collegamento AC,
- C sul collegamento CB,
- D sul collegamento DB o DF,
- F sul collegamento FE.

E4.

/* Utilizziamo la variabile slotNum, l'azione startTimer() e l'evento tiimeout() per modellare la sincronizzazione degli slot (assumendo per semplicità che la sincronizzazione iniziale sia garantita. */



(1) Alternativamente possiamo più semplicemente osservare che 6<$\frac{1405}{222}$<7 e che l'ultimo valore intero assunto da *cwnd* è quindi molto probabilmente stato 6. Notiamo che $6 + \frac{1}{6} = \frac{37}{6}$ e $\frac{37}{6} + \frac{6}{37} = \frac{1405}{222}$ e concludiamo quindi che $cwnd_{t1} = 6$, che $cwnd_{t2} = \frac{37}{6}$ e che $cwnd_{t0} = 5$ (e 5 < $ssthresh_{t0}$ = $ssthresh_{t1}$ = $ssthresh_{t2}$ = $ssthresh_{t3} \leq 6$).