

**E1 [6 punti].** Descrivere con un automa a stati finiti il modo in cui un peer risponde con un messaggio *query-hit* a una richiesta ricevuta su una rete di copertura Gnutella. Ignorando i problemi di autenticazione e assumendo per semplicità che le comunicazioni con l'autorità di certificazione avvengano su un canale sicuro, garantire utilizzando firme digitali l'integrità dei messaggi di richiesta e di risposta. Per la descrizione dell'automa utilizzare gli eventi:

`<Q,Qs,Id>=receiveQuery()` //per ricevere una query Q e la relativa firma Qs generate dal peer Id  
`cert=receiveCertificate()` //per ricevere un certificato dall'autorità di certificazione

e le azioni:

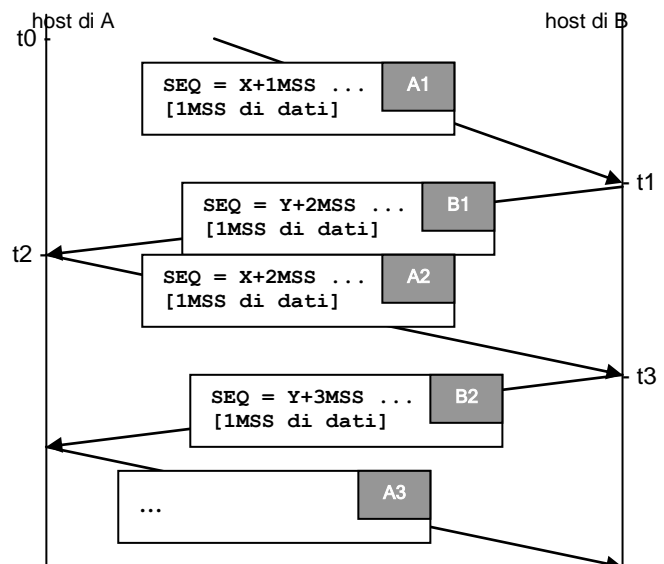
`res=lookUp(Q)` //per verificare se il peer possiede il file (`res!=NULL`) richiesto oppure no (`res=NULL`)  
`sendCertificateRequest(Id)` //per chiedere all'autorità di certif. un certificato contenente la chiave pubblica  $K_{id}^+$  di Id  
`sendQueryHit(Id,reply)` //per inviare la risposta al peer Id

**E2 [8 punti].** Consideriamo un'applicazione A che ha già stabilito una connessione TCP con un suo pari B. Supponiamo che al tempo  $t_0$ :

- il valore della finestra di congestione *CongWin* dell'host di A sia pari a  $1MSS$ , gli indici *sendBase* e *nextSeqNum* di A valgano rispettivamente  $X+1MSS$  e  $X+2MSS$ , e che, sempre al tempo  $t_0$ , l'host di A debba spedire  $1MSS$  di (nuovi) dati;
- il valore di *CongWin* dell'host di B sia pari a  $2MSS$ , gli indici *sendBase* e *nextSeqNum* di B valgano rispettivamente  $Y$  e  $Y+2MSS$ , e che, sempre al tempo  $t_0$ , l'host di B debba spedire  $2MSS$  di (nuovi) dati;
- entrambi i TCP si trovino in *slow start*.

Per semplicità assumiamo che né il pacchetto contenente A1 né nessuno dei pacchetti inviati dopo  $t_0$  venga corrotto né perso, che tutti i segmenti contenenti dati contengano  $1MSS$  di dati,

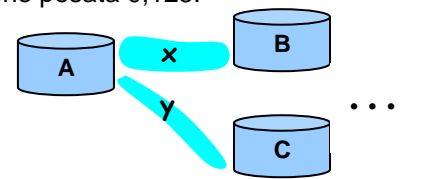
e che non si verifichi nessun timeout. Specificare, motivando la risposta, il valore dei campi ACK e *RcvWin* dei segmenti A1, B1, A2, B2 e il valore dei campi SEQ e ACK di A3.



**E3 [4 punti].** (a) Indicare, giustificando la risposta, di quanto deve essere incrementata la dimensione della finestra di congestione TCP ogni volta che viene ricevuto un riscontro positivo per aumentarne la dimensione di  $\frac{1}{2}MSS$  a ogni RTT. (b) Supporre che TCP aggiorni il valore di *EstimatedRTT*<sub>0</sub> tenendo conto di due nuove misurazioni *SampleRTT*<sub>1</sub> e *SampleRTT*<sub>2</sub>, entrambi pari a *EstimatedRTT*<sub>0</sub>+ $\epsilon$ . Indicare, giustificando la risposta, di quanto varierà il valore di *EstimatedRTT* se ogni nuova misurazione viene pesata 0,125.

**E4 [6 punti].** Considerare un sistema autonomo (una parte del quale è illustrata di lato) che utilizza RIP. Supponendo che la tabella di routing del nodo di A contenga la riga:

Destination	NextRouter	#Hops
w	B	5



indicare – giustificando la risposta – in che modo A aggiorna la propria tabella di instradamento se:

(a) A riceve un annuncio di B che pubblicizza distanza 5 per w e subito dopo riceve un annuncio di C che pubblicizza distanza k per w. (b) A riceve un annuncio di C che pubblicizza distanza k per w e subito dopo riceve un annuncio di B che pubblicizza distanza 5 per w.

**E5 [6 punti].** (a) Descrivere con un automa a stati finiti il comportamento del nodo principale di una rete che utilizza un protocollo di accesso multiplo a rotazione ("taking turns") di tipo *polling*. Per semplicità assumere che gli altri nodi della rete siano rappresentati dai numeri 1,2,...N, che il nodo principale non trasmetta frame di dati e che gli altri nodi possano trasmettere al più un frame di dati quando è il loro turno. Per descrivere l'automa usare le azioni:

`poll(i)` //per trasmettere un messaggio di "sondaggio" al nodo i  
`startTimer()` e `stopTimer()` //per (ri)avviare e per arrestare il timer (associato al sondaggio)

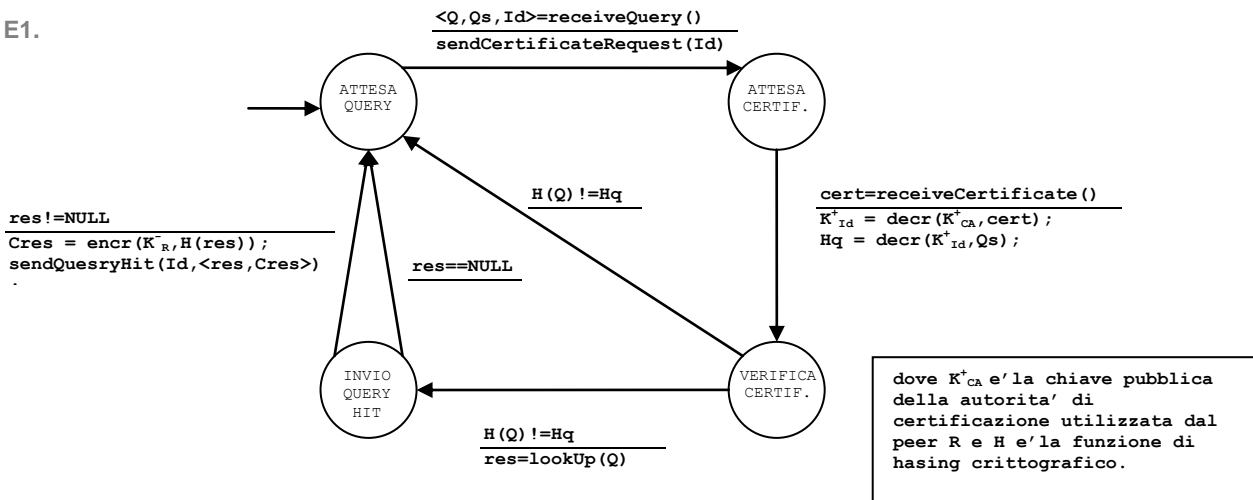
e gli eventi:

`timeout()` //per indicare lo scadere del timeout  
`signalOnChannel()` e `noSignalOnChannel()` //per rilevare la presenza e l'assenza di segnale sul canale

(b) Sia  $T_P$  il tempo necessario per trasmettere un messaggio di sondaggio, sia  $T_F$  il tempo necessario per trasmettere un frame di dati e sia  $T_{TO}$  la durata del timeout associato al sondaggio. Se  $T_P = T_{TO} = \frac{1}{2} T_F$  quanti devono essere al più i nodi A che hanno dati da trasmettere affinché il protocollo polling impieghi meno tempo di TDM per permettere a ciascun nodo di A di trasmettere un frame?

## TRACCIA DELLA SOLUZIONE

E1.



E2. Esaminiamo i diversi casi possibili.

(a) A1 potrebbe essere un riscontro cumulativo di tutti i dati spediti da B, ovvero  $A1.ACK=Y+2MSS$ . In tal caso  $A1.RcvWin=1MSS$  (altrimenti B non potrebbe spedire in  $t1$  1MSS di nuovi dati oppure potrebbe spedire piu' di 1 MSS di nuovi dati). B1 riscontre' i dati contenuti in A1, ovvero  $B1.ACK=X+2MSS$ , e  $B1.RcvWin \geq 1MSS$  (altrimenti A non potrebbe spedire in  $t2$  1MSS di nuovi dati). A2 riscontre' i dati contenuti in B1, ovvero  $A2.ACK=Y+3MSS$  e  $A2.RcvWin \geq 1MSS$  (altrimenti B non potrebbe spedire in  $t3$  1MSS di nuovi dati). B2 riscontre' i dati contenuti in A2, ovvero  $B2.ACK=X+3MSS$  e non possiamo dire niente sul valore di  $B2.RcvWin$ . Infine A3 sara' un riscontro positivo per i dati contenuti in B2, ovvero  $A3.ACK=Y+4MSS$ . A3 non conterra' dati e  $A3.SEQ=X+3MSS$ .

(b) A1 potrebbe essere un riscontro per il segmento piu' vecchio spedito da B e ancora "in volo" per B, ovvero  $A1.ACK=Y+1MSS$ . In questo caso  $A1.RcvWin=2MSS$  (per la stessa ragione menzionata nel caso (a)).  $B1.ACK=X+2MSS$  e  $B1.RcvWin \geq 1MSS$  come nel caso (a).

(b1) Se A ha ricevuto il segmento con  $SEQ=Y+1MSS$  prima di  $t0$  e il relativo riscontro e' andato perso, A2 e' un riscontro cumulativo di tutti i dati spediti da B, ovvero  $A2.ACK=X+3MSS$ . In tal caso  $A2.RcvWin \geq 1MSS$  come nel caso (a) e anche i segmenti B2 e A3 sono gli stessi che nel caso (a).

(b2) Se A non ha ricevuto in  $t2$  il segmento con  $SEQ=Y+1MSS$ , quando riceve B1 invia un riscontro negativo, ovvero  $A2.ACK=Y+1MSS$ , e  $A2.RcvWin \geq 3MSS$  (dato che B spedisce 1MSS di nuovi dati in  $t3$ ). Abbiamo quindi che  $B2.ACK=X+3MSS$  come nel caso (a). Infine A3 sara' di nuovo un riscontro negativo, ovvero  $A3.ACK=Y+1MSS$  (e  $A3.SEQ=X+3MSS$ ).

[Notiamo che A1 non puo' essere un riscontro negativo, ovvero  $A1.ACK$  non puo' essere Y, altrimenti il meccanismo di controllo di congestione non permetterebbe a B di spedire nuovi dati in  $t1$ .]

E3. (a) La dimensione di CongWin deve essere aumentata di  $\Delta = (MSS \cdot MSS) / (2 \cdot CongWin)$  ogni volta che viene ricevuto un riscontro positivo. In questo modo abbiamo infatti che:  $CongWin = \Delta \cdot CongWin / MSS = CongWin + MSS/2$ . (b) Il valore di EstimatedRTT aumentera' di  $(15/64)\epsilon$ . Infatti:

$$EstimatedRTT_2 = (1-\alpha)^2 EstimatedRTT_0 + (1-\alpha)\alpha SampleRTT_1 + \alpha SampleRTT_2 = EstimatedRTT_0 + (2\alpha - \alpha^2) \epsilon \text{ dove } \alpha = 1/8.$$

E4. (a) Quando riceve l'annuncio di B, A aggiorna la sua distanza per w in

Destination	NextRouter	#Hops
w	B	6

dato che la rotta di A per w passa proprio per B. Quando riceve l'annuncio di C, A aggiorna la sua distanza per w in

Destination	NextRouter	#Hops
w	C	k+1

solo se  $k < 5$ .

(b) Quando riceve l'annuncio di C, A aggiorna la sua distanza per w in

Destination	NextRouter	#Hops
w	C	k+1

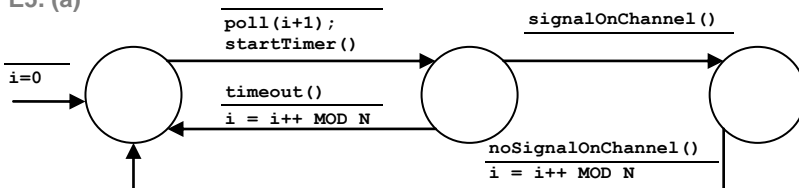
solo se  $k < 4$  e in tal caso quando riceve l'annuncio di B non modifica la sua tabella di instradamento.

Se invece  $k \geq 4$ , A non modifica la sua tabella di instradamento quando riceve l'annuncio di C ma modifichera' la sua distanza per w in

Destination	NextRouter	#Hops
w	B	6

dopo avere ricevuto l'annuncio di B.

E5. (a)



(b) Per permettere a ciascun nodo A di trasmettere un frame, il protocollo polling impieghera'  $t_{polling} = NT_P + (N-A)T_{TO} + AT_F$ . E' facile osservare che, nell'ipotesi che  $T_P = T_{TO} = \frac{1}{2} T_F$ ,  $t_{polling}$  sara' sempre superiore a  $t_{TDM} = NT_F$  necessario a TDM dato che  $t_{TDM} \leq t_{polling} \leq \frac{3}{2} t_{TDM}$ . Formalmente, osserviamo che la disequazione  $NT_P + (N-A)T_{TO} + AT_F \geq NT_F$  vale per qualunque valore di A  $\geq 0$ .

(AB)