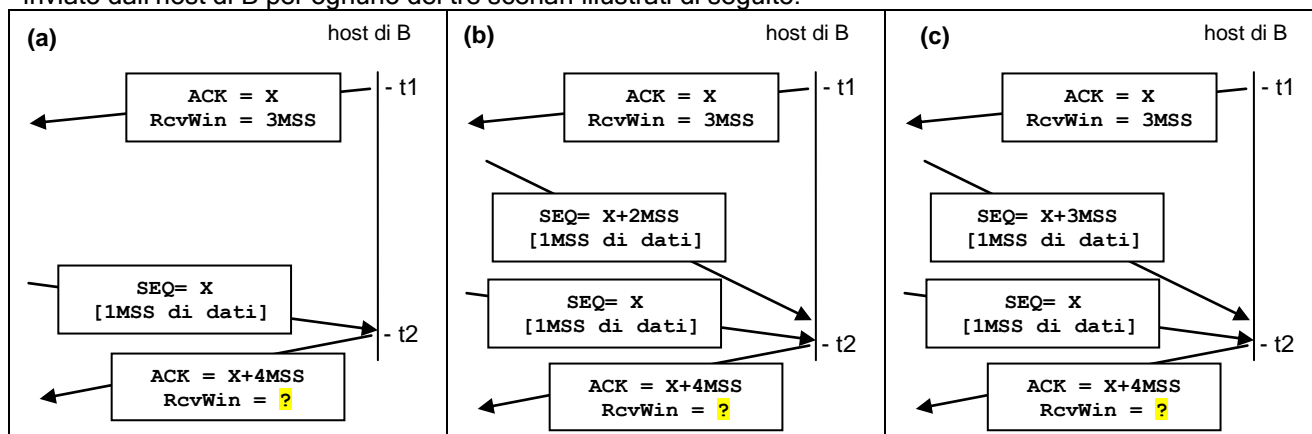


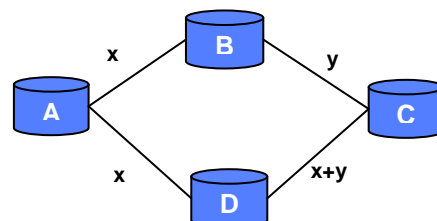
RETI DI CALCOLATORI - Secondo appello a.a. 2008/09

E1 (4 punti). Consideriamo una variante di TCP in cui viene introdotto un parametro `firstThreshold` in modo tale che quando TCP si trova nello stato di *slow start* e `CongWin` (`firstThreshold`, `Threshold`], il valore di `CongWin` sia incrementato a ogni RTT di al più $\frac{1}{2}$ `firstThreshold` (anziché di al più 1 `CongWin`). Descrivere come può essere implementata tale variante, motivando la risposta fornita.

E2 (7 punti). Consideriamo un'applicazione B che ha già stabilito una connessione TCP con un suo pari e supponiamo che nell'intervallo di tempo $[t_1, t_2]$ B riceva soltanto i segmenti indicati nella figura e non legga nessun dato dal buffer. Indicare, motivando la risposta, il valore del campo `RcvWin` dell'ultimo segmento inviato dall'host di B per ognuno dei tre scenari illustrati di seguito.



E3 (6 punti). Consideriamo la rete a lato, dove A,B,C,D sono router e x e y sono costi dei collegamenti. Supponiamo che i router usino un algoritmo di routing *distance vector* con *poisoned reverse*.



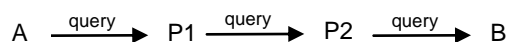
- Indicare i vettori delle distanze ricevuti da B e il vettore delle distanze calcolato da B una volta che tutti i router abbiano raggiunto lo stato di quiescenza S. Quanto deve valere y affinché B inoltri a C i pacchetti destinati a D?
- Se nello stato di quiescenza S, B rileva che il costo del suo collegamento con A è aumentato di $2x+2y$, a quale valore aggiorna la sua distanza per A nel suo vettore delle distanze?
- Se nello stato di quiescenza S, A rileva che il costo del suo collegamento con B è aumentato, di quanto deve essere aumentato tale costo affinché A decida di inoltrare a D i pacchetti destinati a C?

E4 (4 punti). Indicare le informazioni trasportate in un annuncio LSA ("Link State Advertisement") inviato in flooding da un router OSPF e tutte le azioni eseguite da un router OSPF quando riceve un annuncio LSA.

E5 (4 punti). Consideriamo una rete locale composta da 15 stazioni. Rispondere ai seguenti quesiti giustificando la risposta fornita.

- Supponiamo che il protocollo MAC utilizzato sia token-passing e che il tempo T_F necessario per spedire un frame di dati sia cinque volte il tempo T_T necessario per spedire il token. Affinché un nodo X riesca a spedire due frame di dati in al più $4T_F$ di tempo, quanto deve essere la probabilità che le altre stazioni inviino anche esse dati?
- Supponiamo che la rete locale in questione sia invece Ethernet, che due soli nodi abbiano un frame di dati da spedire e che tali nodi cerchino simultaneamente di spedire ciascuno un frame generando una collisione. Quale è la probabilità che la collisione non sia risolta con al più quattro tentativi di ritrasmissione?

E6 (5 punti). Consideriamo una rete Gnutella in cui il peer B è in grado di inviare un messaggio "query-hit" di successo in risposta alla query ricevuta.



Se B aprisse una connessione TCP con A per inviare tale risposta, assumendo che nessun pacchetto vada perso né corrotto:

- diminuirebbe il numero complessivo di pacchetti IP spediti per trasportare tale messaggio?
- A riceverebbe più velocemente il pacchetto IP contenente la risposta?

Giustificare la risposta fornita.

TRACCIA DELLA SOLUZIONE

E1. A ogni riscontro non duplicato ricevuto, se $\text{CongWin} \in (\text{firstThreshold}, \text{Threshold}]$, CongWin deve essere incrementata di una quantità K tale che:
 $\text{CongWin} + K \text{ CongWin}/\text{MSS} = \text{Congwin} + \frac{1}{2} \text{firstThreshold}$
 ovvero $K = \frac{1}{2} \text{firstThreshold}/\text{CongWin MSS}$.

E2. Osserviamo prima di tutto che in tutti e tre scenari gli scenari proposti la versione di TCP utilizzata dall'host di B evidentemente **bufferizza i dati ricevuti non in ordine** - altrimenti non potrebbe inviare il riscontro cumulativo $\text{ACK} = X + 4\text{MSS}$ subito dopo avere ricevuto i byte $[X, X + 1\text{MSS})$. Per determinare il valore di RcvWin inviato in t_2 consideriamo i possibili comportamenti della versione di TCP utilizzata dall'host di B.

- Supponiamo che la versione di TCP utilizzata dall'host di B calcoli la dimensione di RcvWin **considerando non occupata la porzione del buffer in cui ha memorizzato i byte ricevuti non in ordine**, ovvero che determini la dimensione di RcvWin utilizzando la formula:

$$\text{RcvWin} = \text{DimensioneBufferRicezione} - (\text{NextByteExpected} - \text{LastByteRead})$$

dove LastByteRead indica il numero dell'ultimo byte che è stato letto dal processo applicativo e NextByteExpected indica il numero del byte successivo all'ultimo riscontro. Se così fosse, in tutti e tre gli scenari la dimensione di RcvWin dovrebbe ridursi di 4MSS in t_2 (dato che NextByteExpected passa da X in t_1 a $X + 4\text{MSS}$ in t_2), ma ciò non è possibile dato che il valore di RcvWin in t_1 è 3MSS . Non può pertanto essere questo il modo in cui la versione di TCP utilizzata dall'host di B calcola la dimensione di RcvWin .

- Supponiamo allora che la versione di TCP utilizzata dall'host di B determini la dimensione di RcvWin **calcolando esattamente lo spazio occupato da tutti i dati ricevuti e memorizzati nel buffer e che non sono ancora stati letti dal processo applicativo**.
 - Nello scenario (a), dato che i byte $[X, X + 1\text{MSS})$ erano evidentemente gli unici byte dell'intervallo $[X, X + 4\text{MSS})$ che ancora non erano stati ricevuti dal TCP di B in t_1 , il valore di RcvWin in t_2 diminuisce di 1MSS rispetto al valore che aveva in t_1 .
 - Nello scenario (b) dobbiamo distinguere due casi:
 - (b1) se TCP aveva già ricevuto (prima di t_1) i byte $[X + 2\text{MSS}, X + 3\text{MSS} - 1]$ allora, come nel caso precedente, il valore di RcvWin in t_2 diminuisce di 1MSS rispetto al valore che aveva in t_1 ;
 - (b2) se viceversa se TCP non aveva ancora ricevuto i byte $[X + 2\text{MSS}, X + 3\text{MSS} - 1]$ allora il valore di RcvWin in t_2 diminuisce di 2MSS rispetto al valore che aveva in t_1 .
 - Anche nello scenario (c) dobbiamo distinguere due casi:
 - (c1) se TCP aveva già ricevuto (prima di t_1) i byte $[X + 3\text{MSS}, X + 4\text{MSS})$ allora il valore di RcvWin in t_2 diminuisce di 1MSS rispetto al valore che aveva in t_1 ;
 - (c2) se viceversa se TCP non aveva ancora ricevuto i byte $[X + 3\text{MSS}, X + 4\text{MSS})$, allora il valore di RcvWin in t_2 diminuisce di 2MSS rispetto al valore che aveva in t_1 .
- Alternativamente possiamo supporre che la versione di TCP utilizzata dall'host di B determini la dimensione di RcvWin **considerando occupata tutta la porzione del buffer compresa tra $\text{LastByteRead} + 1$ e il byte $\text{HighestByteReceived}$ dal numero più alto tra quelli memorizzati e non ancora letti dal processo applicativo**.
 - Negli scenari (a) e (b) il valore di RcvWin non varia da t_1 a t_2 , dato che né LastByteRead né $\text{HighestByteReceived}$ variano da t_1 a t_2 .
 - Nello scenario (c) dobbiamo distinguere due casi:
 - (c1) se TCP aveva già ricevuto (prima di t_1) i byte $[X + 3\text{MSS}, X + 4\text{MSS})$ il caso è analogo ai due precedenti e il valore di RcvWin non varia da t_1 a t_2 ;
 - (c2) se invece TCP non aveva ancora ricevuto i byte $[X + 3\text{MSS}, X + 4\text{MSS})$ allora il valore di RcvWin in t_2 diminuisce di 1MSS rispetto al valore che aveva in t_1 , dato che il valore di $\text{HighestByteReceived}$ passa da $X + 3\text{MSS} - 1$ in t_1 a $X + 4\text{MSS} - 1$ in t_2 .

E3. (a)

| | | | |
|---|-----|-----|-----|
| | A | C | D |
| A | 0 | inf | x |
| C | inf | 0 | x+y |
| B | x | y | w |

 dove $w = \min(2x, x + 2y)$. Affinche' B inoltri a C i pacchetti destinati a D deve valere: $x + 2y < 2x$ ovvero $y < x/2$ [oppure $y = x/2$ se B ha ricevuto l'advertisement di C prima di quello di A].

(b) B deve aggiornare la sua distanza per A a $3x + 2y$.

(c) Affinche' A decida di inoltrare a D i pacchetti destinati a C, il costo del collegamento AB deve aumentare di piu' di X . Infatti $x + \Delta + y > 2x + y$ se $\Delta > x$.

E4. Un annuncio LSA in OSPF contiene l'identificatore del nodo N che ha creato l'annuncio, i costi dei collegamenti tra N e i suoi vicini, il numero di sequenza e il tempo di vita dell'annuncio. Quando un router OSPF riceve un annuncio A, controlla il numero di sequenza dell'annuncio. Se il router ha precedentemente ricevuto un altro annuncio dallo stesso mittente con numero di sequenza più alto, scarta l'annuncio A. In caso contrario il router aggiorna la sua rappresentazione della topologia del sistema autonomo con i costi dei collegamenti riportati nell'annuncio e, se qualche costo è variato, ricalcola i suoi cammini minimi usando l'algoritmo di Dijkstra. Infine, se il tempo di vita dell'annuncio non è scaduto, inoltra l'annuncio agli altri suoi vicini.

E5. (a) Non è possibile che un nodo riesca a trasmettere due frame di dati in $4T_F$ di tempo. Infatti, anche supponendo che nessuna altra stazione abbia dati da spedire, il tempo necessario affinché il token torni a R è $15T_T$ ovvero $3T_F$. (b) $1/2 \times 1/4 \times 1/8 \times 1/16 = 1/1024$.

E6. (a) Se B aprisse una connessione TCP direttamente con A, il numero complessivo di pacchetti IP spediti diminuirebbe. Nella variante considerata B potrebbe infatti inviare la risposta come dati in piggybacking nel terzo segmento dell'handshake e verrebbero quindi spediti complessivamente soltanto 4 pacchetti IP (il quarto conterrebbe il riscontro dei dati inviato da A). In Gnutella verrebbero invece spediti 6 pacchetti IP, due per ogni hop del cammino inverso della rete overlay.

(b) Non è invece possibile determinare a priori se A riceverebbe più velocemente il pacchetto IP contenente la risposta nel caso in cui B aprisse una connessione TCP direttamente con A. Da un lato, a parte il minor numero di pacchetti IP, il numero di hop fisici da B ad A potrebbe essere minore del numero di hop fisici corrispondenti al cammino B-P2-P1-A della rete overlay e verrebbe evitato l'overhead introdotto dall'intervento del livello application nei nodi intermedi. D'altra parte, A dovrebbe rispondere alle richieste di connessione di tutti i peer in grado di rispondere alla query, potendosi così trovare a dover gestire un alto numero di richieste di connessione.

(AB)