RETI DI CALCOLATORI – 26.06.2013

Per ottenere una valutazione sufficiente all'ammissione alla prova orale è necessario ottenere una valutazione sufficiente della prima parte.

Prima parte (10 punti)

Q1. Consideriamo un messaggio m di posta elettronica avente un unico destinatario D. Indicare, giustificando la risposta, se è possibile o meno che per il trasferimento di m dall'host del mittente al mailserver contenente la mailbox di D vengano create più di 2 connessioni TCP.

Q2. Un sender Selective Repeat con dimensione della finestra uguale a 4 e con 3 segmenti "in volo" riceve un riscontro non duplicato R. Indicare – giustificando la risposta – se è possibile o meno che tale sender **non** possa inviare nuovi segmenti dopo avere ricevuto R.

Q3. Indicare, giustificando la risposta, che cosa succederebbe se il segmento SYNACK di TCP **non** specificasse nessun valore nel campo RcvWin.

Q4. Sia R una rete stub connessa a due reti di provider A e B. Supponiamo che un gateway di R riceva da un router di B un advertisement BGP contenente una rotta per un prefisso P. Indicare, giustificando la risposta, in quali casi e come R propagherà l'informazione ricevuta, nonché i principali attributi della rotta stessa.

Seconda parte

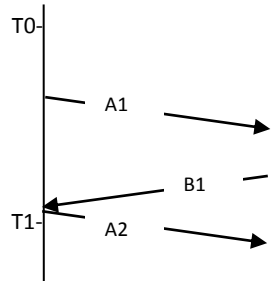
E1 (5 punti). Descrivere, utilizzando un automa a stati finiti, il comportamento di un server FTP che può ricevere, come previsto dal protocollo FTP, i comandi USER, PASS, LIST, RETR, STOR e QUIT ma che, diversamente da quanto previsto dal protocollo FTP, accetta al più un comando STORE. Per semplicità, ignorare nella modellazione le risposte inviate dal server e supporre di avere a disposizione gli eventi USER, PASS, LIST, RETR, STOR e QUIT.

E2 (6 punti). Supponiamo che al tempo T0 il TCP di un processo applicativo A abbia 2 MSS di dati in volo, che il valore di *sendBase* sia X, che la dimensione della sua finestra di congestione sia 3MSS e che l'ultimo valore di RcvWin che ha ricevuto dal suo pari sia 3MSS. Indicare – giustificando la risposta:

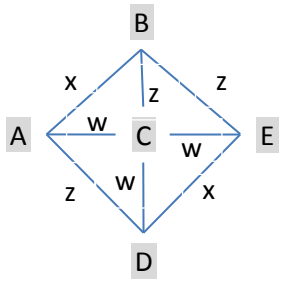
- i possibili valori di AckNum e di RcvWin contenuti in B1,
- i possibili valori di SeqNum contenuti in A1 e A2

nel caso in cui il TCP di A si trovi (a) nello stato di *slow start* e (b) nello stato di *congestion avoidance*.

Per semplicità, supponiamo che tutti i segmenti contenenti dati scambiati da A e B contengano 1 MSS di dati e che non scatti nessun timeout nell'intervallo [T0,T1].



E3 (5 punti). Si consideri la rete a lato, in cui i nodi utilizzano l'algoritmo distance vector con poisoned reverse, e in cui i valori di x, w, z sono interi positivi. Determinare – giustificando la risposta - quali vincoli devono valere tra i valori di x, w e z affinché, quando la rete ha raggiunto lo stato di quiescenza, i vettori che D ha ricevuto dai suoi vicini contengano il **massimo** numero possibile di distanze avvelenate.



E4 (4 punti). Supponiamo di avere a disposizione l'operazione *encrypt(k,t)* per criptare un messaggio *t* utilizzando una chiave *k*, oltre alle operazioni per l'invio e la ricezione di messaggi. Descrivere – con uno pseudo-codice – il comportamento di un'entità *A* che, avendo generato una chiave di sessione K_S non ancora nota a *B*, invia a *B* tale chiave di sessione e quindi un messaggio *m* in modo da garantire, utilizzando (anche) K_S, la sua autenticazione, nonché segretezza e integrità di *m*. Assumiamo che inizialmente A conosca la chiave pubblica K^+_B di *B*, che *B* conosca la chiave pubblica K^+_A di *A* e che *A* e *B* non condividano altre informazioni.

Traccia della soluzione

Q1. Si, ciò avviene per esempio se l'host del mittente trasferisce m al mailserver del mittente e quest'ultimo trasferisce m a un mailserver intermedio.

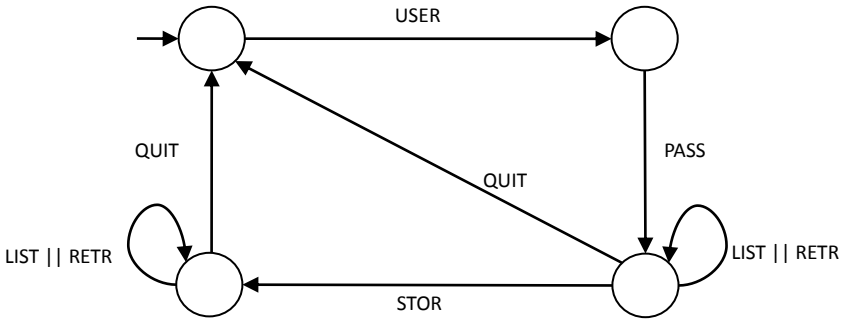
Q2. Si. Considerando le quattro possibili situazioni (illustrate a lato) in cui può trovarsi il buffer di spedizione quando il sender riceve R, ciò accade nei primi tre casi se *R.ackNum>sendBase*.

sendBase	*sendBase+1*	*sendBase+2*	*sendBase+3*
in-flight	acked	in-flight	in-flight
in-flight	in-flight	acked	in-flight
in-flight	in-flight	in-flight	acked
in-flight	in-flight	in-flight	unsent

Q3. Ciò non creerebbe in realtà problemi. Se infatti TCP quando invia un segmento SYNACK non comunicasse con tale segmento la dimensione iniziale della sua finestra di ricezione, il pari potrebbe in ogni caso spedire al più 1 MSS di dati (per il meccanismo di controllo della congestione) e prima di poter inviare altri nuovi dati dovrebbe in ogni caso attendere un riscontro (che potrebbe quindi contenere il valore iniziale di RcvWin).

Q4. Un gateway G di R che riceve da un gateway di B un advertisement contenente una rotta per P accetta o meno tale rotta in base alle proprie politiche di importazione. Se la accetta, G propaga la rotta a tutti gli altri router di R con cui ha una connessione iBGP. I due attributi più importanti associati alla rotta sono AS-PATH, che indica la sequenza degli AS attraversati dall'advertisement, e NEXT-HOP, che indica l'interfaccia del gateway dell'ultimo AS in AS-PATH.

E1. L'automa a lato modella il comportamento del server per le sequenze corrette di comandi[1].



E2. Dato che A1 trasporta nuovi dati spediti: A1.SeqNum=X+2MSS. Analizziamo quindi i possibili casi per B1:

- se B1 è un riscontro non duplicato, ovvero B1.AckNum=X+n·MSS, con n∈[1,3], allora A2 contiene nuovi dati, ovvero A2.SeqNum=X+3MSS, e al momento dell'invio di A2 deve valere min(RcvWin,CongWin)≥(4-n)MSS ovvero, dato che dopo la ricezione di B1 CongWin>3MSS, deve valere: B1.RcvWin)≥(4-n)MSS;
- se invece B1 è un riscontro duplicato, ovvero B1.AckNum≤X, ricevuto per la terza volta[2] allora A2 è la ritrasmissione veloce del segmento più vecchio ancora in volo, ovvero A2.SeqNum=X, indipendentemente dal valore di B1.RcvWin.

Tutte le precedenti osservazioni valgono qualunque sia lo stato in cui si trova il TCP di A.

E3. I vettori ricevuti da D conterranno il minimo dei valori indicati in ogni casella della tabella a lato, dove i valori evidenziati in grassetto causerebbero l'avvelenamento della distanza.
Quindi, affinché i vettori ricevuti da D contengano il massimo numero ossibile di distanze avvelenate, devono valere i vincoli: $\begin{cases} 2w > x + z \\ w + 2z > x + z \end{cases}$ ovvero $\begin{cases} 2w > x + z \\ w + z > x \end{cases}$.

	A	B	C	E
A	-	$min\begin{pmatrix} x, \\ w+z \end{pmatrix}$	$min\begin{pmatrix} w, \\ x+z \end{pmatrix}$	$min\begin{pmatrix} 2w, \\ w+2z, \\ \boldsymbol{x+z} \end{pmatrix}$
C	$min\begin{pmatrix} w, \\ x+z \end{pmatrix}$	$min\begin{pmatrix} z, \\ x+w \end{pmatrix}$	-	$min\begin{pmatrix} w, \\ 2z \end{pmatrix}$
E	$min\begin{pmatrix} 2w, \\ w+2z, \\ \boldsymbol{x+z} \end{pmatrix}$	$min\begin{pmatrix} z, \\ x+2w \end{pmatrix}$	$min\begin{pmatrix} w, \\ 2z \end{pmatrix}$	-

E4. Assumendo di disporre dell'operazione *send* per inviare messaggi e di una funzione di hashing *h* nota a B[3], il comportamento di A può essere descritto dal seguente pseudo-codice:

send (encrypt (K^+_B, encrypt(K^-_A, K_S)));
send (encrypt (K_S, (m, h(m))));

[1] Le risposte inviate dal server sono omesse (come suggerito dal testo dell'esercizio).

[2] B1 non può essere un riscontro duplicato ricevuto non per la terza volta, dato che in tal caso non vi sarebbe ritrasmissione veloce e il controllo di congestione impedirebbe di spedire nuovi dati.

[3] In caso contrario il secondo messaggio può essere cifrato come *encrypt (K_S, (m, encrypt (K^-_A. m))).*