

RETI DI CALCOLATORI – prova scritta del 15/06/2017

Per essere ammessi alla prova orale è necessario ottenere una valutazione sufficiente della prima parte e una votazione totale di almeno 15.

Prima parte (15 punti)

Q1. In un host H, situato in Tahiti, subito dopo essere stato acceso, e come prima azione, al tempo T_1 viene attivato il browser che richiede per la prima volta la pagina web <https://www.di.unipi.it/fr> (che non risiede nella cache del browser) e, al tempo T_2 riceve la risposta *404-page not found*. Quanti segmenti TCP viaggiano in internet tra T_1 e T_2 , supponendo che H sia collocato in una LAN che utilizza un proxy web di LAN, e che non si inviino mai dati in piggybacking? Giustificare la risposta.

Q2. Il TCP di un host A, dopo aver instaurato una connessione con un suo pari B, invia, tra i tempi t_0 e t_1 , due segmenti e misura per essi i seguenti RTT_M (espressi in secondi): 4 e 2. Se i valori di RTT_S , RTT_D , α , β al tempo t_0 sono rispettivamente di 4, 3, $1/6$ e $1/3$, quanto vale RTO al tempo t_1 ? Giustificare la risposta.

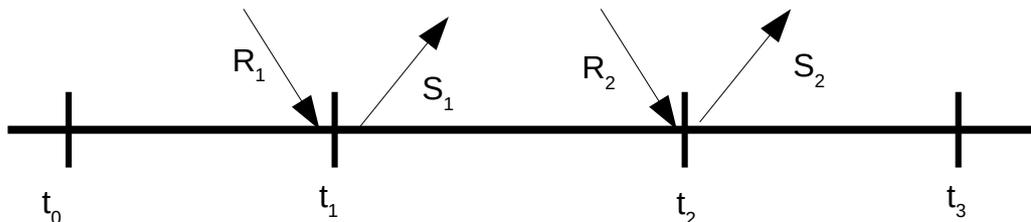
Q3. Un hacker H vuole scoprire quale protocollo di finestra scorrevole viene utilizzato dal sender S per mandare pacchetti al receiver R. Per questo, H “sniffa” (cioè legge) tutti i pacchetti che S ed R si scambiano. Da tale lettura, riesce H a stabilire con certezza se S ed R utilizzano Go-Back -N oppure Selective Repeat? Giustificare la risposta.

Q4. Si consideri un AS formato da 13 router, 17 reti locali e 94 host. Quante connessioni TCP vengono instaurate per generare le sessioni iBGP interne all’AS necessarie al funzionamento di iBGP? Giustificare la risposta.

Q5. In una rete locale Slotted Aloha, il cui protocollo MAC utilizza il metodo della p-persistenza, il frame P del nodo A subisce per la seconda volta una collisione nello slot X. Sapendo che la probabilità che A trasmetta P per la terza volta nello slot $X+3$ è $2/9$, e che le collisioni vengono notificate nello slot successivo a quello in cui si verificano, indicare – giustificando la risposta – qual’è il valore utilizzato dal protocollo per la probabilità p della p-persistenza.

Seconda parte (15 punti)

E1 (7 punti). Al tempo t_0 , il TCP di un host H ha una connessione già stabilita, per la quale ha 4 segmenti full sized in volo con $S_f=Y$, 2 MSS di dati nuovi da spedire, $cwnd=4$ MSS. Dal tempo t_0 al tempo t_3 H riceve e invia i segmenti indicati nella figura sotto, dove in R_1 si ha $ACK=Y$ e $rwnd=4MSS$, e in S_2 $SEQ=Y+4MSS$ e contiene un MSS di dati. Si supponga che tra t_0 e t_3 non scada nessun timeout, che H non spedisca ulteriori segmenti, e che tutti i segmenti inviati da H prima di t_0 siano full sized. Indicare, giustificando la risposta, il valore di SEQ nel segmento S_1 , i valori di ACK e $rwnd$ di R_2 ed i valori di $cwnd$, S_f e S_n e lo stato del TCP di H al tempo t_3 .



E2 (8 punti). Descrivere, con un pseudocodice, le azioni svolte da un router R_g non di confine appartenente ad un sistema autonomo AS_z di transito, che utilizza RIP come protocollo di routing intra-AS, per inserire od aggiornare nella sua tabella di inoltro globale *tabin*, le informazioni relative alla raggiungibilità di un sistema autonomo AS_x , secondo quanto stabilito dal protocollo iBGP. Il messaggio *mess* che contiene tali informazioni, ha i campi: *mess.AS*, contenente il nome del sistema autonomo destinazione (AS_x in questo caso), *mess.reti*: lista delle reti appartenenti ad AS_x , *mess.routconf* identificativo del router di confine di AS_z a cui inoltrare i datagram destinati alle reti di AS_x , e *mess.ASpl*, che indica la lunghezza del cammino per raggiungere AS_x , secondo la metrica usata da BGP. La tabella *tabin* ha (tra le altre) le colonne *dest* (che contiene la rete destinazione), *nexthop* (che contiene il router a cui inoltrare i datagram destinati alla rete in dest), *ASpl* che contiene la lunghezza del cammino per raggiungere AS_x , e *costo*, che contiene il costo (secondo la metrica normalmente usata in iBGP) per raggiungere le reti destinazione. Inoltre, ogni router ha il vettore *distr* che contiene la distanza tra lui e tutti gli n router del suo sistema autonomo. Si supponga, per semplicità, che i router vicini di R_g siano rappresentati dai numeri interi compresi tra 1 ed m, che AS_x contenga una sola rete, che AS_z non utilizzi nessuna preferenza locale, e che, a parità di tutti gli altri attributi, si scelga l’ultimo percorso ricevuto. Si utilizzino le procedure *receive (mess, neigh)* per ricevere il messaggio *mess*, *neigh* è il vicino che l’ha mandato; e *lookup (tabin, rete, i)* che cerca “rete” in *tabin*. Se la trova, *i* è l’indice di *tabin* in cui si trova; altrimenti *i* è negativo. Infine, si supponga che *tabin* sia sovradimensionata, e che *lasttabin* sia l’indice dell’ultimo elemento di *tabin* che contiene informazioni valide (cioè, *tabin* ha informazioni valide comprese tra gli elementi 0 e *lasttabin*).

Q1. 3 per l'apertura della connessione TCP con il proxy +1 per l'invio della GET HTTP al proxy +1 per l'ACK della GET di H +3 per l'apertura della connessione TCP tra il proxy e www.di.unipi.it +1 per l'invio della GET dal proxy verso www.di.unipi.it + 1 per l'ACK della GET del proxy +1 per la ricezione della risposta 404 +1 per l'ACK del proxy verso il web server di di.unipi.it, + 1 per l'invio (da parte del proxy) della risposta 404 ad H +1 per l'ACK di H verso il proxy. A questi vanno aggiunti X segmenti per il calcolo dell'indirizzo IP del proxy e di www.di.unipi.it mediante DNS (X=0 se DNS usa UDP, cioè se i messaggi DNS sono non più lunghi di 512 byte). Quindi, in totale 14+X ipotizzando, (come è altamente probabile) che GET e risposte non superino 1 MSS in lunghezza.

Q2. Dato che $RTT_s = (1-\alpha)RTT_s + \alpha RTT_M$, che $RTT_D = (1-\beta)RTT_D + \beta |RTT_s - RTT_M|$ e che $RTO = RTT_s + 4RTT_D$, si ha che RTT_s tra il tempo t_0 e il tempo t_1 assume i valori $5/6 \times 4 + 1/6 \times 4 = 4$ e $5/6 \times 4 + 1/6 \times 2 = 11/3$, mentre RTT_D assume i valori $2/3 \times 3 + 1/3 \times 0 = 2$ e $2/3 \times 2 + 1/3 \times (11/3 - 2) = 4/3 + 5/9$ si ha che RTO al tempo t_1 vale $11/3 + 4 \times 17/9 = 101/9$.

Q3. No, perché se, ad esempio, nessun pacchetto viene perduto oppure viene corrotto, i due protocolli si comportano allo stesso modo.

Q4. Ogni sessione iBGP utilizza una connessione TCP sulla porta 179. Per far funzionare iBGP si deve instaurare una sessione per ogni coppia di router dell' AS, quindi avremo $13 \times 12 / 2 = 13 \times 6 = 78$ connessioni TCP.

Q5. A trasmette in X+3 se $K(A) = 0$ e per una volta genera un valore R per la p-persistenza superiore a p, e un valore non superiore a p subito dopo, oppure se $K(A) = 1$ ed R è minore od uguale a p. Quindi, $1/4((1-p)p) + 1/4p = 2/9$, cioè $1/4(2p-p^2) = 2/9$ e pertanto $p^2/4 - p/2 + 2/9 = 0$ che ha due radici: 4/3 (che essendo >1 non è valida) e 2/3. Pertanto, $p = 2/3$.

E1. Al tempo t_0 , H non può spedire nuovi dati perché ha raggiunto il limite di cwnd e di rwnd, quindi S_1 è una ritrasmissione del segmento più vecchio della finestra, ed R_1 è il terzo duplicato del riscontro del segmento precedente. Pertanto, SEQ di $S_1 = Y$, ed il TCP di H passa nello stato di *fast recovery*, ponendo ssthresh a 2 e cwnd a 5. Dato che S_2 contiene nuovi dati, R_2 è un riscontro non duplicato, alla sua ricezione il TCP di H passa in congestion avoidance ponendo $cwnd = ssthresh = 2$. Ne segue che il campo ACK di $R_2 = Y+3$ (altrimenti la finestra si vuoterebbe e invieremmo 2 MSS di nuovi dati e non 1 MSS) mentre il campo rwnd è maggiore od uguale a 2 MSS. Il caso in cui il campo ACK di $R_2 = Y+4$ e si invia un solo MSS di dati nuovi implicherebbe che $rwnd = 1MSS$ in R_2 e quindi implicherebbe che il sistema operativo del ricevente ha tolto al TCP spazio utilizzato per la finestra (infatti, inizialmente lo spazio riservato per tutta la finestra di ricezione era di almeno 4 MSS byte, e sarebbe diventato di 1 MSS byte), cosa altamente improbabile, se non impossibile. Al tempo t_3 si ha che lo stato di TCP è congestion avoidance, $S_r = Y+3$ MSS, $S_n = Y+5$ MSS, e $cwnd = 2$ MSS.

E2.

```

receive (mess, neigh); //per ricevere il messaggio mess; neigh è il vicino che l'ha mandato
lookup (tabin, mess.rete, i); // cerca "mess.rete" in tabin. Se lo trova, i è l'indice di tabin in cui si trova; else i è negativo
if ((i>=0)&&(mess.Aspl<=tabin[i].Aspl))
    { tabin[i].Aspl=mess.Aspl;
      tabin[i].nexthop=neigh;
      tabin[i].costo=distr(mess.routconf)+1;
    }
else if (i<0)
    { lasttabin++;
      i=lasttabin;
      tabin[i].Aspl=mess.Aspl;
      tabin[i].nexthop=neigh;
      tabin[i].costo=distr[mess.routconf]+1;
      tabin[i].dest=mess.rete;
    }

```