RETI DI CALCOLATORI – prova scritta dell' 11/04/2017

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

Prima parte (15 punti)

Q1. Tizio manda dal suo account di email (tizio@libero.it) un messaggio di posta elettronica con testo di 256 caratteri a Caio (caio@occupato.it). Specificare il contenuto dei primi due messaggi TCP *inviati* dal mailserver di libero.it, ms.libero.it, per ricevere tale messaggio. Specificare, per ciascun segmento, payload, numero di sequenza, ack number, flags posti ad 1, porta origine e porta destinazione.

Q2. E' possibile che in una connessione TCP tra un client C ed un server S, S abbia in volo (inviati e non ancora riscontrati) X byte di dati, mentre l'ultimo valore di rwnd che S ha ricevuto da C è Y, con X>Y? Giustificare la risposta.

Q3. Il server DNS di *unipi* vuole inviare al server top level del dominio .org una query relativa al nome simbolico *www.moma.org* . Per motivi di sicurezza, si vuole che la richiesta abbia la proprietà della riservatezza. Quali protocolli devono essere utilizzati? Giustificare la risposta.

Q4. Si consideri una rete IEEE 802.11 in cui, al tempo t, un host A trasmette per la prima volta un RTS destinato all'host B, che però è spento, e rimarrà spento nelle successive 2 ore, e quindi non riceve il frame. Supponendo di trascurare i tempi di elaborazione, che A sia l'unico host che deve trasmettere, che la frequenza di trasmissione effettiva della rete sia di 8 Mbps, che i frame RTS e CTS siano lunghi 20 byte, che DIFS sia di 50microsecondi e SIFS sia di 16 microsecondi, che il tempo di slot (tempo da moltiplicare per il numero casuale generato) sia il tempo per trasmettere 512 bit, che il valore del timeout per decidere che il CTS relativo ad un RTS non è stato inviato è uguale a due volte il tempo che intercorre tra l'inizio dell'invio di un RTS e il tempo di ricezione del CTS ad esso relativo, che il raggio di trasmissione sia di 400 metri e la velocità di propagazione sia di $2 \cdot 10^8$m/s, e che il limite per dichiarare che la comunicazione è fallita, e quindi interromperla è 16, qual'è il tempo *minimo* (a partire da t), impiegato da A per dichiarare fallimento? Giustificare la risposta.

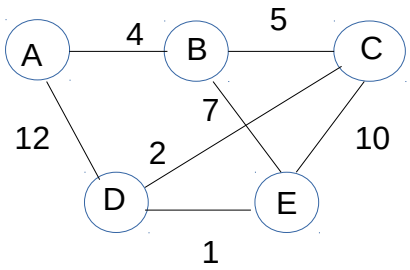
Q5. Alcuni ricercatori vogliono vedere se una variante del protocollo Go Back N in cui il receiver manda solamente negative ack (NACK) se il segmento ricevuto non è corretto oppure non è quello atteso, e non manda niente se il segmento ricevuto è corretto e quello atteso, mentre il sender, alla ricezione del NACK, reinvia il segmento più vecchio in finestra, ed interpreta lo scadere del timeout come avvenuta ricezione corretta da parte del receiver del segmento più vecchio in finestra (cioè lo scadere del timeout equivale alla ricezione di un ack relativo al segmento più vecchio in finestra), e quindi non si usano ack cumulativi. La variante suddetta è equivalente (anche se più lenta) di quella classica? Giustificare la risposta.

Seconda parte

E1 (9 punti). Il TCP di un web server è realizzato mediante due threads, TCPR (il receiver), che realizza la ricezione dei segmenti, e TCPS (il sender), che realizza l'invio dei segmenti. La sincronizzazione tra threads è realizzata con le primitive ad ambiente locale send e receive, bloccanti. Non appena TCPR riceve un segmento, e prima di elaborarlo, ne passa una copia (invocando la primitiva send) a TCPS, che lo riceve con una receive. Descrivere, mediante *pseudocodice*, il comportamento di TCPS relativamente ad una connessione già aperta. Per semplicità, **non** occorre realizzare **né** l'apertura **né** la chiusura della connessione, **né** l'interazione con il livello applicativo, **né** la modifica del valore di cwnd. Si indichi con *S.Z* il campo Z dell'header del segmento S e si utilizzino (tra le altre) le seguenti procedure e strutture dati:

- *receive(A,B)* primitiva per ricevere dal thread A il dato B;
- *send(A,B)* primitiva per inviare il dato B al thread A;
- *makesegment(S,X)* per creare un segmento S che contiene i primi X byte ricevuti dal livello applicativo e non ancora inviati;
- *insert(A,B)* per inserire (con politica FIFO) il dato B nella coda A;
- *extract(A,B)* per estrarre il primo elemento dalla coda A e metterlo in B;
- *first(A,B)* per mettere in B il primo elemento della coda A, senza estrarlo dalla coda;
- *newdata* vettore contenente i byte ricevuti dal livello applicativo e non ancora inviati;
- *sentsegm* coda (FIFO) che contiene i segmenti inviati e non ancora riscontrati (cioè in volo).

E2 (6 punti). Nella rete a lato, i nodi utilizzano il protocollo distance vector con poisoned reverse. Al tempo t, il protocollo ha raggiunto lo stato di quiescenza. Subito dopo t, i nodi D ed E realizzano che il costo del collegamento (D,E) è passato da 1 a 47. Indicare **(a)** i vettori delle distanze che D ed E inviano a C subito dopo aver realizzato che il costo del collegamento (D,E) è passato da 1 a 47 e **(b)** quali costi dovrebbero aver avuto i collegamenti (C,D) e (C,E) affinché i vettori di cui al punto (a) non contengano distanze avvelenate.



Traccia della soluzione

Q1. ms.libero.it riceve i segmenti dall'host di Tizio. Il primo segmento sarà SYNACK, e quindi: payload vuoto, flags SYN e ACK ad 1, numero sequenza Z, ack number Y, porta mittente 25, porta destinazione P. Il secondo segmento conterrà come payload *220 service ready*, nessun flag ad 1, numero sequenza Z+1, ack number Y, porta mittente 25, porta destinazione P.

Q2. In generale non è possibile, altrimenti non si risolverebbe il problema del controllo del flusso (per questo, si inviano al massimo (*rwnd-in volo*) byte di nuovi dati), **ma è possibile se** S ha ricevuto un riscontro con rwnd=0 ed ha inviato un segmento di *zero window probe*, che contiene un byte di nuovi dati. In questo caso, X=Y+1.

Q3. Il sistema DNS utilizza UDP per l'invio delle query. Quindi, la riservatezza può essere garantita solamente da IPsec nella modalità ESP (Encapsulating Security Payload).

Q4. A partire da t, A invia il RTS (tempo impiegato = 160/8 x 10^{-6}=20 microsecondi), quindi attende un timeout (che richiede due volte 20 (tempo di trasmissione di RTS) + 4 (due ritardi di propagazione) + 16 (SIFS)+ 20 (tempo di trasmissione del CTS)=2x60=120 microsecondi)), poi genera (nel caso migliore) R=0, e ripete il protocollo aspettando un DIFS = 50 microsecondi, e poi come sopra per altre 15 volte, il tempo minimo richiesto sarà di 140+15x190 microsecondi= 2990 microsecondi.

Q5. No, perché si possono non ricevere correttamente dei pacchetti. Ad esempio, se il sender ha inviato più di un pacchetto, il più vecchio nella finestra ha numero di sequenza X, ed il NACK di quel pacchetto viene perduto, allora scadrà il timer, il sender scorrerà la finestra cancellando il pacchetto più vecchio che quindi non potrà essere più inviato correttamente al receiver.

E1.

```
<apertura della connessione>;
lastack=0; newdata.first=0; newdata.last=0;  involo=0;//inizializzazione variabili
receive (TCPR, segm);
while segm.fin==0  //se non è richiesta la chiusura della connessione
   {  if segm.ack==1  //se segm contiene un ack
        {   if segm.acknum>lastack          //se contiene un ack nuovo
              {   lastack=segm.acknum;
                  dupack=0;
                  rwnd=segm.rwnd;
                  tosend=newdata.last-newdata.first;
                  if tosend>0    //se ci sono dati nuovi da trasmettere
                        {        segmsize=min((cwnd-involo),(rwnd-involo),tosend,MSS);   //dimensione del segmento da inviare
                                 segment=makesegm(segmsize);
                                 involo=involo+segmsize;
                                 send(IP, segment);
                                 incoda(sentsegm,segment);
                                 newdata.first=newdata.first-segmsize;
                         };
                   while (first(sentsegm).seqnum<segm.acknum)   //finchè ci sono segmenti riscontrati
                         {  extract(sentsegm, S);
                            involo=involo-S.segmsize;
                         }
              }
            else   {     dupack++;          //ack duplicato
                         if dupack==3    //fast retransmit
                                {   dupack=0;
                                    send(IP, first(sentsegm));
                                }
                    }
      receive(TCPR,segm);
   }
<chiusura della connessione>;
```

E2.

(b) Posto costo(C,D) =X e costo(C,E)=Y devono valere: X+9>12 (tra A e D), X+5>16 (tra B e D), X+Y>23 (tra D ed E e tra E e D). Quindi: X>11 ed X+Y >23. Ad esempio, X=12 e Y=12.

(a) vettore ricevuto da D (inf.= infinito):

A	B	D	E
Inf.	Inf.	0	Inf.

vettore ricevuto da E (inf.= infinito):

A	B	D	E
11	7	Inf.	0