RETI DI CALCOLATORI – prova scritta del 12/02/2015

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

Prima parte (10 punti)

Q1. Supponiamo che l'access point di una rete IEEE 802.11 A inizi a trasmettere al tempo t un frame F contenente un datagram di 928 bit su un collegamento con un host B e che l'ultimo bit di F arrivi a B al tempo t+90μs. Determinare – giustificando la risposta– quale è la frequenza di trasmissione se il ritardo di trasmissione è il doppio del ritardo di propagazione.

Q2. L'azienda *ACME s.p.a.* utilizza al suo interno indirizzi IPv4 privati del blocco 169.254.0.0/16. In particolare, al server Web dell'azienda www.acme.it è associato l'indirizzo 169.254.0.111, al server DNS locale all'azienda è associato l'indirizzo 169.254.0.123 e all'unico router NAT dell'azienda sono associati gli indirizzi 169.254.0.7 e 113.131.4.147. Indicare – giustificando la risposta – il contenuto del *resource record* di tipo A relativo a "www.acme.it" presente nel server DNS locale.

Q3. Sia m la dimensione in bit del campo *sequence number* utilizzato da Selective Repeat, sia S_f l'indice del segmento più vecchio spedito dal sender e per esso non ancora riscontrato e sia R_n l'indice del prossimo segmento atteso dal receiver. Indicare – giustificando la risposta – l'intervallo dei possibili valori che S_f può assumere rispetto a R_n e a m.

Q4. Consideriamo l'utilizzo del protocollo *IPSec AH* (*Authentication Header)* in modalità trasporto per il trasferimento di un segmento TCP. Indicare – giustificando la risposta – per quale motivo:

(a) viene inserito nel campo *protocollo* dell'intestazione IP il numero 51 (anziché il numero 6 riservato a TCP), e

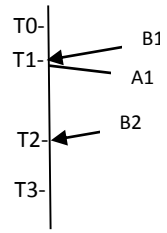
(b) non è garantita la riservatezza.

Seconda parte (20 punti)

E1 (5 punti). Descrivere <u>con uno pseudo-codice</u> il comportamento di un server proxy quando riceve una richiesta GET di tipo "if-modified-since". Assumere di avere a disposizione le operazioni:

```
connection TCPopen (IPaddress, int) //per aprire una connessione
void TCPsend (connection, data)     //per spedire dati su una connessione
data TCPreceive (connection)        //per ricevere dati su una connessione
void TCPclose (connection)          //per chiudere una connessione
int TCPbind (int)                   //per rich. assegnaz. porta su cui attendere rich. di conn.
void TCPunbind (int)                //per liberare una porta
connection TCPaccept (int)          //per attendere richieste di connessione
```

E2 (6 punti). Al tempo T0 il TCP di un host A ha già stabilito una connessione con il TCP di un altro host B, ha 2 MSS di dati in volo (spediti in due segmenti *full-sized*), 2 MSS di dati non ancora spediti, il numero di sequenza del segmento più vecchio in volo è X, la dimensione della finestra di congestione *cwnd* è 6 MSS e il valore di *ssthresh* è 8 MSS. Al tempo T1>T0 riceve da B un riscontro B1 e in conseguenza ciò invia un segmento A1 contenente 1 MSS di dati. Al tempo T2>T1 riceve da B un altro riscontro B2. Sapendo che al tempo T3>T2 *cwnd=ssthresh* e che nell'intervallo [T0,T3] non si verifica alcun altro evento oltre quelli sopra menzionati, indicare – giustificando la risposta – i possibili valori dei campi *ackNum* e *rwnd* contenuti in B1 e B2.



E3 (4 punti). Sia M un router che utilizza il protocollo distance vector con poisoned reverse e supponiamo che M abbia collegamenti diretti solo con i router G, H e L e che il costo di tali collegamenti sia 5, 3 e 1 rispettivamente. Supponiamo che al tempo T1 i contenuti delle ultime copie dei vettori delle distanze ricevuti dai suoi vicini siano quelli riportati di lato.

	D_G	D_H	D_L
G	-	1	2
H	1	-	1
L	2	1	-
F	2	3	4

(a) Indicare il contenuto della tabella delle distanze calcolata e del vettore nextHop determinati da M al tempo T1.

(b) Subito dopo T1, il router M rileva che il collegamento ML non è più disponibile. Indicare in che modo M modifica sia la tabella delle distanze calcolata sia il vettore *nextHop*.

E4 (5 punti). Consideriamo una rete locale che utilizza il protocollo Slotted Aloha e supponiamo che tre soli nodi della rete debbano trasmettere un frame e che inizino tutti e tre a trasmettere il proprio frame per la prima volta nello slot N. Indicare –giustificando la risposta– quale è la probabilità che tutti e tre riescano a trasmettere con successo il proprio frame entro e non oltre la fine dello slot N+4. Per semplicità assumiamo che tutte le collisioni vengano notificate istantaneamente.

TRACCIA DELLA SOLUZIONE

Q1. $d_{trasm} + \frac{1}{2} d_{trasm}$ = 90μs quindi d_{trasm}=60μs. Poiché la lunghezza complessiva di F è di (240+928+32) bit abbiamo che $\frac{1200\ b}{R} = \frac{60}{10^6} s$ ovvero R = 20 Mbps.	**Q3.** Poiché in SR la dimensione massima della finestra di invio è 2^{m-1} abbiamo che: - se $R_n \geq 2^{m-1}$ allora $S_f \in [R_n-2^{m-1}, R_n]$ e - se $R_n < 2^{m-1}$ allora $S_f \in [0, R_n]$ o $S_f \in [R_n+2^{m-1}, 2^m-1]$.
Q2. Affinché il server Web dell'azienda sia raggiungibile dall'esterno della rete NAT, il record contenuto nel server DNS sarà <www.acme.it, A, IN, ..., 113.131.4.147> ovvero associerà "www.acme.it" all'indirizzo pubblico del router NAT (il quale si incaricherà di "tradurre" le richieste destinate a 113.131.4.147, porta 80 a 169.254.01.11, porta 80).	**Q4.** (a) Per permettere al destinatario del datagram IP di rilevare che il payload del datagram contiene un'intestazione AH e quindi di passare tale payload a IPSec (e non a TCP). (b) Perché i byte del segmento TCP non vengono cifrati.

E1.

```
/* Assumiamo di disporre di:

getURL,getHost,getPort
//per estrarre URL,host,porta da rich. HTTP

cacheLookup        //cerca URL in cache

refreshCache, updateCash  //aggiorna cache

valid(cached,request)
//confronta data copia cache e richiesta

build304reply //per costruire risposta 304

isNotModified //per det. tipo risp. HTTP
*/
```

```
int p = TCPbind(8080);
while (true) do {
  c = TCPaccept(p);
  request = TCPreceive(c);
  URL = getURL(request);
  cached = cacheLookup(URL);
  if (cached != NULL && valid(cached,request))
    reply=build304reply(cached);
  else {
    s = TCPopen(getHost(request),getPort(request));
    TCPsend(s,request);
    reply = TCPreceive(s); TCPclose(s);
    if isNotModified(reply) refreshCash(URL);
    else updateCash(reply);
  }
  TCPsend(reply); TCPclose(c);
}
```

E2. Osserviamo che in TO il TCP si trova nello stato di slow start dato che cwnd < ssthresh. Analizziamo i vari casi possibili.

(1) Se B1 è un riscontro non duplicato allora, subito dopo avere ricevuto B1, TCP rimane in slow start e cwnd=7MSS. Affinchè in T3 cwnd=ssthresh, anche B2 deve essere un riscontro non duplicato; in tal caso infatti, subito dopo avere ricevuto B2, cwnd=8MSS.

- Se B1.ackNum=X+1MSS allora, dopo avere ricevuto B1, min(cwnd,B1.rwnd)=2MSS dato che viene spedito 1 solo MSS di nuovi dati, quindi B1.rwnd=2MSS.
  - Se B2.ackNum=X+2MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)≤1MSS dato che non vengono spediti nuovi dati, quindi B2.rwnd≤1MSS.
  - Analogamente, se B2.ackNum=X+3MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=0 MSS e B2.rwnd=0MSS.
- Se B1.ackNum=X+2MSS allora, dopo avere ricevuto B1, min(cwnd,B1.rwnd)=1MSS dato che viene spedito 1 solo MSS di nuovi dati, quindi B1.rwnd=1MSS. Poiché B2.ackNum=X+3MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=0 MSS dato che non vengono spediti nuovi dati, quindi B2.rwnd=0MSS.

(2) Se B1 è un riscontro duplicato ricevuto per la terza volta allora, subito dopo avere ricevuto B1, TCP passa in fast recovery, ssthresh=3MSS e cwnd=6MSS. Affinchè in T3 cwnd=ssthresh, B2 deve essere un riscontro non duplicato; in tal caso infatti, subito dopo avere ricevuto B2, cwnd=3MSS.

Osserviamo che B1.ackNum=X e il valore di B1.rwnd è ininfluente (dato che scatta il meccanismo di fast retransmit).

- Se B2.ackNum=X+1MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=1MSS dato che non vengono spediti nuovi dati, quindi B2.rwnd=1MSS.

Analogamente, se B2.ackNum=X+2MSS allora, dopo avere ricevuto B2, min(cwnd,B2.rwnd)=0MSS e B2.rwnd=0MSS.

E3.

(a)	D_M	$nextHop_G$
G	3	L
H	2	L
L	1	L
F	5	L

(b)	D_M	$nextHop_G$
G	4	H
H	3	H
L	4	H
F	6	H

E4. Analizziamo i casi possibili. (a) Se dopo la prima collisione nello slot N i tre nodi scelgono tutti di ritentare subito la trasmissione, essi collideranno di nuovo nello slot N+1 e dovranno allora scegliere di attendere 0, 1 e 2 slot rispettivamente. Ciò avverrà con probabilità $\left(\frac{1}{2^3} \times \frac{6}{4^3}\right)$. (b) Se dopo la prima collisione due nodi scelgono di ritentare subito la trasmissione mentre il terzo sceglie di attendere uno slot, i primi collideranno di nuovo nello slot N+1 e dovranno allora scegliere di attendere 1 e 2 slot rispettivamente. Ciò avverrà con probabilità $\left(\frac{3}{2^3} \times \frac{2}{4^2}\right)$. (c) Se dopo la prima collisione due nodi scelgono di attendere uno slot mentre il terzo ritenta subito la trasmissione, i primi collideranno di nuovo nello slot N+2 e dovranno allora scegliere di attendere 0 e 1 slot rispettivamente. Ciò avverrà con probabilità $\left(\frac{3}{2^3} \times \frac{2}{4^2}\right)$. (d) Osserviamo infine che se dopo la prima collisione tutti e tre i nodi scegliessero di attendere uno slot essi collidererebbero di nuovo nello slot N+2 e non potrebbero quindi riuscire tutti e tre a trasmettere con successo entro la fine dello slot N+4. - La probabilità che tutti e tre i nodi riescano a trasmettere con successo entro la fine dello slot N+4 è quindi $\left(\frac{1}{2^3} \times \frac{6}{4^3}\right) + 2 \times \left(\frac{3}{2^3} \times \frac{2}{4^2}\right) = \frac{27}{256}$.