

## **Homework assignment**

### **Ipv4 e algoritmi di routing**

**Q1.** 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.

**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 169.254.0.111, porta 80).**

**Q2.** Indicare – giustificando la risposta – quali dei seguenti indirizzi IP appartengono alla rete 100.8.0.0/14: (a) 100.10.7.0 (b) 100.11.7.0 (c) 100.12.7.0 (d) 100.13.7.0 (e) 100.255.7.0

**Un indirizzo IP appartiene alla rete 100.8.0.0/14 (01100100.00001000.00000000.00000000) se i suoi 14 bit più significativi (cioè 01100100.000010) coincidono con il prefisso della rete. Quindi solo (a) e (b) appartengono alla rete 100.8.0.0/14, essendo: (a) 01100100 . 00001010 . 00000111 . 00000000 (b) 01100100 . 00001011 . 00000111 . 00000000 (c) 01100100 . 00001100 . 00000111 . 00000000 (d) 01100100 . 00001101 . 00000111 . 00000000 (e) 01100100 . 11111111 . 00000111 . 00000000**

**Q3.** Il sistema autonomo di una grande azienda è logicamente diviso in un sottosistema che usa indirizzi IP pubblici nel blocco 121.212.64.0/18 e uno che usa indirizzi IP privati nel blocco 192.168.0.0/16. Il sistema autonomo è dotato di 3 router NAT ciascuno dei quali ha tre indirizzi privati e tre indirizzi pubblici. Indicare – giustificando la risposta – se una GET inviata dall'host di indirizzo 121.212.85.93 al web server dello stesso sistema autonomo e di indirizzo 192.168.15.87 deve passare necessariamente per un router NAT di quel sistema autonomo o meno.

**Si, deve necessariamente passare per un router NAT del sistema autonomo perchè, anche se l'host appartiene allo stesso sistema autonomo, non fa parte del sottosistema con indirizzi privati (a meno che non abbia anche un indirizzo privato), e quindi è esterno all'area privata di cui fa parte il server web.**

**Q4.** Supponiamo che un ISP disponga del blocco di indirizzi IP 222.222.222.128/25. (a) Indicare se gli indirizzi 222.222.222.127 e 222.222.222.129 appartengono o meno a tale blocco. (b) Indicare – giustificando la risposta – in che modo l'ISP può ripartire l'insieme di indirizzi IP di cui dispone in 4 sotto blocchi: uno di 64 indirizzi, uno di 32 indirizzi e due di 16 indirizzi

**(a) E' facile osservare che 222.222.222.127 (11011110.11011110.11011110.01111111) non appartiene al blocco 222.222.222.128/25 (11011110.11011110.11011110.10000000/25=11011110.11011110.11011110.1) mentre 222.222.222.129 (11011110.11011110.11011110.10000001) vi appartiene. (b) L'ISP utilizzerà i bit più significativi del blocco per identificare i sottoblocchi, ad esempio 222.222.222.128/26 per 64 indirizzi, (prefisso di sottorete: 11011110.11011110.11011110.10) con 222.222.222.192/27 per 32 indirizzi (prefisso di sottorete: 11011110.11011110.11011110.110) e 222.222.222.224/28 (prefisso di**

**sottorete: 11011110.11011110.11011110.1110) e 222.222.222.240/28 (prefisso di sottorete: 11011110.11011110.11011110.1111) per 16 indirizzi.**

**Q5.** Si consideri il *blocco* di indirizzi 129.66.35.48/X. Quanto può valere X? Giustificare la risposta.

**Siccome il primo indirizzo del blocco si ottiene ponendo a 0 gli ultimi 32-X bit dell'indirizzo (e l'ultimo tutti a 1), e siccome 48 in binario è 00110000, allora X=28, come valore minimo, e X=32 come valore massimo.**

**Q6.** Un sistema autonomo AS che contiene 8000 hosts e quattro server (web, email, ftp e DNS), adotta al suo interno indirizzi privati del gruppo 192.168.0.0/16, ed usa per i suoi router NAT tutti gli indirizzi pubblici del gruppo 114.113.87.24/29. Qual'è il numero massimo di connessioni TCP uscenti da AS e relative agli host che possono essere contemporaneamente attive nell'AS? Giustificare la risposta.

**Il numero massimo di connessioni TCP relative agli host può essere limitato dagli host o dai router NAT. I numeri di porta utilizzabili da ogni entità (host o router) è al massimo 65536 (tutti i numeri di porta) - 1024 (i numeri noti), cioè 64512. Dato che gli host sono 8000 (con almeno 8000 indirizzi IP), il limite è dato dai router NAT. In particolare, ogni indirizzo IP pubblico dei router NAT deve avere assegnata una porta. Gli indirizzi pubblici sono 8, ed il numero delle porte per ciascuno di essi è 65536 (tutti i numeri di porta) - 1024 (i numeri noti), cioè 64512. Pertanto, in totale si possono avere al massimo  $64512 \times 8 = 516096$  connessioni TCP uscenti da AS e relative agli host.**

**Q7.** Un router NAT è dotato di 6 interfacce di rete a cui sono associati 4 indirizzi IP pubblici (113.205.94.0, 113.205.94.1, 113.205.94.2, 113.205.94.3) e 2 indirizzi privati (192.168.0.1 e 192.168.0.2). Indicare – giustificando la risposta – quanti server FTP possono essere simultaneamente attivi nella rete NAT.

**In generale, il numero di server FTP simultaneamente attivi nella rete NAT (cioè come server utilizzabili solamente *dentro* la rete NAT) è limitato superiormente solo dal numero di porte TCP disponibili sulle interfacce presenti in tale rete. Il numero di server FTP accessibili sulla porta 21 *dall'esterno* della rete NAT è invece al più 4, ovvero al più 1 per ogni indirizzo pubblico del router NAT.**

**Q8.** Un router IPv4 R deve inoltrare su un collegamento con MTU a 500 byte un datagram IP che ha ricevuto, la cui intestazione non contiene opzioni e che contiene un segmento TCP di 980 byte. Indicare –giustificando la risposta– i valori dei campi offset e length dei frammenti inviati da R.

**Per inoltrare il datagram di 1000 byte ricevuto (980 byte di dati e 20 byte di intestazione), R dovrà creare 3 frammenti (dovendo inserire una nuova intestazione, 2 frammenti sarebbero lunghi 1020 byte in tutto): il primo con offset=0 e length=500 (480 byte di payload+20 byte di intestazione), il secondo con offset=60 (è sempre relativo al payload) e length=500, il terzo con offset=120 e length=40.**

**Q9.** (a) Un router IPv4 R1 deve inoltrare, su un collegamento con MTU di 500 byte con un router IPv4 R2, un datagram IP che ha ricevuto, la cui intestazione non contiene opzioni e che contiene un segmento TCP di 990 byte. (a) Indicare i valori dei campi offset (“scostamento”) e length (“lunghezza totale”) dei frammenti inviati da R1. (b) R2 deve inoltrare tutti i frammenti ricevuti da R1 su un collegamento con MTU di 260 byte con un router IPv4 R3. Indicare i valori dei campi offset e length dei frammenti inviati da R2.

- a) Per inoltrare il datagram di 1010 byte ricevuto (990 byte di dati e 20 byte di intestazione), R1 invierà a R2 3 frammenti: f1: 480 0 500 payload offset length; f2: 480 60 500 payload offset length; f3: 30 120 50 payload offset length
- b) R2 frammenterà il primo e il secondo frammento ricevuto in due frammenti ciascuno, non dovrà invece frammentare il terzo: f1.1: 240 0 260 payload offset length; f1.2: 240 30 260 payload offset length; f2.1: 240 60 260 payload offset length ; f2.2: 240 90 260 payload offset length; f3: 30 120 50 payload offset length.

**Q10.** Un router IPv4 R deve inoltrare su un collegamento con MTU di 500 byte un datagram IP che ha ricevuto, la cui intestazione contiene 20 byte di opzioni e che contiene un segmento UDP di 960 byte. Indicare –giustificando la risposta– i valori dei campi offset (“scostamento”) e length (“lunghezza totale”) dei frammenti inviati da R.

**Per inoltrare il datagram di 1000 byte ricevuto (960 byte di dati e 40 byte di intestazione), R dovrà inviare 3 frammenti (le “length” devono essere calcolate in modo che gli “offset”, espressi in ottetti di byte, siano ovviamente un numero intero):**

- il primo, contenente 456 byte di dati (se fossero 460, avrei le future offset rappresentate da un numero non intero, in particolare la prossima avrebbe valore 57,5) , con offset=0 e length=496,
- il secondo, contenente 456 byte di dati, con offset=456/8=57 e length=496,
- il terzo, contenente 48 byte di dati, con offset=912/8=114 e length=88.

**Q11.** Consideriamo un sistema autonomo S i cui nodi utilizzano distance vector con poisoned reverse come protocollo di routing intra-dominio, e siano A e B due nodi in S. Indicare – giustificando la risposta – se è possibile che A invii a B un vettore contenente solo distanze con valore infinito mentre B sta anch’esso inviando ad A un vettore contenente solo distanze con valore infinito.

**Si, ciò è possibile se per esempio A e B hanno entrambi un unico altro vicino C attraverso cui instradano tutti i pacchetti. Se C diventa irraggiungibile, ovvero se i collegamenti AC e BC cadono, A e B ricalcolano i loro vettori delle distanze e ciascuno invia all’altro un vettore contenente solo distanze con valore infinito.**

**Q12.** In una rete R, i percorsi vengono calcolati con l’algoritmo Distance Vector con Poisoned Reverse. Se invece si utilizzasse l’algoritmo Distance Vector, ma *senza* Poisoned Reverse, si otterrebbero gli stessi percorsi, o si avrebbero percorsi diversi? Giustificare la risposta.

**Si, si otterrebbero gli stessi percorsi, perché la tecnica poisoned reverse serve solamente per far convergere più velocemente l’algoritmo alla soluzione finale nel caso di peggioramenti nel costo di alcuni link.**

**Q13.** Una rete R di 93 nodi utilizza l’algoritmo distance vector con *split horizon* per determinare i cammini interni ad R. Sia C un router di R che è collegato con altri 8 router della stessa rete. Qual’è il numero *minimo* di righe delle tabelle che C invia ai suoi vicini? Giustificare la risposta.

**Split horizon prevede che la tabella inviata da un router ad un suo vicino contenga solamente una riga per ogni destinazione raggiungibile passando per un router diverso da quello a cui sta inviando la tabella. Quindi, se tutte le destinazioni sono raggiungibili solamente passando per un vicino V, C invierà una tabella vuota a V.**

**Q14.** In una rete i cui router utilizzano il protocollo distance vector con poisoned reverse, il router R ha calcolato che la sua distanza minima per V1 è x, che la sua distanza minima per V2 è x+w e che la sua distanza minima per Z è x+2w, dove V1 e V2 sono gli unici router direttamente collegati

a R e dove  $x$  e  $w$  sono interi positivi. Indicare – giustificando la risposta – in che modo R aggiorna il suo vettore delle distanze se V2 gli comunica che la sua distanza minima da Z è  $w-1$ .

**Non sappiamo a chi R inoltra i datagram per Z prima della comunicazione di V2: quindi, in genere, R aggiornerà la sua distanza per Z solo se  $\min\{C(R,V1)+DV1(Z), C(R,V2)+w-1\} < x+2w$ , cioè se  $C(R,V2) < x+w+1$ .**

**Q15.** Una rete di 24 nodi utilizza l'algoritmo link state per il routing dei pacchetti. Indicare –giustificando la risposta– quale è il numero minimo di iterazioni dell'algoritmo che il nodo X della rete deve effettuare per essere certo di aver trovato i cammini minimi verso tutti i suoi 7 vicini.

**23, perchè la distanza minima da X a uno dei vicini potrebbe essere l'ultima a essere determinata dall'algoritmo.**

**Q16.** In una rete composta da nodi che utilizzano il protocollo path vector, il nodo X riceve un path vector Pathy dal suo vicino Y. Indicare –giustificando la risposta– in quali casi X aggiornerà il suo cammino per la destinazione Z a seguito della ricezione di tale vettore.

**X aggiornerà il suo cammino PathX[Z] se Pathy[Z] non contiene X e se X considera il cammino X+Pathy[Z] migliore –secondo la metrica adottata– del cammino PathX[Z] che aveva precedentemente determinato.**

**Q17.** Sia V un router che utilizza il protocollo distance vector con poisoned reverse e supponiamo che V abbia un collegamento diretto solo con i router E, F e G e che il costo di tali collegamenti sia 1, 3 e 3 rispettivamente. Supponiamo inoltre che il vettore delle distanze di V contenga  $DV(E)=1$ ,  $DV(F)=2$ ,  $DV(G)=3$  e che i vettori che ha ricevuto dai suoi vicini contengano  $DE(Z)=4$ ,  $DF(Z)=2$  e  $DG(Z)=1$ . Determinare –giustificando la risposta– quale è il valore di  $DV(Z)$  determinato da V dopo aver ricevuto un nuovo vettore da G contenente  $DG(Z)=3$ .

**V determinerà il nuovo valore di  $DV(Z)$  applicando la formula di Bellman-Ford:  $DV(Z) = \min \{C(V,E)+ DE(Z), C(V,F)+DF(Z), C(V,G)+ DG(Z)\} = \min \{1+4, 3+2, 3+3\} = 5$ .**

**Q18.** Sia R un router che utilizza il protocollo distance vector con poisoned reverse e siano V1 e V2 gli unici due router con cui R ha un collegamento diretto. Supponiamo che:

- al tempo  $t_1$  V1 determini che la sua distanza per Z è  $x$  e che invii il suo vettore delle distanze ai suoi vicini;
- al tempo  $t_2 > t_1$  V2 determini che la sua distanza per Z è  $2x+y$  e che invii anch'esso il suo vettore delle distanze ai suoi vicini;
- al tempo  $t_3 > t_2$  il vettore delle distanze determinato da R (dopo avere ricevuto gli advertisement spediti da V1 e V2 al tempo  $t_1$  e  $t_2$  rispettivamente) contenga  $DR(V1)=y$ ,  $DR(V2)=x$  e  $DR(Z)=x+y$ ;
- al tempo  $t_4 > t_3$  R rilevi che il collegamento R-V1 è caduto.

Indicare –giustificando la risposta– quale è la nuova distanza per Z che R determina non appena rileva che R-V1 è caduto.

**Vi sono due casi possibili: (a) se V2 aveva “avvelenato” la sua distanza per Z nell'advertisement inviato a R, allora R considererà Z non più raggiungibile; (b) altrimenti la distanza per Z determinata da R sarà  $DR(Z)=C(R,V2)+2x+y$ , dove  $C(R,V2)$  indica il costo del collegamento R-V2.**

**Q19.** Sia R un router che utilizza il protocollo distance vector con poisoned reverse. Indicare – giustificando la risposta – se è possibile o meno che R, pur essendo in grado di instradare pacchetti per una destinazione D, comunichi a tutti i suoi vicini di non essere in grado di raggiungere D.

**Si, se R ha un unico vicino V (diverso da D) allora R, anche se in grado di instradare pacchetti per la destinazione D, “avvelenerà” la sua distanza per D nel vettore che invia a V (dato che R inoltrerà a V i pacchetti destinati a D).**

**E1.** Descrivere con uno pseudocodice il comportamento dell'algoritmo Link State per inviare, ricevere e aggiornare le informazioni contenute nel suo Link-State Database. Utilizzare i comandi:  
*receive()* che restituisce un pacchetto LS e l'interfaccia su cui è stato ricevuto  
*send(i,p)* che invia sull'interfaccia i un pacchetto LS p  
*LSDBlookup(p)* che restituisce la copia di p già presente nell'LSDB (e restituisce NULL se il router non ha ricevuto p precedentemente)  
*LSDBupdate(p)* che aggiorna l'LSDB con p  
 e spiegare il significato di eventuali altri comandi utilizzati.

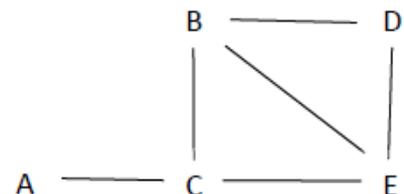
Assumiamo che le interfacce del nodo siano rappresentate dagli interi  $[0, N-1]$  e indichiamo con *welcome* il messaggio di "benvenuto" inviato dal nodo a tutti i suoi vicini e con *p.seqNum* il numero di sequenza del pacchetto p.

```

for (j=0; j<N; j++) { send(j, welcome); }
while true { <p,i>=receive();
  x=LSDBlookup(p);
  if ((x==null) || (x.seqnum<p.seqnum)) {
    LSDBupdate(p);
    for (j=0; j<N; j++)
      { if (j!=i) send(j,p); } }
}
  
```

**E2.** Consideriamo la rete a lato, i cui nodi utilizzano il protocollo distance vector con poisoned reverse. Indicare –giustificando la risposta– quali sono i costi dei collegamenti della rete se gli ultimi vettori ricevuti da B prima che la rete raggiungesse lo stato di quiescenza sono

	C	D	E
A	6	11	$\infty$
C	-	5	$\infty$
D	$\infty$	-	2
E	$\infty$	2	-



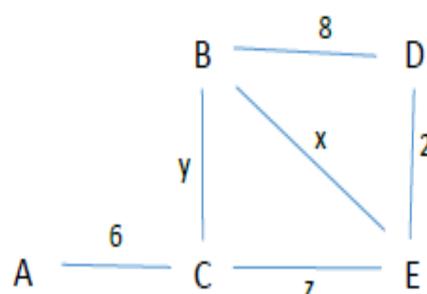
e il costo  $C(B,D)$  del collegamento BD è 8.

E2. Osserviamo che:

- (a)  $D_C(A)=6 \rightarrow C(A,C)=6$  (banalmente)
- (b)  $D_D(E)=2 \rightarrow C(D,E)=2$  (dato che  $C(D,B)>2$ )
- (c)  $D_E(C)=\infty \rightarrow C(C,E) > C(B,E)+C(B,C)$
- (d)  $D_D(C)=5 \wedge (b) \wedge (c) \rightarrow C(B,E)+C(B,C)=3$

e che gli altri valori contenuti nei vettori ricevuti da B non aggiungono altri vincoli.

I costi dei collegamenti sono quindi:



con  $x+y=3 < z$ . Per esempio  $x=1, y=2, z=4$ .

**E3.** Consideriamo una rete che contiene due server DHCP. Supponiamo che un host che desideri richiedere l'assegnazione di un indirizzo IP esegua il protocollo DHCP ma debba attendere di ricevere un'offerta da entrambi i server per poi scegliere l'indirizzo con il lease time più alto. Se l'host non riceve un'offerta da entrambi i server entro un tempo TO ripete il protocollo dall'inizio. Descrivere utilizzando un automa a stati finiti il comportamento del demone di livello application di tale protocollo in esecuzione negli host, utilizzando:

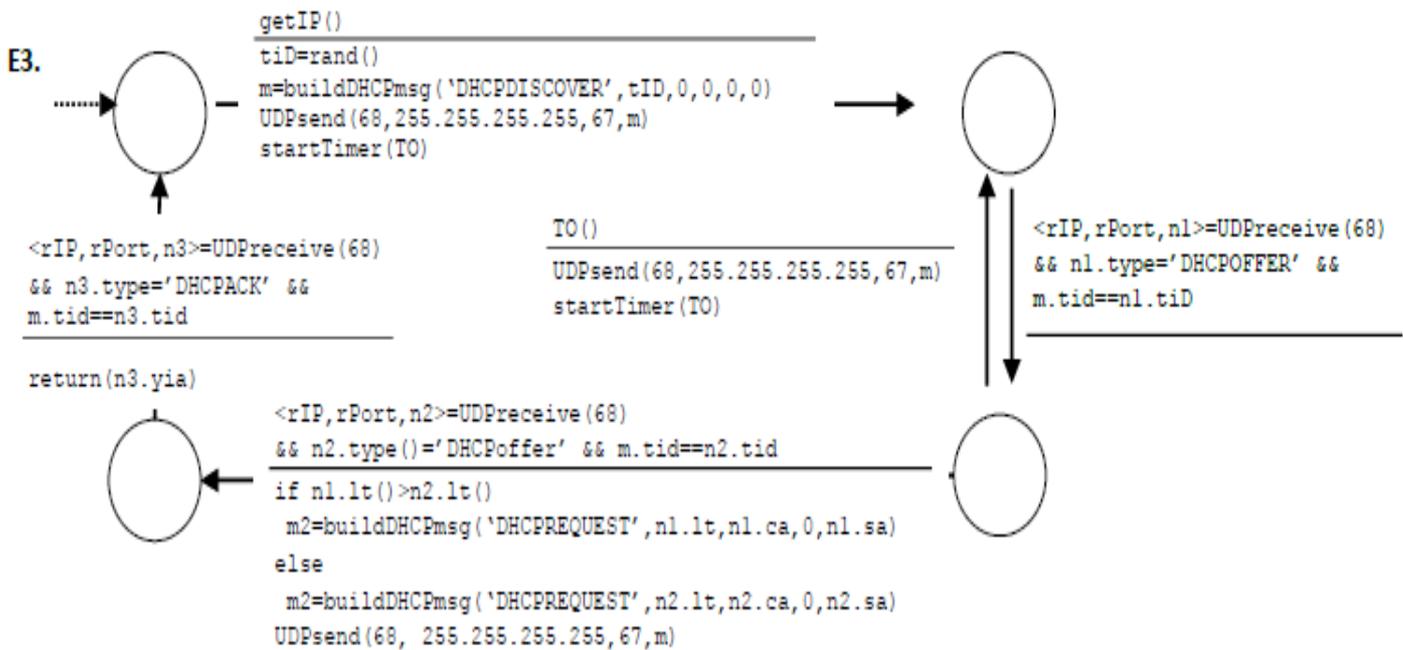
-l'evento *getIP()* per la richiesta al demone,

-le operazioni *UDPsend(localPort, remoteIP, remotePort, message)*

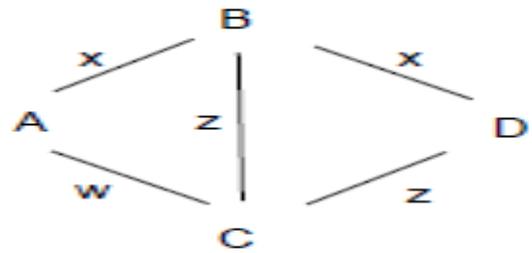
*<remoteIP, remotePort, message> UDPreceive(localPort)*

per l'invio e la ricezione di messaggi DHCP,

-l'operazione *buildDHCPmsg(type,tid,lt,ca,yia,sa)* per costruire un messaggio DHCP, e i metodi *m.type, m.tid, m.lt, m.ca, m.yia, m.sa* per accedere ai campi corrispondenti di un messaggio DHCP m.



**E4.** Consideriamo la rete a lato in cui i nodi A, B, C, D utilizzano l'algoritmo distance vector con poisoned reverse e in cui  $x, w$  e  $z$  sono interi positivi con  $x < w$ . Determinare, giustificando la risposta: (a) quali relazioni devono valere tra  $x, w$  e  $z$  affinché, quando la rete ha raggiunto lo stato di quiescenza al tempo  $t$ , il numero di distanze "avvelenate" presenti nelle ultime copie dei vettori che A ha ricevuto dai suoi vicini sia massimo, e determinare il contenuto del vettore delle distanze calcolato da A in tale caso; (b) il contenuto del vettore delle distanze calcolato da A se, subito dopo  $t$ , A rileva che il collegamento AB non è più disponibile.



**E2. (a)** I vettori ricevuti da A conterranno il minimo dei valori indicati nelle caselle della tabella sottostante, dove i valori evidenziati in grassetto causerebbero l'avvelenamento della distanza.

	$D_B$	$D_C$
B	-	$\min(x+w, z)$
C	$\min(x+w, z)$	-
D	$\min(2z, x)$	$\min(2x+w, z)$

Affinché il numero di distanze "avvelenate" presenti in tali vettori sia massimo deve quindi valere la disequazione:  $2x+w < z$ . Il contenuto del vettore calcolato da A in tale caso sarà:

	$D_A$
B	$x$
C	$w$
D	$2x$

dato che  $2x+w < z \rightarrow x < z \rightarrow x < 2z \rightarrow 2x < x+2z$ .

b) Se subito dopo che la rete ha raggiunto lo stato di quiescenza al tempo  $t$ , A rileva che il collegamento AB non è più disponibile, A ricalcolerà il seguente vettore delle distanze:

	$D_A$
B	$\infty$
C	$w$
D	$\infty$

**E5.** Un sistema autonomo utilizza il protocollo DHCP, ed ha 3 server DHCP, ognuno dei quali ha a disposizione 100 indirizzi IP (esclusivi) da assegnare agli host del sistema autonomo che ne fanno richiesta. Descrivere, con un automa a stati finiti, il comportamento del livello applicativo di uno di questi server quando esegue DHCP. Si supponga che il server utilizzi (tra le altre) le variabili:

- *myaddr* che contiene l'indirizzo IP del server stesso,
- *myLT* che contiene il valore del lease time di quel server,
- *tabaddr* tabella hash indirizzabile sia per indice, che con la chiave uguale al campo indir (vedi sotto), di 100 righe, una per ogni indirizzo che può assegnare, con i campi (tra gli altri)
  - *indir* che contiene l'indirizzo IP,
  - *status* che indica lo stato dell'indirizzo: libero, riservato, occupato,
  - *transid* che contiene il valore della transaction id relativa alla richiesta

e (tra le altre) le funzioni/procedure:

- *selectfreeaddr(tabaddr)* funzione che restituisce un indirizzo IP in *tabaddr* libero, se ci sono indirizzi liberi, e restituisce -1 altrimenti,
- *selectby id(tid)* che restituisce l'indice dell'elemento in *tabaddr* il cui campo *transid* è uguale a *tid*,
- *fhash(addr)* funzione hash per accedere alla tabella in modalità hash,
- *starttimer(ind)* che invoca il gestore del timer (**che non occorre realizzare**) affinché inserisca nella sua coda delle scadenze una nuova scadenza relativa all'indirizzo *ind*,
- *timeout(ind)* che viene invocata dal gestore del timer per comunicare al server che il lease time dell'indirizzo *ind* è scaduto.

**L'automata ha un solo stato. Le transizioni finiscono in quell'unico stato, e sono:**  
**DHCPDISCOVER(M)**

---

```

y==selectfreeaddr(tabaddr);
if y>0
  { DHCPOFFER.transid==M.transid;
    DHCPOFFER.leaseTime==myLT;
    DHCPOFFER.clientaddr==NIL;
    DHCPOFFER.youraddr==y;
    DHCPOFFER.serveraddr==myaddr;
    tabaddr(fhash(M.clientaddr)).status==riservato;
    tabaddr(fhash(M.clientaddr)).transid==M.transid;
      UDPsend(255.255.255.255, sourceport=67, destport=68, DHCPOFFER);
  }

```

**DHCPREQUEST(M)**

---

```

if M.serveraddr==myaddr
  { DHCPACK.transid==M.transid;
    DHCPACK.leaseTime==myLT;
    DHCPACK.clientaddr==NIL;
    DHCPACK.youraddr==M.clientaddr;
    DHCPACK.serveraddr==myaddr;
    UDPsend(255.255.255.255, sourceport=67, destport=68, DHCPACK);
    starttimer(M.clientaddr);
    tabaddr(fhash(M.clientaddr)).status==occupato;
  }
else { j==selectbyid(M.transid);
      tabaddr(j).status==libero;}

```

**timeout(addr)**

---

```

tabaddr(addr).status==libero;

```

**E6.** Un sistema autonomo S è connesso con internet tramite un router NAT R. In particolare, R è dotato di quattro interfacce di rete, I1, I2, I3 ed I4, le prime tre delle quali sono collegate rispettivamente con i tre sottosistemi del sistema autonomo, denominati SAS1, SAS2, e SAS3 ai quali sono assegnati (rispettivamente) i blocchi di indirizzi privati 192.168.0.0/18, 192.168.64.0/18 e 192.168.128.0/18. La quarta interfaccia di rete, I4, che ha l'indirizzo pubblico 113.114.97.0, è collegata con internet. R svolge funzioni di instradamento sia tra i sottosistemi autonomi (è l'unico router che collega tali sottosistemi) che tra il sistema autonomo ed internet. Descrivere, mediante pseudocodice, il funzionamento di R quando deve inoltrare datagram, sia provenienti dal sistema autonomo che da internet. Per brevità, si considerino **solamente i datagram che provengono dalla interfaccia I1**. Inoltre, si hanno a disposizione le seguenti procedure:

*receive(interf,dg)* per ricevere il datagramma dg dalla interfaccia interf;

*send(interf,dg)* per inviare il datagramma dg sulla interfaccia interf;

*inblocco(a,b,c)* che restituisce true se l'indirizzo a è nel blocco di indirizzo iniziale b e finale c, e false altrimenti;

*private\_address\_error* per segnalare un errore sull'indirizzo privato;

*nattablook(ind,port)* che restituisce (a,b), dove a è un booleano che vale true se è stata trovata una riga con la coppia (ind,port), false altrimenti; se a=true, allora b è l'indice della tabella che contiene quell'elemento, altrimenti il valore non è significativo.

*addentrynattable* per aggiungere una riga alla tabella nat: restituisce l'indice della riga aggiunta;

*freeport* che restituisce una porta di R attualmente inutilizzata.

Per semplicità, si supponga che i numeri di porta siano replicati nel campo options del datagramma: *dg.options.sourceport* per la porta sorgente, e *dg.options.destport* per la porta destinazione.

**receive(I1,dg);**

**if inblocco(dg.inddest,192.168.64.0,192.168.127.255)**

**{send(I2,dg)}**

**else if inblocco(dg.inddest,192.168.128.0,192.168.191.255)**

**{send(I3,dg)}**

**else if inblocco(dg.inddest,192.168.192.0,192.168.255.255)**

**{private\_address\_error}**

**else { (exist,i)=nattablelook(dg.indsource,dg.options.sourceport);**

**if ! exist**

**{ i=addentrynattable;**

**nattable[i].privateaddr=dg.indsource;**

**nattable[i].privateport=dg.options.sourceport;**

**nattable[i].newport=freeport;**

**nattable[i].externaladdr=dg.inddest;**

**nattable[i].externalport=dg.options.destport;**

**}**

**dg.indsource=113.114.97.0;**

**dg.options.sourceport=nattable[i].newport;**

**send(I4,dg);**

**}**