

## RETI DI CALCOLATORI – Seconda verifica in itinere, a.a. 2009/2010<sup>1</sup>

*La prova di verifica è strutturata in due parti. Per ottenere una valutazione sufficiente dell'intera prova è necessario ottenere una valutazione sufficiente della prima parte.*

### Prima parte (10 punti)

**Q1.** Indicare quali parametri di ingresso e quali parametri di uscita deve avere l'operazione *receive* offerta da IP.

**Q2.** Consideriamo un router che connette tre sottoreti. Supponiamo che tutte le interfacce presenti nelle tre sottoreti abbiano il prefisso 223.1.17.0/25 e che una debba servire fino a 60 interfacce, mentre le altre due debbano servire fino a 30 interfacce ciascuna. Scrivere tre indirizzi di rete che soddisfino tali vincoli.

**Q3.** Consideriamo una rete stub S dual-homed connessa a due reti A e B di provider e supponiamo che un gateway di S riceva un advertisement BGP da un gateway di A con cui viene annunciata una rotta per un prefisso p. Descrivere il comportamento del gateway di S quando riceve tale advertisement.

**Q4.** Consideriamo una LAN Ethernet in cui si verifica una collisione tra un nodo N1 che cerca di spedire un frame F1 per la prima volta e un nodo N2 che cerca di spedire un frame F2 per la seconda volta. Indicare – giustificando la risposta – con quale probabilità N1 e N2 collideranno di nuovo al successivo tentativo.

### Seconda parte

**E1 (8 punti).** Consideriamo un'estensione del protocollo DV che eviti il problema del "conteggio all'infinito" prevedendo che gli advertisement contengano, oltre alla distanza per ogni destinazione, anche il relativo cammino. Descrivere con pseudo-codice il funzionamento di tale estensione, indicando con  $D_x(y)$  e con  $R_x(y)$  rispettivamente la distanza e il cammino da x a y.

**E2 (4 punti).** Supponiamo che un host A sotto NAT invii una richiesta HTTP a un server S. Indicare il contenuto dei campi (dei diversi preamboli) che contengono informazioni relative all'indirizzamento: (a) nel frame ricevuto dal router di A, (b) nel frame inviato dal router di A e (c) nel frame ricevuto da S.

**E3 (8 punti).** Descrivere con un automa a stati finiti il modo in cui può essere rigenerato il token in un protocollo token-passing. Nella descrizione assumere che il nodo abbia già rilevato la necessità di rigenerare il token. Commentare opportunamente le azioni e gli eventi utilizzati.

---

<sup>1</sup> Come preannunciato, questa seconda prova di verifica non contiene quesiti né esercizi sulla sicurezza nelle reti e sulle applicazioni P2P in considerazione del fatto che tali argomenti sono stati introdotti nelle ultime due lezioni del corso.

## TRACCIA DELLA SOLUZIONE

Q1. L'operazione receive offerta da IP deve prendere in ingresso il numero del protocollo che invoca l'operazione e restituire sia i dati contenuti nel datagram ricevuto che l'indirizzo IP del mittente di tale datagram.

Q2. 223.1.17.0/26, 223.1.17.64/27, 223.1.17.96/27.

Q3. Il gateway G di S che riceve da A l'advertisement applicherà le proprie politiche di importazione per accettare o scartare la rotta per p. Nel primo caso aggiornerà la propria tabella di inoltra e inoltrerà l'informazione sulla raggiungibilità di p agli altri router di S con cui ha stabilito una sessione BGP interna. Dato che S è una rete stub, né G né eventuali altri gateway di S annunceranno a (gateway di) B di poter raggiungere p.

Q4. N1 attenderà un tempo  $K1 \cdot 512\text{bit}$  prima di tornare ad ascoltare il canale, con K1 valore intero scelto in modo casuale nell'intervallo  $[0, 2^1 - 1]$ , mentre N2 attenderà un tempo  $K2 \cdot 512\text{bit}$  con K2 valore casuale nell'intervallo  $[0, 2^2 - 1]$ . La probabilità che N1 e N2 collidano di nuovo al successivo tentativo è quindi  $\frac{1}{4}$  (ovvero la probabilità che entrambi scelgano 0 oppure 1).

E1. /\*Descriviamo lo pseudo-codice del protocollo eseguito da un generico router S. Indichiamo con N l'insieme dei nodi della rete e con  $C(x,y)$  il costo del collegamento tra x e y (con  $C(x,y)=\infty$  se non esiste un collegamento tra x e y).\*/

```
//inizializzazione
forEach y in N-{s}:
    Dx(y)=C(x,y);
    if Dx(y)≠∞ then Rx(y)={s,y} else Rx(y)={}
Ds(s)=0; Rs(s)={s};
forEach v vicino di s:
    invia a v Ds e Rs;
    forEach y in N-{s}: Dv(y)=∞;

//ciclo
loop
    on event "cambia costo collegamento con un vicino" o
        "ricevi adv. da vicino" do
        forEach y in N-{s}:
            sia w tale che: C(s,w)+Dw(y) = minv { C(s,v)+Dv(y) | s≠Rv(y) };
            Ds(y)= C(s,w)+Dw(y);
            Rs(y)= Rw(y)∪{s};
        if Ds è cambiato then forEach v vicino di s: invia a v Ds e Rs;
    forever
```

E2. Sia R il router a cui A è collegato e siano:

- $I_A$ = interfaccia di A
- $I_{R1}$  = interfaccia di R appartenente alla rete di A
- $I_{R2}$  = interfaccia di R appartenente alla rete esterna
- $I_T$ = interf. del router collegato a R sulla rete esterna
- $I_W$ = interfaccia del router collegato a S
- $I_S$ =interfaccia di S
- P=numero di porta utilizzata dal cliente HTTP in A
- NewP=numero di porta associata a (P,A) nella tabella NAT di R

	(a)	(b)	(c)
nel preambolo del frame	ind.MAC di $I_A$ ind.MAC di $I_{R1}$	ind.MAC di $I_{R2}$ ind.MAC di $I_T$	ind.MAC di $I_W$ ind.MAC di $I_S$
nel preambolo del datagram IP	ind.IP di $I_A$ ind.IP di $I_S$	ind.IP di $I_{R2}$ ind.IP di $I_S$	ind.IP di $I_{R2}$ ind.IP di $I_S$
nel preambolo del segmento TCP	P 80	newP 80	newP 80

E3. La rigenerazione del token può essere ottenuta semplicemente prevedendo che ogni nodo (una volta rilevata la necessità di rigenerare il token) cerchi di generare il token inviando un messaggio di broadcast e risolvendo eventuali possibili collisioni ad esempio utilizzando l'algoritmo di regressione esponenziale di CSMA/CD. Supponiamo di avere a disposizione le operazioni:

```
int bcast(data) //spedisce un frame in broadcast, restituisce 0 se si è generata collisione, 1 altrimenti
void startTimer(int) //avvia il timer
void stopTimer() //arresta il timer
int myRand(int) //myRand(x) restituisce un intero casuale nell'intervallo  $[0, 2^x - 1]$ 
data receiveFrame() //riceve (i dati contenuti in) un frame
```

Sia infine T la codifica del token e timeout() l'evento che segnali lo scadere del timer.

