

RETI DI CALCOLATORI – prova di verifica intermedia del 21/12/2016

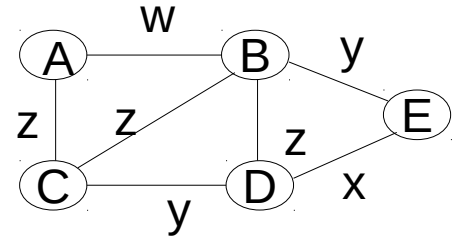
Per essere ammessi alla prova orale è necessario ottenere una valutazione sufficiente della prima parte.

Prima parte (21 punti)

Q1. 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.

Q2. 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.

Q3. Si consideri l’area a lato, in cui si utilizza il protocollo OSPF per il routing intra-sistema autonomo, e dove E è l’unico router di confine di area. Determinare-giustificando la risposta- un insieme (minimo) di vincoli che porti tutti i nodi a far transitare sul link B-E tutti i datagram con destinazioni esterne all’area, ed indicare dei possibili valori (interi non negativi e *tutti diversi tra loro*) per w, x, y, e z che rispettino tali vincoli.



Q4. In una rete Ethernet con topologia a bus, e con velocità di trasmissione di 8 Mbps, il jamming signal è lungo 48 bit, T_{fr} è uguale al tempo necessario per trasmettere 512 bit, e il tempo di sensing è di 56 μ sec. Al tempo t, il nodo A rileva la terza collisione subita dal frame che sta trasmettendo. Indicare-giustificando la risposta- in quale istante inizierà *mediamente* a trasmettere per la quarta volta quel frame, supponendo che la rete utilizzi il metodo 1-persistente, che tale trasmissione abbia successo e che il canale sia libero per tutto il tempo di sensing.

Q5. Una rete Ethernet con topologia a bus utilizza il metodo 1-persistente. I suoi nodi interrompono l’esecuzione del protocollo e dichiarano fallimento dopo 6 trasmissioni consecutive dello stesso frame senza successo. Al tempo t, quattro nodi della rete iniziano simultaneamente a trasmettere un frame per la prima volta. Qual’è la probabilità che tutti questi quattro nodi dichiarino fallimento, nell’ipotesi che nessun altro nodo della rete trasmetta a partire da t? Giustificare la risposta.

Q6. Indicare-giustificando la risposta- se è possibile o meno che in una rete IEEE 802.11, il CTS inviato dal nodo S in risposta ad un RTS inviato dal nodo Q, possa subire collisioni.

Q7. In una piconet Bluetooth, la velocità di trasferimento dati effettiva è di 0,8 Mbps. Un nodo secondario S di tale piconet deve trasmettere un frame F di 500 byte. Supponendo che la trasmissione di F inizi al tempo t, che S sia l’unico secondario attivo di quella piconet, e che il tempo di salto sia 0, indicare-giustificando la risposta- quando terminerà la trasmissione di F.

Seconda parte (9 punti)

E1. Un sistema autonomo S consiste di n nodi, e suddivide tutti gli altri sistemi autonomi di internet in tre categorie: i nemici (memorizzati in una *black list*), gli incerti (memorizzati in una *grey list*), e gli amici (che non sono memorizzati in alcuna lista perché si possono dedurre per esclusione dalle due liste precedenti). Tali liste sono note a tutti i nodi di S. La politica di preferenza locale usata da eBGP prevede di scartare comunque tutti gli aggiornamenti che contengono sistemi autonomi della black list. Un messaggio di aggiornamento X ha preferenza locale superiore a quella di un messaggio di aggiornamento Y se entrambi non contengono nodi nemici, e X contiene un minor numero di sistemi autonomi che sono nella grey list rispetto a Y. Descrivere con uno pseudocodice il comportamento di un router di confine R per decidere se aggiornare o meno la sua tabella di inoltra (eventualmente propagando nel sistema autonomo tale aggiornamento) quando riceve un messaggio di aggiornamento A che pubblicizza la raggiungibilità di un sistema autonomo remoto T. Si supponga per semplicità che i vicini di R siano rappresentati dai primi m numeri interi, e si supponga di avere a disposizione *receive(A)* per ricevere l’aggiornamento A, *send(A,i)* per inviare A al vicino i, la funzione *isin(L,AS)* che restituisce true se e solo se AS è nella lista L. Inoltre, si supponga che la tabella di inoltra di R sia memorizzata in *tabinoltra*, il cui generico elemento h-esimo, associato al percorso per raggiungere il sistema autonomo Y, ha (tra gli altri) i campi *grey(h)*, intero che contiene il numero di sistemi autonomi dell’AS path appartenenti alla grey list nel percorso attualmente utilizzato per raggiungere Y, *pathlength(h)* che contiene la lunghezza di tale AS path, e che sia inizializzato a NULL quando il sistema viene attivato. Inoltre, si supponga che l’AS path contenuto in A sia nel vettore *A.path(.)*, la cui lunghezza è in *A.pathlength*.

Q1. Sì, deve necessariamente passare per un router NAT del sistema autonomo perché, 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.

Q2. Sì, 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.

Q3. I nodi B e D sono gli unici vicini di E, e quindi tutti i datagram verso E (e quindi verso l'esterno) devono passare per uno dei due. Se $y+z < x$, allora tutti i datagram passeranno per il link B-E. Un insieme di valori che rispetta questo vincolo è ad esempio: $w=1, x=6, y=2, z=3$. In modo più esaustivo e più generale, (non minimo), per ogni nodo (tranne E) è sufficiente dare un insieme di vincoli per cui **un** cammino da quel nodo ad E che include il link B-E sia migliore di **ogni** cammino da quel nodo ad E che non include tale link. Per il nodo A abbiamo (ad esempio) $w+x+z > w+y$ e $x+y+z > w+y$ e $3z+x > w+y$; per il nodo B: $x+z > y$; per il nodo C: $x+y > z+y$ e $x+2z > z+y$; per il nodo D: $x > z+y$. Semplificando: $y < x+z$; $w < x+z$; $w+y < x+3z$; $z < x$; $y < x-z$. Quindi, $y < x-z$ e $z < x$ e $w+y < x+3z$ e $w < x+z$. Un insieme di valori che rispetta tutti questi vincoli è (ad esempio): $w=2; y=1; z=3$, e quindi $x > 4$, ad esempio $x=5$.

Q4. Dal tempo t , A trasmetterà il jamming signal (per un tempo di $6 \mu\text{sec}$), quindi aspetterà un tempo casuale che sarà X volte T_{fr} , dove $T_{fr}=64\mu\text{sec}$, quindi un tempo di sensing (uguale a $56\mu\text{sec}$) per un totale di $(6+64X+56)\mu\text{sec}$, dove X è il valore medio di $[0,7]$ e cioè $X=3.5$. Quindi, il tempo di attesa sarà di $6+224+56=286\mu\text{sec}$.

Q5. La probabilità che i quattro nodi collidano per la i -esima volta, cioè che tutti e quattro scelgano per la i -esima volta di aspettare lo stesso tempo, è $2^i/2^{4i} = 1/2^{3i}$. Quindi, la probabilità richiesta è $1 \times 1/2^3 \times 1/2^6 \times 1/2^9 \times 1/2^{12} \times 1/2^{15} = 1/2^{45}$. Il caso più generale in cui si considerino anche le collisioni tra coppie di nodi genera moltissimi sottocasi, troppi per il tempo a disposizione. Nel testo si intendeva solamente il caso indicato all'inizio.

Q6. Sì, è possibile se, ad esempio, un nodo Z per cui Q è nascosto, e che quindi non riceve il RTS, inizia a trasmettere dopo la trasmissione di Q ma prima di ricevere quella di S, ad esempio inizia proprio mentre S invia il suo CTS. In questo caso, però, il CTS arriverà a Q intatto. Affinchè il CTS arrivi a Q colliso bisogna considerare uno scenario con 4 nodi, diciamo Q, S, Y e Z che trasmettono interferendo reciprocamente.

Q7. F è lungo $500 \times 8 = 4000$ bit. In uno slot della piconet (senza salto), vengono trasmessi $625 \times 10^{-6} \times 0,8 \times 10^6 = 500$ bit. Quindi, sono necessari 8 slot dispari in cui S trasmette. Siccome in Bluetooth questi sono intervallati da 7 slot pari in cui trasmette il nodo primario, in totale saranno necessari 15 slot, la cui durata è $15 \times 625 = 9375 \mu\text{sec}$, e quindi la trasmissione terminerà al tempo $t + 9375 \mu\text{sec}$. Se invece consideriamo il salto, allora in ogni slot vengono trasmessi $366 \times 10^{-6} \times 0,8 \times 10^6 = 292$ bit (sarebbe 292,8 ma le frazioni di bit non si possono trasmettere!). In questo caso ci vorrebbero $4000/292 = 13,69863$, cioè 14 slot dispari, intervallati da 13 slot pari, quindi 27 in totale, e pertanto la trasmissione finirebbe al tempo $t + 27 \times 625 = t + 16875 \mu\text{sec}$.

E1.

La soluzione seguente (sufficiente per la votazione massima) prevede che path vector sia già stato utilizzato e quindi non si debba controllare se il sistema autonomo S sia presente o meno nell'AS path. Se si vuol controllare anche questo, basta (ad esempio) modificare la condizione del primo if in questo modo:

```
if (isin(blacklist,A.path(i)) || (A.path(i)==S))
```

```
*****
```

```
receive(A);
```

```
black=false;
```

```
grey=0;
```

```
iter=A.pathlength;
```

```
i=1;
```

```
while ((i<iter)&&(!black))
```

```
  { if isin(blacklist,A.path(i))
```

```
    {black=true}
```

```
    else if isin(greylist,A.path(i))
```

```
      {grey++};
```

```
    i++;}
```

```
if ((!black) && ((tabinoltro(h)!=NULL) || (grey<tabinoltro(h).grey) || ( grey==tabinoltro(h).grey) && (A.pathlength<tabinoltro(h).pathlength)))
```

```
{
```

```
  tabinoltro(h)=A;
```

```
  tabinoltro(h).grey=grey;
```

```
  for i=1 to m
```

```
    {send(A,i)};
```

```
}
```