

RETI DI CALCOLATORI – prova scritta del 20/01/2016

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

Prima parte (12 punti)

Q1. Supponiamo che un server FTP in esecuzione su un host A riceva da un cliente in esecuzione su un host B un segmento S con porta sorgente *ps*, porta destinazione *pd*, numero di sequenza *y*, numero di riscontro *z* e flag ACK a 1 e come dati la stringa "RETR tesi.docx". Indicare i valori di porta sorgente, porta destinazione, numero di sequenza e numero di riscontro dei primi due segmenti che A invia a B dopo avere ricevuto S, nell'ipotesi che il server in A non invii dati in piggybacking.

Q2. Al tempo *t* i valori di *cwnd* e *ssthresh* del TCP di un host A per una connessione *c* sono tali che:

$$ssthresh < cwnd < ssthresh + 1 \text{ MSS}$$

Indicare – giustificando la risposta – quale è il valore assunto da *cwnd* se al tempo *t* viene ricevuto un riscontro.

Q3. 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.

Q4. Consideriamo un anello Chord che utilizza identificatori a 6 bit ed è formato dai nodi con identificatori 0, 6, 12, 28, 32, 41, 51. Indicare –giustificando la risposta– il cammino della richiesta per recuperare l'informazione associata alla chiave 63 assumendo che esso inizi: (a) nel nodo 12, (b) nel nodo 6, (c) nel nodo 0.

Seconda parte

E1 (6 punti). Consideriamo una variante del protocollo Go-Back-N in cui la dimensione *N* della finestra è inizialmente *maxWin* e varia dinamicamente nel modo seguente -rimanendo comunque sempre nell'intervallo $[1, \text{maxWin}]$. Se il mittente riceve un riscontro non duplicato per un segmento *S*:

- se RTT di *S* è maggiore di RTT dell'ultimo segmento precedentemente riscontrato allora *N* viene decrementata di 1,
- se RTT di *S* è minore di RTT dell'ultimo segmento precedentemente riscontrato allora *N* viene incrementata di 1.

Descrivere l'automa del mittente di tale protocollo, che allo scadere del timeout rispedisce al più *N* segmenti, supponendo di disporre della funzione di sistema `readClock()` che restituisce il valore dell'orologio di macchina.

E2 (6 punti). Consideriamo una variante del protocollo RIP che avvelena le distanze tenendo conto sia del *nextHop* che del *nextHop* del *nextHop*. Descrivere –utilizzando uno pseudo-codice– il comportamento di un router *R* che utilizza tale protocollo quando riceve un advertisement da un vicino. Per semplicità assumere che i nodi della rete siano identificati dai primi *N* naturali e che i primi *M* siano i vicini di *R*.

E3 (6 punti). Supponiamo che solo due nodi (*A* e *B*) di una rete Ethernet con topologia a bus debbano trasmettere un frame di 90 byte e che al tempo *t*, dopo aver verificato che il canale è inattivo, inizino tutti e due a trasmettere simultaneamente il proprio frame: *A* per la seconda volta e *B* per la terza volta. Supponendo inoltre che i nodi utilizzino la variante p-persistente del protocollo, con $p=0,8$, che la frequenza di trasmissione sia di 10 Mbps, che la rete sia lunga 250 metri, che la velocità di propagazione sia di 2×10^8 m/sec, che il jamming signal sia di 48 bit, che T_{fr} sia uguale al tempo necessario per trasmettere 512 bit e sia uguale alla lunghezza dello slot di persistenza, indicare – giustificando la risposta – quale è la probabilità che il nodo *A* riesca a terminare di trasmettere con successo il suo frame all'istante $t + 231,65$ microsecondi incorrendo in una sola collisione dopo *t*.

Q1. Dato che il server in A non invia dati in piggybacking, il primo segmento inviato da A sarà il riscontro TCP, conterrà porta sorgente pd , porta destinazione ps , numero di sequenza z e numero di riscontro $y+K$, dove K è il numero di byte corrispondenti alla stringa "RETR tesi.docx", e non conterrà dati. Il secondo segmento inviato da A conterrà la risposta FTP e gli stessi valori di porta sorgente, porta destinazione, numero di sequenza e numero di riscontro del precedente.

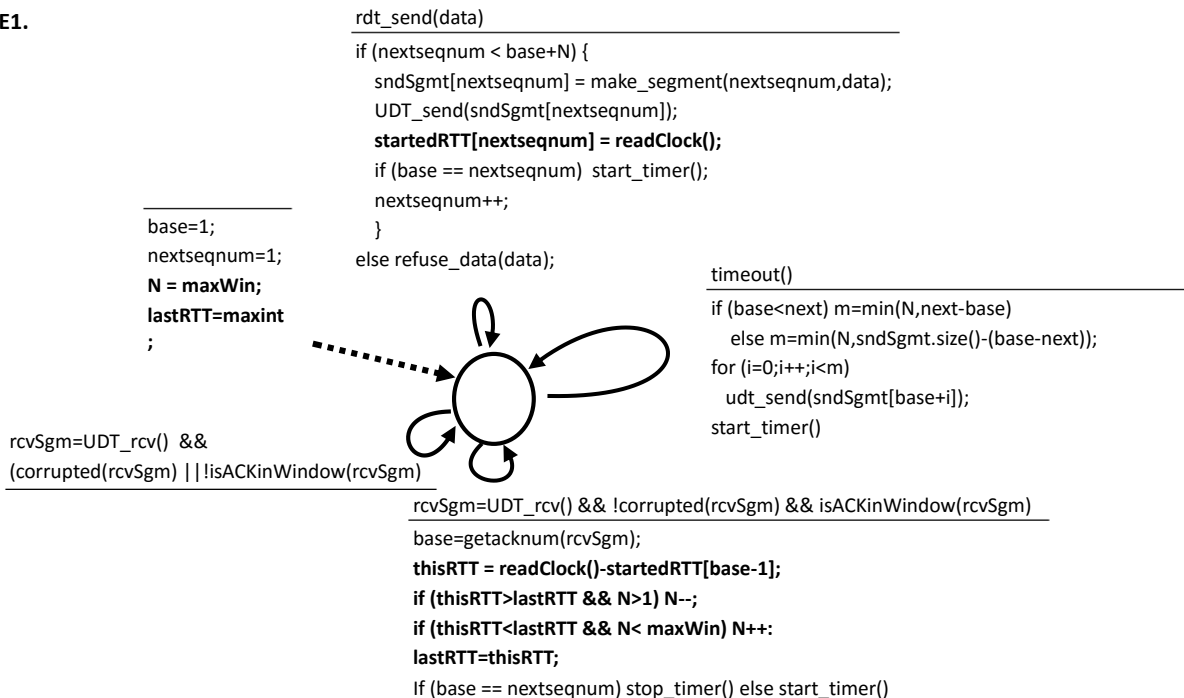
Q2. Al tempo t il TCP di A si trova nello stato *congestion avoidance* dato che non può trovarsi né nello stato *slow start* (poiché $ssthresh < cwnd$) né nello stato *fast recovery* (poiché $cwnd < ssthresh + 1 \text{ MSS}$). Quindi:

- se il riscontro ricevuto è un riscontro non duplicato allora $cwnd$ è posto a $cwnd+1/cwnd$,
- se il riscontro ricevuto è un riscontro duplicato ricevuto per la terza allora $cwnd$ è posto a $cwnd/2 + 3 \text{ MSS}$,
- se il riscontro ricevuto è un riscontro duplicato ricevuto non per la terza allora il valore $cwnd$ non viene modificato.

Q3. In generale il numero di server FTP simultaneamente attivi nella rete NAT è limitato superiormente solo dal numero di porte TCP disponibili sulle interfacce presenti in tale rete¹. 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:

Q4. (a) Il nodo 12, i cui finger sono nell'ordine 28 e 51, inoltra la richiesta al nodo 51. Il nodo 51 determina che il nodo 0 è *successor(63)* dato che $63 \in (51, 0]$. (b) Il nodo 6, i cui finger sono nell'ordine 12, 28 e 41, inoltra la richiesta al nodo 41. Il nodo 41, i cui finger sono nell'ordine 51, 0 e 12, inoltra la richiesta al nodo 51. Il nodo 51 determina che il nodo 0 è *successor(63)* dato che $63 \in (51, 0]$. (c) Il nodo 0 determina di essere esso stesso *successor(63)* dato che $63 \in (51, 0]$.

E1.



E2. // R riceve da V advertisement A e aggiorna la propria tabella

```

for (i=0;i<N;i++)
    if ( D[i].next==NULL || 1+A[i].cost<D[i].cost || V==D[i].next ) {
        D[i].cost = 1+A[i].cost;
        D[i].next = V;
        D[i].nextnext = A[i].next;
    }
// R invia la sua tabella (avvelenata) ai vicini
for (j=0;j<M;j++) {
    for (i=0;i<N;i++) {
        if ( D[i].next==j || D[i].nextnext==j )
            toSend[i].cost = 16;
        else {
            toSend[i].cost = D[i].cost;
            toSend[i].next = D[i].next;
        }
    }
    send(j,toSend);
}
    
```

¹ Ovvero, se vi sono m interfacce nella rete NAT, da $\sum_{i=1}^m p_i$ dove p_i è il numero di porte disponibili sull' i -esima interfaccia.

E3. A rileverà la collisione al tempo $t + 1,25 \mu s$ e, dopo avere inviato il segnale di jam ($4,8 \mu s$), per terminare di trasmettere con successo il suo frame all'istante $t + 231,65 \mu s$ dovrà quindi attendere un tempo X tale che $1,25 \mu s + 4,8 \mu s + X + 72 \mu s = 231,65 \mu s$ ovvero $X = 153,6 \mu s$. A dovrà quindi attendere 3 tempi di trasmissione frame/slot di persistenza. Ciò avverrà se A sceglierà:

- $K=0$ e $p > 0,8$ per tre volte e quindi $p \leq 0,8$ oppure
- $K=1$ e $p > 0,8$ per due volte e quindi $p \leq 0,8$ oppure
- $K=2$ e $p > 0,8$ una volta e quindi $p \leq 0,8$ oppure
- $K=3$ e $p \leq 0,8$

con probabilità $p = \left(\frac{1}{4} \times \frac{1}{5} \times \frac{1}{5} \times \frac{1}{5} \times \frac{4}{5}\right) + \left(\frac{1}{4} \times \frac{1}{5} \times \frac{1}{5} \times \frac{4}{5}\right) + \left(\frac{1}{4} \times \frac{1}{5} \times \frac{4}{5}\right) + \left(\frac{1}{4} \times \frac{4}{5}\right)$.

Affinché non si verifichi una seconda collisione dopo t , B non dovrà trasmettere quando A trasmette. La probabilità che A riesca a terminare di trasmettere con successo il suo frame all'istante $t + 231,65$ microsecondi incorrendo in una sola collisione dopo t è quindi:

$$p \times \left(1 - \left(\left(\frac{1}{8} \times \frac{1}{5} \times \frac{1}{5} \times \frac{1}{5} \times \frac{4}{5}\right) + \left(\frac{1}{8} \times \frac{1}{5} \times \frac{1}{5} \times \frac{4}{5}\right) + \left(\frac{1}{8} \times \frac{1}{5} \times \frac{4}{5}\right) + \left(\frac{1}{8} \times \frac{4}{5}\right)\right)\right).$$