

RETI DI CALCOLATORI - prova scritta del 09/04/2015 (appello straordinario)

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. Un host C vuole scaricare da un server Web S una pagina Web P di 32 kbyte contenente 3 oggetti: O1 di 2 kbyte, O2 di 64 kbyte e O3 di 2 Mbyte. Sapendo che C utilizza HTTP 1.1, indicare –giustificando la risposta– quale è il numero minimo di datagram IP che verranno spediti da C e da S affinché sia completato il trasferimento di P, O1, O2 e O3. Per semplicità assumiamo che il valore di MTU sia 64 kbyte su tutti i collegamenti. Assumiamo inoltre che C non abbia in cache alcuna copia di P, O1, O2 o O3.

Q2. Alberto è seduto a Times Square (New York) in un bar dotato di free wi-fi e accende il suo portatile per inviare un email al suo amico Bruno, il cui indirizzo è `bruno@libero.it`. Indicare, nell'ordine, tutti i messaggi (non solo SMTP) che il portatile di Alberto riceverà dal momento in cui Alberto lo accende fino al momento in cui l'email sarà stato inviato.

Q3. Al tempo t_0 il TCP di un host A ha già stabilito una connessione con il TCP di un altro host B, ha 3 MSS di dati in volo (spediti in tre segmenti *full-sized*), il numero di sequenza del segmento più vecchio in volo è X, e riceve un riscontro R. Subito dopo tale ricezione, il TCP si trova nello stato di fast recovery, il valore di *cwnd* è 6 MSS e quello di *ssthresh* è 2 MSS. Indicare –giustificando la risposta– quali sono i possibili valori di R.ackNum.

Q4. Consideriamo un anello Chord che utilizza identificatori a 6 bit ed è formato dai nodi con identificatori 1, 2, 15, 20, 33, 45, 49, 54. Indicare –giustificando la risposta– il cammino della richiesta che inizia nel nodo 15 per recuperare l'informazione associata: (a) alla chiave 60, e (b) alla chiave 6.

Seconda parte (18 punti)

E1 (6 punti). Una versione “prudente” del protocollo Go-Back-N prevede che il sender, dopo ogni timeout, debba attendere che la finestra di invio si svuoti completamente prima di poter inviare nuovi segmenti. Descrivere con un automa a stati finiti il comportamento di tale sender.

E2 (6 punti). Descrivere con uno pseudocodice il modo in cui un router R che utilizza distance vector con poisoned reverse invia ai suoi vicini le informazioni contenute nel suo vettore D delle distanze. Supporre per semplicità che i nodi della rete siano identificati dagli interi $[0, N-1]$, che i vicini di R siano identificati dagli interi $[0, M-1]$, che R sia identificato da M, e che $D[x]=\langle k, y \rangle$ indichi che la distanza per la destinazione x è k utilizzando come next-hop y.

E3 (6 punti). Supponiamo che solo tre nodi A, B e C di una rete locale che utilizza il protocollo slotted aloha debbano trasmettere un frame di dati. Supponiamo che nello slot N tutti e tre i nodi inizino simultaneamente a tentare di trasmettere ciascuno il proprio frame, A per la seconda volta, e B e C per la terza volta. Indicare –giustificando la risposta– quale è la probabilità che C riesca a trasmettere con successo il proprio frame per primo senza che si verifichino altre collisioni tra i tre nodi dopo lo slot N. Per semplicità assumiamo che tutte le collisioni vengano notificate istantaneamente.

TRACCIA DELLA SOLUZIONE

Q1. Assumendo che non avvengano perdite né corruzioni dei datagram, che C utilizzi una connessione TCP già aperta con S, che O1, O2 e O3 risiedano in S e che le richieste HTTP di O2 e O3 vengano inviate in piggybacking, i datagram che verranno spediti da C e S sono:

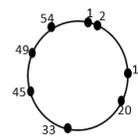
- 3 per il trasferimento di P (1 per richiesta HTTP, 1 per risposta HTTP e 1 per ACK),
- 2 per il trasferimento di O1 (1 per richiesta HTTP e 1 per risposta HTTP),
- 3 per il trasferimento di O2 (1 per richiesta HTTP e 2 per risposta HTTP) e
- 51 per il trasferimento di O3 (1 per richiesta HTTP, 33 per risposta HTTP¹ e 17 ACK).

Q2. Il portatile di Alberto riceverà almeno i seguenti messaggi (a livello application²):

- 2 messaggi DHCP (DHCP OFFER e DHCP ACK, per l'acquisizione dell'indirizzo IP)
- 1 messaggio DNS (contenente l'indirizzo IP del mailserver di Alberto)
- 7 messaggi SMTP: 2 per l'apertura della connessione SMTP ("220 service ready" e "250 OK"), 2 per la busta ("250 OK" e "250 OK"), 1 per le intestazioni ("354 start mail input"), 1 per il corpo ("250 OK") e 1 per la chiusura ("221 service closed").

Q3. Se subito dopo la ricezione di R il TCP si trova nello stato di *fast recovery* e *cwnd*=6 MSS e *ssthresh*=2 MSS, ciò implica che il TCP si trovava già in *fast recovery* in t0 (altrimenti *cwnd* dovrebbe essere *ssthresh*+3 MSS dopo la ricezione di R). Poiché dopo la ricezione di R il TCP rimane in *fast recovery*, R è necessariamente un riscontro duplicato, ovvero R.ackNum ≤ X.

Q4. (a) Il nodo 15, i cui finger sono 20, 33 e 49, inoltra la richiesta al nodo 49. Il nodo 49, i cui finger sono 54, 1 e 20, inoltra la richiesta al nodo 54. Il nodo 54, i cui finger sono 1, 15 e 33, determina che il successore della chiave 60 è il nodo 1. (b) Il nodo 15 determina di essere lui stesso il successore della chiave 6, dato che $6 \in (2,15]$.



E1.

```

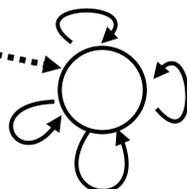
rdt_send(data) && accept==1
if (nextseqnum < base+N) {
    sndSgmt[nextseqnum] = make_segment(nextseqnum,data)
    UDT_send(sndSgmt[nextseqnum])
    if (base == nextseqnum)
        start_timer()
    nextseqnum++
}
else refuse_data(data)

```

```

rcvSgm=UDT_rcv() &&
(corrupted(rcvSgm) || !isACKinWindow(rcvSgm))

```



```

timeout()
udt_send(sndSgmt[base])
udt_send(sndSgmt[base+1])
...
udt_send(sndSgmt[nextseqnum-1])
start_timer()
accept=0

```

```

rcvSgm=UDT_rcv() && !corrupted(rcvSgm) && isACKinWindow(rcvSgm)
base=getacknum(rcvSgm)
If (base == nextseqnum)
    {stop_timer(); accept=1}
else start_timer()

```

E2. //Utilizziamo un vettore E per spedire

```

for (i=0; i<M; i++) do {
    for (j=0; j<N; j++) do
        if D[j]==<k,i> then E[j]= maxInt else E[j]= k;
    send(i,E);
}

```

E3. Analizziamo i casi possibili. Dopo la collisione avvenuta nello slot N: (a) C attende 0 slot, A attende $K_A \in [1,3]$ slot e B attende $K_B \in [1,7]$ slot, con $K_A \neq K_B$. Ciò avverrà con probabilità $\left(\frac{1}{2^3} \times \frac{3}{2^2} \times \frac{6}{2^3}\right)$; (b) C attende 1 slot, A attende $K_A \in [2,3]$ slot e B attende $K_B \in [2,7]$ slot, con $K_A \neq K_B$. Ciò avverrà con probabilità $\left(\frac{1}{2^3} \times \frac{2}{2^2} \times \frac{5}{2^3}\right)$; (c) C attende 2 slot, A attende 3 slot e B attende $K_B \in [4,7]$ slot, con $K_A \neq K_B$. Ciò avverrà con probabilità $\left(\frac{1}{2^3} \times \frac{1}{2^2} \times \frac{4}{2^3}\right)$. La probabilità che C riesca a trasmettere con successo il proprio frame per primo senza che si verifichino altre collisioni tra i tre nodi dopo lo slot N è quindi $\frac{18+10+4}{2^3 \times 2^2 \times 2^3} = \frac{2^5}{2^8} = \frac{1}{8}$.

¹ Assumendo che l'header dei messaggi HTTP response occupi 2 kbyte e che gli header TCP e IP non contengano opzioni, S dovrà inviare 33 datagram (dato che $2 \times 10^6 / 61960 = 32,28$).

² Se considerassimo anche i messaggi relativi ai protocolli dei livelli inferiori, dovremmo menzionare anche i messaggi di 802.11 (beacon, risposta di associazione, e CTS e ACK per ogni frame), di ARP (risposta con indirizzo fisico del router), e i messaggi TCP necessari per realizzare la connessione SMTP.