RETI DI CALCOLATORI – Prima verifica in itinere a.a. 2008/09

Q1 (2 punti). Consideriamo due host A e B direttamente collegati tra loro e supponiamo che A inizi a trasmettere un pacchetto di dati all'istante t=0. Se $d_{prop} = 2d_{trasm}$ dove si troverà il penultimo bit del pacchetto all'istante $t=3d_{trasm}$? Giustificare la risposta.

Q2 (2 punti). Spiegare in che modo un cliente HTTP specifica al livello sottostante quale è il destinatario di una richiesta HTTP che intende inviare.

Q3 (2 punti). Indicare quante connessioni TCP vengono stabilite tra un cliente e un server FTP per trasferire due file dal cliente al server e un file dal server al cliente. Giustificare la risposta.

Q4 (2 punti). Se un server SMTP riceve un comando "**`RCPT`**" con argomento "**`@HOST1,@HOST2:MAILBOX`**" quante connessioni TCP cercherà di apire (e con quali host) per trasmettere il messaggio?

Q5 (8 punti). Descrivere con un automa a stati finiti il comportamento di un resolver che deve determinare l'indirizzo IP del mailserver canonico di un dominio **`D`**. Per la descrizione dell'automa utilizzare gli eventi:

`mailServerFor(dominio)`	// per indicare la richiesta dell'applicazione
`<name, value, type> = UDP_rcv(porta)`	// per indicare la ricezione di una tupla

e le azioni:

`UDP_send(IPserver, porta, <QNAME,QTYPE>)`	// per indicare l'invio di una richiesta DNS con UDP
`deliver(risultato)`	// per indicare la restituzione del risultato // all'applicazione

Si assuma che il resolver risolva una sola richiesta delle applicazioni alla volta. SI assuma inoltre che ogni risposta DNS consista di una sola tupla e che le tuple di risposta di tipo "NS" contengano nel campo **`value`** direttamente l'indirizzo IP (anziché il nome) del server di competenza. Si ignori infine la necessità di gestire timer, ovvero si assuma che nessun pacchetto possa andare perduto.

Q6 (8 punti). Si assuma che il lato mittente di un'applicazione invii dati con UDP, inserendo un numero di sequenza in ogni blocco di dati (inserendo 1 per il primo blocco) ma senza attendere nessun tipo di riscontro dal suo pari. Specificare con un automa a stati finiti il comportamento del lato ricevente dell'applicazione che cerca di mostrare all'utente quanti più possibile dei dati ricevuti senza però violare l'ordine con cui sono stati spediti. Si assuma che il lato ricevente dell'applicazione possa memorizzare (ovvero "bufferizzare") al più un solo blocco di dati. Si assuma infine per semplicità che nessun segmento UDP possa essere corrotto dalla trasmissione.
Si supponga di avere a disposizione l'operazione **`UDP_rcv`** per ricevere dati, l'operazione **`seqNum`** per estrarre il numero di sequenza del blocco di dati ricevuti e l'operazione **`display`** per mostrare dati all'utente

Q7 (6 punti). Supponiamo che un processo debba inviare due pacchetti di dati utilizzando il protocollo stop&wait. Supponiamo inoltre che nessun pacchetto (né di dati né di controllo) vada perso e che soltanto il primo riscontro arrivi corrotto. Indicare quanti pacchetti in totale verranno scambiati tra il processo mittente e il processo destinatario nell'ipotesi che la lunghezza del timeout sia ½ RTT:
(a) se viene utilizzato il protocollo stop&wait standard e
(b) se viene utilizzata una variante del protocollo in cui il mittente rispedisce immediatamente un pacchetto quando riceve un riscontro corrotto.
Giustificare la risposta illustrando con una figura leggibile lo scambio di pacchetti che avverrà nei due casi.

TRACCIA DELLA SOLUZIONE

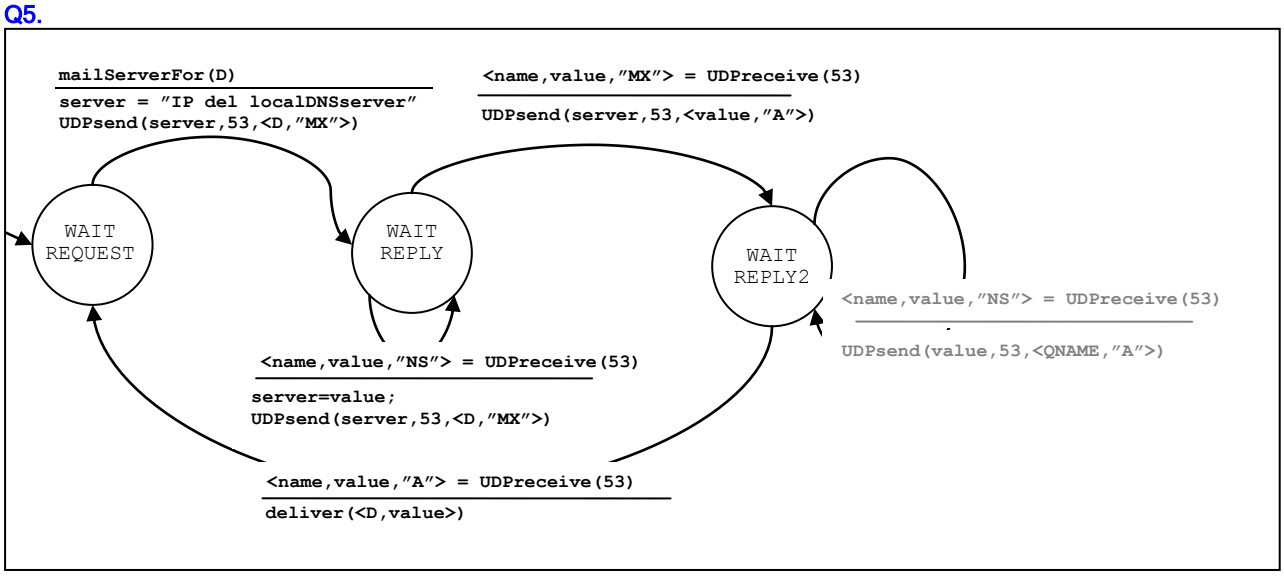
Q1. All'istante $t=3d_{trasm}$ il penultimo bit del pacchetto avrà già raggiunto l'host B. Infatti l'ultimo bit del pacchetto entrerà sul canale all'istante d_{trasm} e raggiungerà quindi l'host B all'istante $d_{trasm}+d_{prop}= 3d_{trasm}$. Quindi all'istante $3d_{trasm}$ il penultimo bit del pacchetto avrà anch'esso già raggiunto l'host B[1].

Q2. HTTP utilizza il servizio di trasporto TCP, che e' orientato alle connessioni e che identifica ogni connessione tra processi con una quadrupla contenente gli indirizzi IP e i numeri di porte dei due processi. Per inviare un messaggio, un cliente HTTP deve quindi specificare nella sua richiesta a TCP su quale connessione desidera inviare il messaggio.

Q3. Quattro connessioni: una connessione (persistente) per i dati controllo e tre connessioni (non persistenti) per i file, una per ciascuno dei file da trasfertire.

Q4. Il server cercherà di aprire una connessione TCP con l'host HOST1 cone specificato nella "source routing list" del comando RCPT.

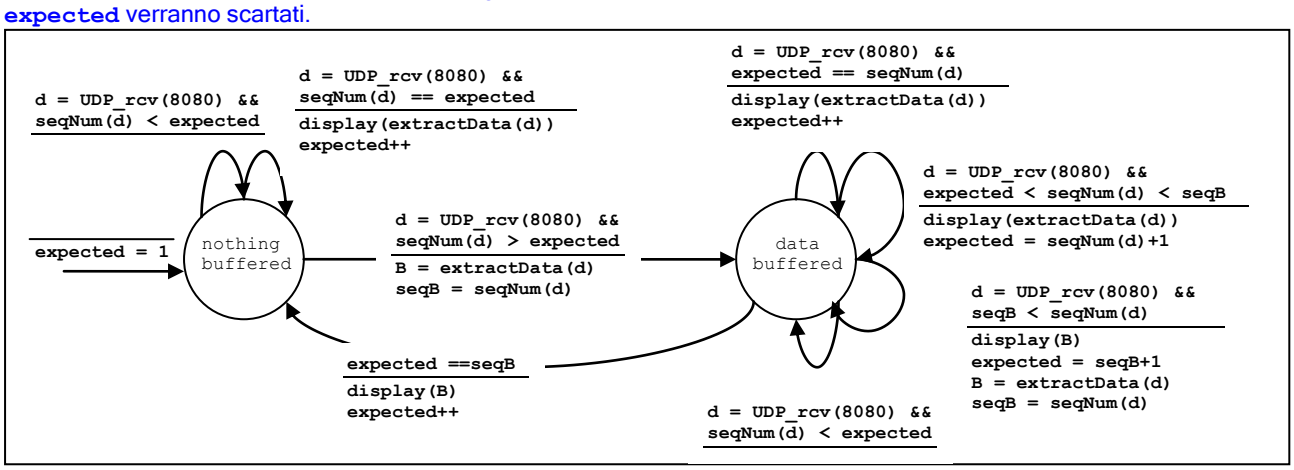
Q5.



Q6. Per cercare di mostrare all'utente quanti più possibile dei dati ricevuti, utilizziamo due variabili:

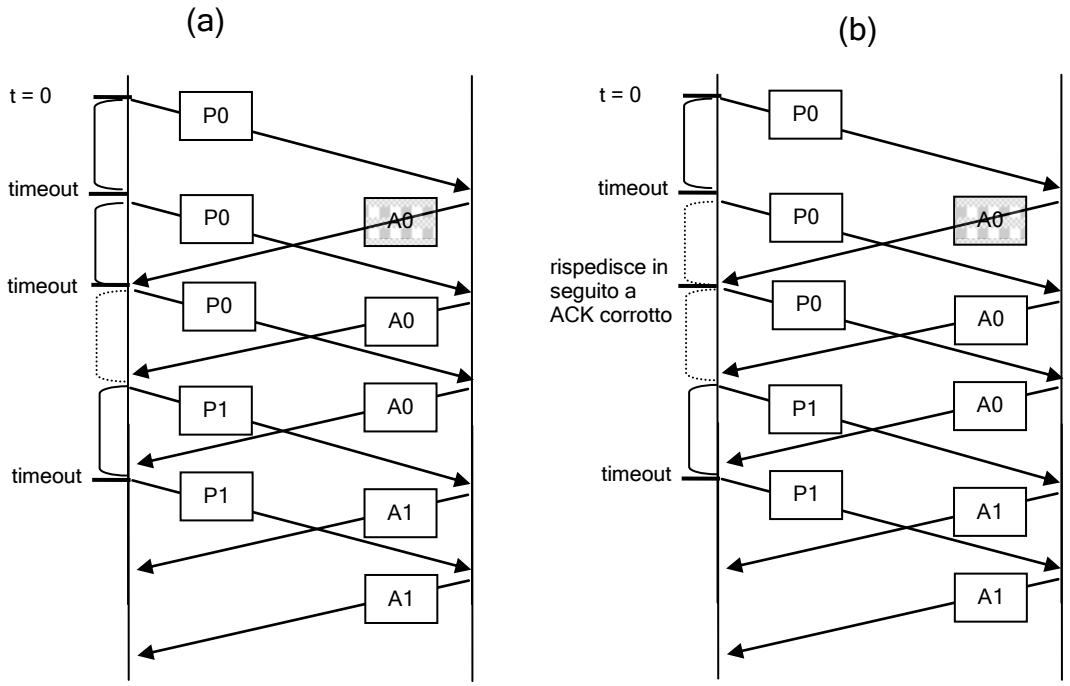
- **`expected`** - per indicare il numero di sequenza che il ricevente si aspetta di ricevere, e
- **`seqB`** - per indicare il numero di sequenza contenuto nel blocco di dati che è stato salvato nel buffer **`B`**.

Intuitivamente cerchiamo di utilizzare il buffer per ridurre il numero di dati scartati a causa di blocchi arrivati non in ordine. In generale, il ricevente si aspetterà di ricevere il blocco di dati con numero di sequenza **`expected`** ed avrà memorizzato nel buffer un blocco di dati già ricevuto ma avente numero di sequenza maggiore di **`expected`**. Se arriverà un blocco di dati con numero di sequenza **`expected`**, esso potrà essere direttamente mostrato all'utente. Se invece arriverà un blocco di dati con numero di sequenza **`n>expected`**, esso verrà mostrato all'utente nel caso in cui **`n`** sia minore del numero di sequenza del blocco di dati bufferizzato, altrimenti il blocco precedentemene bufferizzato verrà mostrato e il nuovo blocco verrà salvato nel buffer. In ogni caso, i blocchi di dati ricevuti aventi numero di sequenza inferiore a **`expected`** verranno scartati.



[1] Per la precisione il penultimo bit del pacchetto lascerà l'host A all'istante $t = (L-1)/R = d_{trasm} -1/R$ e raggiungerà quindi B all'istante $d_{trasm} -1/R + d_{prop} = 3d_{trasm} -1/R$.

Q7. Assumiamo che al tempo t=RTT (cos' come al tempo t=1,5RTT e al tempo t=2RTT) il mittente riceva il riscontro (subito) prima che scada il timeout. Osserviamo che, sotto questa ipotesi, il numero di pacchetti scambiati (5 contenenti dati e 5 contenenti riscontri) è lo stesso nei due casi. Indichiamo con P0 e A0 rispettivamente il primo segmento di dati e il relativo riscontro.



(AB)