

ESERCIZIO 3: L'apertura simultanea di una connessione TCP da parte del *client* e del *server* è una caratteristica del TCP utilizzata raramente in pratica. Il candidato:

1. proponga un cambiamento del diagramma di stato del TCP che vieti tale caratteristica.

Quando il TCP invia un segmento di controllo  $\langle SYN, SequenceNum = x \rangle$  oppure  $\langle FIN, SequenceNum = x \rangle$  l'ACK relativo trasporta come ACK number  $x + 1$ ; ossia SYNs e FINs utilizzano (consumano) una unità nello spazio dei *numeri di sequenza*. Il candidato:

2. dica, giustificando in modo appropriato, se questo è necessario e nel caso di risposta affermativa dia un esempio di ambiguità che potrebbe presentarsi se l'ACK number fosse  $x$  invece di  $x+1$ .

Supponiamo infine che il TCP operi su un link ad 1 Gigabit/s e che abbia una sola connessione aperta. Il candidato:

3. calcoli il tempo necessario ( $t_w$ ) affinché si abbia il *wraparound* (ad esempio sullo zero) sui numeri di sequenza;
4. nel caso in cui venga utilizzata l'opzione TCP di timestamp (campo di 32-bit nell'intestazione del TCP) e supponendo che il campo relativo venga incrementato 1000 volte durante il *wraparound* time calcolato precedentemente, valuti il tempo di *wraparound* per il timestamp ( $t_{wtimestamp}$ ).

#### RISOLUZIONE

1. Per evitare l'apertura simultanea basta che l'estremità della connessione TCP nello stato *SYN\_SENT* non accetti segmenti di *SYN* dall'altra estremità. In altre parole, nel diagramma delle transizioni di stato del TCP la transizione *SYN\_SENT*  $\rightarrow$  *SYN\_RCVD* deve essere rimossa. Invece la risposta ad un *SYN* nello stato *SYN\_SENT* dovrebbe essere il segmento *RST*.

2. L'incremento dell'ACK number trasportato da un ACK relativo ad un *FIN* è essenziale in quanto il mittente del *FIN*, ricevendo tale ACK è sicuro che si riferisce al *FIN* cui è associato e non ad un dato precedente. Per quanto concerne il *SYN* la situazione è diversa in quanto un qualunque segmento dati trasmesso dopo il *SYN* viene riscontrato con un ACK il cui ACK number oltre ad essere incrementato funziona come ACK implicito per il *SYN*. Di

conseguenza, in questo caso, l'incremento dell'*ACK number* è dovuto più ad un motivo di convenienza e di consistenza che di necessità di progettazione.

3. Osserviamo che 1 Gigabit/s corrisponde a 125 MByte/s. Il numero di sequenza "wrap around" su invio di  $2^{32}B = 4GB$ , cui corrispondono  $4GB/(125MBs) = 32$  secondi.

4. Se l'incrementato avviene ogni 32ms questo significa che servono  $32 \times 4 \times 10^9 ms$ , corrispondenti a circa 4 anni, prima che il campo timestamp passi per lo zero!