

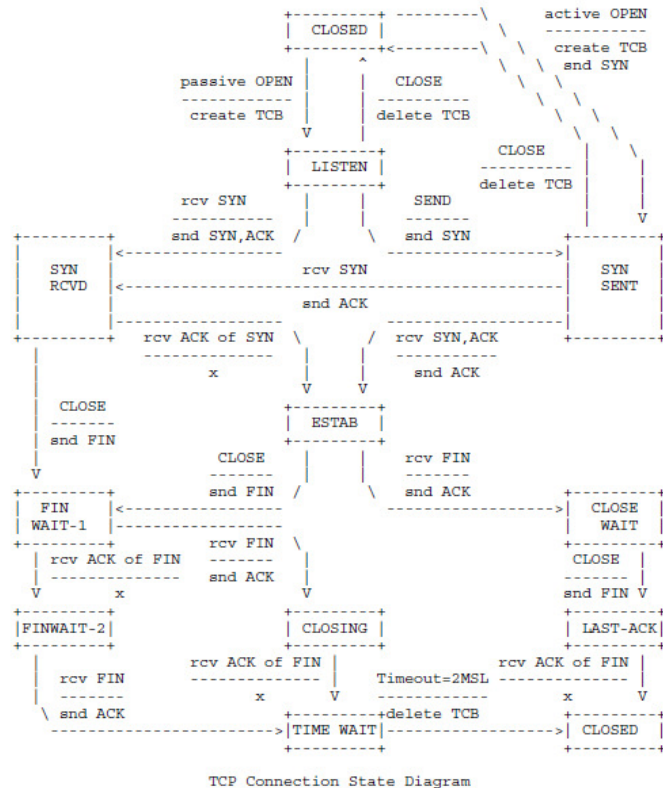
Nota su TCP*

Antonio Brogi

Dipartimento di Informatica, Università di Pisa
(Settembre 2011)

Chiusura di una connessione TCP

Come correttamente illustrato nella figura 3.40 di [KR], affinché una connessione TCP tra due applicazioni venga definitivamente chiusa, entrambe le applicazioni devono richiedere al proprio TCP di chiudere la connessione invocando l'operazione CLOSE disponibile nell'API di TCP. Purtroppo la figura 3.41 di [KR] non illustra bene tale comportamento in quanto nella transizione dallo stato "ESTABLISHED" allo stato "FIN_WAIT_1" non è evidenziato che l'evento "L'applicazione client inizia la chiusura della connessione" è una preconditione per l'invio del segmento FIN da parte di TCP. Purtroppo anche la figura 3.42 di [KR] non illustra correttamente la chiusura di una connessione in quanto nella transizione dallo stato "CLOSE_WAIT" allo stato "LAST_ACK" non è indicata nessuna preconditione. Una descrizione più completa di come TCP gestisce apertura e chiusura delle connessioni è fornita nella RFC 793 dove viene presentato il "TCP connection state diagram" (riportato a lato) che mostra come devono essere trattate richieste – anche "simultanee" – di chiusura (e di apertura) di connessioni.



Ritrasmissione rapida in TCP

Lo pseudo-codice che descrive il funzionamento della ritrasmissione rapida di TCP (pagina 230 di [KR]) non evidenzia che tale meccanismo dovrebbe essere utilizzato solo quando vengono ricevuti 4 riscontri identici *consecutivamente*, come indicato nella RFC 2581: "The fast retransmit algorithm uses the arrival of 3 duplicate ACKs (4 identical ACKs without the arrival of any other intervening packets) as an indication that a segment has been lost. After receiving 3 duplicate ACKs, TCP performs a retransmission of what appears to be the missing segment, without waiting for the retransmission timer to expire."

Controllo di flusso in TCP

L'affermazione (pagina 233 di [KR]) "l'host A si assicura che per tutta la durata della connessione sia rispettata la disuguaglianza $LastByteSent - LastByteAced \leq RcvWindow$ " non si applica alla (eventuale) fase di "zero window probing" descritta subito dopo.

Controllo di congestione in TCP

L'affermazione (pagina 250 di [KR]) "La quantità di dati non riscontrati da un mittente non può eccedere il massimo tra i valori di CongWin e RcvWindow, ossia: $LastByteSent - LastByteAced \leq \min(CongWin, RcvWindow)$ " si riferisce soltanto al vincolo che TCP deve soddisfare quando dovrebbe inviare nuovi dati, come specificato nella RFC 2581: "At any given time, a TCP MUST NOT send data with a sequence number higher than the sum of the highest acknowledged sequence number and the minimum of cwnd and rwnd."

* Lo scopo di questa breve nota è quello di cercare di chiarire solo alcune imprecisioni contenute nel testo adottato nel corso

[KR] J. Kurose & K. Ross. Reti di calcolatori e Internet (quarta edizione). Pearson-Addison Wesley

che potrebbero indurre gli studenti ad acquisire una comprensione non corretta di alcuni concetti importanti delle reti di calcolatori. Questa nota non intende tuttavia in alcun modo rappresentare un *errata corrige* completo del testo.