

Diagramma di stato di TCP

Prof. Vincenzo Auletta

auletta@dia.unisa.it

<http://www.dia.unisa.it/professori/auletta/>

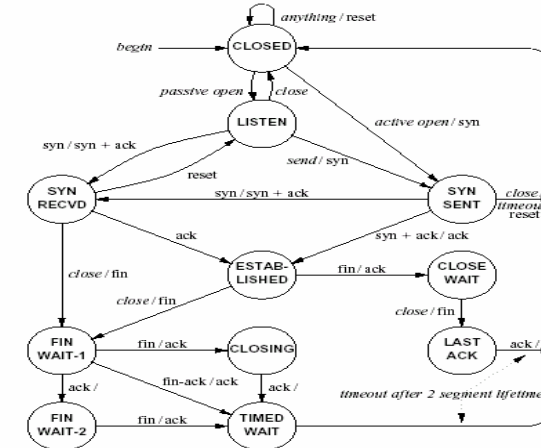


Università degli studi di Salerno
Laurea e Diploma in Informatica



Diagramma di Stato di TCP

macchina a stati finiti che descrive il funzionamento del protocollo TCP



Stati del Protocollo – 1

- ◆ Closed
 - stato iniziale (protocollo non attivo)
 - per uscire da questo stato si deve effettuare un'operazione di open (passiva o attiva)
 - ◆ l'open passiva non manda nulla e passa allo stato Listen
 - ◆ l'open attiva spedisce un messaggio SYN e passa allo stato Syn Sent
- ◆ Listen
 - in questo stato il protocollo è attivo ed è in ascolto su una porta
 - quando riceve un SYN risponde con un SYN+ACK e passa allo stato Syn Received
 - se l'applicazione chiede di inviare dati manda un SYN e passa allo stato Syn Sent



Stati del Protocollo – 2

- ◆ Syn Sent
 - stato in cui si è mandato un SYN e si attende l'ACK corrispondente
 - ◆ raggiunto da Closed con una open attiva o da Listen dopo un'operazione di send
 - attende la risposta al SYN per un certo tempo
 - ◆ se riceve un SYN con ACK passa allo stato Established e manda a sua volta un ACK
 - ◆ se riceve un SYN senza ACK (open simultanea) manda un SYN+ACK e passa allo stato Syn Received
 - ◆ se non riceve risposta effettua una close o una reset
- ◆ Syn Received
 - stato in cui si è ricevuto un SYN
 - ◆ se lo rifiuta, ritorna allo stato Listen con RST
 - ◆ se accetta, passa allo stato Established e manda l'ACK



Stati del Protocollo – 3

◆ Established

- stato in cui è stata stabilita la connessione ed è possibile iniziare il trasferimento dei dati
 - ◆ è stata completata la 3-way handshake
- se l'applicazione decide di chiudere la connessione manda un messaggio FIN e passa allo stato FIN_WAIT_1 (close attiva)
- se riceve un messaggio FIN risponde con un ACK e passa allo stato Close-Wait (close passiva)

5



Stati del Protocollo – 4

◆ Close_Wait

- stato in cui si è ricevuto un messaggio FIN e si attende che l'applicazione chiuda la connessione
- quando l'applicazione decide di chiudere la connessione manda un messaggio FIN e passa allo stato LAST_ACK

◆ Last_ACK

- stato in cui si è ricevuto il FIN dall'altro endpoint e si è risposto con un FIN
 - ◆ il protocollo attende l'ACK al suo FIN
- quando riceve l'ACK risponde con l'ultimo ACK e chiude la connessione

6



Stati del Protocollo – 5

◆ Fin_Wait_1

- stato in cui si è inviato un messaggio FIN e si attende che l'altro endpoint chiuda la connessione
- se riceve un FIN+ACK manda l'ACK e passa allo stato Time_Wait
- se riceve solo un FIN (close simultanea) manda l'ACK e passa allo stato Closing
- se riceve un ACK passa allo stato Fin_Wait_2

◆ Closing

- stato in cui si entrambi gli endpoint hanno mandato un FIN contemporaneamente
- manda l'ACK e passa allo stato Time_Wait

7



Stati del Protocollo – 6

◆ Fin_Wait_2

- stato in cui si è inviato un messaggio FIN per il quale è stato ricevuto l'ACK e si attende il FIN dell'altro endpoint (half-close)
- quando riceve un FIN manda l'ACK e passa allo stato Time_Wait

◆ Time_Wait

- attende un tempo pari a $2 * MSL$ (Maximum Segment Lifetime) prima di chiudere la connessione per attendere eventuali richieste di ritrasmissione dell'ultimo ACK
 - ◆ la durata dipende dall'implementazione
- per tutto questo intervallo di tempo la porta dell'endpoint non è riutilizzabile

8