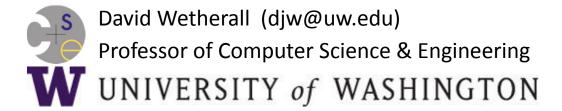
Computer Networks

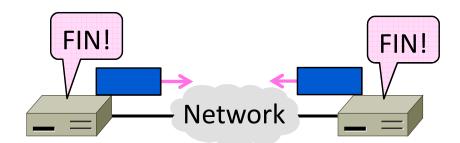
Connection Release

(§6.5.6-6.5.7, §6.2.3)



Topic

- How to release connections
 - We'll see how TCP does it



Connection Release

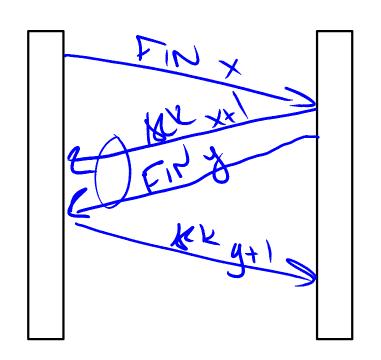
- Orderly release by both parties when done
 - Delivers all pending data and "hangs up"
 - Cleans up state in sender and receiver
- Key problem is to provide reliability while releasing
 - TCP uses a "symmetric" close in which both sides shutdown independently

TCP Connection Release

- Two steps:
 - Active sends FIN(x), passive ACKs
 - Passive sends FIN(y), active ACKs
 - FINs are retransmitted if lost
- Each FIN/ACK closes one direction of data transfer

Active party

Passive party

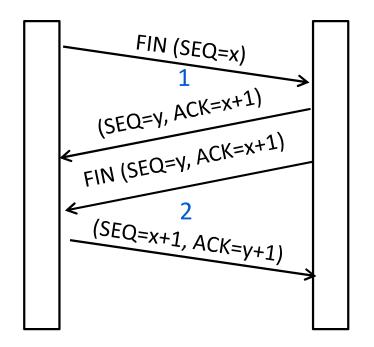


TCP Connection Release (2)

- Two steps:
 - Active sends FIN(x), passive ACKs
 - Passive sends FIN(y), active ACKs
 - FINs are retransmitted if lost
- Each FIN/ACK closes one direction of data transfer

Active party

Passive party

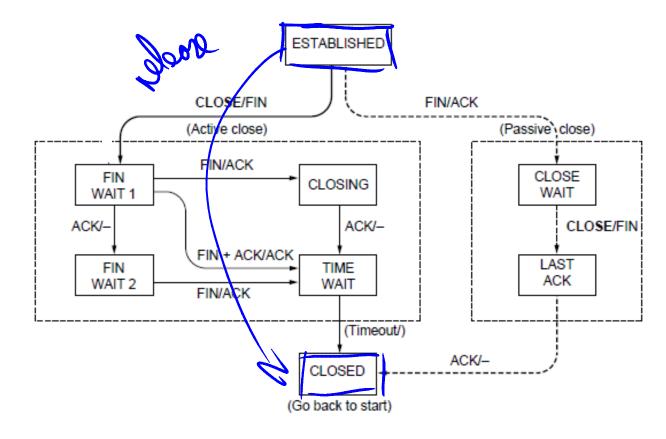


Computer Networks

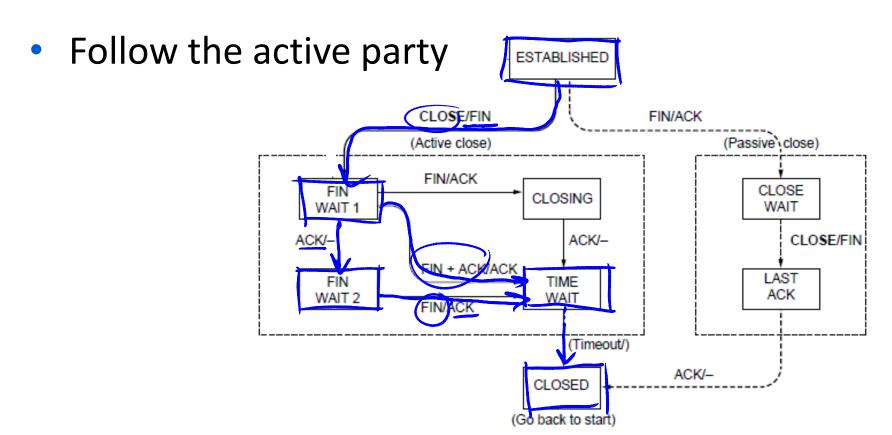
5

TCP Connection State Machine

Both parties run instances of this state machine



TCP Release



TCP Release (2)

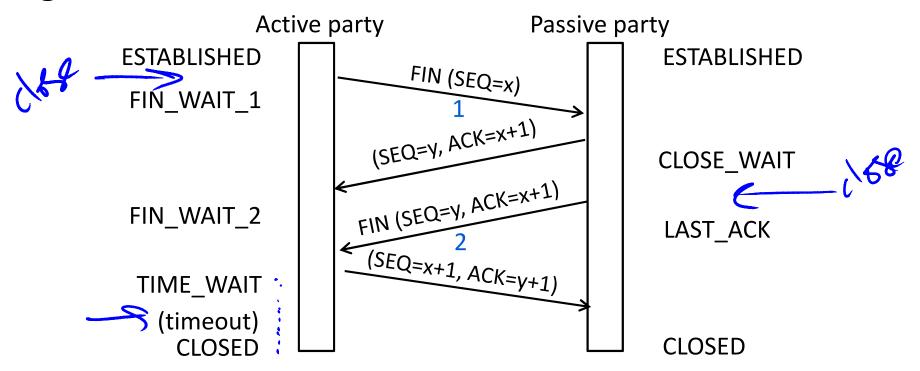
 Follow the passive party ESTABLISHED FINACK CLOSE/FIN (Passive close) (Active close) FIN/ACK FIN CLOSING WAIT 1 ACK/-CLOSE/PIN ACK/-FIN + ACK/ACK LAST FIN TIME ACK WAIT 2 WAIT FIN/ACK (Timeout/) ACIV-CLOSED

Computer Networks 8

(Go back to start)

TCP Release (3)

Again, with states ...



TIME_WAIT State

- We wait a long time (two times the maximum segment lifetime of 60 seconds) after sending all segments and before completing the close
- Why?
 - ACK might have been lost, in which case FIN will be resent for an orderly close
 - Could otherwise interfere with a subsequent connection

END

© 2013 D. Wetherall

Slide material from: TANENBAUM, ANDREW S.; WETHERALL, DAVID J., COMPUTER NETWORKS, 5th Edition, © 2011. Electronically reproduced by permission of Pearson Education, Inc., Upper Saddle River, New Jersey