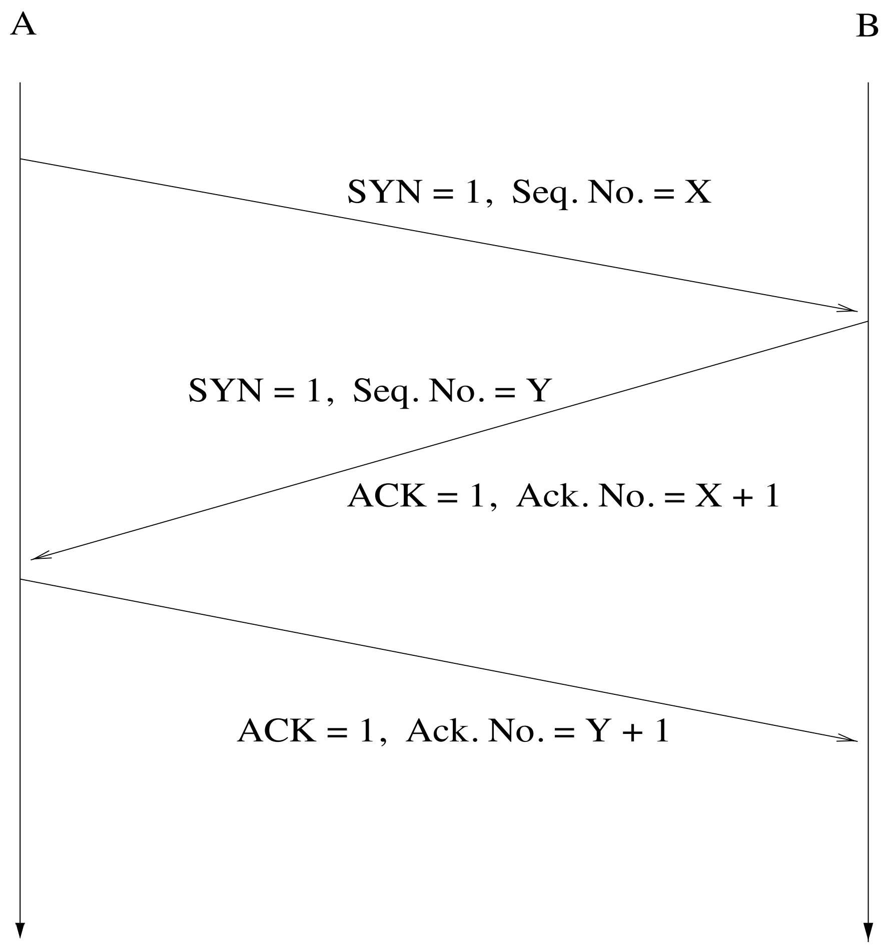
TCP connection establishment (3-way handshake):



- X, Y are chosen randomly

  $\rightarrow$ sequence number prediction

- piggybacking

2-person consensus problem: are A and B in agreement about the state of affairs after 3-way handshake?

$\longrightarrow$ in general: impossible

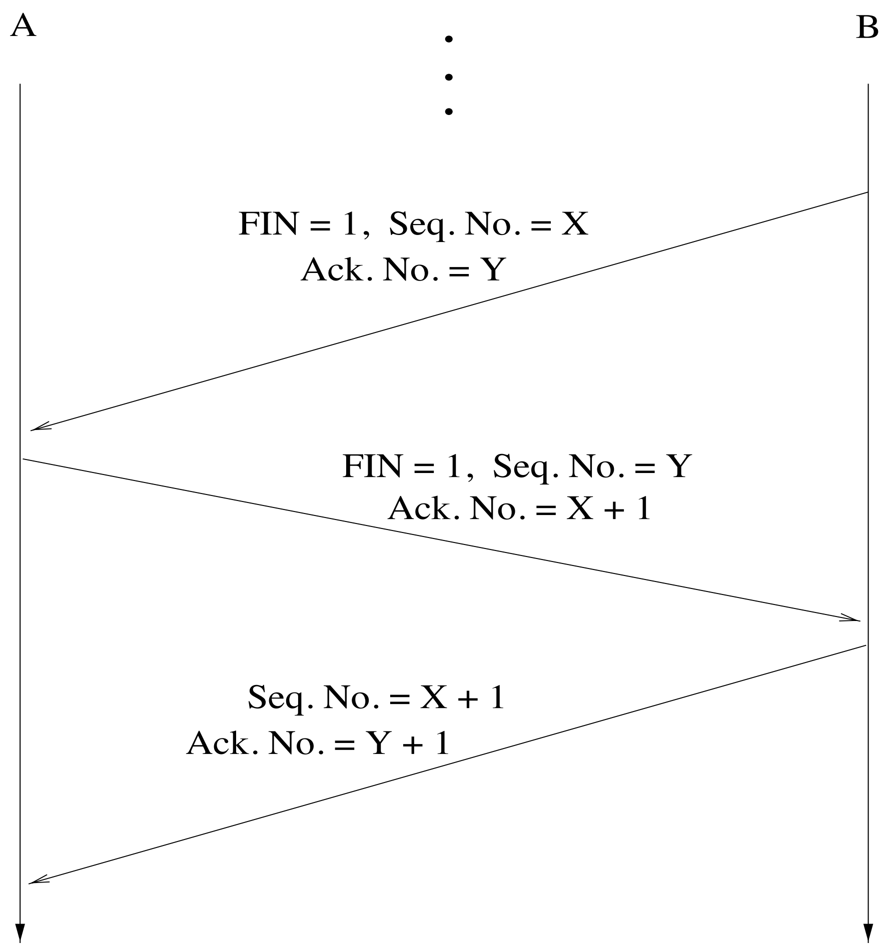
$\longrightarrow$ can be proven

$\longrightarrow$ "acknowledging the ACK problem"

$\longrightarrow$ also TCP session ending

$\longrightarrow$ lunch date problem

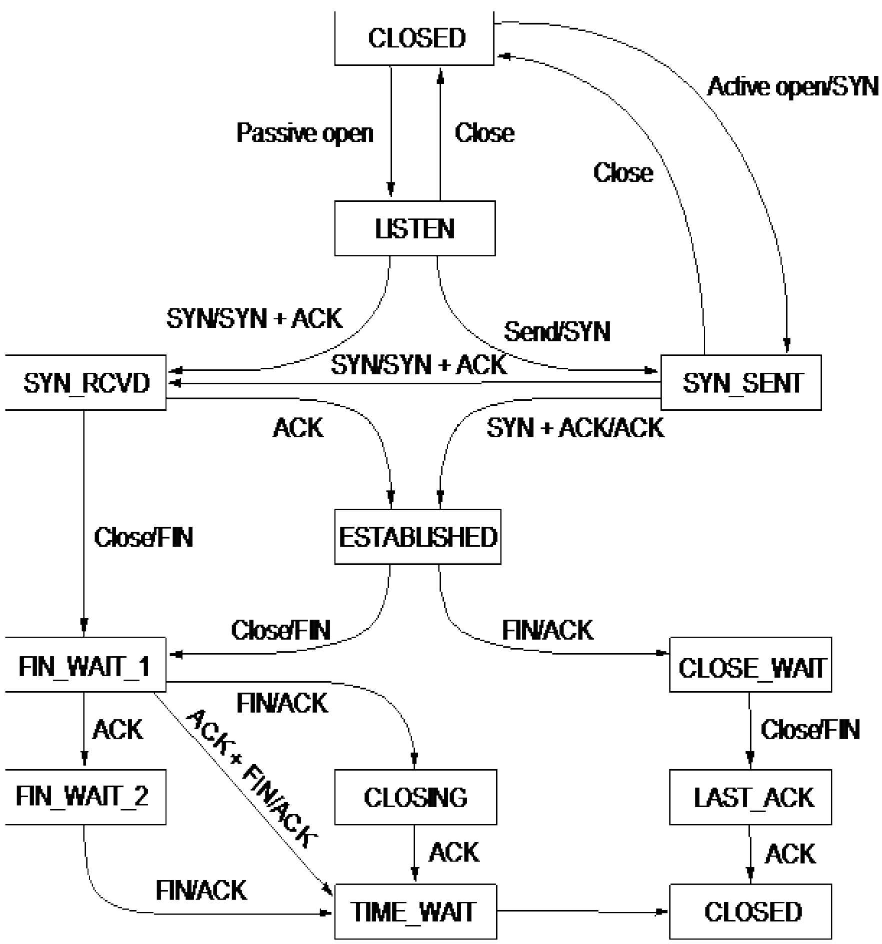
TCP connection termination:



- full duplex
- half duplex

More generally, finite state machine representation of TCP's control mechanism:

$\longrightarrow$ state transition diagram



Features to notice:

- Connection set-up:
  - client's transition to `ESTABLISHED` state without ACK
  - how is server to reach `ESTABLISHED` if client ACK is lost?
  - `ESTABLISHED` is macrostate (partial diagram)
- Connection tear-down:
  - three normal cases
  - special issue with `TIME WAIT` state
  - employs hack

Issues:

How to let sender know of change in receiver window size after `AdvertisedWindow` becomes 0?

- trigger ACK event on receiver side when `AdvertisedWindow` becomes positive
- sender periodically sends 1-byte probing packet

$\longrightarrow$ design choice: smart sender/dumb receiver

$\longrightarrow$ same situation for congestion control

Silly window syndrome: Assuming receiver buffer is full, what if application reads one byte at a time with long pauses?

- can cause excessive 1-byte traffic
- if `AdvertisedWindow` $<$ MSS then set `AdvertisedWindow` $\leftarrow 0$

Do not want to send too many 1 B payload packets.

Nagle's method:

- rule: connection can have only one such unacknowledged packet outstanding
- while waiting for ACK, incoming bytes are accumulated (i.e., buffered)

... compromise between real-time constraints and efficiency.

$\rightarrow$ useful for **`telnet/ssh`**-type interactive applications

Sequence number wrap-around problem: recall sufficient condition

$$\texttt{SenderWindowSize} < (\texttt{MaxSeqNum} + 1)/2$$

$\longrightarrow$ 32-bit sequence space/16-bit window space

However, more importantly, time until wrap-around important due to possibility of roaming packets.

bandwidth	time until wrap-around †
T1 (1.5 Mbps)	6.4 hrs
Ethernet (10 Mbps)	57 min
T3 (45 Mbps)	13 min
F/E (100 Mbps)	6 min
OC-3 (155 Mbps)	4 min
OC-12 (622 Mbps)	55 sec
OC-24 (1.2 Gbps)	28 sec