

Data exchange

Data termination

Half close: When one party is finished, the other party has not finished.

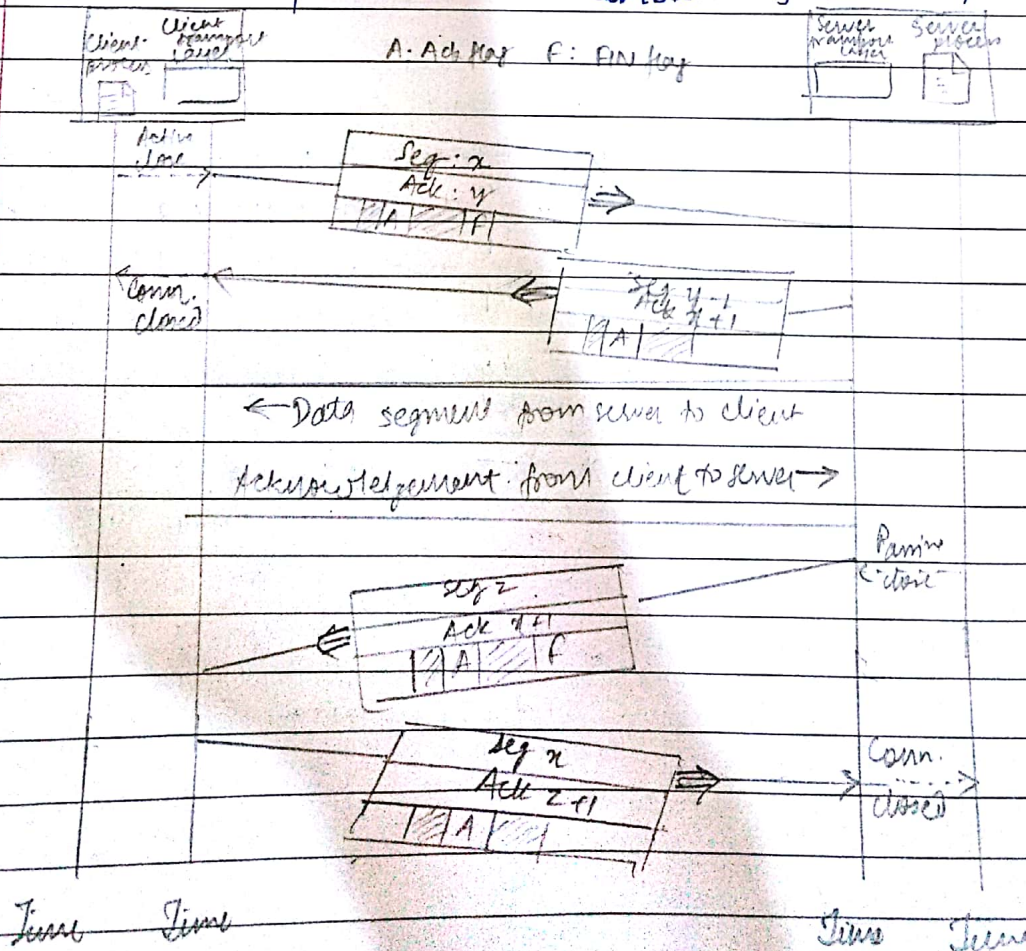
TCP data transfer

- Pushing data - receiving end has to process the data as soon as they get it (not storing)
- Urgent data (an urgent bit tells about the urgent segment of the data)

Half-close

- One end can stop sending data while still receiving data
- This is half close
- ...

Although the client has received sequence number $y-1$ and is expecting y , the server sequence number is still set to $y-1$. When the connect finally closes the sequence no of the last ACK segment is still x because the sequence no. are connected during data transfer in that direction.



Connections Reset

- Dumping a connection
 - TCP on one side has requested a connection to a non-existent host.
- Aborting a connection
 - TCP may want to abort an existing connection due to an abnormal situation.
- Terminating an idle connection
 - TCP on one side may discover that the TCP on the other side has been idle for a long time.

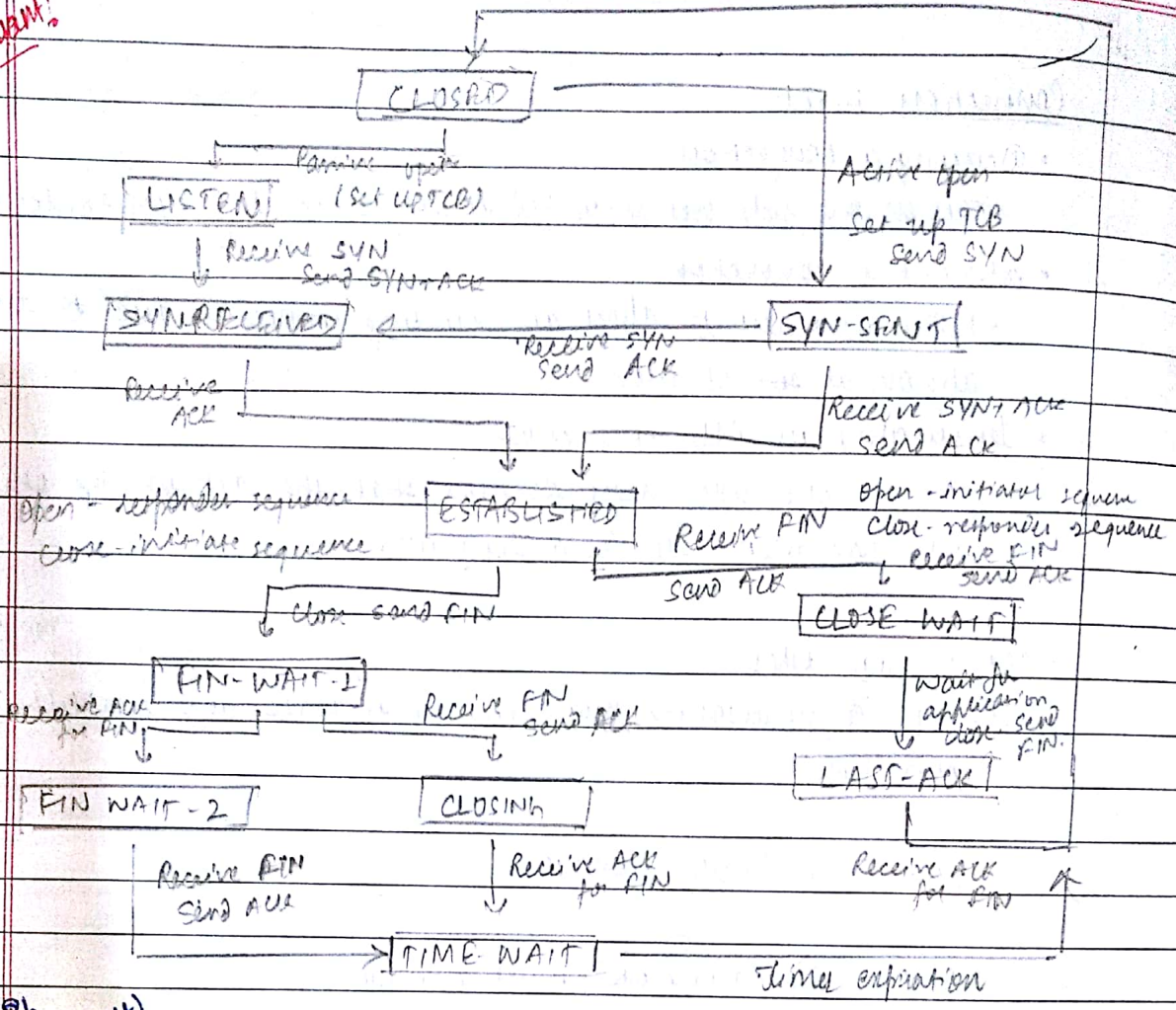
TCP message types

- SYN: A synchronize msg, used to initiate and establish the connection.
- FIN: A finish message.
- ACK: An acknowledgement message.

TCP States and state transition diagram

States	Description
CLOSED	No conn. exists
LISTEN	Passive open received, waiting for syn
SYN-SENT	SYN sent, waiting for ACK
SYN-RECEIVED	SYN-ACK sent, waiting for ACK
ESTABLISHED	Connection established, data transfer programs
CLOSE-WAIT	First FIN received, ACK sent, waiting for app to close
LAST-ACK	Second FIN sent, waiting for ACK.
FIN-WAIT-1	First FIN resent, waiting for ACK.
FIN-WAIT-2	ACK to first FIN received waiting for the second fin.
CLOSING	Both sides decided to close simultaneously.
TIME-WAIT	Second FIN received, ACK sent, waiting for 2MSL time-out

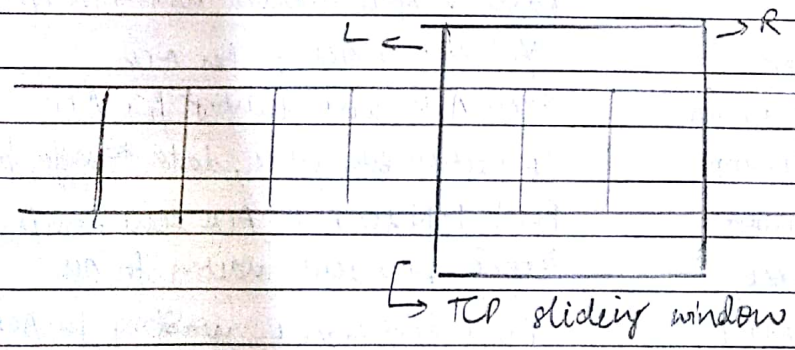
Important!



Lecture 18b
SDE / Second half

28/02/18

TCP/IP flow control



Byte oriented

variable size

If the window size increases then the no. of signals are increased.

Window close - Left wall moves right

Window shrink - Right wall moves left

Sliding window method of implementing flow control in TCP.

Receiver's window : $rwnd$
Congestion window : $cwnd$

Effective window size : $\min(rwnd, cwnd)$

MSL : Message sent ~~lost~~

- TCP's flow control is a mechanism to ensure the sender is not overwhelming the receiver with more data than it can handle.
 - A sliding window is used to make transmission more efficient as well as control the flow of data.
- Sliding window in TCP spans a portion.