Connection Setup:

1. 3 way handshake
    - SYN, ISN – make ISN random to avoid hi-jack
        1. allocate buffer (nbe, ISN) on receiver
        2. ISN can be a cookie to avoid allocating resources
    - SYN/ACK, ack is ISN + 1, Seq = ISN'
    - ACK
    - Sending FIN closes one side of the connection, the other side can still send data. (Can tie resources if did not receive the ACK). Do not need to wait 2 * TTL if seq. number for another connection is not the same as before.
    - Typically number of retry (SYN = 3, DATA = 12).

Sting:

1. Suppose you want to measure the internet, wants to know what the characteristic were – ICMP (control packets). Problem? ICMP tends to be turn off by servers.
2. TCP connection
    a. don't know where the lost occurred
    b. delay ACKs
3. Twist TCP to get the max. connection
    a. Send recv every other byte, stays in the recv buffer
    b. 1 3 5 7 9, sends a 2, gets an ACK for 3
    c. Suppose lost 3, sends a 2, did not get an ACK back
    d. Suppose sends 4, ACK for 5 or ACK for 1
4. Survey: 1/3 did not correctly implement TCP stack (e.g. no fast retransmit, etc.).

Congestion Control:

1. Fairness / Priority
2. Avoid collapse
3. BW matches the net (efficient)
4. Implementation:
    a. Hosts
        1. Incremental deployment path
        2. Easy to upgrade
    b. Routers
5. Marketing
    a. Another way to charge for BW during rush hour

Slow Start:

1. cwnd = 1 (initial window size)
    a. problem: delay ack slows performance
2. exponential growth, back off when timeout by factor of 2
3. slow start every every timeout, additive increase after a threshold
4. fast recovery: use duplicate acks as clocking
5. in general, enough buffering keeps the bw busy
6. 2 TCP sharing the same link (one slower, one faster)
    a. both will get lost when they exceed the bw
    b. progress toward fairness over a period of time with enough buffer space