CMSC417 Spring 2016 Lecture #17 4/6/2016

Agenda:
⇒ office hours change: Colin Wed 9:30-11:30a
⇒ exam on Monday
⇒ let us know if you need p3 grades by Mon@5p

⇒ TCP wrap-up
- ☐ delayed ack
- ☐ sack
- ☐ high bw-delay product links
- ☐ TCP fairness
- ☐ misbehaving receivers
- ☐ examples
- ☐ DDoS/CDNs

CMSC417 Spring 2016 Lecture #17 4/6/2016

Delayed Acks — RFC 1122

⇒ if there are lots of data packets sent very quickly, there are lots of acks
- most acks are redundant b/c they are cumulative
- not all are necessary

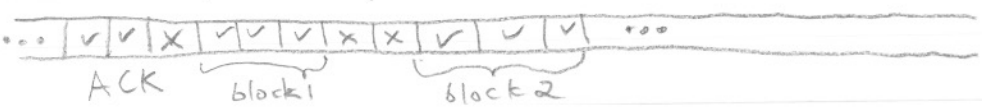
⇒ allow receivers to ack less frequently
- delay at most 500 ms
- ack at least every second segment

⇒ why?
- fewer ACKs is better (fewer packets)
- gives receiver some time to wait for data
  - piggyback ack on data for free

Selective ACKs (sack) — RFC 2018

⇒ it would be great if we knew what (if any) segments the receiver has beyond their cummulative ack

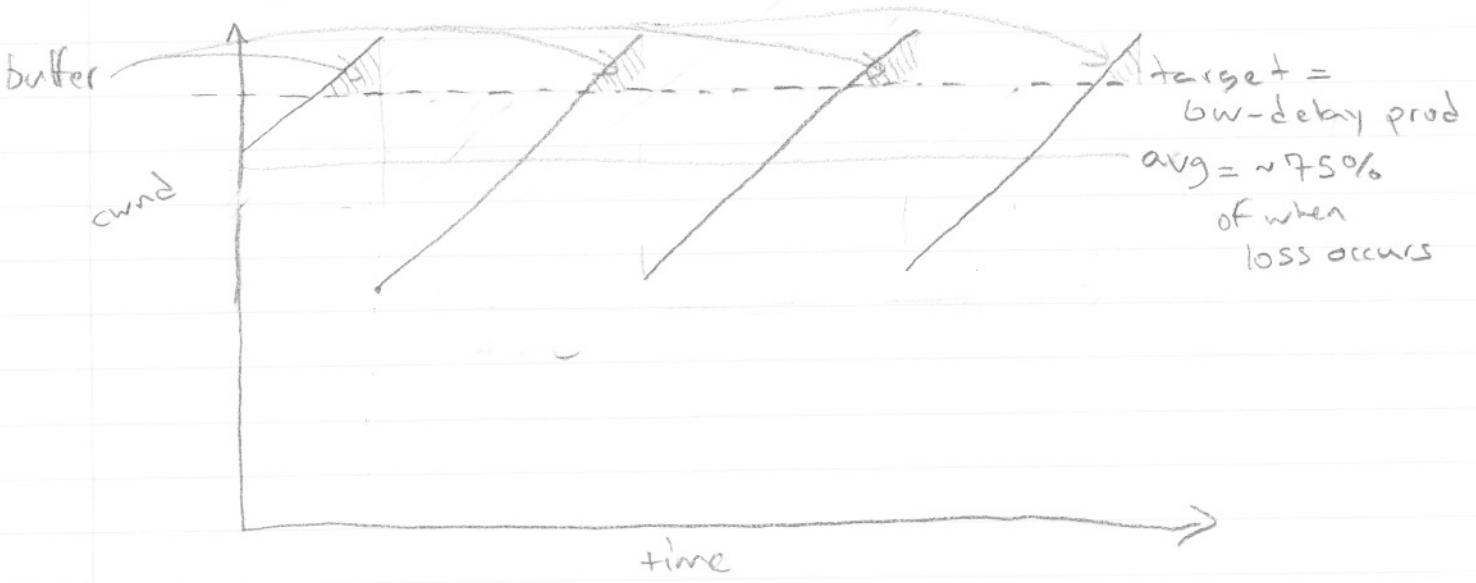
⇒ TCP options
- 8 bits kind (=4 or 5)
- 8 bits length
- 64-bit blocks (32-bit start, 32-bit end)
  - up to 20 bytes of TCP options
  - 4 blocks ⇒ 18 bytes ⇒ 4 blocks max
  - typically used with timestamp option that uses up 12 bytes ⇒ 3 blocks
  - allows notifying sender of 3-4 contiguous blocks past ACK



High bw-delay product connections!

⇒ ideally you want cwnd ≈ RTT × bandwidth

⇒ in practice cwnd does a sawtooth



⇒ for high delay-bw product links
- 75% of cwnd @ loss events << bw-delay prod
- esp. if buffers (at bottleneck switch/router) are small

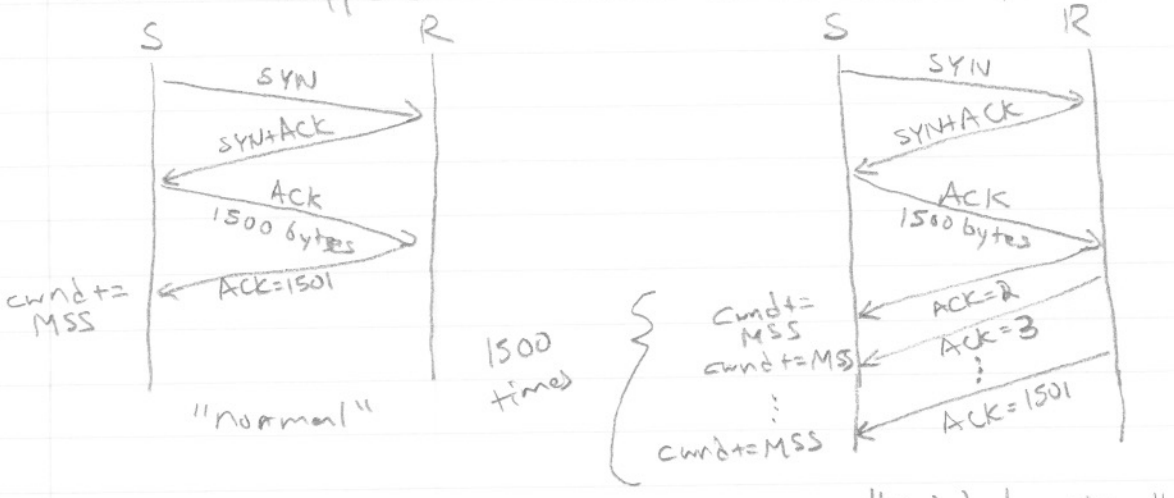
⇒ alternative TCPs target this
- BIC (binary increase congestion)
  - do a binary search for cwnd at high values instead of linear
- CUBIC
  - use shifts of x^3 instead of linear
- many, many more

⇒ all of this assumes you can use window scaling to make the advertised window big enough

TCP w/ Misbehaving Receivers

⇒ recall:
- slowstart: cwnd += MSS per ACK
- normal: cwnd += MSS/cwnd per ACK

⇒ what happens if recver acks each byte individually?



⇒ normal cwnd += MSS

⇒ misbehaving cwnd += MSS²

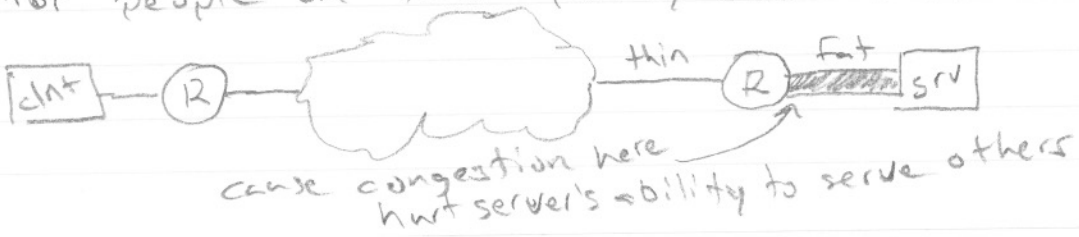
⇒ from a single packet ☐₀

problems:

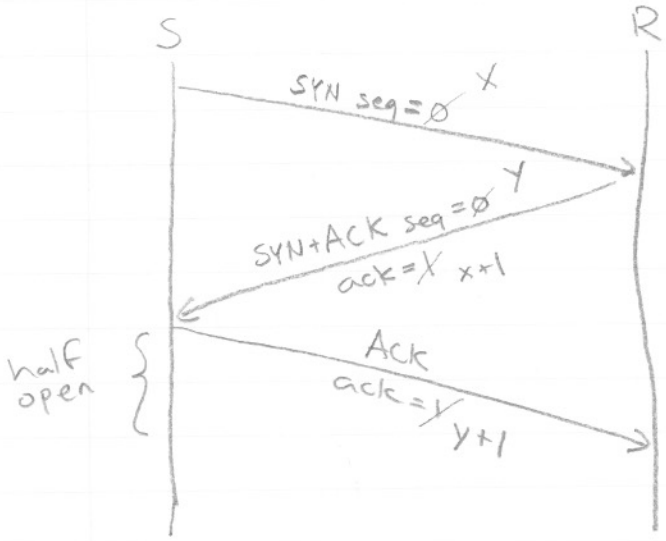
⇒ unfair advantage: get your connection to start faster and stay faster than others

⇒ congestion collapse: trick sender into sending too fast, if you do this to lots of people, might break the internet

⇒ Denial of Sevice: all packets likely won't reach you, but may cause problems for people on the path, even the server



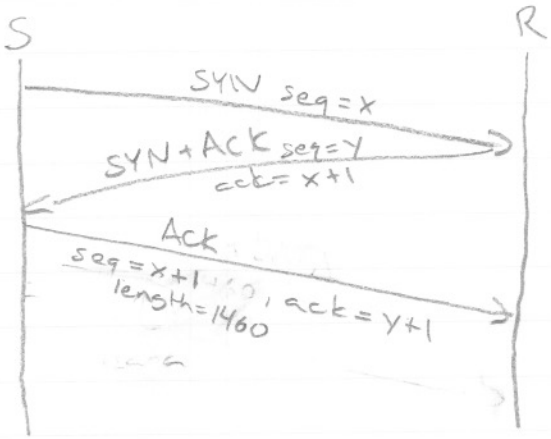
TCP examples



⇒ called the "3-way handshake"

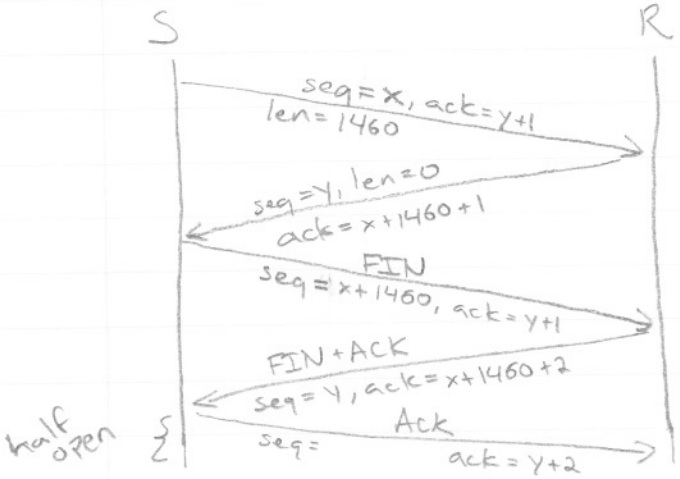
⇒ seq #s need not start at 0 (in practice are randomized)

⇒ first Acks are ACKing the SYNs



⇒ once <u>your</u> SYN is ACKed, you can start sending data

⇒ often you send data with your Ack for their SYN



⇒ first person to call close sends FIN

⇒ can still have to send retransmissions until FIN acked

⇒ FIN can go on a data packet