The Network Stack (2)

• The Transmission Control Protocol (TCP)
  • The TCP state machine
  • TCP congestion control
  • TCP implementations and performance
  • The evolving TCP stack
  • Lab 3 on TCP

• Wrapping up the Advanced Operating Systems lecture series
The Transmission Control Protocol (TCP)


TCP principles and properties

- Assumptions: Network may delay, (reorder), drop, corrupt IP packets
- TCP implements reliable, ordered, stream transport protocol over IP
- Three-way handshake: SYN / SYN-ACK / ACK (mostly!)
- Steady state
  - Sequence numbers ACK’d
  - Round-Trip Time (RTT) measured to time out loss
  - Data retransmitted on loss
  - Flow control via advertised window size in ACKs
  - Congestion control (‘fairness’) detects congestion via loss (and, recently, via delay: BBR)
- NB: “Half close” allows communications in one direction to end while the other continues
TCP congestion control and avoidance

- 1986 Internet CC collapse
  - 32Kbps → 40bps
- Van Jacobson, SIGCOMM 1988
  - Don’t send more data than the network can handle!
  - Conservation of packets via ACK clocking
  - Exponential retransmit timer, slow start, aggressive receiver ACK, dynamic window sizing on congestion, and (later) ABC
- ECN (RFC 3168), ABC (RFC 3465), Compound (Tan, et al, INFOCOM 2006), Cubic (Rhee and Xu, ACM OSR 2008), BBR (Cardwell, ACM Queue 2016)
TCP time/sequence graphs (Van Jacobson)

- Extracted from TCP packet traces (e.g., via `tcpdump`)
- Visualize windows, congestion response, buffering, RTT, etc:
  - X: Time
  - Y: Sequence number
- We can extract this data from the network stack directly using DTrace
  - Allows correlation/plotting with respect to other variables/events
    - E.g., TCP and socket-buffer state
- TCP time/sequence diagrams have since been extended to represent additional information
  - E.g., SACK (selective acknowledgement) blocks