

L41: Lab 5 - TCP Latency and Bandwidth

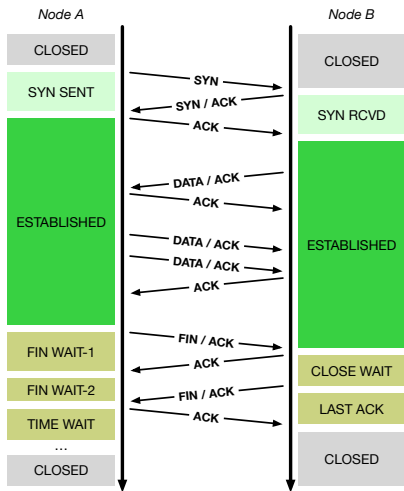
Dr Robert N. M. Watson

10 February 2016

L41: Lab 5 - TCP latency and bandwidth

- ▶ TCP congestion control
- ▶ TCP Protocol Control Block (TCPCB)
- ▶ Exploratory and experimental questions

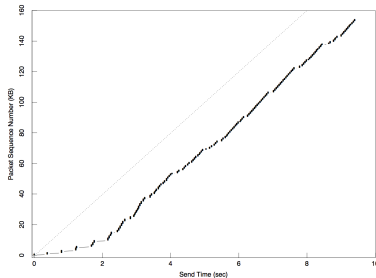
Lect 6: TCP goals and properties



- ▶ Network may delay, (reorder), drop, corrupt packets
- ▶ TCP: Reliable, ordered, stream transport protocol over IP
- ▶ Three-way handshake: SYN / SYN-ACK / ACK (mostly!)
- ▶ Sequence numbers ACK'd; data retransmitted on loss
- ▶ Round-Trip Time (RTT) measured to time out loss
- ▶ Flow control via advertised window size in ACKs
- ▶ Congestion control ('fairness') via packet loss and ECN

Lect 6: TCP congestion control and avoidance

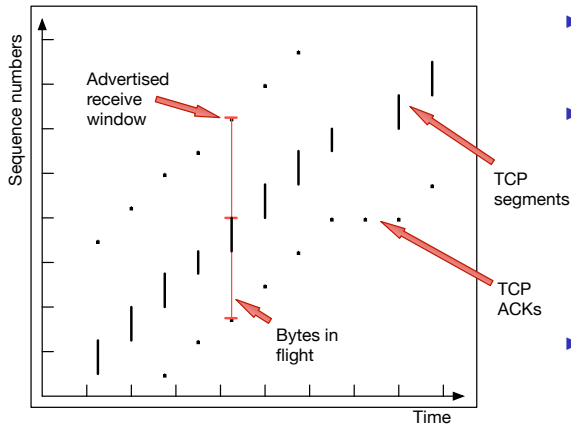
Figure 4: Startup behavior of TCP with Slow-start



Same conditions as the previous figure (same time of day, same Suns, same network path, same buffer and window sizes), except the machines were running the 4.3⁺ TCP with slow-start. No bandwidth is wasted on retransmits but two seconds is spent on the slow-start so the effective bandwidth of this part of the trace is 16 KBps — two times better than figure 3. (This is slightly misleading: Unlike the previous figure, the slope of the trace is 20 KBps and the effect of the 2 second offset decreases as the trace lengthens. E.g., if this trace had run a minute, the effective bandwidth would have been 19 KBps. The effective bandwidth without slow-start stays at 7 KBps no matter how long the trace.)

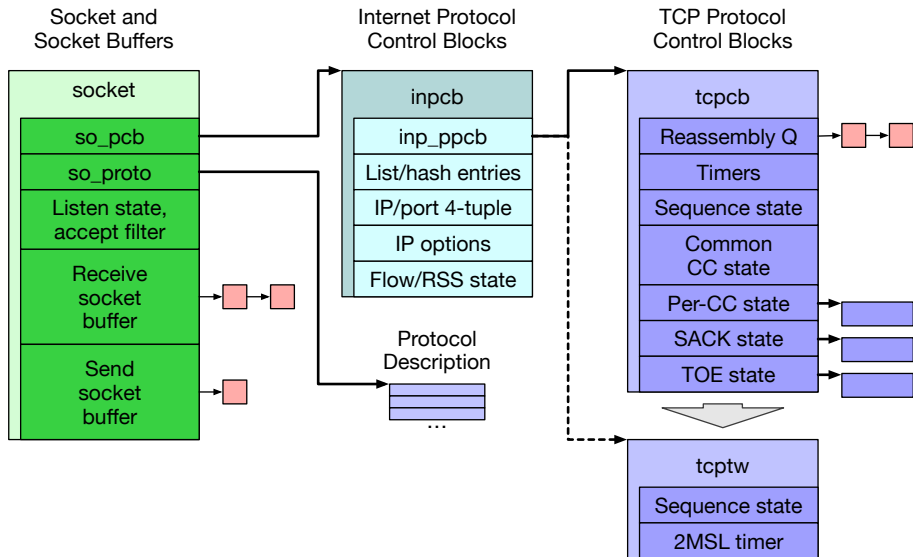
- ▶ 1986 Internet CC collapse
 - ▶ 32Kbps -> **40bps**
- ▶ Van Jacobson, SIGCOMM 1988
 - ▶ Don't send more data than the network can handle!
 - ▶ **Conservation of packets** via ACK clocking
 - ▶ Exponential retransmit timer, slow start, aggressive receiver ACK, and dynamic window sizing on congestion
- ▶ ECN (RFC 3168), ABC (RFC 3465), Compound (Tan, et al, INFOCOM 2006), Cubic (Rhee and Xu, ACM OSR 2008)

Lect 6: TCP time/sequence graphs



- ▶ Extracted from TCP packet traces
- ▶ Visualise windows, congestion response, RTT, etc.
 - ▶ X: Time
 - ▶ Y: Sequence number
- ▶ We can also extract this data from the stack using DTrace

Lect 6: Data structures - sockets, control blocks



`tcpcb` sender-side data-structure fields

Described in more detail in the lab assignment:

`snd_wnd` Last received advertised flow-control window.

`snd_cwnd` Current calculated congestion-control window.

`snd_ssthresh` Current slow-start threshold: if `snd_cwnd` is less than or equal to `snd_ssthresh`, then TCP is in slow start; otherwise, it is in congestion avoidance.

- ▶ Instrument `tcp_do_segment` using DTrace to inspect TCP header fields and `tcpcb` state
- ▶ Packets on 'client' and 'server'; `tcpcb` only on 'server'.
- ▶ Use as input to time–sequence-number or time–bandwidth plots.
- ▶ Make sure to flush the TCP host cache between benchmark runs.

Exploratory questions

- ▶ As latency varies, how does overall bandwidth change?
- ▶ How does using a fixed rather than auto-resized socket buffer affect advertised receive window? Use a fixed buffer size (1MB).
- ▶ How quickly does TCP enter steady state (i.e., shift out of slow start) at a 10ms RTT? Is this a product of the congestion or the socket-buffer limit?

Experimental questions for the lab report

- ▶ Plot network latency vs. TCP bandwidth. Does linear increase in latency mean linear decrease in bandwidth? How does socket-buffer auto-resizing help/hurt/not change performance?
- ▶ Explore the effects of socket-buffer limits and stack graph information on the flow-control versus congestion-control limits. How does socket-buffer auto-resizing help/hurt/not change performance?
- ▶ Explore how latency affects the time taken to leave slow start.

This lab session

- ▶ Ensure that you are able to properly extract both TCP header and `tcpcb` fields from the `tcp_do_segment` FBT probe
- ▶ Generate the data for a time–bandwidth graph
- ▶ Generate the data for a time–sequence-number graph
- ▶ Ask us if you have any questions or need help