Finishing TCP

(and some higher-level networking libraries)

11/11/15
Simulation lab

After 1M time steps, each user has communication probability 0.05
Fraction is actually failing/(delivered+failing)
Rest of the term

• Friday
  – Go over chat program (and HW 6?)
  – Overview and questions

• Final exam
  – Available Friday evening (anticipated)
  – Due at the end of finals (Wednesday at 10pm)
  – 2 programming problems, ~5 conceptual problems
  – Comprehensive, less code-oriented than midterm
How long should the sender wait to retransmit?

A. A fixed short amount of time
B. A fixed long amount of time
C. The time it took the last packet to get acknowledged
D. Depends on the application
E. Why wait? Just retransmit continuously until the packet is acknowledged!
How long should the sender wait to retransmit?

• With ave. round-trip time RTT and standard deviation D, can retransmit after RTT + 4D to timeout on ~1% of non-lost packets
How long should the sender wait to retransmit?

• With ave. round-trip time RTT and standard deviation D, can retransmit after RTT + 4D to timeout on ~1% of non-lost packets

• Per connection, update estimates with each acknowledgement:

\[
RTT_{\text{new}} = \alpha RTT_{\text{old}} + (1 - \alpha) \text{new\_time} \\
D_{\text{new}} = \alpha D_{\text{old}} + (1 - \alpha) |RTT - \text{new\_time}| \\
\]

(\alpha determines relative weight of history)
To handle out of order or duplicate packets, each packet has a sequence number. What should this be?

A. Packet number (1\textsuperscript{st}, 2\textsuperscript{nd}, etc) among packets sent in one direction on that connection

B. Packet number among packets sent in either direction on that connection

C. Byte number of its first byte of data among data sent in one direction on that connection

D. Byte number of its first byte of data among data sent in either direction on that connection

E. The number of times the packet has been sent
To handle out of order or duplicate packets, each packet has a sequence number. What should this be?

A. Packet number (1\textsuperscript{st}, 2\textsuperscript{nd}, etc) among packets sent in one direction on that connection

B. Packet number among packets sent in either direction on that connection

C. Byte number of its first byte of data among data sent in one direction on that connection

D. Byte number of its first byte of data among data sent in either direction on that connection

E. The number of times the packet has been sent
Sequence numbers

- Also used in acknowledgements, which just give the largest sequence number received.
- Don’t start at 0 for every connection to avoid confusion when a connection is closed and reopened.
- Leads to “handshake” at beginning of connection:
  - SYN: client sends its desired sequence number.
  - SYN ACK: server acknowledges and sends its.
  - ACK: client acknowledges.
Flow control

• With every acknowledgement, the receiver includes the amount of available space in its receiving buffer (its window size)
• Window size also updated as receiving application accepts the data
• Sender can’t send more than this
  (one exception: allowed to make 1-byte query for current size)
Silly Window Syndrome

• Naive implementations can suffer if the application reads one byte at a time
  – Why?
Silly Window Syndrome

• Naive implementations can suffer if the application reads one byte at a time
  – Why? (Lots of traffic to update the window size. Analogous to car taking every exit on the freeway to top off the tank.)
Silly Window Syndrome

• Naive implementations can suffer if the application reads one byte at a time
  – Why? (Lots of traffic to update the window size. Analogous to car taking every exit on the freeway to top off the tank.)

• Solution: Receiver delays sending update until more of window is free (sender can also wait)
Similar problem for sender

- What if the data to be sent is generated 1 byte at a time? (e.g. telnet sends chars as soon as they are typed)
Similar problem for sender

• What if the data to be sent is generated 1 byte at a time? (e.g. telnet sends chars as soon as they are typed)

• Nagle’s algorithm: send first byte immediately, but buffer others until the first byte is acknowledged
So what fields does TCP header have?