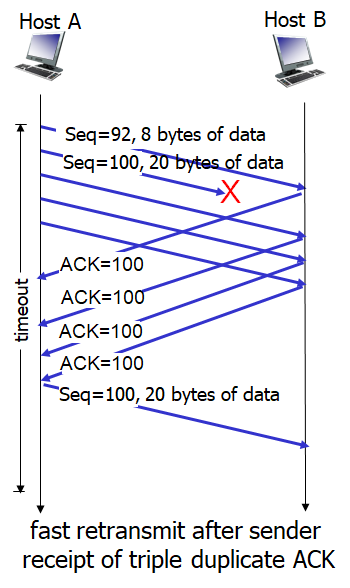
TCP’s implementation of a RDT protocol

* Cumulative ACK
* Single transmission timer
* Pipelining
* Retransmissions triggered by:
  + Duplicate ACKs
  + Timeout events

Sender Events

* Data received from application
  + Create segment with #
    - Seq # is byte stream number of first byte
  + Start timer
* Timeout
  + Retransmit segment and restart timer
* ACK received
  + If ACK for unACKed segment:
    - Update what is known to be ACKed and start timer if still unACKed segment

Fast Transmit

* Timeout can grow rapidly
  + Doubles every time timer expires without ACK
* Detect lost segments with duplicate ACKs
  + Sender often sends segments back-to-back
  + Will result In duplicate ACKs if segment is lost
* If sender receives 3 duplicate ACKs for same segment, it will immediately resend segment with smallest seq #
* Example: Three duplicate ACKs for seq#100
  + Host A dropped pkt with seq#100
  + Host B re-ACKs all new pkts with seq#100
  + Host A resends pkt with seq#100 when it receives three duplicates before timeout
  + Unlike GBN, only sends seq#100

Flow Control

* Congestion control about preventing network from getting overloaded and dropping packets
* Flow control about preventing receiver buffer from getting overloaded and dropping packets
  + Data not delivered to program immediately, buffers until requested
  + Possible to overflow buffer
  + Flow control is a speed matching service, matching speed of requester reading
* Sender maintains receive window that allows it to keep track of free buffer space
  + Receiver communicates buffer size to sender using receive window field in header
    - Typically between 4 kB and 8 MB
* Receive buffer (receiver)
  + Example: Out-of-order segments discarded
    - Compute space used in buffer
    - Put space remaining in TCP header when talking to sender
* Receive buffer (sender)
  + Keep track of how much un-ACKd data has been sent
  + Space remaining => lastByteSent – lastByteAcked
* Receive buffer – problem
  + Host B has 0 space, doesn’t send a packet to Host A
    - Host A will never learn
  + Solve: Host A can keep sending 1 byte segments when space remaining = 0 bytes

Connection management:

* TCP connection established three-way handshake
  + Agreement to establish connection
  + Agreement on parameters
* Three steps:
  + Client sends SYN segment, random seq#
  + Server responds with SYN segment, picks random seq#, sets ACK to client\_seq#+!, allocates buffer
  + Client sets SYN bit to 1, ACKs last segment from server
* Why?
  + Communicates Client\_ISN
  + Ensure client knows server knows Client\_ISN
  + Communicate Server\_ISN to client
  + Ensure server knows client knows Server\_ISN
* Closing a connection:
  + Each must send segment with FIN set to 1, and each must respond to segment with ACK
* Rejecting:
  + If server receives SYN for a bad port, responds to segment with RST bit to 1

# Congestion Control