CS 372 Lecture #17

Reliable data transfer with TCP

- error detection/handling
- sequencing
- acknowledgements
- retransmission

- See textbook for development of reliable data transfer (using finite state diagrams)

Note: Many of the lecture slides are based on presentations that accompany *Computer Networking: A Top Down Approach,* 6th edition, by Jim Kurose & Keith Ross, Addison-Wesley, 2013.

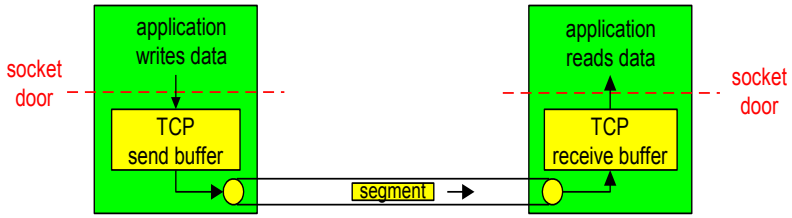
Error handling

- How can sender know that an error has occurred?
  - Bit flipped
  - Missing packet
  - Out-of sequence
- What should the receiver do?
- What should the sender do?

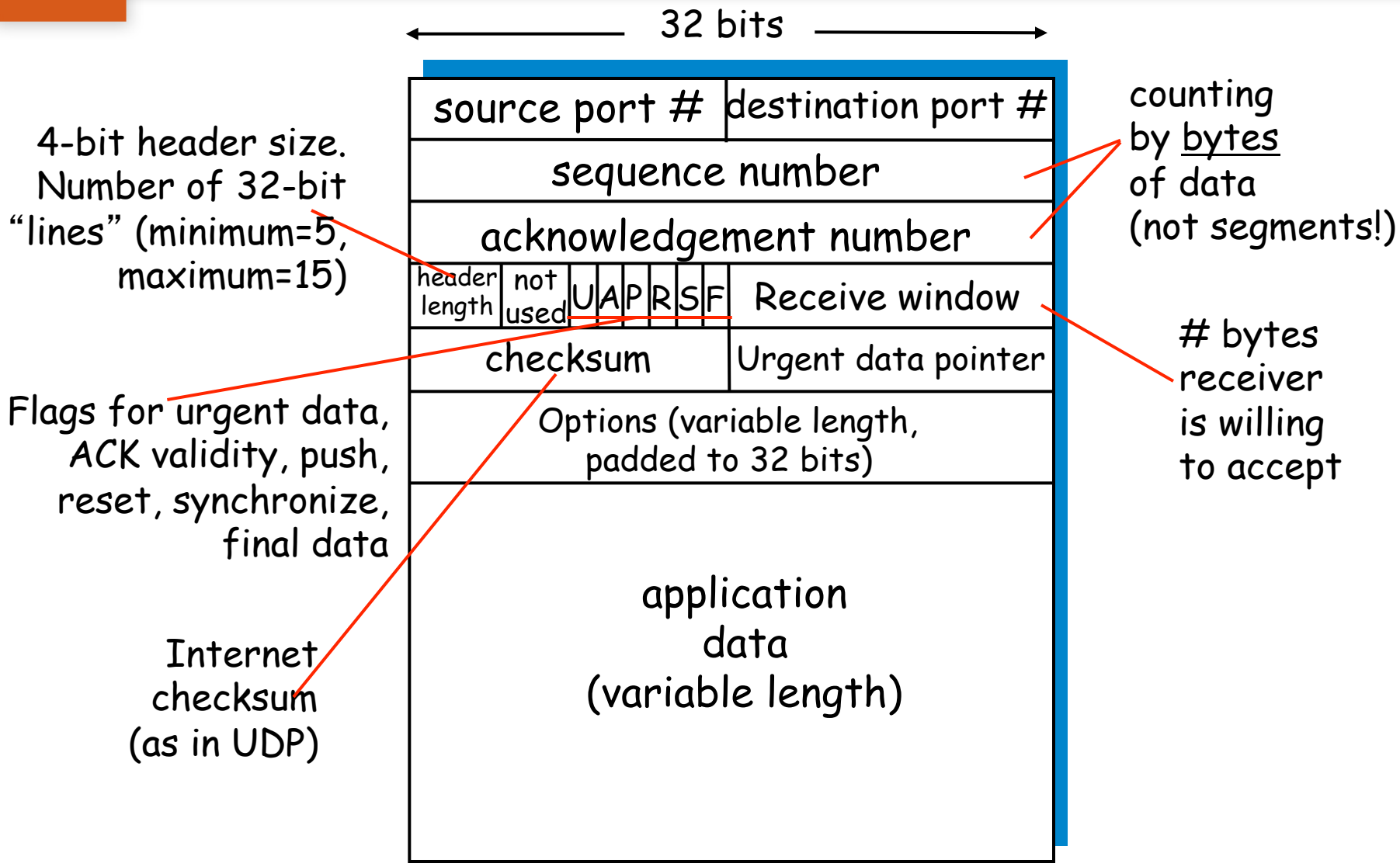
- TCP uses acknowledgement and retransmission

TCP: Overview [RFCs: 793, 1122, 1323, 2018, 2581]

- connection-oriented:
  - handshake (exchange of control messages) initializes sender/receiver state before data exchange
- point-to-point:
  - one sender, one receiver
  - full duplex (bi-directional data flow in same connection)
- in-order byte steam:
  - cumulative byte count
  - acknowledgement of bytes received
- pipelined:
  - TCP congestion and flow control
  - send & receive buffers
- flow controlled:
  - sender will not overwhelm receiver
  - MSS: maximum segment size



TCP segment structure



TCP sequence numbers

- Segments are sent as a stream of bytes
  - Not separated into records, data types, etc.
  - Protocol keeps count of data bytes sent
    - for each segment, protocol puts byte stream number of first byte in segment's data into "sequence number" field
- Sender keeps a copy of each segment until acknowledged by receiver

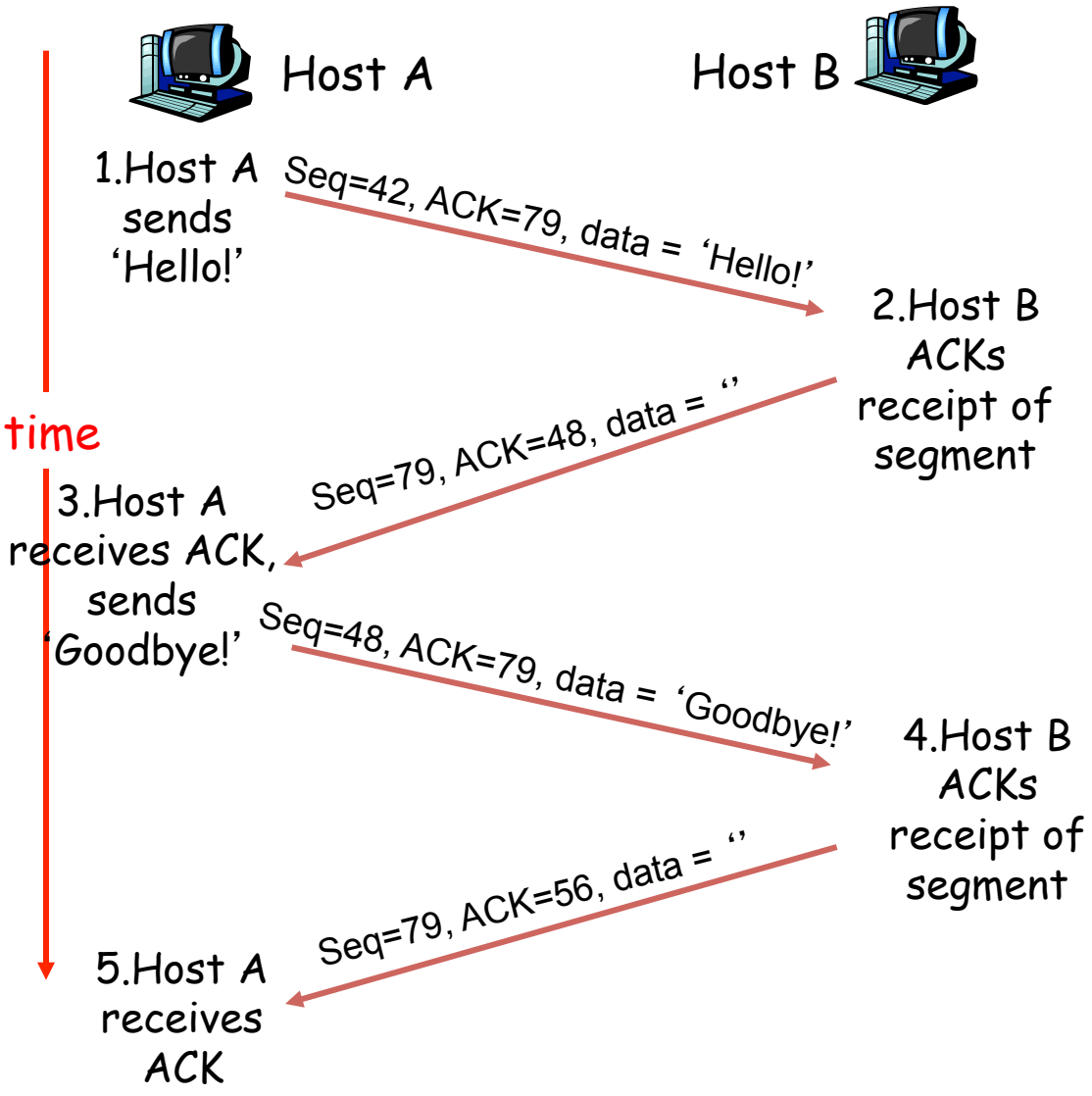
- Sender's segment also has "acknowledgement number"
  - TCP is full-duplex. Data can flow in both directions simultaneously
  - Typical (nothing to ACK) segment has a bogus number in the "acknowledgement number" field

TCP acknowledgement numbers

- Receiver acknowledges all correct segments received.
  - Error segments are not acknowledged at all
- Header for acknowledgement segment (ACK) is the same as for other segments
  - receiver calculates data bytes received by subtracting 4 x "header length" from TCP segment total byte-count
  - "acknowledgement number" field contains the number of the next byte expected
- ACK may also contain data
  - TCP is full-duplex. Data can flow in both directions simultaneously
  - Typical (no data) ACK has a bogus number in the "sequence number" field

Simple TCP scenario (no errors)

1. Host A sends "Hello!" to Host B. Suppose the 'H' is byte #42 in this sequence. ACK (#79) is bogus in this scenario
2. Host B receives segment, determines data length is 6, so next expected byte number is 42+6 = #48. Host B sends ACK, but sends no data.
3. Host A receives ACK, and sends "Goodbye!" to Host B. 'G' is byte #48 in this sequence.
4. Host B receives segment, determines data length is 8, so next expected byte number is 48+6 = #56. Host B sends ACK, but sends no data.
5. Host A receives ACK.

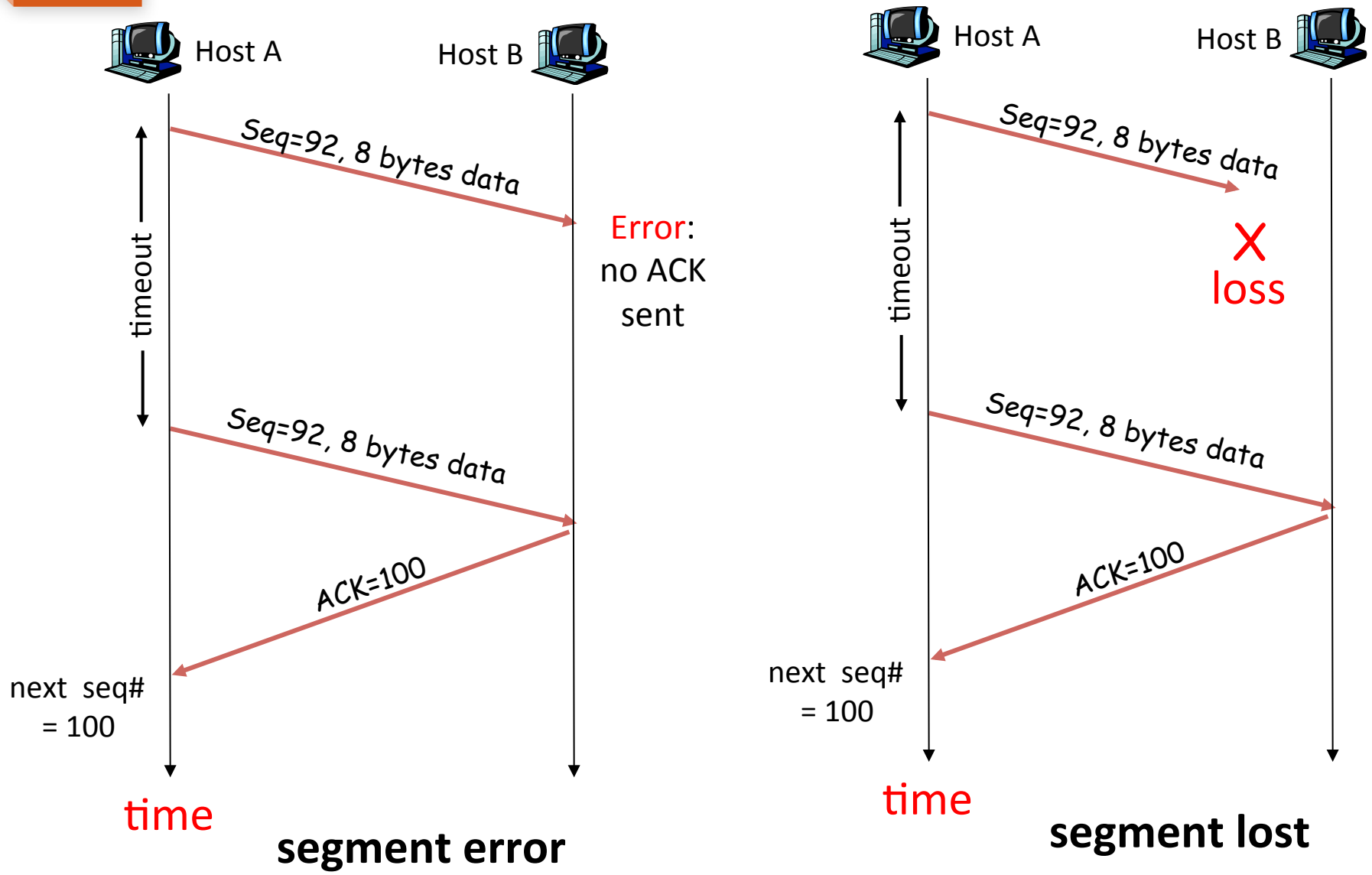


TCP retransmission

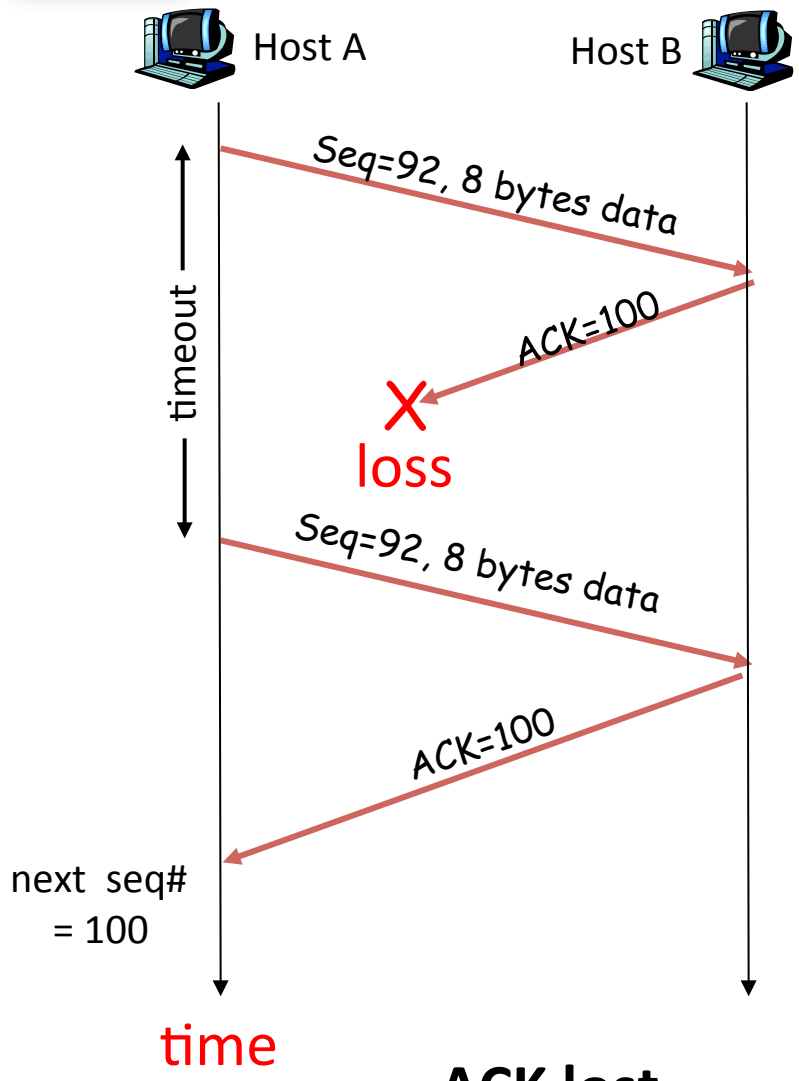
- Sender sets a count-down timer for each segment sent
  - if timer expires before ACK received … re-send
  - if segment error, ACK will never arrive

- Receiver can detect and discard duplicates
- if ACK is delayed (arrives after re-send)
- if ACK is lost

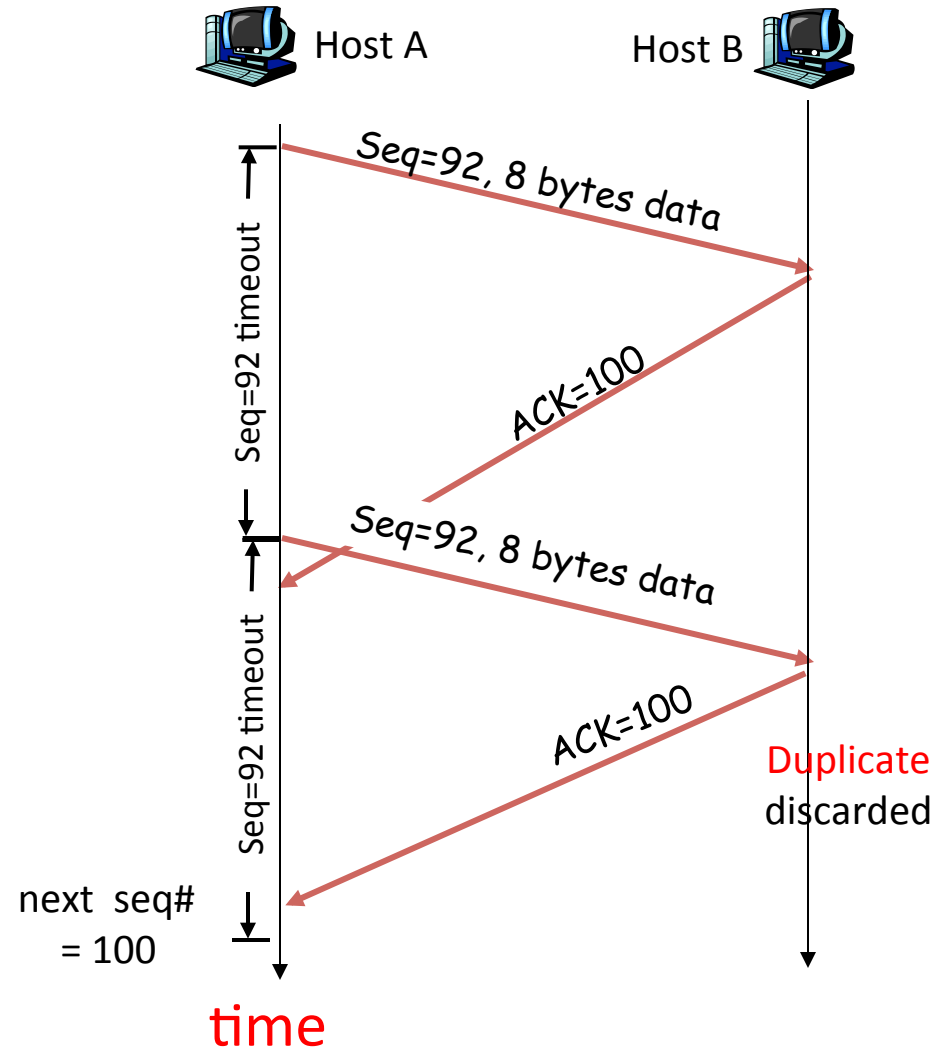
TCP retransmission scenarios



TCP retransmission scenarios



ACK lost



ACK delayed

Summary Lecture #17

- TCP error handling
  - error detection
  - byte sequencing
  - acknowledgement (ACK)
  - retransmission