CS 372 Lecture #19

Reliable data transfer with TCP

- pipeline errors
  - detection
  - handling

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.

TCP sender events

Data comes down from application layer:

- Create segments with sequence numbers
  - sequence # is byte-stream number of first data byte in segment
- Start countdown timer (if not already running for a previous segment)
- Send segments

If countdown timer expires:

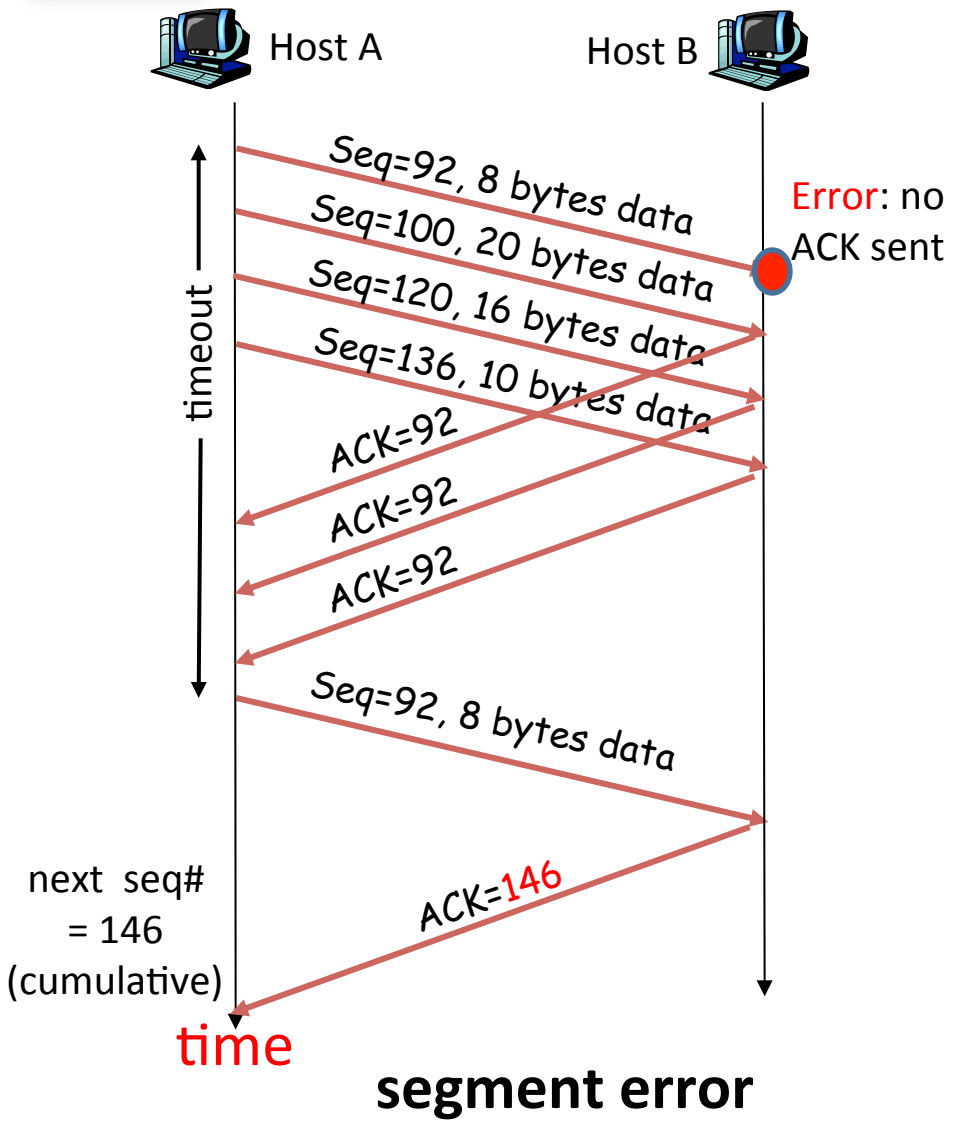
- Retransmit segment that caused timeout
- Restart timer

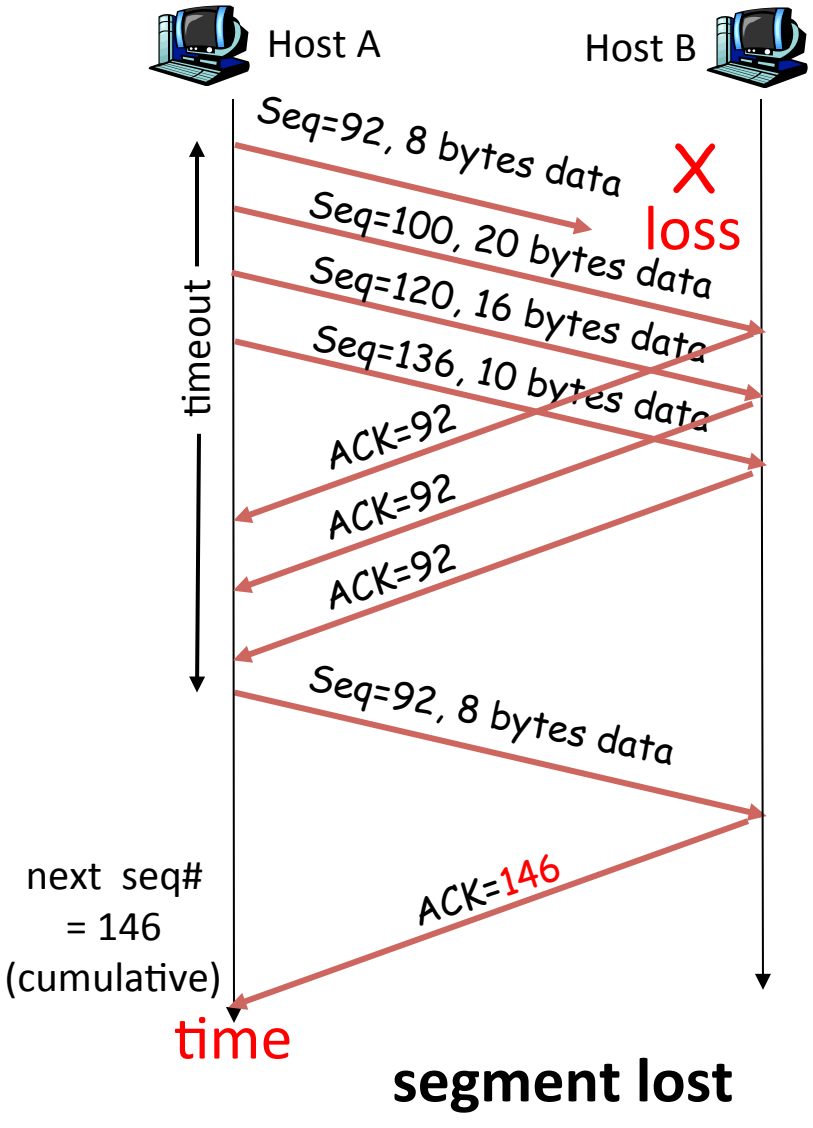
If ACK received:

- Check to see if ACK includes previously unACK'ed segments
  - update what is known to be ACK'ed
  - restart timer if there are outstanding segments

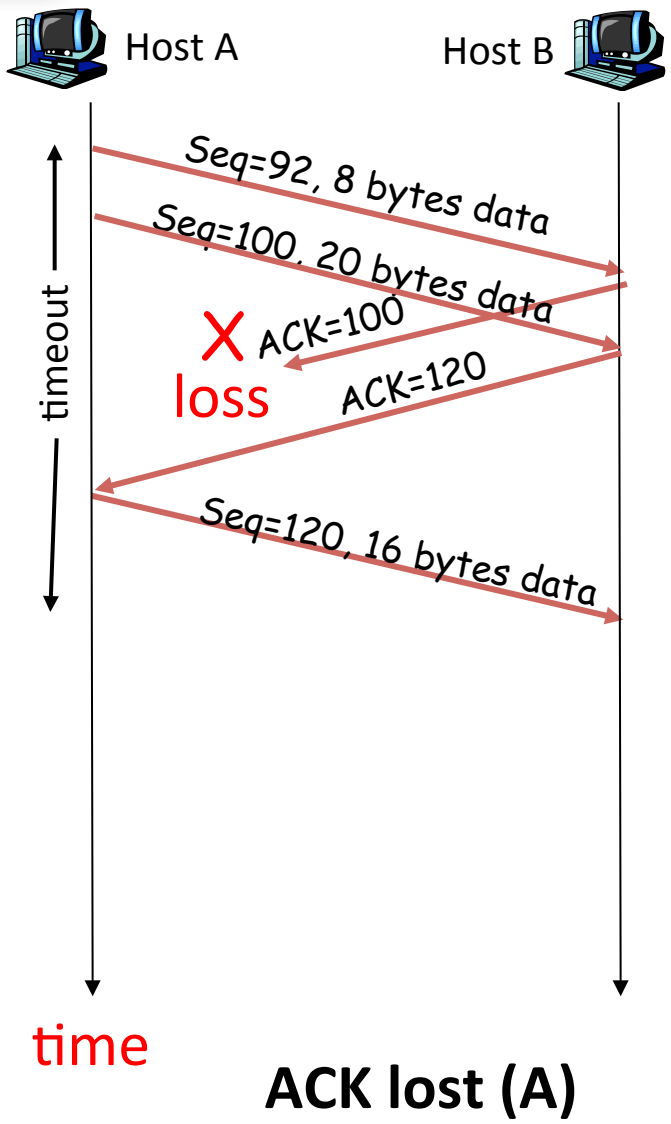
Note: ACK's are cumulative. The ACK # is always the next expected byte number. This implies that all previous bytes have been accounted for.

Pipeline retransmission error scenarios (cumulative ACK)

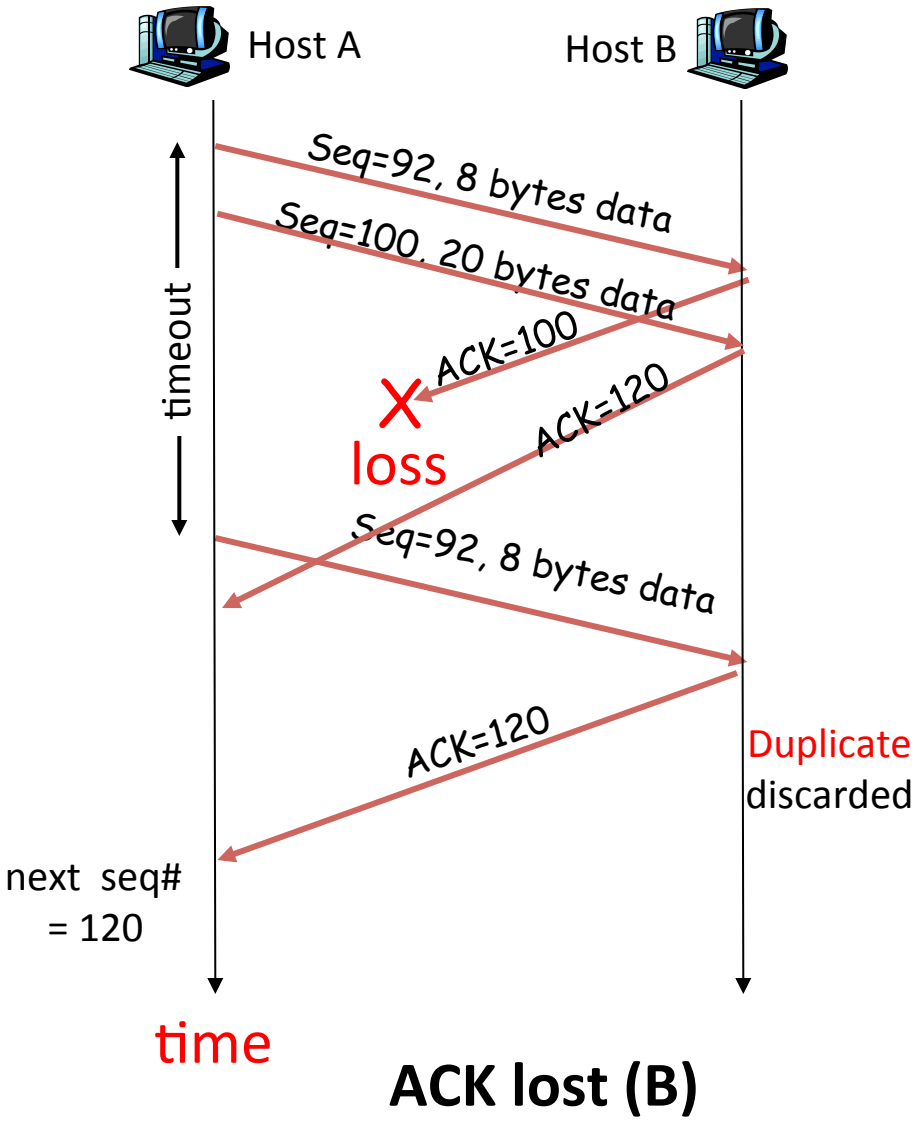




Pipeline retransmission error scenarios (cumulative ACK)

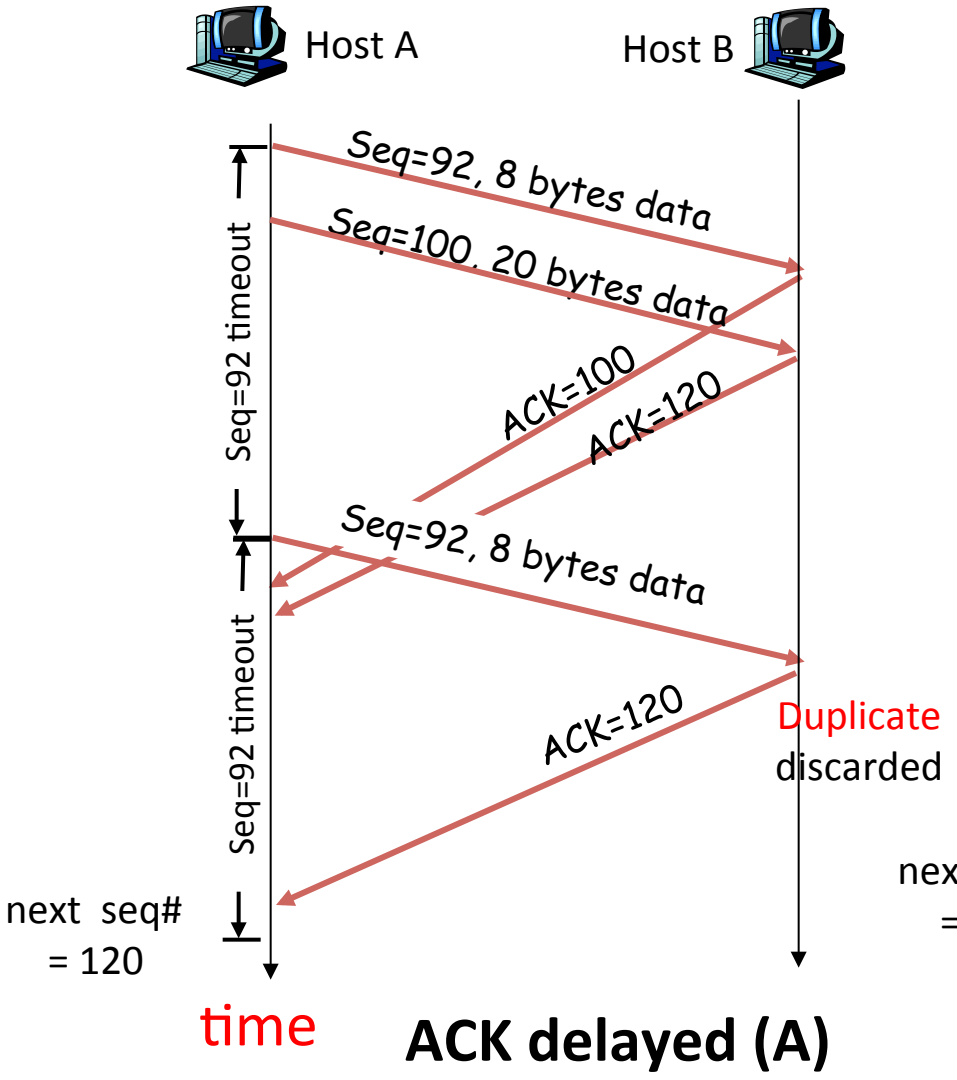


ACK lost (A)

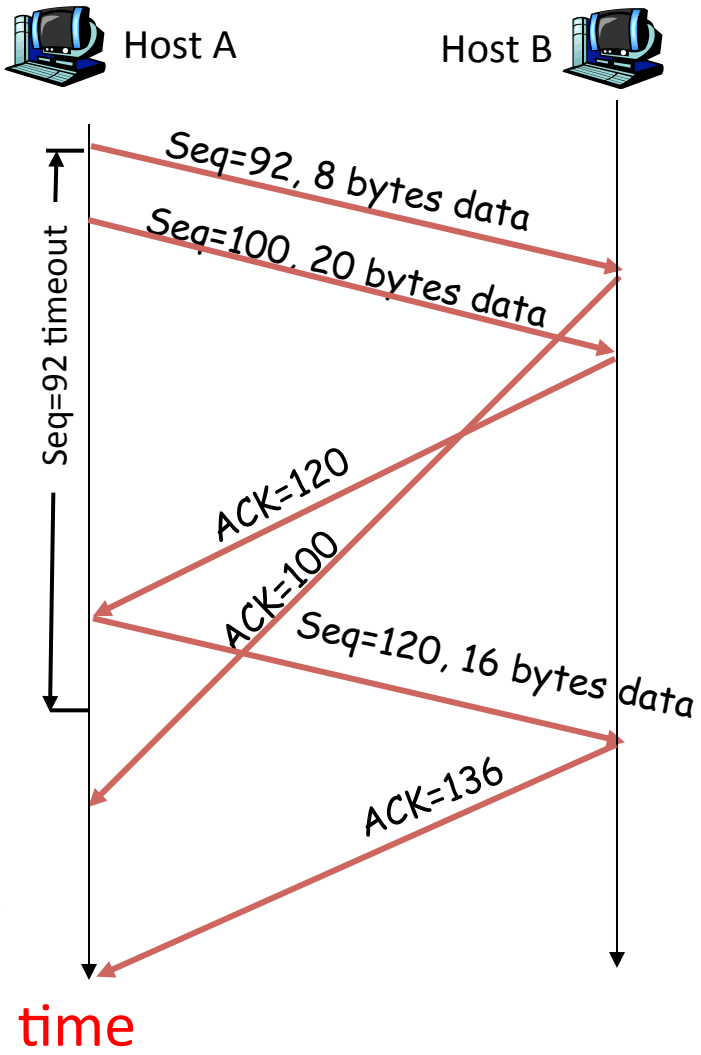


ACK lost (B)

Pipeline retransmission error scenarios (cumulative ACK)



ACK delayed (A)

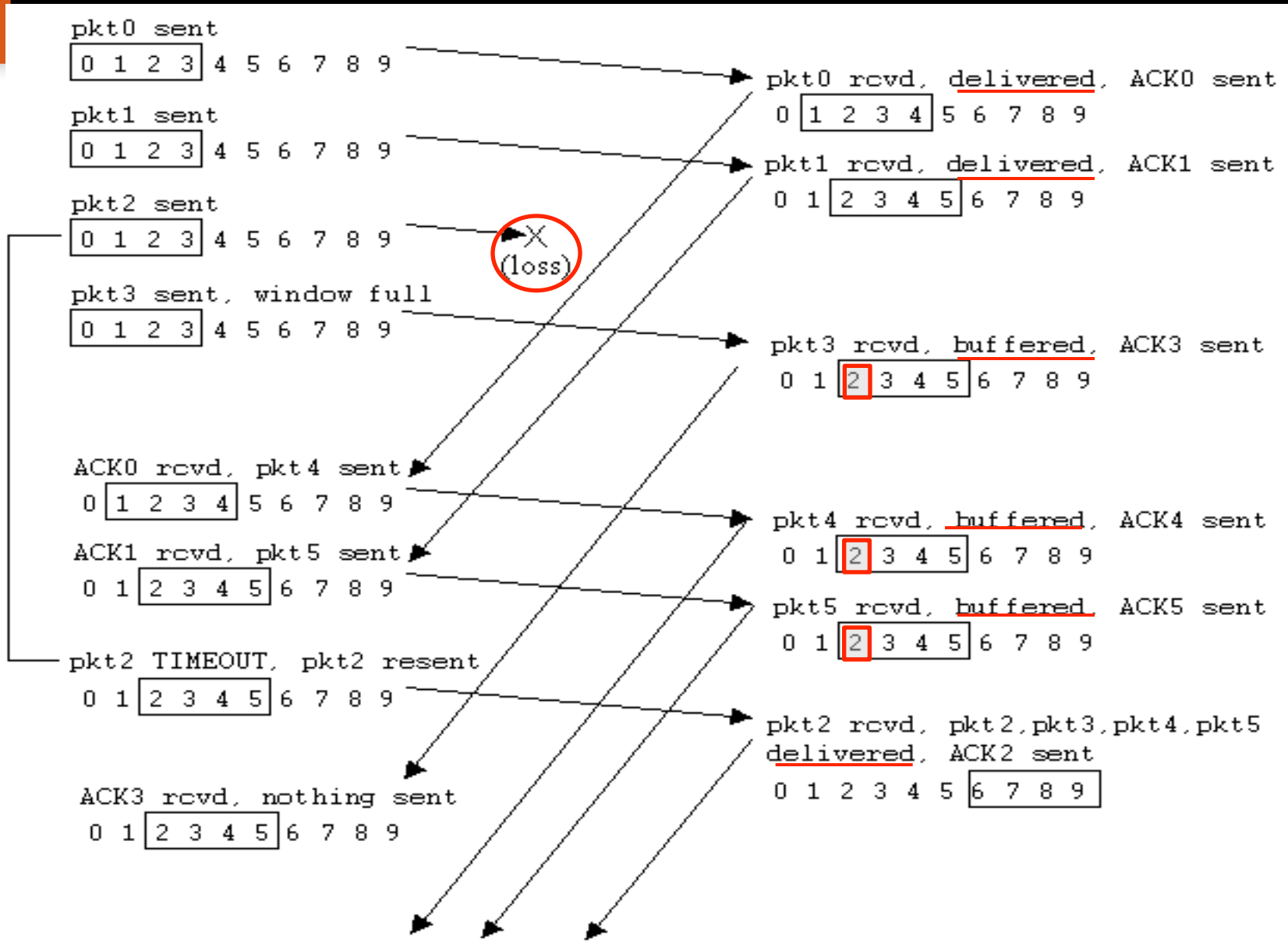


ACK delayed (B)

Selective Repeat protocol

- Sender can have up to "window size" un-ACK'ed packets in pipeline
- Receiver ACKs individual packets
- Sender maintains timer for each un-ACK'ed packet
  - When timer expires, retransmit only the un-ACK'ed packet

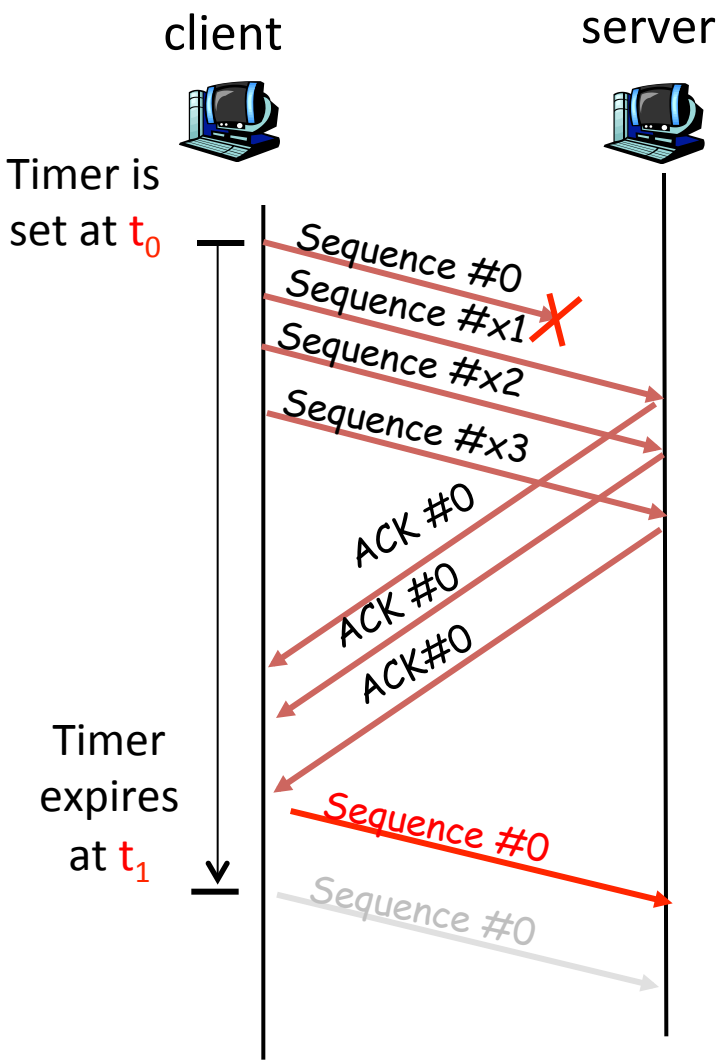
Selective Repeat protocol example



TCP Fast Retransmit

- Suppose that the packet with sequence #0 gets lost
  - Q: When will the packet with sequence #0 get retransmitted?
  - A: typically at t_1 . We think it is lost when the timer expires
- Can we do better??
  - Why wait till timeout?
    - We already know the packet is lost.

Remember:
- **Sequence#** is the number (in the data stream) of the first byte of the sent segment
- **ACK#** is the number of the next byte expected by the receiver.

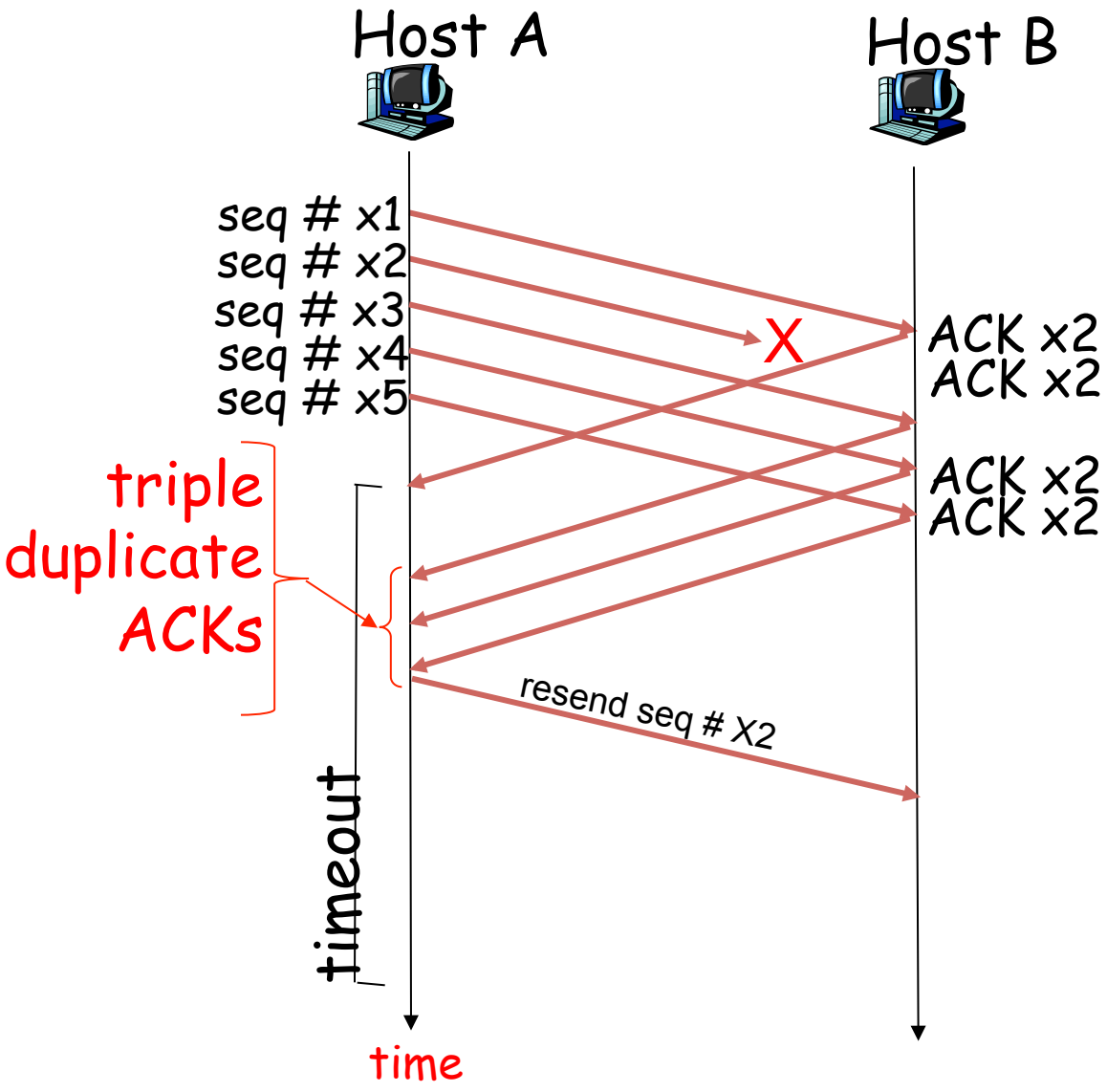


TCP Fast Retransmit

If we receive many duplicate ACKs for sequence #x2 …

… it means packet with sequence #x2 is lost.

Fast retransmit => better performance



TCP Fast Retransmit recap

- Receipt of duplicate ACKs indicate loss of segments
  - Sender usually pipelines segments
  - If segment is lost, there will likely be many duplicate ACKs.

This is how TCP works:

- If sender receives 3 ACKs for the same data, it supposes that segment after ACK'ed data was lost

- fast retransmit:
  - resend segment before timer expires
  - better performance

Summary Lecture #19

- TCP
  - cumulative ACKS
  - count-down timer
  - handling segment errors, segment loss
  - handling lost ACK, delayed ACK
- Retransmission protocols
  - Selective-Repeat
  - fast retransmission

- Next lecture: review for midterm exam