CS 144 Section #2
October 1, 2010

Aki Kobashi

Announcements

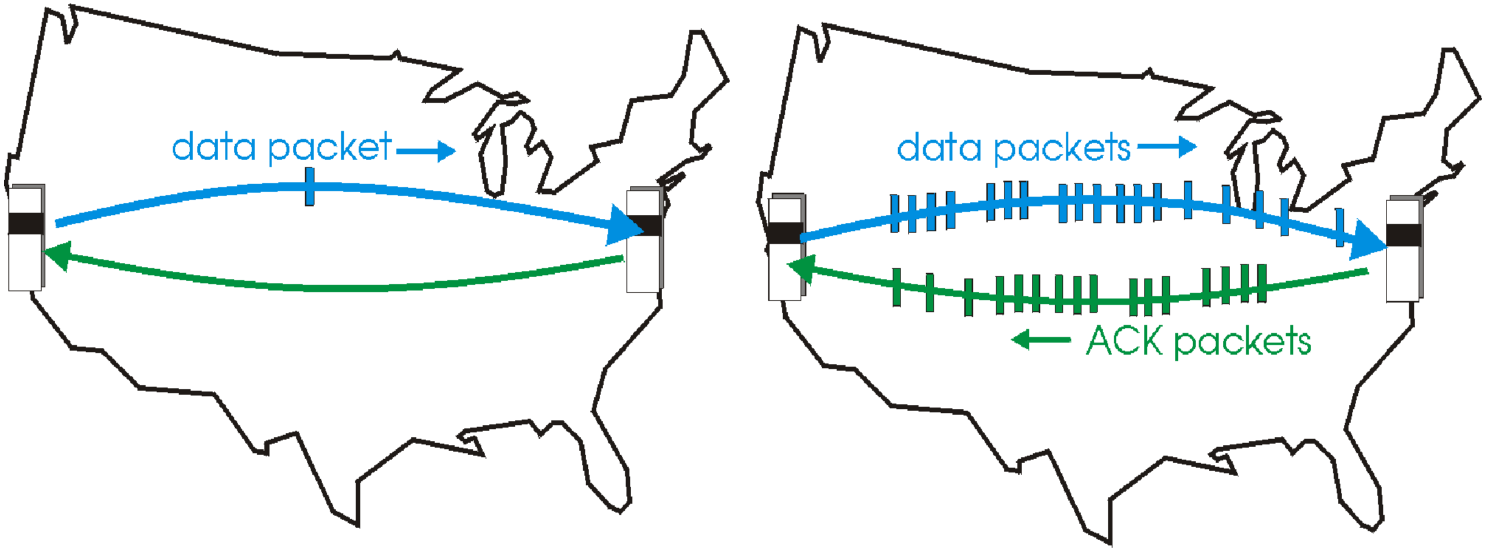
- Lab 1
  - Was due yesterday (Thu)
  - Have until midnight on Saturday to submit for up to 90% credit
- Lab 2
  - Due Thursday, Oct. 7 at beginning of class
  - Go to class and get an automatic extension until midnight that day

Lab 1 Review

- Stop-and-Wait
  - Pros
    - Easy implementation
    - Low memory usage
  - Cons
    - Inefficient use of bandwidth
    - Throughput mostly capped by RTT, not BW

Lab 2 Overview

- Sliding-window
  - Can have multiple unacknowledged packets
  - Need to handle packet reordering
- Lab 1 Grade = Max(Lab 1, Lab 2)



(a) a stop-and-wait protocol in operation

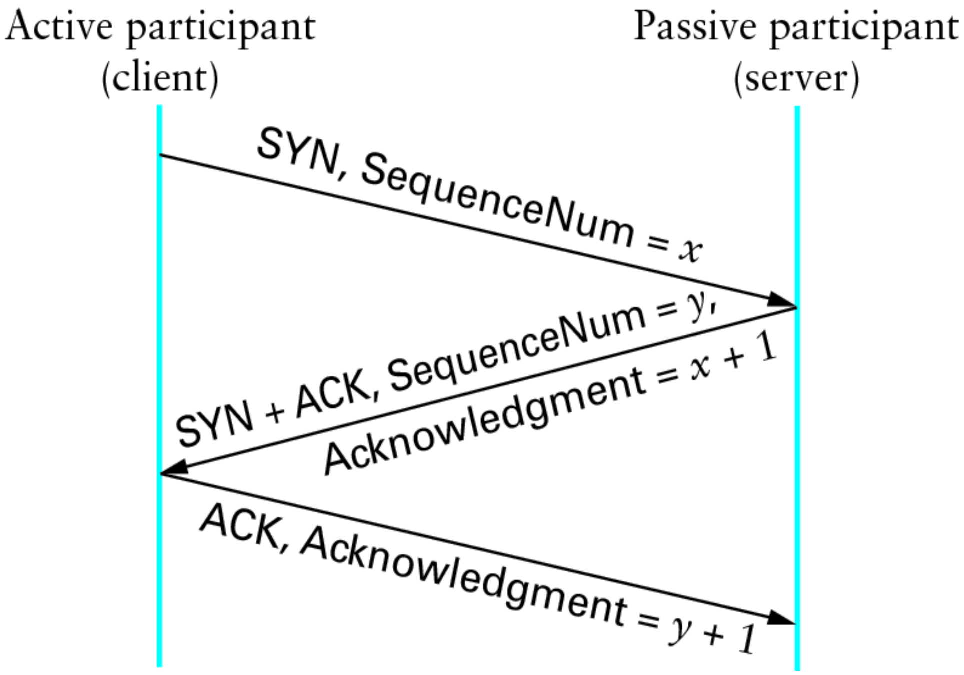
(b) a pipelined protocol in operation

Lab 2 Demo

TCP Review

- Sacrifices timeliness for accuracy
- How would we reliably stream live TV?
- Connection setup
- Connection teardown
- Two types of control:
  - Flow Control
  - Congestion Control

TCP Connection Setup



- 3-way handshake
- Learn each other’s sequence numbers
- Establish window size

TCP Connection Teardown

- Modified 3-way handshake
- Need to verify both sides are truly closed (FINs get ACKed)
- Both sides need to send FINs

- A → B: FIN
- B → A: ACK
- B → A: FIN
- A → B: ACK

Sliding Window Review

- Sliding window protocol tries to fill the pipeline more efficiently than stop and wait
- Sender:
  - Verify packets are reaching destination via ACKs before sending more data
- Receiver:
  - Inform sender if packets are missing via cumulative ACKs
  - Discard packets clearly out of range

Flow Control Review

- Purpose: prevent sender from sending too fast (or too slow)
- Implemented by the sliding window protocol
- Window size advertised by receiver
- Sender adjusts to not fill-up receiver’s buffer
- Receiver sending feedback is what allows for flow control

Congestion Control Review

- Prof. Levis’ second favorite lecture. *hint hint*
- Purpose: prevent over-subscription of data links
- States
  - Slow start
    - cwnd += 1 for every ACK received
    - Exponential growth
  - Congestion avoidance
    - Cwnd += 1/cwnd (Increase by 1 per RTT)
    - Linear growth

Congestion Control Example

- Nodes A, B are happily communicating over a dedicated link in congestion avoidance mode
- Heavy noise on the link disrupts traffic, causing a timeout to occur on a packet
- Go back to CWND = 1 in slow start mode.
- Switch to congestion avoidance once CWND reaches half the original CWND that got us in trouble

Additive Increase, Multiplicative Decrease

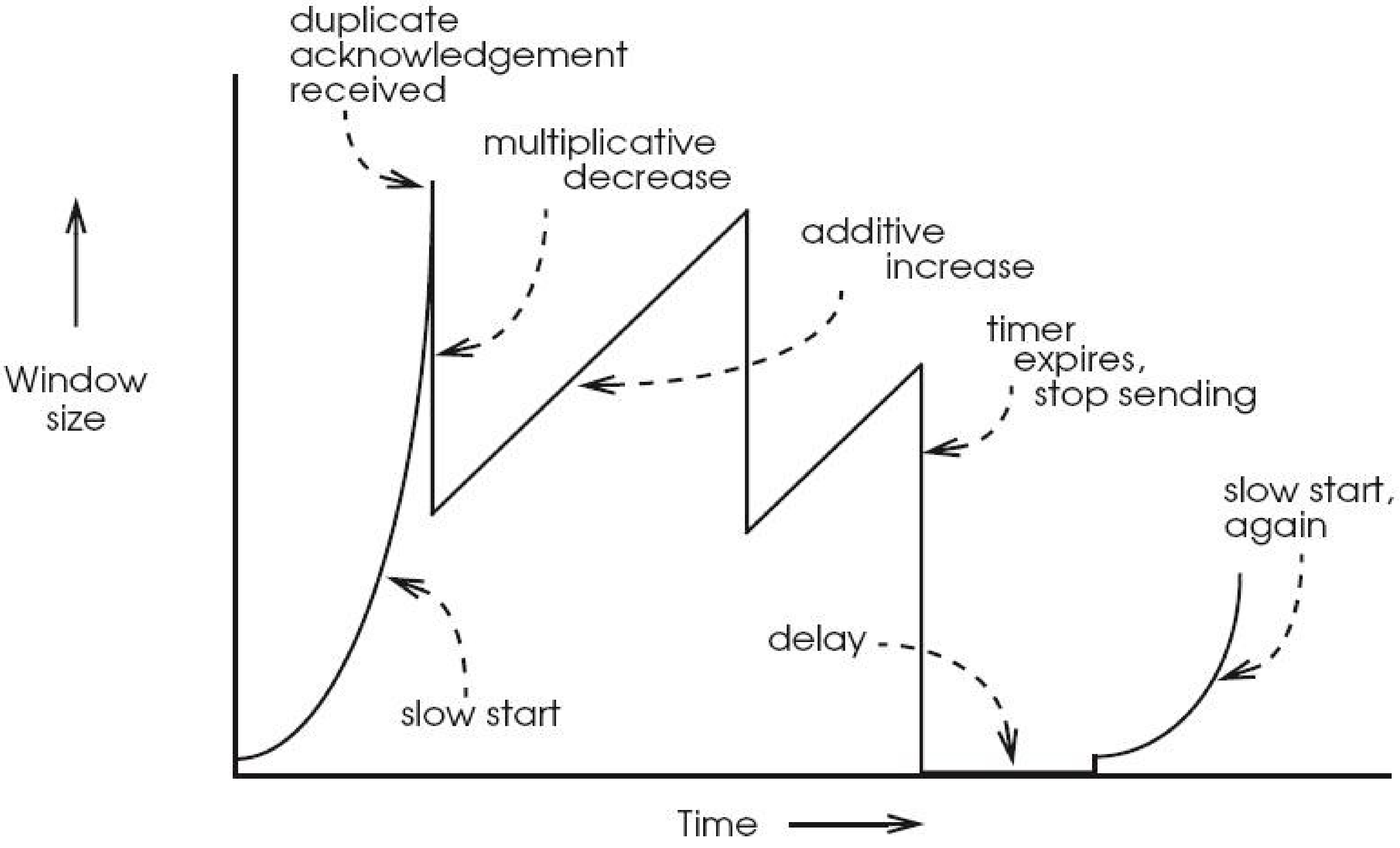


Figure from Brad Karp

Lab 2

Start Early! Good Luck!