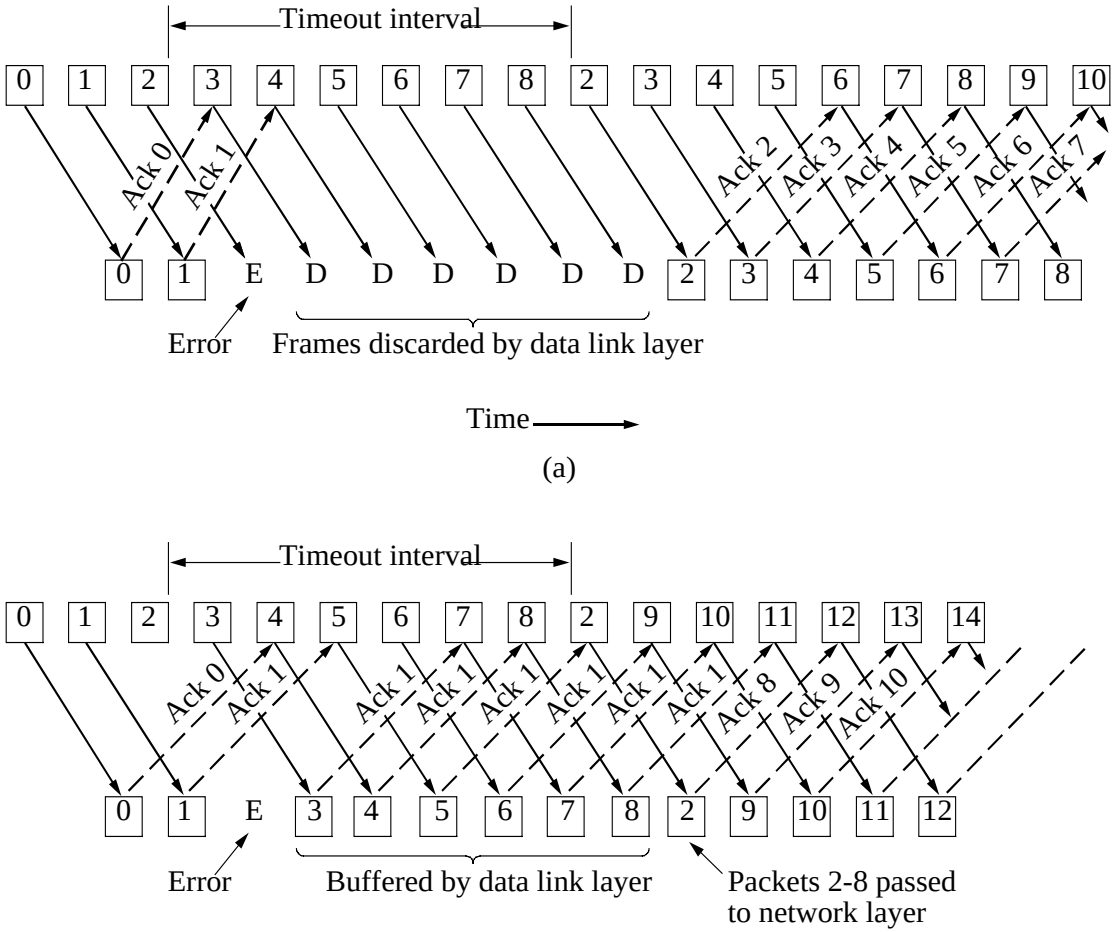
Announcements

- Program #2 regrades requests due by 11:00 today
- HW #2 was collected at the start of class
- Reading
  - Today: 6.3-6.4
  - Thursday: 3.1

Sliding Window Protocol

- Need to
  - have multiple outstanding packets
  - limit total number of outstanding packets
  - permit re-transmissions to occur
- Sliding Window
  - permit at most N outstanding packets
  - when packet is ACK'd advance window to first non-ACK'd pkt
- Retransmission
  - Go-back N
    - when a packet is lost, restart from that packet
    - provides in-order delivery, but wastes bandwidth
  - Selective Retransmission
    - use timeout to re-sent lost packet
    - use NACK as a **hint** that something was lost

Sliding Window Example



From: *Computer Networks*, 3rd Ed. by Andrew S. Tanenbaum, (c)1996 Prentice Hall.

Buffer Management

- Unreliable Network
  - sender must buffer all un-acked packets
  - receiver can buffer if space is available
    - if not, drop packet and wait to re-transmission
- Buffer Size
  - does one size fit all?
    - are TPDUs of uniform size?
  - might use a fixed size buffer smaller than max TPDU
    - requires support for multiple buffers per TPDU
- Possible to decouple buffer allocation from window
  - ACKs contain both buffer credits and ACKSs
- Buffer Copies
  - possible for each layer to copy the buffer, but this is slow
  - handoff pointers to data, but requires coordination between layers

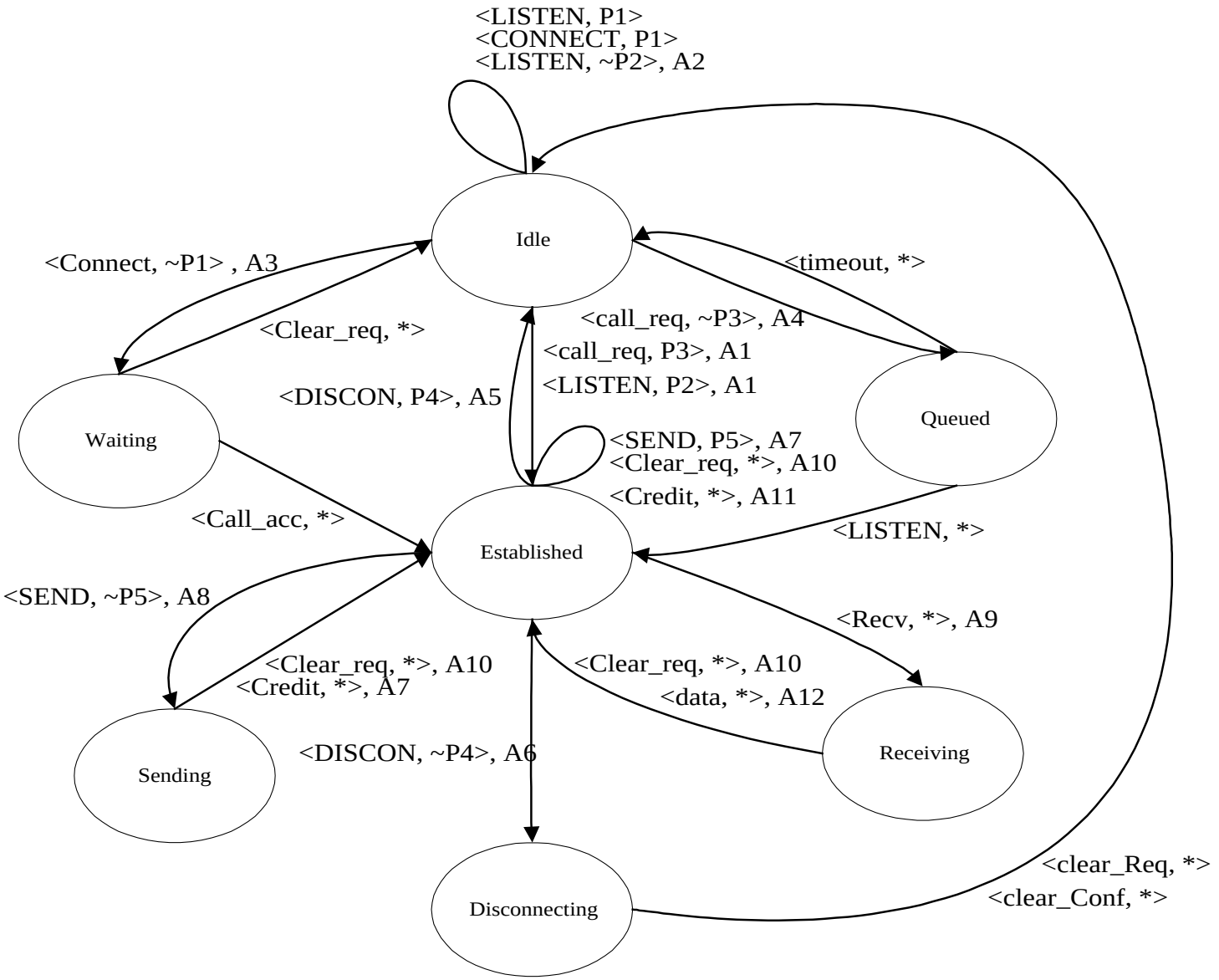
Multiplexing in the Transport Layer

- Upward multiplexing
  - putting multiple transport connections onto one network connection
  - used to accommodate pricing strategies that charge for connections
- Downward multiplexing
  - using several network connections per transport connection
  - permits use of multiple copies of network resources
    - if the network layer uses sliding windows
      - a high latency network may under utilize the link
      - multiple connections each get a window
    - per connection buffer allocation
      - get more buffers
    - round-robin scheduling
      - get a larger share of link bandwidth

Crash Recovery

- Router or Link Crashes
  - Data in transit can be lost.
  - End nodes have sufficient state to recover lost data.
  - Transport protocol can hide network failures from the application.
- Host Crashes
  - Transport level state will be lost at one end.
  - Does the transport layer have sufficient info to recover?, **No!.**
    - Information must flow down to network and up to transport user
      - ACKs go down, and data goes up.
      - It is not possible to make these two operations atomic.
    - lack of stable storage causes this problem
  - Result, higher up layer must deal with host crashes

Protocol State Machines



Predicates And State Transitions

Pred	Meaning
P1	Connection table full
P2	Call_req pending
P3	LISTEN Pending
P4	Clear_req Pending
P5	Credit Available

Act	Meaning
A1	Send Call_acc
A2	Wait for Call_req
A3	Send Call_req
A4	Start Timer
A5	Send Clear_conf
A6	Send Clear_req
A7	Send message
A8	Wait for credit
A9	Send Credit
A10	Set Clr_req_recv flag
A11	Record credit
A12	Accept message