Announcements

- Reading
  - Chapter 6 (6.4 & 6.6)
- Midterm #1
  - Re-grades were returned
    - Changes ranged from –7 to +10

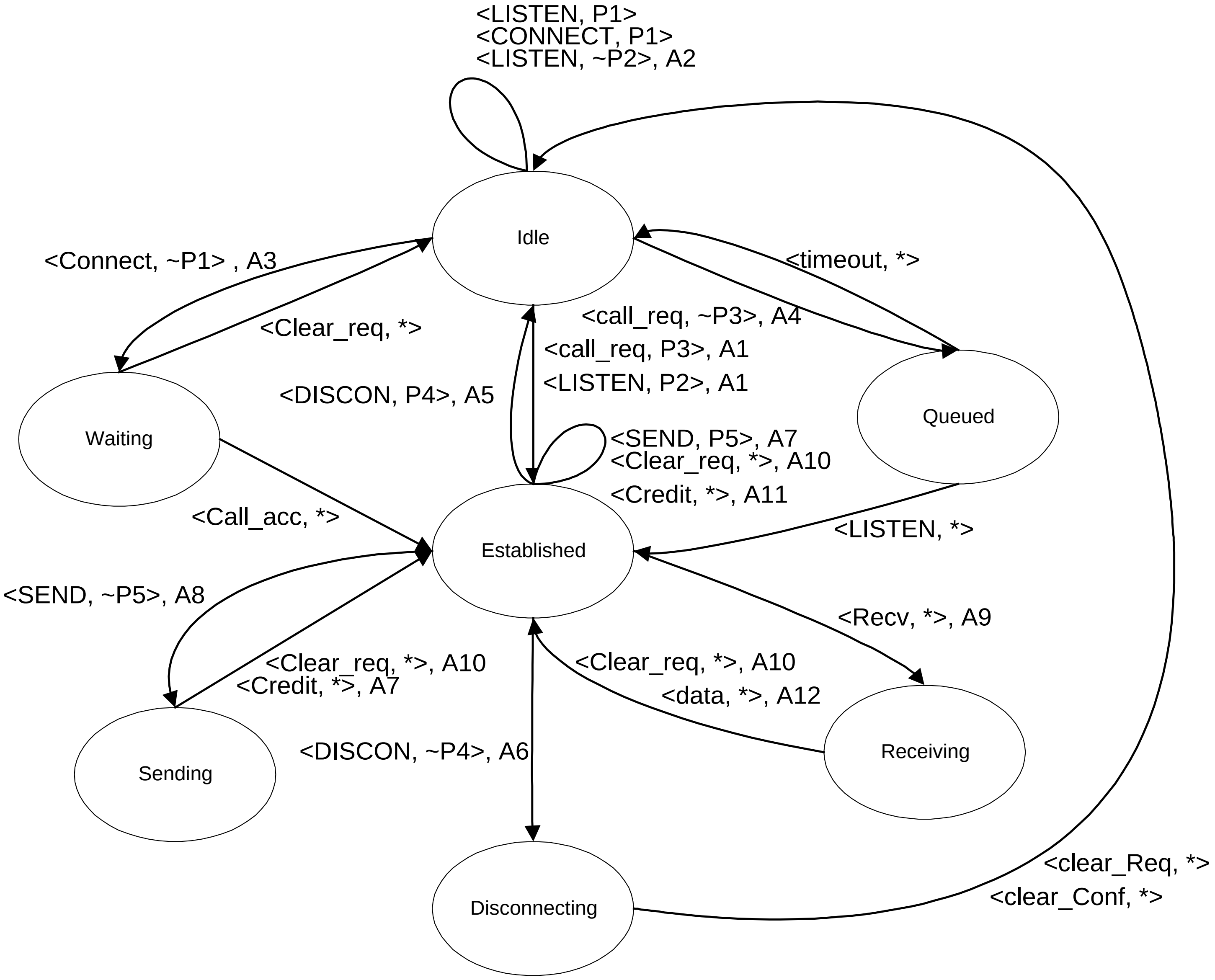
Multiplexing in the Transport Layer

- 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

TCP Timer Management

- Problem: How to pick timeout value?
  - need to estimate round-trip latency
  - need low variance in round trip latency
- Solution: dynamic estimates of RTT
  - $RTT = \alpha RTT + (1 - \alpha) M$

    M time of an ACK

    $\alpha = 7/8$
  - Need to pick retransmission time
    - old policy, use Timeout = RTT β , with $\beta = 2$
    - estimate standard deviation of RTT using mean deviation

      $D = \alpha D + (1 - \alpha) | RTT - M |$

      Timeout = $RTT + 4 * D$
  - How to update RTT on retransmission's
    - double Timeout on a retransmission

Other TCP Timers

- Persistence Timer
  - Prevents deadlock due to dropped window packets
    - This is a problem if the window is set to 0
- Keepalive Timer
  - Prevents half dead connections
  - may consume bandwidth
  - may kill live connections when net hiccups
- TIMED Wait
  - prevents re-use of a connection before max packet life is over
  - set to twice max packet lifetime