**Lab 6**

**Q1 In slow start, a sender doubles its window size every RTT if all sent packets were acknowledged**

1. **True**
2. False

**Q2 In steady state, a sender increases its window size by one packet for each acknowledgement**

1. True
2. **False**

**Q3 A sender that underestimates the round-trip time of a connection may unnecessarily induce a TCP timeout**

1. **True**
2. False

**Q4 After detecting packet loss through a timeout, TCP halves its window size as a response to the path congestion**

1. **True**
2. False

**Q5**

**A Triple duplicate ack**

**Q6**

**No,**  **it could be due to reordering due to queuing or asymmetric paths.**

**Q7**

**D. timeout**

**Q8**

**No, congestion could cause RTT > RTO**

**Q9**

**Less**

**Q10**

**This “slow-start” period quickly discovers the maximum acceptable throughput that the path supports – otherwise, AI (additive increase) could take too long (each a full RTT).**

**Q11**

**(a) How much time has progressed by point B?**

**1 RTT (TCP handshake) + 3-4 RTT in slow-start (1, 2, 4, (8) MSS) = 4 or 5 RTT = 400 or 500 ms**

(**b) How much time has progressed between points C and D?**

**4 MSS to 16 MSS = 12 periods of RTT = 1.2 s**

**(c) How much time has progressed between points E and F?**

**First: slow start to 8K window size**

**(1->2, 2->4, 4->8 MSS), then AI from 8 to 10 MSS window size (8->9->(10) MSS. 5 or 6 RTT = 500 or 600 ms.**

**Q12**

**C. 400ms**

Answer is actually 500 ms

**Q13**

**C. 1200ms**

**Q14**

**B. 600ms**

**Q15**

**Changing cross-traffic by other concurrent senders across same routers**