694684

Q1

Assumption: This is a Tcp zero based graph

(a) At event 13, the decrease in window is due to 3 duplicate acks that the ~~reciever~~ sender received.

(b) No, because it might be possible that the packet is only delayed. It should be noted that even the timeout set for the packet hasn't been completed. However, the packet under discussion will necessarily be retransmitted.

(c) Since the window changed to zero, this is a packet timout event.

(d) Since timeout has occurred, there is a higher probability that the packet has been lost and will be retransmitted. However, it may be posibble that it the packet was only delayed. Again it will necessarily be retransmitted

Rough Work

(c) ~~D~~ B is more Likely. Since, in a highly Light ~~n~~ network, timeout will be ~~rare~~ but with multiple packets in flight, one of them might be only slightly delayed will Lead to event B. However, D is not Likely for the aforesaid reason.

(d) If ~~a~~ A were Linear, the growth will be very slow even for Light networks. For example, if network bandwidth allowed speed of 256 MSS without any congestion, Linear will be at 256 after 256 transmissions while A with exponential growth it will get there ~~at~~ after ~~2^3 to~~ 8 transmissions.

10

Rough Work

C94684

Q2

1- data recieved:
If data is recieved, send ~~increament~~ Length data with seq = sendbase and increament the ~~sendbaseNextseq~~ NextSeqNum by the Length of data.

2- timeout:
If a timout occurs, send oldest packet that hasn't been acknowledged for yet.

3- Ack recieved:
If y > sendbase, then the sender is acknowedging for unacked bytes so we set sendbase = y. (6)

In addition to it, if timer was not running in (1), timer is started, in (3) if there are data packets for which ack hasn't arrived, we restart timer after seting sendbase = y.

Rough Work

Q3
(a) 72 - 61 = 11
(b) 61

4

Q4 ~~RTT=30~~
Estimated RTT = 30
dev RTT = 3 ms

EstimatedRTT = 0.~~875~~ × 30
+ 0.125 × 28

EstimatedRTT = 29.75
devRTT = 0.75(3) + 0.25(2)
~~Timeout~~ devRTT = 2.75

6

Timeout = EstimateRTT + 4(devRTT)
= 29.75 + 4(2.75)

Timout = 40.75

assuming SampleRTT = 28,
$\alpha = \frac{1}{8}$, $\beta = 0.25$

Q5

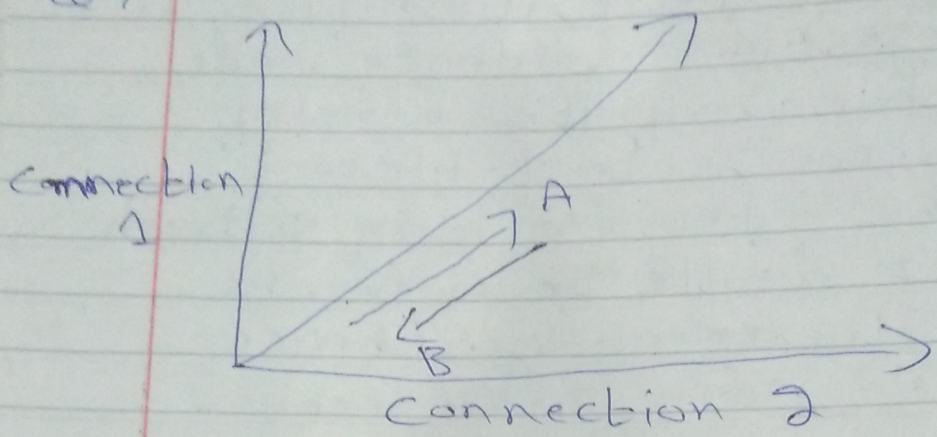
When Tcp congestion control algorithm reached a threshold value (ssthresh) while in slow start mode, it shifts to congestion avoidence mode, ssthresh is half the value of cwnd when the last timeout occurs, thus after reaching ssthresh we can't double the window, we instead increment by 1 (hence called congestion avoidance)

Yes, it can. Every time a timeout occurs, the algo shifts to slow-start irrespective of the current state.

Q6

Ans No. Why would it? Udp doesn't care about timeout or packet loss or congestion control. So, it ~~requires~~ has no use of RTT. It simply sends whatever is being sent to it.

07.



This can be best explained if there are only two connections in the network. In this case if both start at the same time, both will increamentaly increase their window size, as shown in arrow A. Uptill, a point reaches where ~~the~~ packet losses start and tcp would ~~ke~~ decrease ssthresh to de-congest the network. Then, both will go down in the graph till the network is de-congested and then the same process may repeat.

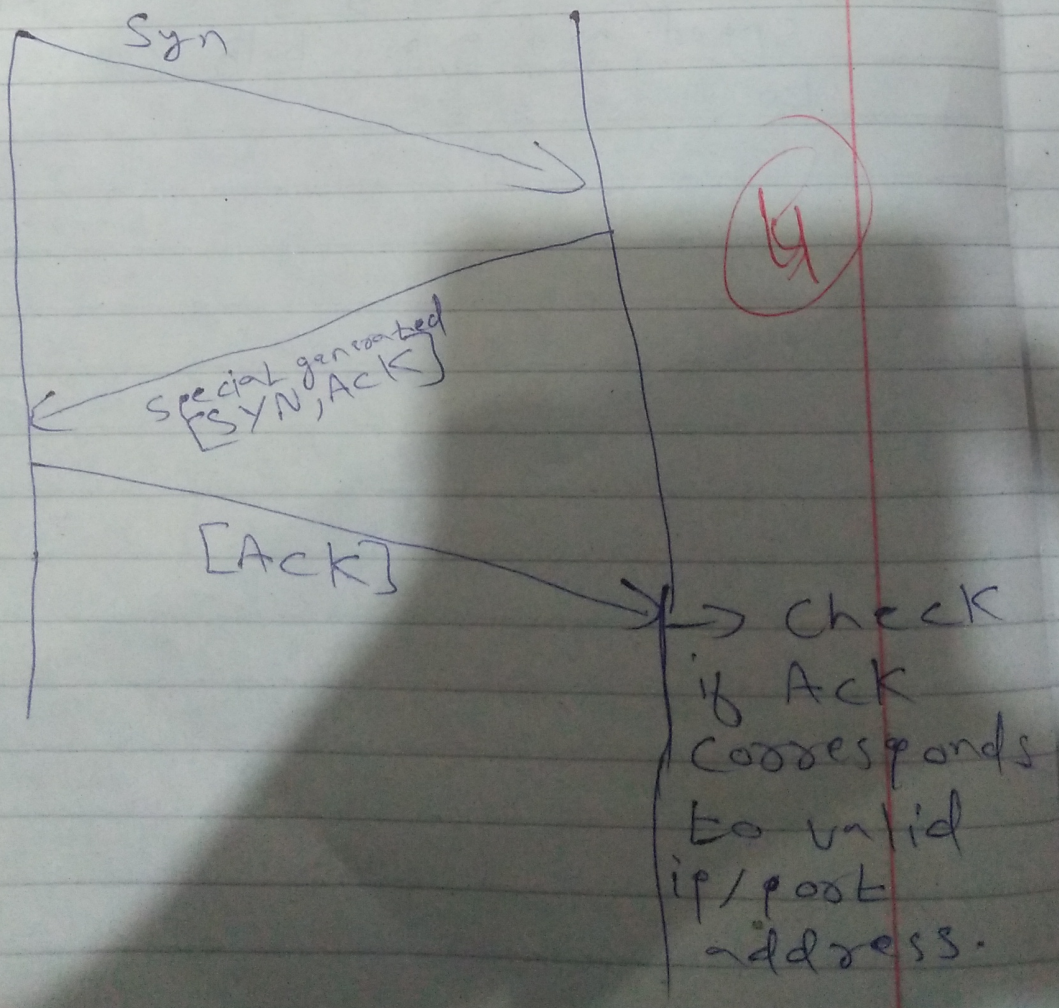
~~All tcp~~ This is because the slow start algorithm allow a client to keep increasing speed until packet loss occurs and then it halves the value to ~~est~~ ensure ~~estab~~ stabilization. So, both clients will eventually converge to similar speeds.

Even if both do not start at same time, say 1 starts before 2. If 1 keeps increasing it's size, at one point ~~it~~ loss will occur and it's speed will be halved, during that time 2 can get upto speed and again both converge to same rate.

Rough Work

Ans

SYN cookies are special Syn sequence numbers that are generated from the server by using the source ip address and source ports. A number is generated ~~using~~ which is a function of the source port or ip and then ~~this number~~ this is stored using a hash function. Once the ack is recieved, the ack number is passed to the hash function to see if it is from a valid client ip/port



694684

This ensures that if a connection is accepted, it is accepted from a valid client. for a Attack, the ack will not correspond to valid ip.