3170103240-张佳瑶-作业六

1

> In both parts of Fig. 6-6, there is a comment that the value of SERVERPORT must be the same in both client and server. Why is this so important?

If the client sends a packet to SERVERPORT and the server is not listening to that port, the packet will not be delivered to the server.

2

> Imagine that a two-way handshake rather than a three-way handshake were used to set up connections. In other words, the third message was not required. Are deadlocks now possible? Give an example or show that none exist.

Deadlocks are possible. For example, a packet arrives at A out of the blue, and A acknowledges it. The acknowledgement gets lost, but A is now open while B knows nothing at all about what has happened. Now the same thing happens to B, and both are open, but expecting different sequence numbers. Timeouts have to be introduced to avoid the deadlocks.

3

> Why does UDP exist? Would it not have been enough to just let user processes send raw IP packets?

No. IP packets contain IP addresses, whichs specify a destination machine. Once such a packet arrived, UDP packets contain a destination to support the newwork handler to know which process to give it to.

4

> A client sends a 128-byte request to a server located 100 km away over a 1-gigabit optical fiber. What is the efficiency of the line during the remote procedure call?

Send a packet needs 128*8/1G = 1.024us. Transport the packet needs 100km/200000 = 50us, so the round needs 100us. The efficiency of the line is about 1.024/100 = 1%.

5

> Datagram fragmentation and reassembly are handled by IP and are invisible to TCP. Does this mean that TCP does not have to worry about data arriving in the wrong order?

No. Although datagram is intact but may arrives in the wrong order. So TCP needs to reassumable the parts of a message properly.

6

> The maximum payload of a TCP segment is 65,495 bytes. Why was such a strange number chosen?

The entire TCP segment must fit in the 65515-byte payload field of an IP packet. Since the TCP header is a minimum of 20 bytes, only 65495 bytes are left for TCP data.

7

> If the TCP round-trip time, RTT, is currently 30 msec and the following acknowledgements come in after 26, 32, and 24 msec, respectively, what is the new RTT estimate using the Jacobson algorithm? Use α=0.9.

RTT = 30 msec

RTT1 = 0.9 * 30 +(1-0.9) * 26 = 29.6

RTT2 = 0.9 * 29.6 +(1-0.9) * 32 = 29.84

RTT3 = 0.9 * 29.84+(1-0.9) * 24 = 29.256

the new RTT is 29.256msec

8

> To get around the problem of sequence numbers wrapping around while old packets still exist, one could use 64-bit sequence numbers. However, theoretically, an optical fiber can run at 75 Tbps. What maximum packet lifetime is required to make sure that future 75-Tbps networks do not have wrap around problems even with 64-bit sequence numbers? Assume that each byte has its own sequence number, as TCP does.

The size of the sequence space is 2^{64} bytes. A 75-Tbps transmitter uses up sequence space at a rate of 9.375 e12 sequence number per second. It takes 2 million seconds to wrap around. It will take over 3 weeks to wrap around, even at 75 Tbps. A maximum packet lifetime of less than 3 weeks will prevent the problem. In short, going to 64 bits is likely to work for quite a while.

9

> Consider that only a single TCP (Reno) connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver's receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion

> avoidance phase, that is, ignore slow start.
>
> – **a.** What is the maximum window size (in segments) that this TCP connection can achieve?
>
> – **b.** What is the average window size (in segments) and average throughput (in bps) of this TCP connection?
>
> – **c.** How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

a.

MMS * Window size / RTT = link capacity

1500 * 8 * W / 0.15= 10Mbps

W = 125

the maximum window size: 125.

b.

average window size = (W+W/2)/2 = 0.75W

average throughput = window size / RTT = 0.75W * 1500 * 8 /0.15 = 7.5Mbps.

c.

T = (125/2)*0.15 = 9.375s