**Lab 5**

**Q1 TCP sequence numbers**

A TCP sender is just about to send a segment of size 100 bytes with sequence number 1234 and ack number 436 in the TCP header. What is the highest sequence number up to (and including) which this sender has received all bytes from the receiver? **C is the answer**

A. 1233

B. 436

**C. 435**

D. 1334

E. 536

**Q2 TCP sequence numbers**

A TCP sender is just about to send a segment of size 100 bytes with sequence number 1234 and ack number 436 in the TCP header. Is it possible that the receiver has received byte number 1335? **No**

1. Yes

**2. No**

**Q3 TCP timeout**

A TCP sender maintains a SmoothedRTT of 100ms. Suppose the next SampleRTT is 108ms. Which of the following is true of the sender?

1. Will increase SmoothedRTT but leave the timeout unchanged

2. Will increase timeout

3. Whether it increases SmoothedRTT depends on the deviation.

**4. Whether it increases the timeout depends on the deviation**

5. Will chomp on fries left over from the rdt question earlier

**Q4 TCP timeout**

A TCP sender maintains a SmoothedRTT of 100ms and DevRTT of 8ms. Suppose the next SampleRTT is 108ms. What is the new value of the timeout in milliseconds? (Numerical question)

RTO = SmoothedRTT + 4 \* DevRTT

SmoothedRTT = 100ms

DevRTT = 8ms

SampleRTT = 108ms

RTO = 100ms + 4 \* 8ms

RTO = 100ms + 32ms

**RTO = 132ms**

**Q5 TCP header fields v Which is the purpose of the receive window field in a TCP header?**

A. Reliability

B. In-order delivery

**C. Flow control**

D. Congestion control

E. Pipelining

**Q6 TCP Roughly how much time does it take for both the TCP sender and receiver to establish connection state since the connect() call?**

A. RTT

B. 1.5RTT

**C. 2RTT**

D. 3RTT

**Q7 TCP uses cumulative ACKs like Go-back-N, but does not retransmit the entire window of outstanding packets upon a timeout. What mechanism lets TCP get away with this?**

**A. Per-byte sequence and ack numbers**

B. Triple duplicate ACKs

C. Receive window-based flow control

D. Using a better timeout estimation method

E. Ketchup (for the fries)

**Q8 A sender that underestimates the round-trip time of a connection may unnecessarily induce a TCP timeout**

1: True

**2: False**

**Q9 Which of the following services use TCP?**

1. DHCP
2. **SMTP**
3. **HTTP**
4. TFTP
5. **FTP**

**Q10**

**100,000 / 125,000 = 0.8**

**Q11**

The router has a buffer size of 100,000 bytes.

The client is sending 1,000 byte packets at a rate of 150 packets per second.

router is receiving data at a rate of 1,000 bytes \* 150 packets/second = 150,000 bytes/second.

Time to fill buffer = Buffer size / Data rate = 100,000 bytes / 150,000 bytes/second = 2/3 seconds.

2/3 seconds, which is slightly less than 1 second. So the closest to that is

**A.2 seconds**

However, it's important to note that the buffer will not reach 100,000 bytes if the packets are only 1,000 bytes each. The router's buffer will continually receive and transmit packets, so it's possible that no packet will be dropped in this scenario as long as the rate of incoming packets does not exceed the router's transmission rate.

**Q12**

Link Capacity: capacity of 100 packets per second

Latency: latency 100 ms, which is equivalent to 0.1 seconds.

Window Size: 4 packets.

The sending rate is limited by the window size and the round-trip time (RTT), which is twice the latency:

Sending rate = Window size / (2 \* RTT)

Sending rate = 4 packets / (2 \* 0.1 seconds

Sending rate = 4 packets / 0.2 second

**Sending rate = 20 packets per second**

1. **20 pkts/s**