

1. Assuming that a TCP endpoint have received everything up to sequence number 2499 successfully, show the acknowledgments in response to the following segments, arriving in that order:

- Seg(2500, 500)
 - ACK(3000)
- Seg(4000, 200)
 - ACK(3000)
- Seg(4200, 300)
 - ACK(3000)
- Seg(3000, 1000)
 - ACK(4500)
- Seg(4500, 100, with FIN=1)
 - ACK(4601)

where Seg(X, Y) denotes a segment with sequence number X and Y bytes of data

2. Using the equations

$$RTT = RTT \times 0.9 + NEW_RTT_SAMPLE \times 0.1,$$

$$Timeout = 2 \times RTT.$$

Starting from $RTT = 100\text{ms}$, calculate the Timeout value after processing 3 RTT samples: 120ms, 50ms, and 400ms. Notice that there probably is a developing congestion situation as implied in the 3rd sample. Your Timeout result will be used as the timeout interval for the next leaving packet. If the round-trip time of that packet is again 400ms, will it suffer retransmission?

$$RTT = 100 \text{ ms}$$

$$RTT_sample = 120\text{ms}$$

$$RTT = 100 * 0.9 + 120 * 0.1 \\ = 102 \text{ ms}$$

$$Timeout = 2 * RTT = 204 \text{ ms}$$

$$RTT_sample = 50 \text{ ms}$$

$$RTT = 102 * 0.9 + 50 * 0.1 \\ = 96.8 \text{ ms}$$

$$Timeout = 2 * 96.8 = 193.6 \text{ ms}$$

$$RTT_sample = 400 \text{ ms}$$

$$RTT = 96.8 * 0.9 + 400 * 0.1 \\ = 127.12 \text{ ms}$$

$$Timeout = 2 * 127.12 = 254.24 \text{ ms}$$

If the round-trip time of that packet is again 400ms which is greater than the timeout, It will suffer retransmission.

3. Calculate Timeout again with the same set of RTT samples but using refined RTT estimation equations:

$$\text{DIFF} = \text{SAMPLE} - \text{RTT}$$

$$\text{RTT} = \text{RTT} + \text{DIFF}/8$$

$$\text{DEV} = \text{DEV} + (|\text{DIFF}| - \text{DEV})/4$$

$$\text{Timeout} = \text{RTT} + \text{DEV} \times 4$$

Initial RTT=100ms and initial DEV=10. Your Timeout result will be used as the timeout interval for the next leaving packet. If the round-trip time of that packet is again 400ms, will it suffer retransmission ?

RTT Samples: 120ms, 50ms, 400ms

Initial RTT = 100ms, initial DEV = 10

i) $\text{DIFF} = 120 - 100 = 20$

$$\text{RTT} = 100 + 20 / 8 = 102.5$$

$$\text{DEV} = 10 + (20 - 10) / 4 = 12.5$$

$$\text{Timeout} = 102.5 + 12.5 * 4 = 152.5$$

ii) $\text{DIFF} = 50 - 102.5 = -52.5$

$$\text{RTT} = 102.5 - 52.5/8 = 95.9$$

$$\text{DEV} = 12.5 + (52.5 - 12.5) / 4 = 22.5$$

$$\text{Timeout} = 95.9 + 4 * 22.5 = 185.9$$

iii) $\text{DIFF} = 400 - 95.9 = 304.1$

$$\text{RTT} = 95.9 + 304.1 / 8 = 133.925$$

$$\text{DEV} = 22.5 + (304.1 - 22.5) / 4 = 92.9$$

$$\text{Timeout} = 133.925 + 4 * 92.9 = 505.525$$

Even if the round-trip time of the packet is again 400ms, that is less than the timeout. So it will not suffer retransmission.

4. Calculate Timeout yet again using the refined RTT estimation equations and with RTT samples: 104ms, 99ms, 97ms. Initial RTT=100ms and initial DEV=10. Notice how confident you are this time to use a Timeout value close to RTT

RTT 104ms, 99ms, 97ms

Initial RTT = 100 ms, Initial DEV = 10

i) $\text{DIFF} = 104 - 100 = 4$

$$\text{RTT} = 100 + 4 / 8 = 100.5$$

$$\text{DEV} = 10 + (4 - 10) / 4 = 8.5$$

$$\text{Timeout} = 100.5 + 8.5 * 4 = 134.5$$

ii) $\text{DIFF} = 99 - 100.5 = -1.5$

$$\text{RTT} = 100.5 - 1.5 / 8 = 100.3$$

$$\text{DEV} = 8.5 + (1.5 - 8.5) / 4 = 6.75$$

$$\text{Timeout} = 100.3 + 4 * 6.75 = 127.3$$

iii) $\text{DIFF} = 97 - 100.3 = -3.3$

$$\text{RTT} = 100.3 - 3.3 / 8 = 99.8$$

$$DEV = 6.75 + (3.3 - 6.75) / 8 = 6.31$$

$$Timeout = 99.8 + 4 * 6.31 = 125.04$$

Since there is not much deviation in the sample, we can be sure that there is no congestion in the network. So we can be confident to use a timeout value close to RTT.

5. Assume that TCP implements an extension that allows window sizes much larger than 64 KB. Suppose that you are using this extended TCP over a 1 Gbps link with a latency of 50 ms to transfer a 10 MB file, and the TCP receive window is 1 MB. If TCP sends 1 KB packets (assuming no congestion and no lost packets):

(a) (10 pts) How many RTTs does it take until slow start opens the send window to 1 MB?

In slow start, the size of the window doubles every RTT. So,

At nth RTT = window size is 2^n KB.

Therefore, $2^n = 1 \text{ MB}$, $2^n = 2^{10} \text{ KB}$

It will take 10 RTTs until slow start opens the send window to 1 MB. (during 11th RTT, the window size will be 1 MB)

(b) (10 pts) How many RTTs does it take to send the file?

In first RTT, 1 KB is sent, and 2 KB in second RTT

So, for the first 10 RTTs, 1023 KB would have been sent. From 11th RTTs, the window size will be 1 MB (1024 KB). So after 19th RTT 10239 Kbytes have been sent.

So, It would take 20 RTTs to send the file.

(c) (10 pts) If the time to send the file is given by the number of required RTTs multiplied by the link latency, what is the effective throughput for the transfer? What percentage of the link bandwidth is utilized?

Time to send the file = 20 RTTs * 2 * 50ms = 2 sec

Effective throughput is given by:

$$10 \text{ MB} / 2\text{sec} = 5 \text{ MB} / \text{sec} = 40 \text{ Mbs}$$

Percentage of bandwidth utilized is:

$$40 \text{ Mbs} / 1 \text{ Gbs} = 4 \%$$

6. During linear increase, TCP computes an increment to the congestion window as Increment = MSS × (MSS/Congestion Window)

Explain why computing this increment each time an ACK arrives may not result in the correct increment. Give a more precise definition for this increment. (Hint: A given ACK can acknowledge more or less than one MSS's worth of data.)

This formula is accurate if each new ACK acknowledges one new MSS-sized segment. However, an ACK can acknowledge more or less than one MSS's worth of data.

Let $N = \text{CongestionWindow} / \text{MSS}$, the window size measured in segments. The goal of the original formula was so that after N segments arrived, the net increment would be MSS, making the increment for one MSS-sized segment MSS / N . If instead we receive an ACK acknowledging more or less than one MSS, we should expand the formula by

Increment = AmountACKed / N

$$= (\text{AmountACKed} * \text{MSS}) / \text{CongestionWindow}$$

7. Suppose a TCP connection has a window size of eight segments and an RTT of 800 ms, the sender sends segments at a regular rate of one every 100 ms, and the receiver sends ACKs

back at the same rate without delay. A segment is lost, and the loss is detected by the fast retransmit algorithm on the receipt of the third duplicate ACK. At the point when the ACK of the retransmitted segment finally arrives, how much total time has the sender lost (compared to lossless transmission) if

(a) (10 pts) The sender waits for the ACK from the retransmitted lost packet before sliding the window forward again?

The sender has lost 1100 ms. 300 ms to detect third duplicate ACK and then 800 ms to receive the ACK of the retransmitted segment.

(b) (10 pts) The sender uses the continued arrival of each duplicate ACK (after three duplicate ACKs) as an indication it may slide the window forward one segment?

The sender lose $1100 - 400 = 700$ ms. Even though the time to resume transmission again is 1100 ms but it had four extra chances to transmit .