TDTS04 - Group A - Assignment 3.
Alexander Jonsson (alejo720)
Jonas Malmlöf (jonma993)

Assignment 3:

Task A:

Now, based on questions 1-12, please write two paragraphs explaining and discussing your observations from the above practice questions. One paragraph should describe and discuss the connections at a high level. The second paragraph should discuss the impact of RTT estimates, packet losses, and interpreted packet loss events. Note that your answer may benefit from explaining and/or referring to some of your observations from the practice questions explicitly. Note that, similar to previous assignments, you are expected to convince us that you understand these aspects of TCP.

High-level:

The first step when uploading a file is to open a connection. This is done by an initial 3-way handshake between the client (sending to port 80) and the server (sending to port 1161). Afterwards, the client starts sending the file. This is done by separating the file in TCP segments. If a segment has been received by the server, the server sends an "ACK"-response acknowledging the client that the segment has been successfully received. If the client doesn't get an "ACK"-response, the client sends the same segment again so that the whole file is guaranteed to arrive.

Low-level:

After the client has sent a TCP-segment and is waiting for the "ACK"-response, the client starts a timer called "Retransmission Timeout"(RTO). If the "ACK"-response hasn't arrived until the end of the timer it retransmits the segment. This RTO can impact the RTT. Each time the same segment is sent, it doubles the RTO. This creates an exponential increase in the SampleRTT (essentially the time it takes to receive the ACK-response after sending). This then increases the EstimatedRTT which explains why our EstimatedRTT values are not only increasing, but also exponentially.

Task B:

Please answer and discuss the following three questions:

13. Use the Time-Sequence-Graph (Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the server (Figure 2a and Figure 2b). For each of the two traces, can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? If you can, explain how. If not, explain why not. To better identify these phases, you may need to find the number of unacknowledged packets (or bytes) at different times and plot the unacknowledged packets (y-axis) as a function of time (x-axis). Note that the number of unacknowledged packets at different times can be found by comparing the number of packets that have been sent with the number of packets that have been acknowledged. After plotting the number of unacknowledged packets versus time, comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

Answer:
In the older trace the slow start is in effect for 7 packets.
Between the first and second segment there is a delay of ~0.015s, then there is a delay of ~0.013s between the second and third. ~0.0006s between the third and fourth. then there's a bigger delay again between fourth and fifth of ~0.023s. then a short, then a big... after that there's a huge leap of ~0.18s where the client waits for ACKs before starting to send again. After this, it's clear skies. In conclusion, the amount of segments sent before waiting for ACK-responses is constant

In the newer trace, if you look at the first three packets, you see that the second is sent after a delay of ~0.004s and the third is sent after a delay of 0.001s. The size of the packets are not the cause of these delays, as the delay would actually be smaller if it were (the first packet is easily half the size of the second or third). In conclusion, the amount of segments sent before waiting for ACK-responses is increasing.

The ideal way that TCP would work - in a world where congestion wouldn't have to be accounted for - the server would never have to intentionally drop packets or tell the client to slow down. thus any delays in TCP segments sent would be entirely from a packet being lost in transit (and as such would appear with far less predictable frequency).

14. Explain the relationship between (i) the congestion window, (ii) the receiver advertised window, (iii) the number of unacknowledged bytes, and (iv) the effective window at the sender.

Answer:

1) The Congestion Window limits how many bytes that can be unacknowledged at a time.

2) The Receiver Advertised Window limits how many bytes the server can receive and buffer at a time.

3) The unacknowledged bytes are either packets in flight, dropped packets at the receiver or packets that were lost on their way there.

4) A function between the amount of bytes in flight, the Congestion Window, and the socket buffer that the sender is sending through on their end.

The relationship between these four is that the top speed that can be attained will always be limited by the slowest of these factors.

15. Is it generally possible to find the congestion window size (i.e. cwnd) and how it changes with time, from the captured trace files? If so, please explain how. If not, please explain when and when not. Motivate your answer and give examples. Your answer may also benefit from trying to describe and discuss your answer in the context of the two prior questions, for example.

Answer:

Yes, you can look at the graph from the old and new traces. In the newer trace's graph the (packets sent/second) first has an exponential growth, followed by a linear growth after 16 packets have been sent. In the older trace, there is no such linear growth at all. meaning the server has no congestion control.

Task C:

Please carefully answer and discuss questions 16-18 as outlined in this section.

16. What is the throughput of each of the connections in bps (bits per second)? What is the total bandwidth of the host on which the clients are running? Discuss the TCP fairness for this case.

Answer:
1 (165095720/521)*8 = 2 535 059 bps
2 (165842766/521)*8 = 2 546 530 bps
3 (165458792/514)*8 = 2 575 234 bps
4 (163235772/512)*8 = 2 550 559 bps

The total bandwidth of the host in this case, is how many bits the host sent over the connections in this segment of time.
2535059+2546530+2575234+2550559 = 10 207 382 bps

There is a slight difference in the results for each user. This could be because of dropped packages or differences in how their different clients work.
Because the different users are on the same network, accessing the same server, it is difficult to see if TCP Fairness even comes into play. the parameters are already as fair as they can get.

17. What is the throughput of each of the connections in bps (bits per second)? What is the total bandwidth of the host on which the clients are running? Discuss the TCP fairness for this case.

Answer:

		bps	RTT
1	(261319130/90)*8	= 23 228 367	13
2	(175995832/90)*8	= 15 644 074	35
3	(151894552/90)*8	= 13 501 738	68
4	(140388568/90)*8	= 12 478 984	73
5	(108610702/90)*8	= 9 654 285	49
6	(70644690/90)*8	= 6 279 528	33
7	(65744938/90)*8	= 5 843 995	135
8	(43212876/90)*8	= 3 841 145	326
9	(39222524/90)*8	= 3 486 447	322

Host's total bandwidth: 93 958 563 bps.

As for the fairness: The connection with the smallest RTT gets the most data during the allotted time, and the two connections with the highest RTT get the least amount of data.

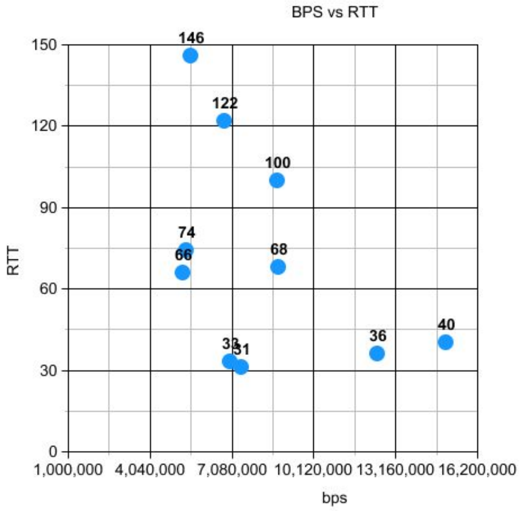
There are outliers though, 5 and 6, get less data than 2, even though they have a smaller RTT. and 8 gets more than 9 despite having a higher RTT.

On the other hand, the difference between 6 and 7 is not as big as the difference in RTT would indicate it should be. All this suggest that the host is deciding that some clients are more equal than others. Whether this is ‘fair’ is entirely up to to the crew running the server and those who connect to it to decide.

18. Discuss the TCP fairness for this case. For all of these questions you must take a closer look at the relationships between the characteristics of the different connections and discuss your findings in the context of the different experiments. You are expected to show that you understand the concept of TCP fairness and how the different scenarios may impact the throughput relationships that you observe and those that you may expect in general. To help the discussion you may for example want to create a scatter plot that show the estimated round trip time (RTT) and throughput against each other (for the different connections).

Answer:

		bps	RTT
1	(108851134/58)*8	= 15 013 950	40
2	(90435681/58)*8	= 12 473 887	36
3	(57971584/53)*8	= 8 750 428	100
4	(32000012/29)*8	= 8 827 590	68
5	(32557334/35)*8	= 7 441 676	31
6	(27199361/31)*8	= 7 019 190	33
7	(26329578/31)*8	= 6 794 730	122
8	(38834490/56)*8	= 5 547 784	146
9	(23571761/35)*8	= 5 387 831	74
10	(36252962/55)*8	= 5 273 158	66



Host's total bandwidth: 82 530 224 bps.
This leaves a strange picture, doesn't it? How come 2 and 6 get such vastly different amounts of data when their RTT (that is to say, electronic time-distance to the host) is almost the same?
The difference between 10 and 8 is also a good example: time is similar, bps is within 200k of each other, but the RTT of 8 is over twice that of 10.

This means that the server doesn't give equal amount of bandwidth to all connections regardless of their RTT. Whether this is fair is all in the eye of the beholder. It is impossible to tell whether the host prioritizes different connections based on what company they are from or based on their internet connection or their RTT.