Haorui Zhang

10/30/2021

CS 455

1. Graphical user interface, application, table

   Description automatically generated
   1. IP address: Source address: 192.169.1.102 Port Number: 1161 Destination Port number: 80
   2. IP of gaia.cs.umass.edu: 128.119.245.12 

Receiving port: 80

Sending port: 1161



* 1. My client IP: 192.168.1.135

TCP port: 62689

* 1. The sequence number counts the bytes of data into byte stream. It’s 0 in this case. The SYN flag is set to 1Graphical user interface, application

     Description automatically generatedGraphical user interface, application

     Description automatically generated
  2. The sequence number is 0.

The value of the ACK in the SYNACK segment is 1. Gaia server determines this value by adding 1 to the SYN segment’s sequence number, which is 0. SYNACK is determined by ACK and SYN flags. Both are set to 1.Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

* 1. The sequence number is 1. Graphical user interface, text

     Description automatically generated
  2. From the first to the sixth segment, the sequence numbers are: 1→566→2026→3486→4946→6406.

|  |  |  |  |
| --- | --- | --- | --- |
| # Segment | Sent Time | Received Time | RTT |
| 1 | 0.026477 | 0.053937 | 0.02746 |
| 2 | 0.041737 | 0.077294 | 0.035557 |
| 3 | 0.054026 | 0.124085 | 0.070059 |
| 4 | 0.054690 | 0.169118 | 0.114428 |
| 5 | 0.077405 | 0.217299 | 0.139894 |
| 6 | 0.078157 | 0.267802 | 0.189645 |

EstimatedRTT = 0.875\*EstimatedRTT+0.125\*SampleRTT

Estimated RTT for 1st segment: 0.02746

Estimated RTT for 2nd segment: 0.875\*0.02746+0.125\*0.035557 = 0.02847

Estimated RTT for 3rd segment: 0.875\*0.02847+0.125\*0.070059=0.0327775

Estimated RTT for 4th segment: 0.875\*0.0327775+0.125\*0.114428 = 0.0429838125

Estimated RTT for 5th segment: 0.875\*0.0429838125+0.125\*0.139894 = 0.05509

Estimated RTT for 6th segment: 0.875\*0.05509 + 0.125\*0.189645 = 0.071916

* 1. Length:

|  |  |
| --- | --- |
| # Segment | Length (byte) |
| 1 | 565 |
| 2 | 1460 |
| 3 | 1460 |
| 4 | 1460 |
| 5 | 1460 |
| 6 | 1460 |

Text

Description automatically generated with low confidence

* 1. Min buffer space (window size) is 5840 bytes. Since the window size increase to 62780 bytes eventually, the sender does not affected by the lack of receiver buffer space. Graphical user interface, text, application

     Description automatically generated
  2. No, based on the sequence numbers graph, the transmission is stable and show no change, which means there is no retransmission. Graphical user interface, application, table

     Description automatically generated
  3. We can calculate data size in an ACK by compare difference between this ACK seq number and next ACK sequence number.

|  |  |  |
| --- | --- | --- |
| # ACK | Cur sequence number | Data size |
| 1 | 566 | 566 |
| 2 | 2026 | 1460 |
| 3 | 3486 | 1460 |
| 4 | 4946 | 1460 |
| 5 | 6406 | 1460 |

…

We can identify cases where the receiver is ACKing every other received segment based on data size. If the data size is different from others except the first one, then the receiver is ACKing every other received segment.

Graphical user interface, text, application, table

Description automatically generated

* 1. The throughput of TCP is the amount of data being sent/received in a unit of time. To calculate the throughput, the total amount of data and the total time for the transmission need to be known. In this case, the total amount of data is determined by the difference between the final sequence number and the first sequence number, which is 164091-1 = 164090 bytes. The amount of time is determined by the difference between the first segment time and the final segment time, which is 5.461175-0.026477 = 5.434698s. The total throughput is 164090 bytes/5.434698s = 30193bytes/second
  2. The slowstart phases starts as beginning and ends where the second phases of segments transmission start. Starts at 0.0 sec and ends by 0.1242 secs.

Chart

Description automatically generatedSlow start phases

The congestion avoidance takes over slow start phase as slow start phase ends: Table

Description automatically generatedTable

Description automatically generatedCongestion avoidance phase.

Ideally TCP has the transmit as fast as possible as long as there is no congestion, and the window size after congestion is the half of the one before. But in this case, it’s much less than the half of the threshold. A picture containing table

Description automatically generated

The threshold is 52560 and the ideal window size should be 26280 but the actual one is only 17520

* 1. My graph:

Slow starts begins at 2.7s and ends at around 2.71s. After that the congestion phases start. Compare to ideal one, the window size is close to ideal case. 

Chart, line chart

Description automatically generated