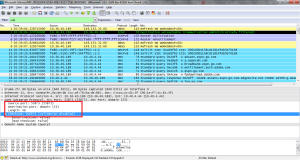
*Start capturing packets in Wireshark and then do something that will cause your host to send and receive several UDP packets. It’s also likely that just by doing nothing (except capturing packets via Wireshark) that some UDP packets sent by others will appear in your trace. In particular, the Simple Network Management Protocol (SNMP – chapter 9 in the text) sends SNMP messages inside of UDP, so it’s likely that you’ll find some SNMP messages (and therefore UDP packets) in your trace.*

*QUESTIONS:*

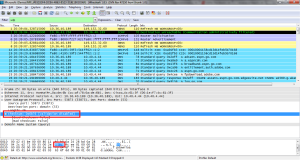
*1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. (You shouldn’t look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.*

* The header only contains 4 fields: the source port, destination port, length, and checksum.

[](https://maxwellsullivan.files.wordpress.com/2013/03/wireshark5-1.png)

*2. By consulting the displayed information in Wireshark’s packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.*

* Each of the UDP header fields is 2 bytes long

[](https://maxwellsullivan.files.wordpress.com/2013/03/5-2.png)

*3. The value in the Length field is the length of what? (You can consult the text for this answer). Verify your claim with your captured UDP packet.*

* The value in the length field, in the example below it is 46, is the sum of the 8 header bytes and the remaining data bytes encapsulated in the packet.

*4. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2. above)*

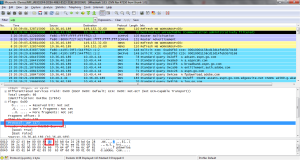
* The maximum number of bytes that can be in the payload is 2^16- the bytes already being used by the header field (8). Therefore the maximum payload is 65535-8= 65527 bytes.

*5. What is the largest possible source port number? (Hint: see the hint in 4.)*

* The largest possible source port number is 2^16 or 65535.

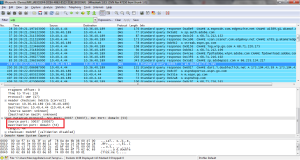
*6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you’ll need to look into the Protocol field of the IP datagram containing this UDP segment (see Figure 4.13 in the text, and the discussion of IP header fields).*

* The protocol number for UDP is 17 in decimal notation which in hexadecimal notation is 0x11.

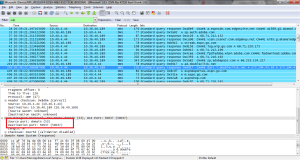
[](https://maxwellsullivan.files.wordpress.com/2013/03/5-4.png)

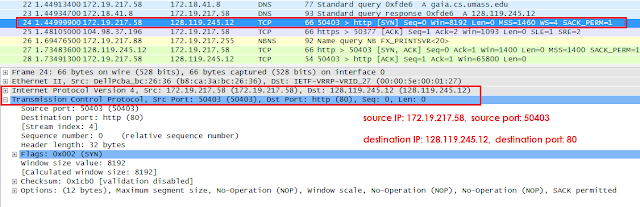
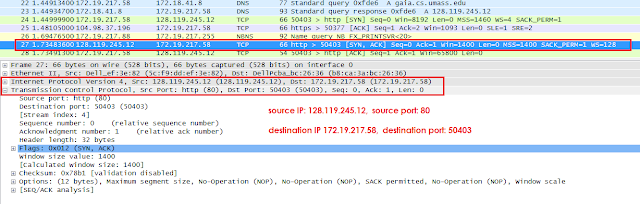
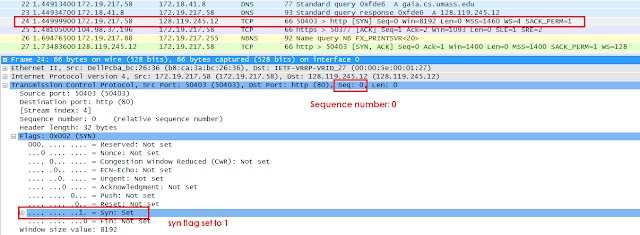
*7. Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets*

UDP Sent by my host

[](https://maxwellsullivan.files.wordpress.com/2013/03/5-5-2.png)

UDP Reply to Host

[](https://maxwellsullivan.files.wordpress.com/2013/03/5-5-3.png)

* The relationship between port numbers is that the source port on the send message is the destination port of the receive message. The destination port for the send message is also the source port for the receive message.
* 1.     What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows.
* Answer:
* The client IP address is *172.19.217.58*, TCP port number is *50403*
* *Screenshot*
* [](https://3.bp.blogspot.com/-BL4nPRX8R_0/VzP88qChsiI/AAAAAAAAG6c/CxT6BQj9QOYX8hnQeYLKLHTwR1JSPmddgCLcB/s1600/1.png)
* Figure 1: SYN packet
* 2.     What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?
* Answer:
* gaia.cs.umass.edu’s IP address is *128.119.245.12*, port number is *80*
* *Screenshot*
* [](https://2.bp.blogspot.com/-hBPnpd26EAo/VzP9DiB4lAI/AAAAAAAAG6g/HGoKgMelpPgr8LDGMEXYJcx8MDOIq6ncwCLcB/s1600/2.png)
* Figure 2: SYN ACK packet
* 4.     What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?
* Answer: The sequence number of the TCP SYN segment is 0 since it is used to imitate the TCP connection between the client computer and gaia.cs.umass.edu. According to the screenshot below, in the Flags section, the SYN flag is set to 1 which indicates that this segment is a SYN segment.
* *Screenshot*
* [](https://1.bp.blogspot.com/-8WBRojU0tcE/VzP9VXozX_I/AAAAAAAAG6k/MEn_YV-TpYcSThWWDxYcixgFpA0ZhmaCgCLcB/s1600/4.png)
* Figure 3: SYN seq num + flag
* 5.     What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?
* Answer: According to the screenshot below, the sequence number of the SYN\_ACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0.
* The value of the acknowledgement field in the SYN\_ACK segment is determined by the server gaia.cs.umass.edu. The server adds 1 to the initial sequence number of the SYN segment from the client computer.
* For this case, the initial sequence number of the SYN segment from the client computer is 0, thus the value of the acknowledgement field in the SYN\_ACK segment is 1
* . A segment will be identified as a SYN\_ACK segment if both SYN flag and ACKnowledgement flag in the segment are set to 1.

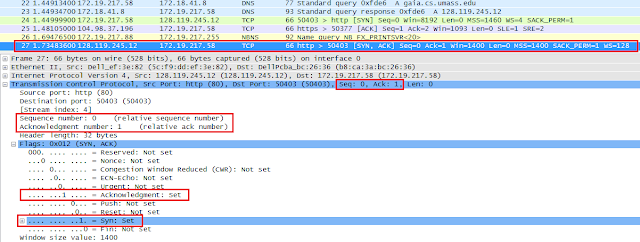
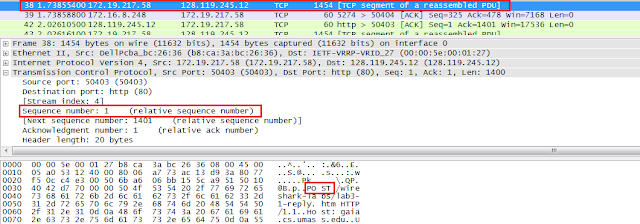
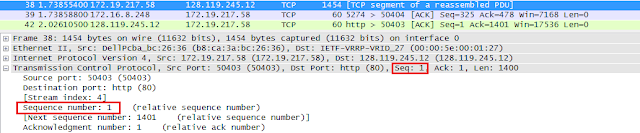
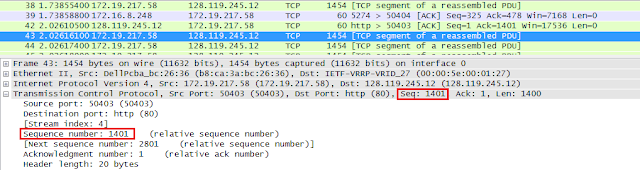
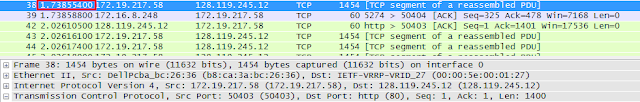
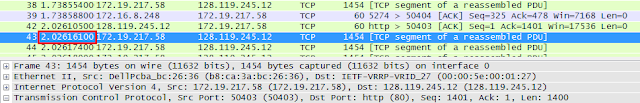
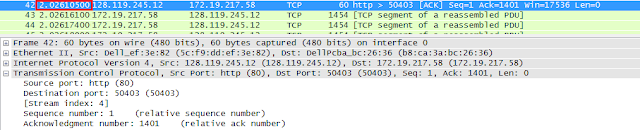
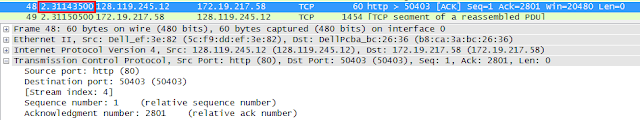
Sequence number of the SYNACK segment fromgaia.cs.umass.edu to the client computer in reply to the SYN has the value of 0 in this trace.

The value of the ACKnowledgement field in the SYNACK segmentis 1.

The value of the ACKnowledgement field in the SYNACK segment is determined by gaia.cs.umass.eduby adding 1 to the initial sequence number of SYN segment fromthe client computer (i.e.the sequence number of the SYN segmentinitiated bythe client computer is 0.).

The SYN flag and Acknowledgement flag in thesegmentare set to 1 and they indicate that this segment is a SYNACK segment

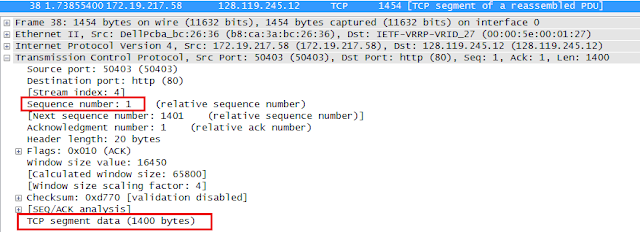
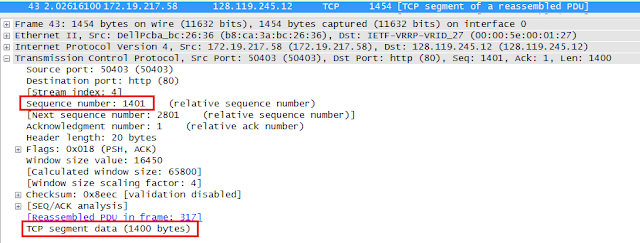
.

* *Screenshot*
* [](https://1.bp.blogspot.com/-Lmn-vbEE75Q/VzP9Y24jHxI/AAAAAAAAG6o/bQ4wh2LdhfI2NBNMKtBTl3Q-C6XtsokXgCLcB/s1600/5.png)
* 6.     What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.
* Answer: The sequence number of the TCP segment containing the HTTP Post command is 1.
* *Screenshot*
* [](https://4.bp.blogspot.com/-8fjQJ9STrCg/VzP9cblcI_I/AAAAAAAAG6s/CpCtkWanEdkxb03KrlL48ExD317bR3zCwCLcB/s1600/6.png)
* Figure 4: HTTP Post
* 7.     Consider the TCP connection.
* a.     What are the sequence numbers of the first six segments in the TCP connection?
* Answer: Sequence number for segment 1 is 1, sequence number for segment 2 is 1401.
* Screenshot:
* [](https://2.bp.blogspot.com/-Be4JV25rHyE/VzP9migoq4I/AAAAAAAAG60/dQAKKpaar4YfyMkwHr8oxB7nXn4EmWARwCLcB/s1600/1.png)
* [](https://4.bp.blogspot.com/-oYvl0hJSGMk/VzP9mjZ-t7I/AAAAAAAAG64/YSQ9g6B7SII6AQuBXvso_4Hw_jDkc8LYwCLcB/s1600/2.png)
* b.     At what time was each segment sent?
* Answer: 1.738554 s for segment 1 and 2.026161 s for segment 2.
* Screenshot:
* [](https://2.bp.blogspot.com/-lydNlzV0L_w/VzP9vO-bmlI/AAAAAAAAG7A/WnF2FJPlFUYm1XGHJDZcam7vqNlvsSVQgCLcB/s1600/3.png)
* [](https://2.bp.blogspot.com/-77GUds5ArE8/VzP9vAJzzLI/AAAAAAAAG7E/uK1c0rb3JDcyD-ny1vvaiEo5UIioLQ1iQCLcB/s1600/4.png)
* c.     When was the ACK for each segment received?
* Answer: ACK for segment 1 was received at 2.026105 s and ACK for segment 2 is received at 2.311435 s.
* Screenshot:
* [](https://2.bp.blogspot.com/-J5B-ksvWjb8/VzP96M1FCLI/AAAAAAAAG7I/cRX55-7HcPs72e30tdhUCHe2fZnFYX4bQCLcB/s1600/1.png)
* [](https://1.bp.blogspot.com/-bsEUvCMgkoE/VzP96Mt69fI/AAAAAAAAG7M/x2emUHM9yagfGnpuONvwJVVOy-o_3euuACLcB/s1600/2.png)
* d.     Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments?
* Answer:
* RTT for segment 1 is 0.287551 seconds, RTT for segment 2 is 0.285274 seconds, RTT for segment 3 is 0.285261 seconds, RTT for segment 4 is 0.285369 seconds, RTT for segment 5 is 0.000176 seconds, RTT for segment 6 is 0.286297 seconds.

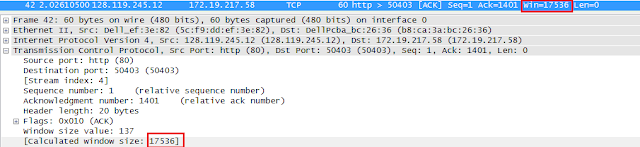
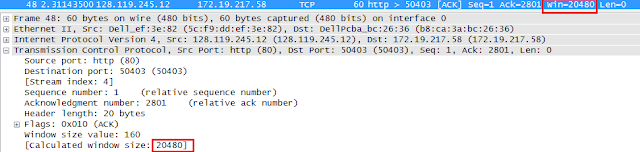
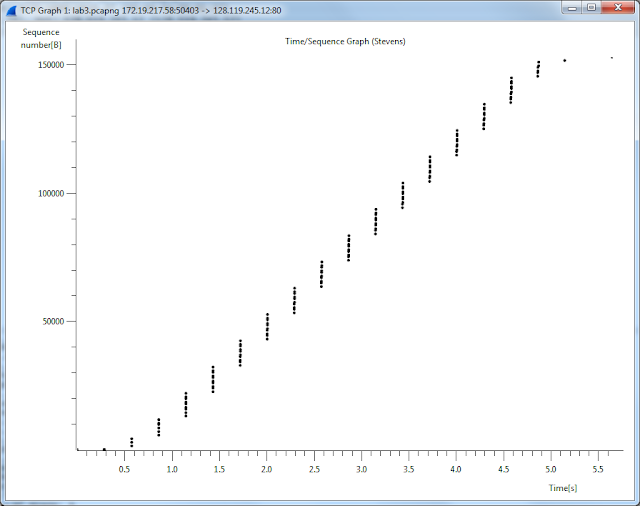
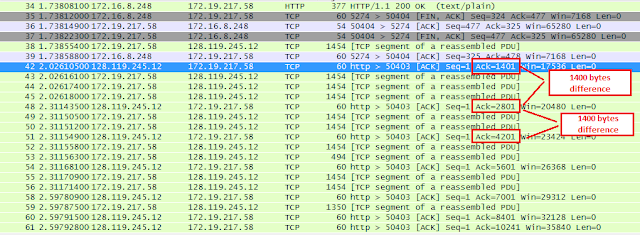
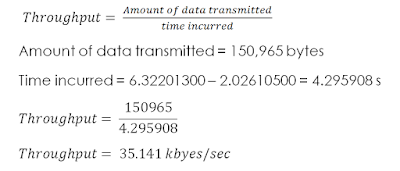
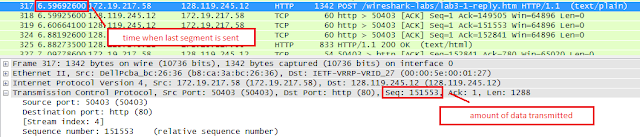
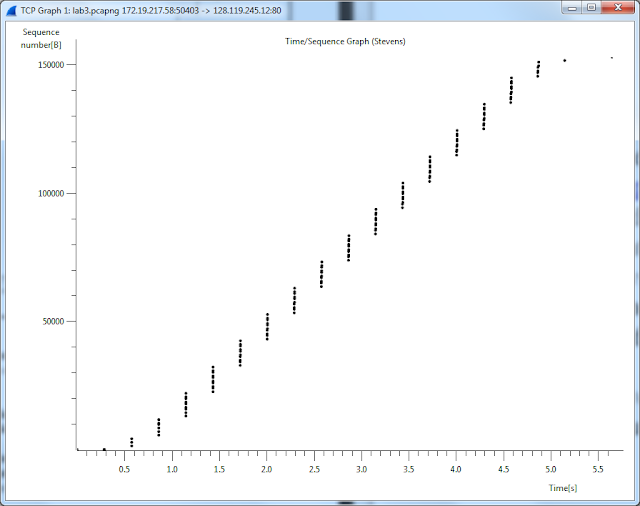
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Segment | Packet Number | Sequence Number | Time Sent(s) | Time ACK received(s) | RTT(s) |
| 1 | 38 | 1 | 1.738554 | 2.026105 | 0.287551 |
| 2 | 43 | 1401 | 2.026161 | 2.311435 | 0.285274 |
| 3 | 44 | 2801 | 2.026174 | 2.311435 | 0.285261 |
| 4 | 45 | 4201 | 2.026180 | 2.311549 | 0.285369 |
| 5 | 49 | 5601 | 2.311505 | 2.311681 | 0.000176 |
| 6 | 50 | 7001 | 2.311512 | 2.597809 | 0.286297 |

* 8.     What is the length of each of the first six TCP segments?
* Answer:
* The length of each of the first 6 TCP segments is 1400 bytes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Segment | Packet Number | Sequence Number | Length(bytes) | Time Sent(s) | Time ACK received(s) | RTT(s) |
| 1 | 38 | 1 | 1400 | 1.738554 | 2.026105 | 0.287551 |
| 2 | 43 | 1401 | 1400 | 2.026161 | 2.311435 | 0.285274 |
| 3 | 44 | 2801 | 1400 | 2.026174 | 2.311435 | 0.285261 |
| 4 | 45 | 4201 | 1400 | 2.026180 | 2.311549 | 0.285369 |
| 5 | 49 | 5601 | 1400 | 2.311505 | 2.311681 | 0.000176 |
| 6 | 50 | 7001 | 1400 | 2.311512 | 2.597809 | 0.286297 |

* Screenshot:
* [](https://2.bp.blogspot.com/-bI3VHj_DSPw/VzP-EerOw5I/AAAAAAAAG7U/xGosc7KrOzAfC9Bd3Sujv7z32qyngsybwCLcB/s1600/1.png)
* Figure 5: Segment 1 Length
* [](https://3.bp.blogspot.com/-3kkL2J7-xi8/VzP-IOl_6dI/AAAAAAAAG7Y/39fNoT9gA84QD6xgnICJHfXtYmhIlebaACLcB/s1600/2.png)
* Figure 6: Segment 2 Length
* 9.     What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
* Answer:
* The minimum amount of available buffer space advertised at the received is 17536 bytes.

|  |  |
| --- | --- |
| Segment | Available Buffer Space Advertised At The Received |
| 1 | 17536 |
| 2 | 20480 |
| 3 | 23424 |
| 4 | 26368 |
| 5 | 29312 |
| 6 | 32128 |

* Screenshot:
* [](https://3.bp.blogspot.com/-Je49zhCaP9U/VzP-PzjWWlI/AAAAAAAAG7g/goupdtNeL78X2zHatFTIOa2AazwAEdwTwCLcB/s1600/1.png)
* Figure 7: Segment 1 Buffer
* [](https://1.bp.blogspot.com/-gCbu8i7UrkM/VzP-SlWzrFI/AAAAAAAAG7k/f2M6FF27agwRFyFRiGOFScIgCDsag7iWwCLcB/s1600/2.png)
* Figure 8: Segment 2 Buffer
* 10.  Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
* Answer:
* No there is no retransmitted segments in the trace file. This can be explained by packets with same sequence number at different time is not found.
* Screenshot:
* [](https://2.bp.blogspot.com/-vIJSXMNSpmI/VzP-ZfLBXxI/AAAAAAAAG7s/kt3zHnZnlEo8VlA-j3nBZqw8FJMSD-dfwCLcB/s1600/3.png)
* 11.  How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).
* Answer: According to the screenshot below, we can see that the ACK numbers increase in the sequence of 1401, 2801, 4201, and so on. The ACK numbers increases by 1400 each time, indicating that the receiver is acknowledging 1400 bytes.
* Screenshot:
* [](https://2.bp.blogspot.com/-HyUyf_NIDrM/VzP-f2-grZI/AAAAAAAAG70/TPCorn6hsUcMGtdXg78hjCZlGuVW2JLEgCLcB/s1600/4.png)
* 12.  What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.
* Answer:
* [](https://1.bp.blogspot.com/-D1p_02DKF5E/VzP_KHBMJ7I/AAAAAAAAG8M/VVkRlv1NK2EA2ik69KLVggeL20jwRo9cQCLcB/s1600/5.PNG)
* Screenshot:
* [https://1.bp.blogspot.com/-UhMJ1U2d8L0/VzP_YDYSVmI/AAAAAAAAG8U/fIz3w48qAJAg1MZJZvCEk2bXcy1AhCyFQCLcB/s640/1.png](https://1.bp.blogspot.com/-UhMJ1U2d8L0/VzP_YDYSVmI/AAAAAAAAG8U/fIz3w48qAJAg1MZJZvCEk2bXcy1AhCyFQCLcB/s1600/1.png)
* Figure 9: Time Send First Segment
* [](https://4.bp.blogspot.com/-7N8DQY8wl78/VzP_cIUZWXI/AAAAAAAAG8Y/ZrXbHrbZiycTeNtYMBCOD4kMnIIhQeb-gCLcB/s1600/2.png)
* Figure 10: Time Send Last Segment
* 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 gaia.cs.umass.edu server. Can you identify where TCP’s slow start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behaviour of TCP that we’ve studied in the text.
* Answer:
* By observing the plot, we can see that the slow-start phase only lasts for first 1-1.5 second. Afterwards, it seems that the TCP session is always in congestion avoidance state. In this case, we do not observe the expected linear increase behaviour, i.e. the TCP transmit window does not grow linearly during this phase. In fact, it appears that the sender transmits packets in batches of 6. This does not seem to be caused by flow control since the receiver advertised window is significantly larger than 5 packets. The reason for this behaviour might be due to the fact that the HTTP server has enforced a rate-limit of some sort.
* Screenshot:
* [](https://3.bp.blogspot.com/-ihQ2XCAKCzo/VzP_iMQI06I/AAAAAAAAG8g/xJC73yLdufgCInlhhga6aj3Sn5_CFJLCgCLcB/s1600/3.png)