CMPSCI 453
HW9, Wireshark Lab 5

Tony Gao

March 13, 2018

1 Lab

1. The source IP is 192.168.1.102. The source port is 1161.

```
1 0.000000       192.168.1.102        128.119.245.12        TCP      62     1161 -> 80
[SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Ethernet II, Src: PremaxPe_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:
73)
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
```

2. The IP of gaia.cs.umass.edu is 128.119.245.12. The port is 80.

3. The IP of the client in my user-generated trace is 10.0.0.157, the port is 56564.

```
314 14.494856      10.0.0.157            128.119.245.12        TCP      66     56564 -> 80
[ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=287860679 TSecr=1425403752
Frame 314: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Apple_10:41:06 (80:e6:50:10:41:06), Dst: ArrisGro_0f:07:81 (10:56:11:0f:07:81)
Internet Protocol Version 4, Src: 10.0.0.157, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 56564, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
```

4. The sequence number of the SYN segment is 0. The TCP header flag field bitmask 0x002 denotes a SYN segment.

```
Flags: 0x002 (SYN)
000. .... .... = Reserved: Not set
...0 .... .... = Nonce: Not set
.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
.... ...0 .... = Acknowledgment: Not set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
.... .... ..1. = Syn: Set
.... .... ...0 = Fin: Not set
[TCP Flags: ··········S·]
```

5. The sequence number of the SYNACK is 0. The ACK number is 1. The server determines this by taking the sequence number of the syn segment plus 1. The bitmask 0x010 | 0x002 = 0x012 denotes syn + ack in the TCP header flags.

```
Flags: 0x012 (SYN, ACK)
000. .... .... = Reserved: Not set
...0 .... .... = Nonce: Not set
.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
.... ...1 .... = Acknowledgment: Set
.... .... 0... = Push: Not set
.... .... .0.. = Reset: Not set
.... .... ..1. = Syn: Set
.... .... ...0 = Fin: Not set
[TCP Flags: ·······A··S·]
```

6. The sequence number of the segment containing the POST is 1

   ```
   4 0.026477 192.168.1.102 128.119.245.12 TCP 619 1161 -> 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP s
   %s pE]!@fw
   P
   4tPDpPOST /ethereal-labs/lab3-1-reply.htm HTTP/1.1
   Host: gaia.cs.umass.edu
   User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20030208 Netscape/7.02
   Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/
   Accept-Language: en-us, en;q=0.50
   Accept-Encoding: gzip, deflate, compress;q=0.9
   Accept-Charset: ISO-8859-1, utf-8;q=0.66, *;q=0.66
   Keep-Alive: 300
   Connection: keep-alive
   Referer: http://gaia.cs.umass.edu/ethereal-labs/lab3-1.htm
   ```

7. First six segments starting from HTTP Post as first segment:

   - Sequence 1, time sent: 0.026477000s, ack 566 received 0.053937000s. RTT=27.46ms, len 565
   - Sequence 566 sent: 0.041737000s, ack 2026 received: 0.077294000s. RTT=35.557ms, len 1460
   - Sequence 2026 sent: 0.054026000s, ack 3486 received: 0.124085000s RTT=70.059ms, len 1460
   - Sequence 3486 sent: 0.054690000s, ack 4946 received: 0.169118000s RTT=114.428ms, len 1460
   - Sequence 4946 sent: 0.077405000s, ack 6406 received: 0.217299000s RTT=139.894ms, len 1460
   - Sequence 6406 sent: 0.078157000s, ack 7866 received: 0.267802000s RTT=189.645ms, len 1460

   - $EstimatedRTT_0 = 27.46$ms
   - $EstimatedRTT_1 = .875 \cdot 27.46 + .125 \cdot 35.557 = 28.472125$ms
   - $EstimatedRTT_2 = .875 \cdot 28.472125 + .125 \cdot 70.059 = 33.670484375$ms
   - $EstimatedRTT_3 = .875 \cdot 33.670484375 + .125 \cdot 114.428 = 43.765173828125$ms
   - $EstimatedRTT_4 = .875 \cdot 43.765173828125 + .125 \cdot 139.894 = 55.781277099609375$ms
   - $EstimatedRTT_5 = .875 \cdot 55.781277099609375 + .125 \cdot 189.645 = 72.514242462158203125$ms

8. First six segment lengths:

   - Sequence 1 len 565
   - Sequence 566 len 1460
   - Sequence 2026 len 1460
   - Sequence 3486 len 1460

- Sequence 4946 len 1460
- Sequence 6406 len 1460

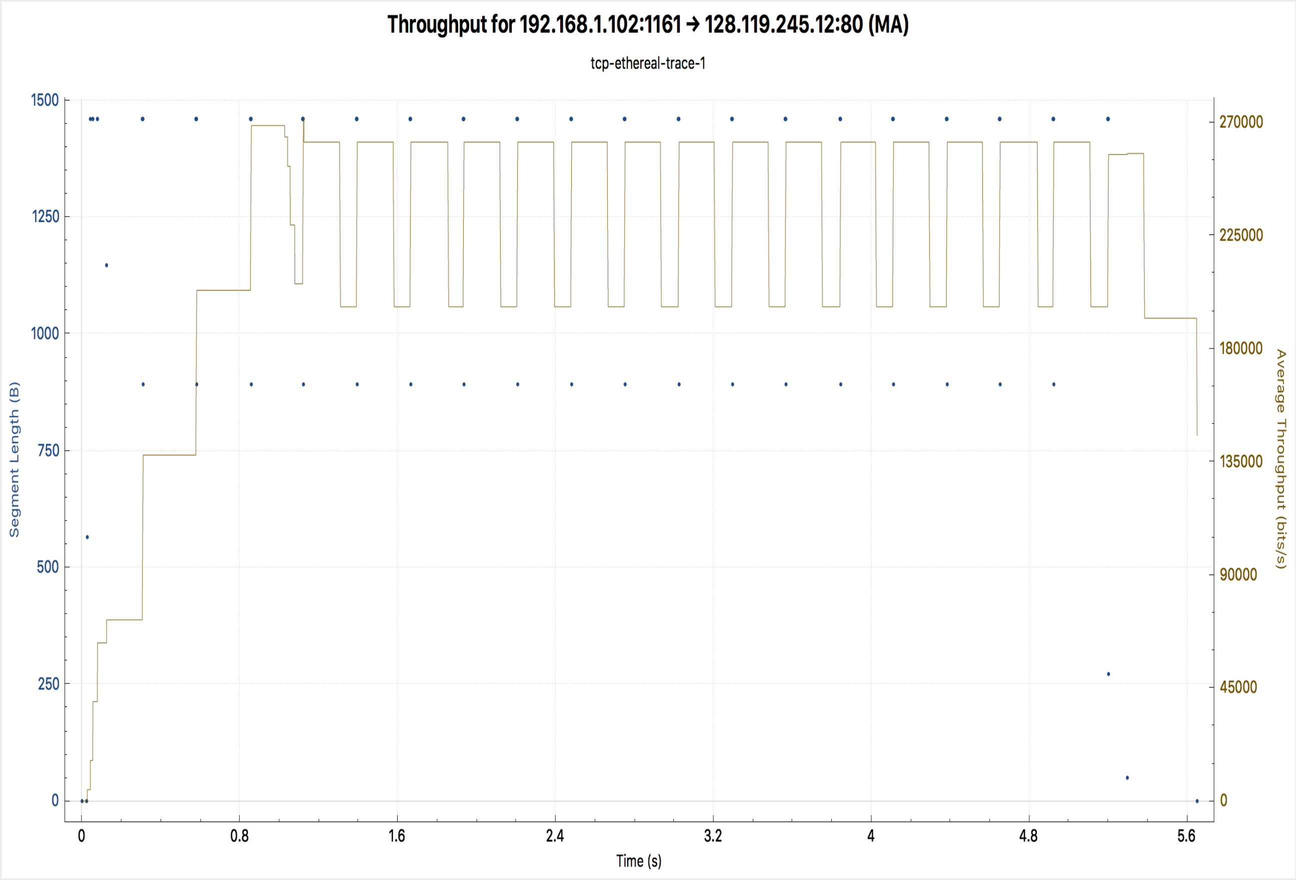
9. The minimum receiver buffer space is advertised in capture packet number 2. It is 5840B. This does not throttle the sender speed because the receiver window size grows faster than the total amount of unacked packets. Total unacked packets reaches a maximum of 6, but receiver window continues to grow far past 6 * 1460B. When the TCP connection is in Slow Start, the receiver window is approximately double the size of the total amount of unacked packets, which means the receiver still has space to buffer more packets, but the sender congestion window size does not allow for any more packets.

   ```
   Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 0, Ack: 1, Len: 0
   Source Port: 80
   Destination Port: 1161
   [Stream index: 0]
   [TCP Segment Len: 0]
   Sequence number: 0    (relative sequence number)
   Acknowledgment number: 1    (relative ack number)
   0111 .... = Header Length: 28 bytes (7)
   Flags: 0x012 (SYN, ACK)
   Window size value: 5840
   [Calculated window size: 5840]
   Checksum: 0x774d [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
   Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
   [SEQ/ACK analysis]
   ```

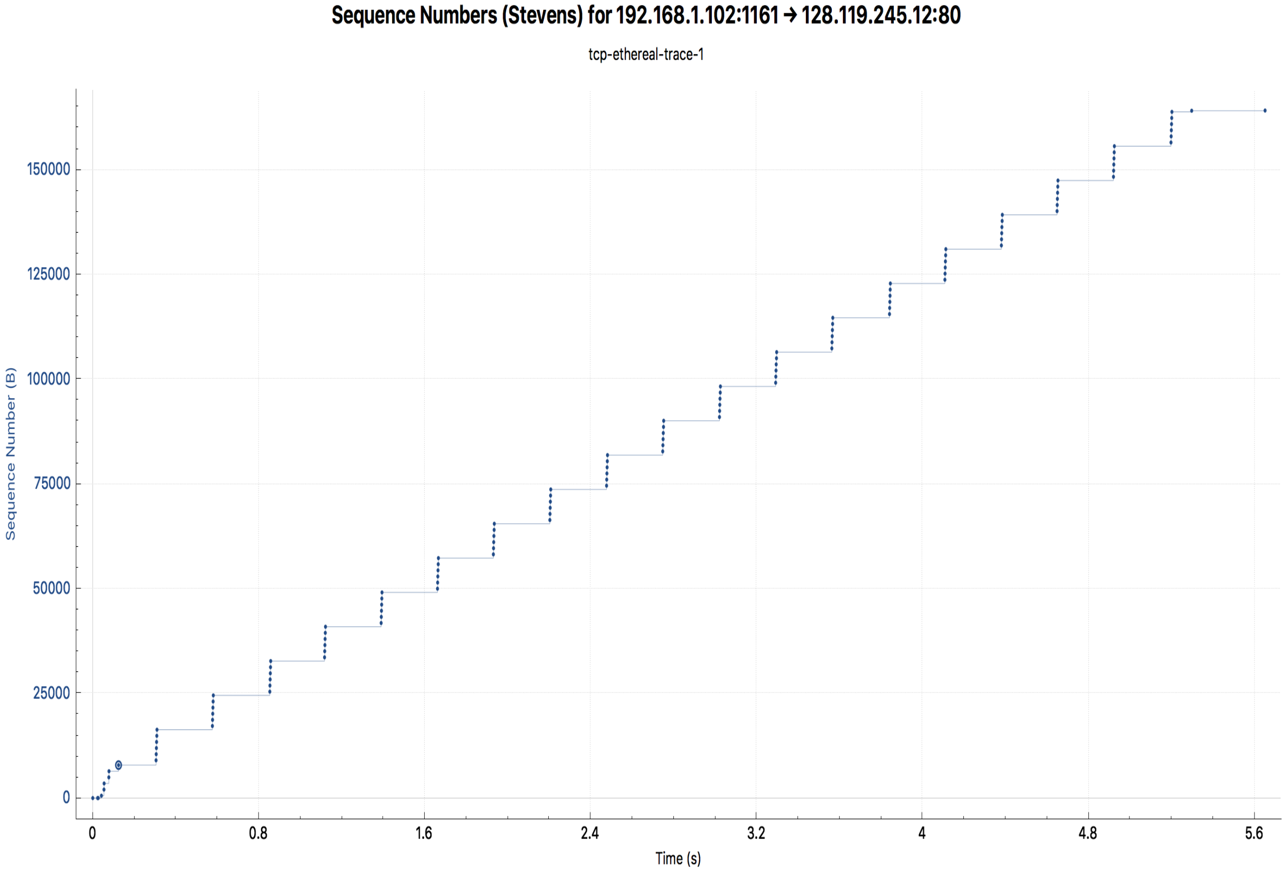
10. There are no retransmissions. This is observed by setting the display filter to ”tcp and (tcp.analysis.retransmission or tcp.analysis.fast_retransmission)” and seeing an empty packet display window.

11. Sequence 41781 (len 1460, ack num 43241) and 43241(len 1460, ack num 44701) were sent back to back, and the receiver only acks 44701, which is the ack number for sequence 43241. The receiver typically acknowledges 1460B in an ack.

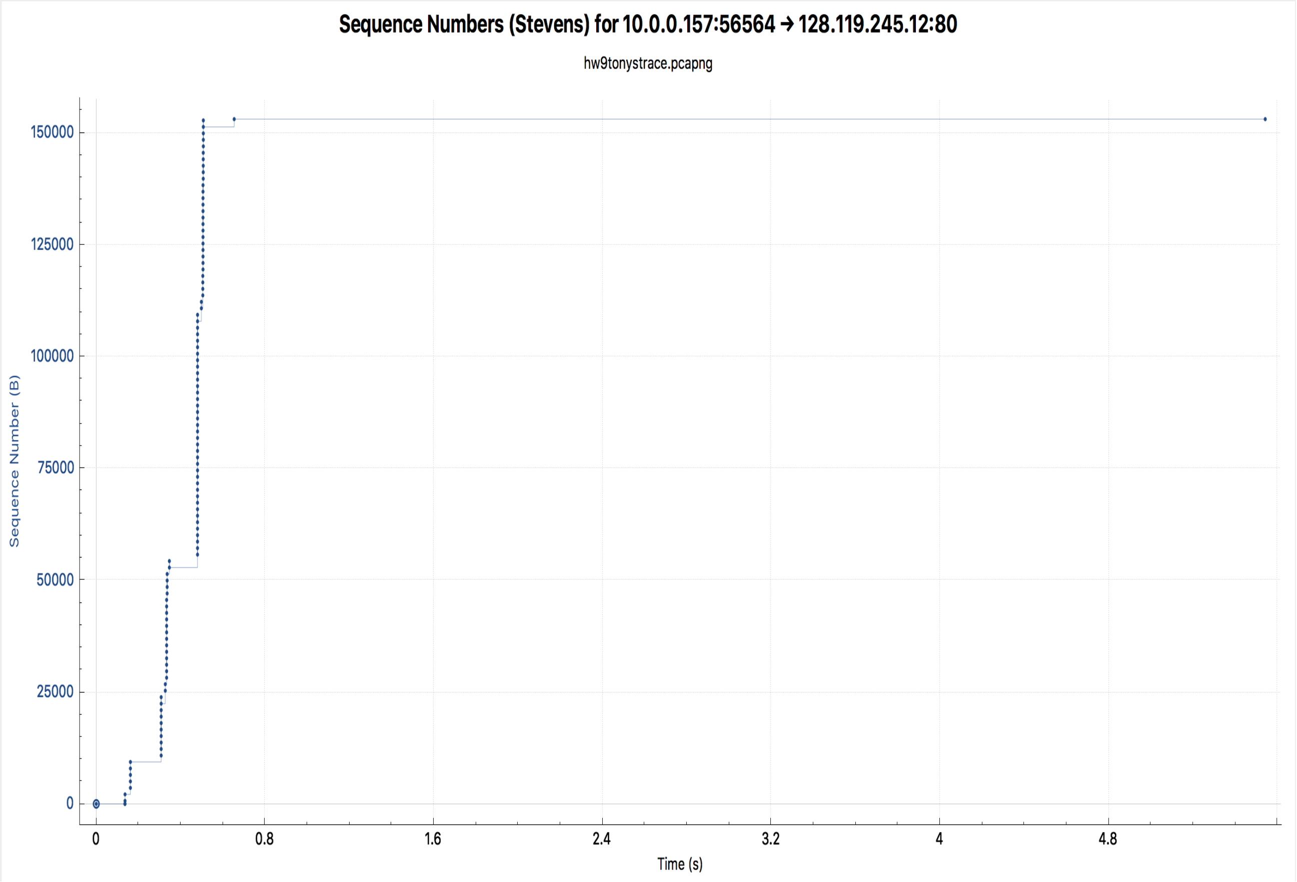
12. The average throughput seems to be approximately 225000 b/s = 28125 B/s = 28.125 KB/s over a 5 second time interval.

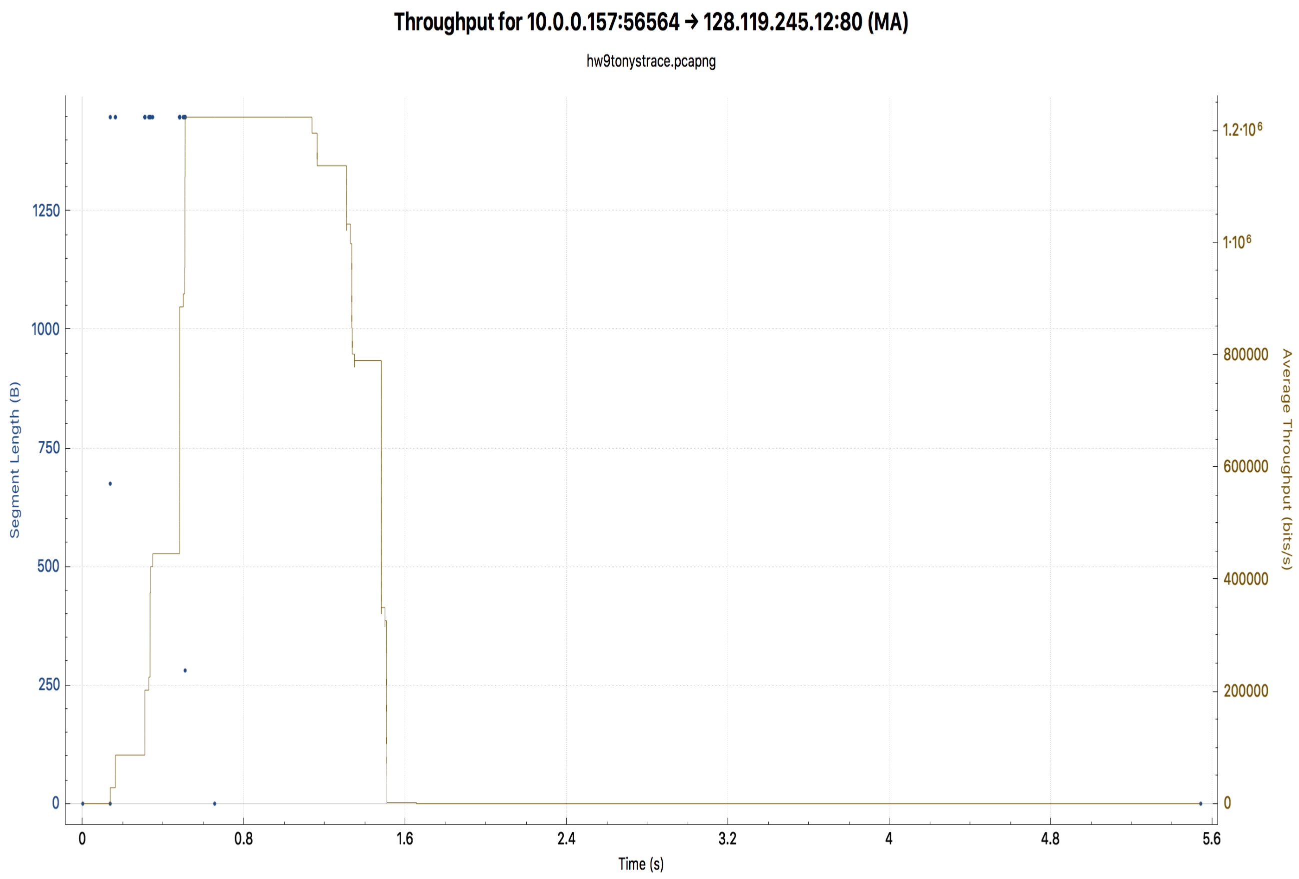


13. Slow start begins with the syn segment, the first segment sent. Slow start ends after sequence number 7866 is received (ack 9013). Congestion avoidance takes over starting from sequence 9013. Congestion avoidance ends after sequence 163769 is sent, which is the end of alice.txt. The TCP congestion avoidance behavior seen in the graph seems to max out at 6 unacked sequences at a time. In the ideal TCP, the number of unacked segments would grow by 1 MSS per RTT, so a sequence of [5, 6, 7, ...] unacked segments would be seen until either three duplicate acks are received or a timeout event occurs, at which point fast recovery would be entered and the congestion window would be decreased depending on which recovery algorithm is used. Instead, the congestion window seems to be a constant size. The maximum unacked segment size does not seem to be influenced by flow control, because the receiver window is much larger than $5 \cdot 1460$.



14. Slow start begins with the first segment sent. Slow start ends after segment 152716 is sent. The average throughput is approximately 700000 b/s = 87500 B/s = 87.5 KB/s over a 1.5s interval. The congestion window seems to grow exponentially until all the alice.txt file bytes are done transferring. There are no error events. Congestion avoidance and Fast recovery are never entered. This seems to closely mirror ideal TCP slow start behavior. My computer appears to have a slow start implementation similar to the reference algorithm suggested in RFC2001, as the slow start threshold appears to be large, at least 65535B. RFC5681 suggests a ssthresh as big as the largest possible rwnd. In the graph, it is clear that on the the command window size continued to double past the reference threshold suggested by RFC2001. 66 sequences are sent from T=.5s to T=.65s back to back. By looking at the wireshark packet capture listing, I see that the acks for those sequences do not begin to arrive until nearly all the sequences are sent, meaning the congestion window is on the order of $2^{log_2(66\cdot1460)} \approx 2^{17}$ bytes large at this point. If the file was larger, maybe my computer would have entered congestion avoidance sometime past this point. It is impossible to tell if there are any differences in congestion avoidance between my computer's TCP implementation and the ideal because congestion avoidance is never entered.




Throughput for 10.0.0.157:56564 → 128.119.245.12:80 (MA)
hw9tonystrace.pcapng
Segment Length (B)
1250
1000
750
500
250
0
Average Throughput (bits/s)
1.2·10⁶
1·10⁶
800000
600000
400000
200000
0
0
0.8
1.6
2.4
3.2
4
4.8
5.6
Time (s)