Erin Alltop
CS372 – Spring 2018
Lab 3

1. The IP address of the client computer (source) is 192.168.1.102

```
Internet Protocol Version 4, Src: 192.168.1.102,
```

IP Address of the source

The TCP port number used by the client computer is 1161

```
Transmission Control Protocol, Src Port: 1161,
```

TCP port number used by the source

2. The IP address of gaia.cs.umass.edu is 128.119.245.12 and the TCP port it is sending a receiving TCP segments for this connection is 80

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
```

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 164041, Ack: 1, Len: 50
```

3. The IP address of my computer is 192.168.0.16 and the TCP port number my computer used to transfer the file to gaia.cs.umass.edu is 62153.

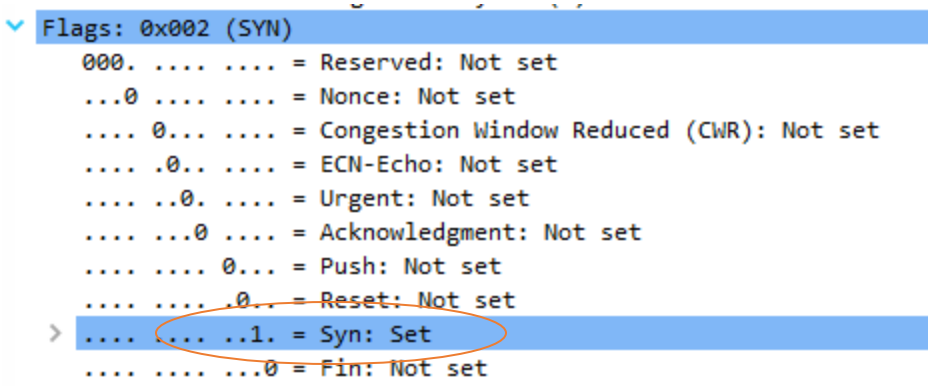
```
Internet Protocol Version 4, Src: 192.168.0.16, Dst: 128.119.245.12
```

```
Transmission Control Protocol, Src Port: 62153,
```

4. 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 is 0

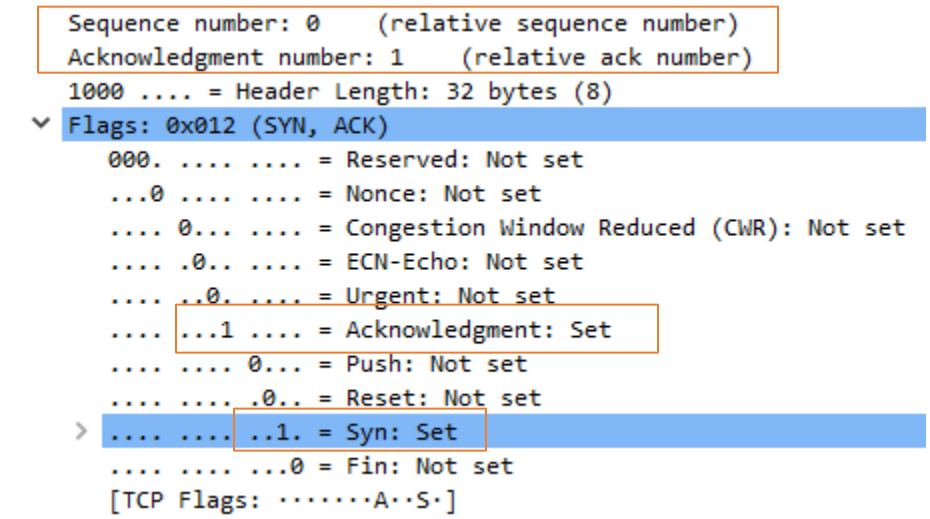
```
Sequence number: 0    (relative sequence number)
```

You can see that this is a SYN segment because the SYN flag is set to 1.



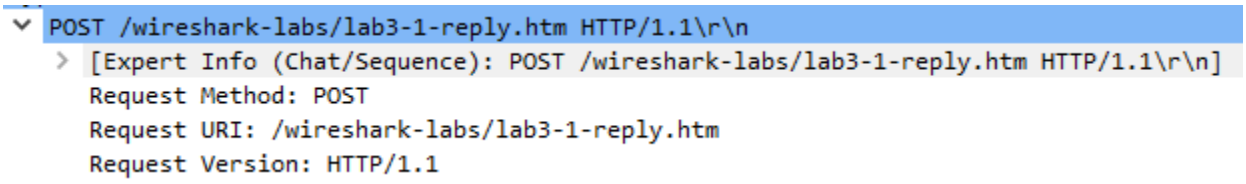
SYN flag is set to 1.

5. The sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is also 0. The value of the acknowledgement field in the SYNACK segment is 1. This value was determined by gaia.cs.umass.edu by adding 1 to the initial sequence number of the CYN segment from the client computer. You can see that this segment is a SYNACK segment because both the SYN and Acknowledgment flags are set to 1.

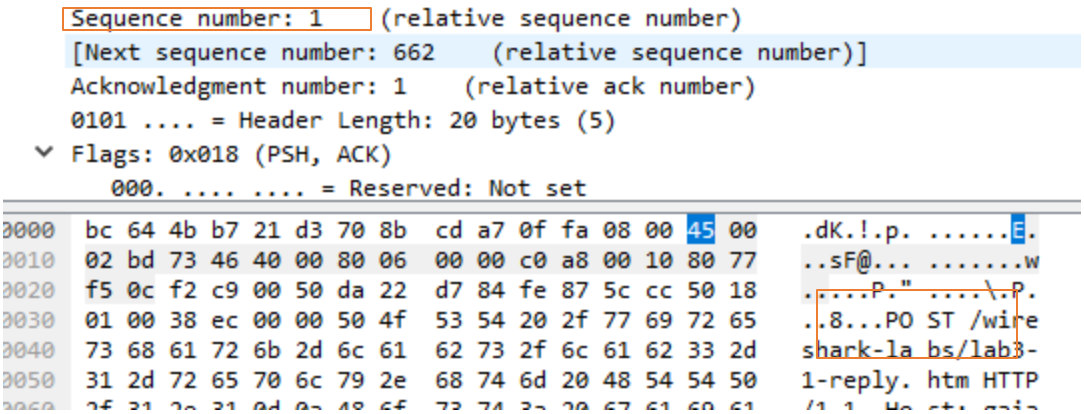


Sequence number, Acknowledgement number, and ACK / SYN flags are set to 1.

6. The sequence number of the TCP segment containing the HTTP POST command is 149765.

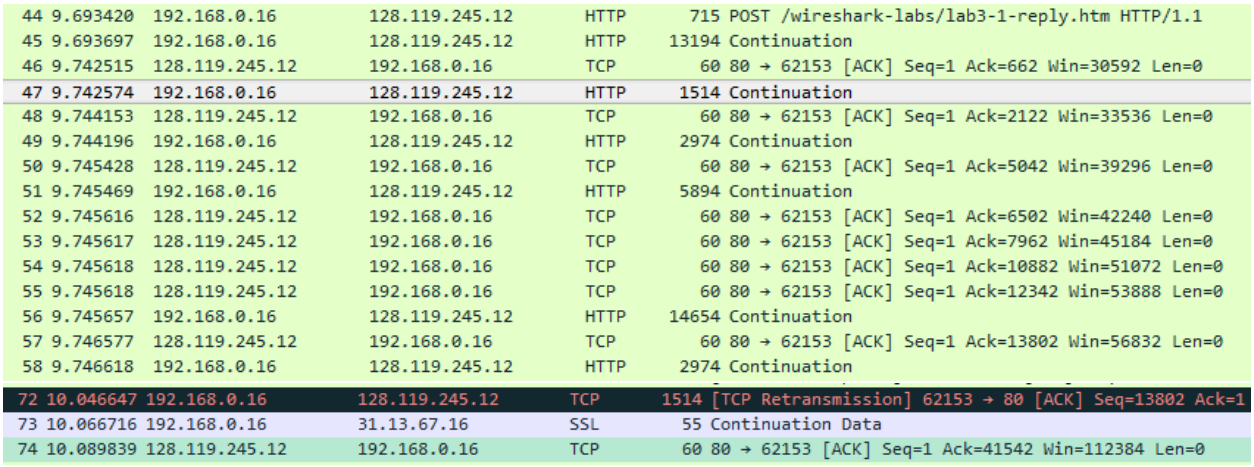


POST Data field



Sequence number of TCP segment with the POST command.

7. The sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST), with the time sent and Acknowledgement received for all:



The next six packets including the POST response with times on the left.

1. Seq = 1 sent at 9.693420; ACK received at 9.742515
   EstimatedRTT = 9.694s

2. Seq = 13802 sent at 9.742574; ACK received at 9.744153
   EstimatedRTT = 8.482s

3. Seq = 15262 sent at 9.744196; ACK received at 9.745428
   EstimatedRTT = 7.422s

4. Seq = 18182 sent at 9.745469; ACK received at 9.745618
   EstimatedRTT = 6.4943s

5. Seq = 24022 sent at 9.745657; ACK received at 9.746577

EstimatedRTT = 5.683s

6. Seq = 38622 sent at 9.746618; ACK received at 10.089839
   EstimatedRTT = 4.973s

8. The lengths of the first six TCP segments:

1. 701 bytes

```
Total Length: 701
```

2. 13180 bytes

```
[Total Length: 13180 bytes
```

3. 1500 bytes

```
Total Length: 1500
```

4. 2960 bytes

```
[Total Length: 2960 bytes (
```

5. 5880 bytes

```
[Total Length: 5880 bytes
```

6. 14640 bytes

```
[Total Length: 14640 bytes
```

9. The minimum amount of available buffer space advertised at the received for the entire trace is 8192, shown in the first ACK from the server.

```
 4 0.014970  192.168.0.16     40.121.213.159  TCP  66 62142 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
 6 0.015460  192.168.0.16     13.107.3.128    TCP  66 62143 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
 7 0.054966  40.121.213.159   192.168.0.16    TCP  66 443 → 62142 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1
 8 0.055036  192.168.0.16     40.121.213.159  TCP  54 62142 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
 9 0.062175  13.107.3.128     192.168.0.16    TCP  66 443 → 62143 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM=1
10 0.062222  192.168.0.16     13.107.3.128    TCP  54 62143 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
11 0.251617  192.168.0.16     13.107.3.128    TCP  54 62143 → 443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
12 0.251732  192.168.0.16     40.121.213.159  TCP  54 62142 → 443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

The lack of receiver buffer does not seem to throttle the sender from what I can tell. The buffer seems to grow as well up to 183296.

```
151 10.314608 128.119.245.12  192.168.0.16  TCP  60 80 → 62153 [ACK] Seq=1 Ack=143925 Win=183296 Len=0
152 10.314637 128.119.245.12  192.168.0.16  TCP  60 80 → 62153 [ACK] Seq=1 Ack=146845 Win=183296 Len=0
153 10.319508 128.119.245.12  192.168.0.16  TCP  60 80 → 62153 [ACK] Seq=1 Ack=149765 Win=183296 Len=0
154 10.320406 128.119.245.12  192.168.0.16  TCP  60 80 → 62153 [ACK] Seq=1 Ack=152983 Win=183296 Len=0
```

10. Yes there was one segment that was retransmitted. By examining the trace it appears as a black box with [TCP Retransmission] as the label. This was the only segment retransmitted:

```
71 9.793024  128.119.245.12   192.168.0.16     TCP      66 [TCP Window Update] 80 → 62153 [ACK] Seq=1 Ack=13802 Win=109440 Len=0 SLE=15262 SRE=41542
72 10.046647 192.168.0.16     128.119.245.12   TCP    1514 [TCP Retransmission] 62153 → 80 [ACK] Seq=13802 Ack=1 Win=65536 Len=1460
73 10.066716 192.168.0.16     31.13.67.16      SSL      55 Continuation Data
74 10.089839 128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=41542 Win=112384 Len=0
75 10.089925 192.168.0.16     128.119.245.12   HTTP  13194 Continuation
```

Retransmitted packet.

11.

By inspecting the Acknowledgement sequence numbers and the amount of data that is transmitted between segments, I can see that the receiver is ACKing every other received segment in a few cases because the data in the segments sent to the receiver are smaller than the size of the data that the receiver is ACKing.

Acknowledgement	ACK Sequence Number	Acknowledged Data
ACK 1	1	1
ACK 2	662	661
ACK 3	2122	1461
ACK 4	5042	3581
ACK 5	6502	2921
ACK 6	7962	5041
ACK 7	10882	5841
ACK 8	12342	6501
ACK 9	13802	7301

```
41 9.641915  192.168.0.16     128.119.245.12   TCP      66 62153 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
42 9.692298  128.119.245.12   192.168.0.16     TCP      66 80 → 62153 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
43 9.692396  192.168.0.16     128.119.245.12   TCP      54 62153 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
44 9.693420  192.168.0.16     128.119.245.12   HTTP    715 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1
45 9.693697  192.168.0.16     128.119.245.12   HTTP  13194 Continuation
46 9.742515  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=662 Win=30592 Len=0
47 9.742574  192.168.0.16     128.119.245.12   HTTP   1514 Continuation
48 9.744153  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=2122 Win=33536 Len=0
49 9.744196  192.168.0.16     128.119.245.12   HTTP   2974 Continuation
50 9.745428  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=5042 Win=39296 Len=0
51 9.745469  192.168.0.16     128.119.245.12   HTTP   5894 Continuation
52 9.745616  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=6502 Win=42240 Len=0
53 9.745617  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=7962 Win=45184 Len=0
54 9.745618  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=10882 Win=51072 Len=0
55 9.745618  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=12342 Win=53888 Len=0
56 9.745657  192.168.0.16     128.119.245.12   HTTP  14654 Continuation
57 9.746577  128.119.245.12   192.168.0.16     TCP      60 80 → 62153 [ACK] Seq=1 Ack=13802 Win=56832 Len=0
58 9.746618  192.168.0.16     128.119.245.12   HTTP   2974 Continuation
```

ACKing segments and their sequence numbers.

NOTE: It was at this point that my computer crashed and I lost the data that I used to answer the above questions (stupidly did not save). I re-did the setup and am using the new trace to answer the remaining questions.

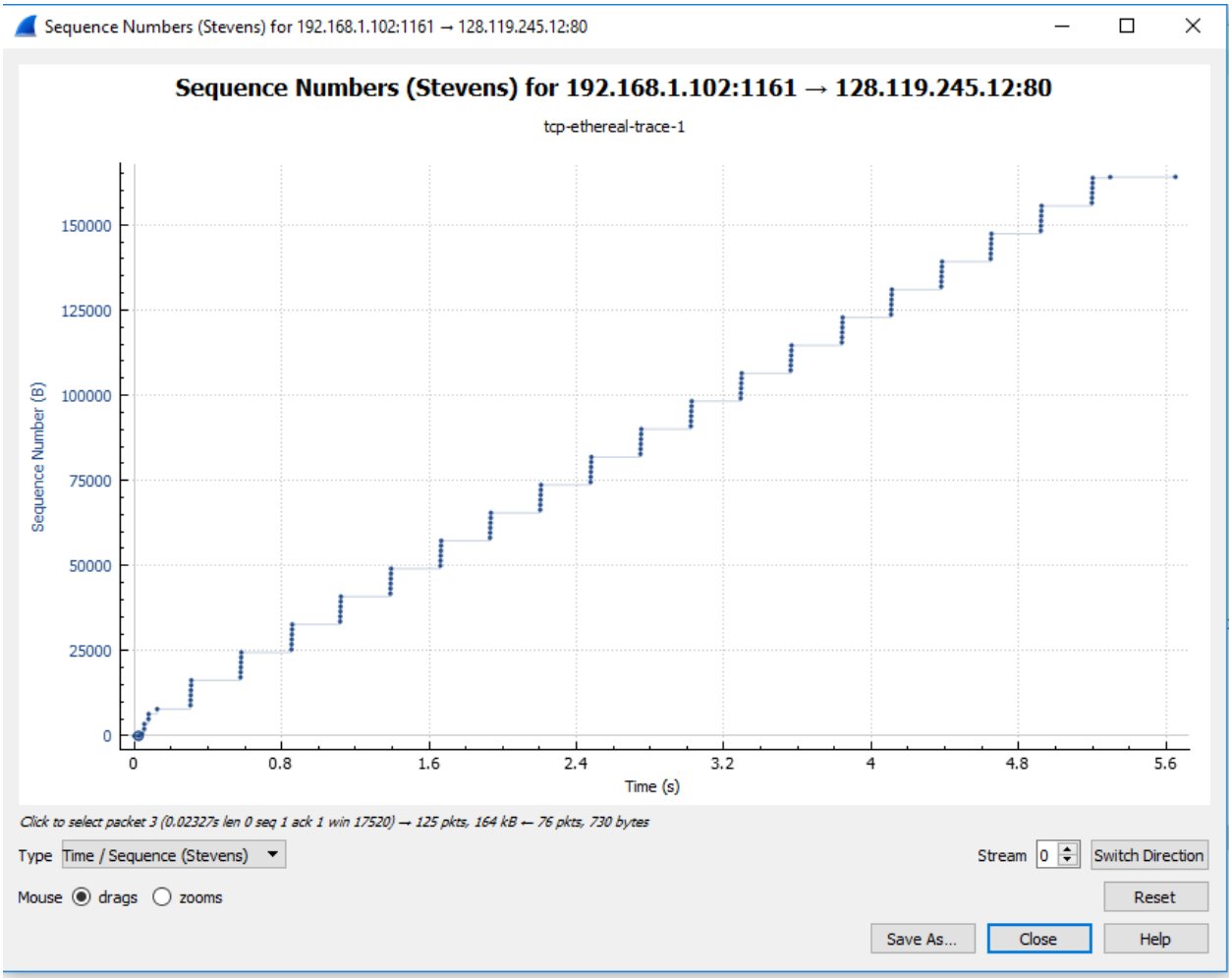
12. I calculated the throughput (bytes transferred per unit time) for the TCP connection by calculating the ratio of the total amount of data transferred and the total transmission time. I found the total

amount of data by subtracting the first sequence number of the first TCP segment by the acknowledgement number of the last ACK.

The first TCP segment has a sequence number of 1 and the last ACK acknowledgem/ent number is 152983. So the total data transmitted can be calculated as 152983 – 1 = 152982 bytes. I calculated the total transmission time in the same way with the time the first TCP segment was transmitted (4.565557) and the transmission time of the last ACK segment (4.771343). 4.771343 – 4.565557 = 0.205786 seconds.

Using the ratio of total amount of data and the total transmission time, the average throughput is 152982 / 0.205786 = 743403.341 = 743.403 KB/sec.

13.



Looking at the graph, it appears that the slow start for this trace starts at about 0.02327s and continues until just after the first ACT at 0.04s or so when congestion avoidance takes over. After that it is consistent with the amount of congestion and no more slowing occurs. I think this is because the packets for this trace are equal so the calculations for congestion avoidance were able to happen quickly and even out the congestion almost right off the bat.

14. This is the Time-Sequence (Stevens) graph for my own computer transferring the file to gaia.cs.umass.edu. The packets on my machine were not nearly as equal as in the sample data above. The first packet (the POST) was packet 31 at about 4.454s. You can see the slow start begin and end at about packet 45 (4.567s) when the rate speeds up a good deal, this slows at packet 64 (4.666s) and increases again at packet 66 (4.7s). Both these spikes indicate that congestion avoidance mode is in effect. The slow start occurs initially when the sender is attempting to find out the available bandwidth and the congestion avoidance occurs when cwnd equals ssthresh in calculating how much bandwidth is available for data transfer.