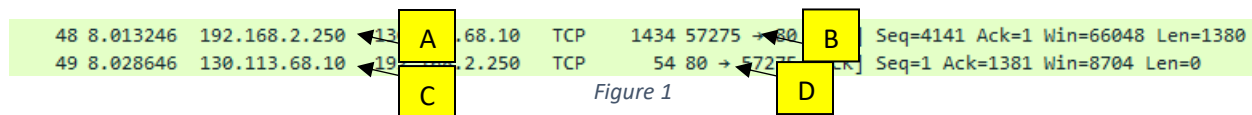


2. A first look at the captured trace

1. IP address: 192.168.2.250 (Figure 1, A)
TCP port number: 57275 (Figure 1, B)
2. IP address: 130.113.68.10 (Figure 1, C)
TCP port number: 80 (Figure 1, D)



3. TCP Basics

3. Sequence number: 0 (relative sequence number) (Figure 2, A)
The 0x002 SYN flag identifies the segment by setting the Syn bit. (Figure 2, B)

```
Transmission Control Protocol, Src Port: 57275, Dst Port: 80, Seq: 0, Len: 0
Source Port: 57275
Destination Port: 80
[Stream index: 8]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number) ← A
Acknowledgment number: 0
1000 .... = Header Length: 32 bytes (8)
▼ Flags: 0x002 (SYN) ← B
  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
```

Figure 2

4. Sequence number: 0 (relative sequence number) (Figure 3, A)
Acknowledge number: 1 (relative ack number) (Figure 3, B), determined by incrementing the SYN sequence number.
The 0x012 SYN, ACK flag identifies the segment. (Figure 3, C)

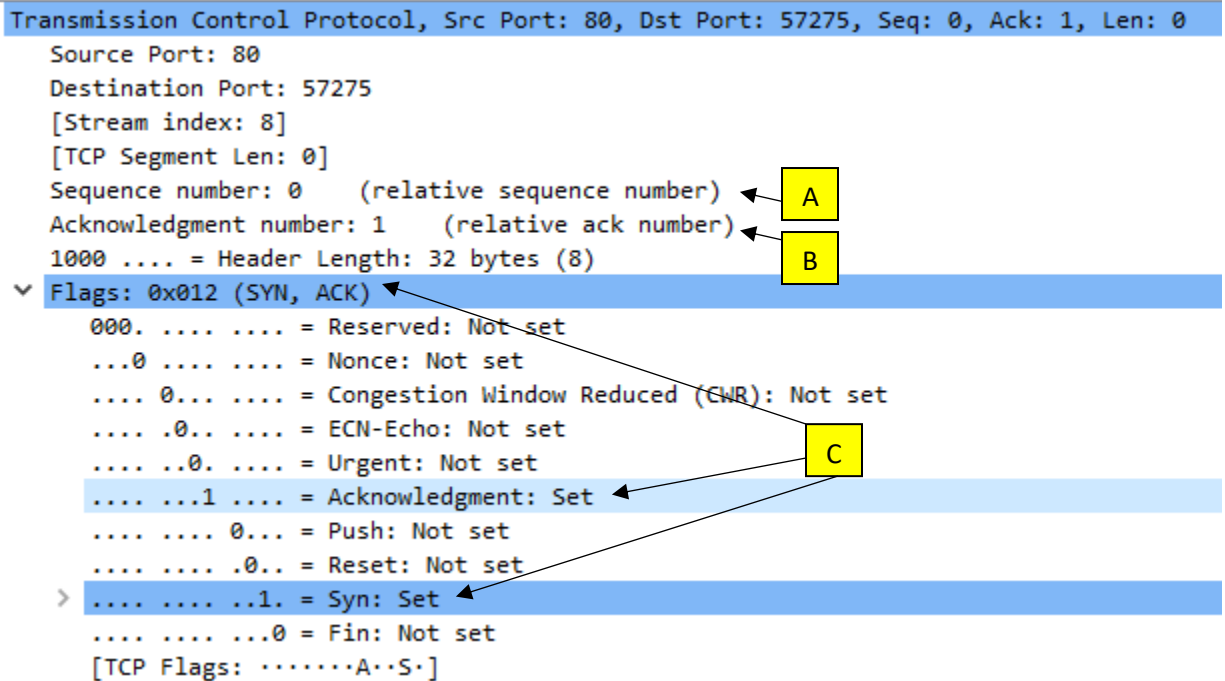


Figure 3

5. Sequence number: 1 (relative sequence number) (Figure 4, A)

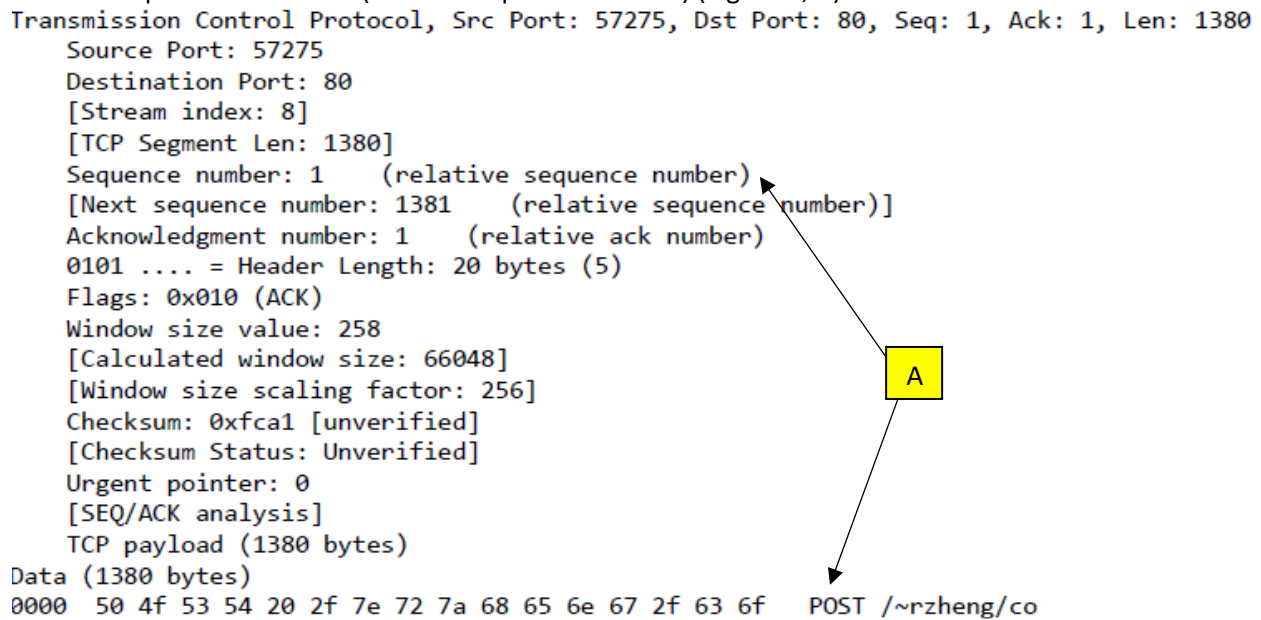


Figure 4

6.

Segment	Sequence Number	Time sent	ACK received	RTT	EstimatedRTT
1	1	8.013127	8.028646	0.015519	0.015519
2	1381	8.013157	8.028729	0.015572	0.015526
3	2761	8.013229	8.030084	0.016855	0.015692
4	4141	8.013246	8.031479	0.018233	0.01601
5	5521	8.028769	8.044303	0.015534	0.015951

6	6901	8.028786	8.044442	0.015656	0.015914
---	------	----------	----------	----------	----------

Sequence Number (Figure 5, A)

Time sent (Figure 5, B)

ACK received (Figure 5, C)

RTT for packet with sequence number 1381 in frame number 46. (Figure 6, A)

Estimated RTT = $(1-0.125) * \text{EstimatedRTT} + 0.125 * \text{RTT}$

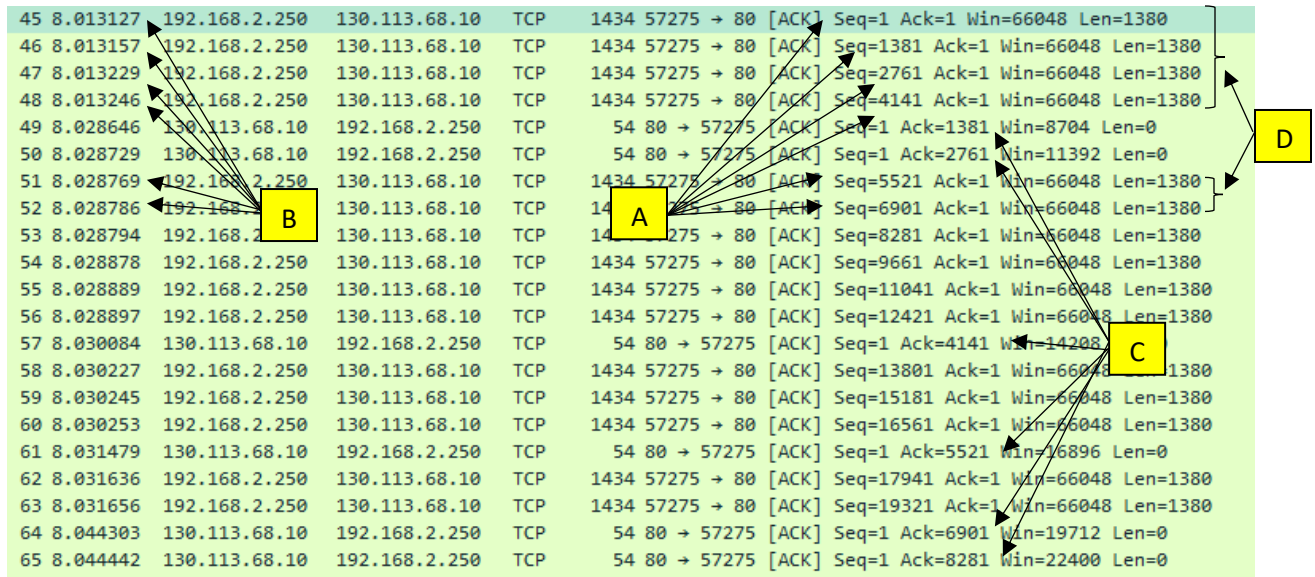


Figure 5

Transmission Control Protocol, Src Port: 80, Dst Port: 57275, Seq: 1, Ack: 2761, Len: 0

Source Port: 80

Destination Port: 57275

[Stream index: 8]

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number)

Acknowledgment number: 2761 (relative ack number)

0101 = Header Length: 20 bytes (5)

> Flags: 0x010 (ACK)

Window size value: 89

[Calculated window size: 11392]

[Window size scaling factor: 128]

Checksum: 0x8046 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

✓ [SEQ/ACK analysis]

[This is an ACK to the segment in frame: 46]

[The RTT to ACK the segment was: 0.015572000 seconds]

[iRTT: 0.015558000 seconds]

Figure 6

7. 1380 bytes for all six packets (Figure 5, D)

8. Minimum available buffer space (receiver window size value): 5840 (Figure 7, A) from the (SYN, ACK). The lack of receiver space does not throttle the sender due to lack of receiver buffer space.

```
Transmission Control Protocol, Src Port: 80, Dst Port: 57275, Seq: 0, Ack: 1, Len: 0
Source Port: 80
Destination Port: 57275
[Stream index: 8]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Acknowledgment number: 1 (relative ack number)
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x012 (SYN, ACK)
Window size value: 5840 ← A
[Calculated window size: 5840]
```

Figure 7

9. Yes, there are retransmitted segments in the trace file. By checking for a TCP duplicate acknowledgement and [TCP Retransmission] packets in the trace. (Figure 8, A)

```
198 8.117924 130.113.68.10 192.168.2.250 54 80 → 57275 [ACK] Seq=1 Ack=98389 Win=64128 Len=0
199 8.120501 130.113.68.10 192.168.2.250 54 80 → 57275 [ACK] Seq=1 Ack=101149 Win=64128 Len=0
200 8.120614 130.113.68.10 192.168.2.250 66 [TCP Dup ACK 199#1] 80 → 57275 [ACK] Seq=1 Ack=101149 Win=64128 Len=0 SLE=143929 SRE=143963
201 8.120793 130.113.68.10 192.168.2.250 66 80 → 57275 [ACK] Seq=1 Ack=102529 Win=63872 Len=0 SLE=143929 SRE=143963
202 8.120843 192.168.2.250 130.113.68.10 TCP 1434 [TCP Spurious Retransmission] 57275 → 80 [ACK] Seq=101149 Ack=1 Win=66048 Len=1380
203 8.120978 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=102529 Ack=1 Win=66048 Len=1380
204 8.121003 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=103909 Ack=1 Win=66048 Len=1380
205 8.121022 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=105289 Ack=1 Win=66048 Len=1380
206 8.121036 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=106669 Ack=1 Win=66048 Len=1380
207 8.121049 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=108049 Ack=1 Win=66048 Len=1380
208 8.121063 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=109429 Ack=1 Win=66048 Len=1380
209 8.121077 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=110809 Ack=1 Win=66048 Len=1380
210 8.121091 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=112189 Ack=1 Win=66048 Len=1380
211 8.121105 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=113569 Ack=1 Win=66048 Len=1380
212 8.121119 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=114949 Ack=1 Win=66048 Len=1380
213 8.121139 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=116329 Ack=1 Win=66048 Len=1380
214 8.121152 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=117709 Ack=1 Win=66048 Len=1380
215 8.121165 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=119089 Ack=1 Win=66048 Len=1380
216 8.121178 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=120469 Ack=1 Win=66048 Len=1380
217 8.121190 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=121849 Ack=1 Win=66048 Len=1380
218 8.121203 192.168.2.250 130.113.68.10 TCP 1434 [TCP Retransmission] 57275 → 80 [ACK] Seq=123229 Ack=1 Win=66048 Len=1380
219 8.122851 130.113.68.10 192.168.2.250 TCP 66 80 → 57275 [ACK] Seq=1 Ack=103909 Win=63872 Len=0 SLE=143929 SRE=143963
220 8.122959 130.113.68.10 192.168.2.250 TCP 66 80 → 57275 [ACK] Seq=1 Ack=105289 Win=63872 Len=0 SLE=143929 SRE=143963
```

Figure 8

10. The difference between ACK sequence numbers of two consecutive ACKs is the amount of data acknowledged by the receiver. As seen below, the receiver typically acknowledges 1380 bytes, but I find some instances, such as acknowledgement sequence number 69001 where 2760 bytes of data was acknowledged. (Figure 9, A)

ACK Sequence Number	Data
1381	1380
2761	1380
4141	1380
5521	1380
6901	1380
8281	1380
9661	1380
...	
66241	...
69001	2760
...	

172	8.092000	192.168.2.250	130.113.68.10	TCP	54	80 → 57275	[ACK] Seq=123989 Ack=66241 Win
173	8.100502	130.113.68.10	192.168.2.250	TCP	54	80 → 57275	[ACK] Seq=123989 Ack=66241 Win
174	8.100639	130.113.68.10	192.168.2.250	TCP	54	80 → 57275	[ACK] Seq=123989 Ack=66241 Win
175	8.100697	130.113.68.10	192.168.2.250	TCP	54	80 → 57275	[ACK] Seq=123989 Ack=66241 Win

Figure 9

11. I calculate the throughput by averaging over the entire time duration during which the file was transferred.

$$\frac{143929 - 1}{8.104559 - 8.013127} = 1574153.47 \text{ bytes/seconds}$$

3. TCP congestion control in action

12. TCP slowstart phase begins at 0.015558 seconds and ends at 0.05 seconds, after which congestion avoidance takes over. The ideal TCP sender behaviour follows the AIMD algorithm, sending plenty of data and then decreasing the window size upon detecting congestion. An application effects TCP behavior. For an application with several small objects where transmission time is less than the length of the slow start phase, transmission experiences long delays.

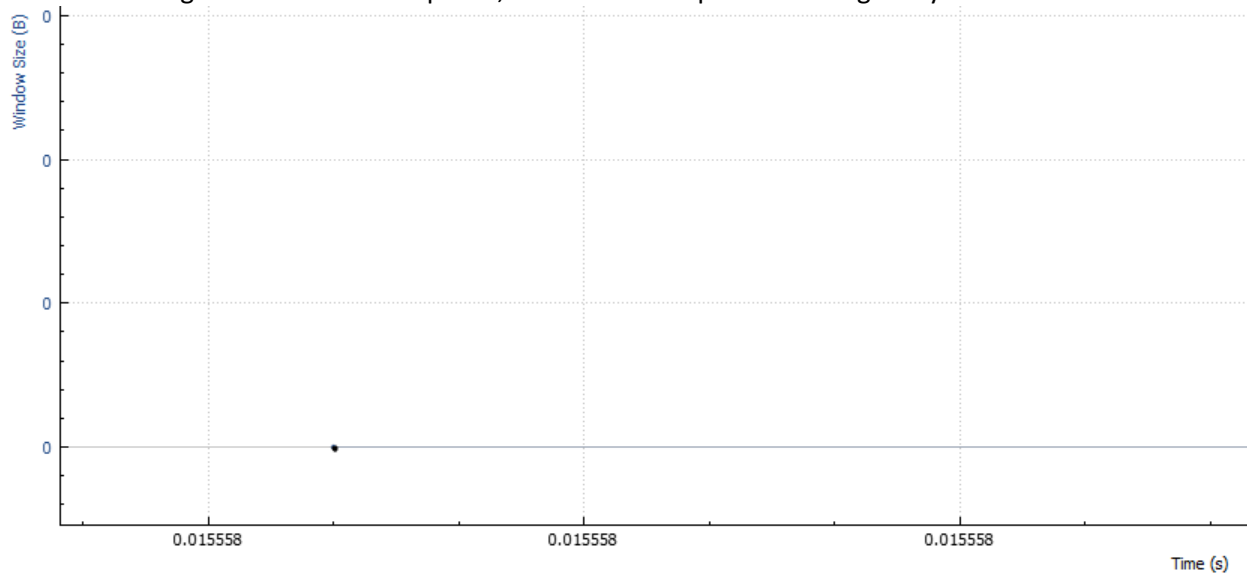


Figure 10

UMME SALMA GADRIWALA - 400021431
4C03 – ASSIGNMENT 2

