BAYANI, Femo Steven Felicisimo

2015-07104

Wireshark Lab 7: TCP

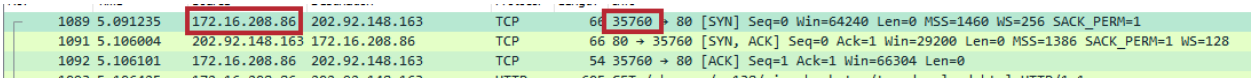
CMSC 138

1. Place the captured three-way handshake packets with their corresponding TCP fields displayed.

```
1089 5.091235    172.16.208.86  202.92.148.163  TCP    66 35760 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1091 5.106004    202.92.148.163 172.16.208.86   TCP    66 80 → 35760 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
1092 5.106101    172.16.208.86  202.92.148.163  TCP    54 35760 → 80 [ACK] Seq=1 Ack=1 Win=66304 Len=0

> Frame 1089: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: Azurewav_15:fd:65 (54:27:1e:15:fd:65), Dst: PaloAlto_67:00:13 (58:49:3b:67:00:13)
> Internet Protocol Version 4, Src: 172.16.208.86, Dst: 202.92.148.163
v Transmission Control Protocol, Src Port: 35760, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 35760
    Destination Port: 80
    [Stream index: 58]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 64240
    [Calculated window size: 64240]
    Checksum: 0x3d59 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted

1091 5.106004    202.92.148.163 172.16.208.86   TCP    66 80 → 35760 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1386 SACK_PERM=1 WS=128
1092 5.106101    172.16.208.86  202.92.148.163  TCP    54 35760 → 80 [ACK] Seq=1 Ack=1 Win=66304 Len=0

> Frame 1091: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: PaloAlto_67:00:13 (58:49:3b:67:00:13), Dst: Azurewav_15:fd:65 (54:27:1e:15:fd:65)
> Internet Protocol Version 4, Src: 202.92.148.163, Dst: 172.16.208.86
v Transmission Control Protocol, Src Port: 80, Dst Port: 35760, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 35760
    [Stream index: 58]
    [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: 29200
    [Calculated window size: 29200]
    Checksum: 0xf46a [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  v [SEQ/ACK analysis]

1092 5.106101    172.16.208.86  202.92.148.163  TCP    54 35760 → 80 [ACK] Seq=1 Ack=1 Win=66304 Len=0

> Frame 1092: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: Azurewav_15:fd:65 (54:27:1e:15:fd:65), Dst: PaloAlto_67:00:13 (58:49:3b:67:00:13)
> Internet Protocol Version 4, Src: 172.16.208.86, Dst: 202.92.148.163
v Transmission Control Protocol, Src Port: 35760, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 35760
    Destination Port: 80
    [Stream index: 58]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 259
    [Calculated window size: 66304]
    [Window size scaling factor: 256]
    Checksum: 0xa600 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  v [SEQ/ACK analysis]
```

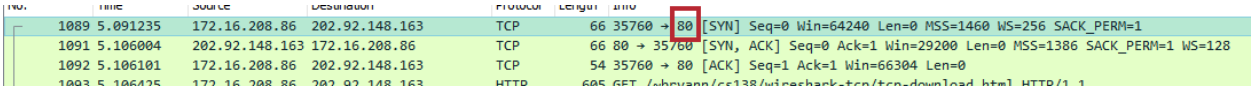
2. What is the IP address and TCP port number used by the client computer (source) that is downloading the file from agila.upm.edu.ph?

The IP address used is **172.16.208.86**, with TCP port number **35760**.



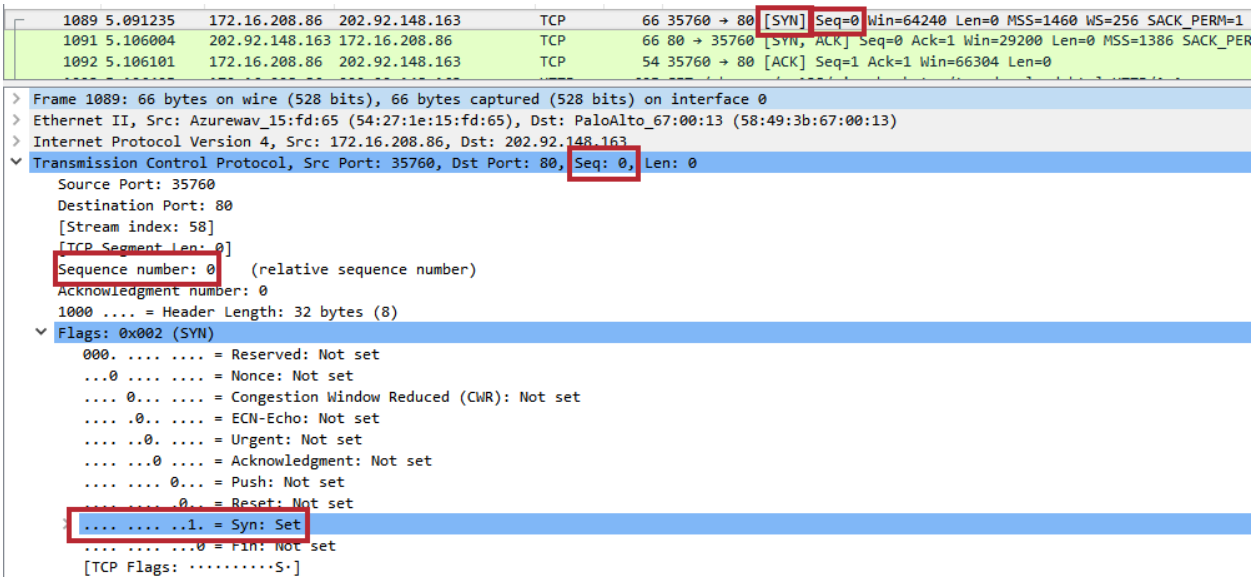
3. On what port number is agila.upm.edu.ph sending and receiving TCP segments for this connection?

Port number **80**



4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and agila.upm.edu.ph that is used to download the installer? What is it in the segment that identifies the segment as a SYN segment?

The sequence number is **0**, and the **SYN flag is set to 1**, identifying it as a SYN segment.



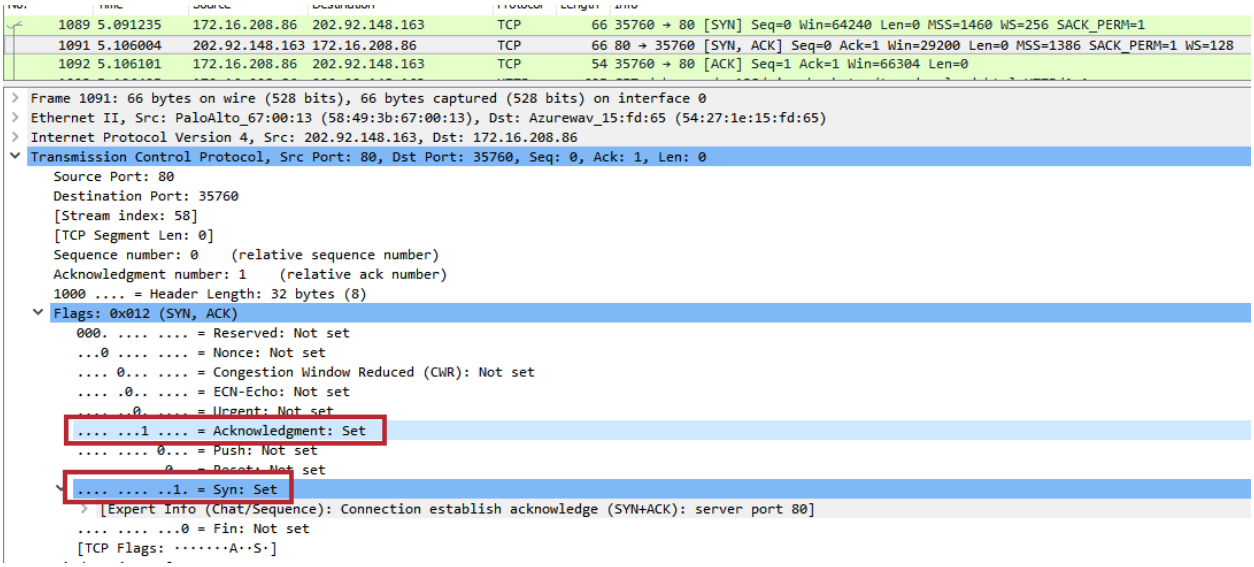
5. With respect to your answer in no. 4, how were you able to determine that this is the correct SYN segment for the TCP connection used in downloading the installer? Note that browsers usually make multiple TCP connections to a server.

6. Using the same TCP connection mentioned in no. 4, what is the sequence number of the SYN ACK segment sent by agila.upm.edu.ph to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYN ACK segment? How did agila.upm.edu.ph determine that value? What is it in the segment that identifies the segment as a SYN ACK segment?

The sequence number of the SYN ACK segment is also **0**.

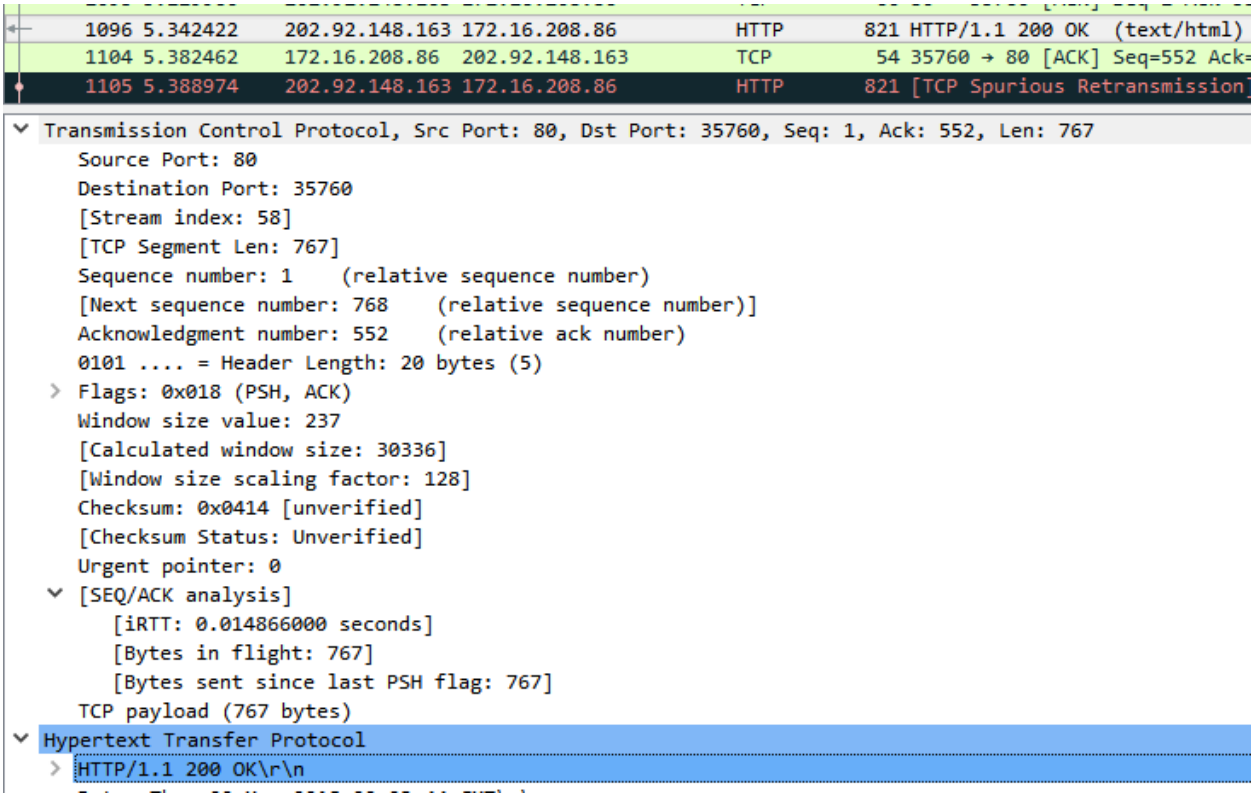
The value of the Acknowledgement field in the SYN ACK segment is **1** and was determined by getting the value of the initial sequence number and adding 1 to it, then setting it as the acknowledgement number, as seen by the “Acknowledgement number: 1”.

The segment is identified as a SYN ACK segment if both the SYN and ACK flags are set to 1.



7. What is the sequence number of the TCP segment containing the HTTP status code (200 OK)? Note that in order to find that segment, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a 200 OK status code within its DATA field.

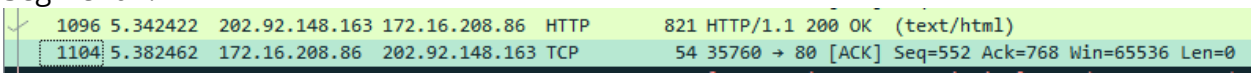
The sequence number is **1**.



8. Consider the TCP segment containing the 200 OK status code as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the 200 OK status code)? At what time was each segment sent? When was the ACK for each segment received? 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? What is the EstimatedRTT value after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation in our lecture for all subsequent segments.

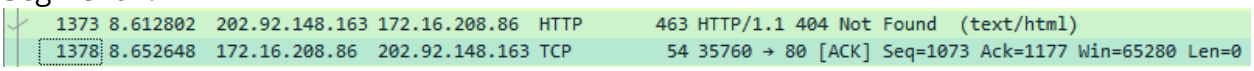
Segment no.	Seq. nos.	Time segment sent	Time ACK received	RTT value
1	768	5.342422	5.842462	0.040040000 s
2	1177	8.612802	8.652648	0.039846000 s
3	1705	11.290583	11.314872	0.024289000 s
4	3949	11.314873	11.314929	0.000056000 s
5	15037	11.318080	11.318172	0.000092000 s
6	17809	11.343678	11.343713	0.000035000 s

Segment 1:



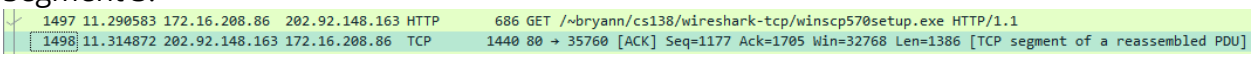
EstimatedRTT = RTT of Segment 1 = 0.04004 seconds

Segment 2:



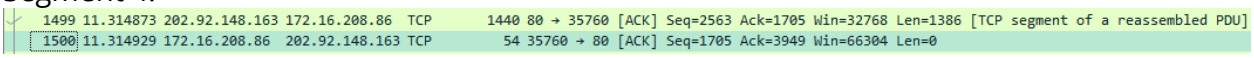
EstimatedRTT = 0.875 * 0.04004 + 0.125 * 0.039846 = 0.0400 seconds

Segment 3:



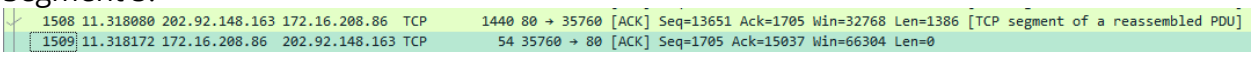
EstimatedRTT = 0.875 * 0.0400 + 0.125 * 0.024289 = 0.0380 seconds

Segment 4:



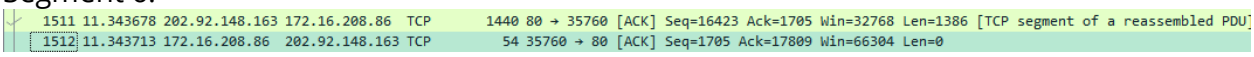
EstimatedRTT = 0.875 * 0.0380 + 0.125 * 0.000056 = 0.0333 seconds

Segment 5:



EstimatedRTT = 0.875 * 0.0333 + 0.125 * 0.000092 = 0.0291 seconds

Segment 6:

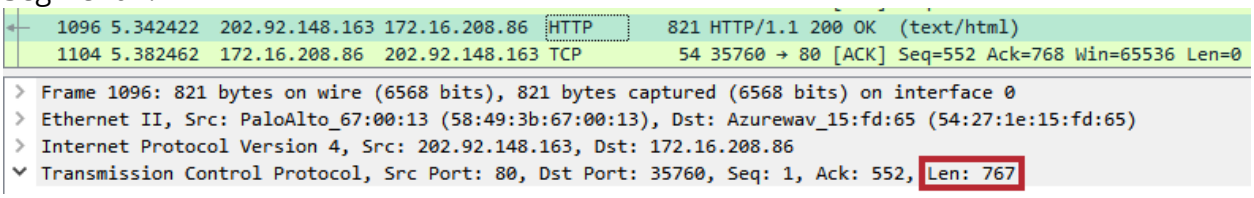


EstimatedRTT = 0.875 * 0.0291 + 0.125 * 0.000035 = 0.0255 seconds

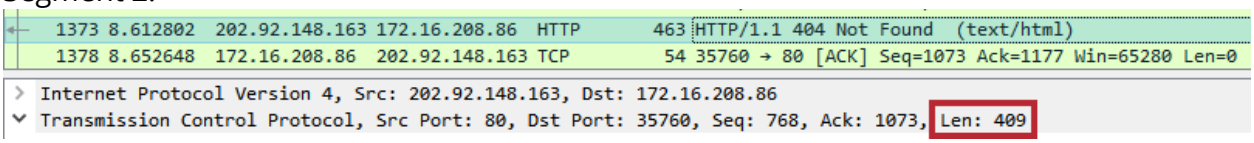
9. What is the length of each of the first six TCP segments?

Segment 1	767
Segment 2	409
Segment 3	632
Segment 4	1386
Segment 5	1386
Segment 6	1386

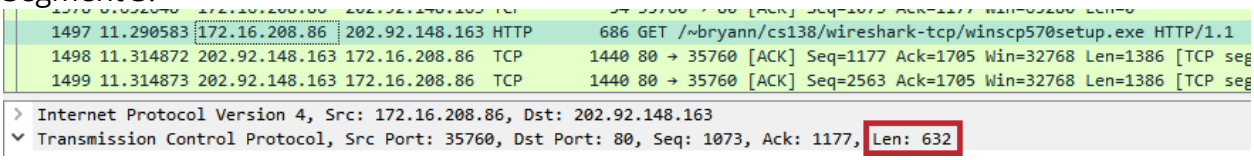
Segment 1:



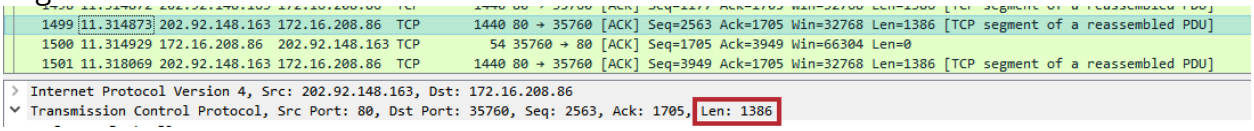
Segment 2:



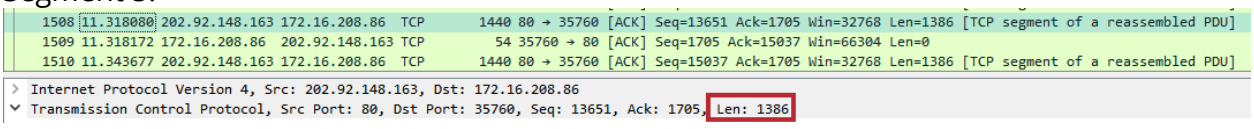
Segment 3:



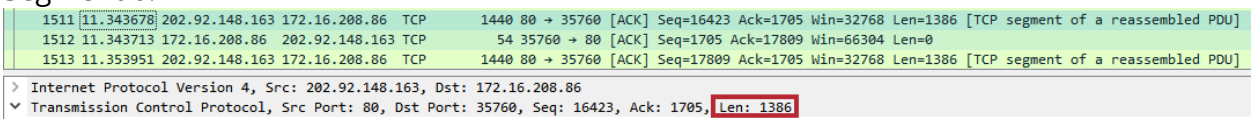
Segment 4:



Segment 5:



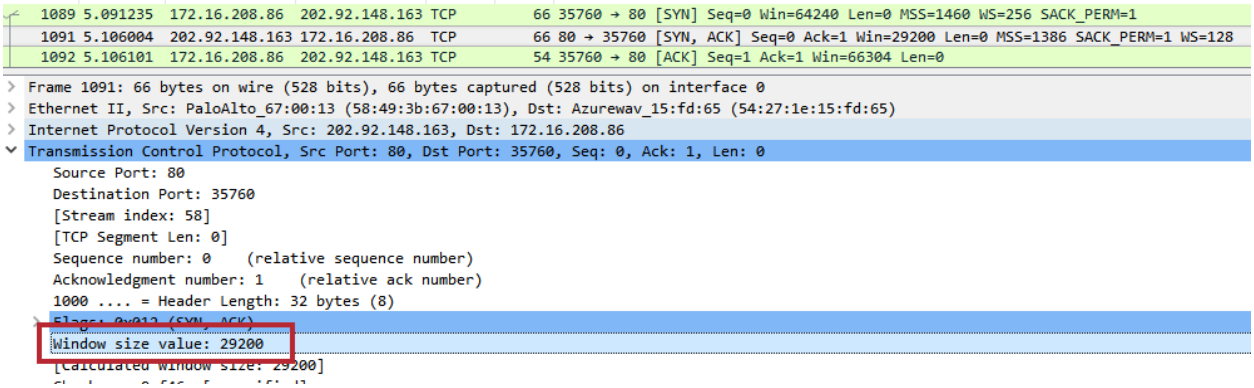
Segment 6:



10. What is the minimum amount of available buffer space advertised at the received window for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

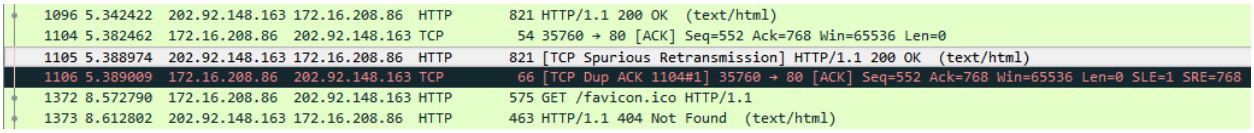
The minimum amount of available buffer space advertised at the received window is **29200 bytes**.

The sender is **not throttled** due to lack of receiver buffer space.



11. Are there any retransmitted segments? What did you check for in order to answer this question?

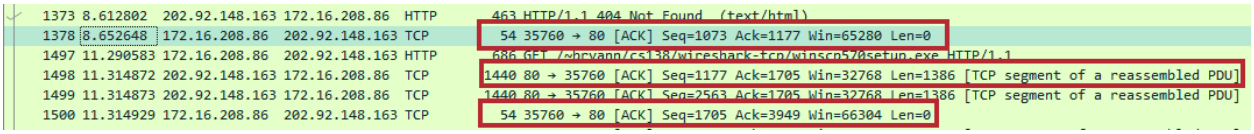
There are transmitted segments, they are indicated by Wireshark as “TCP Spurious Retransmission”.



12. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment?

ACK no.	Acknowledged seq. no.	Acknowledged data
1	768	768
2	1177	409
3	1705	528
4	3949	2244
5	15037	11088
6	17809	2772

There are cases where the receiver is ACKing every other segment as indicated by No. 1378 with No. 1498, and No. 1498 with No. 1500:



13. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Total bytes	5557233 - 1 = 5,557,232 bytes
Time taken	109.374958 - 5.106425 = 104.68533 seconds
Throughput	5557232/104.68533 = 53297.3068682 bytes/second = 53.297 Kbytes/sec

14. 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 agila.upm.edu.ph server. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the lecture.

The trace didn't show the end of the slowstart and congestion control since the sending was not enough to reach the congestion stage.