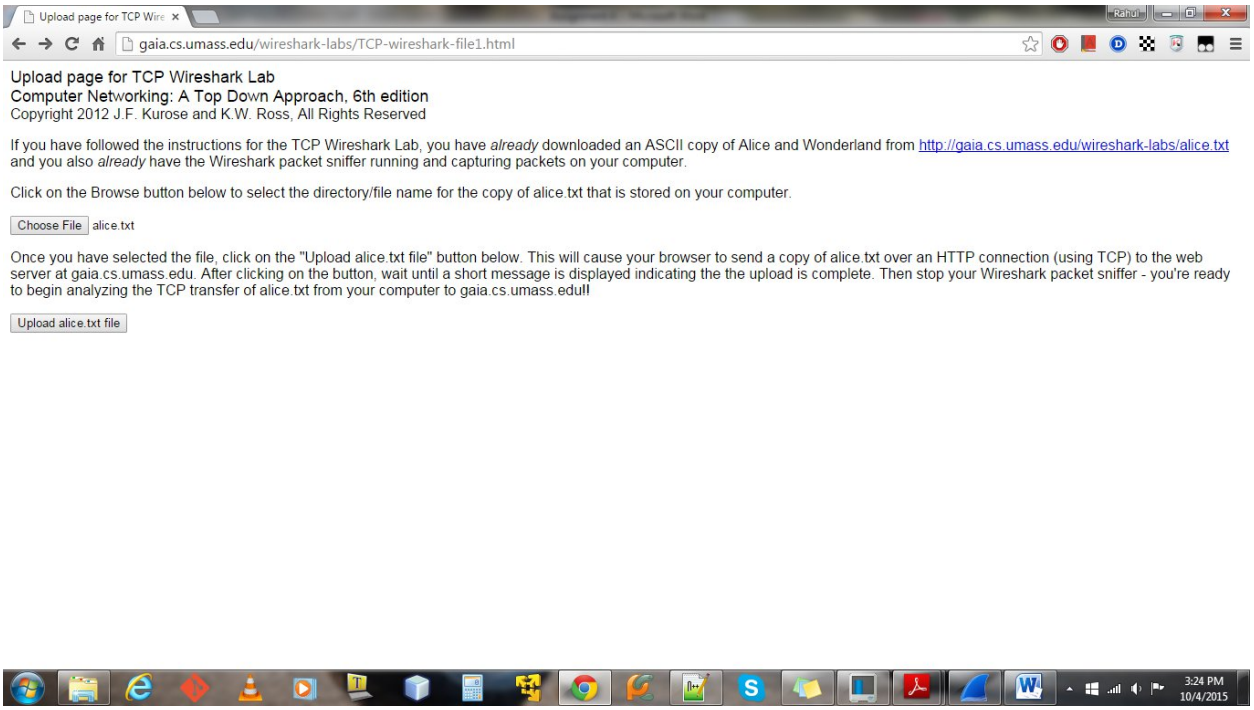
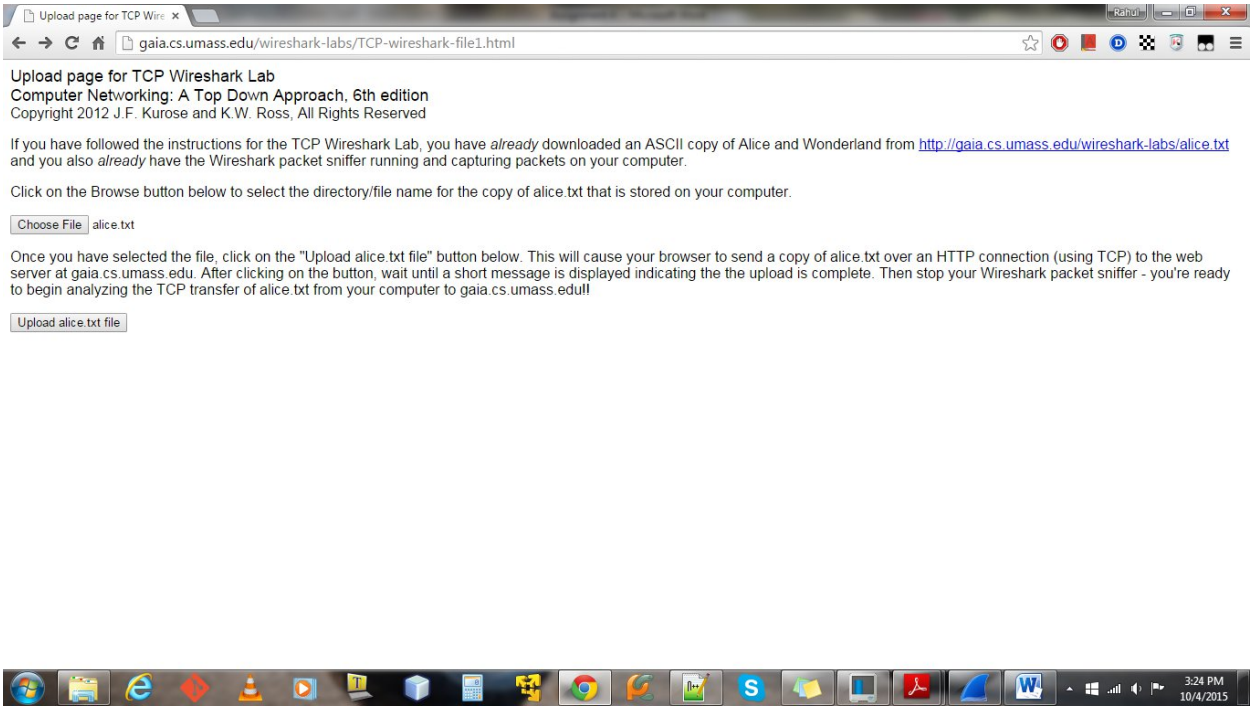
Assignment 4 - Wireshark Lab: TCP v6.0

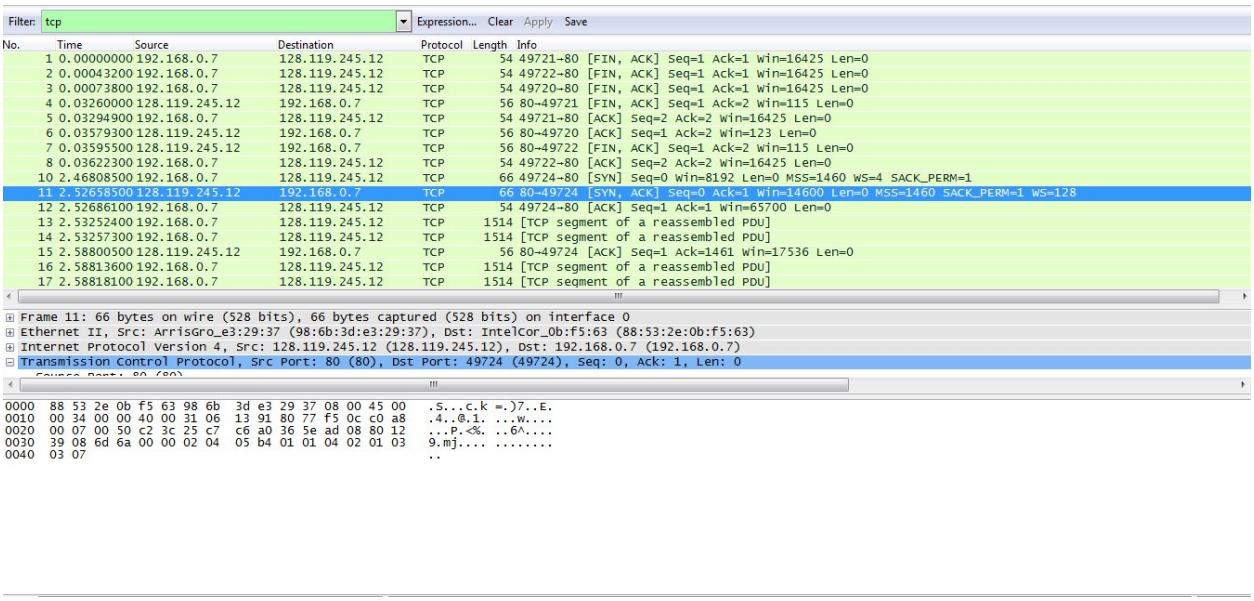
Answer 1: Capturing a bulk TCP transfer from your computer to a remote server



1a. Browser Output



1b. Browser Output



1c. Wireshark Output

2. A first look at the captured trace

1. The IP address of the client is 192.168.1.102 and the Source Port is – 1161

```
    199 5.297341    192.168.1.102        128.119.245.12       HTTP     104    POST /ethereal-labs/lab3-1-reply
.htm HTTP/1.1  (text/plain)
```

```
Frame 199: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
Ethernet II, Src: Actionte_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 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 164041, Ack: 1, Len: 50
    Source Port: 1161 (1161)
    Destination Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 50]
    Sequence number: 164041    (relative sequence number)
    [Next sequence number: 164091    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
```

2. The IP address of gaia.cs.umass.edu is 128.119.245.12 and the port on which it's sending and receiving port is 80 (default)

```
    203 5.461175    128.119.245.12       192.168.1.102        HTTP     784    HTTP/1.1 200 OK  (text/html)
```

```
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 1161 (1161), Seq: 1, Ack: 164091, Len: 730
    Source Port: 80 (80)
    Destination Port: 1161 (1161)
    [Stream index: 0]
    [TCP Segment Len: 730]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 731    (relative sequence number)]
    Acknowledgment number: 164091    (relative ack number)
```

3. The IP address of client computer is 192.168.0.7 and port number is 49724

```
    153 2.787663000    192.168.0.7          128.119.245.12       HTTP     1288   POST /wireshark-labs/lab3-1-r
eply.htm HTTP/1.1  (text/plain)
```

```
Transmission Control Protocol, Src Port: 49724 (49724), Dst Port: 80 (80), Seq: 151741, Ack: 1, Len: 1234
    Source Port: 49724 (49724)
    Destination Port: 80 (80)
    [Stream index: 3]
    [TCP Segment Len: 1234]
    Sequence number: 151741    (relative sequence number)
    [Next sequence number: 152975    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
```

3. TCP Basics

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. The segment has the SYN flag bit set (1<<1) that identifies it as a SYN segment.

```
No.     Time           Source                Destination           Protocol Length Info
     10 2.468085000    192.168.0.7           128.119.245.12        TCP      66     49724   80 [SYN] Seq=0 Win=81
92 Len=0 MSS=1460 WS=4 SACK_PERM=1

Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: IntelCor_0b:f5:63 (88:53:2e:0b:f5:63), Dst: ArrisGro_e3:29:37 (98:6b:3d:e3:29:37)
Internet Protocol Version 4, Src: 192.168.0.7 (192.168.0.7), Dst: 128.119.245.12 (128.119.245.12)
Transmission Control Protocol, Src Port: 49724 (49724), Dst Port: 80 (80), Seq: 0, Len: 0
```

```
.... 0000 0000 0010 = Flags: 0x002 (SYN)
```

5. 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 acknowledge field in the SYNACK segment is = 1. It's determined by the sequence number of the next byte expected from the client computer. The segment has the SYN (1<<1) and ACK(1<<4) bit set in the flags field that identifies it as a SYN ACK segment.

```
No.     Time           Source                Destination           Protocol Length Info
     11 2.526585000    128.119.245.12        192.168.0.7           TCP      66     80    49724 [SYN, ACK] Seq=0 A
ck=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128

Frame 11: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: ArrisGro_e3:29:37 (98:6b:3d:e3:29:37), Dst: IntelCor_0b:f5:63 (88:53:2e:0b:f5:63)
Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.0.7 (192.168.0.7)
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 49724 (49724), Seq: 0, Ack: 1, Len: 0
```

```
.... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
```

6. The sequence number of the TCP segment containing the HTTP POST command is =1.

```
No.     Time           Source                Destination           Protocol Length Info
     13 2.532524000    192.168.0.7           128.119.245.12        TCP      1514   49724   80 [ACK] Seq=1 Ack=1
Win=65700 Len=1460

Frame 13: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: IntelCor 0b:f5:63 (88:53:2e:0b:f5:63), Dst: ArrisGro e3:29:37 (98:6b:3d:e3:29:37)
Internet Protocol Version 4, Src: 192.168.0.7 (192.168.0.7), Dst: 128.119.245.12 (128.119.245.12)
Transmission Control Protocol, Src Port: 49724 (49724), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 1460
Data (1460 bytes)

0000  98 6b 3d e3 29 37 88 53 2e 0b f5 63 08 00 45 00   .k=.)7.S...c..E.
0010  05 dc 1a c6 40 00 80 06 a4 22 c0 a8 00 07 80 77   ....@....".....w
0020  f5 0c c2 3c 00 50 36 5e ad 08 25 c7 c6 a1 50 10   ...<.P6^..%...P.
0030  40 29 bf 1f 00 00 50 4f 53 54 20 2f 77 69 72 65   @)....POST /wire
0040  73 68 61 72 6b 2d 6c 61 62 73 2f 6c 61 62 33 2d   shark-labs/lab3-
```

7. Packet trace taken from author from here onwards

Below table shows the sequence numbers, time sent and time ACK was received

```
  4 2004-08-21 13:44:20.596858  192.168.1.102  128.119.245.12  TCP  619 1161→80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
  5 2004-08-21 13:44:20.612118  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
  7 2004-08-21 13:44:20.624407  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
  8 2004-08-21 13:44:20.625071  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
 10 2004-08-21 13:44:20.647786  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
 11 2004-08-21 13:44:20.648538  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
```

TCP Segments Received

```
  6 2004-08-21 13:44:20.624318  128.119.245.12  192.168.1.102  TCP  60 80→1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
  9 2004-08-21 13:44:20.647675  128.119.245.12  192.168.1.102  TCP  60 80→1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
 12 2004-08-21 13:44:20.694466  128.119.245.12  192.168.1.102  TCP  60 80→1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
 14 2004-08-21 13:44:20.739499  128.119.245.12  192.168.1.102  TCP  60 80→1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
 15 2004-08-21 13:44:20.787680  128.119.245.12  192.168.1.102  TCP  60 80→1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
 16 2004-08-21 13:44:20.838183  128.119.245.12  192.168.1.102  TCP  60 80→1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
```

Sequence number	Time Segment was sent	Time ACK was received	Sample RTT(Micros)	Estimated RTT(Micros)	Estimated RTT(seconds)
1	13:44:20.596858	13:44:20.624318	27460	27460	0.027460
566	13:44:20.612118	13:44:20.647675	35557	28472.1	0.028472
2026	13:44:20.624407	13:44:20.694466	70059	33670.4	0.033670
3486	13:44:20.625071	13:44:20.739499	114428	43765.1	0.043765
4946	13:44:20.647786	13:44:20.787680	139894	55781.2	0.055781
6406	13:44:20.648538	13:44:20.838183	189645	72514.1	0.072514

8. The length of first Six TCP Segments is 565, 1460, 1460, 1460, 1460, and 1460

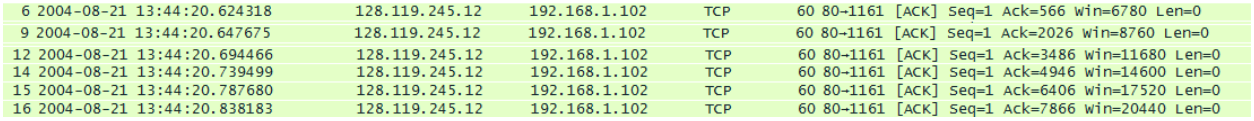
```
  4 2004-08-21 13:44:20.596858  192.168.1.102  128.119.245.12  TCP  619 1161→80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
  5 2004-08-21 13:44:20.612118  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
  7 2004-08-21 13:44:20.624407  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
  8 2004-08-21 13:44:20.625071  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
 10 2004-08-21 13:44:20.647786  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
 11 2004-08-21 13:44:20.648538  192.168.1.102  128.119.245.12  TCP 1514 1161→80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
```

9. The receive window is conveyed to the client by the server during acknowledgement and that is =5840 Bytes. As per TCP flow control implementation as long as the server sends a receive window to client that it can process without dropping and the client makes sure that LastByteSent – LastByteAcked <= Window, there should not be any throttling at client side.

```
  2 2004-08-21 13:44:20.593553  128.119.245.12  192.168.1.102  TCP  62 80→1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
```

10. In the Authors Packet trace there are no re-transmitted segments. It can be seen on Wireshark where info is TCP Retransmission and the number of tries will be performed based on the timeout value. It can also be seen by looking at the sequence number of segment. Two or more segments present in the trace with the same sequence number and spaced between the timeout indicate a retransmission.

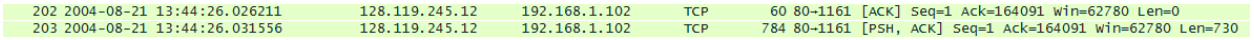
11. The amount of data ACK'ed by Server is known by which is the expected Byte value or sequence number expected from the sending host. Considering these 6 segments and subtracting each sequence will give the amount of data ACK'ed by server.



6	566 - 1 = 565(1 byte for SYN ACK)
9	2026 - 566 = 1460
12	3486 - 2026 =1460
14	4946 – 3486 = 1460
15	6406 – 4946 = 1460
16	7866 – 6406 = 1460

On an average it is 1460 bytes

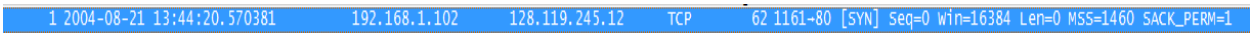
Yes there can be a scenario where receiver is ACK'ing every other received segment. Here the server is ACK'ing for sequence number (164091) twice. The reason could be that there is no more data from client to send .



12. Bytes transferred per unit time for connection = Total bytes transferred for connection/Time from start to end of connection (seconds)

Since the number of bytes can be thought as the last sequence number that is ACK'ed by server and from the Packet trace, it is 164091

Since the first sequence is 1 (for TCP SYN),



This means 164091 bytes were transferred and time for this transfer is 13:44:26.031556 – 13:44:20.570381(SYN and Last ACK) = 5.461175 seconds

164091/5.461175 = 30046.83 Bytes/sec is the throughput for this connection

4. TCP congestion control in action.

13, 14. (13 and 14 answered here both the questions)

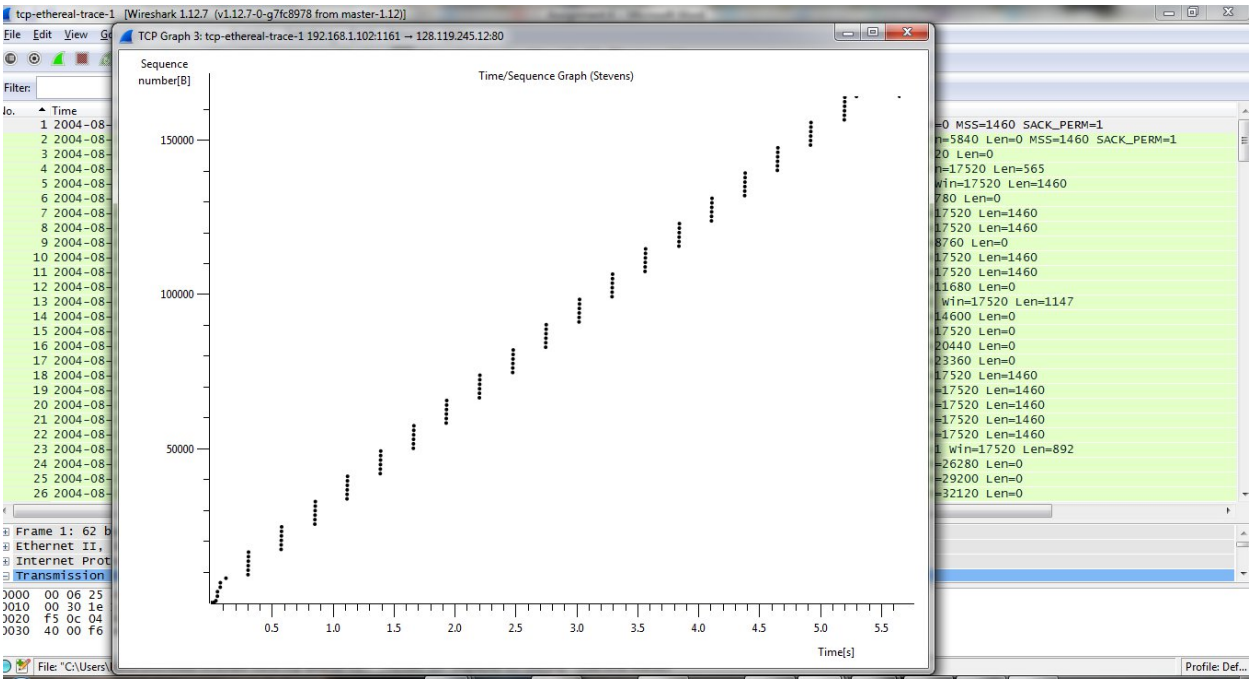


Fig 4.1 Wireshark Time-Sequence-Graph (Stevens)

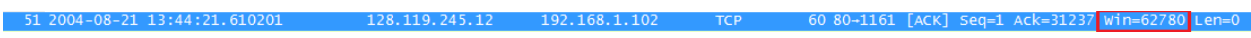
After analyzing the Time-Sequence-Graph (Stevens), these are my observations:

1. The TCP slow start phase is during the initial sequence of 3 way handshake as we can see clearly from the graph that segments are not overlapping which means there is time taken for the TCP stack at either end systems to initialize .

2. But as the TCP segments between the hosts are exchanged during the 3-way handshake, the window size is clearly understood by both the hosts and thereby, the segments after this are sent uniformly (6 segments at a given time).

From the 3-way hand shake segments, it can be seen that the server has MSS = 1460 and the client has MSS = 1460.

From the graph and packet list, we can see that as time advances the window size of server increases and towards the end it stabilizes to 62780(bytes)



At this point the time elapsed is 21.610201 – 20.570381 is 1.039 at which the window size of server stabilizes and the client has agreed to this and send's maximum 6 segments at a time.

1460*6 = 8760 and 8760 <=62780 hence there is no need of flow control in this case since the client is not overwhelming the server with lot of data and there is no congestion.

To really understand if there is congestion, we have to check if server changes window size substantially less than the buffer size at client and also check on wireshark if any segments are lost at receiving end.

This is slightly different from the text, since when the window size is 62780 for server, then the client can send 43 (1460*43 =62780) segments thereby reducing the time for transmission to almost 2 seconds including handshake. However TCP implementation in practice can vary across systems based on buffer sizes and assumptions made for all kinds of networks.