



Name: Abhijit Paul

ID:1201

Lab: WireShark TCP

1. IP address and TCP port number used by the client computer:

Source IP: 10.100.106.14

Source Port: 47430, 47428

We found two port numbers for the first two SYN segments. By looking at the code, we are able to determine that port 47428 communicates with port 80 later.

3	0.128816841	10.100.106.14	128.119.245.12	TCP	74 47428 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=846303048 TSecr=0 WS=128
10	0.379204982	10.100.106.14	128.119.245.12	TCP	74 47430 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=846303298 TSecr=0 WS=128
13	0.478203160	128.119.245.12	10.100.106.14	TCP	74 80 → 47428 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=4105969998 TSecr=...
14	0.478276996	10.100.106.14	128.119.245.12	TCP	66 47428 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=846303397 TSecr=4105969998

2. The IP address and port of gaia.cs.umass.edu:

Destination IP: 128.119.245.12

Destination Port: 80

3	0.128816841	10.100.106.14	128.119.245.12	TCP	74 47428 → 80 [SYN] Seq=0 Win=64240 Len=0
10	0.379204982	10.100.106.14	128.119.245.12	TCP	74 47430 → 80 [SYN] Seq=0 Win=64240 Len=0
13	0.478203160	128.119.245.12	10.100.106.14	TCP	74 80 → 47428 [SYN, ACK] Seq=0 Ack=1 Win=2
14	0.478276996	10.100.106.14	128.119.245.12	TCP	66 47428 → 80 [ACK] Seq=1 Ack=1 Win=64256

3. The IP address and TCP port number used by the client computer (source) to transfer the file to gaia.cs.umass.edu is:

128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK] Seq=1 Ack=87445 Win=183296 Len=0 TSval=4105971068
10.100.106.14	128.119.245.12	HTTP	927 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK] Seq=1 Ack=88833 Win=183296 Len=0 TSval=4105971069

▶ Internet Protocol Version 4, Src: 10.100.106.14, Dst: 128.119.245.12
▼ Transmission Control Protocol, Src Port: 47428, Dst Port: 80, Seq: 148517, Ack: 1, Len: 861
Source Port: 47428
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 861]
Sequence number: 148517 (relative sequence number)
[Next sequence number: 149378 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)

Client IP: 10.100.106.14

Client Port: 47428

4. The sequence number of the TCP SYN segment:

▼ Transmission Control Protocol, Src Port: 47428, Dst Port: 80, Seq: 0, Len: 0
Source Port: 47428
Destination Port: 80
[Stream index: 0]

SYN segment number = 0

```

▼ 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

```

SYN FLAG is set(1).

5. Sequence number of the SYNACK segment is = 0

10.100.106.14	128.119.245.12	TCP	74 47430 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=846303298 TS
128.119.245.12	10.100.106.14	TCP	74 80 → 47428 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1400 SACK_PERM=1 TSval=4
10.100.106.14	128.119.245.12	TCP	66 47428 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=846303397 TSecr=4105969998

The value of the Acknowledgement field in the SYNACK segment = 1

```

▼ Flags: 0x012 (SYN, ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR)
  .... .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.]

```

How did gaia.cs.umass.edu determine that value of SYNACK, ACK Flag?

Answer: The server understood that from the previous SYN TCP segment. Under TCP protocol 3-way handshake, the destination returns SYNACK when it receives SYN and wishes to establish the connection.

What is it in the segment that identifies the segment as a SYNACK segment?

SYN flag = 1

ACK flag = 1

6. What is the sequence number of the TCP segment containing the HTTP POST command?

Sequence Number = 1

```

▶ Internet Protocol Version 4, Src: 10.100.106.14, Dst: 128.119.245.12
▼ Transmission Control Protocol, Src Port: 47428, Dst Port: 80, Seq: 1, Ack: 1, Len: 1388
  Source Port: 47428
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 1388]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 1389 (relative sequence number)]

0020 f5 0c b9 44 00 50 17 b8 5b 29 c6 5c 81 df 80 10 ...D.P...[]\.....
0030 01 f6 bb c1 00 00 01 01 08 0a 32 71 90 a6 f4 bc .....2q....
0040 21 4e 50 4f 53 54 20 2f 77 69 72 65 73 68 61 72 INPOST / wireshar
0050 0b 2d 6c 61 62 73 2f 6c 61 62 33 2d 31 2d 72 65 k-labs/1 ab3-1-re
0060 70 6c 79 2e 68 74 6d 20 48 54 54 50 2f 31 2e 31 ply.htm HTTP/1.1
0070 0d 0a 48 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e ..Host: gaia.cs.
0080 75 6d 61 73 73 2e 65 64 75 0d 0a 55 73 65 72 2d umass.ed u..User-
0090 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 Agent: Mozilla/5
00a0 2e 30 20 28 58 31 31 3b 20 55 62 75 6e 74 75 3b .0 (X11; Ubuntu;
00b0 20 4c 69 6e 75 78 20 78 38 36 5f 36 34 3b 20 72 Linux x86_64; rv:
00c0 76 3a 39 38 2e 30 29 20 47 65 63 6b 6f 2f 32 30 v:98.0) Gecko/20
00d0 31 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f 39 100101 Firefox/9
00e0 38 2e 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 8.0..Accept: text/
00f0 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 t/html,application
0100 0f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 on/xhtml+xml,application
0110 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 lication/xml;q=0.
0120 2e 39 2c 69 6d 61 67 65 2f 61 76 69 66 2c 69 6d .9,image/avif,image

```

7. If TCP POST Seq = 0

```

▶ Frame 187: 927 bytes on wire (7416 bits), 927 bytes captured (7416 bits) on interface 0
▶ Ethernet II, Src: Dell_c2:f9:cf (8c:ec:4b:c2:f9:cf), Dst: Routerbo_c7:55:df (6c:3b:6b:c7:55:df)
▶ Internet Protocol Version 4, Src: 10.100.106.14, Dst: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 47428, Dst Port: 80, Seq: 148517, Ack: 1, Len: 861
▼ [108 Reassembled TCP Segments (149377 bytes): #15(1388), #16(1388), #17(1388), #18(1388), #19(1388), #20(1388), #21(1388), #22(1388)
  [Frame 15, payload: 0-1387 (1388 bytes)]
  [Frame 16, payload: 1388-2775 (1388 bytes)]
  [Frame 17, payload: 2776-4163 (1388 bytes)]
  [Frame 18, payload: 4164-5551 (1388 bytes)]
  [Frame 19, payload: 5552-6939 (1388 bytes)]
  [Frame 20, payload: 6940-8327 (1388 bytes)]
  [Frame 21, payload: 8328-9715 (1388 bytes)]
  [Frame 22, payload: 9716-11103 (1388 bytes)]
  [Frame 23, payload: 11104-12491 (1388 bytes)]
  [Frame 24, payload: 12492-13879 (1388 bytes)]

00000000 50 4f 53 54 20 2f 77 69 72 65 73 68 61 72 6b 2d POST /wireshark-
00000010 6c 61 62 73 2f 6c 61 62 33 2d 31 2d 72 65 70 6c labs/lab 3-1-repl
00000020 79 2e 68 74 6d 20 48 54 54 50 2f 31 2e 31 0d 0a y.htm HT TP/1.1..
00000030 48 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75 6d Host: ga ia.cs.um
00000040 61 73 73 2e 65 64 75 0d 0a 55 73 65 72 2d 41 67 ass.edu..User-Ag
00000050 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 ent: Mozilla/5.0
00000060 20 28 58 31 31 3b 20 55 62 75 6e 74 75 3b 20 4c (X11; U buntu; L
00000070 69 6e 75 78 20 78 38 36 5f 36 34 3b 20 72 76 3a inux x86 _64; rv:
00000080 39 38 2e 30 29 20 47 65 63 6b 6f 2f 32 30 31 30 98.0) Ge cko/2010
00000090 30 31 30 31 20 46 69 72 65 66 6f 78 2f 39 38 2e 0101 Fir efox/98.
000000a0 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 2f 0..Accep t: text/
000000b0 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e html,application
000000c0 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 /xhtml+xml,appli
000000d0 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e 39 cation/x ml;q=0.9
000000e0 2c 69 6d 61 67 65 2f 61 76 69 66 2c 69 6d 61 67 ,image/a vif,imag
000000f0 65 2f 77 65 62 70 2c 2a 2f 2a 3b 71 3d 30 2e 38 e/webp,* /*;q=0.8

```

What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)?

tcp.ack == 2777

Sequence Number	Sent(seconds)	ACK Received	RTT
0	0.478668828	0.828767161	0.350098333
1389	0.478683740	0.828846947	0.350163207

2777	0.478788173	0.828856711	0.350068538
4165	0.478795434	0.829377634	0.3505822
5553	0.478973377	0.829418433	0.350445056
6941	0.478987045	0.829426628	0.350439583

Tcp.ack == 1 is sent time

Tcp.seq == 1 is receive time

27	0.828767161	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=1389 Win=31872
30	0.828846947	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=2777 Win=34816
31	0.828856711	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=4165 Win=37760
36	0.829377634	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=5553 Win=40576
39	0.829418433	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=6941 Win=43520
40	0.829426628	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=8329 Win=46336
41	0.829434022	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=9717 Win=49280
42	0.829448849	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=11105 Win=52224
45	0.829533307	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=12493 Win=55040
47	0.829697634	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=13881 Win=57984
57	1.104277070	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=15269 Win=60928

Estimated RTT:

Assuming, $\alpha=0.125$ and EstimatedRTT for segment 1 = 0.350098333

EstimatedRTT for segment 2 = $0.875 \cdot 0.350098333 + 0.125 \cdot 0.350163207 = 0.35010644225$

EstimatedRTT for segment 3 =

$0.875 \cdot 0.35010644225 + 0.125 \cdot 0.350068538 = 0.35010170421$

EstimatedRTT for segment 4 =

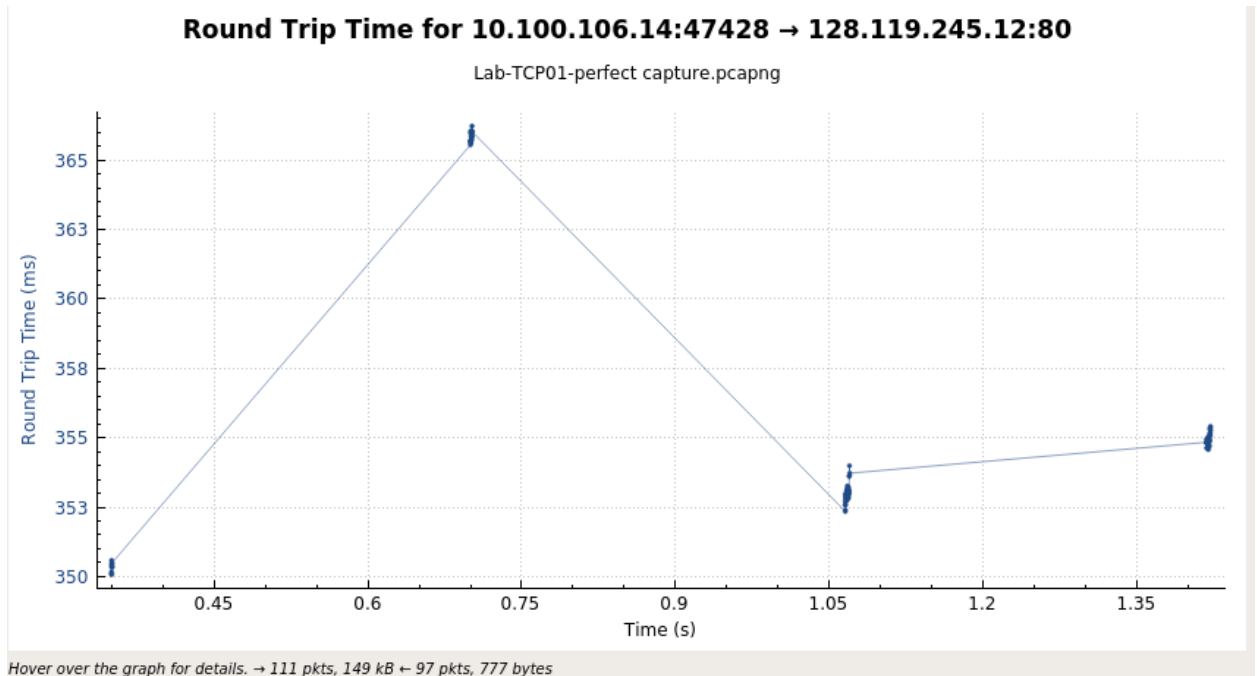
$0.875 \cdot 0.35010170421 + 0.125 \cdot 0.3505822 = 0.35016176618$

EstimatedRTT for segment 5 =

$0.875 \cdot 0.35016176618 + 0.125 \cdot 0.350445056 = 0.3501971774$

EstimatedRTT for segment 6 =

$0.875 \cdot 0.3501971774 + 0.125 \cdot 0.350439583 = 0.3502274781$



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

1388 bytes

9. What is the minimum amount of available buffer space advertised at the received for the entire trace?

1499 bytes

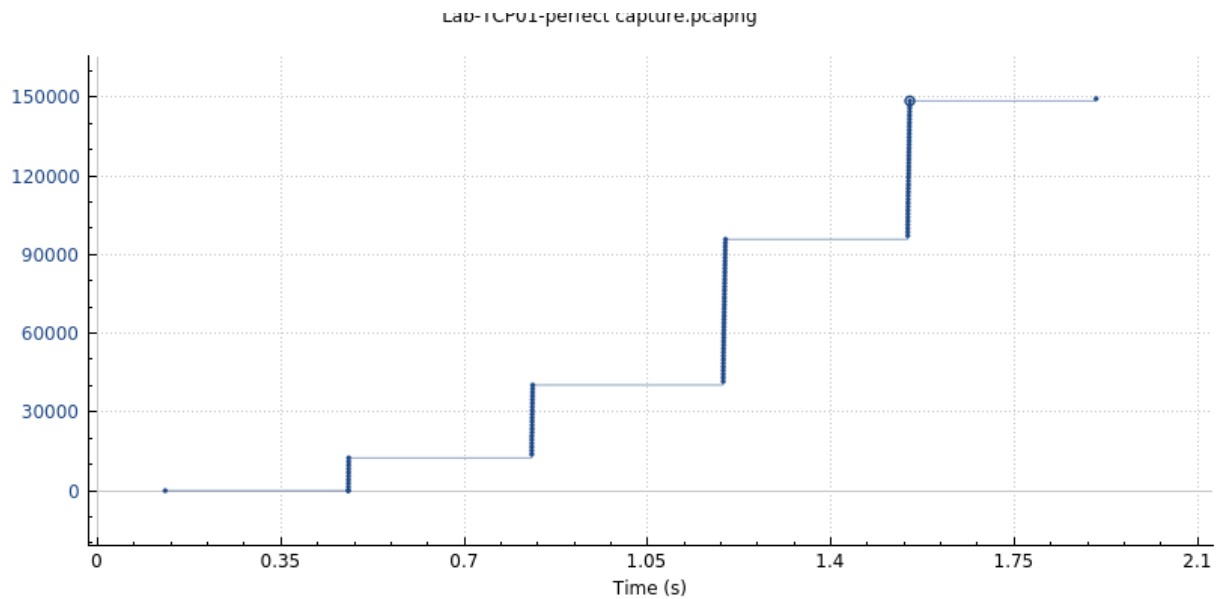
```

[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 124921 (relative ack number)
1000 .... = Header Length: 32 bytes (8)
▶ Flags: 0x010 (ACK)
Window size value: 1499
[Calculated window size: 191872]
[Window size scaling factor: 128]
Checksum: 0x4762 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
▶ [SEQ/ACK analysis]
▶ [Timestamps]

```

Does the lack of receiver buffer space ever throttle the Sender?

No, there was no packet loss as we can see in the graph. The congestion control algorithm has increased the congestion-window size over the whole time.



10. Are there any retransmitted segments in the trace file?
What did you check for (in the trace) in order to answer this question?

No, no retransmission happened. From the above graph, we can be sure that no package loss occurred. I checked the graph for it.

11. How much data does the receiver typically acknowledge in an ACK?

```

Internet Protocol Version 4, Src: 10.10.10.14, Dst: 10.10.10.12
Transmission Control Protocol, Src Port: 47428, Dst Port: 80, Seq: 11105, Ack: 1, Len: 1388
  Source Port: 47428
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 1388]
  Sequence number: 11105 (relative sequence number)
  [Next sequence number: 12493 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x010 (ACK)

```

The segment size is 1388 bytes.

12. What is the throughput (bytes transferred per unit time) for the TCP connection?

Total bytes sent = last ACK from server - 1 = 149378 - 1 = 149377

Total time = last ACK time - first Segment time = 1.906035922 - 0.128816841

So throughput = Total bytes sent / Total time = 84050.9769431 bytes/second

215	1.905331223	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=142965 Win=229632 Len=0 TSval=4105971421 TSecr=846304469
216	1.905338335	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=144353 Win=232448 Len=0 TSval=4105971421 TSecr=846304469
217	1.905450910	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=145741 Win=235392 Len=0 TSval=4105971421 TSecr=846304469
218	1.905881865	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=147129 Win=238208 Len=0 TSval=4105971421 TSecr=846304470
219	1.906085922	128.119.245.12	10.100.106.14	TCP	66 80 → 47428 [ACK]	Seq=1 Ack=149378 Win=242816 Len=0 TSval=4105971421 TSecr=846304470
220	1.906480109	128.119.245.12	10.100.106.14	HTTP	843 HTTP/1.1 200 OK (text/html)	
221	1.906512019	10.100.106.14	128.119.245.12	TCP	66 47428 → 80 [ACK]	Seq=149378 Ack=778 Win=64128 Len=0 TSval=846304826 TSecr=4105971422

13-14. Congestion - Slow Start and Avoidance state

From the graph, we can see that it never had to go to congestion avoidance state. It remained in slow start state the whole time.

