LAB 3

COURSE: COMPUTER NETWORK

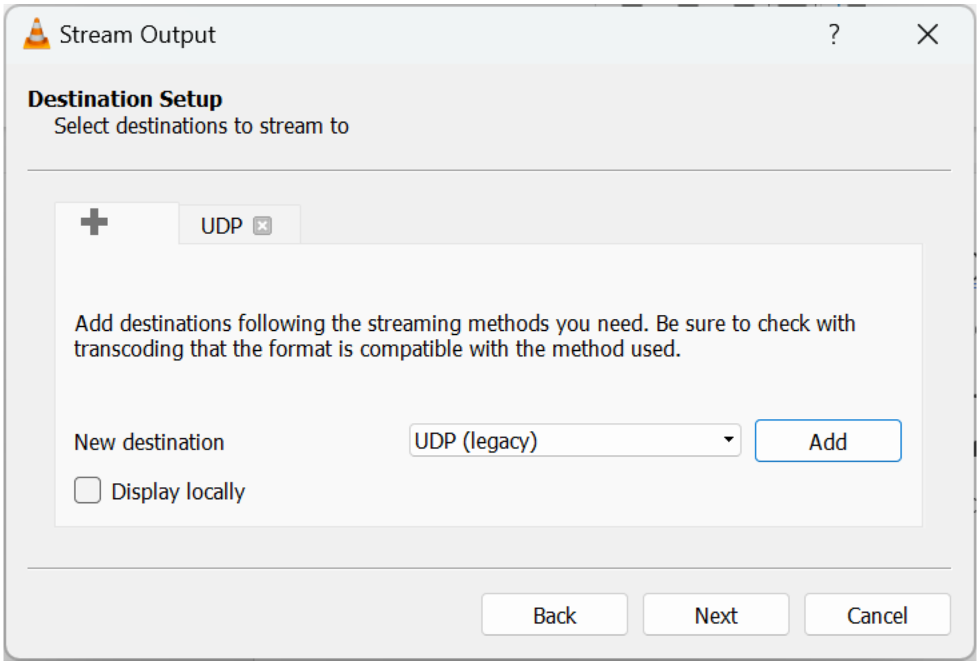
TCP AND UDP

1 Objective

- Stream video between 2 devices with VLC

2 Streaming Video using UDP

- Download and install VLC into student’s computer (as stream server) and install VLC on smartphone (as client). https://www.videolan.org/
- Stream a video from server using UDP protocol: Media -> Stream -> Choose file (Choose a short video ~ 10 seconds)
- Choose destination as UDP -> Add



Destination Setup

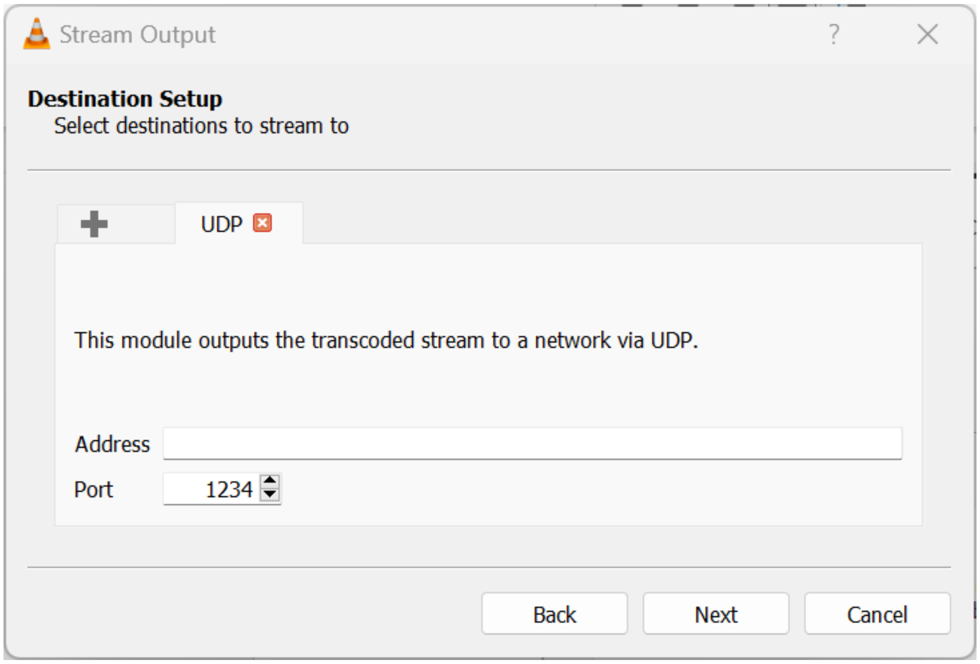
Select destinations to stream to

Add destinations following the streaming methods you need. Be sure to check with transcoding that the format is compatible with the method used.

New destination: UDP (legacy)

□ Display locally

- Add the IP of the smartphone (as client)



This module outputs the transcoded stream to a network via UDP.

Address

Port 1234

- Open Wireshark and capture.
- Client gets video from server: Network -> Open Network Stream -> Enter the URL: udp://@:1234/
- After the video is finished, stop Wireshark and save the capture with name MSSV-udp.pcap

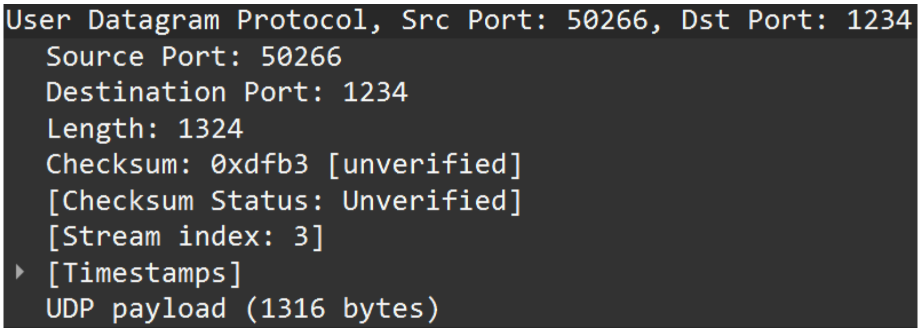
3 Streaming Video using TCP

- Stream a video from server using HTTP protocol: Media -> Stream -> Choose file -> Choose destination as HTTP -> Add -> Next -> Stream.
- Open wireshark and capture.
- Client gets video from server: Network -> Open Network Stream -> Enter the URL: http://IPofServer:8080/
- After the video is finished, stop wireshark and save the capture with name MSSV-http.pcap

4 Analyze UDP

- Open file MSSV-udp.pcap.
- Answer the following questions and capture the screens to verify the answers.

1. Using filter to filter the “udp” packet. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header, name these field.

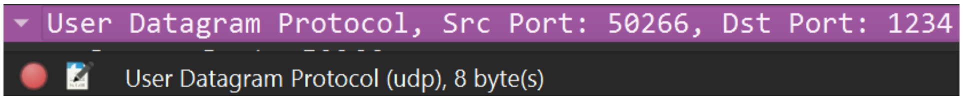


```
User Datagram Protocol, Src Port: 50266, Dst Port: 1234
  Source Port: 50266
  Destination Port: 1234
  Length: 1324
  Checksum: 0xdfb3 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 3]
  [Timestamps]
  UDP payload (1316 bytes)
```

There are 5 fields total in the UDP header:

- **Source Port: 50266**
- **Destination Port: 1234**
- **Length: 1324**
- **Checksum: 0xdfb3**
- **UDP payload: 1316 bytes**

2. By consulting the displayed information in Wireshark’s packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.?

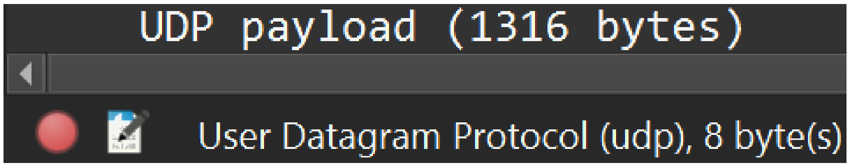


```
User Datagram Protocol, Src Port: 50266, Dst Port: 1234
User Datagram Protocol (udp), 8 byte(s)
```

Each UDP header has a length of 8 bytes.

3. The value in the **Length** field is the length of what? Verify your claim with your captured UDP packet.

The length of UDP packet is the sum of UDP header’s length and UDP payload.



```
UDP payload (1316 bytes)
User Datagram Protocol (udp), 8 byte(s)
```

4. What is the maximum number of bytes that can be included in a UDP payload?

(Hint: the answer to this question can be determined by your answer to 2. above)

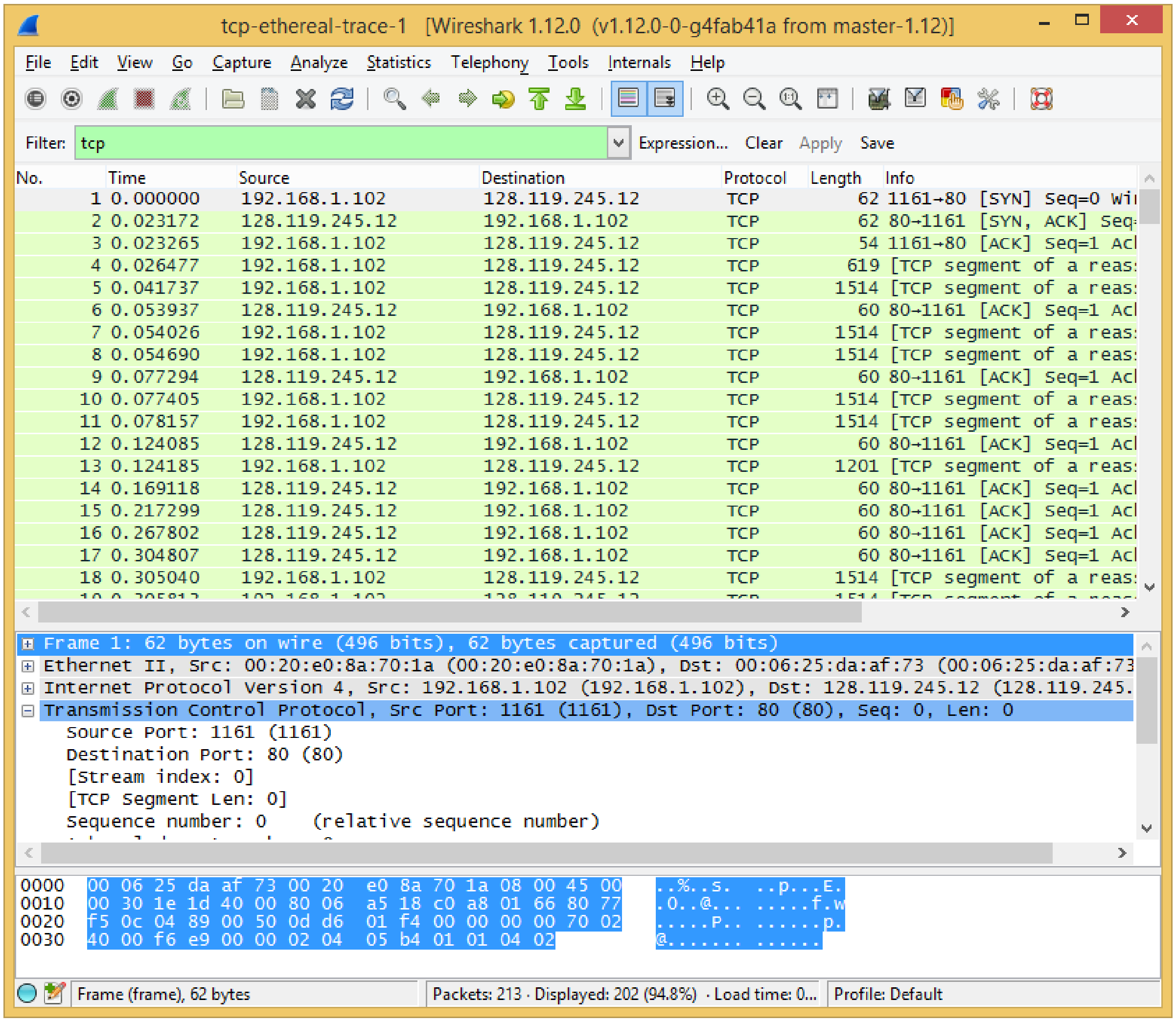
The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) bytes plus the header bytes. This gives 65535 bytes – 8 bytes = 65527 bytes.

5. What is the largest possible source port number? (Hint: see the hint in 4.)

The largest possible source port number is (2^16 – 1) = 65535.

5 Analyze TCP

- Open file MSSV-http.pcap
- Type “tcp” into the display filter specification window towards the top of the Wireshark window to filter the TCP packet.


tcp-ethereal-trace-1 [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: tcp

```
No.     Time      Source          Destination     Protocol Length Info
      1 0.000000  192.168.1.102   128.119.245.12  TCP        62 1161→80 [SYN] Seq=0 Wi
      2 0.023172  128.119.245.12  192.168.1.102   TCP        62 80→1161 [SYN, ACK] Seq
      3 0.023265  192.168.1.102   128.119.245.12  TCP        54 1161→80 [ACK] Seq=1 Ac
      4 0.026477  192.168.1.102   128.119.245.12  TCP       619 [TCP segment of a reas
      5 0.041737  192.168.1.102   128.119.245.12  TCP      1514 [TCP segment of a reas
      6 0.053937  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
      7 0.054026  192.168.1.102   128.119.245.12  TCP      1514 [TCP segment of a reas
      8 0.054690  192.168.1.102   128.119.245.12  TCP      1514 [TCP segment of a reas
      9 0.077294  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
     10 0.077405  192.168.1.102   128.119.245.12  TCP      1514 [TCP segment of a reas
     11 0.078157  192.168.1.102   128.119.245.12  TCP      1514 [TCP segment of a reas
     12 0.124085  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
     13 0.124185  192.168.1.102   128.119.245.12  TCP      1201 [TCP segment of a reas
     14 0.169118  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
     15 0.217299  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
     16 0.267802  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
     17 0.304807  128.119.245.12  192.168.1.102   TCP        60 80→1161 [ACK] Seq=1 Ac
     18 0.305040  192.168.1.102   128.119.245.12  TCP      1514 [TCP segment of a reas
```

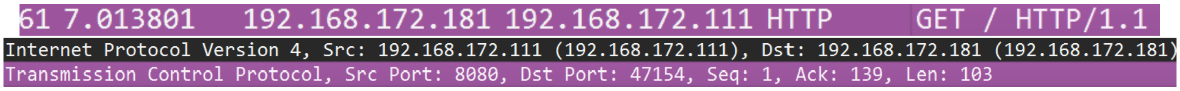
```
⊞ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: 00:20:e0:8a:70:1a (00:20:e0:8a:70:1a), Dst: 00:06:25:da:af:73 (00:06:25:da:af:73
⊞ Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.
⊟ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 0, Len: 0
    Source Port: 1161 (1161)
    Destination Port: 80 (80)
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ..%..s.  ..p...E.
0010  00 30 1e 1d 40 00 80 06  a5 18 c0 a8 01 66 80 77   .0..@...  .....f.w
0020  f5 0c 04 89 00 50 0d d6  01 f4 00 00 00 00 70 02   .....P..  ......p.
0030  40 00 f6 e9 00 00 02 04  05 b4 01 01 04 02         @.......  ......
```

Frame (frame), 62 bytes | Packets: 213 · Displayed: 202 (94.8%) · Load time: 0... | Profile: Default

Answer the following questions and capture the screen to verify the answers.

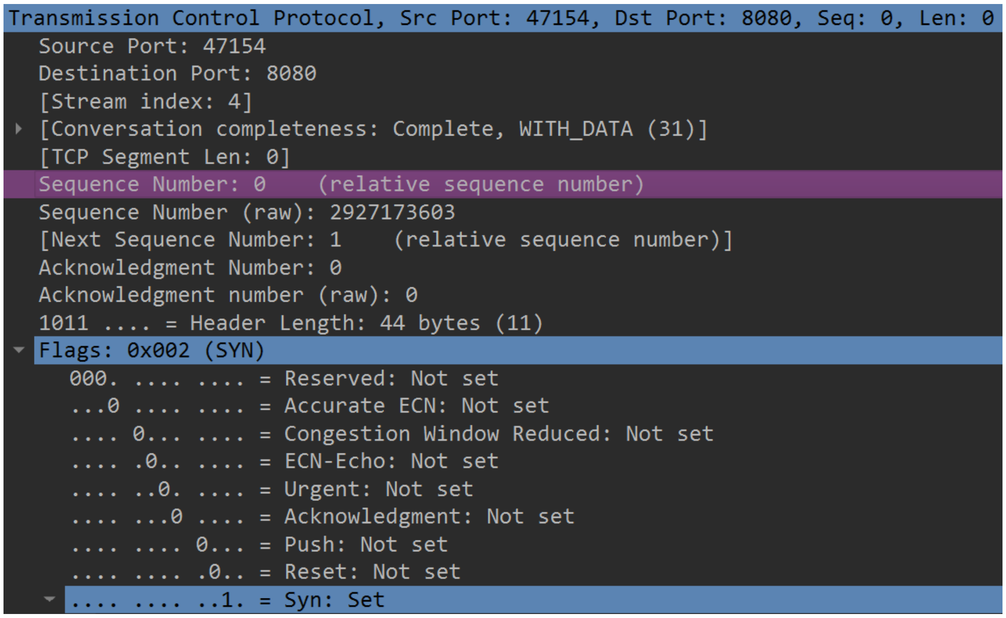
6. What is the IP address and TCP port number used by the client?



```
61 7.013801    192.168.172.181 192.168.172.111 HTTP       GET / HTTP/1.1
Internet Protocol Version 4, Src: 192.168.172.111 (192.168.172.111), Dst: 192.168.172.181 (192.168.172.181)
Transmission Control Protocol, Src Port: 8080, Dst Port: 47154, Seq: 1, Ack: 139, Len: 103
```

The client’s IP address is 192.168.172.181, using TCP port number 47154.

7. What is0020the IP address server? On what port number is it sending and receiving TCP segments for this connection?
   - **Server's IP address: 192.168.172.111.**
   - **Server's TCP port: 8080.**
8. What is the **sequence number** of the TCP SYN segment that is used to initiate the TCP connection between the client and server?



```
Transmission Control Protocol, Src Port: 47154, Dst Port: 8080, Seq: 0, Len: 0
   Source Port: 47154
   Destination Port: 8080
   [Stream index: 4]
 ▸ [Conversation completeness: Complete, WITH_DATA (31)]
   [TCP Segment Len: 0]
   Sequence Number: 0    (relative sequence number)
   Sequence Number (raw): 2927173603
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 0
   Acknowledgment number (raw): 0
   1011 .... = Header Length: 44 bytes (11)
 ▾ Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: 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
```

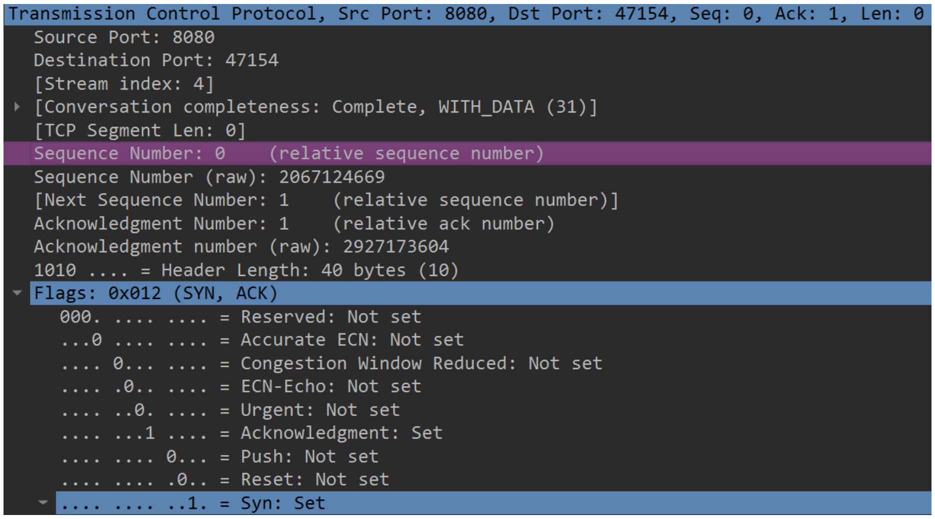
The sequence number of the TCP SYN segment has the value of 2927173603 (raw).

What is it in the segment that identifies the segment as a **SYN segment**?

Hint: Find the field "Flags"

The SYN flag is set to 1 and it indicates that this segment is a SYN segment.

9. What is the **sequence number** of the SYN/ACK segment sent by server to the client computer in reply to the SYN?



```
Transmission Control Protocol, Src Port: 8080, Dst Port: 47154, Seq: 0, Ack: 1, Len: 0
   Source Port: 8080
   Destination Port: 47154
   [Stream index: 4]
 ▸ [Conversation completeness: Complete, WITH_DATA (31)]
   [TCP Segment Len: 0]
   Sequence Number: 0    (relative sequence number)
   Sequence Number (raw): 2067124669
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 2927173604
   1010 .... = Header Length: 40 bytes (10)
 ▾ Flags: 0x012 (SYN, ACK)
     000. .... .... = Reserved: Not set
     ...0 .... .... = Accurate ECN: Not set
     .... 0... .... = Congestion Window Reduced: Not set
     .... .0.. .... = ECN-Echo: Not set
     .... ..0. .... = Urgent: Not set
     .... ...1 .... = Acknowledgment: Set
     .... .... 0... = Push: Not set
     .... .... .0.. = Reset: Not set
   ▾ .... .... ..1. = Syn: Set
```

The sequence number of SYN/ACK segment has the value of 2067124669 (raw).

What is the value of the **Acknowledgement** field in the SYNACK segment?

Value of Acknowledgement field is 2927173604 (raw).

How did the server determine that value?

The value of the Acknowledgement field in the SYN/ACK segment is determined by adding 1 to the initial sequence number of SYN segment from the client computer.

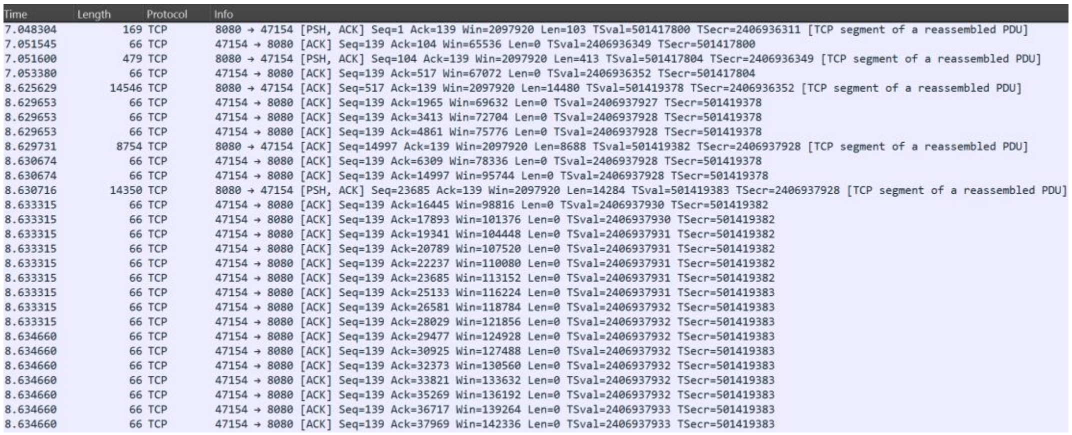
What is it in the segment that identifies the segment as a **SYN/ACK segment**?

The SYN flag and Acknowledgement flag in the segment are set to 1 and they indicate that this segment is a SYNACK segment.

10. Find the first 6 segments that server sent to client.

- What are the sequence numbers and the length of the first six segments?
- 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 (Round trip time) value for each of the six segments?

Student can answer with the following table:



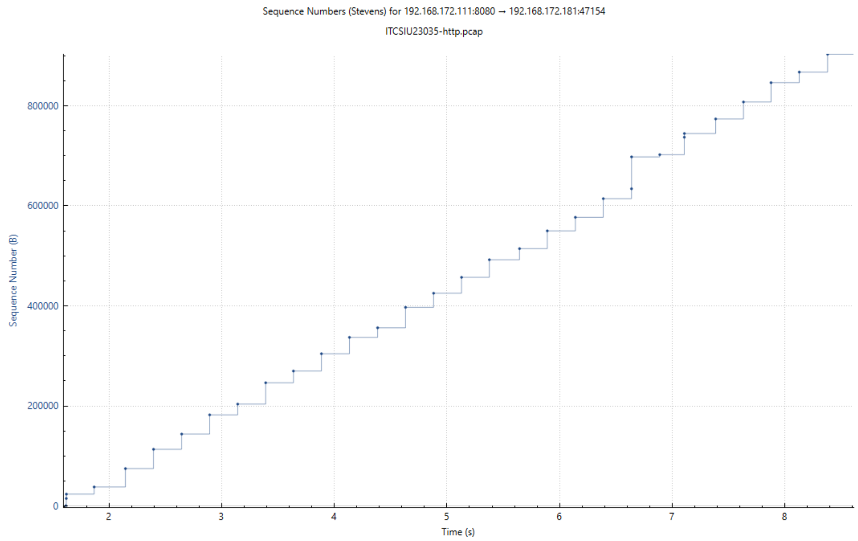
```
Time        Length  Protocol  Info
7.048304       169 TCP       8080 → 47154 [PSH, ACK] Seq=1 Ack=139 Win=2097920 Len=103 TSval=501417800 TSecr=2406936311 [TCP segment of a reassembled PDU]
7.051545        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=104 Win=65536 Len=0 TSval=2406936349 TSecr=501417800
7.051600       479 TCP       8080 → 47154 [PSH, ACK] Seq=104 Ack=139 Win=2097920 Len=413 TSval=501417804 TSecr=2406936349 [TCP segment of a reassembled PDU]
7.053380        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=517 Win=67072 Len=0 TSval=2406936352 TSecr=501417804
8.625629     14546 TCP       8080 → 47154 [ACK] Seq=517 Ack=139 Win=2097920 Len=14480 TSval=501419378 TSecr=2406936352 [TCP segment of a reassembled PDU]
8.629653        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=1965 Win=69632 Len=0 TSval=2406937927 TSecr=501419378
8.629653        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=3413 Win=72704 Len=0 TSval=2406937928 TSecr=501419378
8.629653        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=4861 Win=75776 Len=0 TSval=2406937928 TSecr=501419378
8.629731      8754 TCP       8080 → 47154 [ACK] Seq=14997 Ack=139 Win=2097920 Len=8688 TSval=501419382 TSecr=2406937928 [TCP segment of a reassembled PDU]
8.630674        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=6309 Win=78336 Len=0 TSval=2406937928 TSecr=501419378
8.630674        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=14997 Win=95744 Len=0 TSval=2406937928 TSecr=501419378
8.630716     14350 TCP       8080 → 47154 [PSH, ACK] Seq=23685 Ack=139 Win=2097920 Len=14284 TSval=501419383 TSecr=2406937928 [TCP segment of a reassembled PDU]
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=16445 Win=98816 Len=0 TSval=2406937930 TSecr=501419382
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=17893 Win=101376 Len=0 TSval=2406937930 TSecr=501419382
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=19341 Win=104448 Len=0 TSval=2406937931 TSecr=501419382
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=20789 Win=107520 Len=0 TSval=2406937931 TSecr=501419382
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=22237 Win=110080 Len=0 TSval=2406937931 TSecr=501419382
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=23685 Win=113152 Len=0 TSval=2406937931 TSecr=501419382
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=25133 Win=116224 Len=0 TSval=2406937931 TSecr=501419383
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=26581 Win=118784 Len=0 TSval=2406937932 TSecr=501419383
8.633315        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=28029 Win=121856 Len=0 TSval=2406937932 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=29477 Win=124928 Len=0 TSval=2406937932 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=30925 Win=127488 Len=0 TSval=2406937932 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=32373 Win=130560 Len=0 TSval=2406937932 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=33821 Win=133632 Len=0 TSval=2406937932 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=35269 Win=136192 Len=0 TSval=2406937932 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=36717 Win=139264 Len=0 TSval=2406937933 TSecr=501419383
8.634660        66 TCP       47154 → 8080 [ACK] Seq=139 Ack=37969 Win=142336 Len=0 TSval=2406937933 TSecr=501419383
```

STT	**Sequence number**	**Length (bytes)**	**Sent time (s)**	**Receive ACK time (s)**	**RTT (ms)**
1	1	103	7.048304	*104* - 7.051545	3.241
2	104	413	7.051600	*517* - 7.053380	1.78
3	517	14480	8.625629	*14997* - 8.629653	4.024
4	14997	8688	8.629731	*23685* - 8.630674	0.943
5	23685	14284	8.630716	*37969* - 8.634660	3.944
6	37969	36739	8.877340	*74780* - 8.882472	5.132

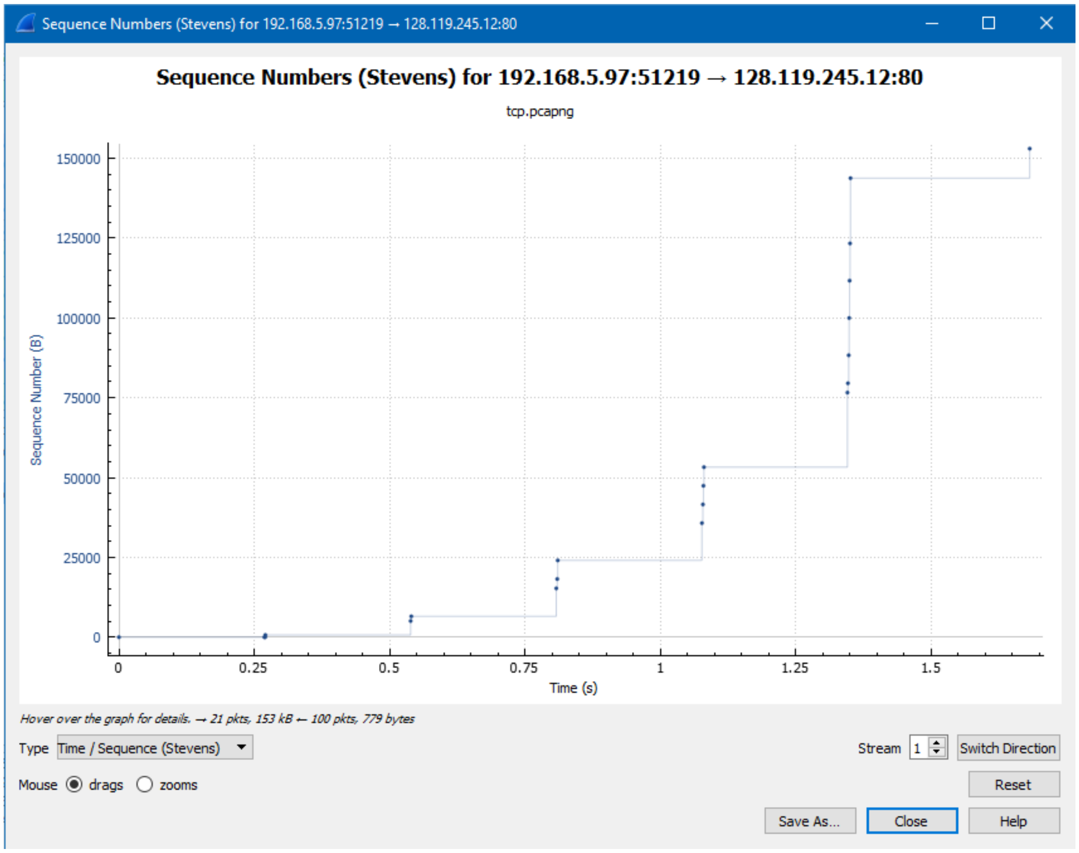
11. What is the minimum amount of available buffer space advertised at the received for the entire trace? Hint: The field Windows size “Win=xxxx”

The smallest amount of available buffer space being received is 65535.

12. Are there any retransmitted segments in the trace file? **No, there are not any retransmitted segments in the trace file. If it is, there will be two or more dots in a period of a sequence number.**



Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to server. Then select: Statistics->TCP Stream Graph -> Time/Sequence (Steven)


Sequence Numbers (Stevens) for 192.168.5.97:51219 → 128.119.245.12:80
Sequence Numbers (Stevens) for 192.168.5.97:51219 → 128.119.245.12:80
tcp.pcapng
Sequence Number (B)
150000
125000
100000
75000
50000
25000
0
0
0.25
0.5
0.75
1
1.25
1.5
Time (s)
Switch Direction
Reset
Save As...
Close
Help

Hover over the graph for details. → 21 pkts, 153 kB ← 100 pkts, 779 bytes

Type Time / Sequence (Stevens)

Stream 1

Mouse ● drags ○ zooms

Sequence Number (Stevens) Graph

Note that: Each dot in the graph represents the sequence number and sending time of a TCP segment. If there are 2 dots with the same sequence number but they are in different time => there are restransmission.