CS 6043/5143: Computer Networking

FALL 2017

PROJECT 2

Given: Oct. 22, 2017

Due: Nov. 3 (Friday), 2017 (NO LATER THAN 11:59PM)

Team Member: Anshul Gautam

Submission Instructions:

1. Submit only on-line files on Blackboard before midnight. No hard copy will be accepted.

2. For students who are working in a team, *one* submission for the team is sufficient.

3. Wireshark files for this project can be found in the zip file "Project_2_Wireshark_Traces.zip".

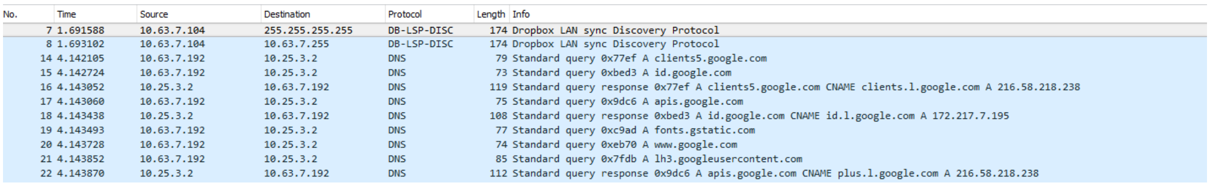
Total possible points: 10

Part I: UDP

Open the file 'UDP_project_2.pcapng' in Wireshark and answer the following questions. Provide screenshots with necessary annotations in each case.

1. (1 pts) Find a UDP packet in the trace file and determine the name and length (in bytes) of each of the UDP header fields.

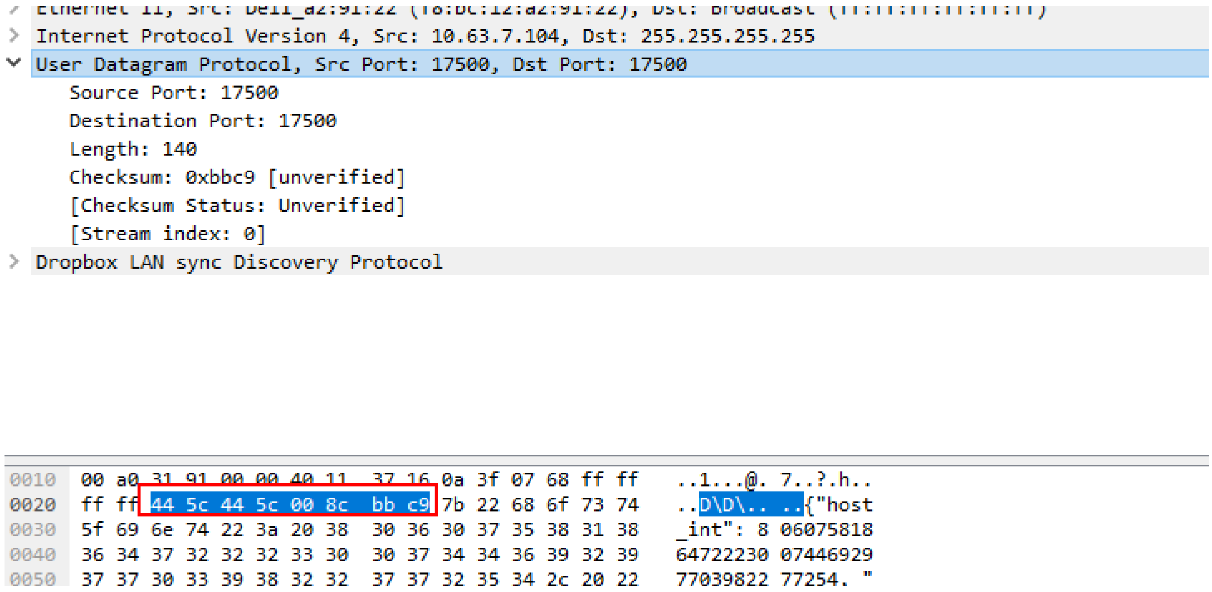
   Answer:



No.	Time	Source	Destination	Protocol	Length	Info
7	1.691588	10.63.7.104	255.255.255.255	DB-LSP-DISC	174	Dropbox LAN sync Discovery Protocol
8	1.693102	10.63.7.104	10.63.7.255	DB-LSP-DISC	174	Dropbox LAN sync Discovery Protocol
14	4.142105	10.63.7.192	10.25.3.2	DNS	79	Standard query 0x77ef A clients5.google.com
15	4.142724	10.63.7.192	10.25.3.2	DNS	73	Standard query 0xbed3 A id.google.com
16	4.143052	10.25.3.2	10.63.7.192	DNS	119	Standard query response 0x77ef A clients5.google.com CNAME clients.l.google.com A 216.58.218.238
17	4.143060	10.63.7.192	10.25.3.2	DNS	75	Standard query 0x9dc6 A apis.google.com
18	4.143438	10.25.3.2	10.63.7.192	DNS	108	Standard query response 0xbed3 A id.google.com CNAME id.l.google.com A 172.217.7.195
19	4.143493	10.63.7.192	10.25.3.2	DNS	77	Standard query 0xc9ad A fonts.gstatic.com
20	4.143728	10.63.7.192	10.25.3.2	DNS	74	Standard query 0xeb70 A www.google.com
21	4.143852	10.63.7.192	10.25.3.2	DNS	85	Standard query 0x7fdb A lh3.googleusercontent.com
22	4.143870	10.25.3.2	10.63.7.192	DNS	112	Standard query response 0x9dc6 A apis.google.com CNAME plus.l.google.com A 216.58.218.238

   The first UDP packet is the one used by Dropbox LAN sync discovery protocol as seen in the snapshot above. Other application using UDP in the above snapshot is DNS.

   The header length of UDP in Dropbox LAN sync discover protocol is **8 bytes**

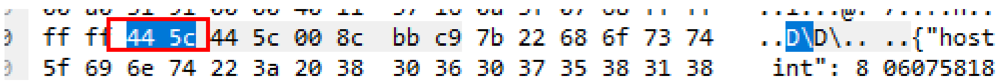


```
> Ethernet II, Src: Dell_a2:91:22 (18:bc:12:a2:91:22), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.63.7.104, Dst: 255.255.255.255
v User Datagram Protocol, Src Port: 17500, Dst Port: 17500
    Source Port: 17500
    Destination Port: 17500
    Length: 140
    Checksum: 0xbbc9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
> Dropbox LAN sync Discovery Protocol
```

```
0010  00 a0 31 91 00 00 40 11  37 16 0a 3f 07 68 ff ff   ..1...@. 7..?.h..
0020  ff ff 44 5c 44 5c 00 8c  bb c9 7b 22 68 6f 73 74   ..D\D\.. ..{"host
0030  5f 69 6e 74 22 3a 20 38  30 36 30 37 35 38 31 38   _int": 8 06075818
0040  36 34 37 32 32 32 33 30  30 37 34 34 36 39 32 39   64722230 07446929
0050  37 37 30 33 39 38 32 32  37 37 32 35 34 2c 20 22   77039822 77254, "
```

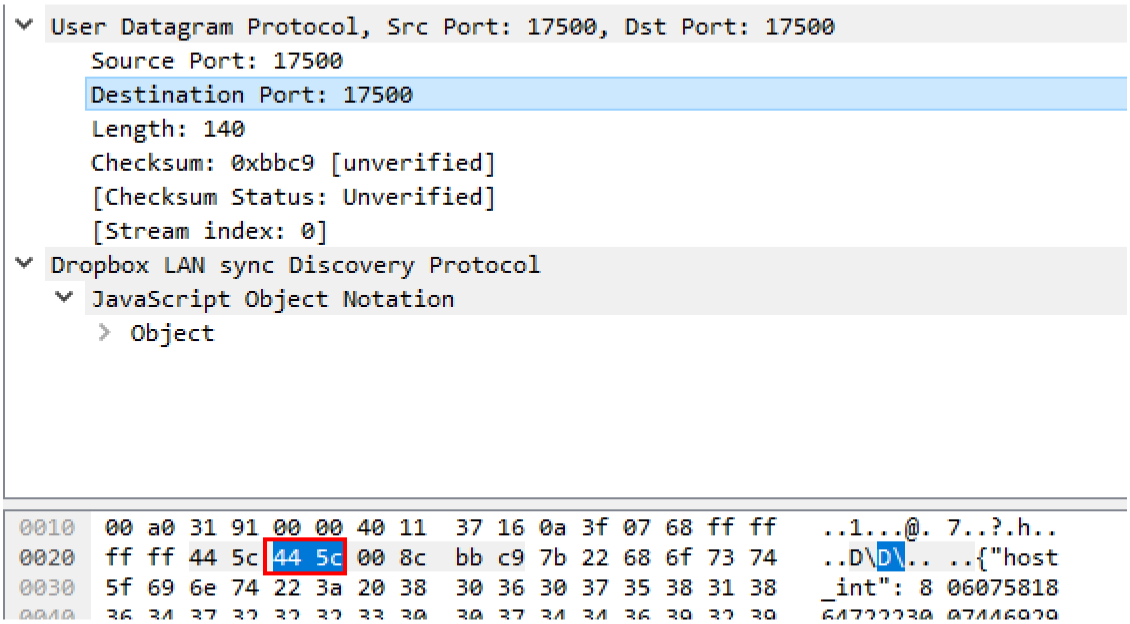
The length of each header field is

1. Source port is 2 bytes



```
00 a0 31 91 00 00 40 11  37 16 0a 3f 07 68 ff ff   ..1...@. 7..?.h..
ff ff 44 5c 44 5c 00 8c  bb c9 7b 22 68 6f 73 74   ..D\D\.. ..{"host
5f 69 6e 74 22 3a 20 38  30 36 30 37 35 38 31 38    int": 8 06075818
```

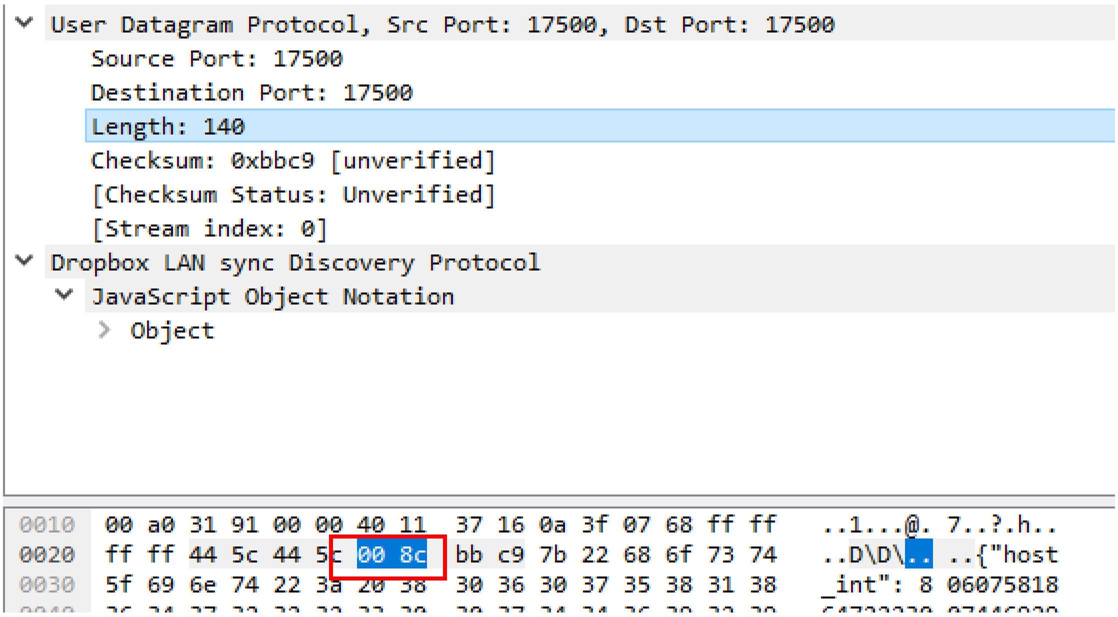
2. Destination port is 2 bytes



```
v User Datagram Protocol, Src Port: 17500, Dst Port: 17500
    Source Port: 17500
    Destination Port: 17500
    Length: 140
    Checksum: 0xbbc9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
v Dropbox LAN sync Discovery Protocol
  v JavaScript Object Notation
    > Object
```

```
0010  00 a0 31 91 00 00 40 11  37 16 0a 3f 07 68 ff ff   ..1...@. 7..?.h..
0020  ff ff 44 5c 44 5c 00 8c  bb c9 7b 22 68 6f 73 74   ..D\D\.. ..{"host
0030  5f 69 6e 74 22 3a 20 38  30 36 30 37 35 38 31 38   _int": 8 06075818
0040  36 34 37 32 32 32 33 30  30 37 34 34 36 39 32 39   64722230 07446929
```

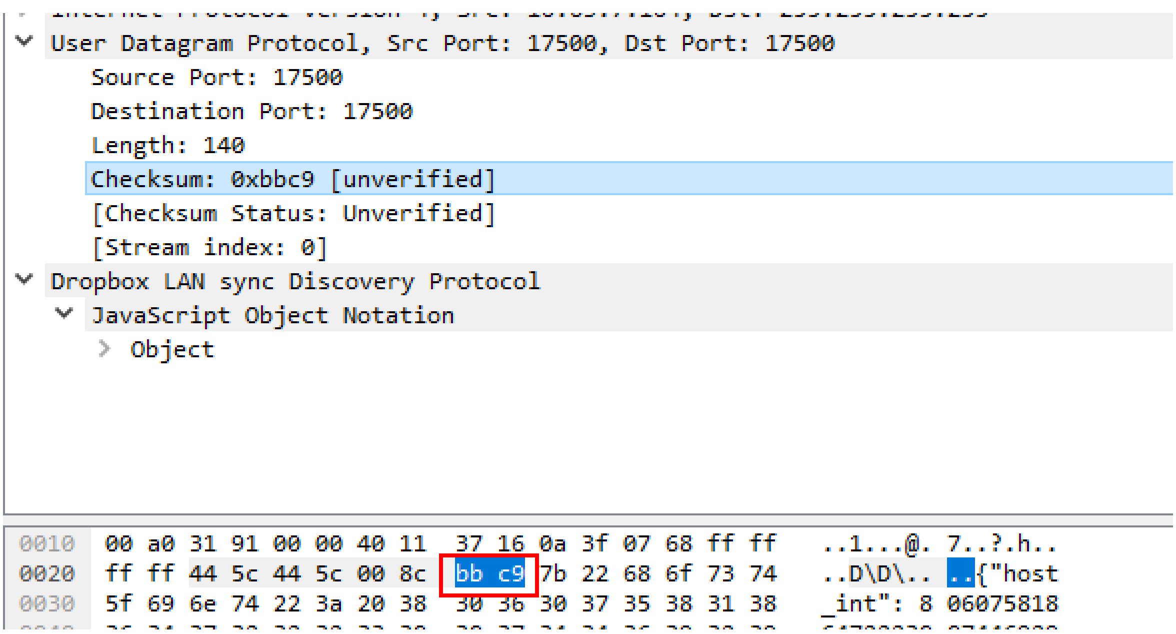
3. Length field is 2 bytes



```
v User Datagram Protocol, Src Port: 17500, Dst Port: 17500
     Source Port: 17500
     Destination Port: 17500
     Length: 140
     Checksum: 0xbbc9 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 0]
v Dropbox LAN sync Discovery Protocol
  v JavaScript Object Notation
     > Object

0010  00 a0 31 91 00 00 40 11  37 16 0a 3f 07 68 ff ff   ..1...@. 7..?.h..
0020  ff ff 44 5c 44 5c 00 8c  bb c9 7b 22 68 6f 73 74   ..D\D\.. ..{"host
0030  5f 69 6e 74 22 3a 20 38  30 36 30 37 35 38 31 38   _int": 8 06075818
```

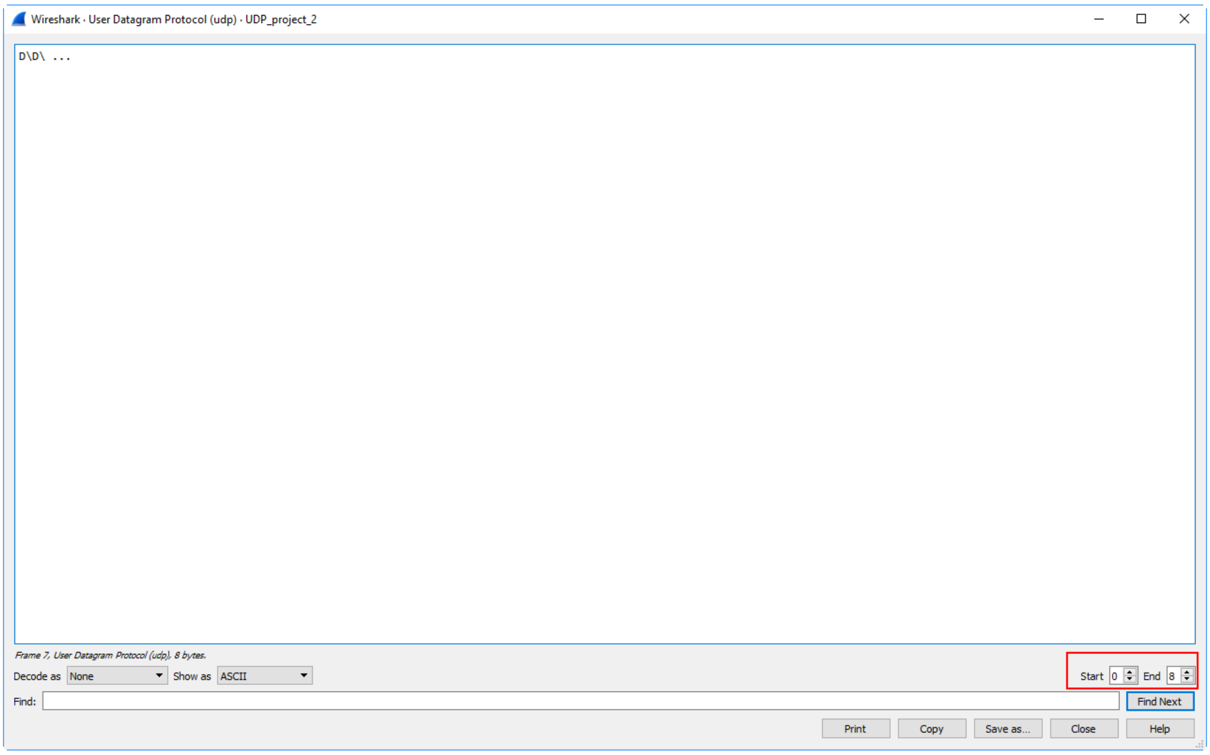
4. Checksum status field is 2 bytes



```
v User Datagram Protocol, Src Port: 17500, Dst Port: 17500
     Source Port: 17500
     Destination Port: 17500
     Length: 140
     Checksum: 0xbbc9 [unverified]
     [Checksum Status: Unverified]
     [Stream index: 0]
v Dropbox LAN sync Discovery Protocol
  v JavaScript Object Notation
     > Object

0010  00 a0 31 91 00 00 40 11  37 16 0a 3f 07 68 ff ff   ..1...@. 7..?.h..
0020  ff ff 44 5c 44 5c 00 8c  bb c9 7b 22 68 6f 73 74   ..D\D\.. ..{"host
0030  5f 69 6e 74 22 3a 20 38  30 36 30 37 35 38 31 38   _int": 8 06075818
```

Show Packet Bytes Option under Wireshark



Wireshark · User Datagram Protocol (udp) · UDP_project_2

```
D\D\ ...
```

Frame 7, User Datagram Protocol (udp), 8 bytes.

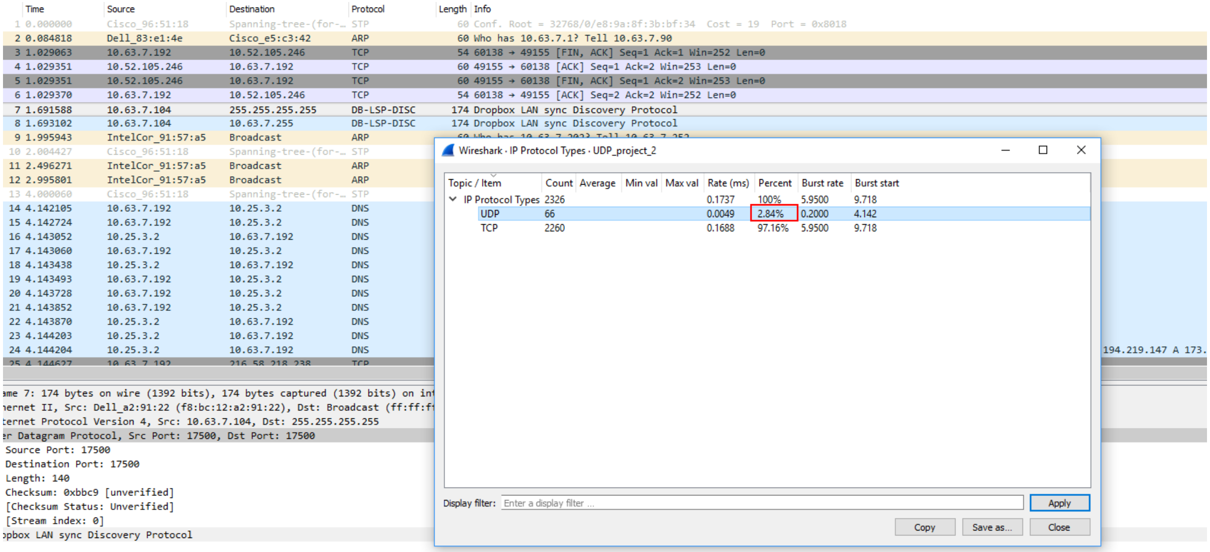
Decode as: None Show as: ASCII Start: 0 End: 8

Find:

2. (1 pts) Using statistics feature of Wireshark, determine the percentage of IPv4 UDP packets in the capture.

   Answer:

   The percentage of IPv4 UDP packets in the capture is **2.84%** as seen from the snapshot below.



```
   Time        Source             Destination           Protocol      Length Info
 1 0.000000    Cisco_96:51:18     Spanning-tree-(for-…  STP               60 Conf. Root = 32768/0/e8:9a:8f:3b:bf:34  Cost = 19  Port = 0x8018
 2 0.084818    Dell_83:e1:4e      Cisco_e5:c3:42        ARP               60 Who has 10.63.7.1? Tell 10.63.7.90
 3 1.029063    10.63.7.192        10.52.105.246         TCP               54 60138 → 49155 [FIN, ACK] Seq=1 Ack=1 Win=252 Len=0
 4 1.029351    10.52.105.246      10.63.7.192           TCP               60 49155 → 60138 [ACK] Seq=1 Ack=2 Win=253 Len=0
 5 1.029351    10.52.105.246      10.63.7.192           TCP               60 49155 → 60138 [FIN, ACK] Seq=1 Ack=2 Win=253 Len=0
 6 1.029370    10.63.7.192        10.52.105.246         TCP               54 60138 → 49155 [ACK] Seq=2 Ack=2 Win=252 Len=0
 7 1.691588    10.63.7.104        255.255.255.255       DB-LSP-DISC      174 Dropbox LAN sync Discovery Protocol
 8 1.693102    10.63.7.104        10.63.7.255           DB-LSP-DISC      174 Dropbox LAN sync Discovery Protocol
 9 1.995943    IntelCor_91:57:a5  Broadcast             ARP
10 2.004427    Cisco_96:51:18     Spanning-tree-(for-…  STP
11 2.496271    IntelCor_91:57:a5  Broadcast             ARP
12 2.995801    IntelCor_91:57:a5  Broadcast             ARP
13 4.000060    Cisco_96:51:18     Spanning-tree-(for-…  STP
14 4.142105    10.63.7.192        10.25.3.2             DNS
15 4.142724    10.63.7.192        10.25.3.2             DNS
16 4.143052    10.25.3.2          10.63.7.192           DNS
17 4.143060    10.63.7.192        10.25.3.2             DNS
18 4.143438    10.25.3.2          10.63.7.192           DNS
19 4.143493    10.63.7.192        10.25.3.2             DNS
20 4.143728    10.63.7.192        10.25.3.2             DNS
21 4.143852    10.63.7.192        10.25.3.2             DNS
22 4.143870    10.25.3.2          10.63.7.192           DNS
23 4.144203    10.25.3.2          10.63.7.192           DNS
24 4.144204    10.25.3.2          10.63.7.192           DNS                                                   194.219.147 A 173.

ame 7: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits) on int
hernet II, Src: Dell_a2:91:22 (f8:bc:12:a2:91:22), Dst: Broadcast (ff:ff:ff
ternet Protocol Version 4, Src: 10.63.7.104, Dst: 255.255.255.255
er Datagram Protocol, Src Port: 17500, Dst Port: 17500
 Source Port: 17500
 Destination Port: 17500
 Length: 140
 Checksum: 0xbbc9 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 0]
opbox LAN sync Discovery Protocol
```

Wireshark · IP Protocol Types · UDP_project_2

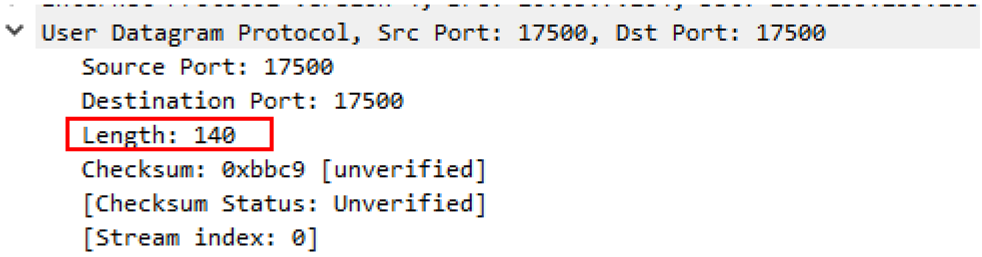
Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
IP Protocol Types	2326				0.1737	100%	5.9500	9.718
UDP	66				0.0049	2.84%	0.2000	4.142
TCP	2260				0.1688	97.16%	5.9500	9.718

Display filter: Enter a display filter ...

3. (1 pts) The value in the Length field is the length of what? (You can consult the textbook for this answer). Verify your claim with your captured UDP packet.

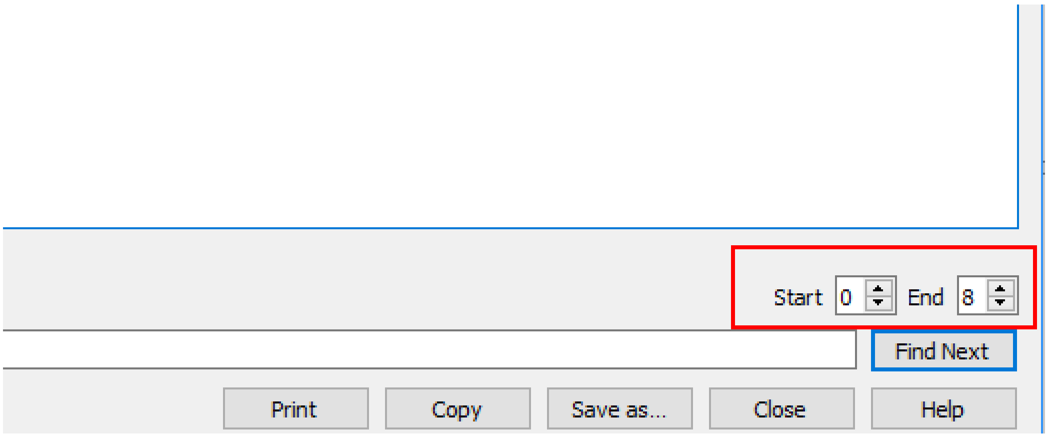
Answer:

The length indicates the combined size of UDP header and the payload. For the Dropbox UDP packet, the length field has 140 bytes as seen in the snapshot below. 8 bytes belong to header and 132 bytes belong to payload.



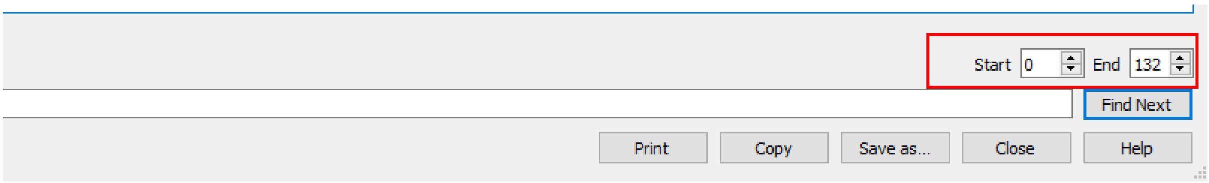
```
User Datagram Protocol, Src Port: 17500, Dst Port: 17500
    Source Port: 17500
    Destination Port: 17500
    Length: 140
    Checksum: 0xbbc9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
```

Image below shows the start and end of UDP header which is 8 bytes.



Start 0 End 8

Image below shows the size of payload which is 132 bytes

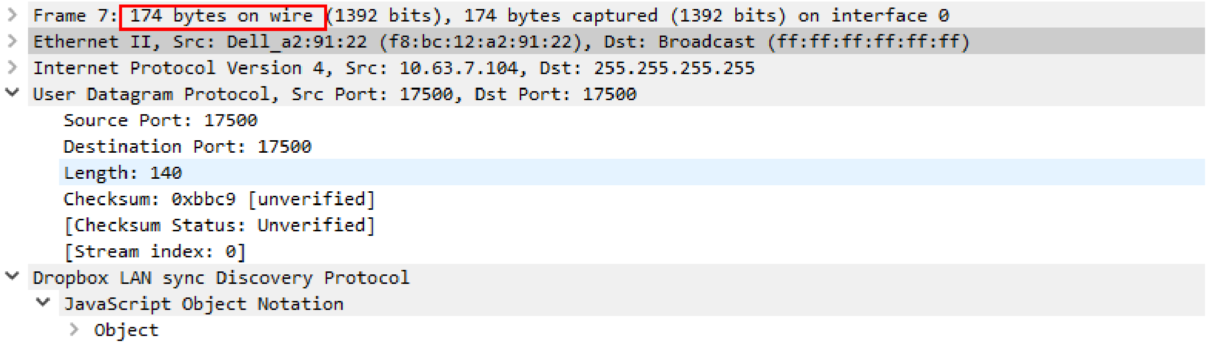


Start 0 End 132

4. (1 pts) What are the source port and length of the first UDP packet in the trace file? What is the largest possible source port number?

Answer:

The source port is 17500 and the total length of the first UDP packet in the trace is 174 bytes. The length of UDP header is 8 bytes, and its combined length with payload is 140 bytes.



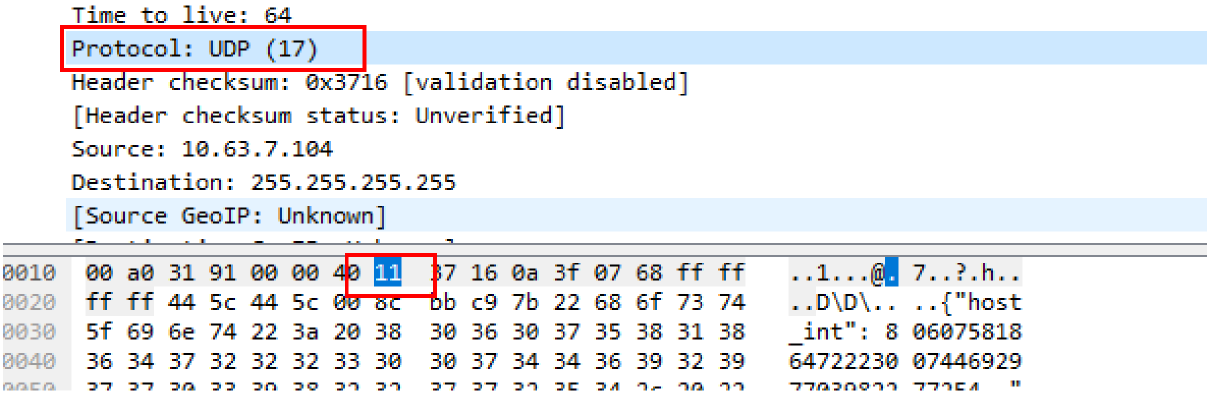
```
> Frame 7: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits) on interface 0
> Ethernet II, Src: Dell_a2:91:22 (f8:bc:12:a2:91:22), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.63.7.104, Dst: 255.255.255.255
v User Datagram Protocol, Src Port: 17500, Dst Port: 17500
    Source Port: 17500
    Destination Port: 17500
    Length: 140
    Checksum: 0xbbc9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
v Dropbox LAN sync Discovery Protocol
  v JavaScript Object Notation
    > Object
```

The maximum possible port number is $2^{16} - 1 = 65535$

5. (1 pts) What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notations along with a screenshot of Wireshark showing those values.

Answer:

The protocol number of UDP is 17 and the equivalent hexadecimal value is 11 as seen in the snapshot.



```
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0x3716 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.63.7.104
    Destination: 255.255.255.255
    [Source GeoIP: Unknown]

0010  00 a0 31 91 00 00 40 11  37 16 0a 3f 07 68 ff ff   ..1...@. 7..?.h..
0020  ff ff 44 5c 44 5c 00 8c  bb c9 7b 22 68 6f 73 74   ..D\D\.. ..{"host
0030  5f 69 6e 74 22 3a 20 38  30 36 30 37 35 38 31 38   _int": 8 06075818
0040  36 34 37 32 32 32 33 30  30 37 34 34 36 39 32 39   64722230 07446929
```

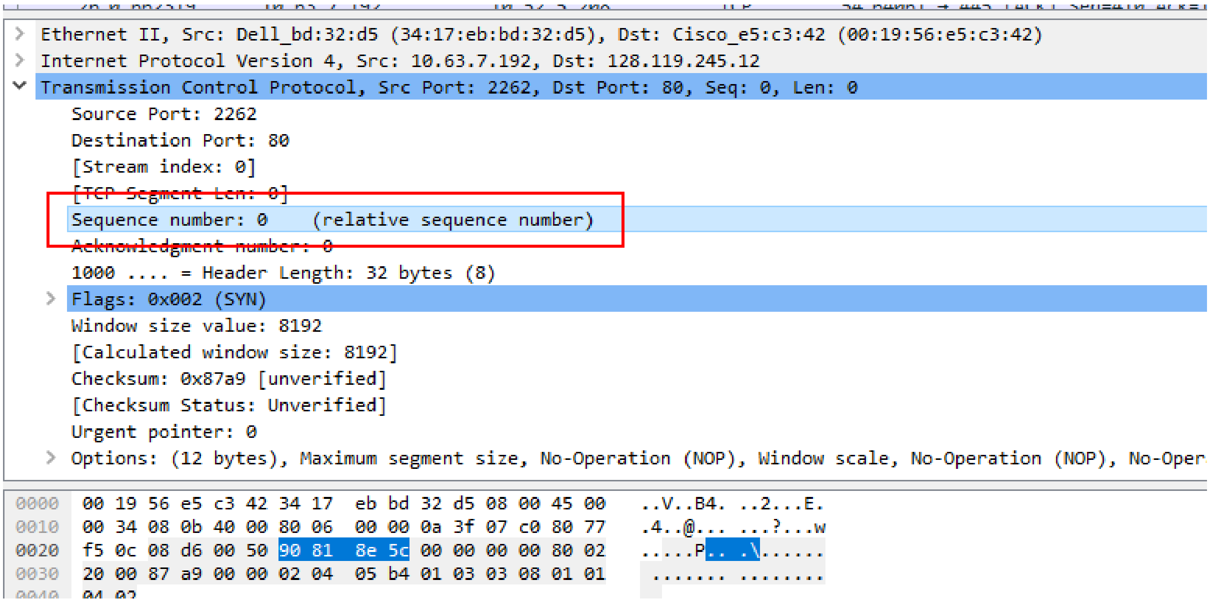
Part II: TCP

Open the file ‘TCP_project_2.pcapng’ in Wireshark and answer the following questions. The trace file was captured while uploading ‘1600.txt’ file from a computer (10.63.7.192) to *gaia.cs.umass.edu* web server (128.119.245.12) using the HTTP POST method.

1. (1 pts) What is 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*?

Answer:

The TCP sequence number is 0 for SYN segment used to initiate the TCP connection



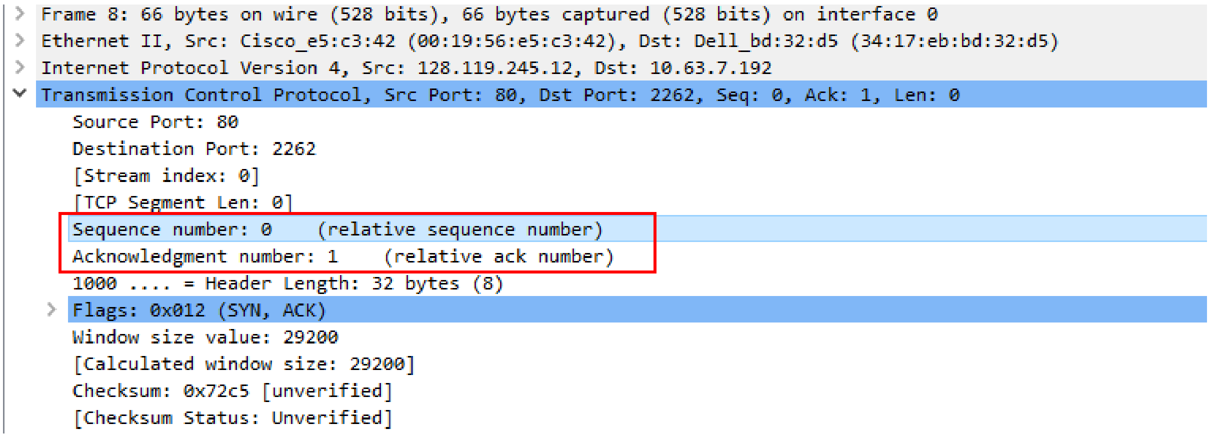
```
> Ethernet II, Src: Dell_bd:32:d5 (34:17:eb:bd:32:d5), Dst: Cisco_e5:c3:42 (00:19:56:e5:c3:42)
> Internet Protocol Version 4, Src: 10.63.7.192, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 2262, Dst Port: 80, Seq: 0, Len: 0
     Source Port: 2262
     Destination Port: 80
     [Stream index: 0]
     [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: 8192
     [Calculated window size: 8192]
     Checksum: 0x87a9 [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Oper

0000  00 19 56 e5 c3 42 34 17  eb bd 32 d5 08 00 45 00   ..V..B4. ..2...E.
0010  00 34 08 0b 40 00 80 06  00 00 0a 3f 07 c0 80 77   .4..@... ...?...w
0020  f5 0c 08 d6 00 50 90 81  8e 5c 00 00 00 00 80 02   .....P.. .\......
0030  20 00 87 a9 00 00 02 04  05 b4 01 03 03 08 01 01   ........ ........
0040  04 02
```

2. (1 pts) What are the sequence number and acknowledgement number of the first SYNACK packet sent from the server to the client computer? How were the values determined by the server? *(hint: relative seq and ack values displayed by Wireshark is fine, no need to show actual numbers)*

Answer:

The sequence number is 0 and acknowledgement number is 1 in the first SYNACK packet sent from server to client. In the first packet sent by the client, both sequence number and acknowledgement number were 0. The server chooses its own sequence number and the acknowledgement number contains previous_client_sequence_number + 1. The sequence number of the SYN packet sent from client to server is 0. Hence, the acknowledgement number of SYNACK packet is 0 + 1 = 1. This indicates that the server is expecting the next packet from client with sequence number 1.

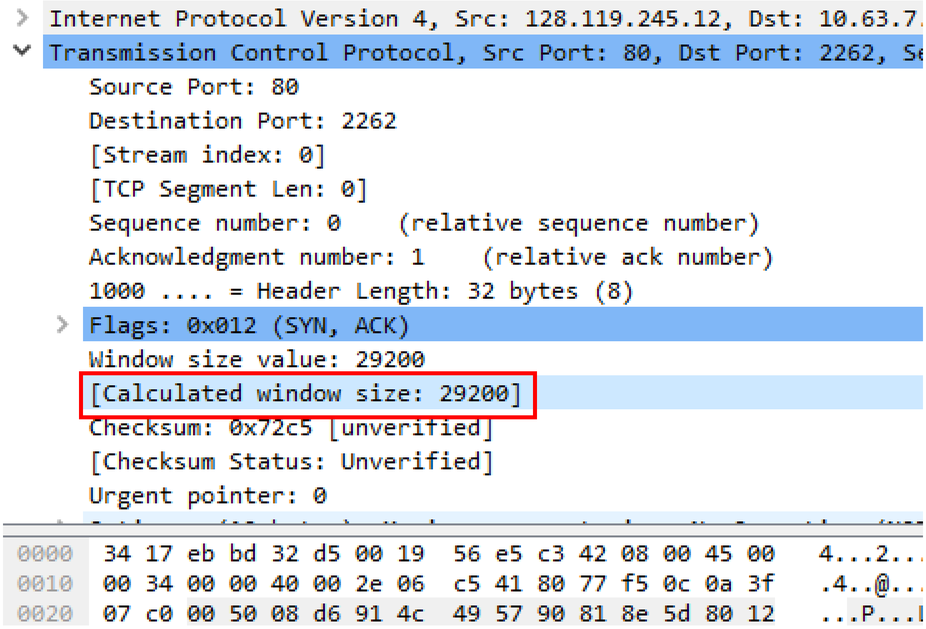


```
> Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: Cisco_e5:c3:42 (00:19:56:e5:c3:42), Dst: Dell_bd:32:d5 (34:17:eb:bd:32:d5)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.63.7.192
v Transmission Control Protocol, Src Port: 80, Dst Port: 2262, Seq: 0, Ack: 1, Len: 0
     Source Port: 80
     Destination Port: 2262
     [Stream index: 0]
     [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: 0x72c5 [unverified]
     [Checksum Status: Unverified]
```

3. (1 pts) What are the minimum and maximum amount of available buffer spaces advertised at the receiver for the entire trace?

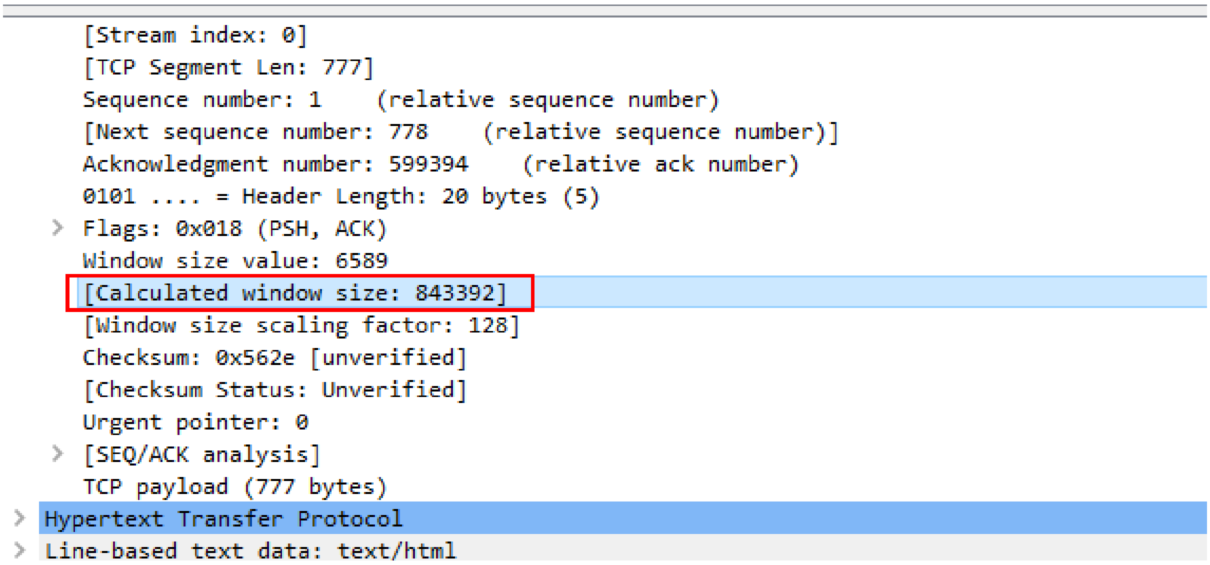
Answer:

The minimum buffer space advertised at the receiver is 29200 and the maximum is 843392 as seen in the snapshots below.



```
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.63.7.
v Transmission Control Protocol, Src Port: 80, Dst Port: 2262, Se
     Source Port: 80
     Destination Port: 2262
     [Stream index: 0]
     [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: 0x72c5 [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0

0000  34 17 eb bd 32 d5 00 19  56 e5 c3 42 08 00 45 00   4...2...
0010  00 34 00 00 40 00 2e 06  c5 41 80 77 f5 0c 0a 3f   .4..@...
0020  07 c0 00 50 08 d6 91 4c  49 57 90 81 8e 5d 80 12   ...P...I
```

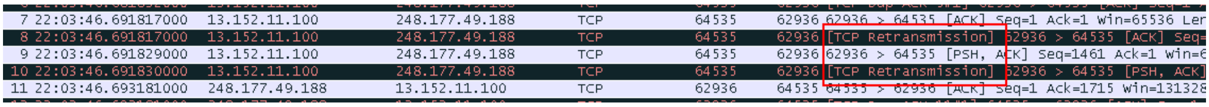


```
        [Stream index: 0]
        [TCP Segment Len: 777]
        Sequence number: 1    (relative sequence number)
        [Next sequence number: 778    (relative sequence number)]
        Acknowledgment number: 599394    (relative ack number)
        0101 .... = Header Length: 20 bytes (5)
      > Flags: 0x018 (PSH, ACK)
        Window size value: 6589
        [Calculated window size: 843392]
        [Window size scaling factor: 128]
        Checksum: 0x562e [unverified]
        [Checksum Status: Unverified]
        Urgent pointer: 0
      > [SEQ/ACK analysis]
        TCP payload (777 bytes)
  > Hypertext Transfer Protocol
  > Line-based text data: text/html
```

4. (1 pts) Are there any retransmitted segments in the trace file? What is the reasoning behind your answer?

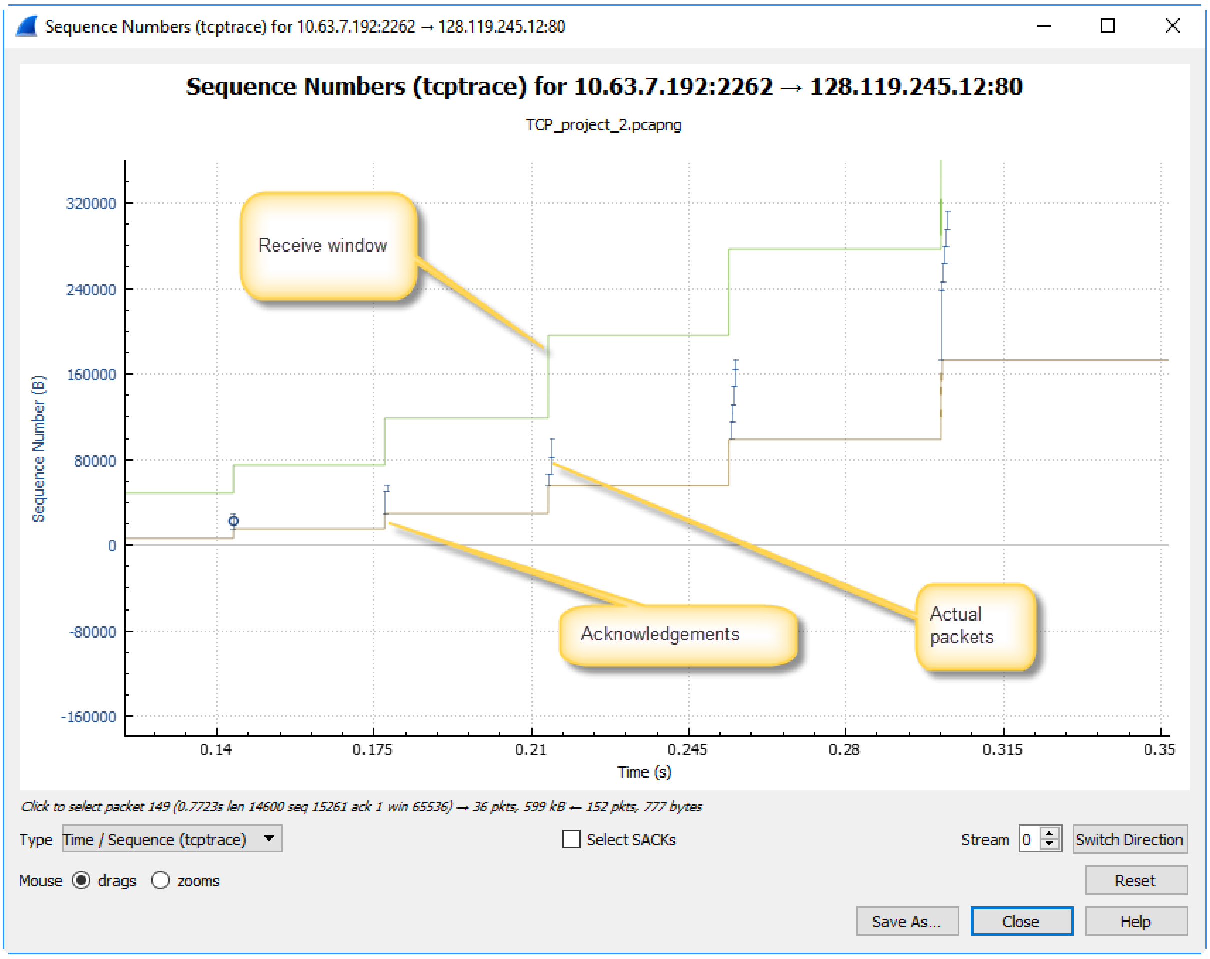
Answer:

There are no retransmitted segments in the trace file. By evaluating the sequence numbers across the trace, there were no duplicate sequence numbers and no packet marked with TCP Retransmission. For example, here is a snapshot of another packet trace which shows how a packed is marked when retransmitted.



```
 7 22:03:46.691817000  13.152.11.100   248.177.49.188  TCP  64535  62936 62936 > 64535 [ACK] Seq=1 Ack=1 Win=65536 Ler
 8 22:03:46.691817000  13.152.11.100   248.177.49.188  TCP  64535  62936 [TCP Retransmission] 62936 > 64535 [ACK] Seq=
 9 22:03:46.691829000  13.152.11.100   248.177.49.188  TCP  64535  62936 62936 > 64535 [PSH, ACK] Seq=1461 Ack=1 Win=6
10 22:03:46.691830000  13.152.11.100   248.177.49.188  TCP  64535  62936 [TCP Retransmission] 62936 > 64535 [PSH, ACK]
11 22:03:46.693181000  248.177.49.188  13.152.11.100   TCP  62936  64535 64535 > 62936 [ACK] Seq=1 Ack=1715 Win=131328
```

Also, the TCP trace does not show any retransmission



Click to select packet 149 (0.7723s len 14600 seq 15261 ack 1 win 65536) → 36 pkts, 599 kB ← 152 pkts, 777 bytes

5. (1 pts) What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Answer:

The file size transferred is 599393 bytes. Time taken to transfer this file is the difference between the time when the first data packet is sent which is number 106 and the time when complete data is received at the receiver which is packet number 781.

1.0670 – 0.6756 = 0.3914 s

The last ack number – 1

599394 – 1 = 599393 bytes

Therefore, throughput = 599393 / 0.3914 = 1531407 Bps = 1495 KBps = 11964 Kbps