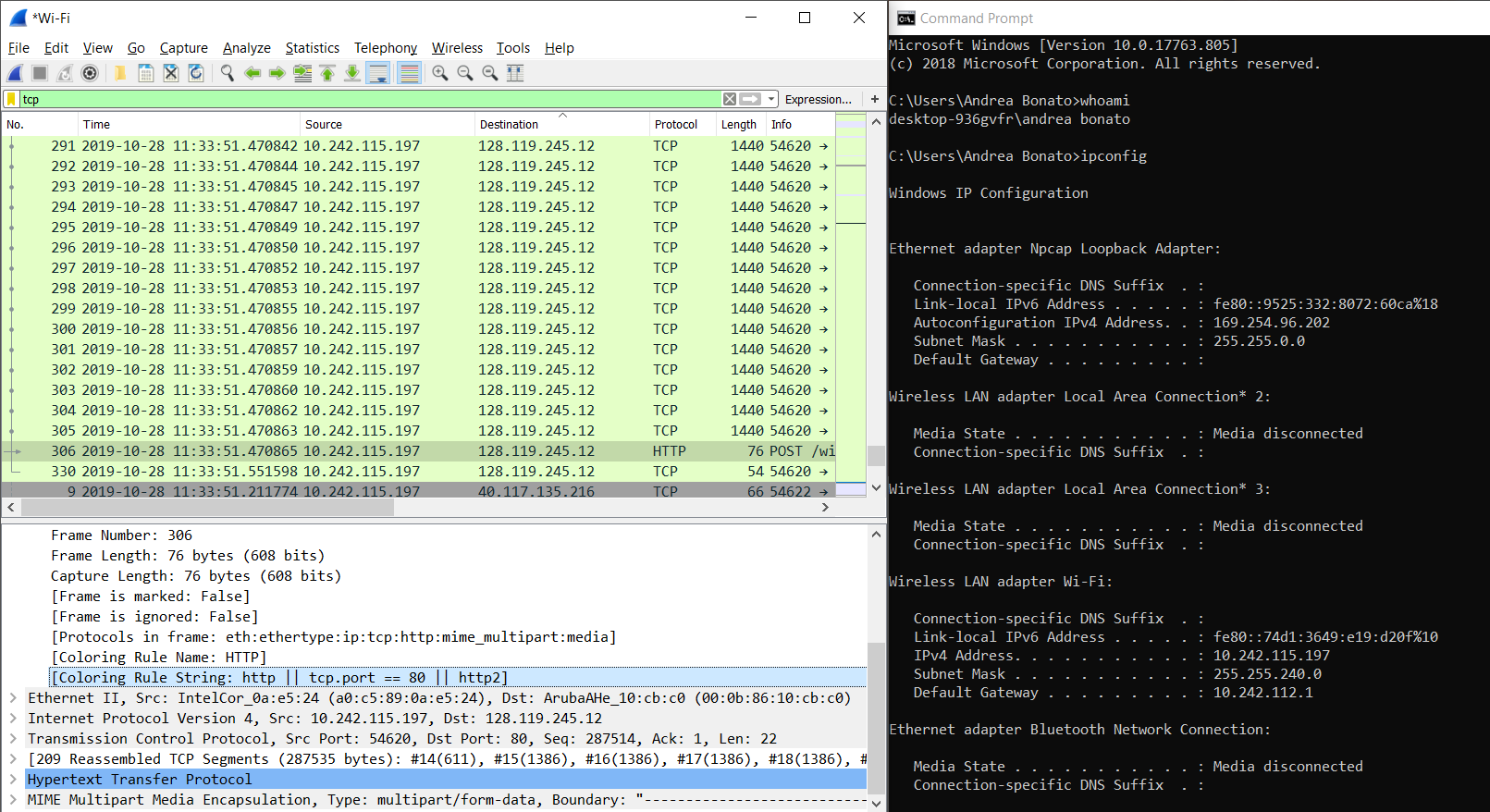
# COMP-3670 Fall 2019

## Assignment 2

## Andrea Bonato (104760390)

## Wireshark Lab: TCP v7.0

#### Figure 1 – Uploading a file



### What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

The source IP address used by the client computer is 10.242.115.197. The TCP port number used by the client computer is 54620. This is shown in **figure 1.**

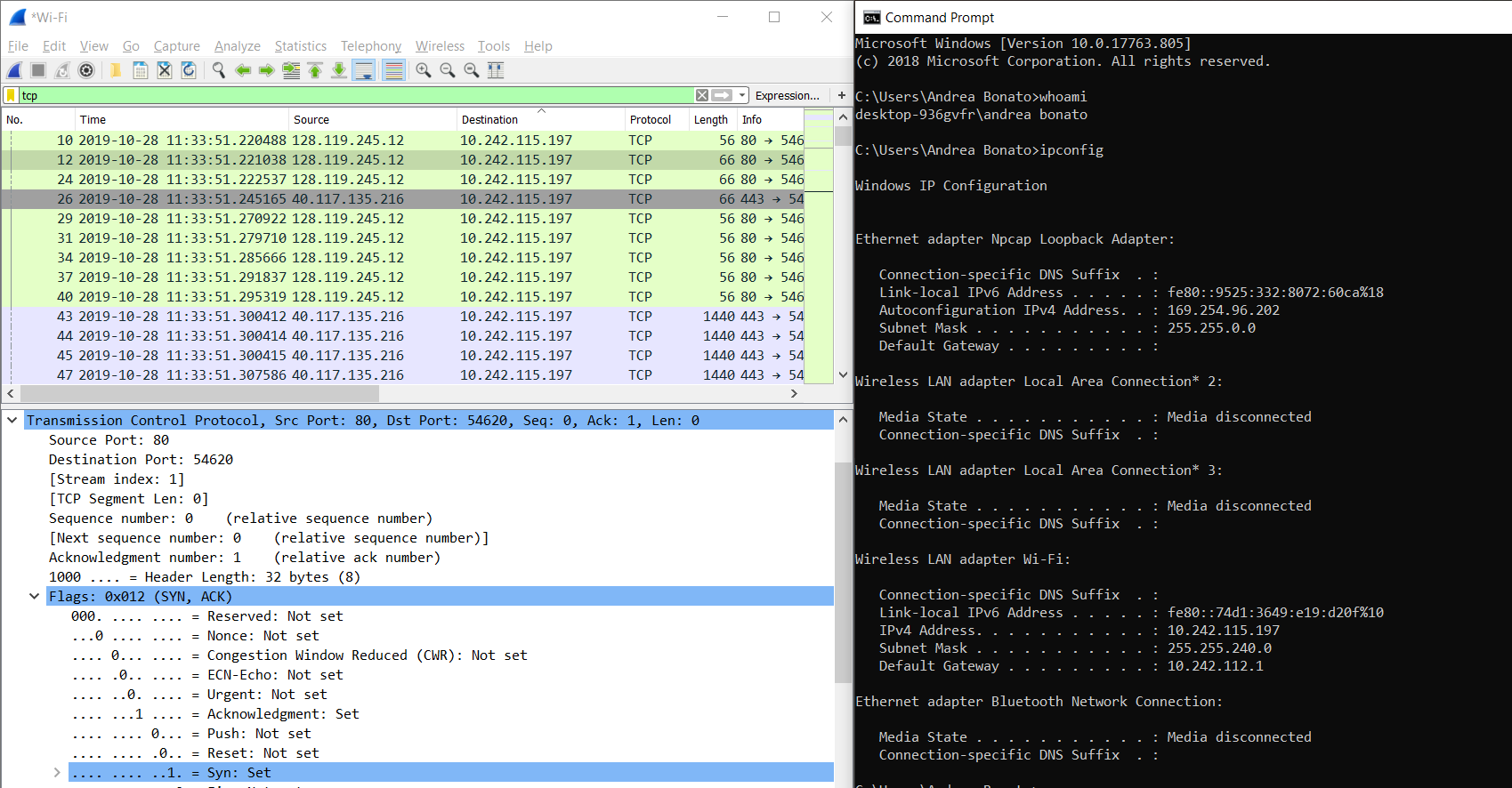
### What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

The IP address of gaia.cs.umass.edu is 128.119.245.12. The port number that it is sending and receiving TCP segments through is 80. This is shown in **figure 1.**

### What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?

The source IP address used by my computer is 10.242.115.197. The TCP port number used by the client computer is 54620. This is shown in **figure 1.**

#### Figure 2 – SYN flag and sequencing

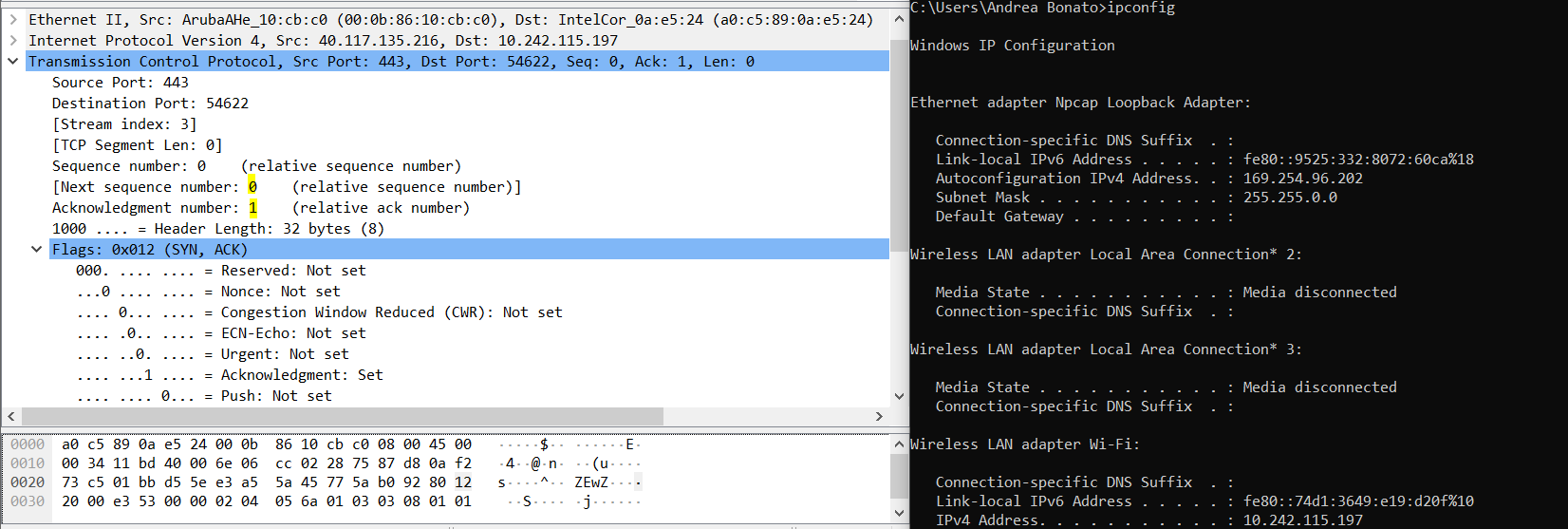


### What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as SYN segment?

As shown in **figure 2**, the sequence number of the TCP SYN is 0 since it is used to start the TCP connection between the client and the website gaia.cs.uass.edu. It shows that the segment is a SYN segment because the SYN flag equals 1.

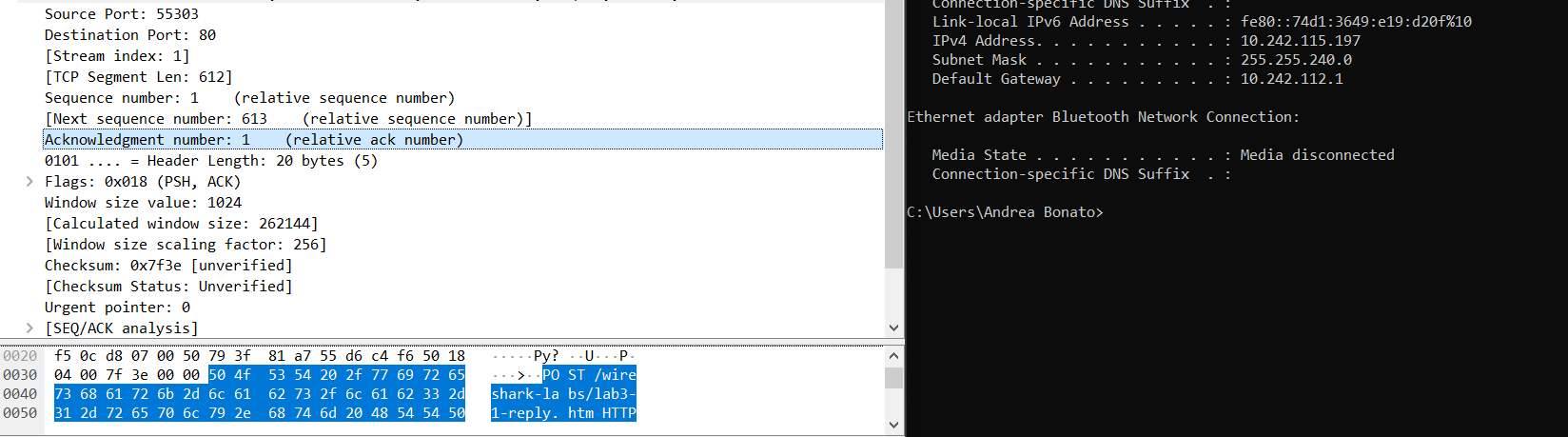
### What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the acknowledgeme3nt field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

As shown in the figure below, the sequence number of the SYNACK segment sent by gaia.cs.umass.edu is 0 and the value of the acknowledgement field is 1. This segment can be identified as a SYNACK segment if both SYN flag and the ACK flags are set to 1.



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

The sequence number oft the TCP segment containing the HTTP POST command is 1. This is shown in the image below.



### Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection? 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 estimated RTT value after the receipt of each ACK?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Segment | Sent Time | ACK Received Time | Round Trip Time | EstimatedRTT |
| 1 | 0.066123 | 0.122956 | 0.056833 | 0.056833 |
| 1387 | 0.122958 | 0.122959 | 0.000001 | 0.048308175 |
| 2773 | 0.122960 | 0.123223 | 0.000265 | 0.042302778125 |
| 5545 | 0.123640 | 0.123836 | 0.000196 | 0.037039430859375 |
| 5708 | 0.270897 | 0.328977 | 0.058080 | 0.039669502001953125 |
| 6991 | 0.500629 | 0.510955 | 0.010326 | 0.0360015642551708984375 |

### What is the length of each of the first six TCP segments?

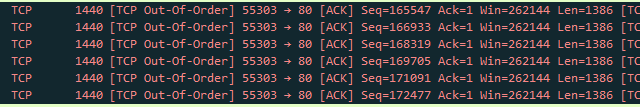
The length of the first segment is 448 bytes and the remaining segments are 1440 bytes per segment.

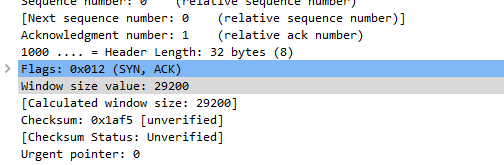
### What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

The minimum amount of buffer space that is seen on the gaia.cs.umass.edu server is 33408 bytes and shown in the image below from the win section.



The receiver window grows until a maximum buffer size of 29200 as shown in the image below.

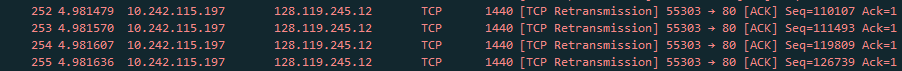




Looking at this trace, this trace throttled due to the size of the buffer.

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





As shown in the images above, there were several times where there was a retransmission segment. In order to check this, I checked the segment stack that is outputted when filtered by TCP.

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

|  |  |  |
| --- | --- | --- |
| ACK | Sequence Number | Acknowledge Data |
| 1 | 1999 | 448 |
| 2 | 3385 | 1440 |
| 3 | 4771 | 1440 |
| 4 | 6157 | 1440 |
| 5 | 7543 | 1440 |
| 6 | 8929 | 1440 |
| 7 | 10315 | 1440 |

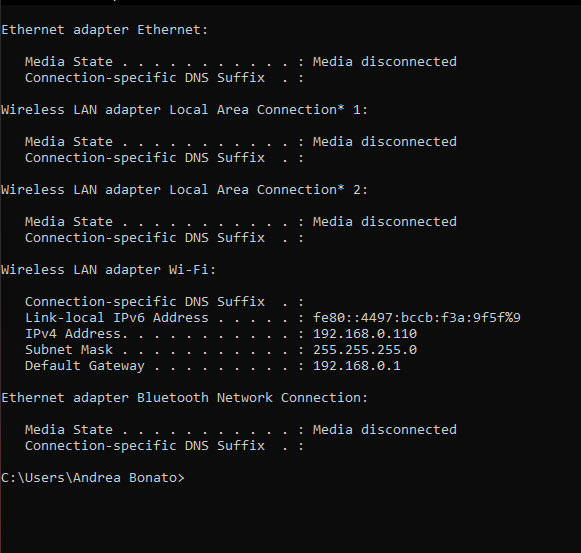
1440 \* 2 = 2880 is when the segment is ACKing every other segment. It is shown in No. 214



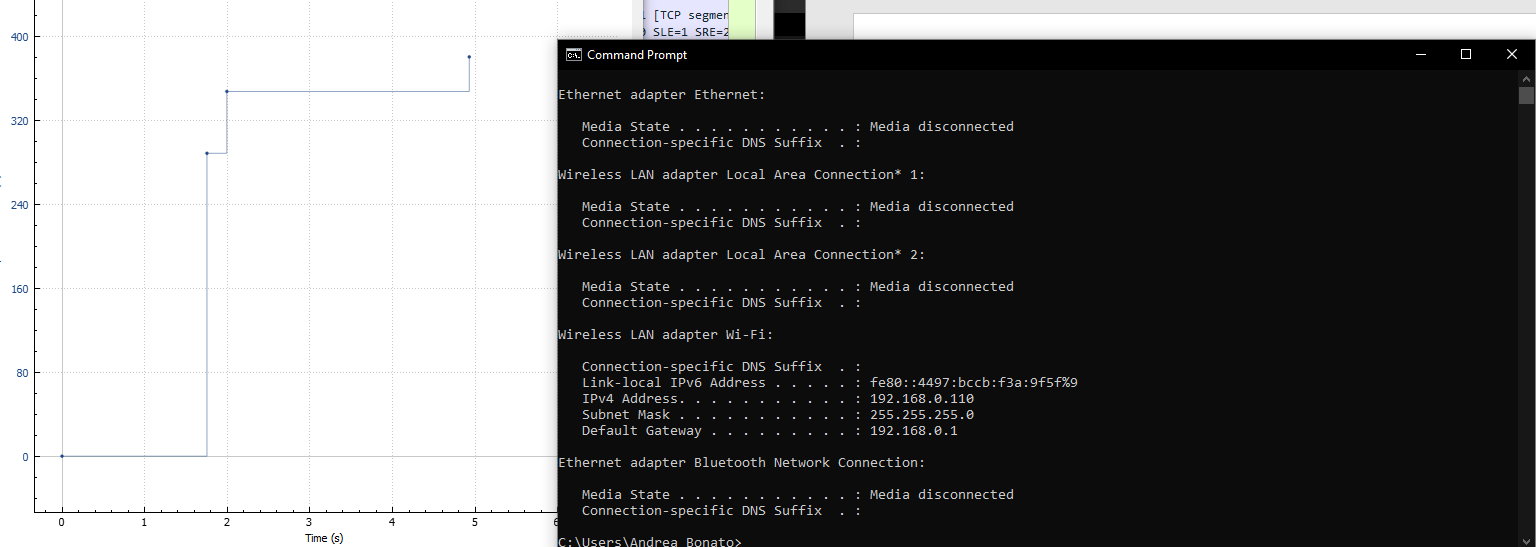
### What is the throughput for the TCP connection? Explain how you calculated this value?

When taking the window size, and converting it to bits, one can divide it by the latency to calculate the throughput for the TCP connection. The size of the given file is 280KB which is equivalent two 280,000 bytes. The first time is 0.399183s and the final time is 0.536283s. The difference in time is 0.1371 seconds. Thus, the throughput is 280,000/0.1371 = 2,041,304.88694383 bytes/second.

#### Figure 3 – Change of location

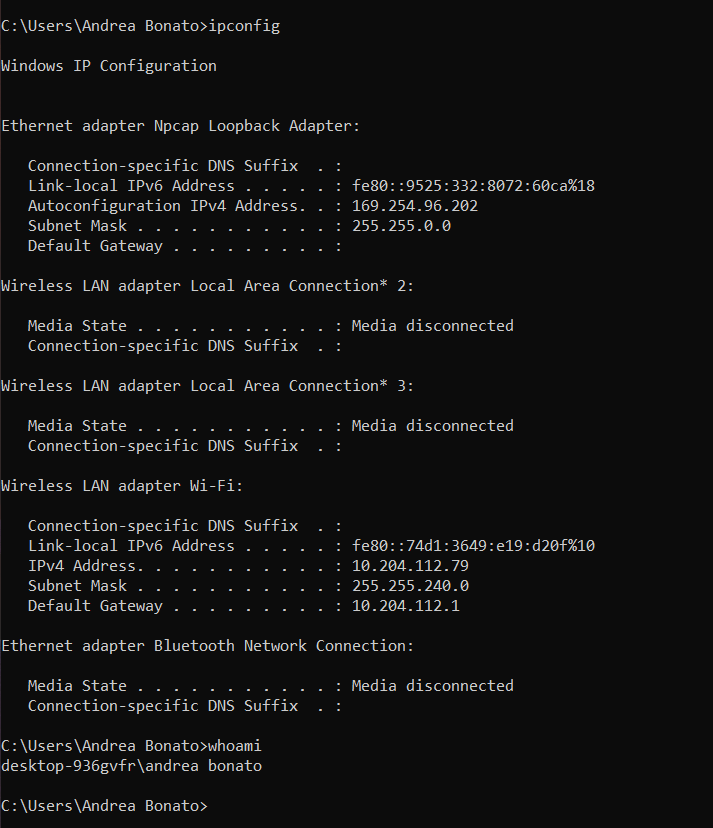


### Use the time-sequence-graph plotting tool to view the sequence number versus time plot o segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP’s slow start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behaviour of TCP that we’ve studied in the text.

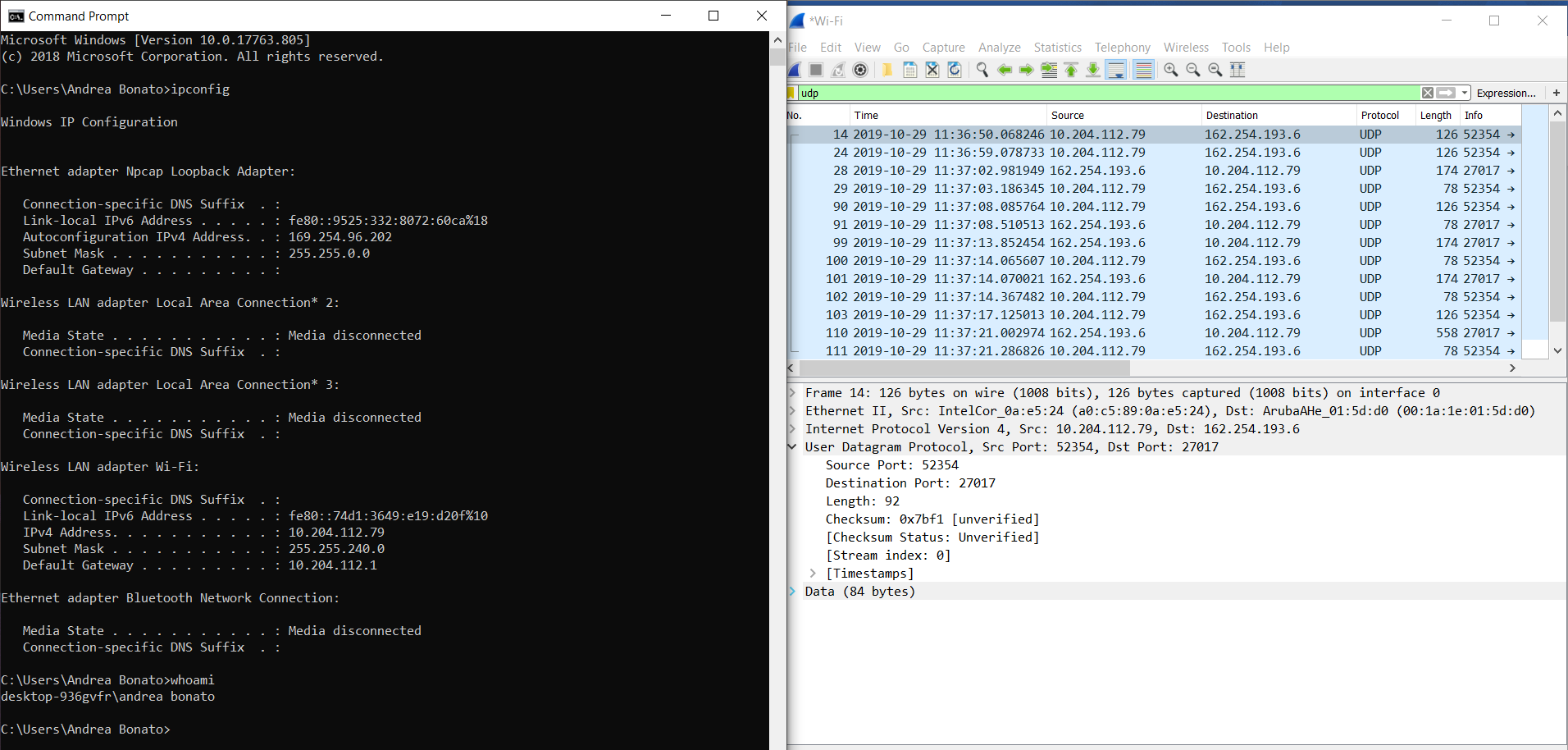


By looking at the graph we can see that the phase started slowly around 0-2 seconds which can be considered the slow start. The congestion avoidance takes over after the 2nd second. The measured data differs from the behaviour of TCP that we studied because it seems that the HTTP server has an enforced rate limit of some sort. This does not seem to be caused by the flow control of the window and server.

## Section 2 – Wire Shark Lab UDP



### Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. Name these fields.



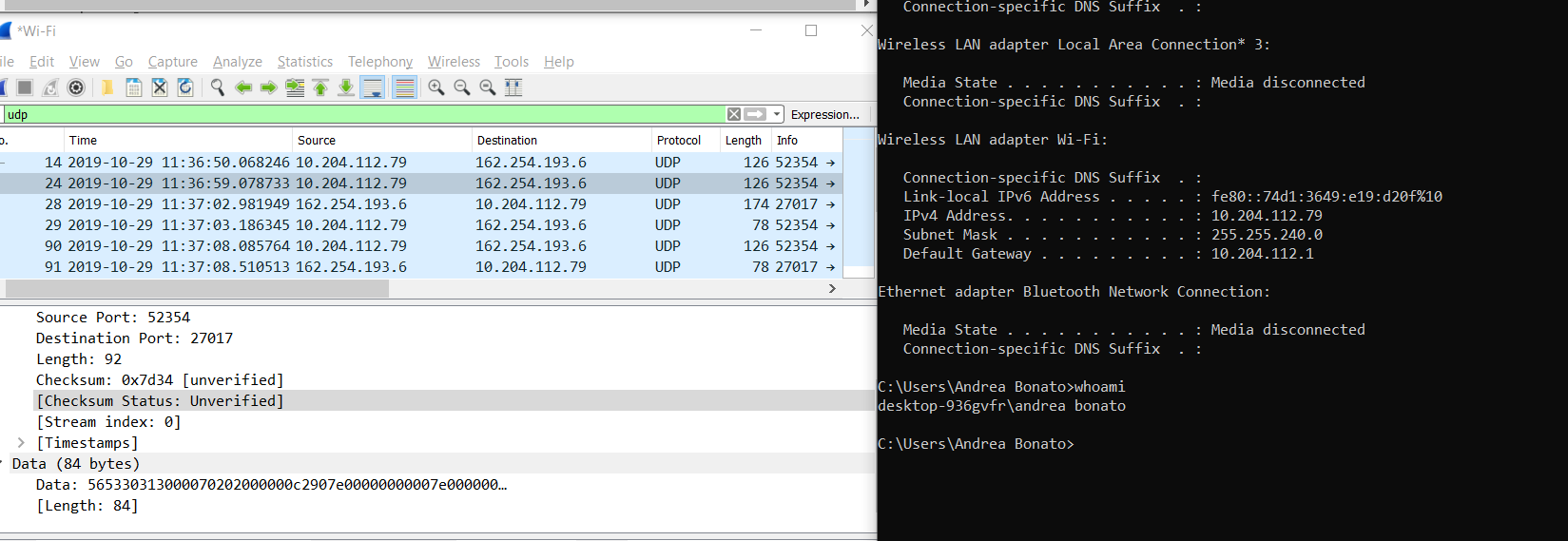
As shown above, there are 4 fields in the UDP header. These fields are source port, destination port, length, and checksum.

### By consulting the displayed information in Wireshark’s packet content field for this packet, determine the length of each of the UDP header fields.

The header is 8 bytes, so each header field must be 2 bytes long.

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

The length field in the header is the length of the UDP segment. In includes the size of the data and the header combines. It is 92 bytes in total, where the data itself is 84 bytes and the header is 8 bytes.



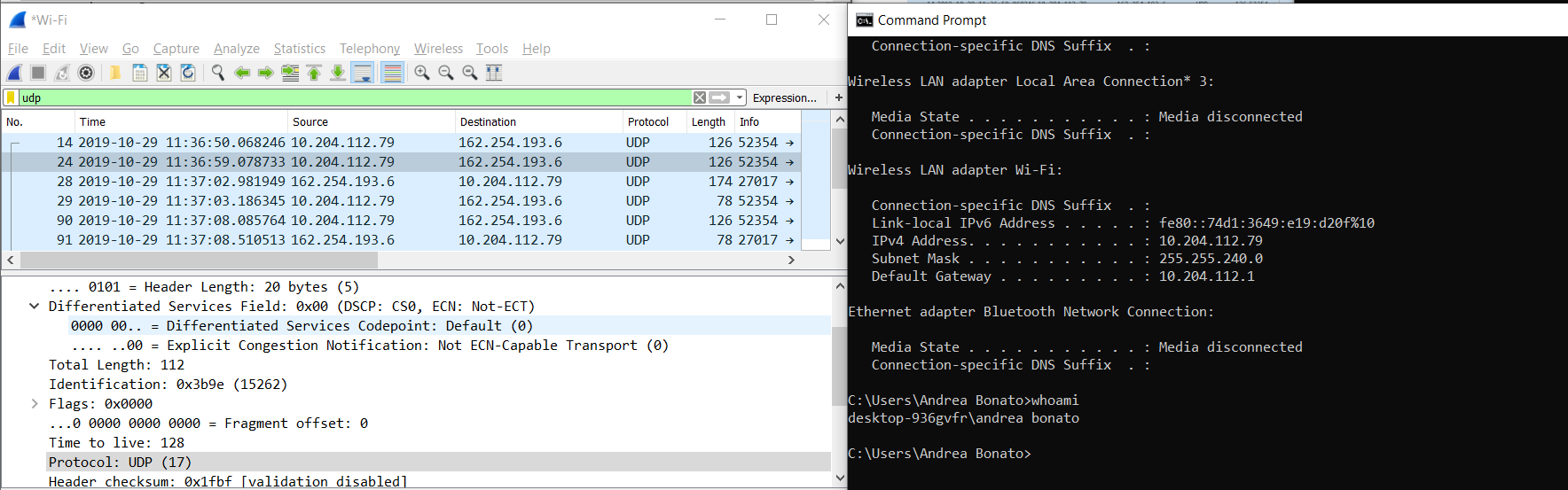
### What is the maximum number of bytes that be included in a UDP payload?

The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) - the header all in bytes. This meant that 65536 bytes – 1 byte – 8 bytes = 65527 bytes.

### What is the largest possible source port?

Like the question 4, the largest source port will be 2^16 – 1 which is calculated to be 65535.

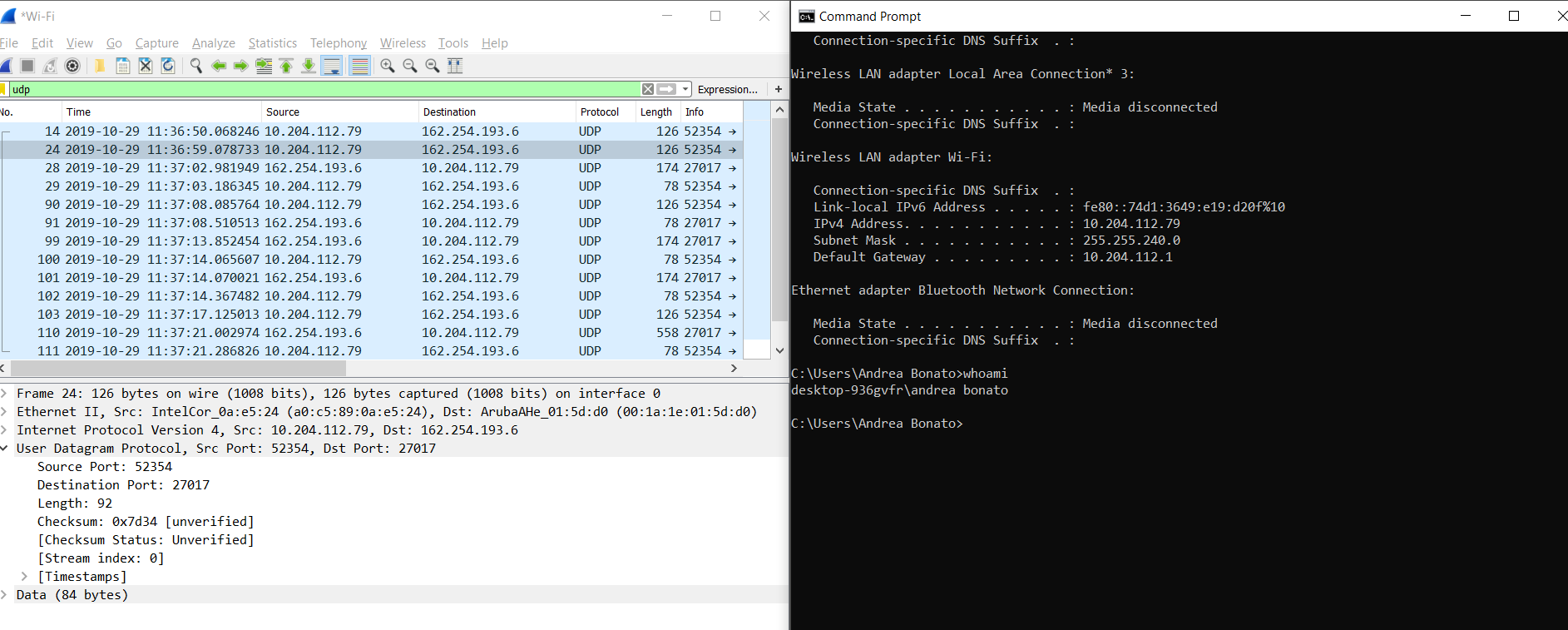
### What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you’ll need to look into the Protocol field of the IP diagram containing this UDP segment.



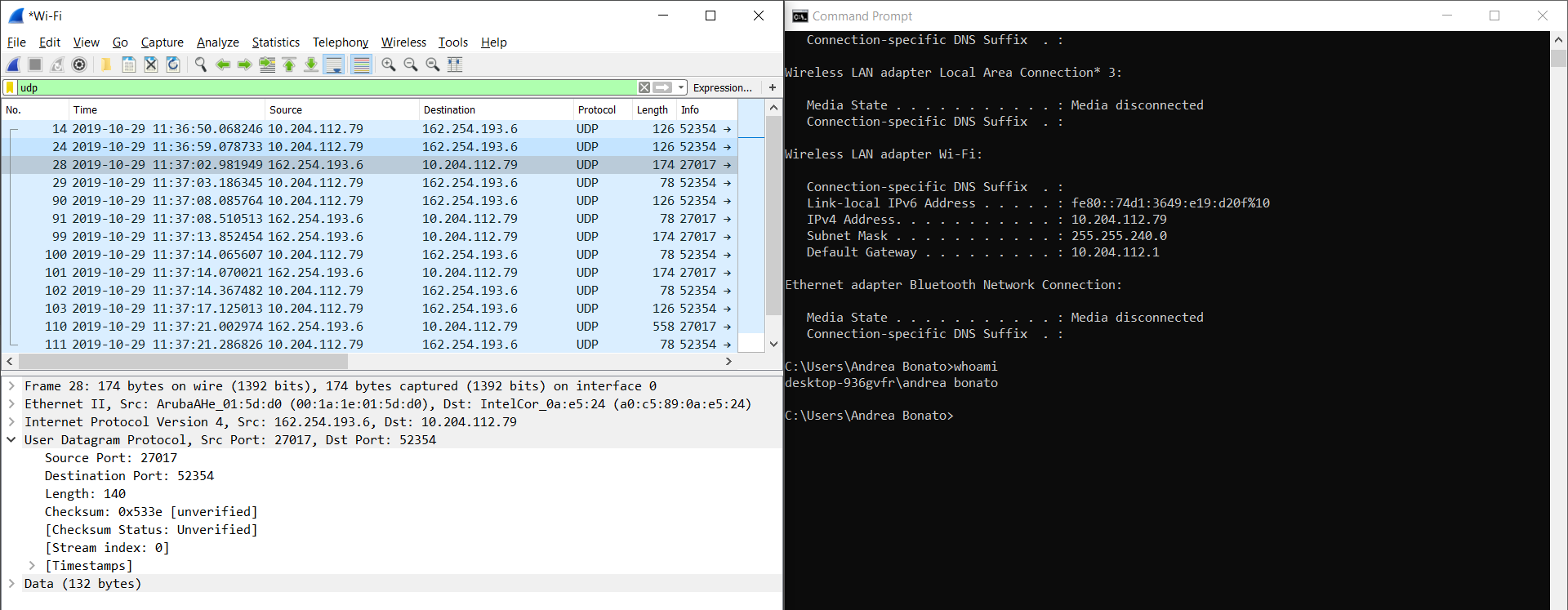
As shown in the screen shot above, the protocol UDP number is 17 in decimal notation and 0x11 when converted to hexadecimal

### Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. Describe the relationship between the port numbers in the two packets.

Packet 1: Sent

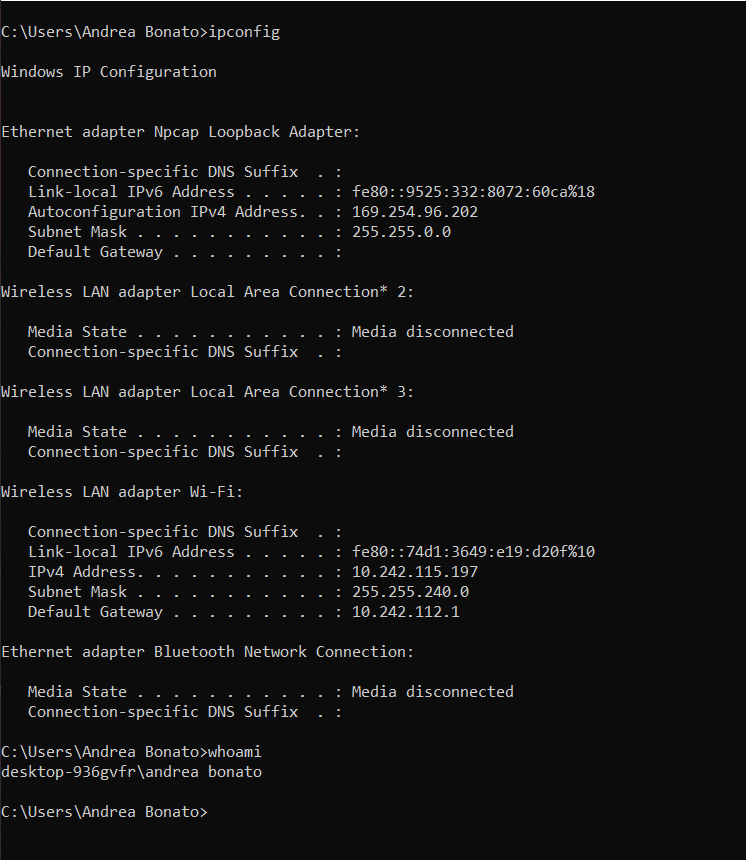


Packet 2: Received

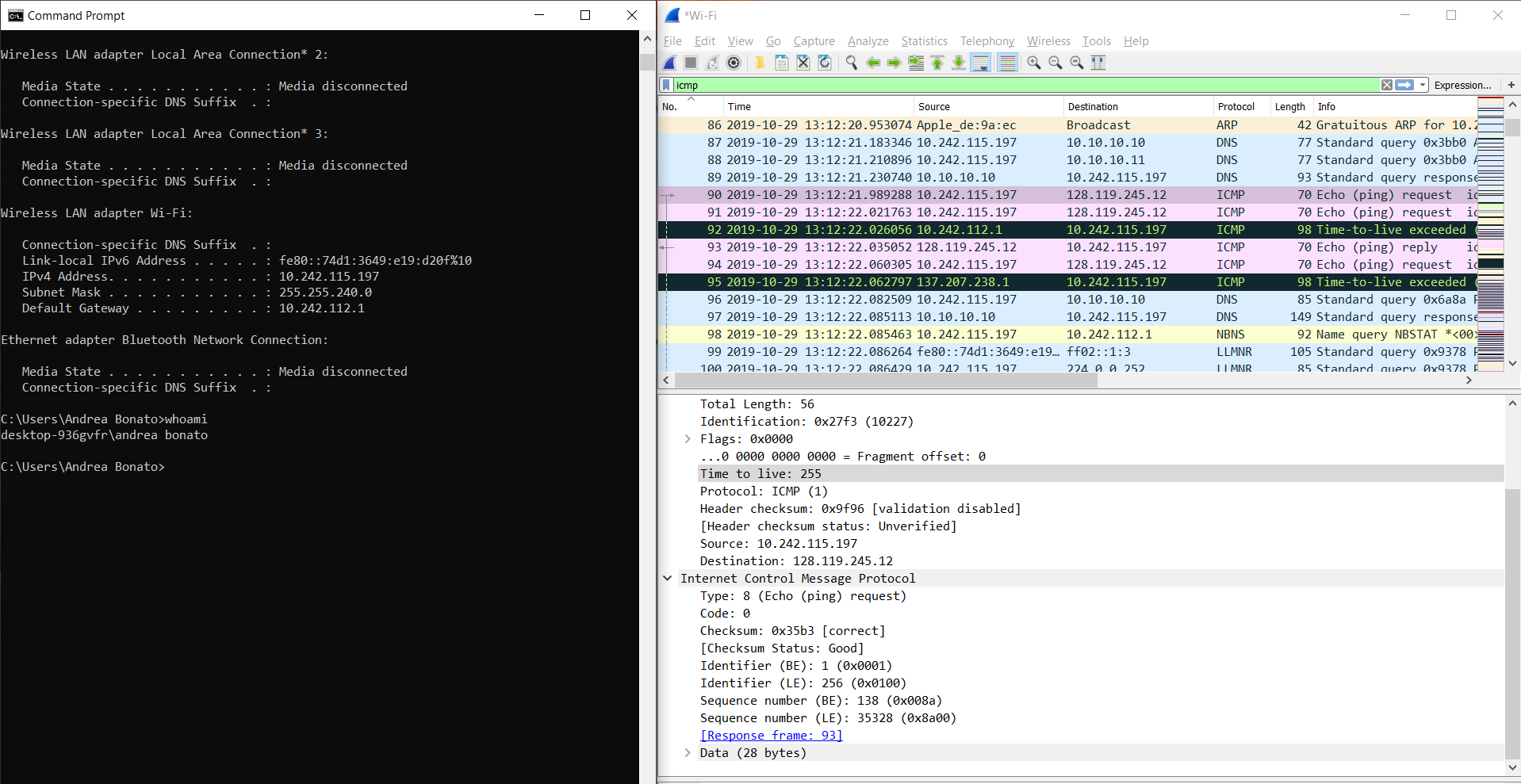
.

The source port of the UDP packet sent by the client is the same as the destination of the packet received, similarly, the destination port of the client is the source port number of the response.

## Section 3 – Wireshark Lab IP

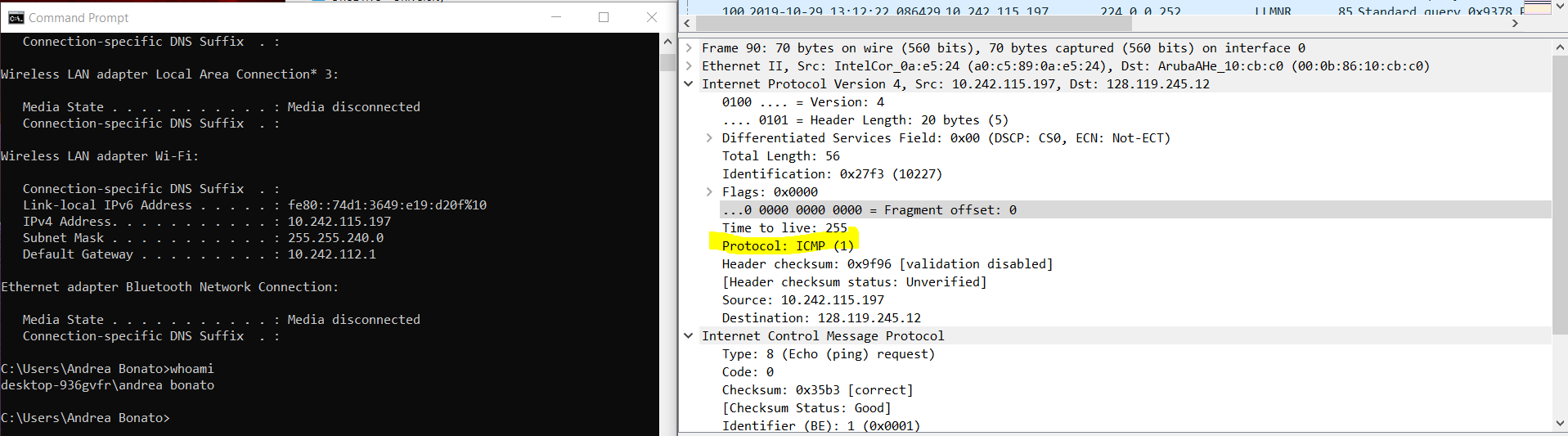


### Select the first ICMP echo request message sent by your computer and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?



The IP address of my computer, as given by the Source for the echo request, is 10.242.115.197.

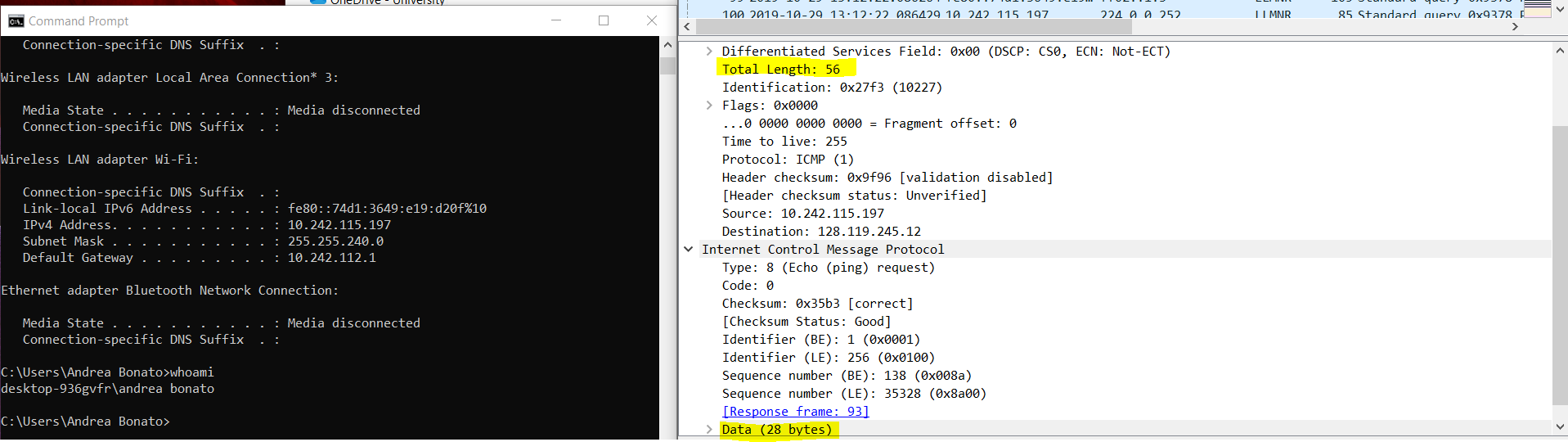
### Within the IP packet header, what is the value in the upper layer protocol field?



The IP packet header contains the upper layer protocol field that states ICMP(1)

### How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

There are 28 bytes in the header, and the payload contains the length – header size bytes. This calculates to 28 bytes for the payload

. 

### Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

There has been no fragmentation here since we do not see any IPv4 fragments in either of the previous screen shots.

### Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

There are two fields in the datagram that change from one datagram to another. The Time to Live field is changing between IP datagrams and so is the identification field.

### Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

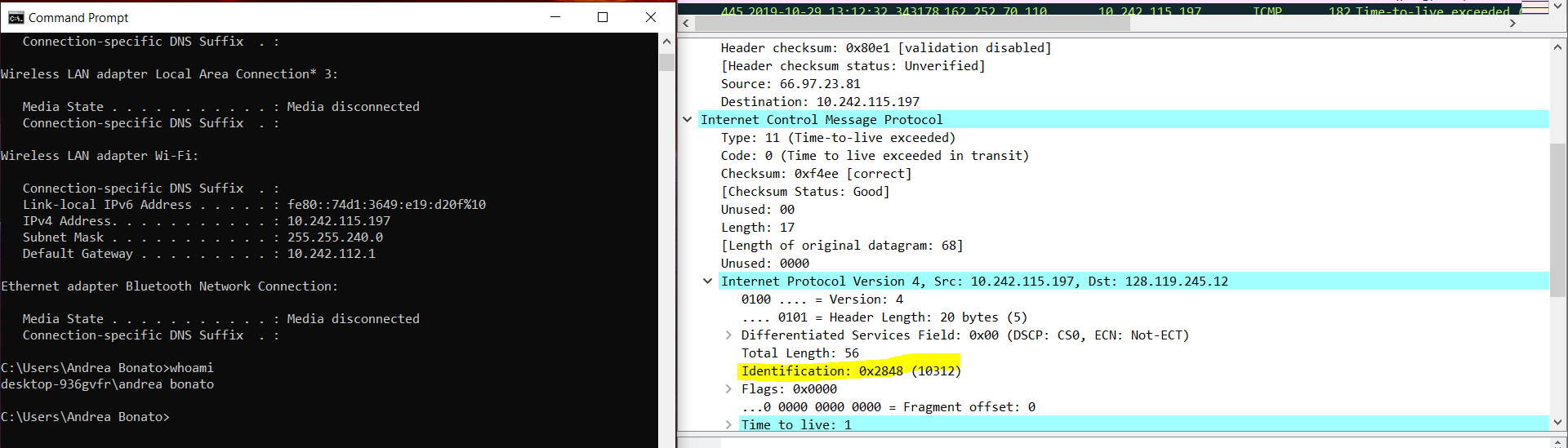
There are several fields that stay constant from one datagram to another. One field is the version. In every datagram, IPv4 was used. Similarly, because of IPv4, the length of the header itself remains the same between each datagram. Finally, the source and destination IP address remain the same between the client and the host.

The fields that change are the identification field and the time to live field. This is how Wireshark are able to trace the packets.

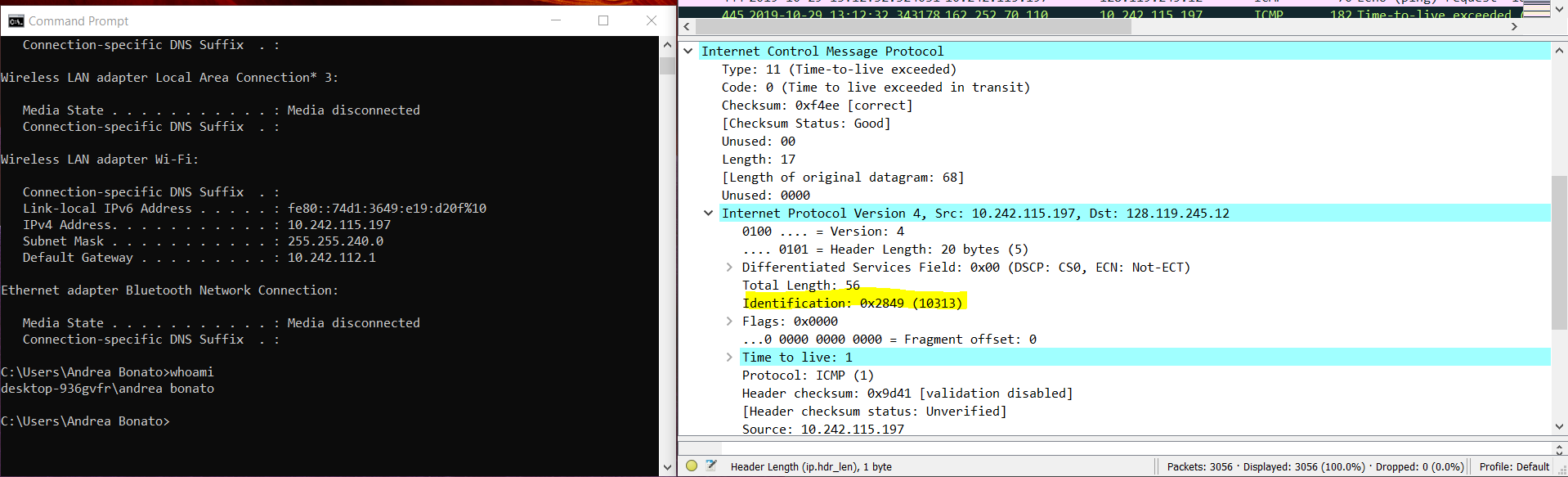
### Describe the pattern you see in the values in the Identification field of the IP datagram. Next find the series of ICMP TTL exceeded replies sent to your computer by the nearest router.

The values in the identification field seem to be increasing and incrementing per datagram. The following images will be the comparison of two datagrams and their identifications.

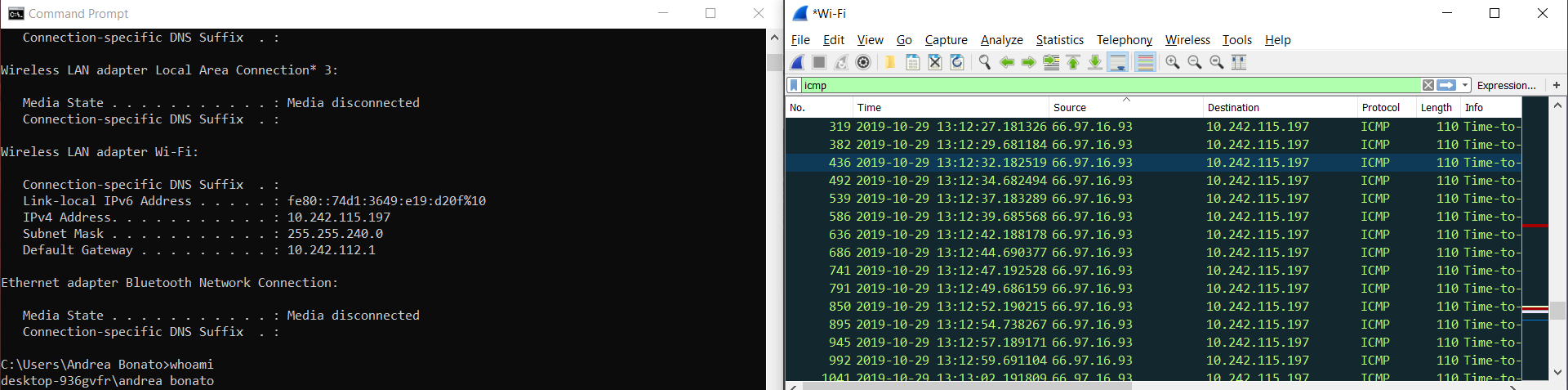
Datagram 1



Datagram 2



The following image is a list of time limit exceeded with the closest router.



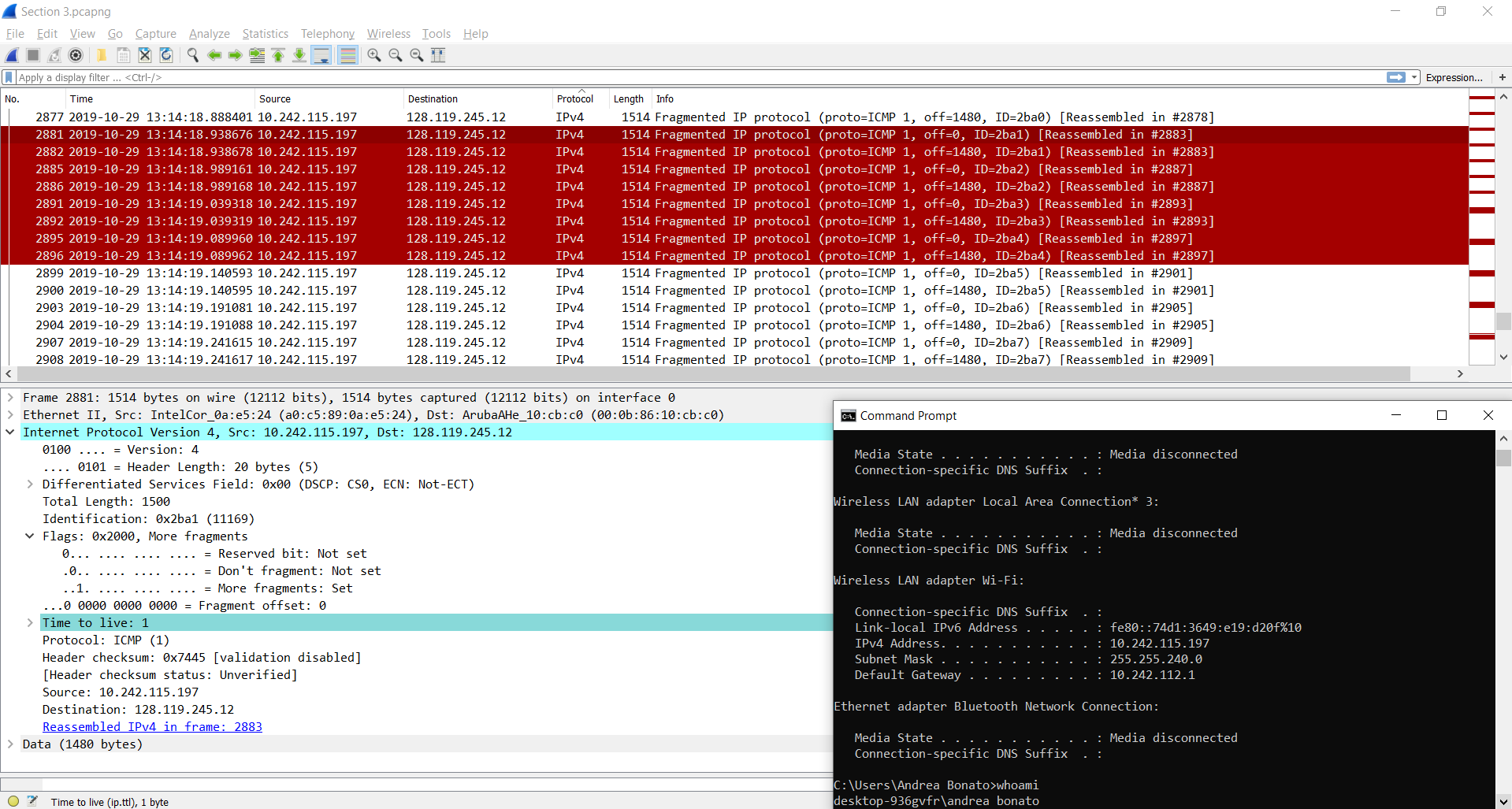
### What is the value in the Identification field and the TTL field?

The value in the identification field is 10313 as shown in the screen shots. The TTL field in this example is 1.

### Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

In this test, the identification values increment and increase between each ICMP TTL-Exceeded reply. The TTL field varies between time and does cycle through 1 on multiple occasions.

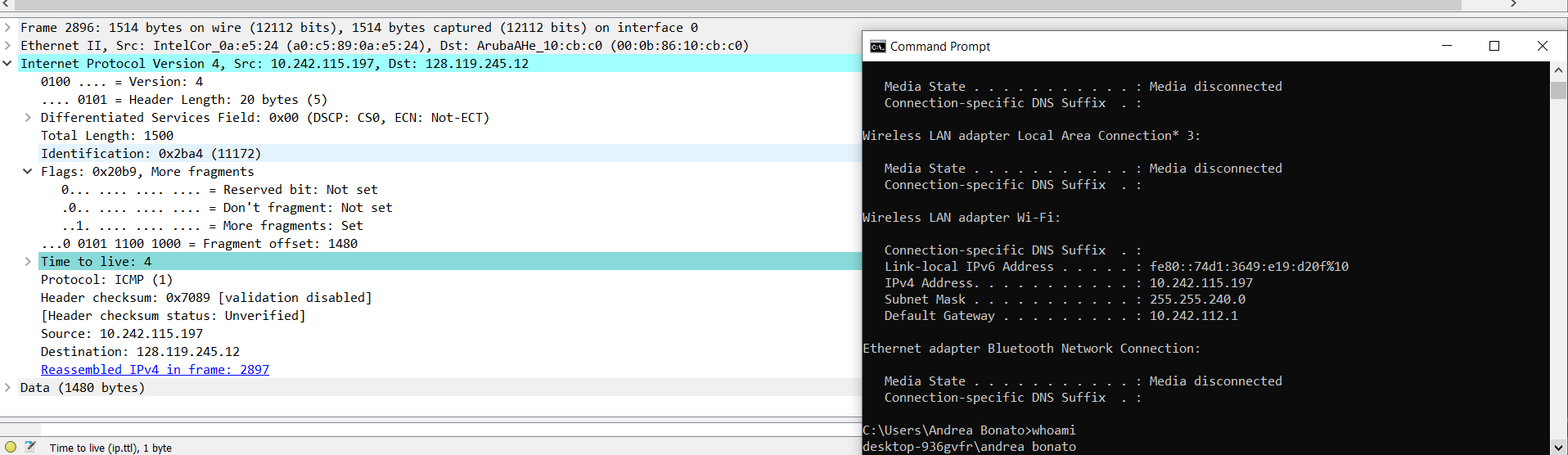
### Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in ping plotter to be 2000. Has that message been fragmented across more than one IP datagram?

As shown in the following image. There is the presence of IPv4 fragments. Wireshark shows the breakdown of the different sequences and segments and it generally resembles fragmented datagrams. 

### Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

As shown in the image above, the flag that states that More fragments are “set” is how the header explains that it is a fragment and there are more of them. The fragment offset is equal to 0 which indicates that it is the first fragment.

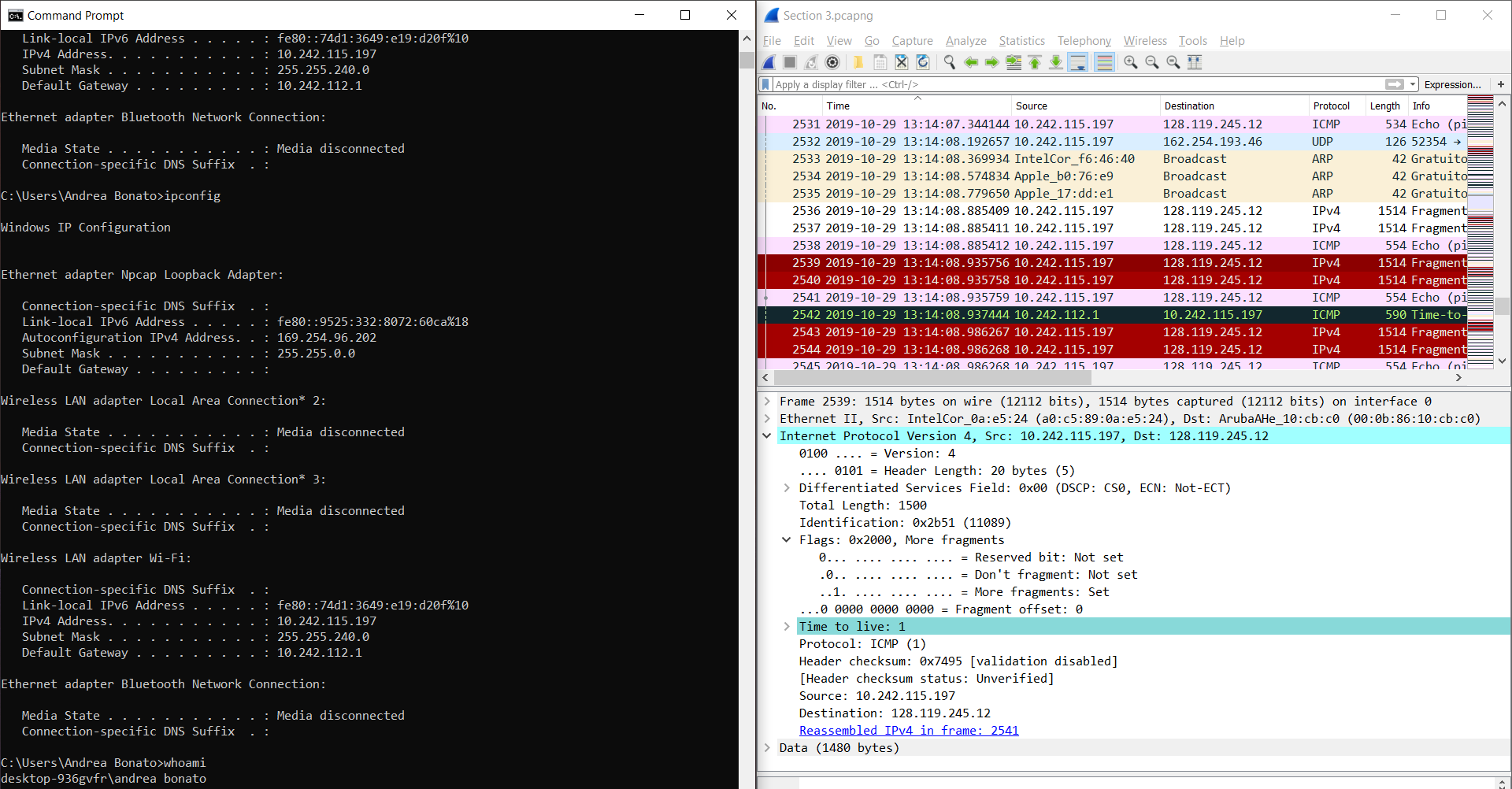
### Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?



As shown above, the fragment offset is 1480. This means that it is the second fragment. The More Fragment flag still says set, that means that there more fragments available. If it was set to “clear” then it would be the last fragment

### What fields change in the IP header between the first and second fragment? Now find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in ping plotter to be 3500. How many fragments were created from the original datagram? What fields change in the IP header among the fragments?

The total length, and the fragment offset are the only two values that changed between the first and second fragment.



As shown in the screen shot above, there were 3 total fragments that were made. Compared to the first fragment, the other fragments have different fragment offsets. However, all fragments seem to have different lengths, and other offsets.