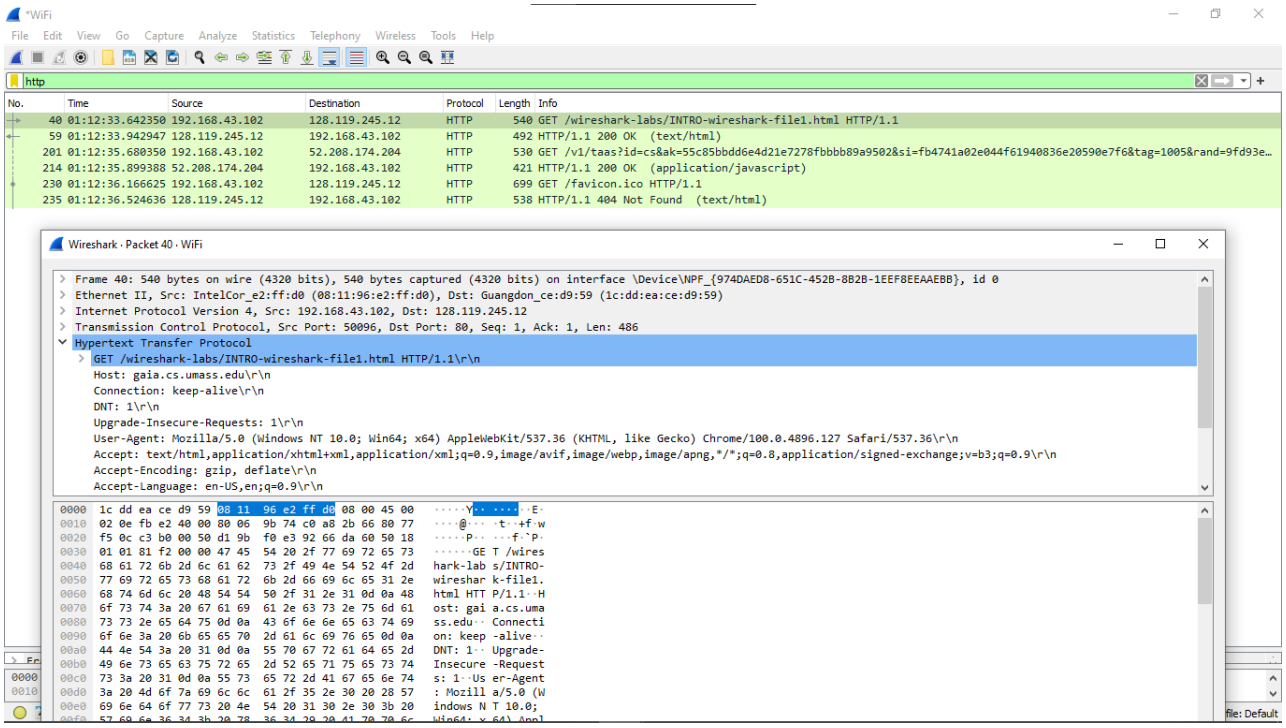
Wireshark Lab : Intro



1. **List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.**

   http , DNS , TCP

2. **How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)**

   Ans: **0.300597 seconds** or **300 ms**

3. **What is the Internet address of the gaia.cs.umass.edu (also known as wwwnet.cs.umass.edu)? What is the Internet address of your computer?**

   Internet address of the gaia.cs.umass.edu : **128.119.245.12**

4. **Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select Print from the Wireshark File command menu, and select the "Selected Packet Only" and "Print as displayed" radial buttons, and then click OK.**

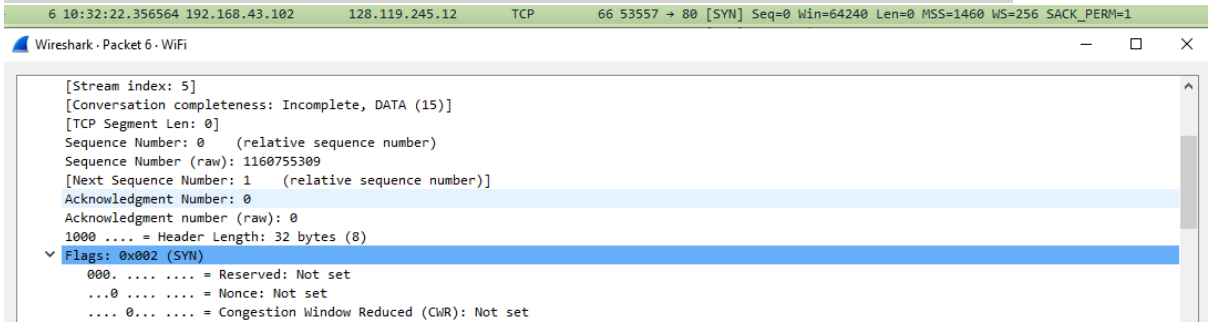
```
No.     Time               Source                Destination           Protocol Length Info
     40 01:12:33.642350    192.168.43.102        128.119.245.12        HTTP     540    GET /wireshark-labs/INTRO-wireshark-
file1.html HTTP/1.1
Frame 40: 540 bytes on wire (4320 bits), 540 bytes captured (4320 bits) on interface
\Device\NPF_{974DAED8-651C-452B-8B2B-1EEF8EEAAEBB}, id 0
Ethernet II, Src: IntelCor_e2:ff:d0 (08:11:96:e2:ff:d0), Dst: Guangdon_ce:d9:59 (1c:dd:ea:ce:d9:59)
Internet Protocol Version 4, Src: 192.168.43.102, Dst: 128.119.245.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 526
    Identification: 0xfbe2 (64482)
    Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x9b74 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.43.102
    Destination Address: 128.119.245.12
Transmission Control Protocol, Src Port: 50096, Dst Port: 80, Seq: 1, Ack: 1, Len: 486
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    DNT: 1\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/
537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
```

```
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    [HTTP request 1/2]
    [Response in frame: 59]
    [Next request in frame: 230]
No.     Time               Source                Destination           Protocol Length Info
     59 01:12:33.942947    128.119.245.12        192.168.43.102        HTTP     492    HTTP/1.1 200 OK  (text/html)
Frame 59: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface
\Device\NPF_{974DAED8-651C-452B-8B2B-1EEF8EEAAEBB}, id 0
Ethernet II, Src: Guangdon_ce:d9:59 (1c:dd:ea:ce:d9:59), Dst: IntelCor_e2:ff:d0 (08:11:96:e2:ff:d0)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.43.102
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 478
    Identification: 0x2fdd (12253)
    Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 42
    Protocol: TCP (6)
    Header Checksum: 0xbdaa [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 128.119.245.12
    Destination Address: 192.168.43.102
Transmission Control Protocol, Src Port: 80, Dst Port: 50096, Seq: 1, Ack: 487, Len: 438
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 20 May 2022 23:20:13 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.28 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Fri, 20 May 2022 05:59:02 GMT\r\n
    ETag: "51-5ddb09f63b5f3"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 81\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/2]
    [Time since request: 0.300597000 seconds]
    [Request in frame: 40]
    [Next request in frame: 230]
    [Next response in frame: 235]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    File Data: 81 bytes
Line-based text data: text/html (3 lines)
```

Wireshark Lab : TCP

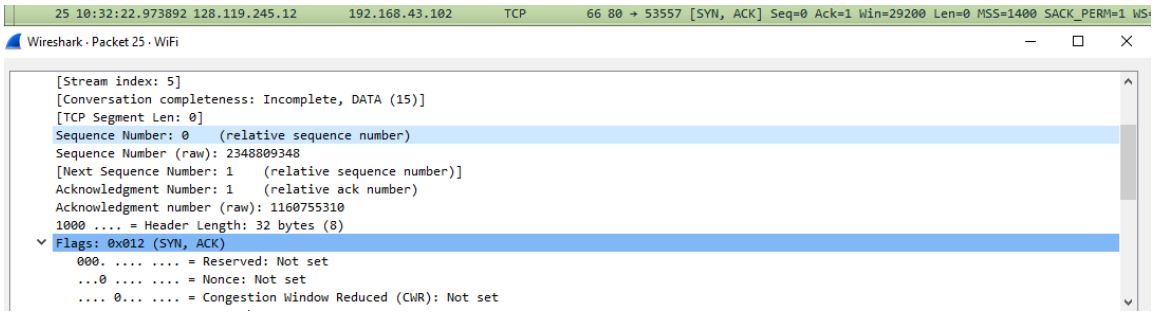
1. **What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?**
   IP address: 192.168.43.102
   TCP port number: 53557
2. **What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?**
   The destination IP address is 128.119.245.12 receiving on port 80
3. **What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?**
   IP address : 172.67.147.40
   TCP port number : 53520

4. **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 a SYN segment?**
   The sequence number of the segment used to initiate the TCP connection is 0. We can see that the message contains a SYN flag indicating that it is a SYN segment.



5. **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 Acknowledgement 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?**

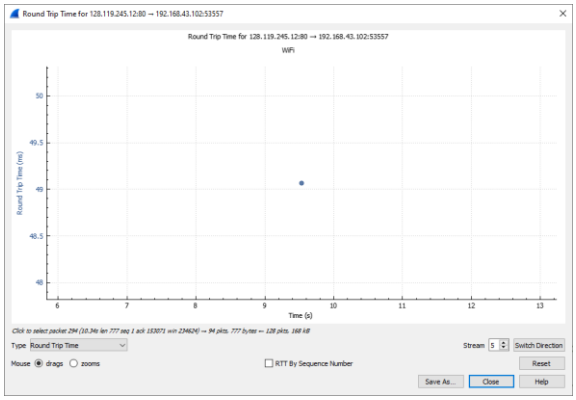
   - The sequence number of the SYNACK segment is 0.
   - The value of the acknowledgement field is 1. This value is determined by the initial sequence number +1.
   - The message carries flags that show it to be a SYN ACK message



6. **What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.**

   The sequence number of the TCP segment containing the HTTP Post Command is 149571.

7. **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 (including the segment containing the HTTP POST)? 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 value for each of the six segments? What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 239 for all subsequent segments. window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph.**

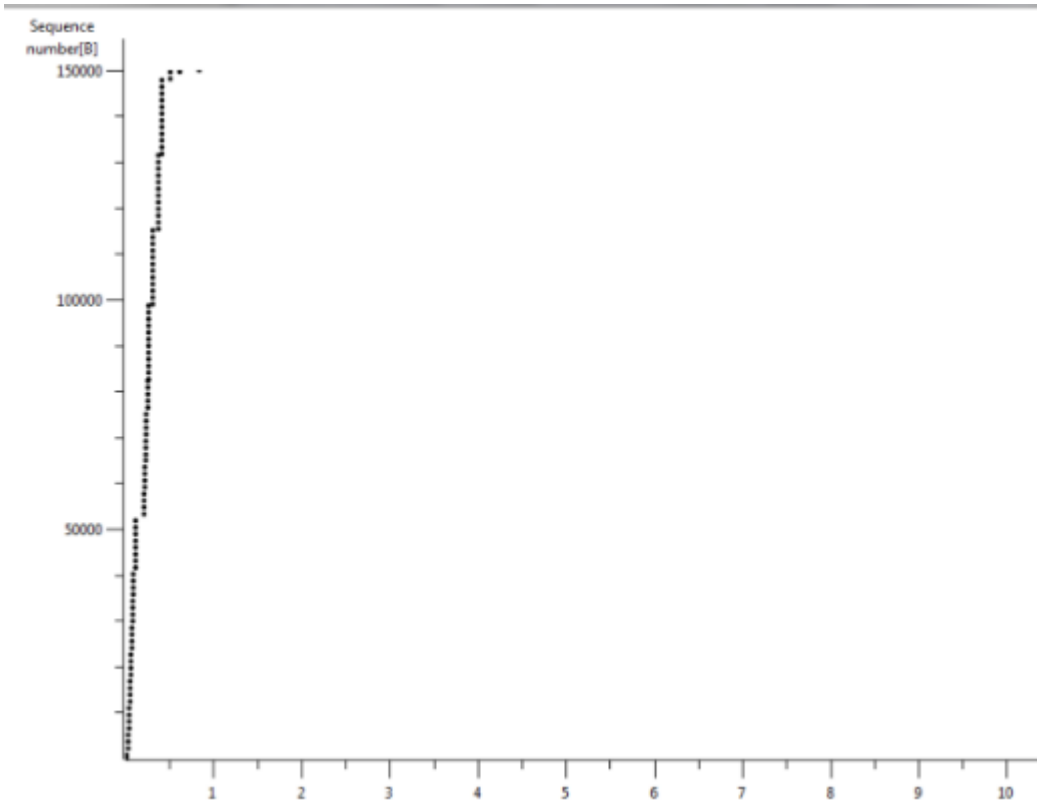


8. **What is the length of each of the first six TCP segments?**
   The length of each of the first TCP segment is 1360
9. **What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?**
   The minimum amount of available buffer space is listed as 65535. The sender is never throttled because we never reach full capacity of the window
10. **Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question? YES**

```
105 10:32:26.771972 192.168.43.102      128.119.245.12       TCP      1414 [TCP Retransmission] 53557 → 80 [ACK] Seq=11638 Ack=1 Win=65792 Len=1360
106 10:32:26.771972 192.168.43.102      128.119.245.12       TCP      1414 [TCP Retransmission] 53557 → 80 [ACK] Seq=12998 Ack=1 Win=65792 Len=1360
```

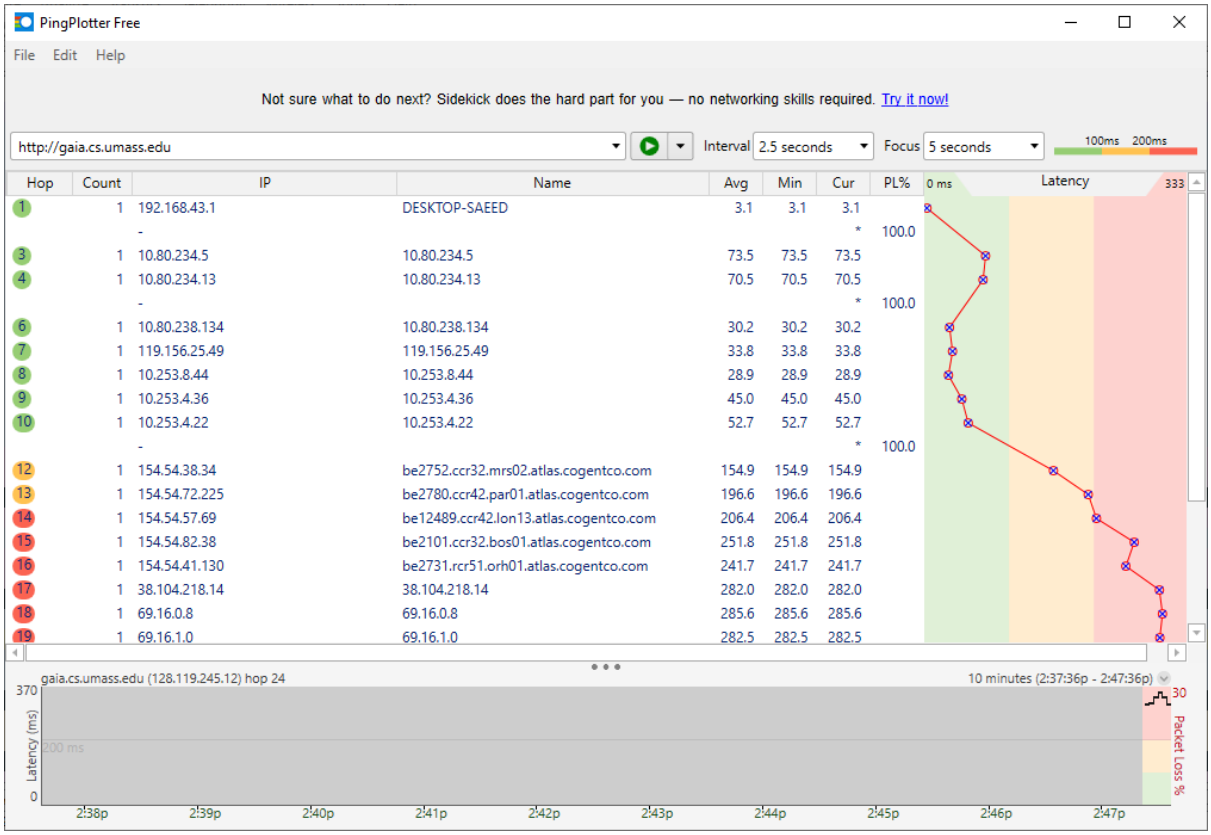
11. **How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).**
    The receiver is typically acking 432 bits. There are cases where the receiver acks every other segment. This is shown when more than one ack occurs in a row.
12. **What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.**
    The throughput can be calculated by using the value of the last ack(149,629)- the first sequence number(1) divided by the time since first frame (1.6) = 93517.6 bps.

13. **Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart 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.**
    The TCP slowstart phase begins at just above seq number 5000, and ends just before sequence number 10000. Congestion avoidance takes over at 10000.

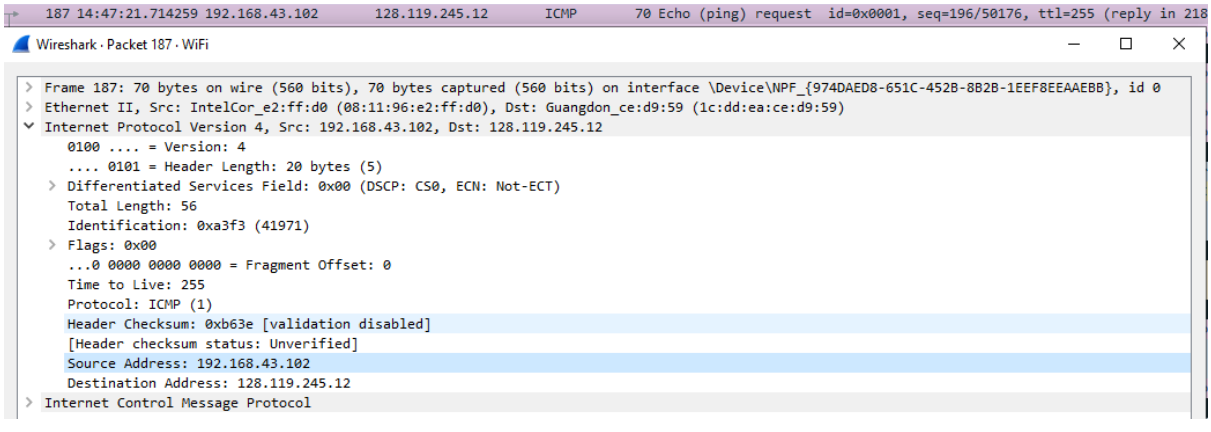


14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu

Wireshark Lab : IP



1. **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?**



IP address of computer: 192.168.43.102

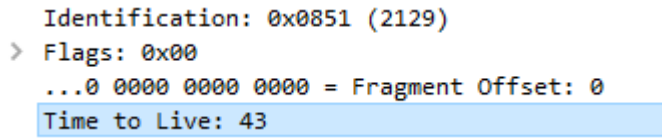
2. **Within the IP packet header, what is the value in the upper layer protocol field?**
The value of the upper layer protocol field is ICMP (0X01)

3. **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 20 bytes in the IP header which leaves 36 bytes for the payload of the IP datagram because we were sending a packet of length 56 bytes.
4. **Has this IP datagram been fragmented? Explain how you determined whether the datagram has been fragmented.**
   The fragment offset is set to 0, therefore, the packet has not been fragmented
5. **Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?**
   The header checksum and the Identification changes from each datagram to the next.
6. **Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?**
   Fields that stay constant:
   - **Version(IPv4)**
   - **Length of header**
   - **Source IP(sending from same place)**
   - **Destination IP(contacting same site)**
   - **Upper layer protocol(always using ICMP)**

   Fields that must stay constant:
   - **Same as above**

   The fields that must change are:
   - **The header checksum (header changes)**
   - **Identification(to verify packets)**

7. **Describe the pattern you see in the values in the Identification field of the IP datagram**
   The pattern in the identification field is that the field increases by one in each strand of echo requests
8. **What is the value in the Identification field and the TTL field?**



9. **Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?**
   - **The Identification field changes from all of the replies because this field has to have a unique value. If they(2 or more replies) have the same value then the replies must be fragments of a bigger packet.**
   - **The TLL field does not change because the time to live to the first hop router is always the same.**
10. **Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram? [Note: if you find your packet has not been fragmented, you should download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the ipethereal-trace-1packet trace. If your**

computer has an Ethernet interface, a packet size of 2000 should cause fragmentation.3]

Yes, that message has been fragmented across more than one IP datagram

11. 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?

    The fact that the flag is set for more segments shows that the the datagram has been fragmented (see above).The fragment offset is set to 0 indicating that this is the first fragment rather than a latter fragment where that value is is set to (1480). The datagram has a total length of 1500.

12. 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?

    The second fragment is obvious because it now has a fragment offset of 1480. There are no more fragments because it no longer has a flag set for more fragments

13. What fields change in the IP header between the first and second fragment?

    The fields that change are

    - Length
    - Flags Set
    - Fragment offset
    - header checksum

14. How many fragments were created from the original datagram?

    After switching , 3 fragments are created

15. What fields change in the IP header among the fragments?

    The fields that change are the fragment offset (0, 1480, 2960) and checksum. The first 2 packets also have lengths of 1500 and more fragments flags set, while the last fragment is shorter (540) and does not have a flag set.