CS5700 - Computer Networking

Module 8 Quiz
Semaa Amin
Question 2 in collaboration with:
Bo Song
JiaHui He

Module 8 Quiz

Module 8 Quiz is due at 1:00PM on Monday November 1, and will be submitted on Canvas. **To do this, upload a PDF file to Assignment "Module 8 quiz".** Your solution to the individual question should be different than other students submissions. Your solution to the group question should be identical to the submission of your group mates.

There will be no group discussions in class for this quiz.

This Quiz will be marked out of 10, and is worth 1% of the final course grade.

Question #1 *Individual Question*

Go to the textbook companion website at the following URL:
https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/

Then go to VideoNotes and watch the video **Traceroute** under chapter 1.

On your laptop, open a Command Prompt window. (Do a google search if you need help in opening a command prompt on your laptop)

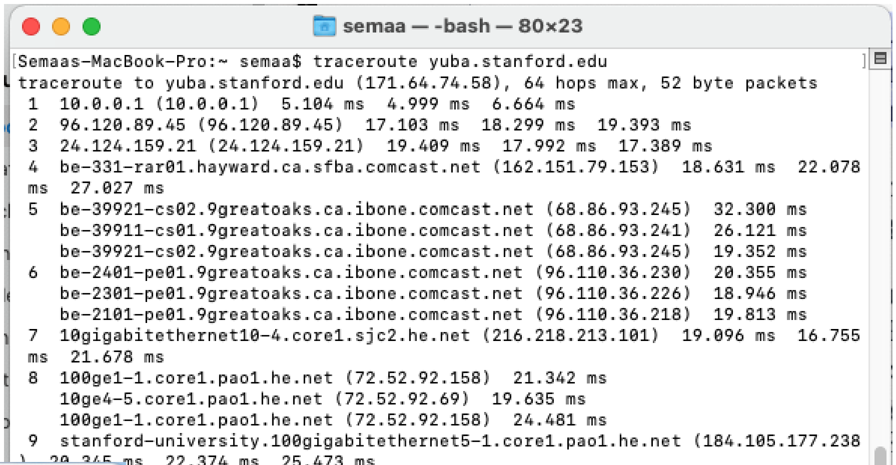
To run the traceroute program, you may want to type **tracert** or **traceroute**, followed by the hostname.
Example: tracert gaia.cs.umass.edu

For each host you selected in parts a and b,

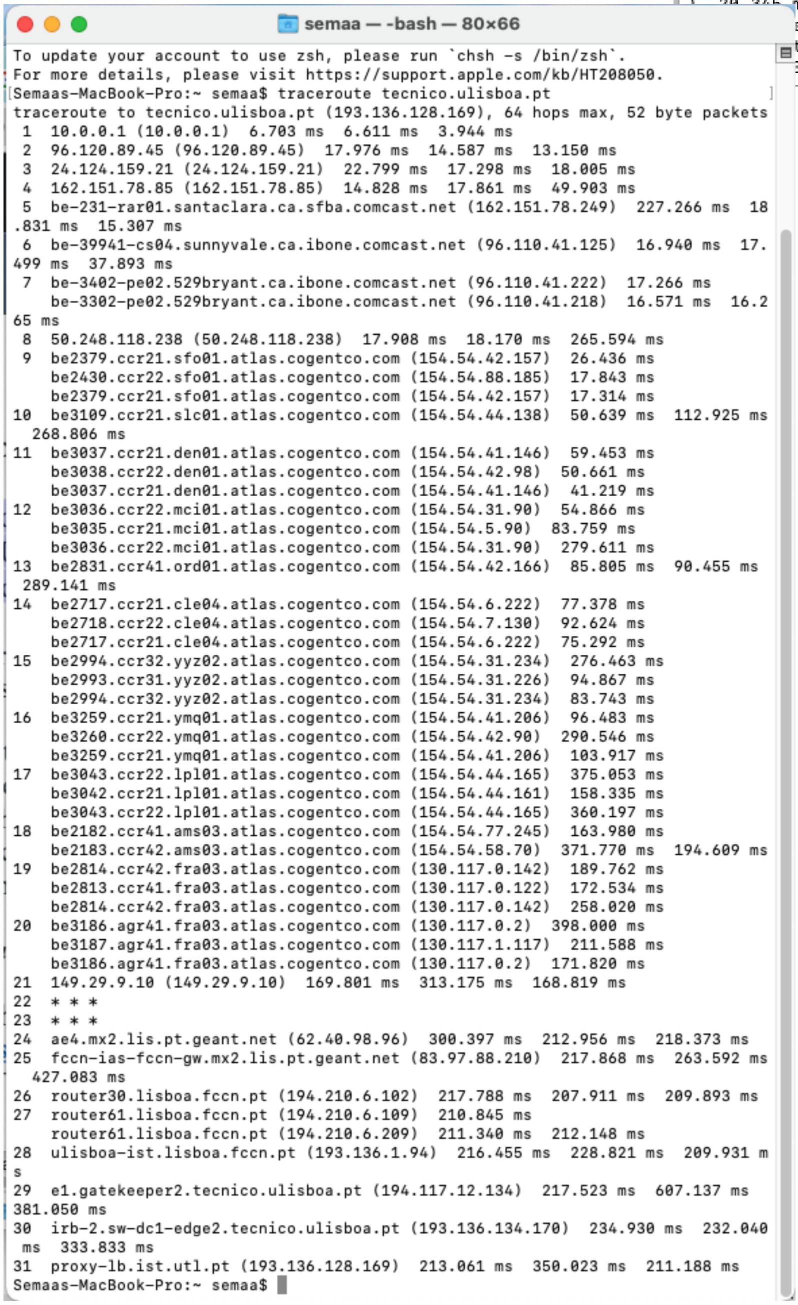
- give its IP address as shown in your traceroute messages
- paste a screenshot of your command window showing the routers between your computer and the university you chose
- give the RTT in msec

Run the traceroute program on:
a) a university in the US (other than the example used in the video)

- Stanford University - yuba.stanford.edu
    - IP: 171.64.74.58
    - →
    - RTT: average of 23.057ms, 24.228ms, and 23.349ms = 23.545ms



```
semaa — -bash — 80×23
Semaas-MacBook-Pro:~ semaa$ traceroute yuba.stanford.edu
traceroute to yuba.stanford.edu (171.64.74.58), 64 hops max, 52 byte packets
 1  10.0.0.1 (10.0.0.1)  5.104 ms  4.999 ms  6.664 ms
 2  96.120.89.45 (96.120.89.45)  17.103 ms  18.299 ms  19.393 ms
 3  24.124.159.21 (24.124.159.21)  19.409 ms  17.992 ms  17.389 ms
 4  be-331-rar01.hayward.ca.sfba.comcast.net (162.151.79.153)  18.631 ms  22.078
ms  27.027 ms
 5  be-39921-cs02.9greatoaks.ca.ibone.comcast.net (68.86.93.245)  32.300 ms
    be-39911-cs01.9greatoaks.ca.ibone.comcast.net (68.86.93.241)  26.121 ms
    be-39921-cs02.9greatoaks.ca.ibone.comcast.net (68.86.93.245)  19.352 ms
 6  be-2401-pe01.9greatoaks.ca.ibone.comcast.net (96.110.36.230)  20.355 ms
    be-2301-pe01.9greatoaks.ca.ibone.comcast.net (96.110.36.226)  18.946 ms
    be-2101-pe01.9greatoaks.ca.ibone.comcast.net (96.110.36.218)  19.813 ms
 7  10gigabitethernet10-4.core1.sjc2.he.net (216.218.213.101)  19.096 ms  16.755
ms  21.678 ms
 8  100ge1-1.core1.pao1.he.net (72.52.92.158)  21.342 ms
    10ge4-5.core1.pao1.he.net (72.52.92.69)  19.635 ms
    100ge1-1.core1.pao1.he.net (72.52.92.158)  24.481 ms
 9  stanford-university.100gigabitethernet5-1.core1.pao1.he.net (184.105.177.238
)  20.345 ms  22.374 ms  25.473 ms
          est-rtr-vl12.sunet (171.66.0.238)  23.442 ms  18.300 ms  22.332 ms
          tanford.edu (171.64.74.58)  23.057 ms  24.228 ms  23.349 ms
          Book-Pro:~ semaa$
```



```
semaa — -bash — 80×66
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Semaas-MacBook-Pro:~ semaa$ traceroute tecnico.ulisboa.pt
traceroute to tecnico.ulisboa.pt (193.136.128.169), 64 hops max, 52 byte packets
 1  10.0.0.1 (10.0.0.1)  6.703 ms  6.611 ms  3.944 ms
 2  96.120.89.45 (96.120.89.45)  17.976 ms  14.587 ms  13.150 ms
 3  24.124.159.21 (24.124.159.21)  22.799 ms  17.298 ms  18.005 ms
 4  162.151.78.85 (162.151.78.85)  14.828 ms  17.861 ms  49.903 ms
 5  be-231-rar01.santaclara.ca.sfba.comcast.net (162.151.78.249)  227.266 ms  18
.831 ms  15.307 ms
 6  be-39941-cs04.sunnyvale.ca.ibone.comcast.net (96.110.41.125)  16.940 ms  17.
499 ms  37.893 ms
 7  be-3402-pe02.529bryant.ca.ibone.comcast.net (96.110.41.222)  17.266 ms
    be-3302-pe02.529bryant.ca.ibone.comcast.net (96.110.41.218)  16.571 ms  16.2
65 ms
 8  50.248.118.238 (50.248.118.238)  17.908 ms  18.170 ms  265.594 ms
 9  be2379.ccr21.sfo01.atlas.cogentco.com (154.54.42.157)  26.436 ms
    be2430.ccr22.sfo01.atlas.cogentco.com (154.54.88.185)  17.843 ms
    be2379.ccr21.sfo01.atlas.cogentco.com (154.54.42.157)  17.314 ms
10  be3109.ccr21.slc01.atlas.cogentco.com (154.54.44.138)  50.639 ms  112.925 ms
  268.806 ms
11  be3037.ccr21.den01.atlas.cogentco.com (154.54.41.146)  59.453 ms
    be3038.ccr22.den01.atlas.cogentco.com (154.54.42.98)  50.661 ms
    be3037.ccr21.den01.atlas.cogentco.com (154.54.41.146)  41.219 ms
12  be3036.ccr22.mci01.atlas.cogentco.com (154.54.31.90)  54.866 ms
    be3035.ccr21.mci01.atlas.cogentco.com (154.54.5.90)  83.759 ms
    be3036.ccr22.mci01.atlas.cogentco.com (154.54.31.90)  279.611 ms
13  be2831.ccr41.ord01.atlas.cogentco.com (154.54.42.166)  85.805 ms  90.455 ms
 289.141 ms
14  be2717.ccr21.cle04.atlas.cogentco.com (154.54.6.222)  77.378 ms
    be2718.ccr22.cle04.atlas.cogentco.com (154.54.7.130)  92.624 ms
    be2717.ccr21.cle04.atlas.cogentco.com (154.54.6.222)  75.292 ms
15  be2994.ccr32.yyz02.atlas.cogentco.com (154.54.31.234)  276.463 ms
    be2993.ccr31.yyz02.atlas.cogentco.com (154.54.31.226)  94.867 ms
    be2994.ccr32.yyz02.atlas.cogentco.com (154.54.31.234)  83.743 ms
16  be3259.ccr21.ymq01.atlas.cogentco.com (154.54.41.206)  96.483 ms
    be3260.ccr22.ymq01.atlas.cogentco.com (154.54.42.90)  290.546 ms
    be3259.ccr21.ymq01.atlas.cogentco.com (154.54.41.206)  103.917 ms
17  be3043.ccr22.lpl01.atlas.cogentco.com (154.54.44.165)  375.053 ms
    be3042.ccr21.lpl01.atlas.cogentco.com (154.54.44.161)  158.335 ms
    be3043.ccr22.lpl01.atlas.cogentco.com (154.54.44.165)  360.197 ms
18  be2182.ccr41.ams03.atlas.cogentco.com (154.54.77.245)  163.980 ms
    be2183.ccr42.ams03.atlas.cogentco.com (154.54.58.70)  371.770 ms  194.609 ms
19  be2814.ccr42.fra03.atlas.cogentco.com (130.117.0.142)  189.762 ms
    be2813.ccr41.fra03.atlas.cogentco.com (130.117.0.122)  172.534 ms
    be2814.ccr42.fra03.atlas.cogentco.com (130.117.0.142)  258.020 ms
20  be3186.agr41.fra03.atlas.cogentco.com (130.117.0.2)  398.000 ms
    be3187.agr41.fra03.atlas.cogentco.com (130.117.1.117)  211.588 ms
    be3186.agr41.fra03.atlas.cogentco.com (130.117.0.2)  171.820 ms
21  149.29.9.10 (149.29.9.10)  169.801 ms  313.175 ms  168.819 ms
22  * * *
23  * * *
24  ae4.mx2.lis.pt.geant.net (62.40.98.96)  300.397 ms  212.956 ms  218.373 ms
25  fccn-ias-fccn-gw.mx2.lis.pt.geant.net (83.97.88.210)  217.868 ms  263.592 ms
  427.083 ms
26  router30.lisboa.fccn.pt (194.210.6.102)  217.788 ms  207.911 ms  209.893 ms
27  router61.lisboa.fccn.pt (194.210.6.109)  210.845 ms
    router61.lisboa.fccn.pt (194.210.6.209)  211.340 ms  212.148 ms
28  ulisboa-ist.lisboa.fccn.pt (193.136.1.94)  216.455 ms  228.821 ms  209.931 m
s
29  e1.gatekeeper2.tecnico.ulisboa.pt (194.117.12.134)  217.523 ms  607.137 ms
381.050 ms
30  irb-2.sw-dc1-edge2.tecnico.ulisboa.pt (193.136.134.170)  234.930 ms  232.040
 ms  333.833 ms
31  proxy-lb.ist.utl.pt (193.136.128.169)  213.061 ms  350.023 ms  211.188 ms
Semaas-MacBook-Pro:~ semaa$
```

b) a university in Europe

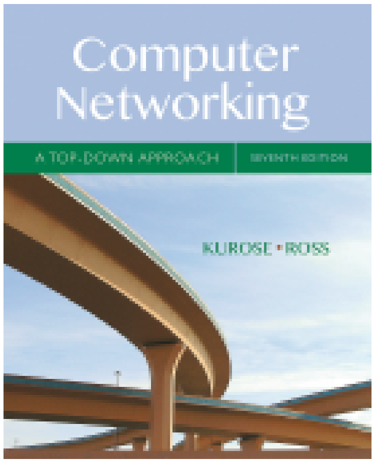
- University of Lisbon - tecnico.ulisboa.pt
- IP: 193.136.128.169
- ←
- RTT average of 213.061 ms 350.023 ms 211.188 ms = 212.437 ms

Complete the “Wireshark Lab: IP” that you can find at the following URL:
http://www-net.cs.umass.edu/wireshark-labs/Wireshark_IP_v7.0.pdf

Write the answers to all questions asked in this lab, and provide screenshots to backup your answers.

Wireshark Lab: IP v7.0

Supplement to *Computer Networking: A Top-Down Approach, 7th ed.,* J.F. Kurose and K.W. Ross



“Tell me and I forget. Show me and I remember. Involve me and I
understand.” Chinese proverb

In this lab, we’ll investigate the IP protocol, focusing on the IP datagram. We’ll do so by analyzing a trace of IP datagrams sent and received by an execution of the traceroute program (the traceroute program itself is explored in more detail in the Wireshark ICMP lab). We’ll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

Before beginning this lab, you’ll probably want to review sections 1.4.3 in the text [1] and section 3.4 of RFC 2151 [ftp://ftp.rfc-editor.org/in-notes/rfc2151.txt] to update yourself on the operation of the traceroute program. You’ll also want to read Section 4.3 in the text, and probably also have RFC 791 [ftp://ftp.rfc-editor.org/in-notes/rfc791.txt] on hand as well, for a discussion of the IP protocol.

1. Capturing packets from an execution of traceroute

In order to generate a trace of IP datagrams for this lab, we’ll use the traceroute program to send datagrams of different sizes towards some destination, *X*. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series of datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1 (actually, RFC 791 says that the router must decrement the TTL by *at least* one). If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. As a result of this behavior, a datagram with a TTL of 1 (sent

[1] References to figures and sections are for the 7th edition of our text, *Computer Networks, A Top-down Approach, 7th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2016.*

by the host executing traceroute) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram sent with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this manner, the host executing traceroute can learn the identities of the routers between itself and destination *X* by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

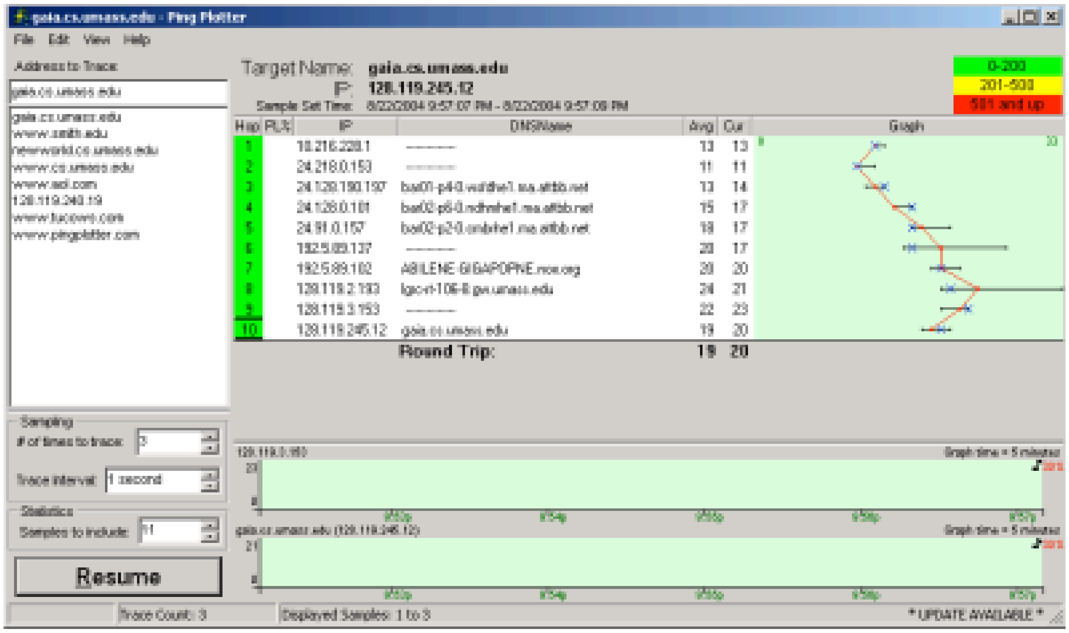
We'll want to run traceroute and have it send datagrams of various lengths.

- **Windows.** The tracert program (used for our ICMP Wireshark lab) provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nicer Windows traceroute program is *pingplotter*, available both in free version and shareware versions at http://www.pingplotter.com. Download and install *pingplotter*, and test it out by performing a few traceroutes to your favorite sites. The size of the ICMP echo request message can be explicitly set in *pingplotter* by selecting the menu item *Edit-> Options->Packet Options* and then filling in the *Packet Size* field. The default packet size is 56 bytes. Once *pingplotter* has sent a series of packets with the increasing TTL values, it restarts the sending process again with a TTL of 1, after waiting *Trace Interval* amount of time. The value of *Trace Interval* and the number of intervals can be explicitly set in *pingplotter*.
- **Linux/Unix/MacOS.** With the Unix/MacOS traceroute command, the size of the UDP datagram sent towards the destination can be explicitly set by indicating the number of bytes in the datagram; this value is entered in the traceroute command line immediately after the name or address of the destination. For example, to send traceroute datagrams of 2000 bytes towards gaia.cs.umass.edu, the command would be:

```
%traceroute gaia.cs.umass.edu 2000
```

Do the following:

- Start up Wireshark and begin packet capture *(Capture->Start)* and then press *OK* on the Wireshark Packet Capture Options screen (we'll not need to select any options here).
- If you are using a Windows platform, start up *pingplotter* and enter the name of a target destination in the "Address to Trace Window." Enter 3 in the "# of times to Trace" field, so you don't gather too much data. Select the menu item *Edit- >Advanced Options->Packet Options* and enter a value of 56 in the *Packet Size* field and then press OK. Then press the Trace button. You should see a *pingplotter* window that looks something like this:



Target Name: gaia.cs.umass.edu
IP: 128.119.245.12
Sample Set Time: 8/22/2004 9:57:07 PM - 8/22/2004 9:57:09 PM

Hop	PL%	IP	DNSName	Avg	Cur
1		10.216.228.1	----------	13	13
2		24.218.0.153	----------	11	11
3		24.128.190.197	bar01-p4-0.wsfdhe1.ma.attbb.net	13	14
4		24.128.0.101	bar02-p6-0.ndhmhe1.ma.attbb.net	15	17
5		24.91.0.157	bar02-p2-0.cmbrhe1.ma.attbb.net	18	17
6		192.5.89.137	----------	20	17
7		192.5.89.102	ABILENE-GIGAPOPNE.nox.org	20	20
8		128.119.2.193	lgcr-rt-106-8.gw.umass.edu	24	21
9		128.119.3.153	----------	22	23
10		128.119.245.12	gaia.cs.umass.edu	19	20
			Round Trip:	19	20

# of times to trace: 3
Trace interval: 1 second
Samples to include: 11
Trace Count: 3
Displayed Samples: 1 to 3

Next, send a set of datagrams with a longer length, by selecting *Edit->Advanced Options->Packet Options* and enter a value of 2000 in the *Packet Size* field and then press OK. Then press the Resume button.

Finally, send a set of datagrams with a longer length, by selecting *Edit->Advanced Options->Packet Options* and enter a value of 3500 in the *Packet Size* field and then press OK. Then press the Resume button.

Stop Wireshark tracing.

- If you are using a Unix or Mac platform, enter three traceroute commands, one with a length of 56 bytes, one with a length of 2000 bytes, and one with a length of 3500 bytes.

  Stop Wireshark tracing.

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author's Windows computers[2]. You may well find it valuable to download this trace even if you've captured your own trace and use it, as well as your own trace, when you explore the questions below.

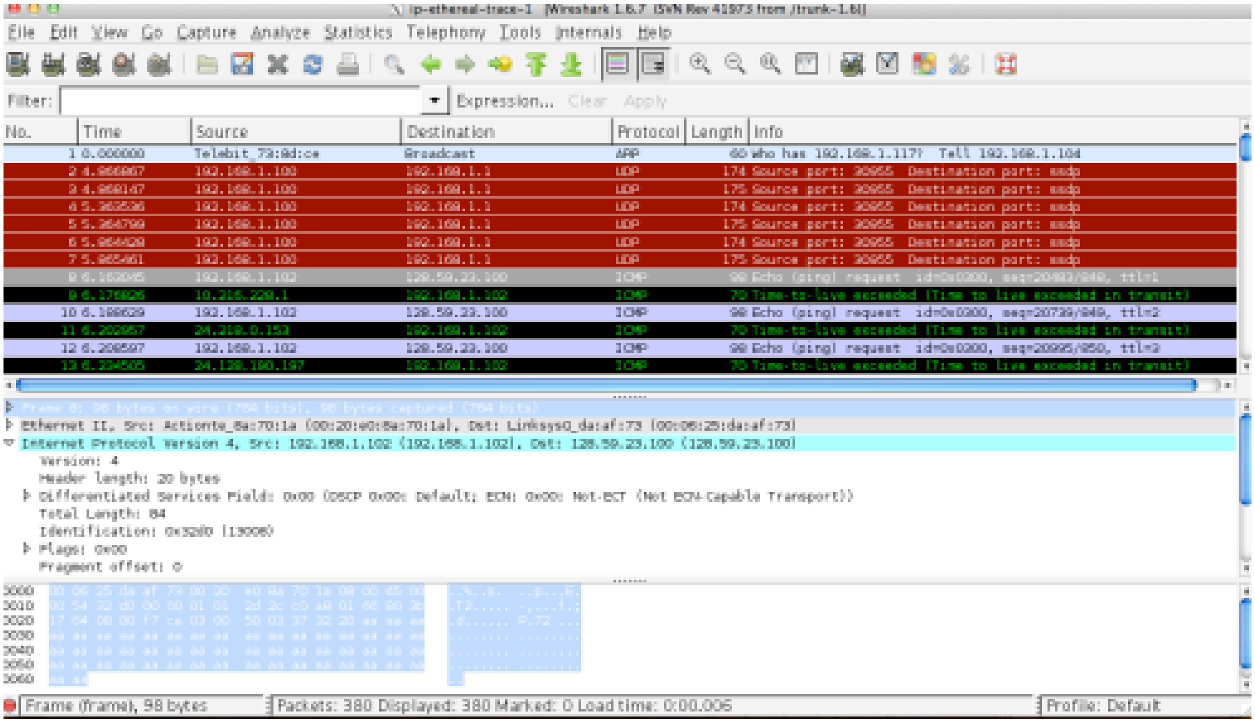
2. A look at the captured trace

In your trace, you should be able to see the series of ICMP Echo Request (in the case of Windows machine) or the UDP segment (in the case of Unix) sent by your

[2] Download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the file *ip ethereal-trace-1*. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the ip-ethereal-trace-1 trace file.

computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. In the questions below, we'll assume you are using a Windows machine; the corresponding questions for the case of a Unix machine should be clear. Whenever possible, when answering a question below you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. When you hand in your assignment, annotate the output so that it's clear where in the output you're getting the information for your answer (e.g., for our classes, we ask that students markup paper copies with a pen, or annotate electronic copies with text in a colored font).To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

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.


ip-ethereal-trace-1 [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.  Time        Source            Destination       Protocol Length Info
  1 0.000000    Telebit_73:8d:ce  Broadcast         ARP        60 Who has 192.168.1.117?  Tell 192.168.1.104
  2 4.866867    192.168.1.100     192.168.1.1       UDP       174 Source port: 30955  Destination port: ssdp
  3 4.868147    192.168.1.100     192.168.1.1       UDP       175 Source port: 30955  Destination port: ssdp
  4 5.363536    192.168.1.100     192.168.1.1       UDP       174 Source port: 30955  Destination port: ssdp
  5 5.364799    192.168.1.100     192.168.1.1       UDP       175 Source port: 30955  Destination port: ssdp
  6 5.864428    192.168.1.100     192.168.1.1       UDP       174 Source port: 30955  Destination port: ssdp
  7 5.865461    192.168.1.100     192.168.1.1       UDP       175 Source port: 30955  Destination port: ssdp
  8 6.163045    192.168.1.102     128.59.23.100     ICMP       98 Echo (ping) request  id=0x0300, seq=20483/848, ttl=1
  9 6.176826    10.216.228.1      192.168.1.102     ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
 10 6.188629    192.168.1.102     128.59.23.100     ICMP       98 Echo (ping) request  id=0x0300, seq=20739/849, ttl=2
 11 6.202957    24.218.0.153      192.168.1.102     ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
 12 6.208597    192.168.1.102     128.59.23.100     ICMP       98 Echo (ping) request  id=0x0300, seq=20995/850, ttl=3
 13 6.234505    24.128.190.197    192.168.1.102     ICMP       70 Time-to-live exceeded (Time to live exceeded in transit)
```

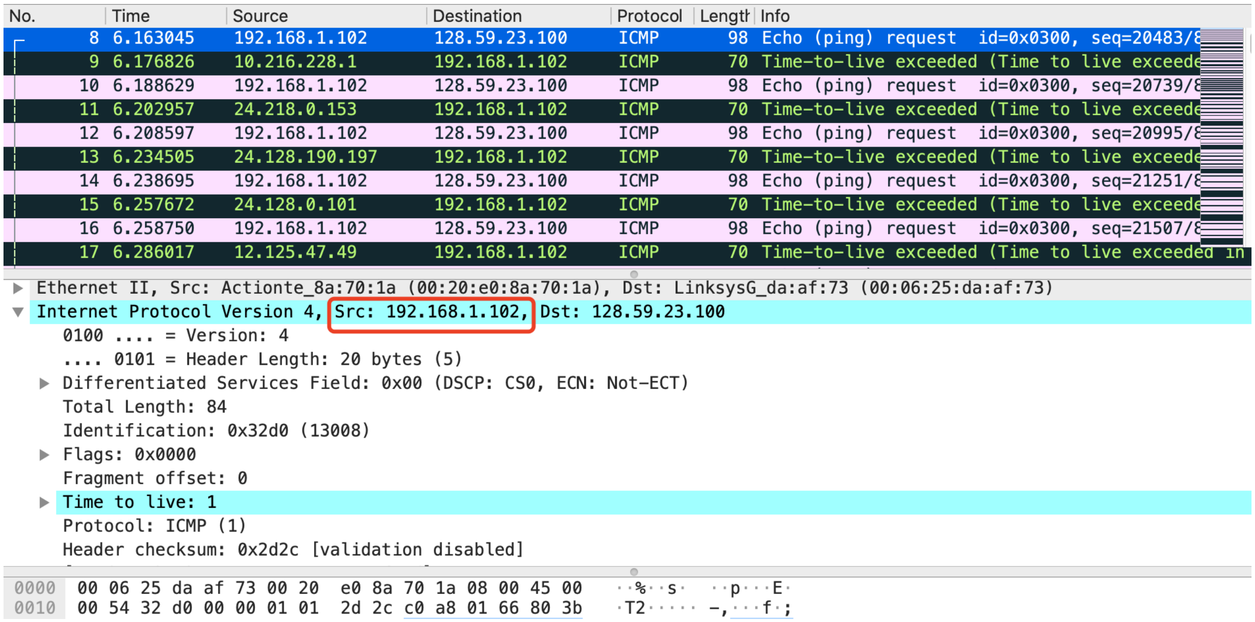
```
▷ Frame 8: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
▷ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▽ Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.59.23.100 (128.59.23.100)
    Version: 4
    Header length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▷ Flags: 0x00
    Fragment offset: 0
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ..%..s. ..p...E.
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   .T2..... -,...f.;
0020  17 64 08 00 f7 ca 03 00  50 03 37 32 20 aa aa aa   .d...... P.72 ...
0030  aa aa aa aa aa aa aa aa  aa aa aa aa aa aa aa aa   ........ ........
0040  aa aa aa aa aa aa aa aa  aa aa aa aa aa aa aa aa   ........ ........
0050  aa aa aa aa aa aa aa aa  aa aa aa aa aa aa aa aa   ........ ........
0060  aa aa                                              ..
```

```
Frame (frame), 98 bytes | Packets: 380 Displayed: 380 Marked: 0 Load time: 0:00.006 | Profile: Default
```

NOTE: Our group used the existing packet trace file '*ip-ethereal-trace-1*' to do the lab.

1.What is the IP address of your computer?



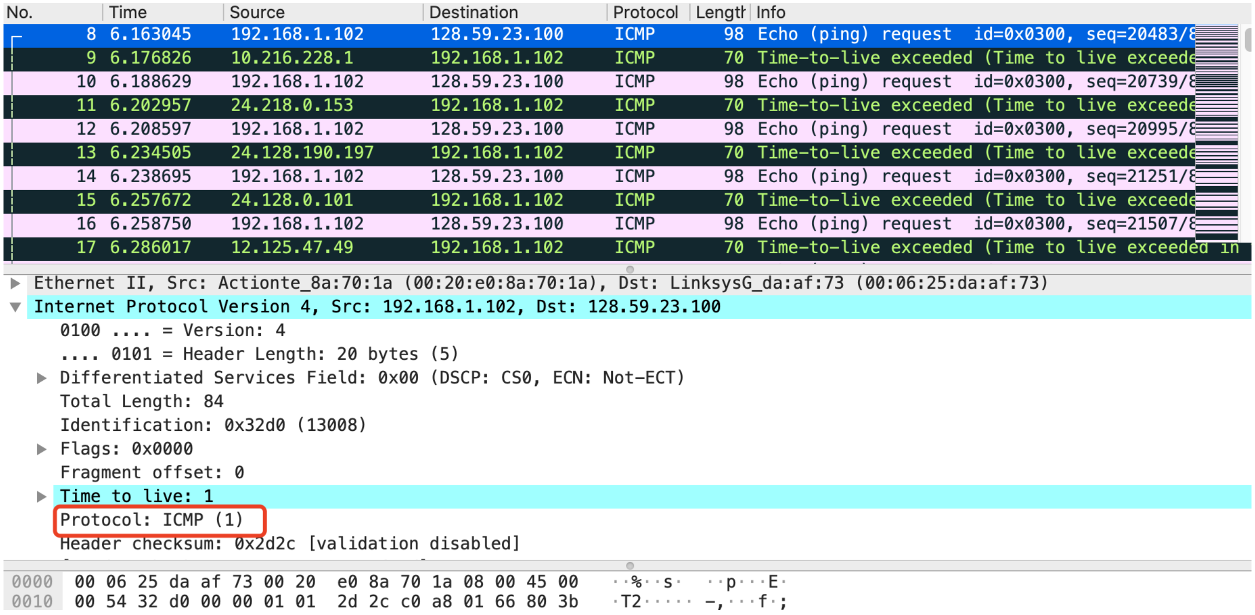
```
No.    Time      Source          Destination     Protocol  Length  Info
  8 6.163045    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20483/8
  9 6.176826    10.216.228.1    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 10 6.188629    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20739/8
 11 6.202957    24.218.0.153    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 12 6.208597    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20995/8
 13 6.234505    24.128.190.197  192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 14 6.238695    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=21251/8
 15 6.257672    24.128.0.101    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 16 6.258750    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=21507/8
 17 6.286017    12.125.47.49    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceeded in

▶ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▼ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▶ Flags: 0x0000
    Fragment offset: 0
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2d2c [validation disabled]

0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   ·T2·····  –,···f·;
```

- IP address: 192.168.1.102

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



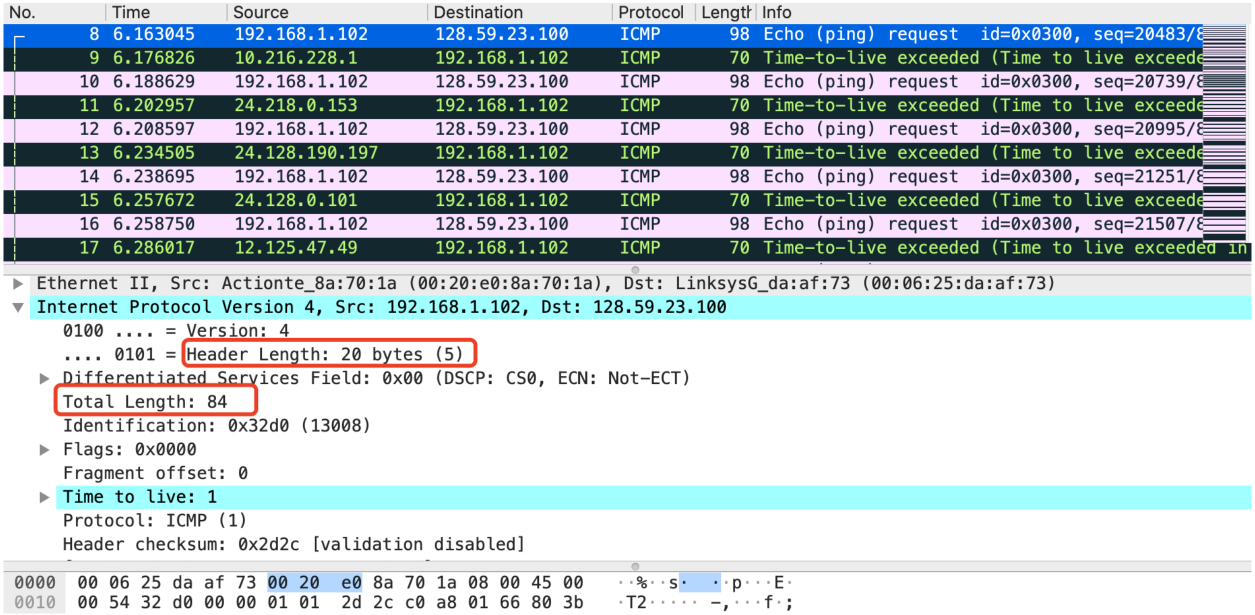
```
No.    Time      Source          Destination     Protocol  Length  Info
  8 6.163045    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20483/8
  9 6.176826    10.216.228.1    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 10 6.188629    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20739/8
 11 6.202957    24.218.0.153    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 12 6.208597    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20995/8
 13 6.234505    24.128.190.197  192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 14 6.238695    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=21251/8
 15 6.257672    24.128.0.101    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
 16 6.258750    192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=21507/8
 17 6.286017    12.125.47.49    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceeded in

▶ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▼ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▶ Flags: 0x0000
    Fragment offset: 0
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2d2c [validation disabled]

0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   ·T2·····  –,···f·;
```

- Within the header, the value in the upper layer protocol field is ICMP (1)

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.



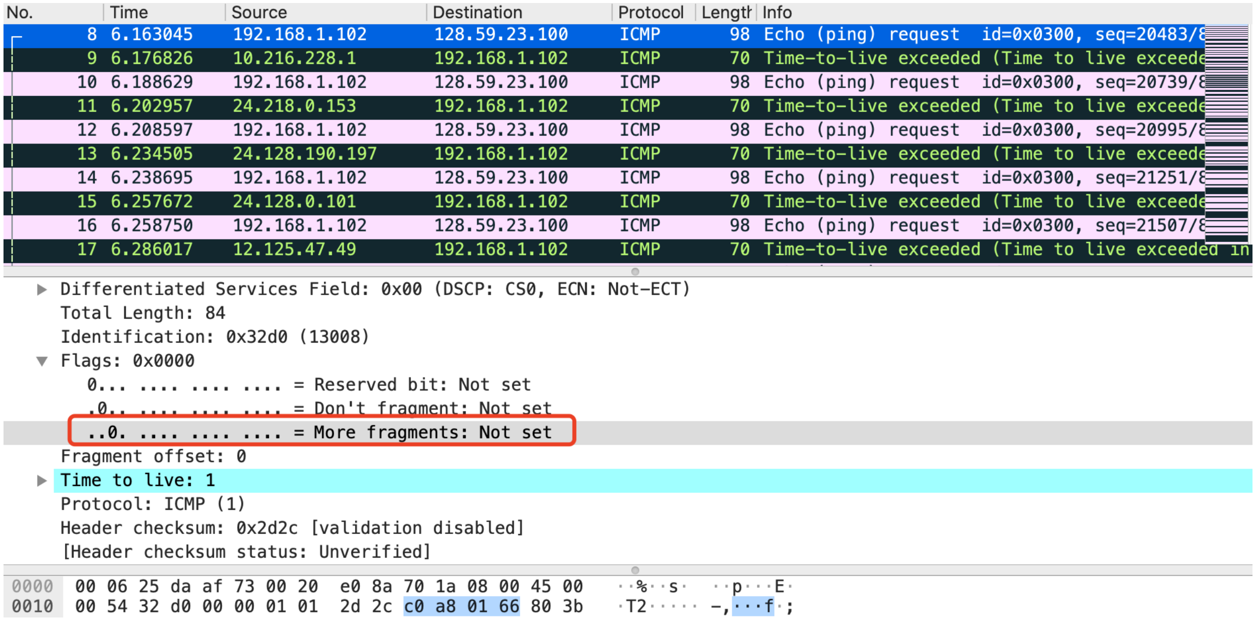
```
No.   Time       Source           Destination     Protocol Length Info
    8 6.163045   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=20483/8
    9 6.176826   10.216.228.1     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   10 6.188629   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=20739/8
   11 6.202957   24.218.0.153     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   12 6.208597   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=20995/8
   13 6.234505   24.128.190.197   192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   14 6.238695   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=21251/8
   15 6.257672   24.128.0.101     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   16 6.258750   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=21507/8
   17 6.286017   12.125.47.49     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceeded in

▶ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▼ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▶ Flags: 0x0000
    Fragment offset: 0
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2d2c [validation disabled]

0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   ·T2····· -,···f·;
```

- the IP header has 20 bytes
- the payload of the IP datagram = 84-20 = 64
- the payload of the IP datagram = total length - IP header

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



```
No.   Time       Source           Destination     Protocol Length Info
    8 6.163045   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=20483/8
    9 6.176826   10.216.228.1     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   10 6.188629   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=20739/8
   11 6.202957   24.218.0.153     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   12 6.208597   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=20995/8
   13 6.234505   24.128.190.197   192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   14 6.238695   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=21251/8
   15 6.257672   24.128.0.101     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceede
   16 6.258750   192.168.1.102    128.59.23.100   ICMP      98 Echo (ping) request  id=0x0300, seq=21507/8
   17 6.286017   12.125.47.49     192.168.1.102   ICMP      70 Time-to-live exceeded (Time to live exceeded in

  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▼ Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2d2c [validation disabled]
    [Header checksum status: Unverified]

0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   ·T2····· -,···f·;
```

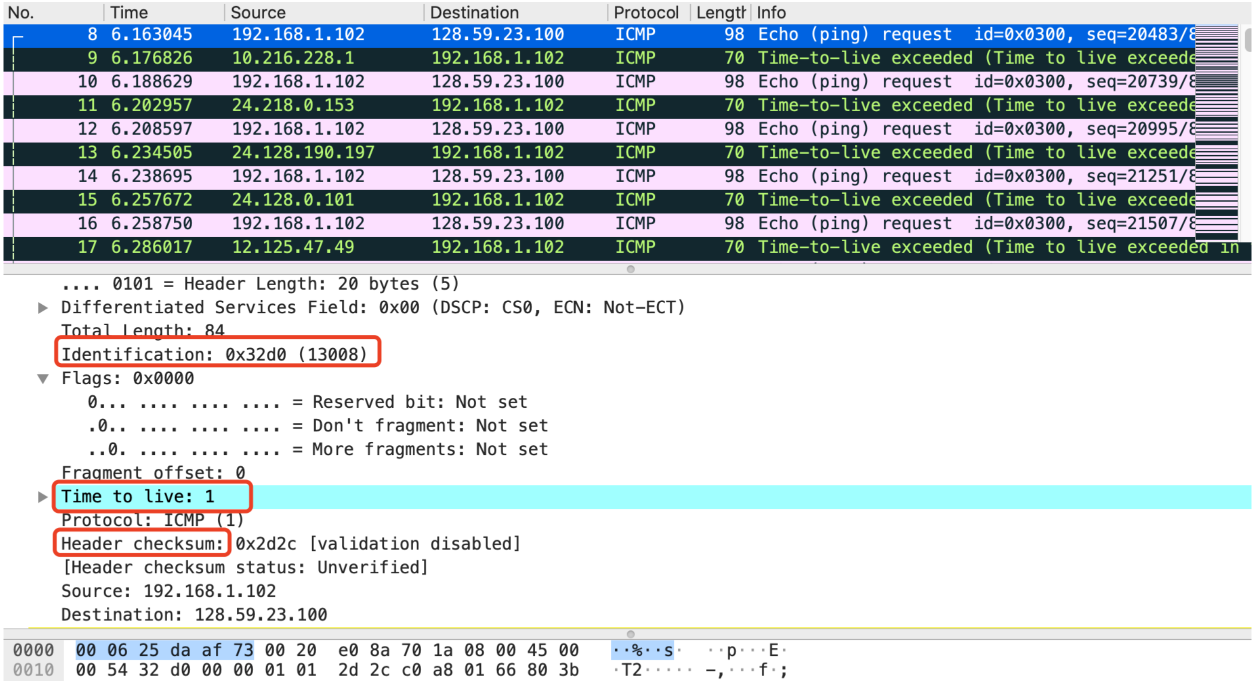
- The more fragments flag is 0, so the data is not fragmented.

Next, sort the traced packets according to IP source address by clicking on the *Source* column header; a small downward pointing arrow should appear next to the word *Source*. If the arrow points up, click on the *Source* column header again.

Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the "details of selected packet header" window. In the "listing of captured packets" window, you should see all of the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your computer) below this first ICMP. Use the down arrow to move through the ICMP messages sent by your computer.

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?

- Identification
- Time to live
- Header checksum



```
No.   Time      Source          Destination     Protocol Length Info
    8 6.163045  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20483/8
    9 6.176826  10.216.228.1    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
   10 6.188629  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20739/8
   11 6.202957  24.218.0.153    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
   12 6.208597  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=20995/8
   13 6.234505  24.128.190.197  192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
   14 6.238695  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=21251/8
   15 6.257672  24.128.0.101    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceede
   16 6.258750  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=21507/8
   17 6.286017  12.125.47.49    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live exceeded in
```

```
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▼ Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2d2c [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.102
    Destination: 128.59.23.100
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   ·T2·····  –,···f·;
```

6.

a) Which fields stay constant? Why?

- Header length - using same IPv4 for all packers
- Version - same for all IPv4 packets
- Source IP - sending from same source
- Destination - sending to same location/destination
- Differentiated Services - all packets are ICMP which use same service class
- Upper Layer Protocol - ICMP packets

b) Which of the fields must stay constant? Why?

- All the fields in (a) for same reasons

c) Which fields must change? Why?

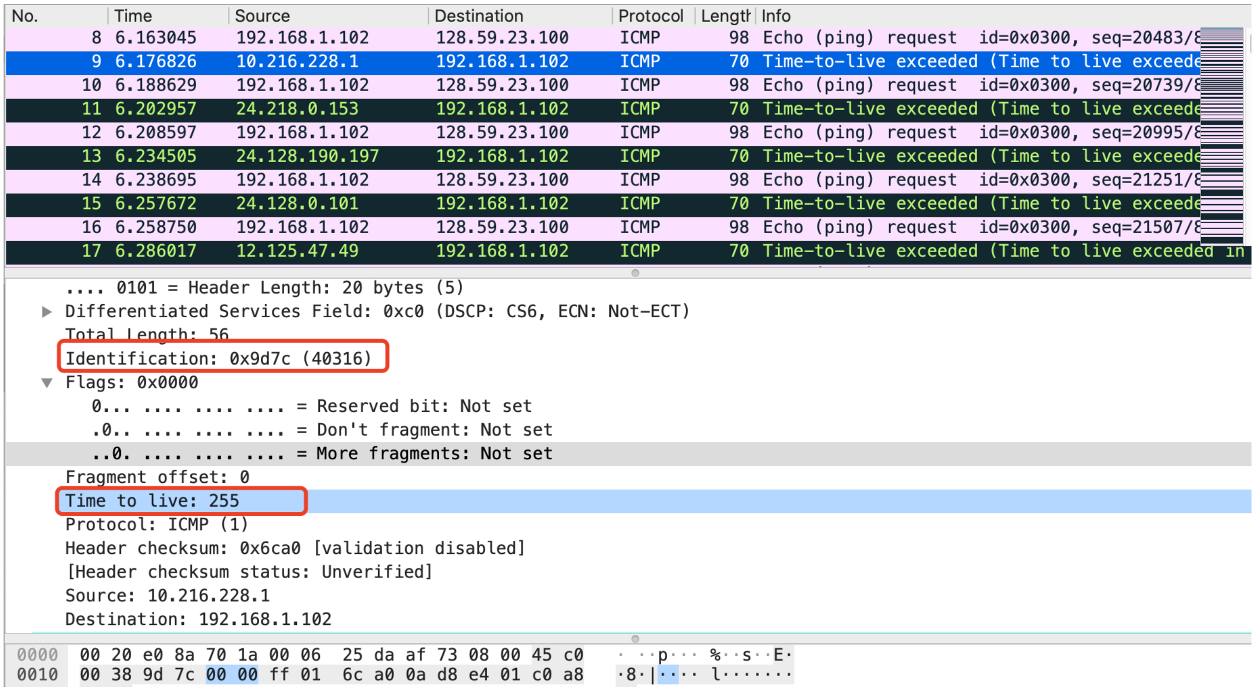
- Identification - to differentiate the varying IP packets
- Time to Live - traceroute changes the next packets
- Header checksum - the checksum must change when the header changes

7. Describe the pattern you see in the values in the Identification field of the IP datagram

- The IP header Identification fields increment with each ICMP Echo (ping) request.

Next (with the packets still sorted by source address) find the series of ICMP TTL exceeded replies sent to your computer by the nearest (first hop) router.

8. What is the value in the Identification field and the TTL field?



```
No.   Time      Source           Destination    Protocol Length Info
   8 6.163045   192.168.1.102    128.59.23.100  ICMP      98 Echo (ping) request  id=0x0300, seq=20483/8
   9 6.176826   10.216.228.1     192.168.1.102  ICMP      70 Time-to-live exceeded (Time to live exceede
  10 6.188629   192.168.1.102    128.59.23.100  ICMP      98 Echo (ping) request  id=0x0300, seq=20739/8
  11 6.202957   24.218.0.153     192.168.1.102  ICMP      70 Time-to-live exceeded (Time to live exceede
  12 6.208597   192.168.1.102    128.59.23.100  ICMP      98 Echo (ping) request  id=0x0300, seq=20995/8
  13 6.234505   24.128.190.197   192.168.1.102  ICMP      70 Time-to-live exceeded (Time to live exceede
  14 6.238695   192.168.1.102    128.59.23.100  ICMP      98 Echo (ping) request  id=0x0300, seq=21251/8
  15 6.257672   24.128.0.101     192.168.1.102  ICMP      70 Time-to-live exceeded (Time to live exceede
  16 6.258750   192.168.1.102    128.59.23.100  ICMP      98 Echo (ping) request  id=0x0300, seq=21507/8
  17 6.286017   12.125.47.49     192.168.1.102  ICMP      70 Time-to-live exceeded (Time to live exceeded in
```

```
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x9d7c (40316)
  ▼ Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x6ca0 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.216.228.1
    Destination: 192.168.1.102
```

```
0000  00 20 e0 8a 70 1a 00 06  25 da af 73 08 00 45 c0   · ··p··· %··s··E·
0010  00 38 9d 7c 00 00 ff 01  6c a0 0a d8 e4 01 c0 a8   ·8·|···· l·······
```

- ID = 40316
- TTL = 255

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 TTL remains unchanged because the TTL for the first hop router is always the same.
- The identification field changes for all the ICMP TTL-exceeded replies because the identification field is a unique value.
- When two or more IP datagrams have the same identification value, then it means that these IP datagrams are fragments of a single large IP datagram.

Fragmentation

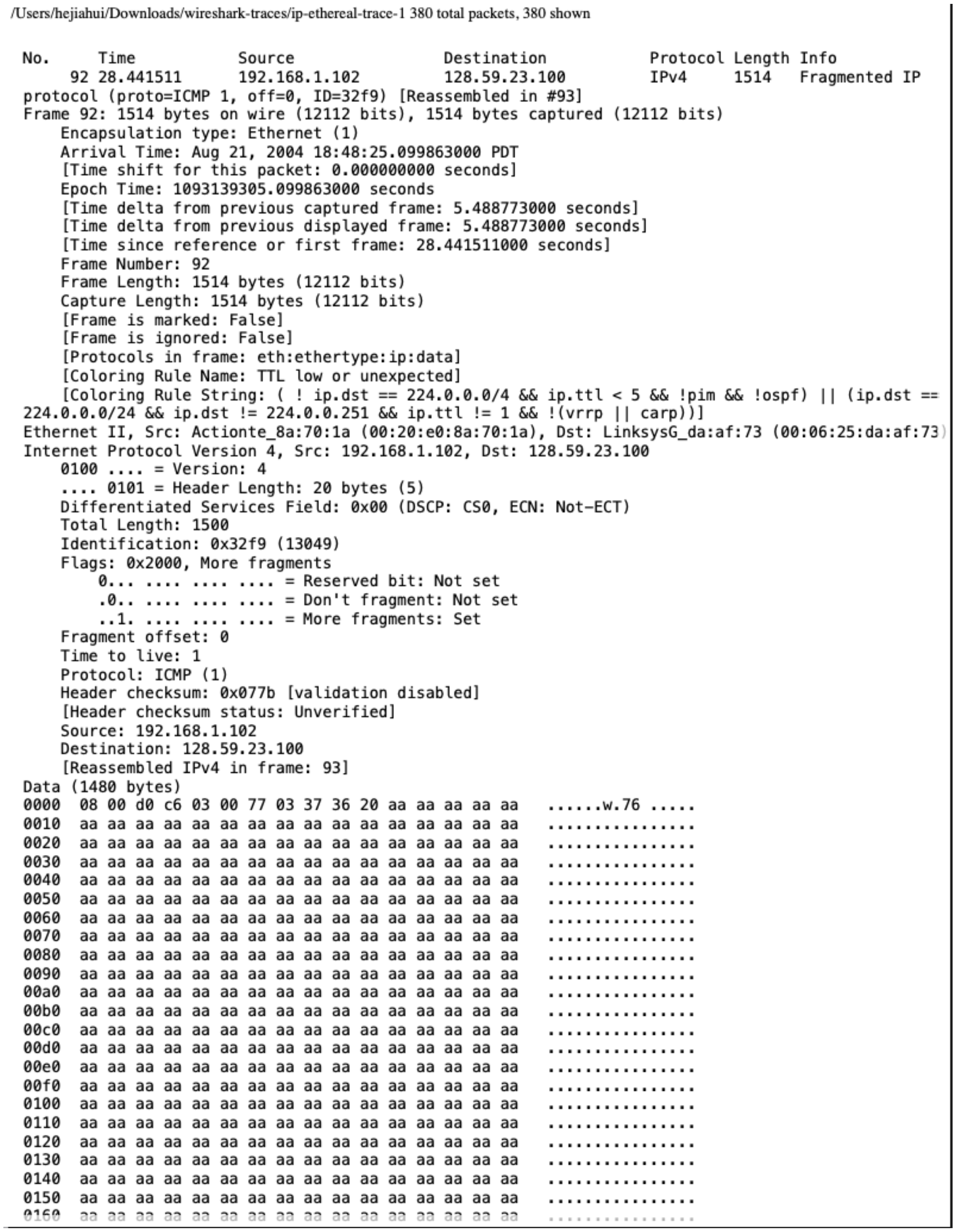
Sort the packet listing according to time again by clicking on the *Time* column.

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 *ip ethereal-trace-1*packet trace. If your computer has an Ethernet interface, a packet size of 2000 *should* cause fragmentation.[3]]

- Yes, this packet has been fragmented.

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?

[3] The packets in the *ip-ethereal-trace-1* trace file in http://gaia.cs.umass.edu/wireshark-labs/wireshark traces.zip are all less that 1500 bytes. This is because the computer on which the trace was gathered has an Ethernet card that limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP/IP header data and 1460 bytes of upper-layer protocol payload). This 1500 byte value is the standard maximum length allowed by Ethernet. If your trace indicates a datagram longer 1500 bytes, and your computer is using an Ethernet connection, then Wireshark is reporting the wrong IP datagram length; it will likely also show only one large IP datagram rather than multiple smaller datagrams.. This inconsistency in reported lengths is due to the interaction between the Ethernet driver and the Wireshark software. We recommend that if you have this inconsistency, that you perform this lab using the *ip-ethereal-trace-1* trace file.

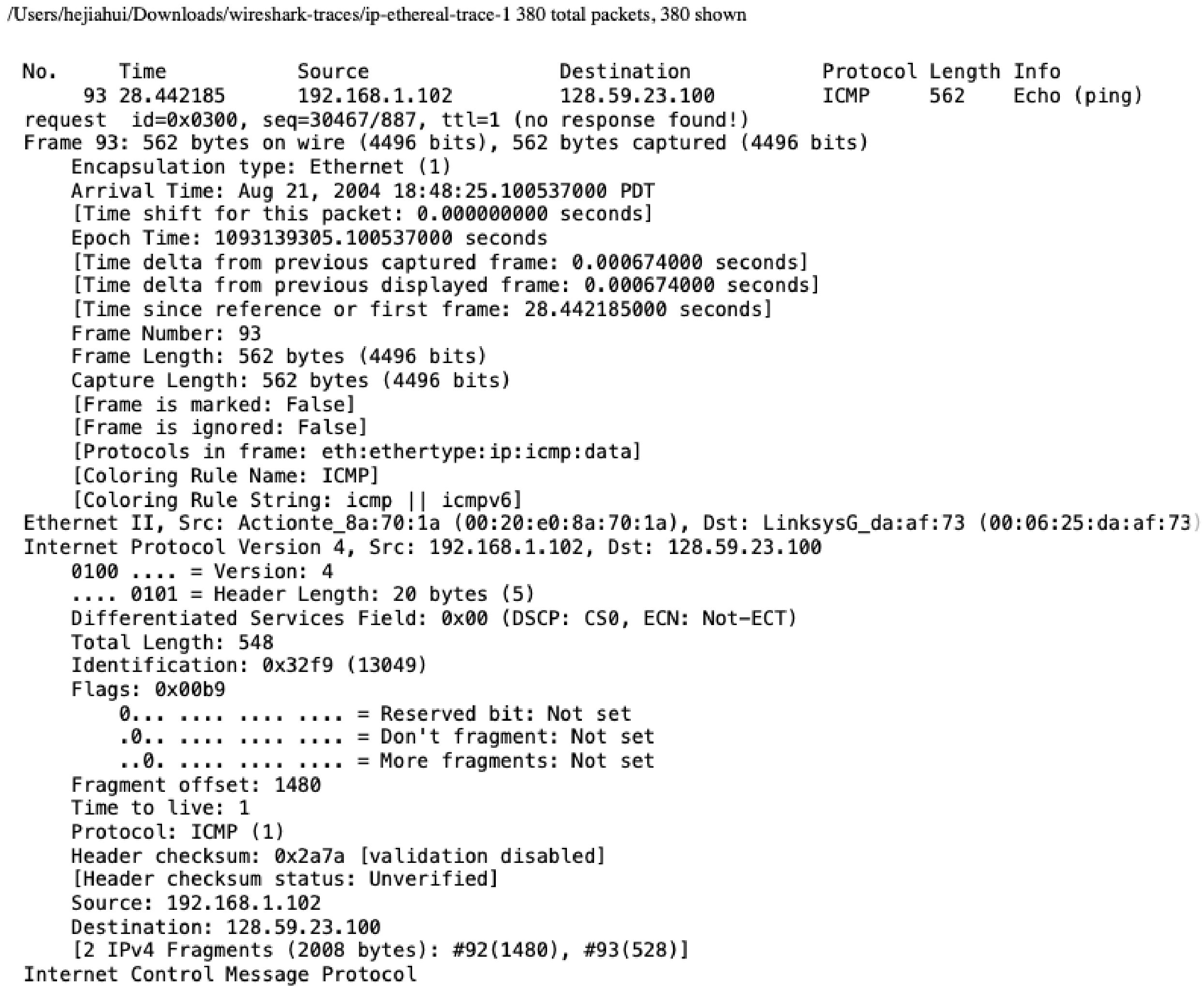


```
/Users/hejiahui/Downloads/wireshark-traces/ip-ethereal-trace-1 380 total packets, 380 shown

No.     Time           Source                Destination           Protocol Length Info
     92 28.441511      192.168.1.102         128.59.23.100         IPv4     1514   Fragmented IP
protocol (proto=ICMP 1, off=0, ID=32f9) [Reassembled in #93]
Frame 92: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Aug 21, 2004 18:48:25.099863000 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1093139305.099863000 seconds
    [Time delta from previous captured frame: 5.488773000 seconds]
    [Time delta from previous displayed frame: 5.488773000 seconds]
    [Time since reference or first frame: 28.441511000 seconds]
    Frame Number: 92
    Frame Length: 1514 bytes (12112 bits)
    Capture Length: 1514 bytes (12112 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:data]
    [Coloring Rule Name: TTL low or unexpected]
    [Coloring Rule String: ( ! ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim && !ospf) || (ip.dst ==
224.0.0.0/24 && ip.dst != 224.0.0.251 && ip.ttl != 1 && !(vrrp || carp))]
Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x32f9 (13049)
    Flags: 0x2000, More fragments
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..1. .... .... .... = More fragments: Set
    Fragment offset: 0
    Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x077b [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.102
    Destination: 128.59.23.100
    [Reassembled IPv4 in frame: 93]
Data (1480 bytes)
0000  08 00 d0 c6 03 00 77 03 37 36 20 aa aa aa aa aa   ......w.76 .....
0010  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0020  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0030  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0040  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0050  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0060  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0070  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0080  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0090  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
00a0  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
00b0  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
00c0  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
00d0  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
00e0  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
00f0  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0100  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0110  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0120  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0130  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0140  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0150  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
0160  aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa   ................
```

- The Flags bit for more fragments is set, indicating that the datagram has been fragmented.
- The fragment offset is 0 indicating this is the first fragment.
- This first datagram has a total length of 1500, including the header.

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 there more fragments? How can you tell?



/Users/hejiahui/Downloads/wireshark-traces/ip-ethereal-trace-1 380 total packets, 380 shown

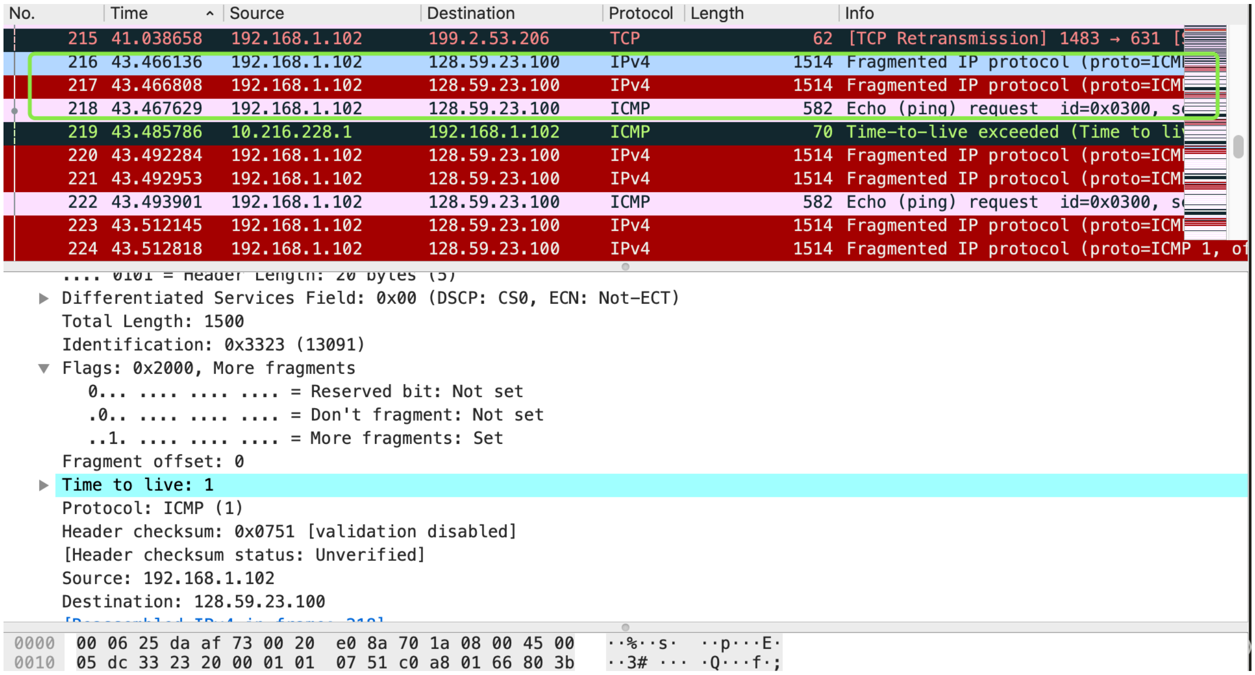
```
No.     Time           Source                Destination           Protocol Length Info
     93 28.442185      192.168.1.102         128.59.23.100         ICMP     562    Echo (ping)
request  id=0x0300, seq=30467/887, ttl=1 (no response found!)
Frame 93: 562 bytes on wire (4496 bits), 562 bytes captured (4496 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Aug 21, 2004 18:48:25.100537000 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1093139305.100537000 seconds
    [Time delta from previous captured frame: 0.000674000 seconds]
    [Time delta from previous displayed frame: 0.000674000 seconds]
    [Time since reference or first frame: 28.442185000 seconds]
    Frame Number: 93
    Frame Length: 562 bytes (4496 bits)
    Capture Length: 562 bytes (4496 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 548
    Identification: 0x32f9 (13049)
    Flags: 0x00b9
        0... .... .... .... = Reserved bit: Not set
        .0.. .... .... .... = Don't fragment: Not set
        ..0. .... .... .... = More fragments: Not set
    Fragment offset: 1480
    Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2a7a [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.102
    Destination: 128.59.23.100
    [2 IPv4 Fragments (2008 bytes): #92(1480), #93(528)]
Internet Control Message Protocol
```

- The fragment offset is 1480.
- It is the last fragment because the More fragment flag is not set.

13. Now find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* in *pingplotter* to be 3500.

- Total length,
- flags,
- fragment offset, and
- checksum.

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



```
No.   Time        Source          Destination     Protocol Length  Info
 215  41.038658   192.168.1.102   199.2.53.206    TCP         62   [TCP Retransmission] 1483 → 631 [
 216  43.466136   192.168.1.102   128.59.23.100   IPv4      1514   Fragmented IP protocol (proto=ICM
 217  43.466808   192.168.1.102   128.59.23.100   IPv4      1514   Fragmented IP protocol (proto=ICM
 218  43.467629   192.168.1.102   128.59.23.100   ICMP       582   Echo (ping) request  id=0x0300, se
 219  43.485786   10.216.228.1    192.168.1.102   ICMP        70   Time-to-live exceeded (Time to li
 220  43.492284   192.168.1.102   128.59.23.100   IPv4      1514   Fragmented IP protocol (proto=ICM
 221  43.492953   192.168.1.102   128.59.23.100   IPv4      1514   Fragmented IP protocol (proto=ICM
 222  43.493901   192.168.1.102   128.59.23.100   ICMP       582   Echo (ping) request  id=0x0300, se
 223  43.512145   192.168.1.102   128.59.23.100   IPv4      1514   Fragmented IP protocol (proto=ICM
 224  43.512818   192.168.1.102   128.59.23.100   IPv4      1514   Fragmented IP protocol (proto=ICMP 1, of
```

```
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x3323 (13091)
  ▼ Flags: 0x2000, More fragments
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..1. .... .... .... = More fragments: Set
    Fragment offset: 0
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x0751 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.102
    Destination: 128.59.23.100
    [Reassembled IPv4 in frame: 218]
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  05 dc 33 23 20 00 01 01  07 51 c0 a8 01 66 80 3b   ··3# ···  ·Q···f·;
```

- There are three packets created.

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

- Fragment offset, and checksum.
- The total length also changed between the first two fragments and the last fragments.
- The first two fragments have a total length of 1500.
- The last fragment has a total length of 568.
- The More fragment flag changed.
- The first two fragments is 1 and the last fragment is 0.