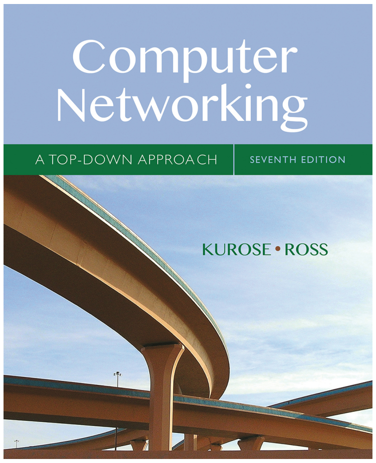
Wireshark Lab: IP v7.0

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.

在本实验中，我们将研究 IP 协议，重点关注 IP 数据报(IP datagram)。我们将通过分析在执行 traceroute 程序发送和接收的一系列 IP 数据报的过程来完成这个实验（traceroute 程序本身则是在 Wireshark ICMP 实验室中进行了更详细的探讨），我们将研究 IP datagram 中的各个字段(fields)，并详细研究 IP fragmentation 的方法。

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.

在开始本实验之前，希望您复习课本中的 1.4.3 节 和观看 RFC 2151 文件的 3.4 节中的内容[ftp://ftp.rfc-editor.org/in-notes/rfc2151.txt]，让自己更了解 traceroute 程序的操作。您还需要阅读文中的第 4.4 节，或许还需要看看 RFC 791 [ftp://ftp.rfc-editor.org/in-notes/rfc791.txt]，让自己对 IP 协议有基础的认识。

1. Capturing packets from an execution of traceroute (捕获执行 traceroute 的数据包)

[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.*

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 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.

为了生成本实验的一系列 IP 数据报，我们将使用 traceroute 程序向不同的目的地 X 发送不同大小的数据报。回想一下，traceroute 通过首先发送一个或多个带有生存时间(TTL: Time-to-Live)字段设置为 1 的数据报；然后发送一个或多个带有生存时间(TTL: Time-to-Live)字段设置为 2 的数据报到同一个目的地；然后发送一个或多个带有生存时间(TTL: Time-to-Live)字段设置为 3 的数据报到同一个目的地，以此类推，直到目的地真正收到此数据报为止。回想一下，路由器必须将每个接收到的数据报中的 TTL 减 1（实际上，RFC 791 文献中表示路由器必须将 TTL 减少至少一个）。如果 TTL 达到 0，路由器会向来源主机发送 ICMP 消息（类型 11 - 超出 TTL）。由于这种行为，TTL 为 1 的数据报（由执行 traceroute 的主机发送）将导致距发送方一次跳跃的路由器，将 ICMP TTL 超出的消息发送回发送方主机；以 TTL 为 2 发送的数据报将导致距离为两次跳跃的路由器，将 ICMP 消息发送回发送方主机；以 TTL 为 3 发送的数据报将导致距离为两次跳跃的路由器，将 ICMP 消息发送回发送方主机，等等。以这种方式，执行 traceroute 的主机可以通过查看包含 ICMP TTL 超出消息的数据报中的来源 IP 地址来获知其自身与目的地 X 之间的路由器的身份。

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

我们想要运行 traceroute 并让它发送各种长度的数据报。

- **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*.

- Windows 操作系统：Windows 提供的 tracert 程序（曾被使用于我们的 ICMP Wireshark 实验中）不允许更改 tracert 程序发送的 ICMP echo 请求（ping）消息的大小。因此，一个更好的 Windows traceroute 程序是 pingplotter，可在 http://www.pingplotter.com 上以免费版和共享软件版本获得。下载并安装 pingplotter，并通过对您喜欢的站点执行一些 traceroute 来测试它。通过选择菜单项 Edit-> Options-> Packet Options 然后填写 Packet Size 字段，可以在 pingplotter 中显式设置 ICMP echo 请求消息的大小。默认数据包大小为 56 个字节。一旦 pingplotter 发送了一系列具有递增的 TTL 值的数据包，它会在等待 Trace Interval 时间后再次以 TTL 为 1 重新启动发送进程。同时，我们可以在 pingplotter 中明确设置 Trace Interval 的值和间隔数。(备注：PinPlotter 5 需要使用到 Standar 版或是 Professional 版才能够自定义 packet 参数，有 14 天的试用版可以使用)

- **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:

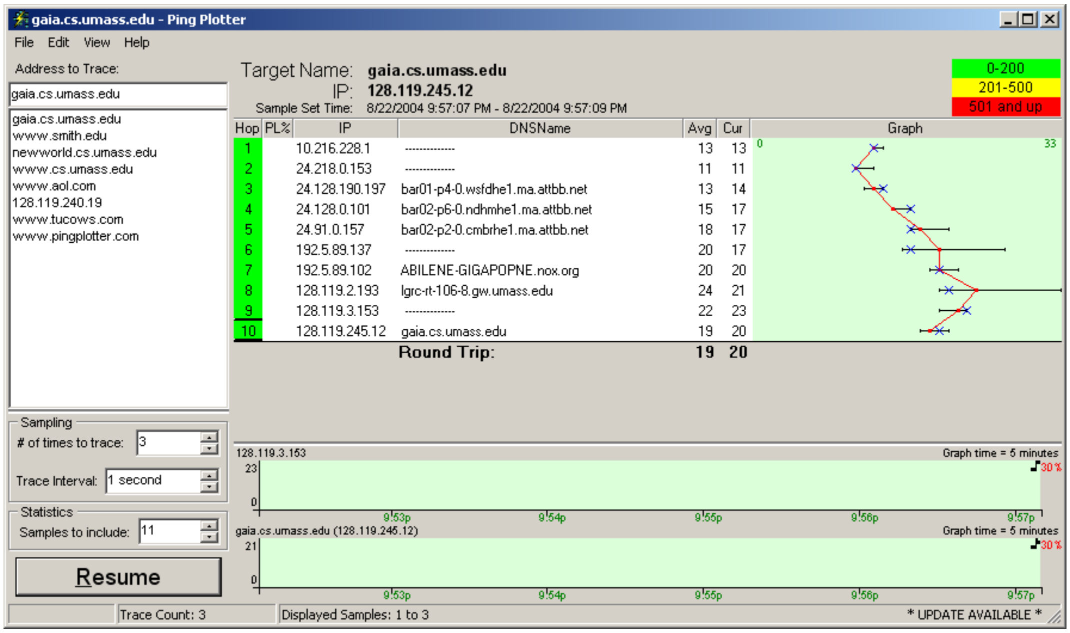
```
%traceroute gaia.cs.umass.edu 2000
```

Do the following: (依照下列步骤执行：)

- 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).
  启动 Wireshark 并开始数据包捕获（Capture-> Start），然后在 Wireshark 数据包捕获选项屏幕上按 OK（我们不需要在此处选择任何选项）。

- 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:

如果您使用的是 Windows 平台，请启动 pingplotter 并在“要跟踪的地址窗口”中输入目标目标的名称。在“要跟踪的次数”字段中输入 3，这样您就不会收集太多数据。选择菜单项编辑 - >高级选项 - >数据包选项，然后在数据包大小字段中输入值 56，然后按确定。然后按 Trace 按钮。你应该看到一个看起来像这样的 pingplotter 窗口：(备注：新版 PingPlotter5 中没有跟踪次数的设定，可以在 count 到达 3 的时候按下暂停键，停止收集数据包)



Address to Trace: gaia.cs.umass.edu

gaia.cs.umass.edu
www.smith.edu
newworld.cs.umass.edu
www.cs.umass.edu
www.aol.com
128.119.240.19
www.tucows.com
www.pingplotter.com

Sampling
# of times to trace: 3
Trace Interval: 1 second

Statistics
Samples to include: 11

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

🟩 0-200
🟨 201-500
🟥 501 and up

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	lgrc-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**

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.
接下来，通过选择编辑 ->高级选项 ->数据包选项并在数据包大小字段中输入值 2000，然后按确定，发送一组长度较长的数据报。然后按“继续”按钮。

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.
最后，通过选择 Edit-> Advanced Options-> Packet Options 并在 Packet Size 字段中输入值 3500，然后按 OK，发送一组长度较长的数据报。然后按“继续”按钮。

Stop Wireshark tracing. (停止 Wireshark 数据包撷取)

- 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.

如果您使用的是 Unix 或 Mac 平台，请输入三个 traceroute 命令，一个长度为 56 个字节，一个长度为 2000 个字节，另一个长度为 3500 个字节。

Stop Wireshark tracing. (停止 Wireshark 数据包撷取)

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.
如果您无法在实际的网络连接上运行 Wireshark，则可以下载在作者的某台 Windows 计算器上执行上述步骤时捕获的数据包跟踪文件。当您探索下面的问题时，即使您已经捕获了自己的跟踪数据并使用它，如同您自己的跟踪数据一般，您也可能会发现下载此跟踪数据对你的实验很有帮助。

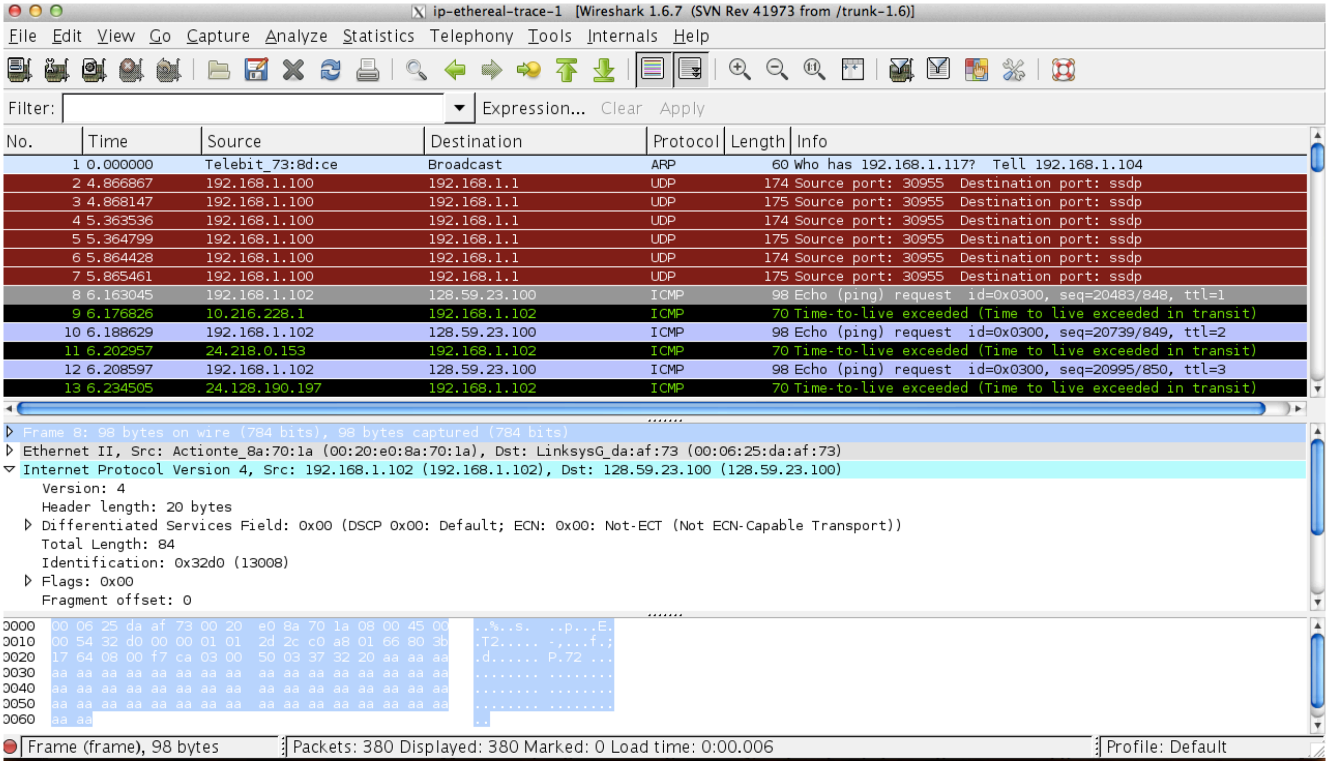
2. A look at the captured trace

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 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.
在您的跟踪数据包中，您应该能够看到计算器发送的一系列的 ICMP Echo 请求讯息（在 Windows 计算器的情况下）或 UDP 区段（在 Unix 的情况下）以及由中间路由器发送到计算器的 ICMP TTL 超出的讯息。在下面的问题中，我们假设您使用的是 Windows 机器；对于 Unix 机器的相应问题应该是清楚的。只要有可能，在回答下面的问题时，您应该提交用于回答问题的跟踪内的数据包的打印输出。当您提交作业时，请对输出进行注释，以便清楚地显示输出中您获得答案信息的位置（例如，对于我们的课程，我们要求学生用笔标记纸本答案，或者使用带注释的电子副本。若要打印数据包，请使用文件 - >打印，选择仅选择数据包，选择数据包

[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.

摘要行，然后选择回答本问题时所需的最小数据包的详细信息量。

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
```

What is the IP address of your computer?
选择计算器发送的第一个 ICMP Echo Request 消息，然后在 packet details window 中展开数据包的 Internet 协议部分。您的计算器的 IP 地址是多少？

2. Within the IP packet header, what is the value in the upper layer protocol field?
在 IP header 中，上层协议字段的值是多少？

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.
IP header 有多少 bytes？ IP datagram 的有效负载中有多少 bytes？ 说明如何确定 payload bytes 的数。

4. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.
此 IP 数据报是否已被分段(fragmented)？解释您如何确定数据报是否已被分段(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.
接下来，通过单击 Source 列标题，根据 IP 源地址对跟踪的数据包进行排序，一个小的向下箭头应出现在 Source 旁边，如果箭头指向上方请再次单击“Source column header”。选择计算器发送的第一个 ICMP Echo Request 消息，然后展开“details of selected packet header”窗口中的 Internet 协议部分。在“listing of captured packets”窗口中，您应该在第一个 ICMP 下面看到所有后续 ICMP 消息（可能还有计算器上运行的其他协议发送的其他散布数据包），使用向下箭头浏览计算器发送的 ICMP 消息。

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?
   在您的计算器发送的这一系列 ICMP 消息中，IP 数据报中的哪些字段“always”从有改变？

6. Which fields stay constant? Which of the fields *must* stay constant? Which fields must change? Why?
   哪些字段保持不变？ 哪个字段必须保持不变？ 哪些字段必须更改？ 为什么？

7. Describe the pattern you see in the values in the Identification field of the IP datagram
   描述您在 IP datagram 的 Identification field 中的值中所看到的

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.
下一步（数据包仍按来源地址排序）查找最近的（第一跳）路由器发送到您的计算器的一系列 ICMP TTL 超出的回复讯息。

8. What is the value in the Identification field and the TTL field?
   ID 字段和 TTL 字段的值是多少？
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?
   对于最近（第一跳）路由器发送到您的计算器的所有 ICMP TTL 超出的回复，这些值是否保持不变？为什么？

Fragmentation

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

单击“时间”列，再次按时间对数据包列表进行排序。

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]]

    在将 pingplotter 中的数据包大小更改为 2000 后，查找计算机发送的第一个 ICMP Echo Request 消息。该消息是否已碎片化为多个 IP 数据报？ [注意：如果你发现你的数据包没有被分割，你应该下载 zip 文件 http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip 并提取 ip-ethereal-trace-1packet 跟踪。 如果您的计算机具有以太网接口，则数据包大小为 2000 会导致碎片。]

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?

    打印出碎片 IP 数据报的第一个片段。 IP 头中的哪些信息表明数据报已碎片化？ IP 头中的哪些信息表明这是第一个片段还是后一个片段？ 这个 IP 数据报有多长？

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?

    打印出碎片 IP 数据报的第二个片段。 IP 标头中的哪些信息表明这不是第一个数据报片段？ 是否还有更多的片段？ 你是如何知道的？

[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.

13. What fields change in the IP header between the first and second fragment?
    在第一个和第二个片段中，IP 标头中哪些字段发生了变化？

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.
现在，在将 pingplotter 中的数据包大小更改为 3500 后，找到计算机发送的第一个 ICMP Echo Request 消息。

14. How many fragments were created from the original datagram?
    从原始数据报创建了多少个片段？
15. What fields change in the IP header among the fragments?
    片段中 IP 标头中的哪些字段发生了变化？