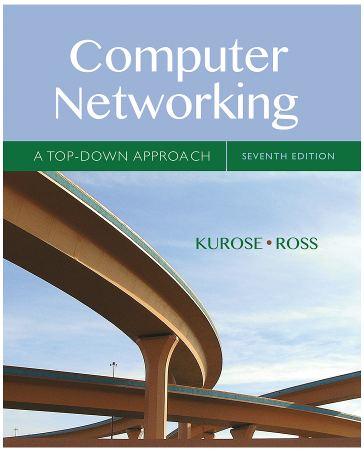
Wireshark Lab: ICMP v7.0

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 explore several aspects of the ICMP protocol:
在本实验中，我们将探讨 ICMP 协议的几个方面：

- ICMP messages generating by the Ping program;
  Ping 程序生成的 ICMP 消息;
- ICMP messages generated by the Traceroute program;
  Traceroute 程序生成的 ICMP 消息;
- the format and contents of an ICMP message.
  ICMP 消息的格式和内容。

Before attacking this lab, you're encouraged to review the ICMP material in section 5.6 of the text[1]. We present this lab in the context of the Microsoft Windows operating system. However, it is straightforward to translate the lab to a Unix or Linux environment.
在开始本实验之前，我们建议您查看课本的 5.6 节中的 ICMP 章节。此实验是在 Windows 下完成的，如果您使用其他系统也不用担心，因为大体都相同。

1. ICMP and Ping ICMP 协议和 Ping 程序

Let's begin our ICMP adventure by capturing the packets generated by the Ping program. You may recall that the Ping program is simple tool that allows anyone (for example, a network administrator) to verify if a host is live or not. The Ping program in the source host sends a packet to the target IP address; if the target is live, the Ping program in the target host responds by sending a packet back to the source host. As you might have guessed (given that this lab is about ICMP), both of these Ping packets are ICMP packets.
让我们通过捕获 Ping 程序生成的数据包开始我们的 ICMP 实验。 您可能还记得 Ping 程序是一个简单的工具，允许任何人（例如：网络管理员）验证主机是否存

[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.
中文版：计算机网络自顶向下方法第 7 版（机械工业出版社）

在。源主机中的 Ping 程序将数据包发送到目标 IP 地址; 如果目标是在线的，则目标主机中的 Ping 程序将会发送响应 Ping 数据包证明他在线，这两个 Ping 数据包都是 ICMP 数据包。因此您可能猜出我们这个实验都是关于 ICMP 的实验了。

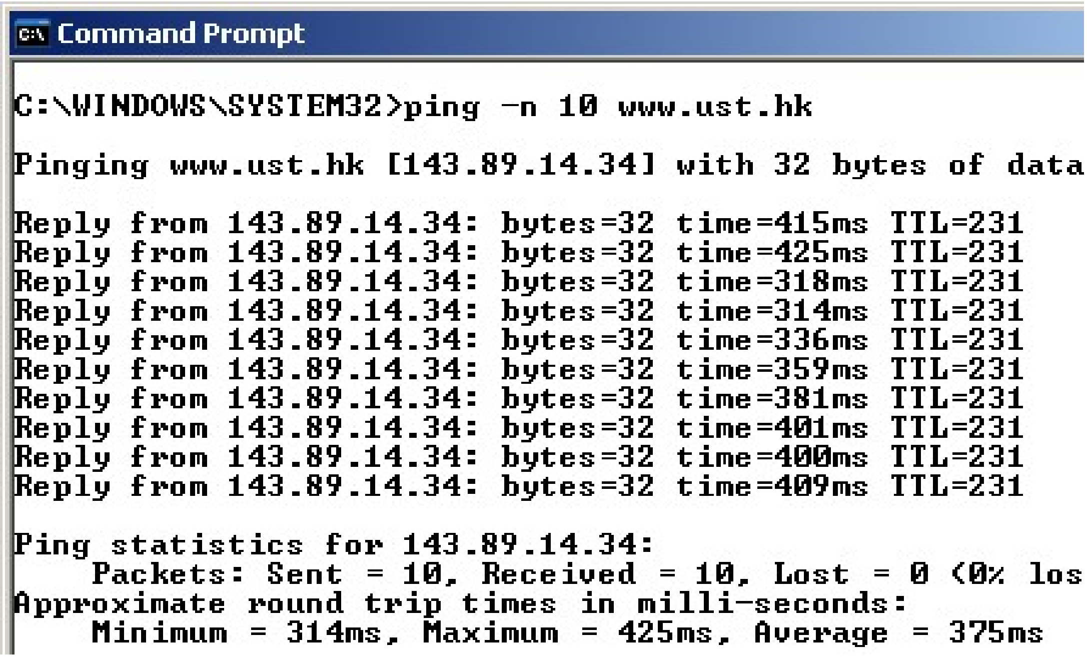
Do the following[2]: 请按照以下动作做:

- Let’s begin this adventure by opening the Windows Command Prompt application (which can be found in your Accessories folder).
  打开 windows 的命令提示符。
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
  启动 Wireshark 数据包嗅探器，并开始 Wireshark 数据包捕获。
- The *ping* command is in c:\windows\system32, so type either “*ping –n 10 hostname*” or “*c:\windows\system32\ping –n 10 hostname*” in the MS-DOS command line (without quotation marks), where hostname is a host on another continent. If you’re outside of Asia, you may want to enter www.ust.hk for the Web server at Hong Kong University of Science and Technology. The argument *“-n 10*” indicates that 10 ping messages should be sent. Then run the Ping program by typing return.
  Ping 程序在 c:\windows\system32 目录中，所以您在命令提示符中输入“*ping –n 10 hostname*” 或 “*c:\windows\system32\ping –n 10 hostname*”都是正确的（注意命令是引号里的内容）。其中 hostname 是另一个大陆的主机名。如您如果在非亚洲地区，建议您访问香港科技大学 www.ust.hk ， 参数-n 10 代表发送 10 个 Ping 消息，然后按下回车执行命令。
- When the Ping program terminates, stop the packet capture in Wireshark.
  当 Ping 程序终止时，停止在 Wireshark 中捕获数据包。

At the end of the experiment, your Command Prompt Window should look something like Figure 1. In this example, the source ping program is in Massachusetts and the destination Ping program is in Hong Kong. From this window we see that the source ping program sent 10 query packets and received 10 responses. Note also that for each response, the source calculates the round-trip time (RTT), which for the 10 packets is on average 375 msec.
在实验结束时，您的命令提示符窗口应如图 1 所示。在此示例中，源 ping 程序位于马萨诸塞州，目标 Ping 程序位于香港。从这个窗口我们看到源 ping 程序发送了

[2] If you are unable to run Wireshark live on a computer, you can download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the file *ICMP-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 *ICMP-ethereal-trace-1* trace file. You can then use this trace file to answer the questions below.
同样如果您无法抓包，建议您下载作者的抓包结果 http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip 解压并且使用 Wireshark 打开 *ICMP-ethereal-trace-1* 进行分析。

10 个查询包并收到了 10 个响应。 另请注意，对于每个响应，源计算往返时间（RTT），对于 10 个数据包平均为 375 毫秒。

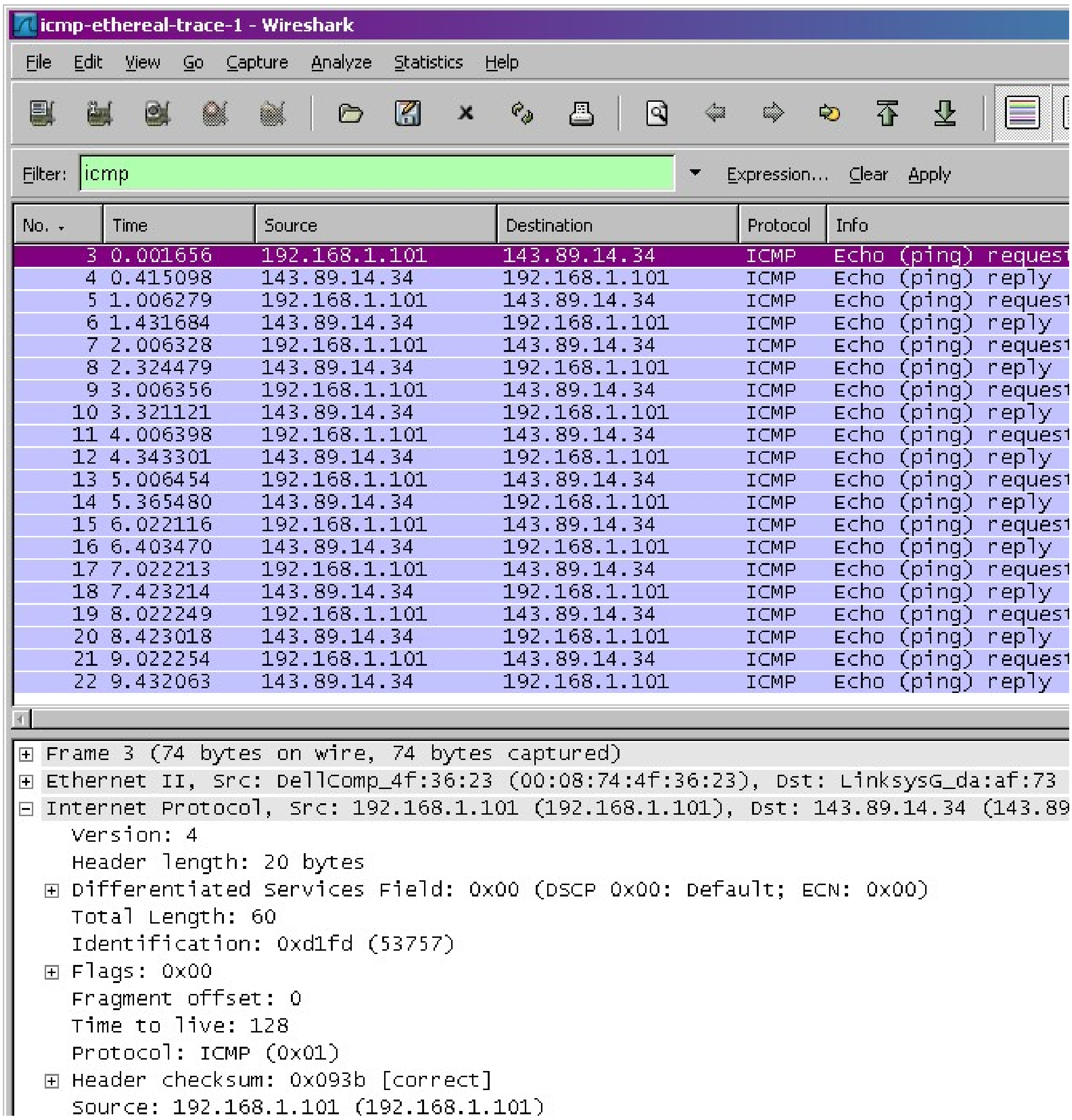


```
C:\WINDOWS\SYSTEM32>ping -n 10 www.ust.hk

Pinging www.ust.hk [143.89.14.34] with 32 bytes of data

Reply from 143.89.14.34: bytes=32 time=415ms TTL=231
Reply from 143.89.14.34: bytes=32 time=425ms TTL=231
Reply from 143.89.14.34: bytes=32 time=318ms TTL=231
Reply from 143.89.14.34: bytes=32 time=314ms TTL=231
Reply from 143.89.14.34: bytes=32 time=336ms TTL=231
Reply from 143.89.14.34: bytes=32 time=359ms TTL=231
Reply from 143.89.14.34: bytes=32 time=381ms TTL=231
Reply from 143.89.14.34: bytes=32 time=401ms TTL=231
Reply from 143.89.14.34: bytes=32 time=400ms TTL=231
Reply from 143.89.14.34: bytes=32 time=409ms TTL=231

Ping statistics for 143.89.14.34:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% los
Approximate round trip times in milli-seconds:
    Minimum = 314ms, Maximum = 425ms, Average = 375ms
```

Figure 1 Command Prompt window after entering Ping command.
图 1 输入 Ping 命令后的命令提示符窗口。

Figure 2 provides a screenshot of the Wireshark output, after “icmp” has been entered into the filter display window. Note that the packet listing shows 20 packets: the 10 Ping queries sent by the source and the 10 Ping responses received by the source. Also note that the source’s IP address is a private address (behind a NAT) of the form 192.168/12; the destination’s IP address is that of the Web server at HKUST. Now let’s zoom in on the first packet (sent by the client); in the figure below, the packet contents area provides information about this packet. We see that the IP datagram within this packet has protocol number 01, which is the protocol number for ICMP. This means that the payload of the IP datagram is an ICMP packet.

如图 2，在将“icmp”输入过滤器显示窗口后 Wireshark 输出的屏幕截图。 请注意，数据包列表显示 20 个数据包：源发送的 10 个 Ping 查询和源接收的 10 个 Ping 响应。 另请注意，源的 IP 地址是 192.168 / 12 格式的私有地址（通过 NAT 协议连接互联网）；目的地的 IP 地址是香港科技大学网络服务器的 IP 地址。 现在让我们点击显示第一个由客户端发送的数据包; 在图中的下方，数据包内容区域提供有关此数据包的信息。 我们看到该数据包中的 IP 数据报具有协议号 01，这是 ICMP 的协议号。 这意味着 IP 数据报的有效载荷是 ICMP 数据包。


icmp-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Expression... Clear Apply

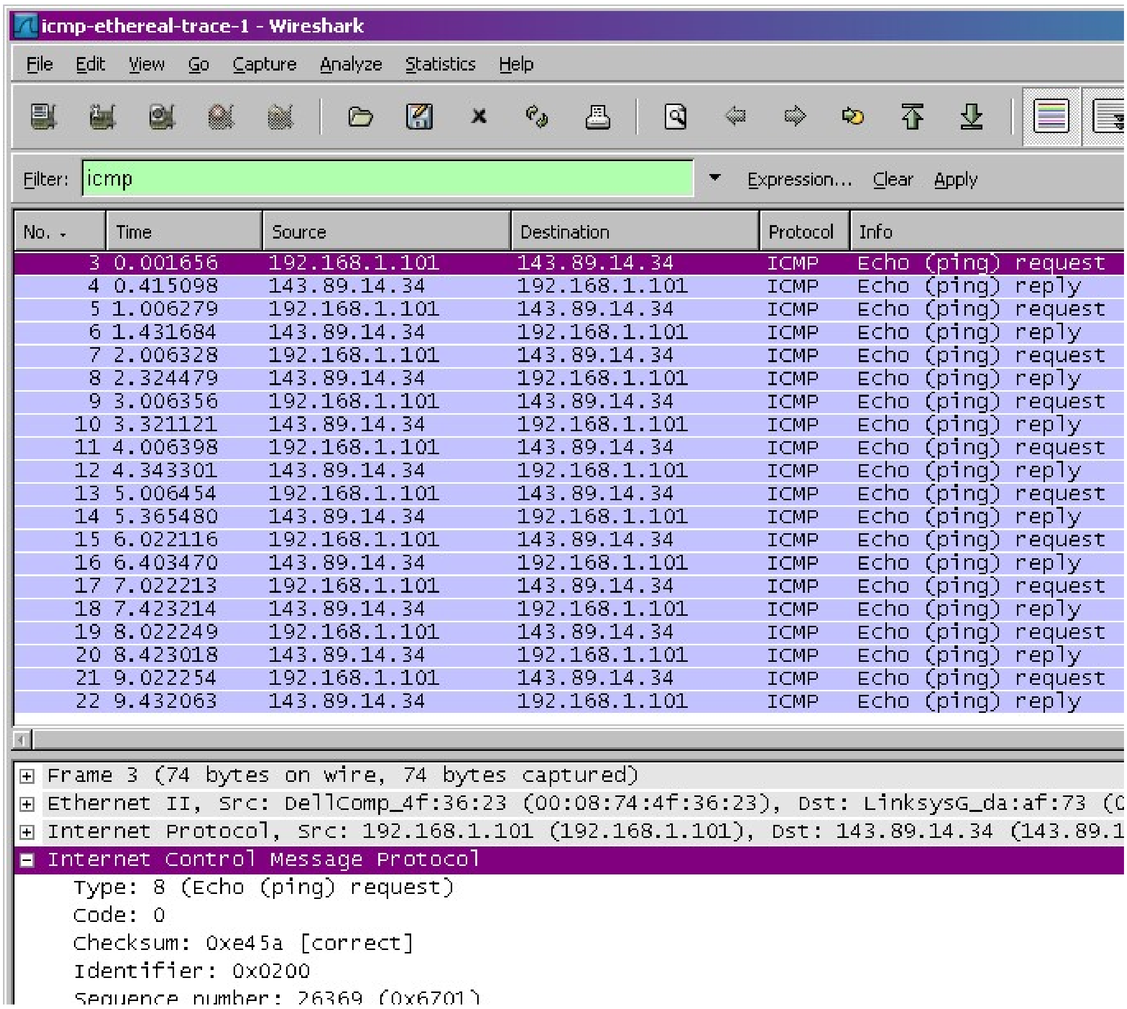
Filter: icmp

No.	Time	Source	Destination	Protocol	Info
3	0.001656	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
4	0.415098	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
5	1.006279	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
6	1.431684	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
7	2.006328	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
8	2.324479	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
9	3.006356	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
10	3.321121	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
11	4.006398	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
12	4.343301	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
13	5.006454	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
14	5.365480	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
15	6.022116	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
16	6.403470	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
17	7.022213	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
18	7.423214	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
19	8.022249	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
20	8.423018	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply
21	9.022254	192.168.1.101	143.89.14.34	ICMP	Echo (ping) request
22	9.432063	143.89.14.34	192.168.1.101	ICMP	Echo (ping) reply

```
⊞ Frame 3 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: DellComp_4f:36:23 (00:08:74:4f:36:23), Dst: LinksysG_da:af:73
⊟ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 143.89.14.34 (143.89
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0xd1fd (53757)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x093b [correct]
    Source: 192.168.1.101 (192.168.1.101)
```

Figure 2 Wireshark output for Ping program with Internet Protocol expanded.
图 2 在 Wireshark 中过滤 ICMP 协议并且选中某个 ICMP 消息

Figure 3 focuses on the same ICMP but has expanded the ICMP protocol information in the packet contents window. Observe that this ICMP packet is of Type 8 and Code 0 - a so-called ICMP “echo request” packet. (See Figure 5.19 of text.) Also note that this ICMP packet contains a checksum, an identifier, and a sequence number.
图 3 同样使用相同的 ICMP 过滤器，展开可以看到该 ICMP 包的详情信息。 观察到该 ICMP 数据包是类型 8 和代码 0 ——所谓的 ICMP“回应请求”数据包。 （请参见书本的图 5.19。）此外，此 ICMP 数据包包含校验和，标识符和序列号。



Filter: icmp

```
No. -    Time        Source          Destination     Protocol  Info
      3 0.001656    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
      4 0.415098    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
      5 1.006279    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
      6 1.431684    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
      7 2.006328    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
      8 2.324479    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
      9 3.006356    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     10 3.321121    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
     11 4.006398    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     12 4.343301    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
     13 5.006454    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     14 5.365480    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
     15 6.022116    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     16 6.403470    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
     17 7.022213    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     18 7.423214    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
     19 8.022249    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     20 8.423018    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
     21 9.022254    192.168.1.101   143.89.14.34    ICMP      Echo (ping) request
     22 9.432063    143.89.14.34    192.168.1.101   ICMP      Echo (ping) reply
```

```
⊞ Frame 3 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: DellComp_4f:36:23 (00:08:74:4f:36:23), Dst: LinksysG_da:af:73 (0
⊞ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 143.89.14.34 (143.89.1
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xe45a [correct]
    Identifier: 0x0200
    Sequence number: 26369 (0x6701)
```

Figure 3 Wireshark capture of ping packet with ICMP packet expanded.
图 3 选择展开 ICMP 包的详情信息

What to Hand In: 回答问题

You should hand in a screen shot of the Command Prompt window similar to Figure 1 above. 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. Annotate the printout[3] to explain your answer. 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.

[3] What do we mean by “annotate”? If you hand in a paper copy, please highlight where in the printout you’ve found the answer and add some text (preferably with a colored pen) noting what you found in what you ‘ve highlight. If you hand in an electronic copy, it would be great if you could also highlight and annotate.
请善用标记来展示您的答案

请尽量清晰的展示您的答案，必要时您可以在图中用标记辅以说明。您的答案应该简单可读。

You should answer the following questions: 请回答以下问题：

1. What is the IP address of your host? What is the IP address of the destination host? 您的主机的 IP 地址是多少？ 目标主机的 IP 地址是多少？
2. Why is it that an ICMP packet does not have source and destination port numbers? 为什么 ICMP 数据包没有源端口号和目的端口号？
3. Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? 查看任意的请求 ICMP 数据包， ICMP 类型和代码是什么？ 该 ICMP 数据包还有哪些其他字段？ 校验和，序列号和标识符字段有多少字节？
4. Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields? 查看任意的响应 ICMP 数据包， ICMP 类型和代码是什么？ 该 ICMP 数据包还有哪些其他字段？ 校验和，序列号和标识符字段有多少字节？

2. ICMP and Traceroute ICMP 协议和 Traceroute 命令

Let’s now continue our ICMP adventure by capturing the packets generated by the Traceroute program. You may recall that the Traceroute program can be used to figure out the path a packet takes from source to destination. Traceroute is discussed in Section 1.4 and in Section 5.6 of the text.
现在让我们通过捕获 Traceroute 程序生成的数据包继续我们的 ICMP 实验。 您可能还记得 Traceroute 程序可用于确定数据包从源到目的地的路径。 Traceroute 命令可以在课本中的 1.4 节和 5.6 节中找到。

Traceroute is implemented in different ways in Unix/Linux/MacOS and in Windows. In Unix/Linux, the source sends a series of UDP packets to the target destination using an unlikely destination port number; in Windows, the source sends a series of ICMP packets to the target destination. For both operating systems, the program sends the first packet with TTL=1, the second packet with TTL=2, and so on. Recall that a router will decrement a packet’s TTL value as the packet passes through the router. When a packet arrives at a router with TTL=1, the router sends an ICMP error packet back to the source. In the following, we’ll use the native Windows *tracert* program. A shareware version of a much nice Windows Traceroute program is *pingplotter* (www.pingplotter.com). We’ll use *pingplotter* in our Wireshark IP lab since it provides additional functionality that we’ll need there.

每个系统有不同路由跟踪实现办法，在 Unix / Linux 中，路由跟踪 traceroute 使用发送不可到达（无使用的）端口的 UDP 包来实现，在 Windows 中，路由跟踪 tracert 仅使用 ICMP 数据包来实现，但是对于他们来说，都是发送 TTL 增加的数据包，例如 TTL=1,TTL=2，回想下，每经过一个路由器，TTL 就会减一，当 TTL=1 的包达到路由器，该路由器会将该包丢弃，并且发送 ICMP 错误给请求的机器，在本次实验，我们使用 window 自带的 tracert。有一个跨平台 Windows Traceroute 程序的是 pingplotter（www.pingplotter.com）（收费但可以试用）。我们将在 Wireshark IP 实验室中使用 pingplotter，因为它提供了我们在那里需要的其他功能。

Do the following[4]: 请执行以下步骤

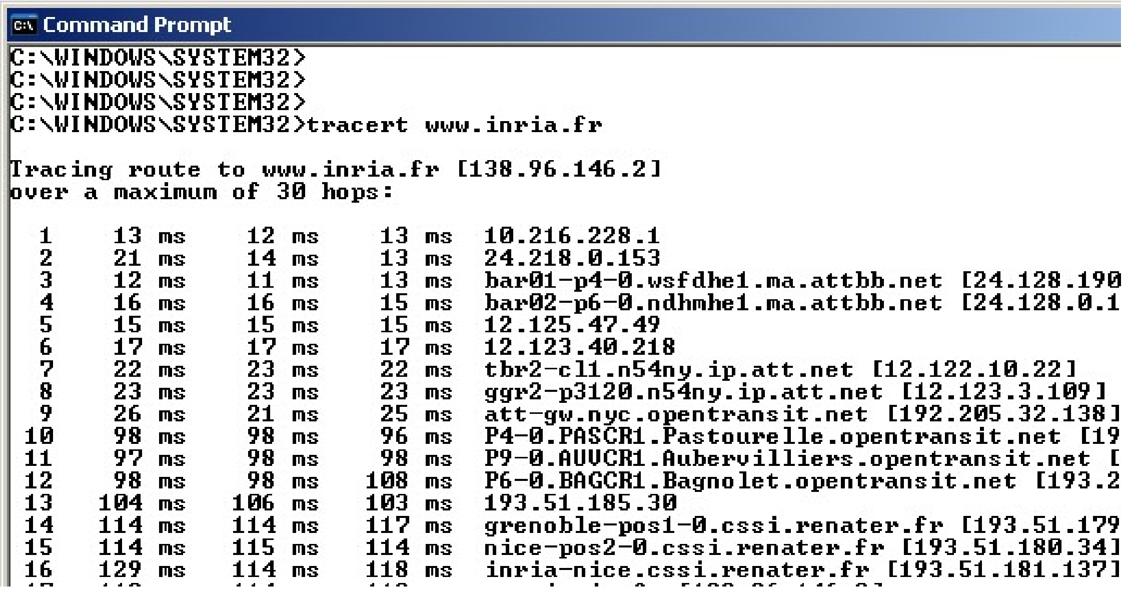
- Let's begin by opening the Windows Command Prompt application (which can be found in your Accessories folder).
  打开 windows 的命令提示符。
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
  启动 Wireshark 数据包嗅探器，并开始 Wireshark 数据包捕获。
- The *tracert* command is in c:\windows\system32, so type either "*tracert hostname*" or "*c:\windows\system32\tracert hostname*" in the MS-DOS command line (without quotation marks), where hostname is a host on another continent. (Note that on a Windows machine, the command is "*tracert*" and not "*traceroute*".) If you're outside of Europe, you may want to enter www.inria.fr for the Web server at INRIA, a computer science research institute in France. Then run the Traceroute program by typing return.
  tracert 程序在 c:\windows\system32 目录中，所以您在命令提示符中输入 "*tracert hostname*" 或 "*c:\windows\system32\ tracert hostname*"都是正确的（注意命令是引号里的内容）。其中 hostname 是另一个大陆的主机名。如您如果在非欧洲地区，建议您访问法国 INRIA（计算机科学研究所）www.inria.fr 然后按下回车执行命令。
- When the Traceroute program terminates, stop packet capture in Wireshark.
  当 Traceroute 程序终止时，停止在 Wireshark 中捕获数据包。

At the end of the experiment, your Command Prompt Window should look something like Figure 4. In this figure, the client Traceroute program is in Massachusetts and the target destination is in France. From this figure we see that for each TTL value, the

[4] If you are unable to run Wireshark live on a computer, you can download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the file *ICMP-ethereal-trace-2*. 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 *ICMP-ethereal-trace-2* trace file. You can then use this trace file to answer the questions below.
同样如果您无法抓包，建议您下载作者的抓包结果 http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip 解压并且使用 Wireshark 打开 *ICMP-ethereal-trace-2* 进行分析。

source program sends three probe packets. Traceroute displays the RTTs for each of the probe packets, as well as the IP address (and possibly the name) of the router that returned the ICMP TTL-exceeded message.

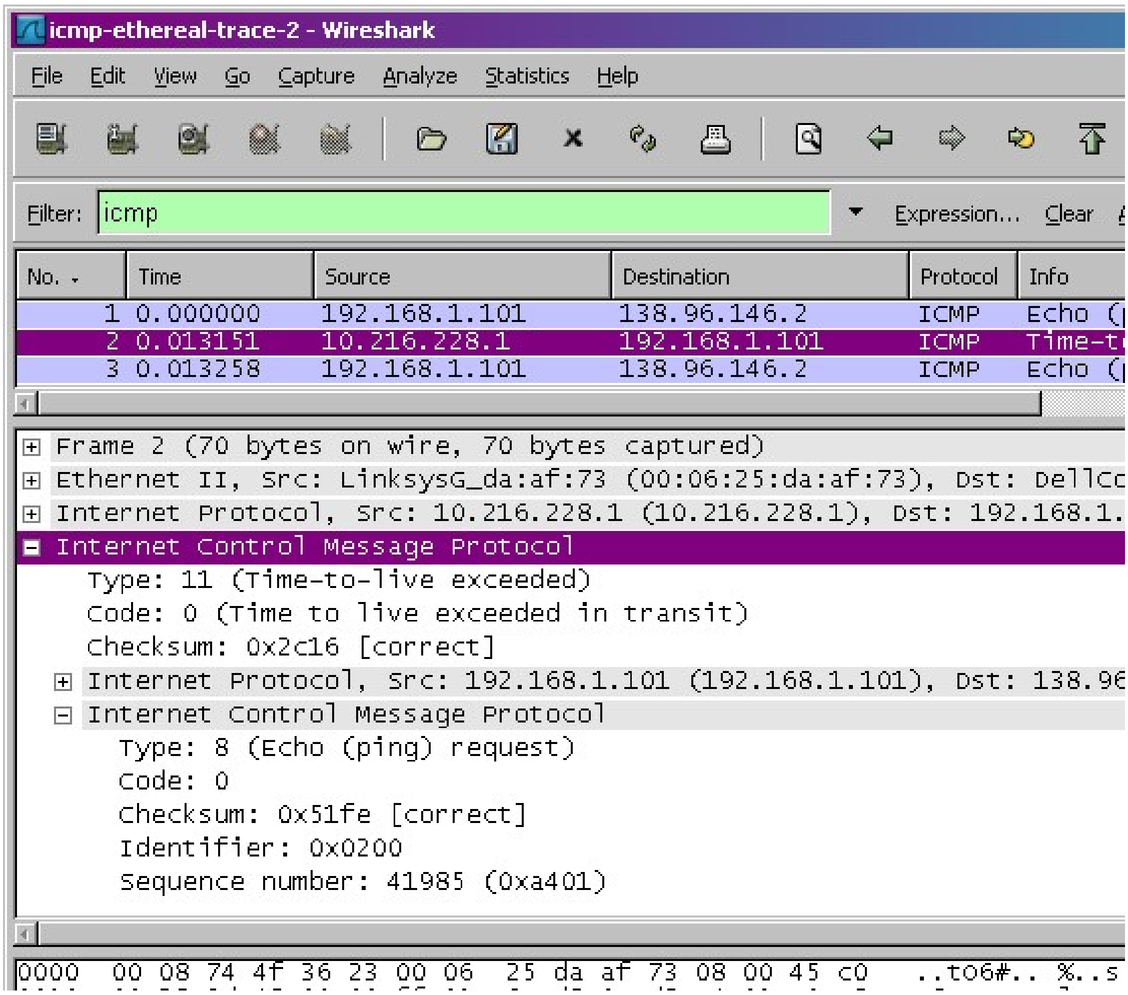
在实验结束时，您的命令提示符窗口应如图 4 所示。在此图中，客户端 Traceroute 程序主机位于马萨诸塞州，目标主机位于法国。 从该图中我们看到，对于每个 TTL 值，源程序发送三个探测包。 Traceroute 显示每个探测包的 RTT，以及返回 ICMP TTL 超出消息的路由器的 IP 地址（可能还有名称）。



```
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>tracert www.inria.fr

Tracing route to www.inria.fr [138.96.146.2]
over a maximum of 30 hops:

  1    13 ms    12 ms    13 ms  10.216.228.1
  2    21 ms    14 ms    13 ms  24.218.0.153
  3    12 ms    11 ms    13 ms  bar01-p4-0.wsfdhe1.ma.attbb.net [24.128.190
  4    16 ms    16 ms    15 ms  bar02-p6-0.ndhmhe1.ma.attbb.net [24.128.0.1
  5    15 ms    15 ms    15 ms  12.125.47.49
  6    17 ms    17 ms    17 ms  12.123.40.218
  7    22 ms    23 ms    22 ms  tbr2-cl1.n54ny.ip.att.net [12.122.10.22]
  8    23 ms    23 ms    23 ms  ggr2-p3120.n54ny.ip.att.net [12.123.3.109]
  9    26 ms    21 ms    25 ms  att-gw.nyc.opentransit.net [192.205.32.138]
 10    98 ms    98 ms    96 ms  P4-0.PASCR1.Pastourelle.opentransit.net [19
 11    97 ms    98 ms    98 ms  P9-0.AUVCR1.Aubervilliers.opentransit.net [
 12    98 ms    98 ms   108 ms  P6-0.BAGCR1.Bagnolet.opentransit.net [193.2
 13   104 ms   106 ms   103 ms  193.51.185.30
 14   114 ms   114 ms   117 ms  grenoble-pos1-0.cssi.renater.fr [193.51.179
 15   114 ms   115 ms   114 ms  nice-pos2-0.cssi.renater.fr [193.51.180.34]
 16   129 ms   114 ms   118 ms  inria-nice.cssi.renater.fr [193.51.181.137]
```

Figure 4 Command Prompt window displays the results of the Traceroute program.
图 4“命令提示符”窗口显示 Traceroute 程序的结果。

Figure 5 displays the Wireshark window for an ICMP packet returned by a router. Note that this ICMP error packet contains many more fields than the Ping ICMP messages.
图 5 显示了路由器返回的 ICMP 数据包的 Wireshark 窗口。 请注意，此 ICMP 错误数据包包含的字段比 Ping ICMP 消息多得多。


icmp-ethereal-trace-2 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: icmp
Expression... Clear

```
No. -   Time       Source          Destination     Protocol  Info
    1   0.000000   192.168.1.101   138.96.146.2    ICMP      Echo (
    2   0.013151   10.216.228.1    192.168.1.101   ICMP      Time-t
    3   0.013258   192.168.1.101   138.96.146.2    ICMP      Echo (

⊞ Frame 2 (70 bytes on wire, 70 bytes captured)
⊞ Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: DellCc
⊞ Internet Protocol, Src: 10.216.228.1 (10.216.228.1), Dst: 192.168.1.
⊟ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0x2c16 [correct]
  ⊞ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 138.96
  ⊟ Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0
      Checksum: 0x51fe [correct]
      Identifier: 0x0200
      Sequence number: 41985 (0xa401)

0000  00 08 74 4f 36 23 00 06  25 da af 73 08 00 45 c0   ..tO6#.. %..s
```

Figure 5 Wireshark window of ICMP fields expanded for one ICMP error packet.
图 5 为一个 ICMP 错误数据包扩展的 ICMP 字段的 Wireshark 窗口。

What to Hand In: 回答问题

For this part of the lab, you should hand in a screen shot of the Command Prompt window. 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. Annotate the printout to explain your answer. 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.
请尽量清晰的展示您的答案，必要时您可以在图中用标记辅以说明。您的答案应该简单可读。

Answer the following questions: 请回答以下问题:

5. What is the IP address of your host? What is the IP address of the target destination host?
   您的主机的 IP 地址是多少? 目标目标主机的 IP 地址是多少?
6. If ICMP sent UDP packets instead (as in Unix/Linux), would the IP protocol number still be 01 for the probe packets? If not, what would it be?
   如果 ICMP 发送了 UDP 数据包（如在 Unix / Linux 中），那么探测数据包的 IP 协议号仍然是 01 吗? 如果没有，它会是什么?
7. Examine the ICMP echo packet in your screenshot. Is this different from the ICMP ping query packets in the first half of this lab? If yes, how so?
   检查屏幕截图中的 ICMP 响应数据包。 这与本实验的前半部分中的 ICMP ping 查询数据包不同吗? 如果不同，请解释为什么?
8. Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?
   检查屏幕截图中的 ICMP 错误数据包。 它具有比 ICMP 响应数据包更多的字段。 这个数据包含哪些内容?
9. Examine the last three ICMP packets received by the source host. How are these packets different from the ICMP error packets? Why are they different?
   检查源主机收到的最后三个 ICMP 数据包。 这些数据包与 ICMP 错误数据包有何不同? 他们为什么不同?
10. Within the tracert measurements, is there a link whose delay is significantly longer than others? Refer to the screenshot in Figure 4, is there a link whose delay is significantly longer than others? On the basis of the router names, can you guess the location of the two routers on the end of this link?
    在 tracert 跟踪测量中，是否有一个连接的延迟比其他连接长得多? 请参阅图 4 中的屏幕截图，是否有连接的延迟明显长于其他连接? 根据路由器名称，您能猜出这个连接末端的两个路由器的位置吗?

3. Extra Credit 额外问题

For one of the programming assignments you created a UDP client ping program. This ping program, unlike the standard ping program, sends UDP probe packets rather than ICMP probe packets. Use the client program to send a UDP packet with an unusual destination port number to some live host. At the same time, use Wireshark to capture any response from the target host. Provide a Wireshark screenshot for the response as well as an analysis of the response.
对于一个编程任务，您可能创建了一个 UDP 客户端 ping 程序。 与标准 ping 程序不同，此 ping 程序发送 UDP 探测包而不是 ICMP 探测包。 使用客户端程序将具有异常目标端口号的 UDP 数据包发送到某个活动主机。 同时，使用 Wireshark 捕获目标主机的任何响应。 提供响应的 Wireshark 屏幕截图以及响应分析。