Wireshark Lab 1, part b: ICMP

In this portion of the lab, we will explore several aspects of the ICMP protocol:

- ICMP messages generating by the Ping program;
- ICMP messages generated by the Traceroute (*tracert*) program;
- the format and contents of an ICMP message.

Before attacking this lab, you are encouraged to review the ICMP material in the textbook (Section 4.4.3 in the 4th edition of the textbook.) 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.

1. ICMP and Ping

You will begin 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 on-line 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.

To capture the packet stream generated by a *ping* do the following:

- Begin by opening the Windows Command Prompt window: Select "Run…" from the Windows Start menu and type "cmd" (without the quotes) in the text box and hit "OK"

- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.

- At the Windows command prompt, type:

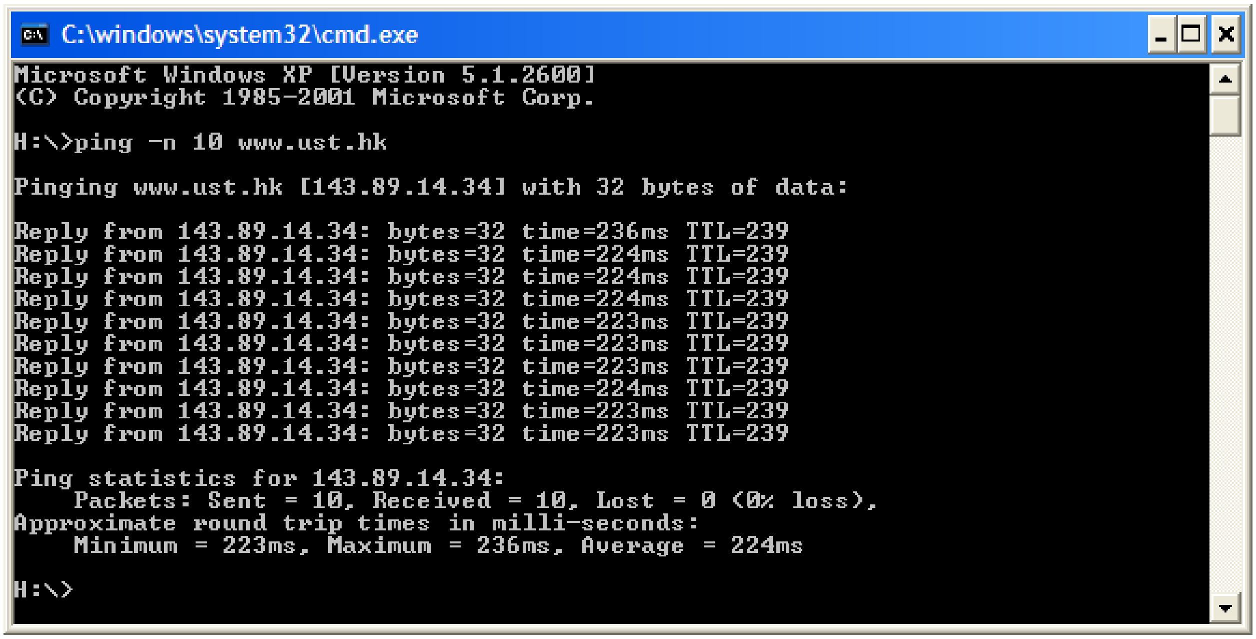
  ```
  ping –n 10   www.ust.hk
  ```

  to ping the web server at the Hong Kong University of Science and Technology. The argument "*-n 10*" indicates that 10 ping messages should be sent. If your path environment is not set up to find the ping command, you may need to use the complete path name:

  ```
  c:\windows\system32\ping –n 10   www.ust.hk
  ```

- When the Ping program terminates, stop the packet capture in Wireshark.

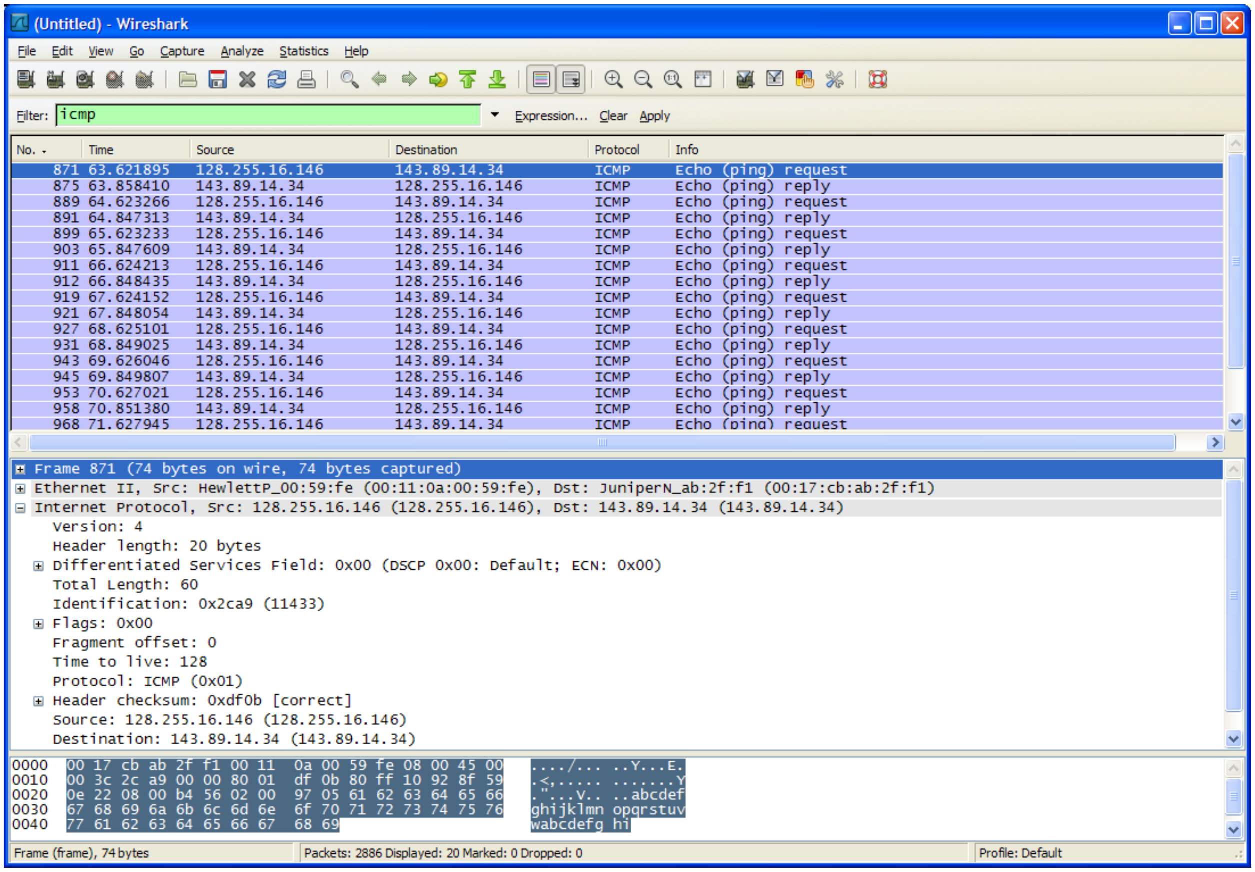
At the end of the experiment, your Command Prompt Window should look something like Figure 1. From this window we see that the source ping program sent 10 query packets and received 10 responses. Note also that for each response, the source calculates the round-trip time (RTT), which for the 10 packets is on average 224 msec.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

H:\>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=236ms TTL=239
Reply from 143.89.14.34: bytes=32 time=224ms TTL=239
Reply from 143.89.14.34: bytes=32 time=224ms TTL=239
Reply from 143.89.14.34: bytes=32 time=224ms TTL=239
Reply from 143.89.14.34: bytes=32 time=223ms TTL=239
Reply from 143.89.14.34: bytes=32 time=223ms TTL=239
Reply from 143.89.14.34: bytes=32 time=223ms TTL=239
Reply from 143.89.14.34: bytes=32 time=224ms TTL=239
Reply from 143.89.14.34: bytes=32 time=223ms TTL=239
Reply from 143.89.14.34: bytes=32 time=223ms TTL=239

Ping statistics for 143.89.14.34:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 223ms, Maximum = 236ms, Average = 224ms

H:\>
```

Figure 1 Command Prompt window after entering Ping command.

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 128.255.16.146 and 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.



Filter: icmp

```
No. -  Time         Source           Destination      Protocol  Info
   871 63.621895    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   875 63.858410    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   889 64.623266    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   891 64.847313    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   899 65.623233    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   903 65.847609    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   911 66.624213    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   912 66.848435    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   919 67.624152    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   921 67.848054    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   927 68.625101    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   931 68.849025    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   943 69.626046    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   945 69.849807    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   953 70.627021    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
   958 70.851380    143.89.14.34     128.255.16.146   ICMP      Echo (ping) reply
   968 71.627945    128.255.16.146   143.89.14.34     ICMP      Echo (ping) request
```

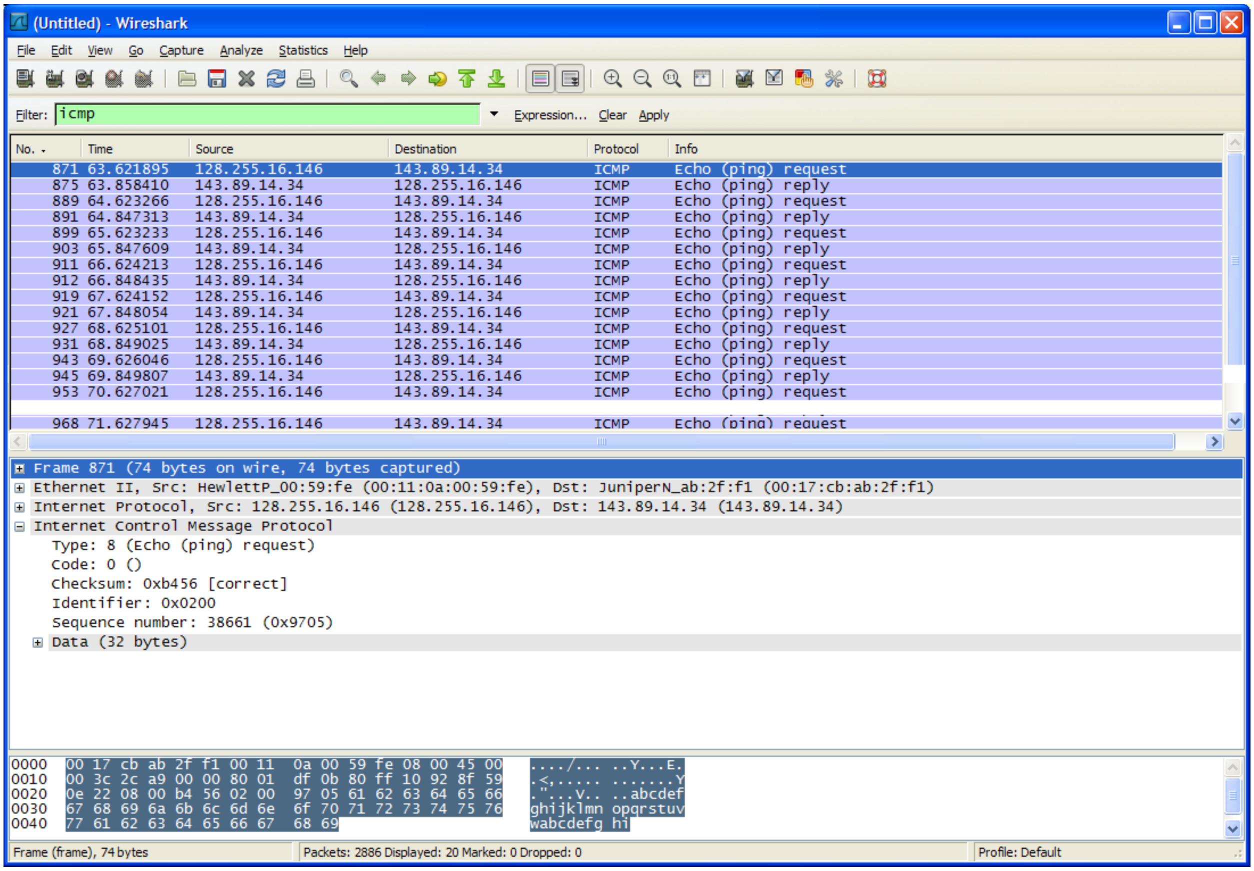
```
⊞ Frame 871 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: HewlettP_00:59:fe (00:11:0a:00:59:fe), Dst: JuniperN_ab:2f:f1 (00:17:cb:ab:2f:f1)
⊟ Internet Protocol, Src: 128.255.16.146 (128.255.16.146), Dst: 143.89.14.34 (143.89.14.34)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0x2ca9 (11433)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0xdf0b [correct]
    Source: 128.255.16.146 (128.255.16.146)
    Destination: 143.89.14.34 (143.89.14.34)
```

```
0000  00 17 cb ab 2f f1 00 11  0a 00 59 fe 08 00 45 00   ..../... ..Y...E.
0010  00 3c 2c a9 00 00 80 01  df 0b 80 ff 10 92 8f 59   .<,..... .......Y
0020  0e 22 08 00 b4 56 02 00  97 05 61 62 63 64 65 66   ."...V.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Frame (frame), 74 bytes | Packets: 2886 Displayed: 20 Marked: 0 Dropped: 0 | Profile: Default

Figure 2 Wireshark output for Ping program with Internet Protocol expanded.

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 4.23 of text.) Also note that this ICMP packet contains a checksum, an identifier, and a sequence number.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: icmp
Expression... Clear Apply

```
No.     Time        Source            Destination       Protocol  Info
    871 63.621895   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    875 63.858410   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    889 64.623266   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    891 64.847313   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    899 65.623233   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    903 65.847609   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    911 66.624213   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    912 66.848435   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    919 67.624152   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    921 67.848054   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    927 68.625101   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    931 68.849025   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    943 69.626046   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
    945 69.849807   143.89.14.34      128.255.16.146    ICMP      Echo (ping) reply
    953 70.627021   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request

    968 71.627945   128.255.16.146    143.89.14.34      ICMP      Echo (ping) request
```

```
⊞ Frame 871 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: HewlettP_00:59:fe (00:11:0a:00:59:fe), Dst: JuniperN_ab:2f:f1 (00:17:cb:ab:2f:f1)
⊞ Internet Protocol, Src: 128.255.16.146 (128.255.16.146), Dst: 143.89.14.34 (143.89.14.34)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0 ()
    Checksum: 0xb456 [correct]
    Identifier: 0x0200
    Sequence number: 38661 (0x9705)
  ⊞ Data (32 bytes)
```

```
0000  00 17 cb ab 2f f1 00 11  0a 00 59 fe 08 00 45 00   ..../... ..Y...E.
0010  00 3c 2c a9 00 00 80 01  df 0b 80 ff 10 92 8f 59   .<,..... .......Y
0020  0e 22 08 00 b4 56 02 00  97 05 61 62 63 64 65 66   ."...V.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

```
Frame (frame), 74 bytes    Packets: 2886 Displayed: 20 Marked: 0 Dropped: 0    Profile: Default
```

Figure 3 Wireshark capture of ping packet with ICMP packet expanded.

What to Hand In:

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 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.

You should answer the following questions:

1. What is the IP address of your host? What is the IP address of the destination host?
2. Why is it that an ICMP packet does not have source and destination port numbers?
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?
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?

2. ICMP and Traceroute

Now we will continue our investigation of ICMP by capturing the packets generated by the *tracert* program. You may recall that the *tracert* program can be used to figure out the path a packet takes from source to destination. *Tracert* is discussed in Section 1.6 and in Section 4.4 of the text.

Traceroute (*tracert*) is implemented in different ways in Unix/Linux 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 will use the native Windows *tracert* program.

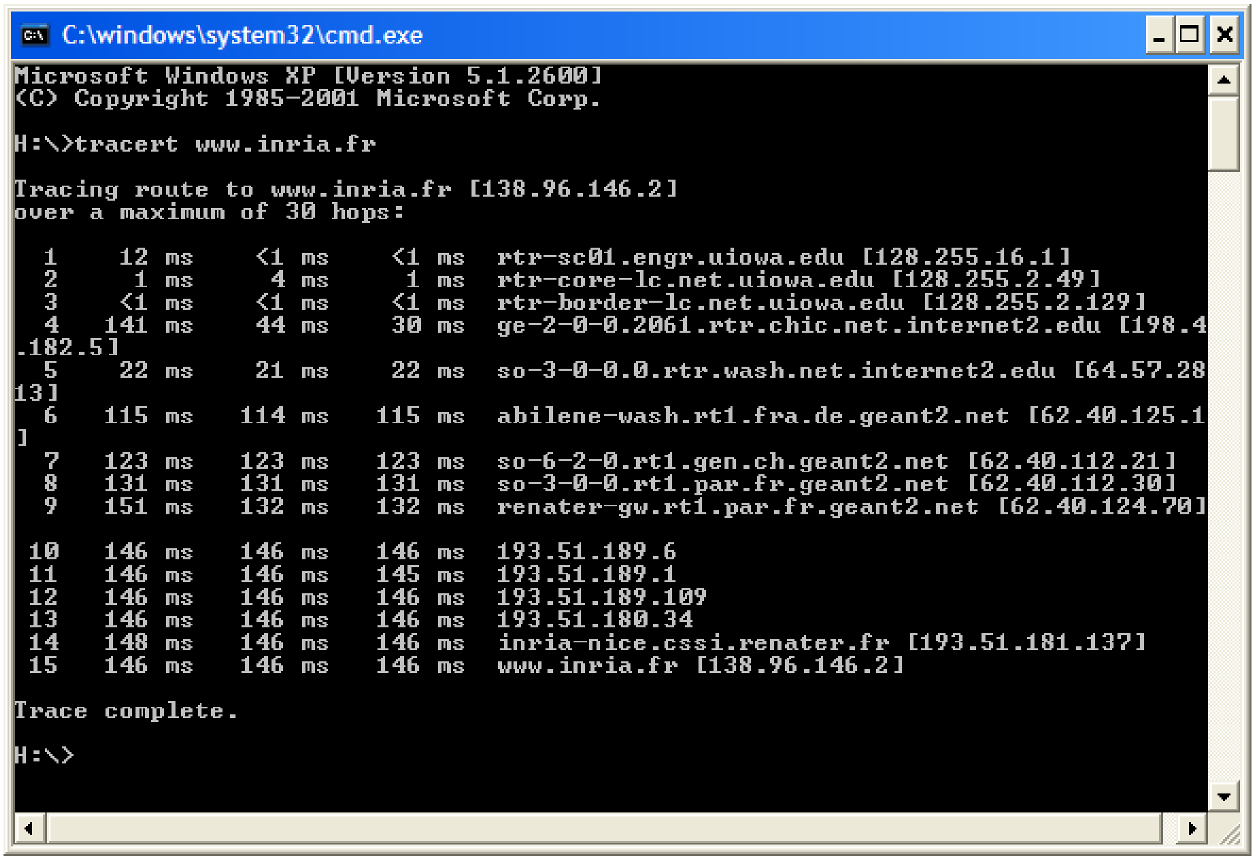
To capture the packet stream generated by a *tracert* command, do the following:

- Begin by opening the Windows Command Prompt window
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
- At the Windows command prompt, enter:

  tracert www.inria.fr

  to trace the route to the web server at INRIA, a research institute in France. If your path environment is not set up, you may need to use the full path name:
  c:\windows\system32\tracert www.inria.fr
- When the Traceroute program terminates, stop packet capture in Wireshark.

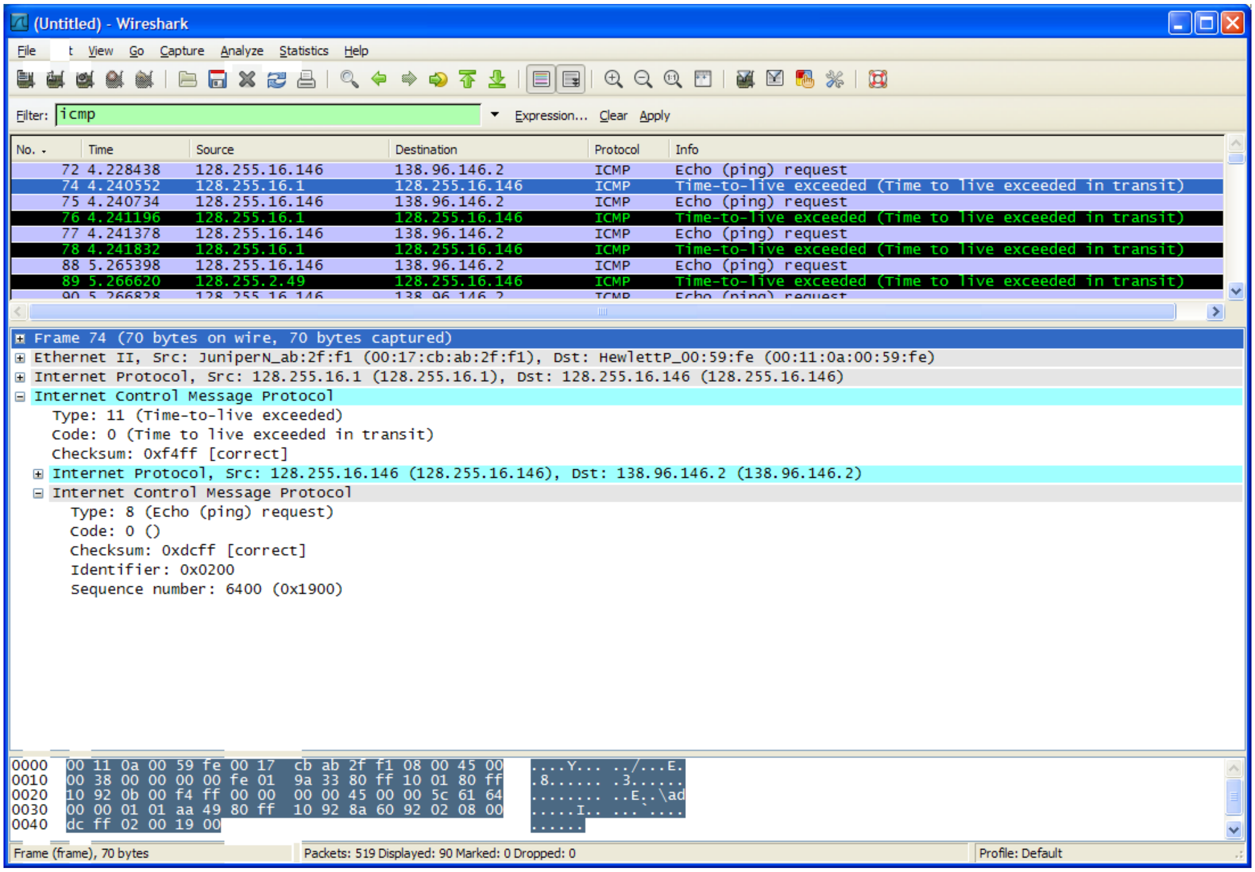
At the end of the experiment, your Command Prompt Window should look something like Figure 4. From this figure we see that for each TTL value, the source program sends three probe packets. *Tracert* displays the RTTs for each of the probe packets, as well as the IP address (and possibly the name) of the router that returned the ICMP TTL-exceeded message.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

H:\>tracert www.inria.fr

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

  1    12 ms    <1 ms    <1 ms  rtr-sc01.engr.uiowa.edu [128.255.16.1]
  2     1 ms     4 ms     1 ms  rtr-core-lc.net.uiowa.edu [128.255.2.49]
  3    <1 ms    <1 ms    <1 ms  rtr-border-lc.net.uiowa.edu [128.255.2.129]
  4   141 ms    44 ms    30 ms  ge-2-0-0.2061.rtr.chic.net.internet2.edu [198.4
.182.5]
  5    22 ms    21 ms    22 ms  so-3-0-0.0.rtr.wash.net.internet2.edu [64.57.28
13]
  6   115 ms   114 ms   115 ms  abilene-wash.rt1.fra.de.geant2.net [62.40.125.1
]
  7   123 ms   123 ms   123 ms  so-6-2-0.rt1.gen.ch.geant2.net [62.40.112.21]
  8   131 ms   131 ms   131 ms  so-3-0-0.rt1.par.fr.geant2.net [62.40.112.30]
  9   151 ms   132 ms   132 ms  renater-gw.rt1.par.fr.geant2.net [62.40.124.70]

 10   146 ms   146 ms   146 ms  193.51.189.6
 11   146 ms   146 ms   145 ms  193.51.189.1
 12   146 ms   146 ms   146 ms  193.51.189.109
 13   146 ms   146 ms   146 ms  193.51.180.34
 14   148 ms   146 ms   146 ms  inria-nice.cssi.renater.fr [193.51.181.137]
 15   146 ms   146 ms   146 ms  www.inria.fr [138.96.146.2]

Trace complete.

H:\>
```

Figure 4 Command Prompt window displays the results of the Traceroute program.

Figure 5 displays the Wireshark window for an ICMP packet returned by a router. Note that this ICMP error packet contains more fields than the Ping ICMP messages.



Filter: icmp

```
No. .  Time        Source          Destination     Protocol  Info
   72 4.228438   128.255.16.146  138.96.146.2    ICMP      Echo (ping) request
   74 4.240552   128.255.16.1    128.255.16.146  ICMP      Time-to-live exceeded (Time to live exceeded in transit)
   75 4.240734   128.255.16.146  138.96.146.2    ICMP      Echo (ping) request
   76 4.241196   128.255.16.1    128.255.16.146  ICMP      Time-to-live exceeded (Time to live exceeded in transit)
   77 4.241378   128.255.16.146  138.96.146.2    ICMP      Echo (ping) request
   78 4.241832   128.255.16.1    128.255.16.146  ICMP      Time-to-live exceeded (Time to live exceeded in transit)
   88 5.265398   128.255.16.146  138.96.146.2    ICMP      Echo (ping) request
   89 5.266620   128.255.2.49    128.255.16.146  ICMP      Time-to-live exceeded (Time to live exceeded in transit)
   90 5.266828   128.255.16.146  138.96.146.2    ICMP      Echo (ping) request
```

```
⊞ Frame 74 (70 bytes on wire, 70 bytes captured)
⊞ Ethernet II, Src: JuniperN_ab:2f:f1 (00:17:cb:ab:2f:f1), Dst: HewlettP_00:59:fe (00:11:0a:00:59:fe)
⊞ Internet Protocol, Src: 128.255.16.1 (128.255.16.1), Dst: 128.255.16.146 (128.255.16.146)
⊟ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
  ⊞ Internet Protocol, Src: 128.255.16.146 (128.255.16.146), Dst: 138.96.146.2 (138.96.146.2)
  ⊟ Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0 ()
      Checksum: 0xdcff [correct]
      Identifier: 0x0200
      Sequence number: 6400 (0x1900)
```

```
0000  00 11 0a 00 59 fe 00 17  cb ab 2f f1 08 00 45 00   ....Y... ../...E.
0010  00 38 00 00 00 00 fe 01  9a 33 80 ff 10 01 80 ff   .8...... .3......
0020  10 92 0b 00 f4 ff 00 00  00 00 45 00 00 5c 61 64   ........ ..E..\ad
0030  00 00 01 01 aa 49 80 ff  10 92 8a 60 92 02 08 00   .....I.. ...`....
0040  dc ff 02 00 19 00                                  ......
```

Frame (frame), 70 bytes | Packets: 519 Displayed: 90 Marked: 0 Dropped: 0 | Profile: Default

Figure 5 Wireshark window of ICMP fields expanded for one ICMP error packet.

What to Hand In:

For this part of the lab, you should hand in a screen shot of the Command Prompt window. Whenever possible, when answering the questions 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.

Answer the following questions:

1. What is the IP address of your host? What is the IP address of the target destination host?
2. 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?
3. 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?
4. Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?
5. 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?
6. 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?