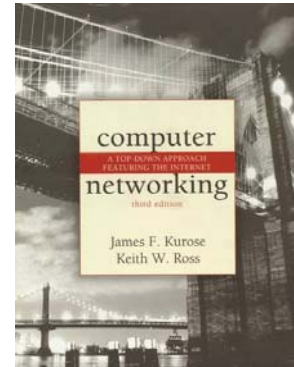# Ethereal Lab: ICMP

Computer Networking: A Top-down Approach Featuring the Internet, 3rd edition.

Version: July 2005

In this lab, we'll explore several aspects of the ICMP protocol:

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

Before attacking this lab, you're encouraged to review the ICMP material in the textbook (Section 4.4.3 in the 3rd edition.) 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

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.

Do the following[1]:

---

[1] If you are unable to run Ethereal live on a computer, you can download the zip file http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip and extract the file *ICMP-ethereal-trace-1*. The traces in this zip file were collected by Ethereal running on one of the author's computers, while performing the steps indicated in the Ethereal lab. Once you have downloaded the trace, you can load it into Ethereal 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.

- Let's begin this adventure by opening the Windows Command Prompt application (which can be found in your Accessories folder).
- Start up the Ethereal packet sniffer, and begin Ethereal packet capture.
- 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.
- When the Ping program terminates, stop the packet capture in Ethereal.

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.

```
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% loss),
Approximate round trip times in milli-seconds:
    Minimum = 314ms, Maximum = 425ms, Average = 375ms

C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>_
```

Figure 1 Command Prompt window after entering Ping command.

Figure 2 provides a screenshot of the Ethereal 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.
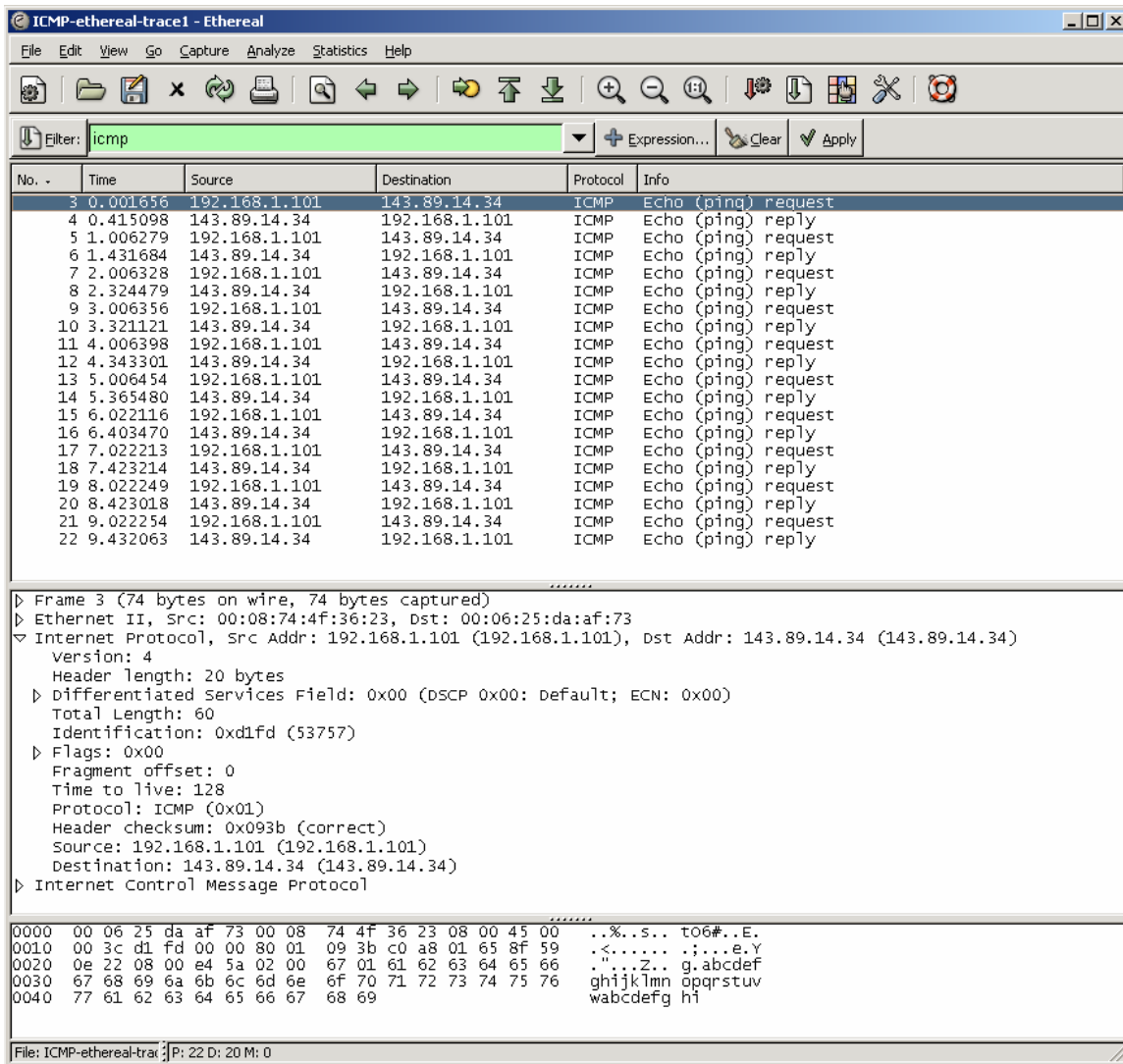
Figure 2 Ethereal output for Ping program with Internet Protocol expanded.

Figure 3 focuses on the same ICMP packet 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 4.21 of text.) Also note that this ICMP packet contains a checksum, an identifier, and a sequence number.
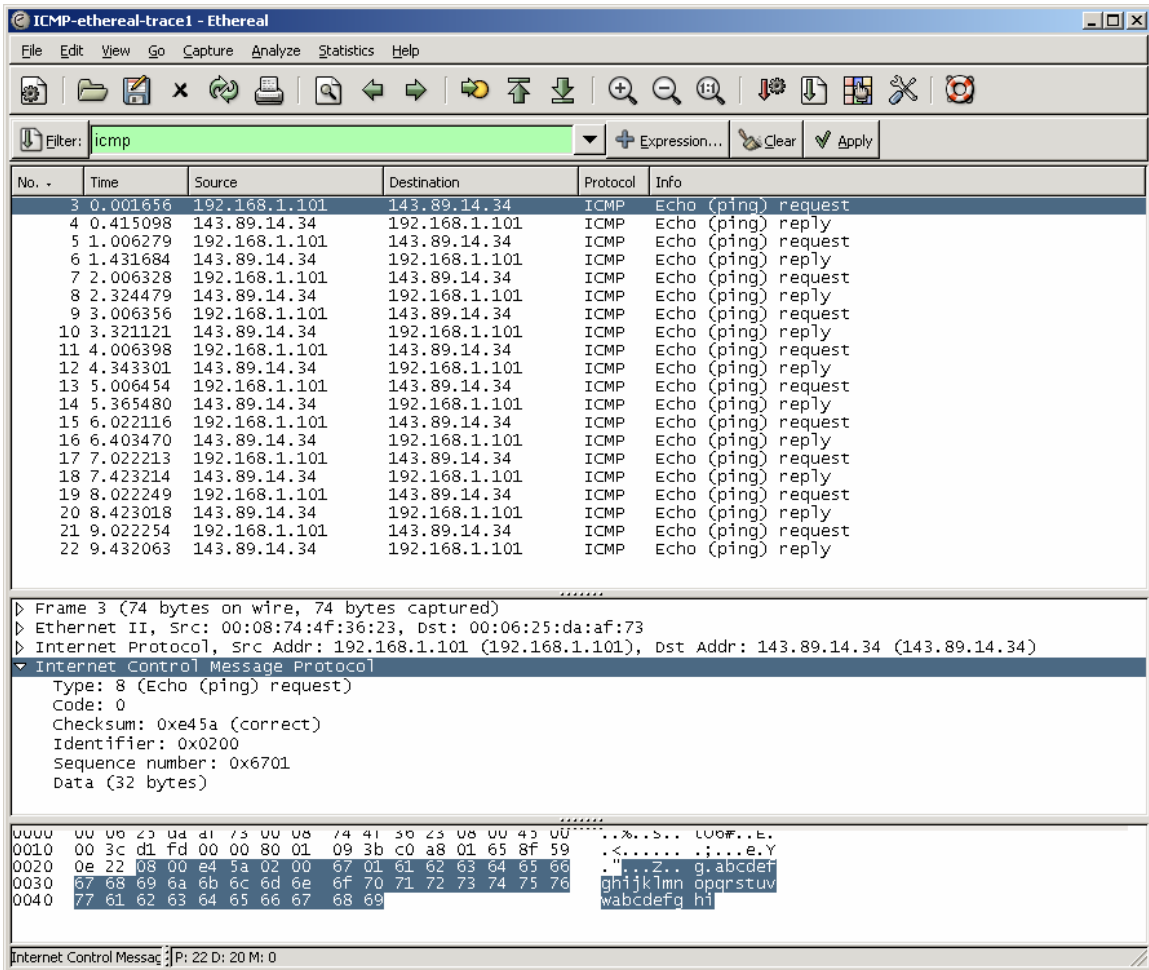
Figure 3 Ethereal capture of ping packet with ICMP packet expanded.

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

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 1.6 and in Section 4.4 of the text.

Traceroute 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'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 Ethereal IP lab since it provides additional functionality that we'll need there.

Do the following[2]:

- Let's begin by opening the Windows Command Prompt application (which can be found in your Accessories folder).
- Start up the Ethereal packet sniffer, and begin Ethereal packet capture.
- 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.

[2] If you are unable to run Ethereal live on a computer, you can download the zip file http://gaia.cs.umass.edu/ethereal-labs/ethereal-traces.zip and extract the file *ICMP-ethereal-trace-2*. The traces in this zip file were collected by Ethereal running on one of the author's computers, while performing the steps indicated in the Ethereal lab. Once you have downloaded the trace, you can load it into Ethereal 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.

- When the Traceroute program terminates, stop packet capture in Ethereal.

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 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.

```
Command Prompt                                                          _ □ ×
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 [24.128.190.197]
  4    16 ms    16 ms    15 ms  bar02-p6-0.ndhmhe1.ma.attbb.net [24.128.0.101]
  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 [193.251.241.133]
 11    97 ms    98 ms    98 ms  P9-0.AUUCR1.Aubervilliers.opentransit.net [193.251.243.29]
 12    98 ms    98 ms   108 ms  P6-0.BAGCR1.Bagnolet.opentransit.net [193.251.241.93]
 13   104 ms   106 ms   103 ms  193.51.185.30
 14   114 ms   114 ms   117 ms  grenoble-pos1-0.cssi.renater.fr [193.51.179.238]
 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]
 17   113 ms   114 ms   112 ms  www.inria.fr [138.96.146.2]

Trace complete.

C:\WINDOWS\SYSTEM32>_
```
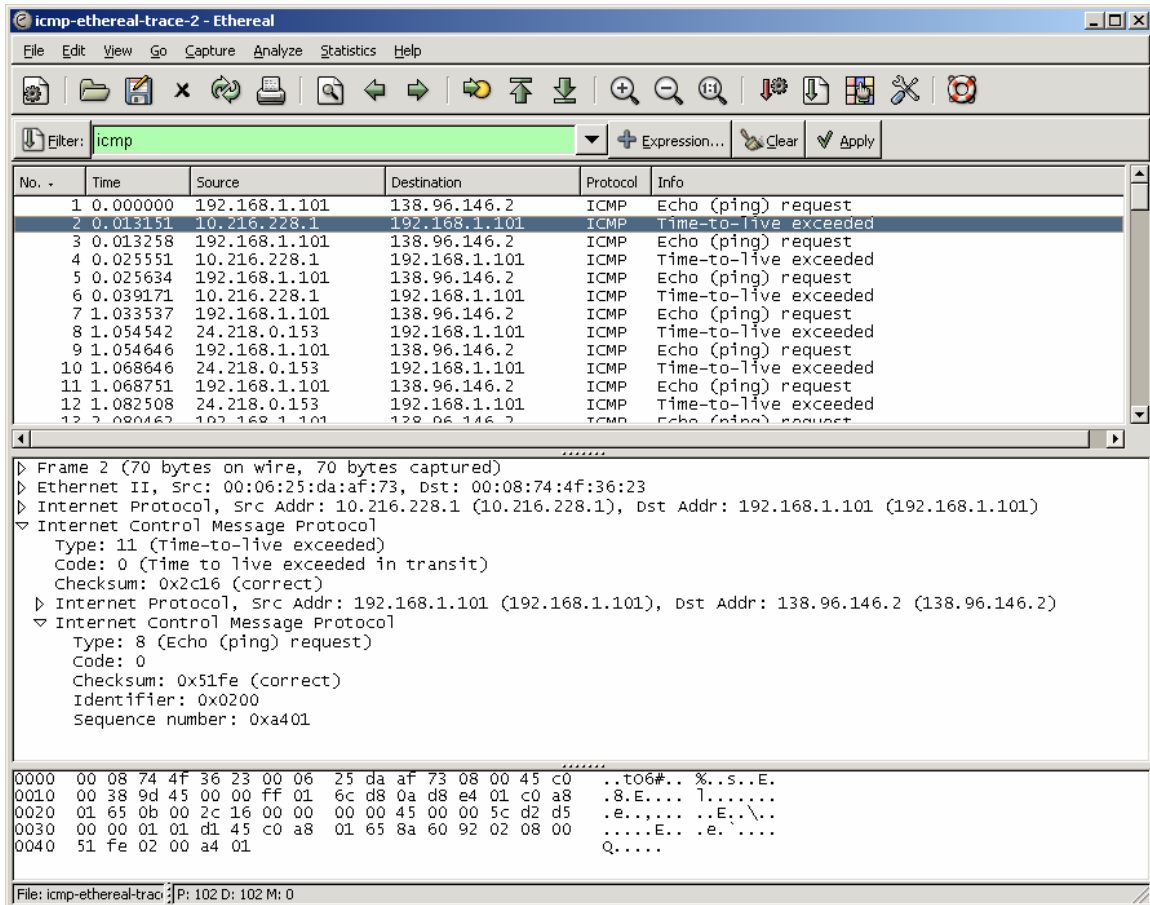
Figure 4 Command Prompt window displays the results of the Traceroute program.

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

Figure 5 Ethereal window if 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 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.

Answer the following questions:

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

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?
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?

3. Extra Credit

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 Ethereal to capture any response from the target host. Provide an Ethereal screenshot for the response as well as an analysis of the response.