



Assignments > Lab 7: ICMP

Lab 7: ICMP

[▼ Hide Assignment Information](#)

Instructions

Lab 7: ICMP

Instructions:

- Support all answers by a screenshot of your Wireshark and Command Prompt windows. Annotate screenshots to explain your answer.
- Submissions must be through Brightspace.
- There is a 24-hour grace period after the due date without a penalty. Late submissions will not be accepted.

Note: This lab is mostly adapted from materials provided by the authors of *Computer Networking: A Top-Down Approach*. All rights reserved.

Introduction

In this lab, we will explore several aspects of the ICMP protocol including ICMP messages generating by the Ping program and the Traceroute program. 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.

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:

- Open the Windows Command Prompt.
- Start up the Wireshark packet sniffer and begin Wireshark packet capture.
- Type `ping -n 10 hostname` in the command line (without quotation marks), where `hostname` is a host of your choice (e.g., `www.uwindsor.ca`, `www.ust.hk`, `www.ahmedsakr.com`, `www.inria.fr`, etc.). 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 Wireshark.

At the end of the experiment, your Command Prompt Window should show that the source ping program sent 10 query packets and received 10 responses. Note also that the source calculates the round-trip time (RTT) for each response and their average.

Figure 1 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. In this example, the source ping program is in Massachusetts and the destination Ping program is in Hong Kong. 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 a public address of the form 143.89/16. 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.

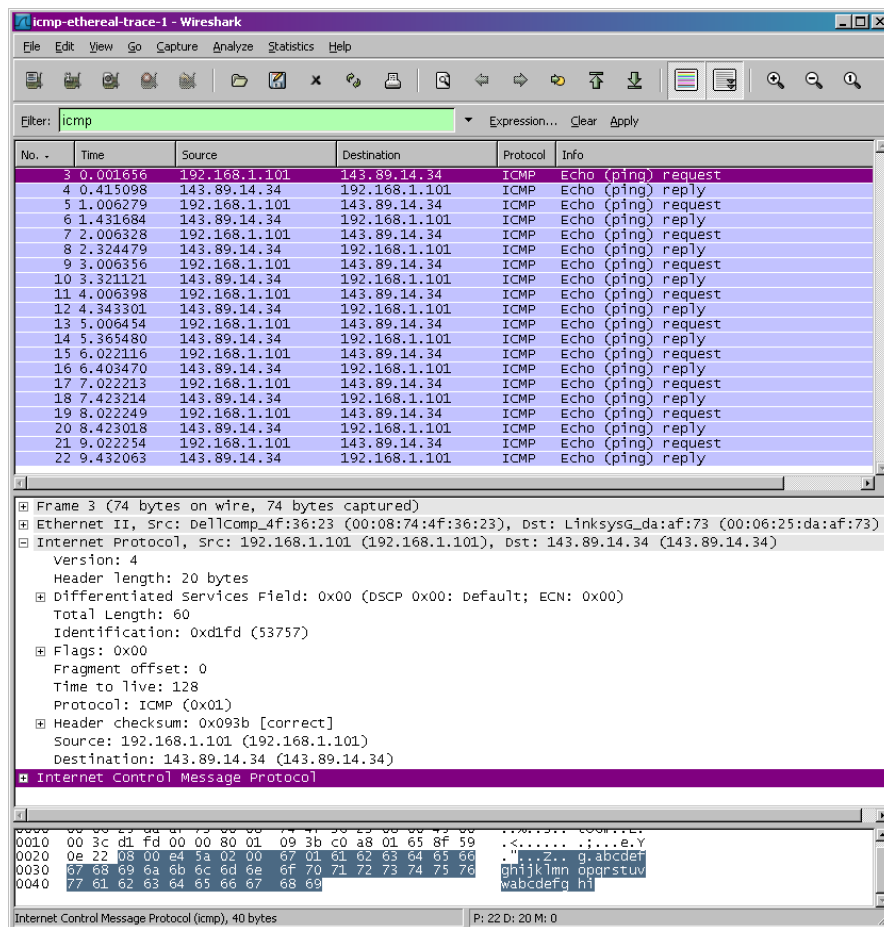


Figure 1 Wireshark output for Ping program with Internet Protocol expanded.

Figure 2 shows the same ICMP with expanded ICMP protocol information. Observe that this ICMP packet is of Type 8 and Code 0; a so-called ICMP “echo request” packet. Also note that this ICMP packet contains a checksum, an identifier, and a sequence number.

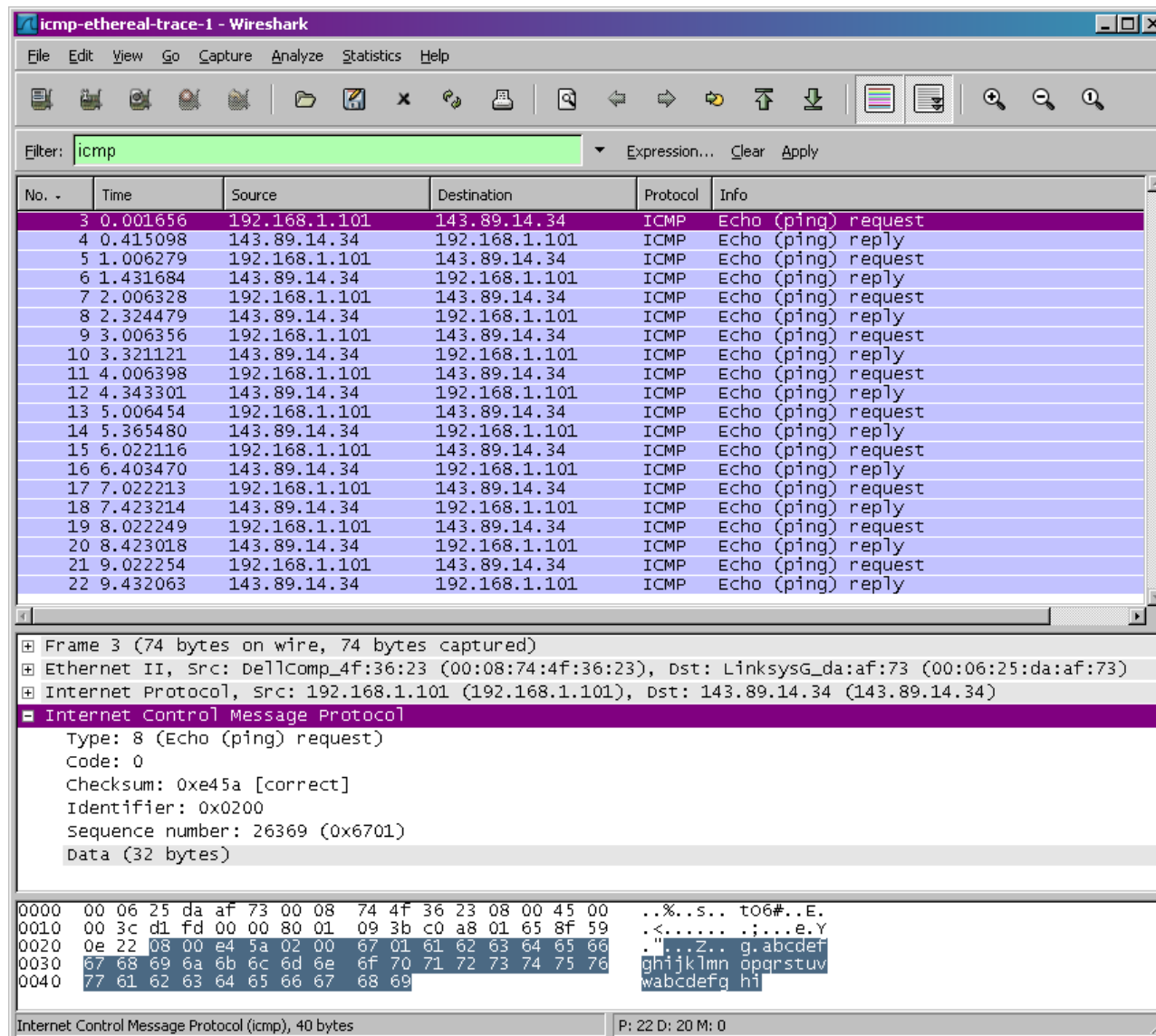


Figure 2 Wireshark capture of ping packet with ICMP packet expanded.

Questions

Now answer the following questions:

1. What is the IP address of your host? What is the IP address of the destination host?
2. 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?
3. 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?

ICMP and Traceroute

Let's now continue our ICMP adventure by capturing the packets generated by the Traceroute program. Traceroute program can be used to figure out the path a packet takes from source to destination.

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

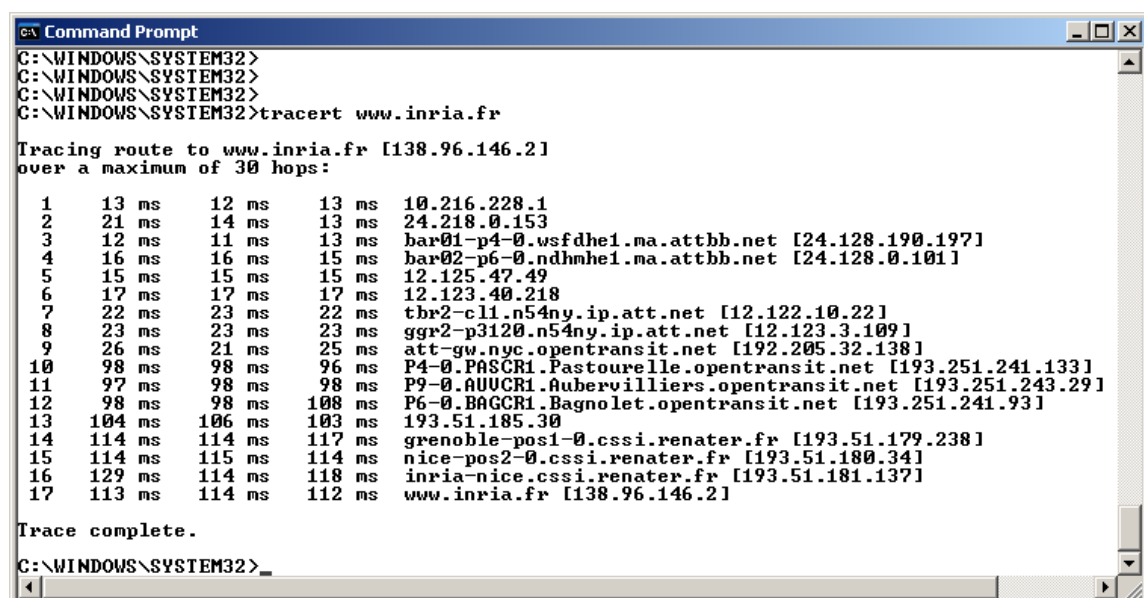
number. In Windows, the source sends a series of ICMP packets to the target destination. For both operating systems, the program operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same 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 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.

Do the following:

- Open the Windows Command Prompt.
- Start up the Wireshark packet sniffer and begin Wireshark packet capture.
- Type either `tracert hostname` in the command line (without quotation marks), where hostname is a host of your choice (e.g., `www.uwindsor.ca`, `www.ust.hk`, `www.ahmedsagr.com`, `www.inria.fr`, etc.). Note that on a Windows machine, the command is `tracert` and not `traceroute`. Then run the Traceroute program by typing return.
- 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 3. 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.

Figure 4 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.



```

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:
  0  13 ms  12 ms  13 ms  10.216.228.1
  1  21 ms  14 ms  13 ms  24.218.0.153
  2  12 ms  11 ms  13 ms  bar01-p4-0.wsfdhe1.ma.attbb.net [24.128.190.197]
  3  16 ms  16 ms  15 ms  bar02-p6-0.ndhmhe1.ma.attbb.net [24.128.0.101]
  4  15 ms  15 ms  15 ms  12.125.47.49
  5  17 ms  17 ms  17 ms  12.123.40.218
  6  22 ms  23 ms  22 ms  tbr2-cl1.n54ny.ip.att.net [12.122.10.22]
  7  23 ms  23 ms  23 ms  ggr2-p3120.n54ny.ip.att.net [12.123.3.109]
  8  26 ms  21 ms  25 ms  att-gw.nyc.opentransit.net [192.205.32.138]
  9  98 ms  98 ms  96 ms  P4-0.PASCR1.Pastourelle.opentransit.net [193.251.241.133]
 10  97 ms  98 ms  98 ms  P9-0.AUUCR1.Aubervilliers.opentransit.net [193.251.243.29]
 11  98 ms  98 ms  108 ms  P6-0.BAGCR1.Bagnolet.opentransit.net [193.251.241.93]
 12  104 ms  106 ms  103 ms  193.51.185.30
 13  114 ms  114 ms  117 ms  grenoble-pos1-0.cssi.renater.fr [193.51.179.238]
 14  114 ms  115 ms  114 ms  nice-pos2-0.cssi.renater.fr [193.51.180.34]
 15  129 ms  114 ms  118 ms  inria-nice.cssi.renater.fr [193.51.181.137]
 16  113 ms  114 ms  112 ms  www.inria.fr [138.96.146.2]

Trace complete.
C:\WINDOWS\SYSTEM32>_

```

Figure 3 Command Prompt window displays the results of the Traceroute program.

Questions

Now answer the following questions: Support all answers by a screenshot of Wireshark window as needed. Annotate screenshots to explain your answer. You should also include in a screenshot of the Command Prompt window.

4. What is the IP address of your host? What is the IP address of the target destination host?
5. 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?
6. 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?
7. Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?
8. 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?

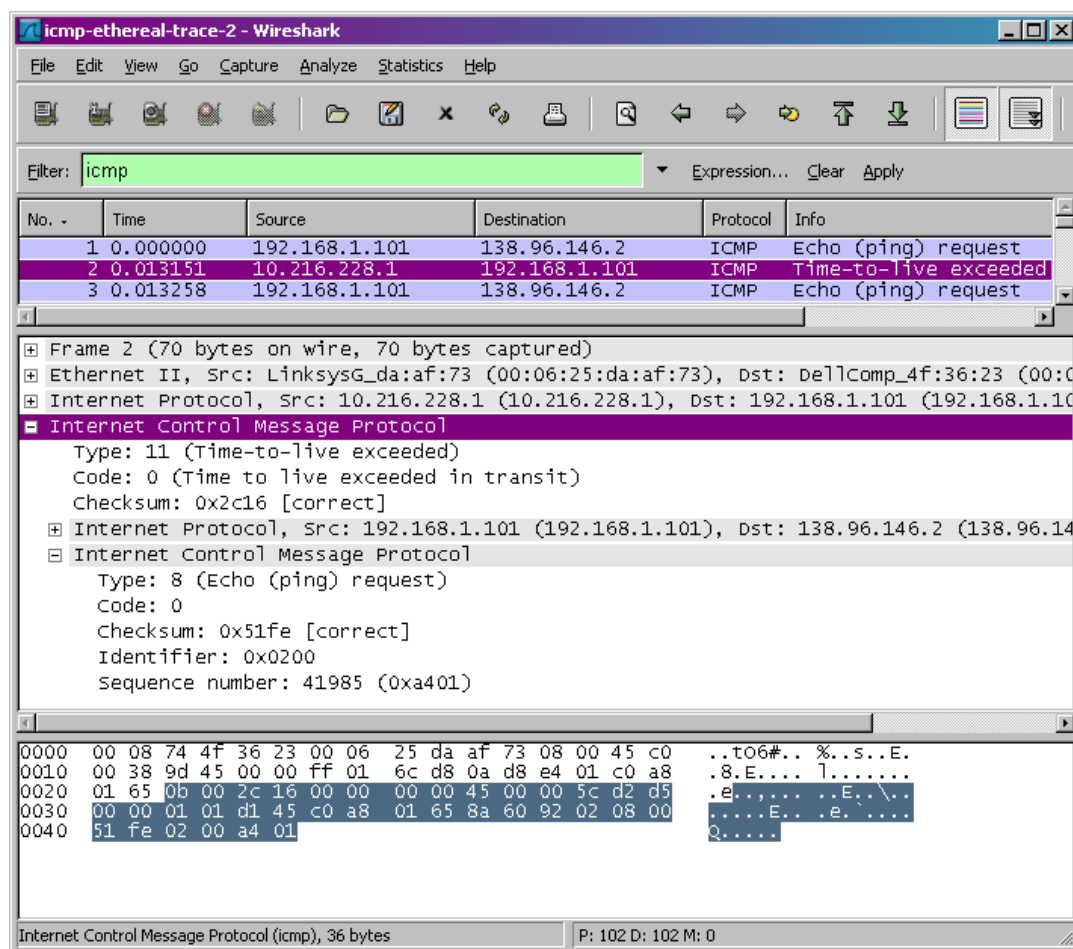


Figure 4 Wireshark window of ICMP fields expanded for one ICMP error packet.

Due on Nov 10, 2023 11:59 PM

Available on Oct 30, 2023 12:01 AM. **Access restricted before availability starts.**

Available until Nov 11, 2023 11:59 PM. **Submission restricted after availability ends.**

Submit Assignment

Submission is restricted outside of availability dates.