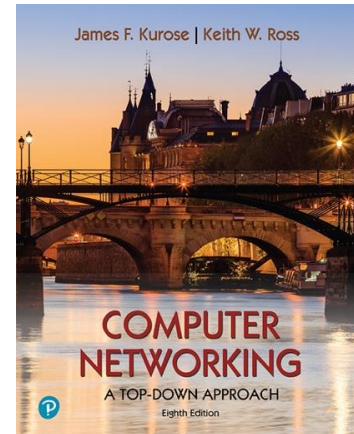


Wireshark Lab: ICMP v8.0

Supplement to *Computer Networking: A Top-Down Approach, 8th 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

© 2005-2020, J.F Kurose and K.W. Ross, All Rights Reserved



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

- ICMP messages generated 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 section 5.6 of the text¹. 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.

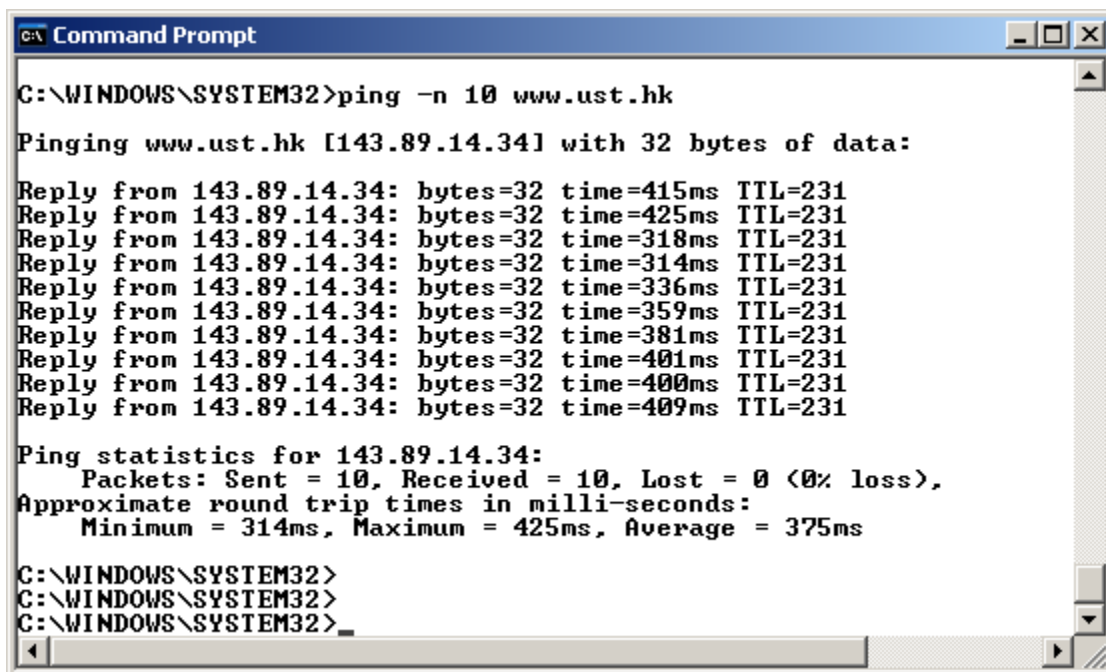
Do the following²:

¹ References to figures and sections are for the 8th edition of our text, *Computer Networks, A Top-down Approach, 8th ed.*, J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2020.

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.

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

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 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.

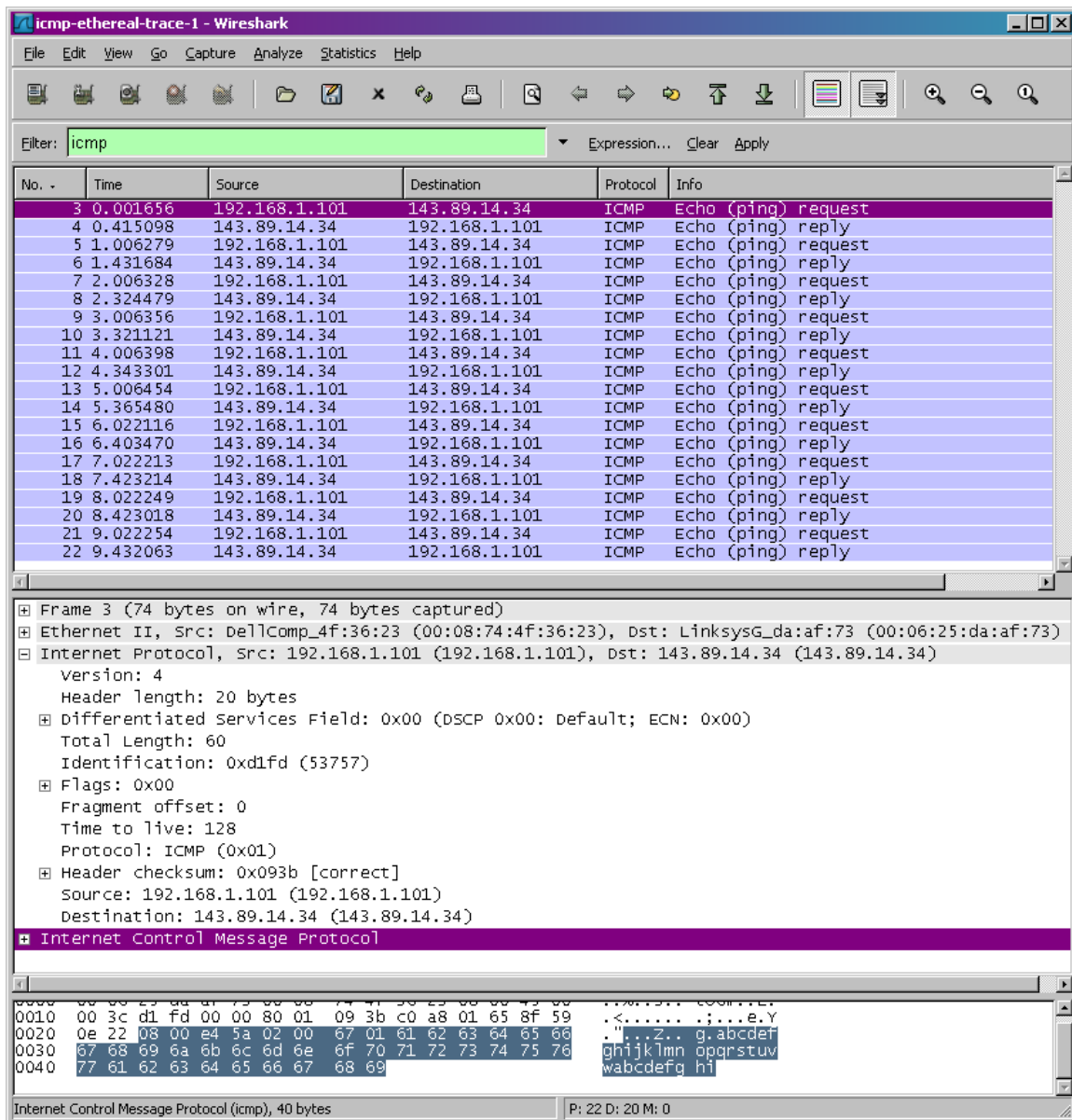


Figure 2 Wireshark output for Ping program with Internet Protocol expanded.

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

icmp-ethereal-trace-1 - Wireshark					
File Edit View Go Capture Analyze Statistics Help					
Filter: icmp Expression... Clear Apply					
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 (00:06:25:da:af:73)					
Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 143.89.14.34 (143.89.14.34)					
Internet Control Message Protocol					
Type: 8 (Echo (ping) request)					
Code: 0					
Checksum: 0xe45a [correct]					
Identifier: 0x0200					
Sequence number: 26369 (0x6701)					
Data (32 bytes)					

0000	00 06 25 da af 73 00 08 74 4f 36 23 08 00 45 00	..%..s.. tO6#..E.
0010	00 3c d1 fd 00 00 80 01 09 3b c0 a8 01 65 8f 59	.<..... :;....e.Y
0020	0e 22 08 00 e4 5a 02 00 67 01 61 62 63 64 65 66	."...Z.. g.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	wabcedfg hi

Internet Control Message Protocol (icmp), 40 bytes				P: 22 D: 20 M: 0	
--	--	--	--	------------------	--

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