Wireshark Lab: ICMP v6.0

Supplement to *Computer Networking: A Top-Down Approach*, 6^{th} *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

© 2005-21012, J.F Kurose and K.W. Ross, All Rights Reserved



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 section 4.4.3 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:

² If you are unable to run Wireshark live on a computer, you can download the zip file

¹ References to figures and sections are for the 6th edition of our text, *Computer Networks, A Top-down Approach, 6th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2012.*

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.

```
🔤 Command Prompt
                                                                                                     - 🗆 ×
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
Reply from 143.89.14.34: bytes=32 time=318ms
                                                                         TTL=231
                                                                         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.

🗖 icr	np-eth	ereal-	trace-1	- Wire	eshark															_ 🗆	×
Eile	<u>E</u> dit	<u>V</u> iew	<u>Go</u> <u>C</u> a	apture	<u>A</u> nalyze	<u>S</u> tati	stics <u>I</u>	<u>H</u> elp													
	<u>i</u>	0	@ (D	<u> </u>	x	e,	8	٩	4	⇔	¢)	Ŧ	⊉		Ţ	€,	Θ	0	
Eilter	: licm	p									▼ E	xpression	n <u>⊂</u> le	ar	Apply						
	<u> </u>							Deatin	abian			Drokoso	-								
NO. 4	2 0	ine 1. 001 (856	1 9 2	168 1	1.01		1/2	80 1/	2/		TCMP	Ech	0.1	(ning)	reques	+				
	4 0	.415	098	143	.89.14	1.34		192.	168.1	.101		ICMP	Ech		(ping)	reply					
	5 1	.006	279	192	.168.1	L.101		143.	89.14	.34		ICMP	Ech	0 ((ping)	reques	t				
	61		584 278	143	.89.14 168.1	1.34 101		192.	168.1 80 14	.101		ICMP	Ech		(ping)	reply	+				
	82	.3244	479	143	.89.14	1.34		192.	168.1	.101		ICMP	Ech		(ping)	reply					
	93	.006	356	192	.168.1	L.101		143.	89.14	.34		ICMP	Ech	0 ((ping)	reques	t				
	10 3	.3211	121	143	.89.14	1.34		192.	168.1	.101		ICMP	Ech		(ping)	reply	+				
	12 4	.343	301	143	.89.14	1.34		192.	168.1	. 101		ICMP	Ech		(ping)	reply	L.				
	13 5	.0064	454	192	.168.1	1.101		143.	89.14	.34		ICMP	Ech	0 ((ping)	reques	t				
	14 5	.3654	480	143	.89.14	1.34		192.	168.1	.101		ICMP	Ech	0 0	(ping)	reply					
	$15 \ 6$. 0221	L16 170	192	.168.1	L.101		143.	169.14	. 34		ICMP	ECN		(ping)	reques	t				
	17 7	.022	213	192	.168.1	L.101		143.	89.14	.34		ICMP	Ech		(ping)	reques	t				
	18 7	.423	214	143	.89.14	1.34		192.	168.1	.101		ICMP	Ech	0 ((ping)	reply					
	19 8	3.0223	249	192	.168.1	L.101		143.	89.14	.34		ICMP	Ech	0 0	(ping)	reques	t				
	20 8	0.423	J18 254	192	168.14	+.34 101		143	108.1	34		TCMP	ECH		(ping) (ning)	repty	+				
	22 9	.432	063	143	.89.14	4.34		192.	168.1	.101		ICMP	Ech	0	(ping)	reply					
1																		-		J	
	- ama	2 (7)	1 hvt.	as or	wire	7/	but as	cant	urad)	1											<u> </u>
	therr	et T	T. Sri	с: De		, / / a 4f:	36:23	(00)	:08:74	.:4f:3	36:23). Dst	t: Lir	nks	vsG da	:af:73	(00:	06:25	: da : a	of:73	0
E I	nterr	net Pi	notoci	ol. s	inc: 19	92.16	8.1.1	.01 (1	92.16	8.1.1	101).	Dst:	143.8	39.	14.34	(143.8	9.14.	34)			<u></u>
	Vers	ion:	4	- , -							,,					•					
	Неас	ler 10	enqth	: 20	bytes																
Đ	Diff	erent	tiate	d ser	vices	Fiel	d: 0>	(00 (D	SCP 0)x00:	Defa	ult; E	ECN: (0×0	0)						
	Total Length: 60																				
	Ider	tifi	catio	n: O×	d1fd	(5375	7)														
+	Flag	js: 0:	×00																		
	Frag	ment	offs	et: C)																
	⊤im∈	e to	live:	128																	
	Prot	ocol	: ICM	P (0x	:01)	-	-														
+	Head	ler cl	hecksi	um: C)x093b	[cor	rect]														
	Sour	ce: :	192.1	68.1.	101 (192.1	68.1.	101)													
	Dest	inati	ion: .	143.8 1 Mos	9.14.: 5200	34 (1 Droto	43.89 col	9.14.3	54) 54)												
	iceri		Shiel O	I Mes	saye i	-1000	COT														
1																					Þ
0010	00	20 2	5 dd 1 fd	<u>.</u>		- 1 0	2 26	-0 -2- -0 -9	00 0	5 95 5 0f	50	•••/	J C		· · · L ·						
0020	00 0e	22 🚺	18 0 <u>0</u>	e4 5	a 02 <u>0</u>	u 09 10 <u>67</u>	2 50 7 01_	cu að 61 6 <u>2</u>	63 <u>6</u>	4 65	66	· · · ·	z <u>a</u>	. al	ocdefi						
0030	67	68 6	9 6a	6b 6	c 6d 6	e 6f	70	71 72	73 74	4 75	76	ghijk	lmn õ	pqr	rstuv						
0040	77	61 6	2 63	64 6	5 66 6	17 68	5 69					wabćd	erg h								-
Intern	et Cont	rol Mess	age Pro	tocol (ic	mp), 40 b	ytes					P: 22	2 D: 20 M	:0								

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.

🕂 icmp-ethereal-trace-1 - Wireshark	<u>- 0 ×</u>
Eile Edit View Go Capture Analyze Statistics Help	
	0
Eilter: 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 143.89.14.34 192.1081.101 ICMP ECDO (ping) reply	
6 1 431684 143 89 14 34 192 168 1 101 TMP Echo (pring) 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 5.000434 192.108.1.101 143.09.14.34 1CMP ECHO (ping) reply	
14 5.6 02116 192 168 1 101 143 89 14 34 TCMP Echo (ping) request	
16 6.403470 143.89.14.34 192.168.1.101 ICMP Echo (ning) 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	
F Frame 3 (74 bytes on wire, 74 bytes captured)	
E Ethernet II. Src: DellComp 4f:36:23 (00:08:74:4f:36:23) Dst: Linksvs5 da:af:73 (00:06:25:da	af:73)
E Toternet Protocol Src: 192 168 1 101 (192 168 1 101) Dst: 143 89 14 34 (143 89 14 34)	u
Internet Costrol Message Protocol	
Type: 8 (Echo (ning) request)	
code, o (colo (ping) request)	
Checksum: Uxe45a [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%	
0010 00 3c d1 fd 00 00 80 01 09 3b c0 a8 01 65 8f 59	
0020 0e 22 08 00 e4 5a 02 00 67 01 61 62 63 64 65 66	
10030 67 68 69 63 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijkimn opgrstuv	
0040 77 62 63 64 63 66 67 68 69 Wabcderg ni	
Internet Control Message Protocol (icmp), 40 bytes P: 22 D: 20 M: 0	

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 printout³ 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:

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.

- 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 1.4 and in Section 4.4 of the text.

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.

Do the following⁴:

- Let's begin 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 *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.

⁴ If you are unable to run Wireshark live on a computer, you can download the zip file <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u> 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.

(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.

• 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. 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.

🔤 Con	imand Pro	ompt					
C:\WI	NDOWS\S NDOWS\S	YSTEM3	2> 2>				
Č:\WI	NDOWS	YSTEM3	2>				
C:\WI	NDOWS\S	YSTEM3	2≻tı	acert	www	.inria.fr	
Tuaci	ng nout	o to u		innia f	. г	120 06 146 21	
over	a maxir	un of	30 1	10105:	r i	130.70.140.23	
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]	
1 4	16 ms	16	ms	15	MS	bar02-p6-0.ndhmhe1.ma.attbb.net [24.128.0.10]]	
2	15 MS	15	ms	15	ms	12.125.47.49	
1 2	17 MS	17	ms	17	ms	12.123.40.218	
1 6	ZZ MS	23	ms	22	ms	tDr2-c11.n54ny.1p.att.net [12.122.10.22]	
N N	23 MS	23	ms	23	ms	ggr2-p3120.n54ny.1p.att.net [12.123.3.107]	
14	20 MS	21	ms	25	ms	att-gw.nyc.opentransit.net 1172.205.32.1361	22.1
11	78 MS	70	ms	70	MS	PG-G AUUCP1 Aubanuillians anatusasit act [102 251 241.1	331
112	77 MS	70	ms	100	ms	PC-0 PACCPI Pagealat apartuanait pat [102 2E1 241 02]	. 47]
12	76 MS	100	ms	100	ms	102 E4 40E 20	
14	114 ms	114	IIIS DO	117	ms mo	173.31.103.30 guapable-peri-0 cosi perstan fu [193 E1 179 229]	
12	114 ms	115	115	114	ms me	grenulle=pos1=0.0001. $relater.rr [173.31.177.230]$	
16	179 ms	114	115	119	me me	invia-nice cosi venatev fv [193 51 181 137]	
17	113 ms	114	me	112	me	$\mu_{\mu\mu}$ invia fy [138 96 146 2]	
111	113 M3	111	115	114	пъ	www.infid.if [136.70.140.2]	
Trace	comple	te.					
C:\WI	NDOWS	YSTEM3	2>_				-

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 many more fields than the Ping ICMP messages.



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 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. 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?
- 9. 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?