Lab Exercise - ICMP

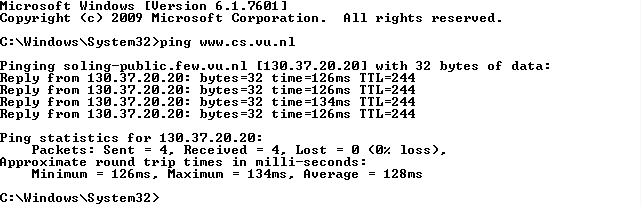
**Objective**

To see how ICMP (Internet Control Message Protocol) is used. ICMP is a companion protocol to IP that helps IP to perform its functions by handling various error and test cases. It is covered in §5.6.4 of your text. Review that section before doing this lab.

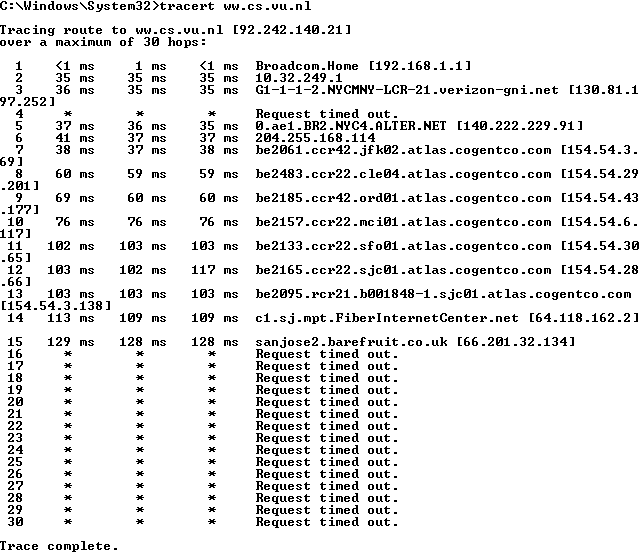
**Step 1: Capture a Trace**

*Proceed as follows to capture a trace of ICMP traffic that results from* ping *and* traceroute*; alternatively, you may use a supplied trace.*

1. *Pick a remote computer such as a server as a target, e.g., www.cs.vu.nl, and check that you can successfully ping it.* To run ping, simply bring up a command line and type, e.g., “ping www.cs.vu.nl” for our choice of computer. This command should work across Windows, Mac or Linux for our choice of computer. You may have to stop the ping program from end-lessly sending pings by pressing ctrl-C. You are looking at the ping output to see that the re-mote computer replies to your ping requests.



2. *Check to see that you can* traceroute *to the same computer while invoking* traceroute *with options to send ICMP traffic rather than other traffic such as UDP*. To do this, run it as “tracert www.cs.vu.nl” (Windows, uses ICMP) or “traceroute –I www.cs.vu.nl” (Mac/Linux). That was a “minus capital i” option, which tells traceroute to send ICMP probes. It may take a little while to run, printing the next line of output after a pause of several seconds or more. You are looking at the output to see that you can find most of the path to the remote computer. Each line of output gives information about the next segment of the network path. When part of the path cannot be found, then the traceroute output will be a “\*” entry that denotes unanswered probes; it is common to have some “\*” entries, and this is OK, but for an interesting trace they should not be the majority of the path.



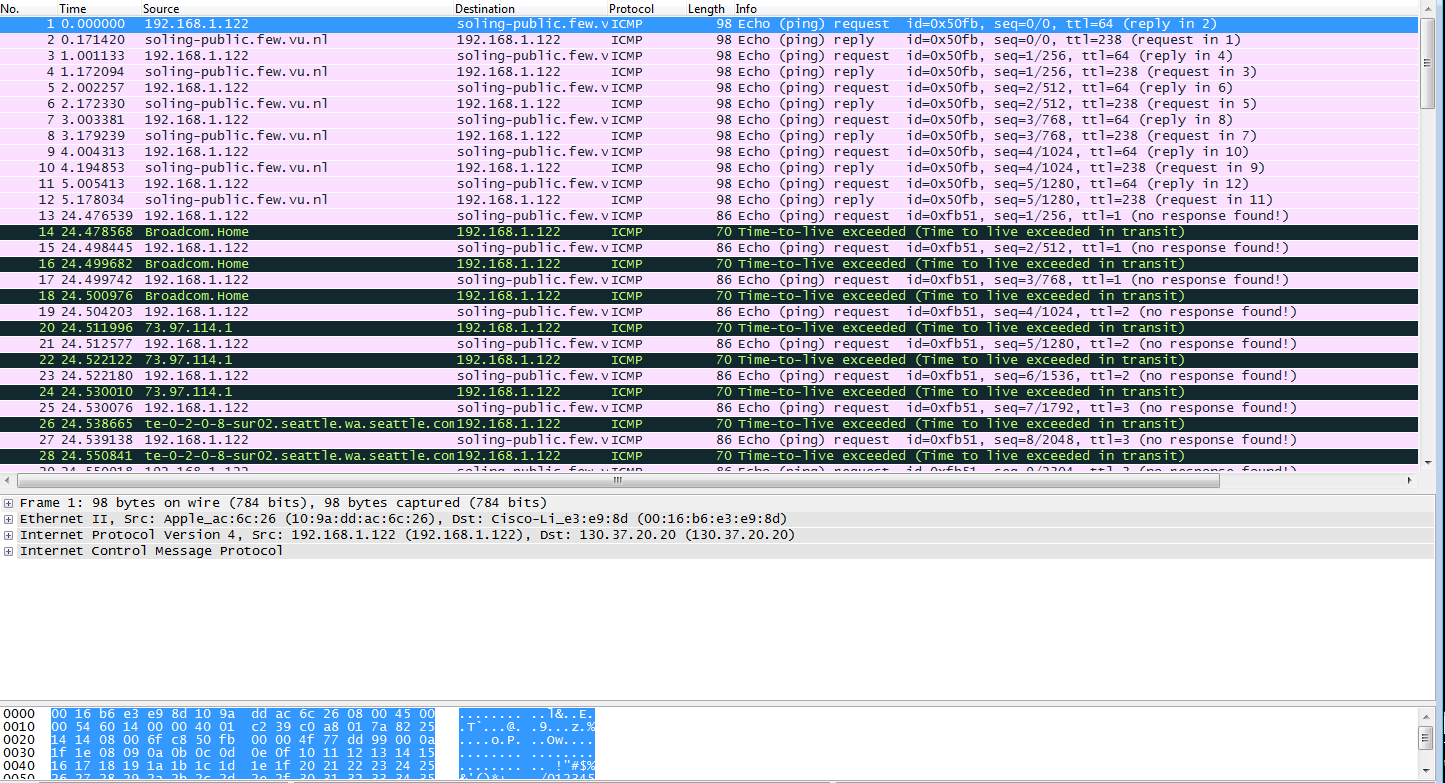
3. *Launch Wireshark and start a capture with a filter of* “icmp*“. Make sure to check “enable net-work name resolution”. This will translate the IP addresses of the computers and routers sending packets into names, which will help you to recognize the organizations on the network path taken by your packets.*

Since I am using the icmp.pcap, I shall skip this step.

4. *When the capture is started, repeat the* ping *command you tested, wait a few seconds, and then repeat the* traceroute *command as well.* This time, the ICMP packets sent and received by these two programs will be recorded by Wireshark.

Since I am using the icmp.pcap, I shall skip this step.

5. *After the commands are complete, return to Wireshark and use the menus or buttons to stop the trace*. You should have a short trace similar to that shown in the figure below. We have expanded the detail of the ICMP header for a ping request packet in our view. Be sure to save the output from the ping and traceroute commands. You will need it for the later steps.



**Step 2: Echo (ping) Packets**

*Start your exploration by selecting an echo (ping) request and reply packet at the start of the trace. Expand the ICMP block (by using the “+” expander or icon) to see the ICMP header and payload details:*

• The ICMP header starts with a Type and Code field that identify the kind of ICMP message. Look to see the values for an echo request and an echo reply and how they compare.

• The Type/Code is followed by a 16-bit checksum over the complete ICMP message, to check that it was received correctly.

• Next comes an Identifier and Sequence Number field. These fields are used to link echo request and reply packets together. Look at the Identifier and Sequence Number values for several requests and replies. Compare the values of a request and matching reply, and of successive re-plies. Note that Wireshark may show these fields in two ways: as a Big Endian (BE) value and a Little Endian (LE). The difference is the order in which the bits are organized into bytes, e.g., 00000001 on the wire might represent “1” or “256” depending on whether the first bit transmit-ted is the least (LE) or most (BE) significant bit.

• Finally, there is a Data field as the ICMP payload. This field is variable length.

*Answer the following questions to demonstrate your understanding of ICMP echo messages:*

1. *What are the Type/Code values for an ICMP echo request and echo reply packet, respectively?*

IMCP echo request : Type = 8 Code =0

IMCP echo reply : Type = 0 Code = 0

2. *How do the Identifier and Sequence Number compare for an echo request and the corresponding echo reply?*

IMCP echo request and corresponding reply packets have the same identifier and sequence number.

3. *How do the Identifier and Sequence Number compare for successive echo request packets?*

For successive echo request packets the identifier is still the same however the sequence number is increased by one (BE.)

4. *Is the data in the echo reply the same as in the echo request or different?*

Yes, the data in echo reply and echo request are the same.

**Step 3: TTL Exceeded (traceroute) Packets**

*Next, explore* traceroute *traffic by selecting any Time Exceeded ICMP packet in your trace. Expand the ICMP block (by using the “+” expander or icon) to see the ICMP header and payload details:*

• The ICMP header starts with a Type and Code field that identify the kind of ICMP message, just as for echo packets. Look to see the values for a TTL Exceeded packet and how they compare to the echo packets.

• The Type/Code is followed by a 16-bit checksum over the complete ICMP message to check that it was received correctly, just as for echo packets.

• The next structure you will see (apart from a possible length field that is part of modern implementations of ICMP) is an IP header! How can this be? For ICMP error messages that are produced by an error during forwarding, such as TTL Exceeded, the start of the packet that triggered the error is included in the payload of the ICMP packet. That is why there is an IP header inside the ICMP packet. Depending on how much of the packet that caused the error is included in the ICMP payload, you may see its headers beyond IP. In the case of our traceroute traffic, this will be an ICMP header of the echo request packet.

*Draw a picture of one ICMP TTL Exceeded packet to make sure that you understand its nested structure. On your figure, show the position and size in bytes of the IP header, ICMP header with details of the Type/Code and checksum subfields, and the ICMP payload. Within the ICMP payload, draw another rec-tangle that shows the overall structure of the contents of the payload.* As usual, your figure can simply show the packet as a long, thin rectangle. To work out sizes, observe that when you click on a protocol block in the middle panel (the block itself, not the “+” expander) then Wireshark will highlight the bytes it corresponds to in the packet in the lower panel and display the length at the bottom of the window.

**ICMP Payload**

Size = 28 bytes

**ICMP**

**Header**

Type = 11

Code = 0

Checksum = 0xf4ff

Size = 8 bytes

**IP**

**Header**

Size = 20 bytes

**IMCP Header**

Type = 8

Code = 0

Checksum = 0xfca3

Size = 8 Bytes

**IP Header**

Size = 20 bytes

*Answer the following questions:*

*1. What is the Type/Code value for an ICMP TTL Exceeded packet?*

Type = 11

Code = 0

2. *Say how the receiver can safely find and process all the ICMP fields if it does not know ahead of time what kind of ICMP message to expect.* The potential issue, as you have probably noticed, is that different ICMP messages can have different formats. For instance, Echo has Sequence and Identifier fields while TTL Exceeded does not.

All IMCP fields seem to all have the same code ( 0 ), which may be used to safely find and process all IMCP packets.

3. *How long is the ICMP header of a TTL Exceeded packet?* Select different parts of the header in Wireshark to see how they correspond to the bytes in the packet.

The IMCP header of a TTL exceeded packet has a size of 8 bytes: 1 byte for to type, 1 bytes for code, and 2 byes for the checksum. I assume that there are 4 empty bits at the end to round out the number.

4. *The ICMP payload contains an IP header. What is the TTL value in this header? Explain why it has this value*. Guess what it will be before you look!

I guessed that the TTL value of the IP header in the ICMP payload would be 0, but it was actually 1. I will assume that this is the smallest possible time to live.

**Step 4: Internet Paths**

The source and destination IP addresses in an IP packet denote the endpoints of an Internet path, not the IP routers on the network path the packet travels from the source to the destination. traceroute is a utility for discovering this path. It works by eliciting ICMP TTL Exceeded responses from the router 1 hop away from the source towards the destination, then 2 hops away from the source, then 3 hops, and so forth until the destination is reached. The responses will identify the IP address of the router. The output from traceroute normally prints the information for one hop per line, including the measured round trip times and IP address and DNS names of the router. The DNS name is handy for working out the organization to which the router belongs. Since traceroute takes advantage of common router implementations, there is no guarantee that it will work for all routers along the path, and it is usual to see “\*” responses when it fails for some portions of the path.

*Look at the* traceroute *portion of the trace, which will have a series of ICMP echo request packets fol-lowed by ICMP TTL Exceeded packets.* The echo requests are sent from the source (your computer) to the destination whose path is being probed. The TTL Exceeded packets are coming from routers along the path back to your computer, triggered by the TTL field counting down to zero.

*By looking at the details of the packets, answer the following questions:*

1. *How does your computer (the source) learn the IP address of a router along the path from a TTL exceeded packet? Say where on this packet the IP address is found.* You might proceed by looking at an echo packet to see the source and destination IP addresses. The routers along the path will have a different IP address, and this address will be present on the TTL Exceeded packet. If you are unsure, you can examine the traceroute text output to see the IP addresses of routers and look for these addresses on the TTL Exceeded packets.

The IP header inside the IMCP payload contains the IP information of the router along the path at that point in the path.

2. *How many times is each router along the path probed by* traceroute*?* Look at the TTL Ex-ceeded responses and see if you can discern a pattern.

3 times.

3. *How does your computer (the source) craft an echo request packet to find (by eliciting a TTL Ex-ceeded response) the router N hops along the path towards the destination? Describe the key attributes of the echo request packet.* The echo request packets sent by traceroute are probing successively more distant routers along the path. You can look at these packets and see how they differ when they elicit responses from different routers.

The source computer sets the echo request to N. If it is hoping to the first router, N=1 so the time to live is one, if it is the second router, N= 2 so time to live is 2, etc.