uc3m | Universidad Carlos III de Madrid	**IP and ICMP traffic analysis lab session** [1] **S06**	
Redes y Servicios de Comunicaciones		2023-2024

1. Introduction and Objectives

Traffic analysis is a valuable tool for understanding the operation of communication protocols. This protocol analysis allows us to observe the sequence of messages exchanged, as well as to delve into the details of how protocols operate.

The objective of this lab session is to handle a traffic analyzer that allows us to obtain data about the traffic that circulates through the Laboratory's network. By means of traffic analysis we can make sure that the network is working properly, as well as explore how the TCP/IP protocol stack works in a real environment.

In particular, in this lab session we will study several aspects of the ICMP protocol: the ICMP packets generated by the ping and traceroute programs, as well as the format and content of ICMP messages, and also that of the IP datagrams that carry them. In previous networking courses you have already used the Wireshark program, and even analyzed ICMP packets generated with ping (and packets from other protocols), but now you will be able to deepen your understanding of the protocols by taking advantage of the greater theoretical knowledge about them.

2. Lab session rules

Carefully read the statement until the end before the session.

It is recommended that the practice is done by groups of two students (although it is possible to do it individually).

In order to carry out the lab session, it is necessary to have a user account in the teaching laboratories of the Department of Telematics Engineering. The hosts use the Linux operating system. Note that this lab session can be also performed and/or trained using the virtual labs of the Telematics Department, available at https://aulavirtual.lab.it.uc3m.es/. It can also be done using the virtual machine providing the UC3M Virtual Laboratory [1] (note that this is different from the former).

To perform the lab, the `wireshark` traffic capture and analysis program will be used.

 *The parts of the statement between these symbols pose questions that are proposed throughout the session.* 

Estimated Lab duration: 100 minutes.

[1] This lab session is inspired by lab sessions proposed in "J. F. Kurose, K. W. Ross; "Computer Networking, a top-down approach", 5th edition, Pearson - Addison Wesley, 2009."

3. Work prior to the start of the lab session

Review in the course book the operation of the traceroute and ping commands (sections 1.4.3 and 4.4.3) and go through the preparation exercises available in Aula Global.

Review the operation of the `wireshark` program. For the use of the program, you can refer to the system manual pages or the online documentation (Wireshark Documentation), available at http://www.wireshark.org/docs/. It is also recommended to carefully read the wireshark help guide available together with this practice before performing the practice.

Prepare for the lab session by reading the exercise carefully. Think about the expected results of the practice. Review the theory where necessary. Study the manuals and command help pages.

4. Description of the lab session

1. ICMP and ping

Connect to a computer in the lab. All the lab questions refer to the lab computer, regardless of whether you are communicating with it locally or remotely through another computer. First, we will analyze the ICMP packets generated by the `ping` command. The `ping` tool is a very simple program that allows us to verify whether a host has IP connectivity (with another host) or not. The ping program on the source machine sends a packet addressed to the IP address of the destination machine; if the destination is working it sends a packet back to the source machine. The packets exchanged are ICMP packets.

a. Start the wireshark program and enable traffic capture on the eth0 interface (the one that connects the computer to the Internet).

b. Open a terminal and ping the machine "doc00X.lab.it.uc3m.es" (with X equal to the last digit of your NIA). Wait a few seconds and terminate the execution of the ping command using 'Ctrl-C'. In case you get no response from the target machine, repeat the section with "doc01X.lab.it.uc3m.es" or with "doc02X.lab.it.uc3m.es".

c. Once the ping command has finished executing, stop capturing packets in wireshark.

d. *Analyze the output of the ping command (do not analyze yet the traffic capture, this will be done later) and answer the following points (d.1-d.4):*

   d.1. How many ping packets did your machine send? How do you know?

   d.2. What is the IP address of doc00X.lab.it.uc3m.es (or wherever you pinged, for the rest of the sections assume we are referring to the successful ping)?

d.3. The output of the ping execution shows, for each ICMP message sent, a time value and a ttl value. Find and explain what those values are, and think of at least one way the ping could calculate them (not necessarily how it actually does it, just one way it could be done).

d.4. Use the ping command again but this time with the machine gbien.tel.uva.es (machine located in Valladolid). Compare the results in the time and ttl values with those obtained in d.3.

Next we are going to analyze the traffic generated by the ping command with the machine doc00X.lab.it.uc3m.es and that you will have captured with wireshark.

a. *Using a display filter, show the ICMP packets generated by the ping to doc00X.lab.it.uc3m.es. Which filter did you use?*

b. Display in wireshark the information about the first packet sent by your machine due to the ping command. Wireshark offers two views of the packet contents, one decoded and one with the contents in hexadecimal. The decoded one offers the information divided by header, so it is easy to go to the specific header you want to analyze. In our case we will see Ethernet, IP, and ICMP headers. Scroll down the IP header information and look at the different fields. For example, the protocol field has a value of 01 which means that there is an ICMP message in the IP datagram data field. If you click on any field you will see that field highlighted in the hexadecimal sequence that represents the frame in the other window. The fields in square brackets are not data included in the packet but additional information that wireshark has been able to deduce from the captured packet. For example, in the Echo request there is a [Response frame:] that indicates the packet number in the wireshark capture in which the corresponding Echo reply can be found (if it does not appear it means that wireshark has not been able to determine it or that the packet has not been captured). And in the Echo reply there is a [Request frame:] indicating the packet number in the wireshark capture in which to find the corresponding Echo request.

c. *Based on what you see in wireshark, draw the first packet sent by your machine due to the ping command. Specifically plot the various headers present in the packet, and in the correct order. It is not necessary to draw all the fields within each header, but do indicate all the address fields (with their value) in each header.*

d. Now let's explore the content of the ICMP message. So, display the ICMP information. Note that the ICMP message is of Type 8 and Code 0 - an Echo Request message. Also note that the ICMP message contains a checksum, an identifier and a sequence number.

e. *In fact, two identifiers and two sequence numbers appear in the decoded information. Are there really two identifiers and two sequence numbers in the ICMP message? Search the Internet for an explanation.*

f. *Examine the rest of the Echo Requests sent by your machine, you will see that the identifier is the same for all of them, while the sequence number is increasing. Why is that, how do you imagine the ping command uses these*

fields? Think that on the Internet, IP packets can get lost and jumbled, and the effect that has on what the ping is trying to do.

g. *Now examine the corresponding Echo Reply packets. What is the type and code for these packets? How many bytes are in the other fields of the ICMP header: checksum, sequence number and identifier?*

2. ICMP and traceroute

Next we will analyze the packets generated by the `traceroute` command. The `traceroute` tool can be used to discover the path that packets follow from a source to a destination.

The traceroute program is by default implemented differently on Unix/Linux and Windows. In this practice we will use the "-I" parameter of the Linux traceroute command (which requires administrator permissions), so it will work the same as the Windows tracert (equivalent command). The way it works is that the source sends a series of ICMP packets of type Echo Request (like those used by the ping program) to the destination machine: a first series of packets with TTL=1, a second series of packets with TTL=2, and so on. Remember that when a router forwards an IP packet it decrements the value of the TTL field of that packet by one. When a packet arrives at a router (which is not the destination of the packet) with TTL=1, the router discards the packet and sends an ICMP error packet back to the source.

h. Start wireshark and enable traffic capture.

i. Run the traceroute command to the machine www.uio.no with the "-I" parameter and launching it as superuser (sudo traceroute).

j. When the command has finished executing, stop the packet capture.

k. *Analyze the output of the traceroute command (not the traffic capture, which we will analyze later). You will notice that for each TTL value the source machine (your machine) has sent three packets. And the output of the command shows the RTT for each of these packets, as well as the IP address and in some cases the name of the router that returned the ICMP TTL-exceeded packet. Now answer the following points:*

k.1) Repeat the command execution but now also include the "-n" option. Explain the effect of the "-n".

k.2) Through how many routers do your packets traverse until they reach www.uio.no?

k.3) What is the IP address of the server www.uio.no?

k.4) If you look at the RTT measurements you have obtained with the traceroute command, there are hops where the delay increases significantly more than others (i.e. comparing the RTT to one router with the RTT to the previous router, the growth is higher than in other cases). Looking at the names of the routers (output of the command without the "-n" option) that are at the ends of those links, can you imagine what is the physical location of those routers? Compare locations with increments in the RTT between hops.

Next we are going to analyze the traffic generated by the traceroute command and that you have captured with the wireshark. Keep the output of the traceroute command in a window.

l. Using a display filter, locate among the captured packets, the packets generated by the executed traceoute.

m. Regarding the first IP packet generated by your machine (due to the traceroute), what does the IP packet contain in the data field, to which machine is the IP packet addressed, what value appears in the TTL field?

n. Examine the first ICMP reporting an error (TTL exceeded), which device generated this message, from which source IP address, what type and code does the ICMP message have, what does the message contain (in the data field of the ICMP message)?

o. Examine the last ICMP packets received by your machine, what are the differences between these packets and the rest of the previous ICMP packets received, why are they different?

References

[1] UC3M Virtual Laboratory: https://www.it.uc3m.es/uc3m_lab_virtual/

Complementary references

- GNU/Linux man pages.
- RFC 792. ICMP: INTERNET CONTROL MESSAGE PROTOCOL. J. Postel, 1981.
- Linux Networking HOWTO, Available at: http://www.tldp.org/HOWTO/Net-HOWTO/