CS3506 - Wireshark Lab 3: ICMP and Fragmentation Lab.

In this lab, we'll explore several aspects of the Internet Control Message Protocol (ICMP):

- ICMP messages generating by the ping program;
- ICMP messages generated by the traceroute program;
- ICMP responses generated by "false" UDP and TCP probes using the traceroute program;
- The format and contents of an ICMP message.

Ensure your PC is running Linux and remember that you have to run wireshark as the administrative user, so at the command line type "**sudo wireshark**" (without the quotation marks). When prompted for a password please enter your own password. You may first also need to run "**xhost +**" to ensure that the X Window System allows wireshark access to the display.

1. ICMP and Ping

Let's begin by capturing the packets generated by the Ping program. Recall that Ping is a simple tool that allows anyone (for example, a network administrator) to verify if a host is live or not. The Ping program sends a packet to the target IP address from the source host; if the target is alive, it 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:
Start up wireshark, and begin the packet capture.

- At the command line type "**ping –c 10 -n <hostname>**" where <hostname> is within the UCC network, e.g. **www.ucc.ie**. The argument "**-c 10**" indicates that 10 ping messages should be sent, while "**-n**" disables hostname lookup.
- When ping terminates, stop the packet capture in wireshark.

At the end of the experiment, your command window should show 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).
Note the summary statistics produced by ping and include in "Submission" sheets below.

Enter "**icmp**" in the wireshark filter display window and study the wireshark output. Assuming that no ping requests/responses were lost, 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 in the UCC subnet 143.239.0.0/16, as is the destination address.

Now let's zoom in on the first packet (sent by the client). We see that the IP datagram within this packet has protocol number 01, which is the protocol number for ICMP. This is what declares the payload of the IP datagram to be an ICMP packet. 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.

- *Now answer questions 1.1-1.4*

2. ICMP and Traceroute

Let's continue by capturing the packets generated by the traceroute program. The traceroute program can be used to figure out the path a packet takes from source to destination. 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 reduces it to 0 and sends an ICMP error packet back to the source.

We will now study the differences when using ICMP and TCP probes. Do the following:

- Start up wireshark, and type "**ip.host == <destination host IP>**". Make sure that you click the Apply button. The <destination host IP> is the IP of www.ucc.ie you found in the first part of the lab. The filter will clear up the display for you, allowing only the packets sent or received from the destination IP to be displayed. If the destination IP is wrong, the filter will hide all packets, or display irrelevant traffic.
- Run "**sudo traceroute -n -I -m 3 www.ucc.ie**". "**-n**" disables intermediary node hostname lookup, "**-I**" forces ICMP probes and "**-m 3**" sets the maximum TTL to 3. This command will send 3 packets with TTL=1, 3 packets with TTL=2, and 3 packets with TTL=3 at once.
- Run "**sudo traceroute -n -T -p 34567 -m 3 www.ucc.ie**". "**-T**" forces TCP probes, and "**-p 34567**" sets the destination port to 34567. TCP probes have the SYN flag set.
- Run "**sudo traceroute -n -T -p 80 -m 3 www.ucc.ie**". "**-p 80**" will send the probes to destination port 80, which is the default http port a web server usually operates on, and www.ucc.ie is a web server.
- When traceroute terminates, stop the packet capture in wireshark.

Notice that for each TTL value, the source sends three probe packets. traceroute displays the RTT for probe packets, as well as the IP address (and possibly the name) of the router that returned the ICMP TTL-exceeded message.
As the processing of messages is more CPU intensive and complicated than forwarding for routers, you might see the responses of the destination host arrive before the responses from the intermediary routers.

- *Now answer questions 2.1-2.4*

3. Fragmentation

Let's produce traffic with packets exceeding the Ethernet fragmentation threshold. Recall that the Ethernet Maximum Transmission Unit (MTU) is 1500 Bytes. You should be able to verify this by running "**ping -c 10 -M do -s 2500 -n www.ucc.ie**". "**-M**" do forces the DF (Don't Fragment) flag to be set, and "**-s 2500**" defines the ICMP payload size to be 2500 Bytes. Now find the maximum bytes of ICMP payload that don't cause fragmentation. Verify your answer.

- *Now answer question 3.1*

Do the following:

- Start up wireshark
- Select **"Edit" -> "Preferences" -> expand "Protocols" -> locate "IPv4"**
- Make sure "**Reassemble fragmented IP datagrams**" is **NOT CHECKED**
- Begin packet capture.
- Run "**ping -c 10 -s 2000 www.ucc.ie**".
- When ping terminates, stop the packet capture in wireshark.

Type "**ip.addr == 143.239.128.67**" in the wireshark filter display window and study the wireshark output. Verify that fragmented Probe Responses exist.

- *Now answer questions 3.2-3.4*

Submission - LAB 3: ICMP / Fragmentation

Please submit during this lab session or else at next week's lab session.

Student name:

Student ID:

1. ICMP and Ping
You should answer the following questions:

1.1 What is the IP address of your host? What is the IP address of the destination host?

1.2. Why is it that an ICMP packet does not have source and destination port numbers?

2. ICMP and Traceroute

2.1. Imagine you are a network administrator. You don't want to give any information about your network map, so you want to allow ping, but block incoming traceroute ICMP from the Internet. How do you think you could discern between the two? Do you think it would be easier to identify and block the traceroute request or the reply messages? Explain your answer.

2.2. Examine the last three ICMP packets sent by www.ucc.ie as a response to the ICMP probes. How are these packets different from the ICMP error packets? Why are they different?

2.3. Examine the three ICMP packets sent by www.ucc.ie as a response to the TCP probes to port 34567. Compare them to the responses sent by www.ucc.ie as a response to the TCP probes to port 80. How are they different?

2.4. After running this experiment, how do you think that a port scanner works like? (A port scanner finds all the "open" ports of a network host for UDP and TCP - “open” ports are called the ones that a service listens to and responds from)

3. Fragmentation

3.1. What is the maximum number of bytes of ICMP payload that doesn’t cause fragmentation? How can you verify your answer with “ping”?

3.2. Examine the first of the echo request packets. Where is the Don’t Fragment (DF) bit?

3.3. Examine the first echo response packet. It should be the first fragment. What information in the IP header indicates that the datagram has been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment?

3.4. Examine the second fragment of the echo response. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?