CS 145 Lab Exercise 3
Wireshark Lab: Internet Protocol and Traceroute Operation
A.Y. 2016-2017, 2nd Semester

1 Introduction

In this laboratory exercise you are going to explore the IP protocol (focusing on the IP datagram), as well as the operation of the `traceroute` program. As such, it would be helpful for you to review Lecture 8 (as well as Laboratory Exercise 1) before doing the laboratory exercise.

In order to generate a trace of IP datagrams for this lab, you will use the `traceroute` program to send a series of UDP packets (which are encapsulated within IP datagrams) towards a destination (in our case, the HTTP server hosting the website of the *Philippine Daily Inquirer* [`www.inquirer.net`]). The `traceroute` program operates by by first sending three datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of three datagrams towards the same destination with a TTL value of 2; it then sends a series of three datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1. If the TTL reaches 0, it returns an ICMP message (type 11 - TTL-exceeded) to the sending host. As a result of this behavior, a datagram with TTL of 1 (sent by the host executing `traceroute`) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram send with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram send with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this manner, the host executing `traceroute` can learn the identities of the routers between itself and the destination by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

2 Restrictions

For this laboratory exercise the following restrictions apply:

- Trace generation must be done in a DCS Teaching Laboratory machine, using the Ethernet (not WiFi) connection.
- Trace analysis may be done in any machine with the Wireshark software installed.

3 Instructions: Trace Generation

1. Open up the terminal.
2. Start up the Wireshark software.
3. Select the appropriate interface and begin packet capture.
4. In the terminal, type and execute the command `traceroute www.inquirer.net`. Wait for the `traceroute` program to finish executing. Leave the terminal open.
5. Stop packet capture.
6. Save the packet trace file as `labexercise3.pcapng`. It is **important** that you do not skip this step, as this tracefile will be used in a latter laboratory exercise. You can also use this tracefile if you wish to work on the laboratory report at a later time (if you are going to do this however, also save a screenshot of `traceroute`'s terminal output!).
7. At this stage, you are now ready to work on the laboratory report.

4 What to hand in

Answer the following questions in the laboratory report, based on your Wireshark experimentation:

1. How many routers are between your computer and the server hosting the Philippine Daily Inquirer? List them. Include an annotated screenshot supporting your answer (Hint: You do not need a Wireshark screenshot for this).
2. Filter (remove) all non-UDP-containing packets in the packet-listing window. Note that some packets that have nothing to do with `traceroute` may still remain. Identify the *first* packet sent by `traceroute` - you can do this searcing for the first UDP-containing packet with the TTL value within the IP Header set to *1*. Include a screenshot or image of the packet, with an annotation showing why you think it is the packet we are looking for.
3. Use the packet you identified in (2) as reference. What is the IP address of your computer? What is the IP address of the server hosting the Philippine Daily Inquirer? Include an annotated screenshot supporting your answer.

4. Use the packet you identified in (2) as reference. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes. Include an annotated screenshot supporting your answer.

5. Find the two other UDP-containing packets sent by `traceroute` with TTL = 1. Compare the values contained in the IP headers of the three packets. Which fields in the IP datagram *always* change from one datagram to the next within this set?

6. With reference again to the set in (5), which fields stay constant?

7. Describe the pattern you see in the values in the **Identification** field of the IP datagram.

8. `traceroute` sends the three packets in a (TTL) set in quick succession: that is, it does not wait for the resulting ICMP message from the first one to arrive before sending the second one, etc.. Each packet in the set will cause a router to send back an ICMP message. Does `traceroute` simply assume that the ICMP messages will arrive in order? That is, that the first ICMP message that that will arrive is caused by the first UDP packet, the second ICMP message by the second UDP packet and the third ICMP message by the third UDP packet? (Super Hint: The answer is no.) How does `traceroute` associate an ICMP message with the UDP packet which caused it? (Hint: The answer is *not* be found in the IP header or the network layer, for that matter.) Include an annotated screenshot supporting your answer.

9. Identify the ICMP TTL-exceeded message caused by the *first* UDP packet in the TTL = 3 set. Include a screenshot or image of the packet, with an annotation showing why you think it is the packet we are looking for.

10. `traceroute` reports the round trip delay for each UDP packet it sent. That is, it reports the time of time which elapsed between the sending of the UDP-containing packet and the reception of the ICMP message which it caused. Wireshark associates its own timestamps with packets ("Time" column in the packet listing window). For each UDP packet in the TTL = 3 set, compare the values reported by `traceroute` against the values that one will get if one computes the delays based on Wireshark timestamps. Show the computations, report actual numbers, and include annotated screenshots supporting your answer.

Note: If you have not done so, exit the Wireshark software.

5 Submission

The laboratory report is due on Sunday, February 26, 2016, 2359 hours. You can submit the laboratory report via UVLE.