COMP3234B Computer and Communication Networks

Lab 4: IP Trace Analysis

Introduction

In this lab, we'll investigate the IP protocol, focusing on the IP datagram. We'll analyze a trace of IP datagrams sent and received by the traceroute program. We'll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

Capture IP Trace using Wireshark

We will use the Wireshark packet sniffer which you have set up and used in Lab 3, to capture a trace of IP datagrams. We will use the traceroute program to send datagrams of different sizes towards some destination, *X*. Recall that traceroute operates by first sending multiple IP datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a number of datagrams towards the same destination with a TTL value of 2; it then sends a number of 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, the router returns an ICMP message (type 11 – TTL exceeded or expired) to the sending host. As a result of this behavior, a datagram with a 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; a datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; a datagram sent 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 destination *X* by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

We'll run traceroute and have it send datagrams of various lengths.

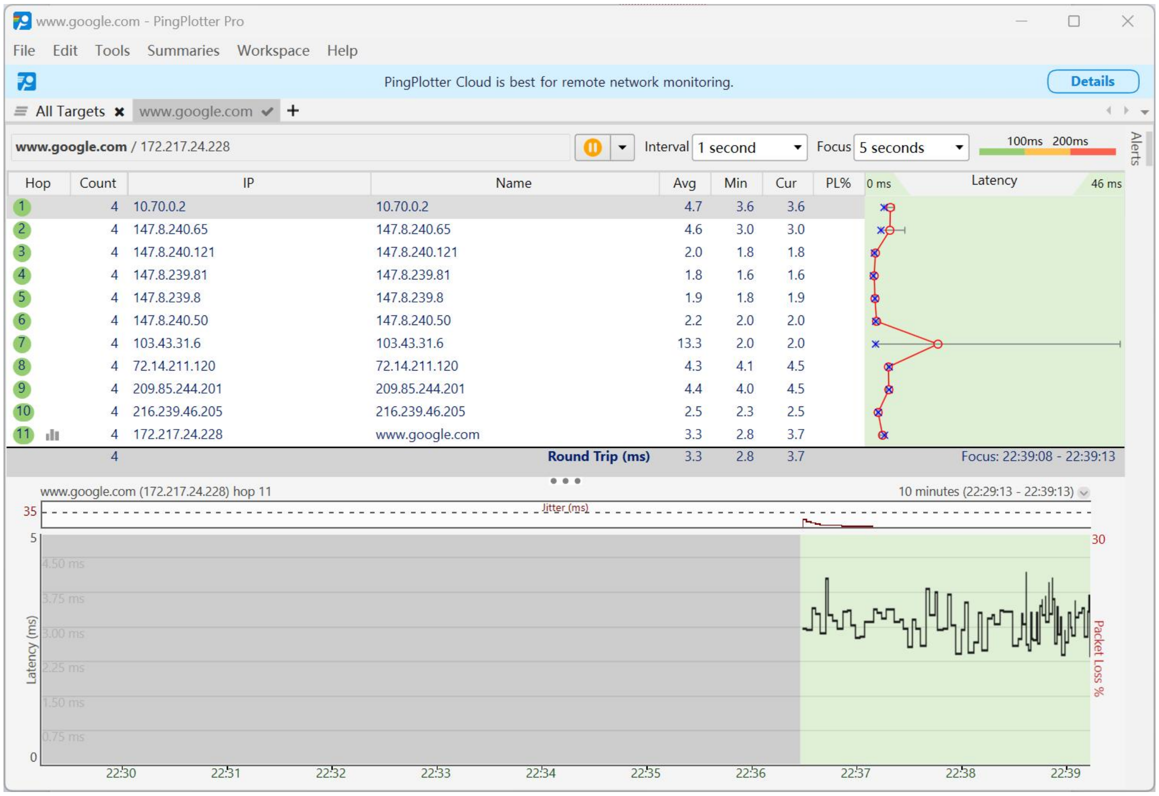
- **If you are using a Windows platform:** The tracert program provided with Windows may not allow one to change the size of the IP datagram sent by the tracert program. A nicer Windows traceroute program is *pingplotter*, available in free versions at http://www.pingplotter.com. Download and install *pingplotter*, and try it out by performing a few traceroutes to your favorite sites. The size of the IP datagrams can be explicitly set in *pingplotter* by selecting the menu item *Edit->Options->Default Settings->Engine* and then filling in the *Packet Size* field. The default packet size is 56 bytes. Once *pingplotter* has sent a series of packets with the increasing TTL values, it restarts the sending process again with a TTL of 1, after waiting *Trace Interval* amount of time. The value of *Trace Interval* can be explicitly set in *pingplotter*.

- **If you are using Linux/Unix/MacOS**: With the Linux/Unix/MacOS traceroute command, the size of the IP datagram sent towards the destination can be explicitly set by indicating the number of bytes in the packetlen option; this value is entered in the traceroute command line immediately after the name or address of the destination. For example, to send traceroute datagrams of 80 bytes towards google.com, the command would be:

```
traceroute google.com 80
```

Launch Wireshark and begin packet capture *(Capture–>Start)*. Next, depending on which operating system you are using:

- If you are using a Windows platform, launch *pingplotter* and enter the name of a destination host in the "Target name or IP" field. Select the menu item *Edit–> Options–>Default Settings–>Engine* and enter a value of 80 in the *Packet Size* field and then press OK. Then press the Trace button. You should see a *pingplotter* window that looks like the following:



www.google.com / 172.217.24.228

Interval 1 second

Focus 5 seconds

Hop	Count	IP	Name	Avg	Min	Cur	PL%
1	4	10.70.0.2	10.70.0.2	4.7	3.6	3.6	
2	4	147.8.240.65	147.8.240.65	4.6	3.0	3.0	
3	4	147.8.240.121	147.8.240.121	2.0	1.8	1.8	
4	4	147.8.239.81	147.8.239.81	1.8	1.6	1.6	
5	4	147.8.239.8	147.8.239.8	1.9	1.8	1.9	
6	4	147.8.240.50	147.8.240.50	2.2	2.0	2.0	
7	4	103.43.31.6	103.43.31.6	13.3	2.0	2.0	
8	4	72.14.211.120	72.14.211.120	4.3	4.1	4.5	
9	4	209.85.244.201	209.85.244.201	4.4	4.0	4.5	
10	4	216.239.46.205	216.239.46.205	2.5	2.3	2.5	
11	4	172.217.24.228	www.google.com	3.3	2.8	3.7	
	4		**Round Trip (ms)**	3.3	2.8	3.7	

Focus: 22:39:08 - 22:39:13

www.google.com (172.217.24.228) hop 11

10 minutes (22:29:13 - 22:39:13)

Next, send a set of datagrams with a longer length, by selecting *Edit-> Options->Default Settings->Engine* and enter a value of 4000 in the *Packet Size* field and then press OK. Then press the Resume button.

Finally, send a set of datagrams with a longer length, by selecting *Edit-> Options->Default Settings->Engine* and enter a value of 8000 in the *Packet Size* field and then press OK. Then press the Resume button.

Stop Wireshark tracing.

- If you are using a Unix/Linux or Mac platform, enter three traceroute commands, one with a length of 80 bytes, one with a length of 4000 bytes, and one with a length of 8000 bytes.

  Stop Wireshark tracing.

Lab Exercises

In your trace, you should be able to see the series of ICMP Echo Requests (in the case of Windows machine, as pingplotter sends out ICMP Echo Request messages in IP datagrams when running traceroute) or the UDP segments (in the case of Unix/Linux/MacOS, which sends UDP segments in IP datagrams when running traceroute) sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. In the questions below, we'll assume you are using a Windows machine; the corresponding questions for the case of a Unix/Linux/MacOS machine should be clear (e.g., replace "ICMP Echo Request message" in the questions by "UDP segment" if you are using a Unix/Linux/MacOS machine). When answering a question, you should add a screenshot or a printout of the packet(s) within the trace that you use to answer the question.

Select the second ICMP Echo Request message sent by your computer. In the case of Unix/Linux/MacOS, for each TTL, the traceroute command may send a group of messages instead of one message to ensure reliability and accuracy. As a result, here 'the second' means the first message in **the second group.** In other words, we want you to select the message for TTL=2. Expand the Internet Protocol part of the packet in the packet details window.

1. What is the IP address of your computer?

2. Within the IP datagram header, what is the value in the upper layer protocol field?

3. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determine the number of payload bytes.

4. Has this IP datagram been fragmented? Explain how you determine whether or not the datagram has been fragmented.

Next, sort the traced packets according to IP source address by clicking on the *Source* column's header in the WireShark window; a small upward pointing arrow should appear next to the word *Source*. If the arrow points down, click on the *Source* column header again. Select the first ICMP Echo Request message sent by your computer to the destination that you specified when running the traceroute program, and expand the Internet Protocol portion in the packet details window. In the listing of the captured packets window, you should see all the subsequent ICMP Echo Request messages (possibly with additional interspersed packets sent by other protocols running on your computer) below this first ICMP Echo Request message. Use the down arrow to move through the ICMP Echo Request messages sent by your computer.

5. Which field(s) in the IP datagram (carrying such an ICMP Echo Request message) *always* change from one datagram to the next within this series of ICMP Echo Request messages sent by your computer?

6. Which fields remain constant? Why?

Next, find the series of ICMP TTL-exceeded replies sent to your computer by the first hop router.

7. Explain how you identify the ICMP TTL-exceeded messages sent to your computer by the first hop router.

Now we investigate datagram fragmentation. Make sure you have sorted the packet listing according to time (you can do so by clicking on the *Time* column's header in the WireShark window).

8. Find the second ICMP Echo Request message (similarly, here 'the second' means the message for TTL=2) that was sent by your computer after you changed the *Packet Size* in *pingplotter* to 4000. Has that message been fragmented across more than one IP datagram?

9. Print out the first fragment of the fragmented IP datagram of the second ICMP Echo Request message. What information in the IP header indicates that this datagram is a fragment? What information in the IP header indicates whether this is the first fragment versus a latter fragment? What is the total length of this IP datagram?

10. Print out the second fragment of the fragmented IP datagram of the second ICMP Echo Request message. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

11. What fields change in the IP header between the first and second fragments?

Now find the second ICMP Echo Request message (similarly, here 'the second' means the message for TTL=2) that was sent by your computer after you changed the *Packet Size* in *pingplotter* to 8000.

12. How many fragments were created from the original datagram?

13. What fields change in the IP header among the fragments?

Submission:

Please put all your answers to the above questions in a word document and insert screenshots of your Wireshark window (or printout of packet information) where needed. Please convert the word document to a ***lab4-yourstudentid.pdf*** file and submit the PDF file on Moodle before 23:59 Wednesday March 27, 2024:

(1) Login Moodle.
(2) Find "Labs" in the left column and click "Lab 4".
(3) Click "Add submission", browse your .pdf file and save it. Done.
(4) You will receive an automatic confirmation email, if the submission was successful.
(5) You can "Edit submission" to your already submitted file, but ONLY before the deadline.