

Computer Networking and Network Programming (ECE 5650)

Lab 5

Instructions:

- **Original Work:** this assignment must represent the original work of the team members.
 - You must not look at other solutions or show your solutions to anyone else.
 - You must not collaborate or discuss the assignment with other groups or individuals.
 - You must not get any portion of the code from Internet sources, other current or prior students, or other groups or individuals.
 - You must not ask or pay others to help you with the solution.

Policy on Cheating, Fabrication, and Plagiarism: Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Therefore, avoid all appearance of improper behavior! Students who witness cheating should report the incident to the instructor as soon as possible. **Academic misconduct will result in at least failing the course.**

- **Late Submissions:** no late submissions will be accepted.
- **Teamwork:** You must work with **one other student** and **must submit only one copy with both your names.**
- **Have questions?** Please contact the GTA/Grader. You can ask him by e-mail or send an e-mail to schedule a meeting on Microsoft Teams. **No help will be given on the day the lab is due.**

Objectives:

In this lab, we'll investigate the IP protocol, focusing on the IP datagram. We'll do so by analyzing a trace of IP datagrams sent and received by an execution of the traceroute program (the traceroute program itself is explored in more detail in the Wireshark ICMP

lab). We'll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

Prerequisites:

- Please review the class notes on IP.
- Other suggested reading material includes RFC 2151 [<http://ftp.rfc-editor.org/in-notes/rfc2151.txt>] (especially Section 3.4) as well as RFC 791 [<http://ftp.rfc-editor.org/in-notes/rfc791.txt>].

Submission Requirements:

- Submit a professional report in **pdf format** including your name(s) and your answers to all questions in the lab. **Only one file must be submitted with all screenshots included within.**
- The answers must be numbered in a consistent way to the questions in this assignment.

Lab Assignment

1. Capturing packets from an execution of traceroute

In order to generate a trace of IP datagrams for this lab, we'll use the traceroute program to send datagrams of different sizes towards some destination, *X*. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series 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 (actually, RFC 791 says that the router must decrement the TTL by *at least* one). If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) 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; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the 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.

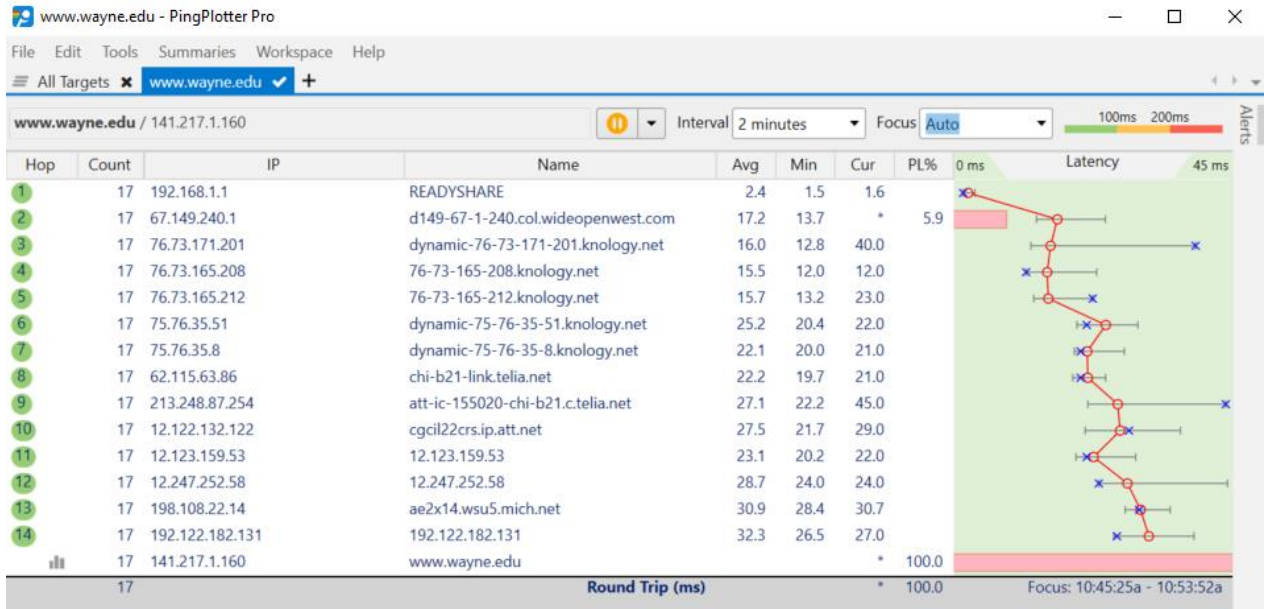
We'll want to run traceroute and have it send datagrams of various lengths.

- **Windows.** The `tracert` program (used for our ICMP Wireshark lab) provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the `tracert` program. A nicer Windows traceroute program is *pingplotter*, available both in free version and shareware versions at <http://www.pingplotter.com>. Download and install *pingplotter*, and test it out by performing a few traceroutes to your favorite sites. The size of the ICMP echo request message 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* and the number of intervals can be explicitly set in *pingplotter*.
- **Linux/Unix/MacOS.** With the Unix/MacOS traceroute command, the size of the UDP datagram sent towards the destination can be explicitly set by indicating the number of bytes in the datagram; 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 2000 bytes towards `gaia.cs.umass.edu`, the command would be

```
%tracert gaia.cs.umass.edu 2000
```

Do the following:

- Start up Wireshark and begin packet capture (*Capture->Start*) and then press *OK* on the Wireshark Packet Capture Options screen (we'll not need to select any options here).
- If you are using a Windows platform, startup *pingplotter* and enter the name of a target destination in the "Enter Target name or IP." If supported by your downloaded version, limit "# of times to Trace" field to 3, so you don't gather too much data. Otherwise change the time "Interval" to an appropriate value and stop *pingplotter* manually after three full traces are collected. Select the menu item *Edit-> Options->Default Settings->Engine* and enter a value of 56 in the *Packet Size* field and then press *OK*. Then press the *Trace* button. You should see a *pingplotter* window that looks something like this:



Next, send a set of datagrams with a longer length, by selecting *Edit-> Options->Default Settings->Engine* and enter a value of 2000 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 3500 in the *Packet Size* field and then press OK. Then press the Resume button.

Stop Wireshark tracing.

- If you are using a Unix or Mac platform, enter three traceroute commands, one with a length of 56 bytes, one with a length of 2000 bytes, and one with a length of 3500 bytes.

Stop Wireshark tracing.

Optional:

Consider the following packet trace file that was captured while following the steps above:

- Download the zip file <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip> and extract the file *ipethereal-trace-1*.
- Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the *ipethereal-trace-1* trace file.

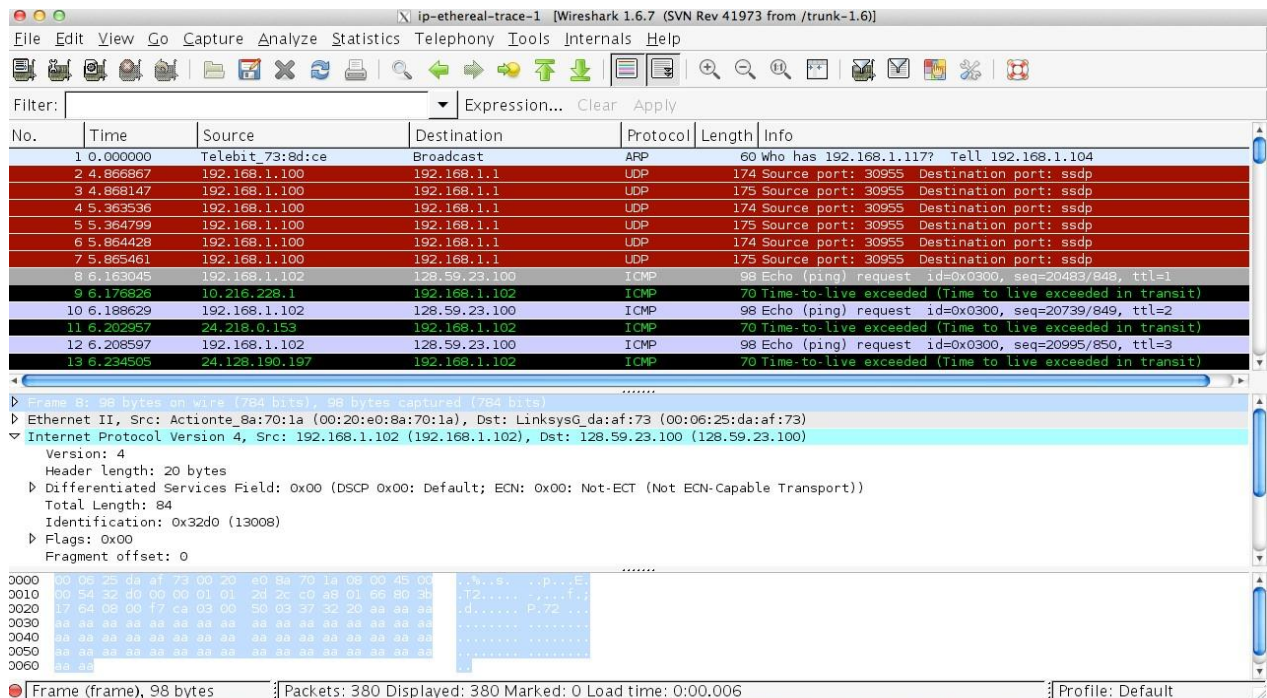
You may well find it valuable to download this trace even if you've captured your own trace.

Answer all questions in this lab based on your own trace.

2. A look at the captured trace

In your trace, you should be able to see the series of ICMP Echo Request (in the case of Windows machine) or the UDP segment (in the case of Unix) 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 machine should be clear. Whenever possible, when answering a question below you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. When you hand in your assignment, annotate the output so that it's clear where in the output you're getting the information for your answer (e.g., for our classes, we ask that students markup paper copies with a pen, or annotate electronic copies with text in a colored font). To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

1. Select the first ICMP Echo Request message sent by your computer and expand the Internet Protocol part of the packet in the packet details window.



- What is the IP address of your computer?
2. Within the IP packet 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 determined the number of payload bytes.

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

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

5. Which fields in the IP datagram *always* change from one datagram to the next within this series of ICMP messages sent by your computer?
6. Which fields stay constant? Which of the fields *must* stay constant? Which fields must change? Why?
7. Describe the pattern you see in the values in the Identification field of the IP datagram

Next (with the packets still sorted by source address) find the series of ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router.

8. What is the value in the Identification field and the TTL field?
9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

Fragmentation

Sort the packet listing according to time again by clicking on the *Time* column.

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* in *pingplotter* to be 2000. Has that message been fragmented across more than one IP datagram? [Note: if you find your packet has not been fragmented, you should download the zip file <http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip> and extract the *ipethereal-trace-1* packet trace. If your computer has an Ethernet interface, a packet size of 2000 *should* cause fragmentation.¹]

¹ The packets in the *ip-ethereal-trace-1* trace file in <http://gaia.cs.umass.edu/wireshark-labs/wiresharktraces.zip> are all less than 1500 bytes. This is because the computer on which the trace was gathered has an Ethernet card that limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP/IP header data and 1460 bytes of upper-layer protocol payload). This 1500 byte value is the standard maximum length allowed by Ethernet. If your trace indicates a datagram longer than 1500 bytes, and your computer is using an Ethernet connection, then Wireshark is reporting the wrong IP datagram length; it will likely also show only one large IP datagram rather than multiple smaller datagrams. This inconsistency in

11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?
12. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?
13. What fields change in the IP header between the first and second fragment?

Now find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* in *pingplotter* to be 3500.

14. How many fragments were created from the original datagram?
15. What fields change in the IP header among the fragments?

reported lengths is due to the interaction between the Ethernet driver and the Wireshark software. We recommend that if you have this inconsistency, that you perform this lab using the *ip-ethereal-trace-1* trace file.