Wireshark Lab: IP

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 send and received by an execution of the traceroute program, which you have used in last lab activity on ICMP. We'll investigate the various fields in the IP datagram and study IP fragmentation in detail.

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. We will be running traceroute to www.upm.edu.ph. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) fields 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 in the datagrams containing the ICMP TTL-exceeded messages.

We'll want to run traceroute and have it send datagrams of various lengths. In Linux, 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 www.upm.edu.ph, the command would be:

```
traceroute www.upm.edu.ph 2000
```

Do the following:

- Start up Wireshark and begin packet capture.
- Send a usual traceroute to any server (e.g. www.upm.edu.ph)
- Send a traceroute with datagrams of longer length (e.g. 2000) to the same server.
- Send a traceroute with datagrams of even longer length (e.g. 3500) to the same server.
- Stop Wireshark packet capture.

If you are using Windows, although tracert is the counterpart of traceroute, you can't use tracert as it does not allow you to change the size of the ICMP echo request. You should use pingplotter. You can download pingplotter from https://www.pingplotter.com/. You should use the 14-day demo version of pingplotter as the free version doesn't allow you to change the size of the ICMP echo request. The size of the ICMP echo request message can be set in pingplotter by selecting the menu item Edit → Options → Packet and then change the value in Packet size (in bytes) field. The default packet size is 56 bytes. Once pingplotter has sent a series of packets with increasing TTL values, it restarts the sending process again

with a TTL of 1, after awaiting *Trace Interval* amount of time. The value of *Trace Interval* and the number of intervals can be set in pingplotter.

Note: It's possible that pingplotter's newer version might have change the function to change the size of the ICMP echo request message. It's up to you to determine how to change it.

A look at the captured packets

In your trace, you should be able to see the series of UDP segment sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. If you are using pingplotter, you should see the series of ICMP echo request instead. Whenever applicable, when answering a question below you should include a screenshot of packet(s) with the appropriate fields highlighted.

1. Select the first UDP segment sent by your computer that is part of the traceroute, 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 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 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 UDP message sent by your computer that is part of the traceroute, and expand the Internet Protocol portion in the "details of the selected packet header" window. In the "listing of captured packets" window, you should see all of the subsequent messages (perhaps with interspersed packets sent by other protocols running on your computer) below this first message. Use the down arrow to move through the UDP messages that is part of the traceroute 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 fields of 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 UDP segment that is part of the traceroute sent by your computer after you changed the packet size to 2000. Has that message been fragmented across more than one IP datagram?
11. Show the first fragment of the fragmented IP datagram. 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? How long is this IP datagram?
12. Show the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?
13. What fields change in the IP header between the first and second fragment?

Now find the first UDP segment that is part of traceroute sent by your computer after you changed the packet size to 3500.

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