Lab 6: IP

Hide Assignment Information

Instructions

Lab 6: IP

Instructions:

- Support all answers by a screenshot of your Wireshark and Command Prompt windows. Annotate screenshots to explain your answer.
- Submissions must be through Brightspace.
- There is a 24-hour grace period after the due date without a penalty. Late submissions will not be accepted.

Note: This lab is mostly adapted from materials provided by the authors of *Computer Networking: A Top-Down Approach*. All rights reserved.

Introduction

In this lab, we will investigate the IP protocol, focusing on the IPv4 and IPv6 datagram. This lab has two parts. In the first part, we will analyze packets in a trace of detail in the ICMP lab. We will study IP fragmentation in Part 2 of this lab.

Capturing Packets from an Execution of Traceroute

In order to generate a trace of IPv4 datagrams for the first two parts of this lab, we will use the Traceroute program to send datagrams of two different sizes to ga

- `traceroute` command 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 c series of datagrams towards the same destination with a TTL value of 3; and so on.
- 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 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 a IC
uter 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

Cancel

`traceroute` can learn the IP addresses of the routers between itself and the destination by looking at the source IP addresses in the datagrams containing the ICI

With the Linux/MacOS `traceroute` command, the size of the UDP datagram sent towards the final destination can be explicitly set by indicating the number of b after the name or address of the destination. For example, to send `traceroute` datagrams of 2000 bytes towards gaia.cs.umass.edu, the command is:

```
>traceroute gaia.cs.umass.edu 2000
```

The `tracert` program provided with Windows does not allow one to change the size of the ICMP message sent by `tracert`. So, it will not be possible to use a W fragmentation. However, you can use `tracert` to generate small, fixed length packets to perform Part 1 of this lab. At the command prompt enter:

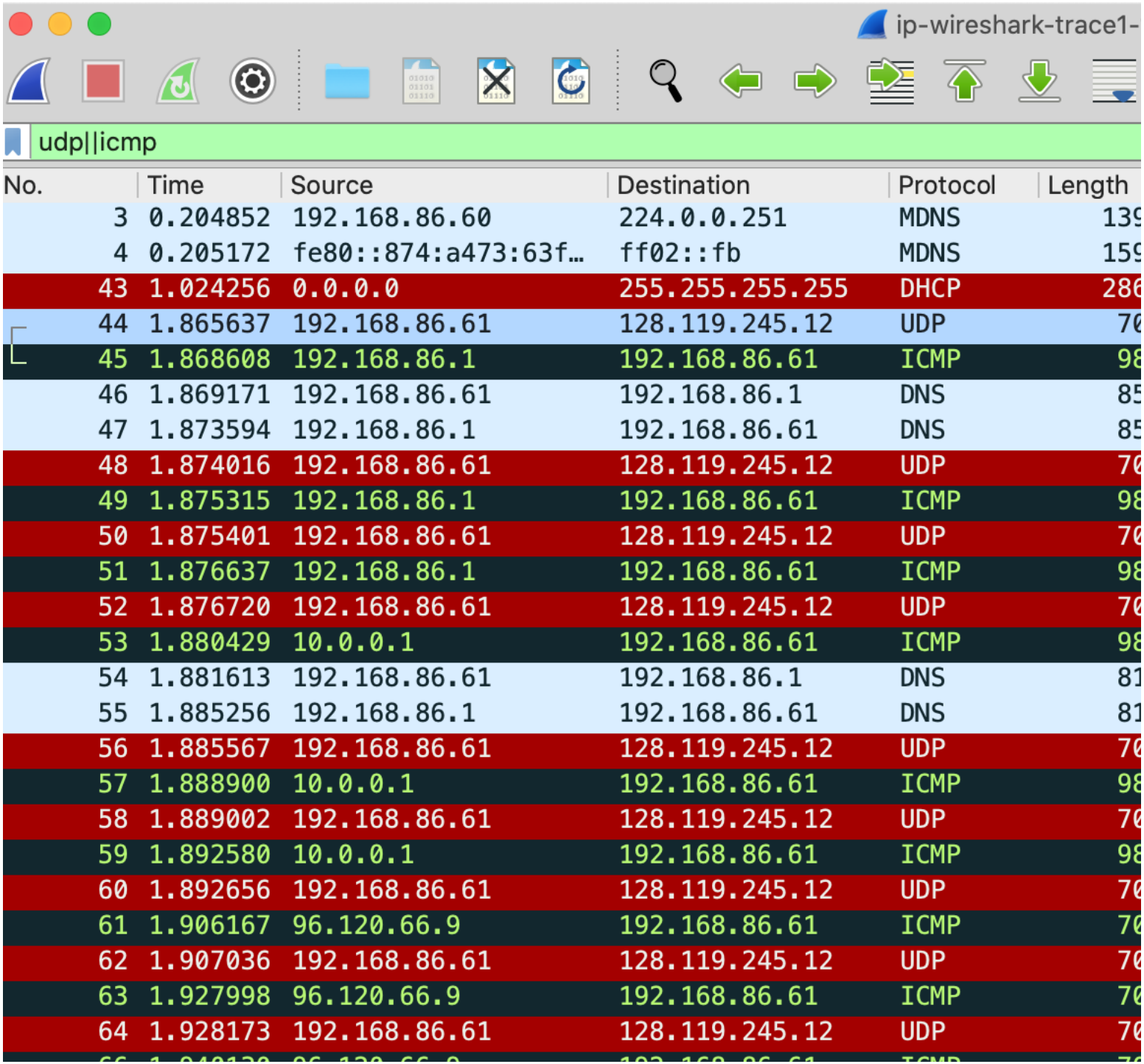
```
>tracert gaia.cs.umass.edu
```

Do the following:

- Start up Wireshark and begin packet capture.
- Windows: Enter two `tracert` commands, using a destination host of your choice.
- MacOS/Linux: Enter two `traceroute` commands, using a destination host of your choice, the first with a length of 56 bytes. Once that command has finish length of 3000 bytes.
- When the Traceroute program terminates, stop packet capture in Wireshark.

Basic IPv4

In your trace, you should be able to see a series of ICMP Echo Request messages (Windows) or UDP segments (MacOS/Linux) sent by Traceroute on your compute routers. Your screen should look similar to the screenshot in Figure 1, where we have used the display filter *icmp* (Windows) or *udp||icmp* (MacOS) so that only UD



No.	Time	Source	Destination	Protocol	Length
3	0.204852	192.168.86.60	224.0.0.251	MDNS	139
4	0.205172	fe80::874:a473:63f…	ff02::fb	MDNS	159
43	1.024256	0.0.0.0	255.255.255.255	DHCP	286
44	1.865637	192.168.86.61	128.119.245.12	UDP	70
45	1.868608	192.168.86.1	192.168.86.61	ICMP	98
46	1.869171	192.168.86.61	192.168.86.1	DNS	85
47	1.873594	192.168.86.1	192.168.86.61	DNS	85
48	1.874016	192.168.86.61	128.119.245.12	UDP	70
49	1.875315	192.168.86.1	192.168.86.61	ICMP	98
50	1.875401	192.168.86.61	128.119.245.12	UDP	70
51	1.876637	192.168.86.1	192.168.86.61	ICMP	98
52	1.876720	192.168.86.61	128.119.245.12	UDP	70
53	1.880429	10.0.0.1	192.168.86.61	ICMP	98
54	1.881613	192.168.86.61	192.168.86.1	DNS	81
55	1.885256	192.168.86.1	192.168.86.61	DNS	81
56	1.885567	192.168.86.61	128.119.245.12	UDP	70
57	1.888900	10.0.0.1	192.168.86.61	ICMP	98
58	1.889002	192.168.86.61	128.119.245.12	UDP	70
59	1.892580	10.0.0.1	192.168.86.61	ICMP	98
60	1.892656	192.168.86.61	128.119.245.12	UDP	70
61	1.906167	96.120.66.9	192.168.86.61	ICMP	70
62	1.907036	192.168.86.61	128.119.245.12	UDP	70
63	1.927998	96.120.66.9	192.168.86.61	ICMP	70
64	1.928173	192.168.86.61	128.119.245.12	UDP	70

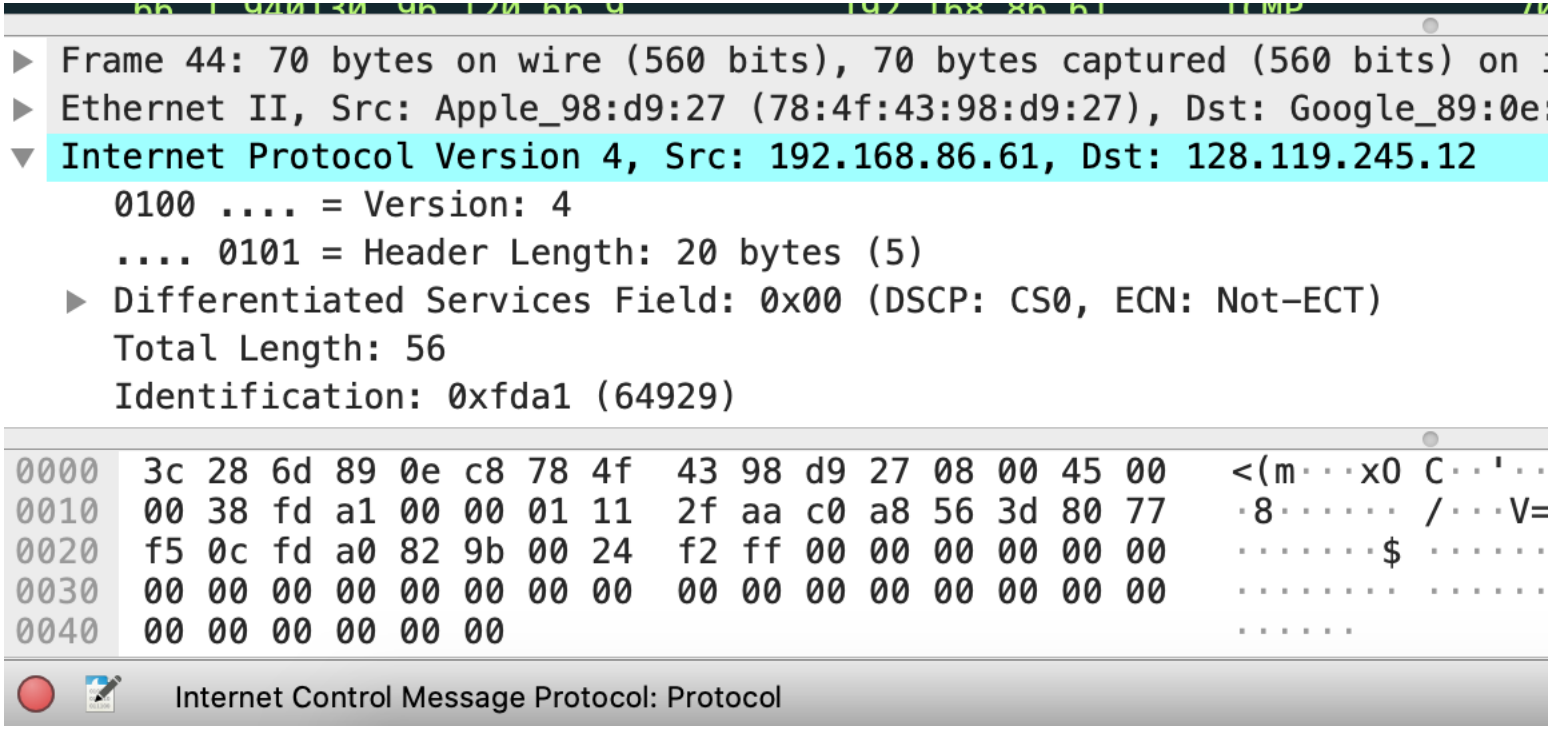


Figure 1: Wireshark screenshot, showing UDP and ICMP packets on a MacOS computer

Questions

Now answer the following questions:

1. Select the first ICMP Echo Request messages (Windows) or UDP segments (MacOS/Linux) sent by your computer via the Traceroute program. Expand the I your computer?
2. What is the value in the time-to-live (TTL) field in this IPv4 datagram's header?
3. What is the value in the upper layer protocol field in this IPv4 datagram's header? [Note: the answers for Linux/MacOS differ from Windows].
4. How many bytes are in the IP header?
5. How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.
6. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Next, let's look at the sequence of the ICMP Echo Request messages (Windows) or UDP segments (MacOS/Linux) being sent from your computer via Traceroute.

7. Which fields in the IP datagram *always* change from one datagram to the next within this series segments sent by your computer via Traceroute? Why?

8. Which fields in this sequence of IP datagrams stay constant? Why?

9. Describe the pattern you see in the values in the Identification field of the IP datagrams being sent by your computer.

Now let's take a look at the ICMP packets being returned to your computer by the intervening routers where the TTL value was decremented to zero (and hence c

10. What is the upper layer protocol specified in the IP datagrams returned from the routers? [Note: the answers for Linux/MacOS differ from Windows].

11. Are the values in the Identification fields (across the sequence of all of ICMP packets from all of the routers) similar in behavior to your answer to Question

12. Are the values of the TTL fields similar, across all of ICMP packets from all of the routers?

Fragmentation

In this section, we will look at a large (3000-byte) UDP segment sent by the `traceroute` program that is fragmented into multiple IP datagrams. For this part, you 1.pcapng. Once you have downloaded a trace file, you can load it into Wireshark and view the trace using the File pull down menu, choosing Open, and then selec

Questions

Now answer the following questions:

Note: clear any display filters and make sure packets are sorted according to time.

13. Find the first IP datagram containing the first part of the segment sent to 128.119.245.12 sent by your computer via the `traceroute` command to gaia.cs.u This is packet 179 in the trace file. Packets 179, 180, and 181 are three IP datagrams created by fragmenting the first single 3000-byte UDP segment sent datagram? (Hint: the answer is yes!)

14. What information in the IP header indicates that this datagram been fragmented?

15. What information in the IP header for this packet indicates whether this is the first fragment versus a latter fragment?

16. How many bytes are there in is this IP datagram (header plus payload)?

17. Now inspect the datagram containing the second fragment of the fragmented UDP segment. What information in the IP header indicates that this is *not* th

18. What fields change in the IP header between the first and second fragment?

19. Now find the IP datagram containing the third fragment of the original UDP segment. What information in the IP header indicates that this is the last fragm

Due on Nov 7, 2023 11:59 PM

Available on Oct 30, 2023 12:01 AM. **Access restricted before availability starts.**

Available until Nov 8, 2023 11:59 PM. **Submission restricted after availability ends.**

Submit Assignment

Submission is restricted outside of availability dates.