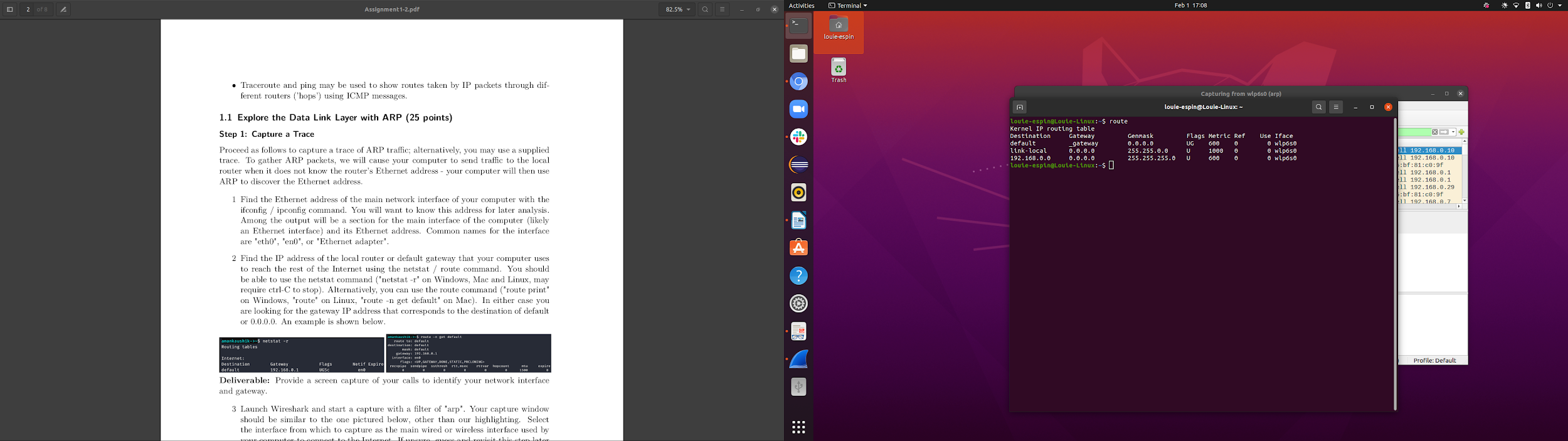
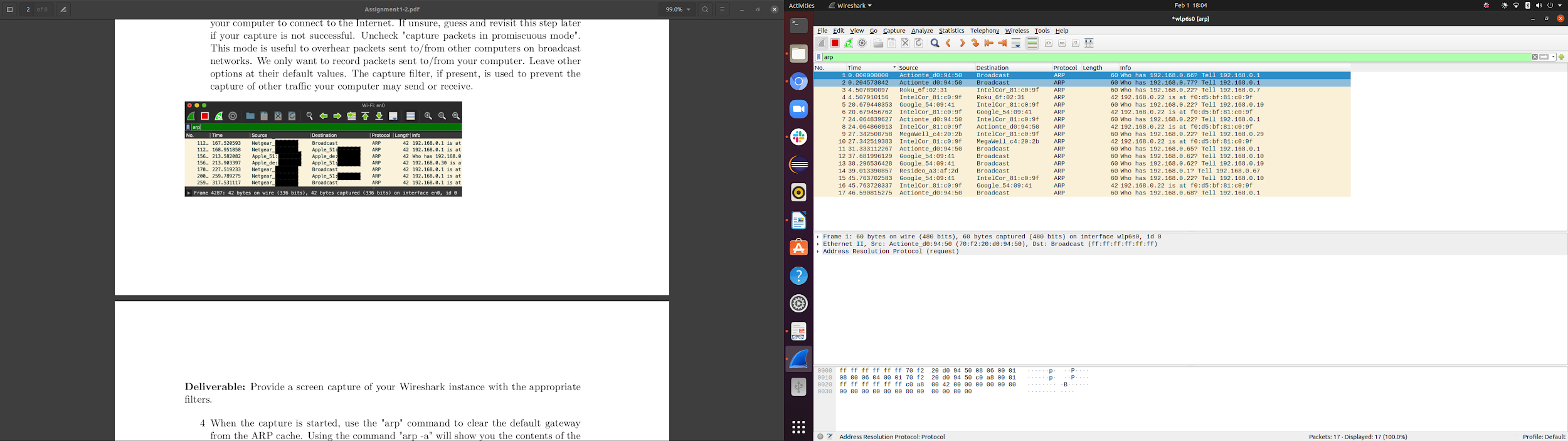
**1.1 Explore the Data Link Layer with ARP**

*1. Capture a Trace*

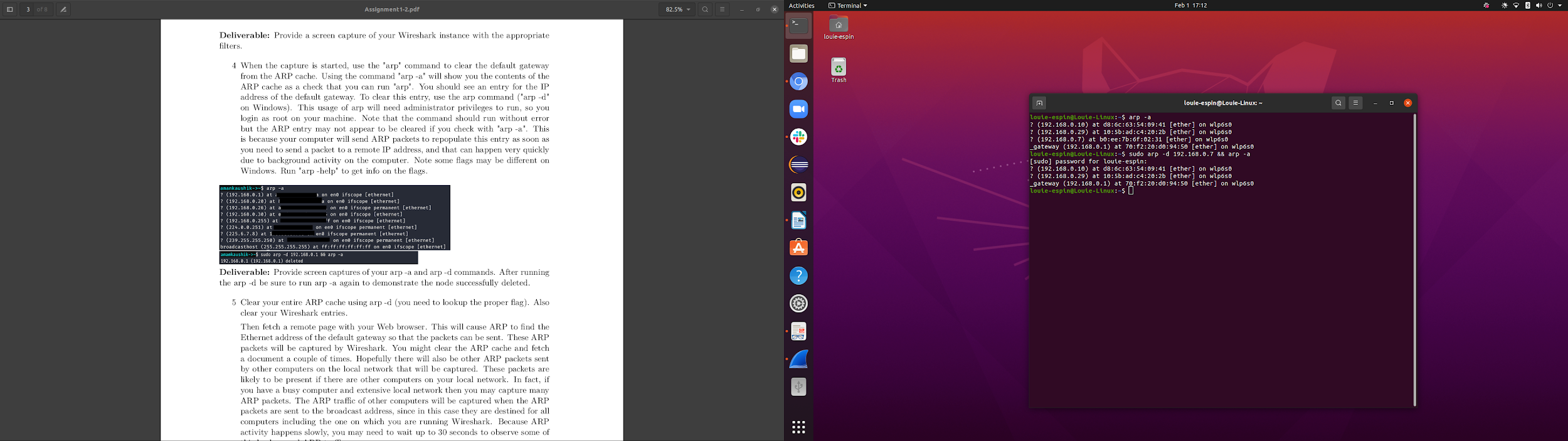
**Steps 1 and 2:** ifconfig and netstat



**Step 3:** Launch Wireshark with arp filter



**Step 4:** arp -a and arp -d

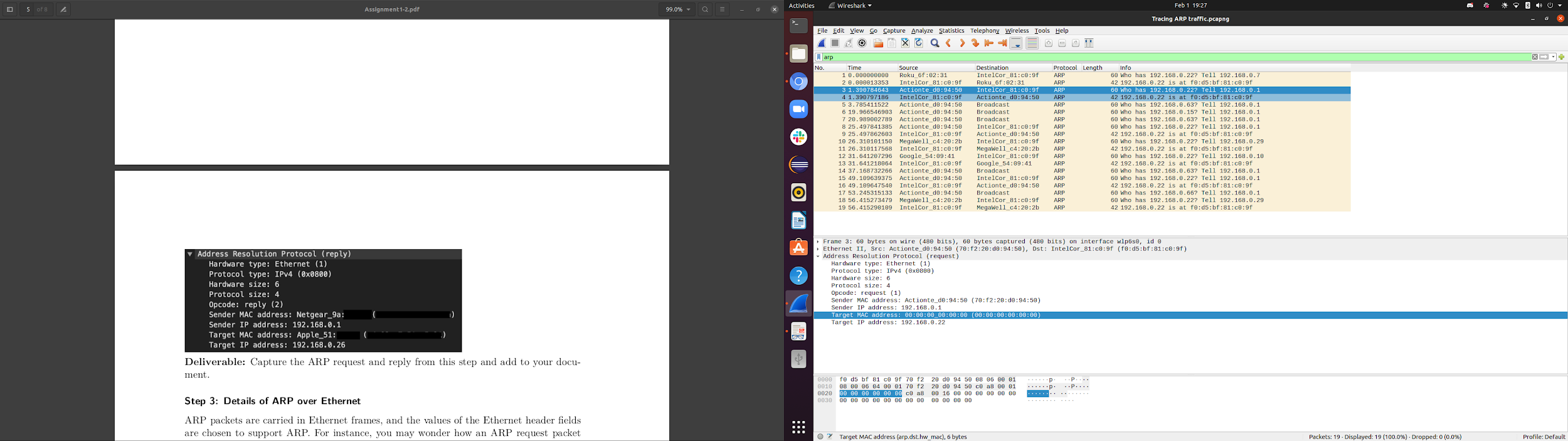


**Steps 5 and 6:**

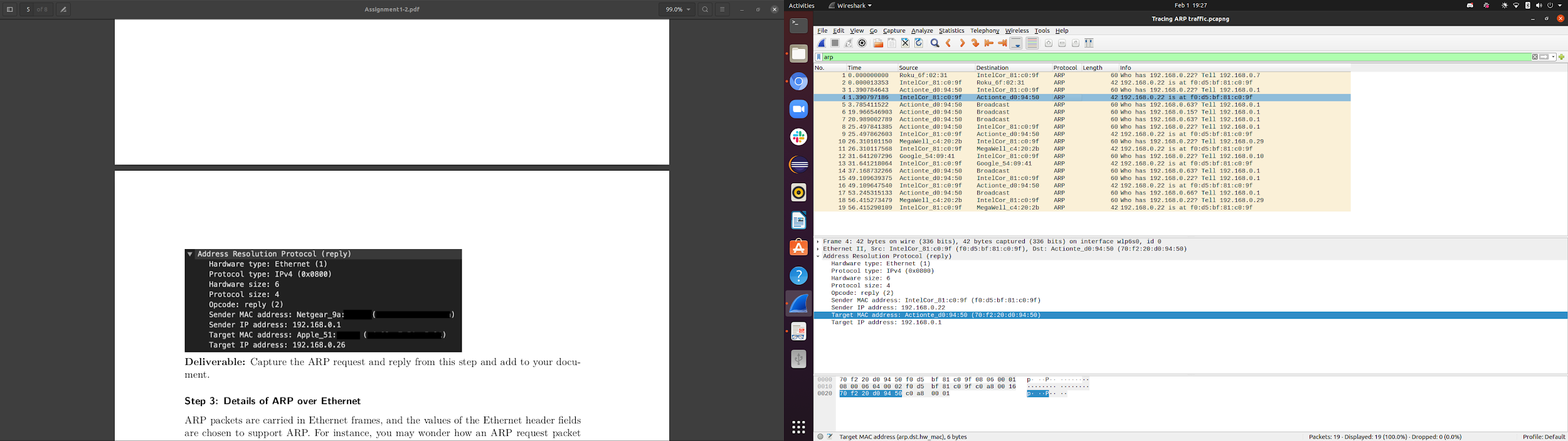
Wireshark Trace of ARP traffic uploaded to GitHub repository on folder “1.1 - Explore the Data Link Layer with ARP”

*2. Inspect a Trace*

**Step 1:** The request



**Step 2:** The reply



*3. Details of ARP over Ethernet*

**1. What opcode is used to indicate a request? What about a reply?**

The opcode for a request is 1, and 2 for a reply

**2. How large is the ARP header for a request? What about for a reply?**

They are both 28 bytes

**3. What value is carried on a request for the unknown target MAC address?**

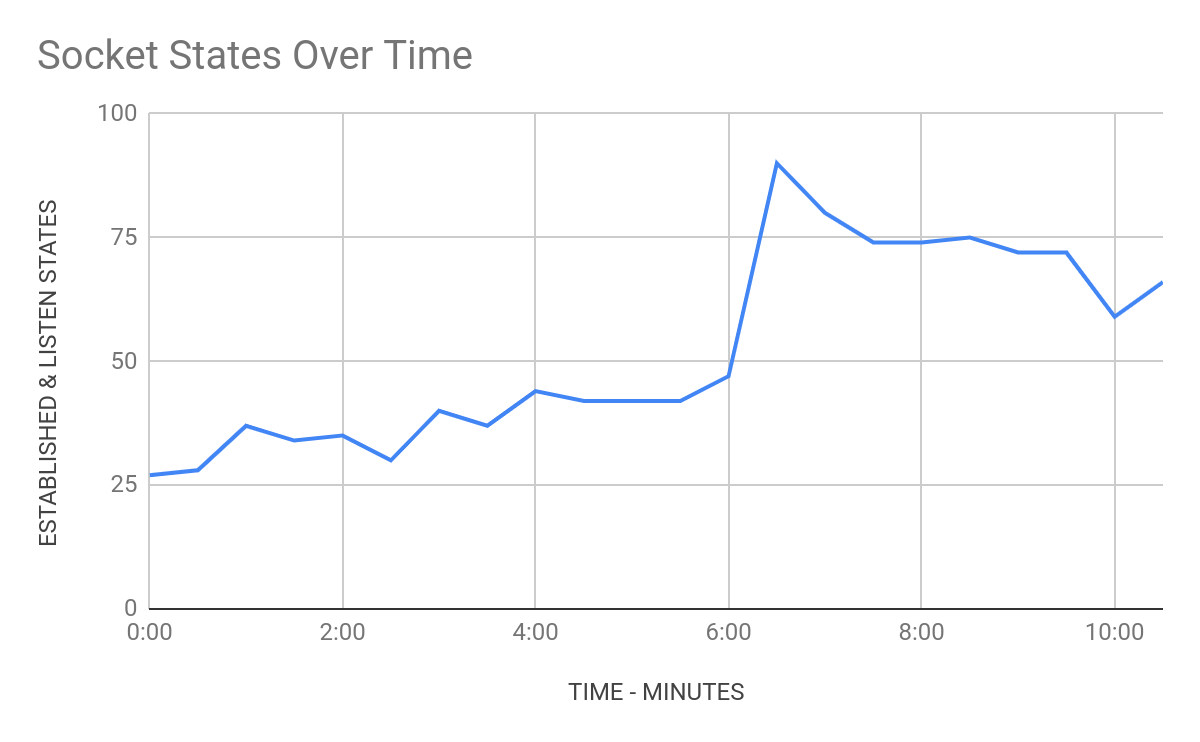
It is denoted as (00:00:00:00:00:00) since it is not known

**4. What Ethernet Type value indicates that ARP is the higher layer protocol?**

Type: ARP (0x0806)

**1.2 Understanding TCP network sockets**

**Command:** watch -n 30 "netstat -at | grep 'ESTABLISHED\|LISTEN' | tee -a TCPlog.txt"

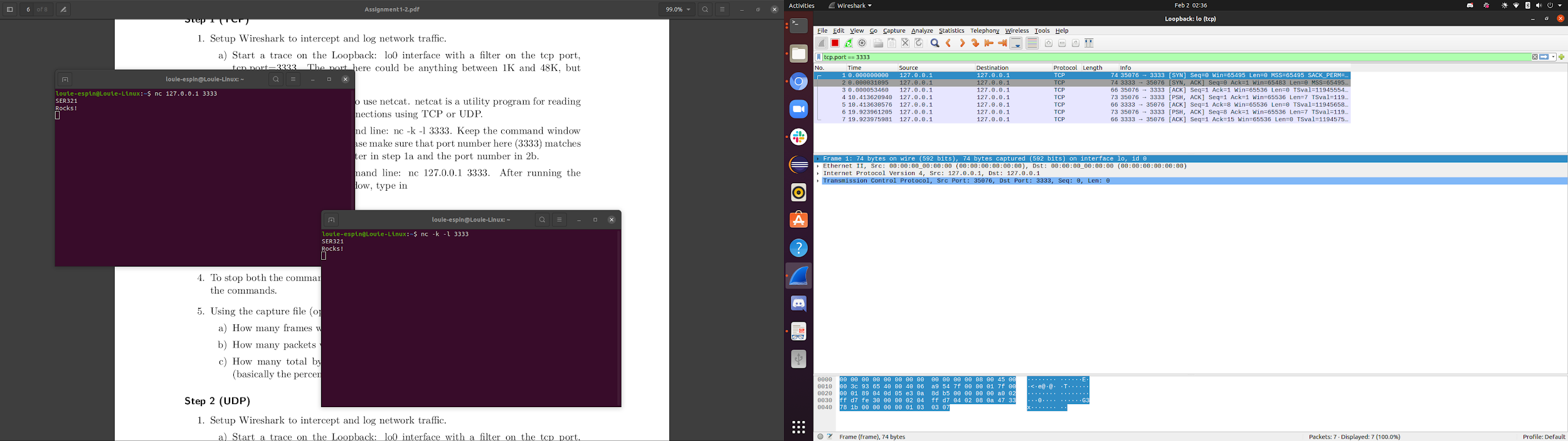


**1.3 Sniffing TCP/UDP traffic**

TCP: Wireshark capture

Wireshark capture added to github repository.

TCP: netcat commands



TCP: Questions

**1. How many frames were needed to capture those 2 lines?**

Seven frames

**2. How many packets were needed to capture those 2 lines?**

While seven packets were captured in the connection, the lines “SER321 Rocks!” showed up in two packets. “SER321” was seen in packet 4, and “Rocks!” in packet 6.

**3. How many total bytes went over the wire? How much overhead was there**

**(basically the percentage of traffic that was not needed to send SER321 Rocks!)?**

There were seven total packets in the TCP connection, but only 2 packets actually transferred information. There were 492 bytes in total of packets captured, while the packets that actually carried information amounted to 146 bytes. Meaning 5 packets that carried no data to transfer equalled 346 bytes. For comparison, “SER321 Rocks!” is 14 bytes in total, equivalent to only 3% of the bytes being transferred.

UDP: Wireshark Capture

Wireshark capture added to github repository.

UDP: netcat commands



UDP: Questions

**1. How many frames were needed to capture those 2 lines?**

The first line is shown to be on frame 9, the second line is shown on frame 14.

**2. How many packets were needed to capture those 2 lines?**

Two packets.

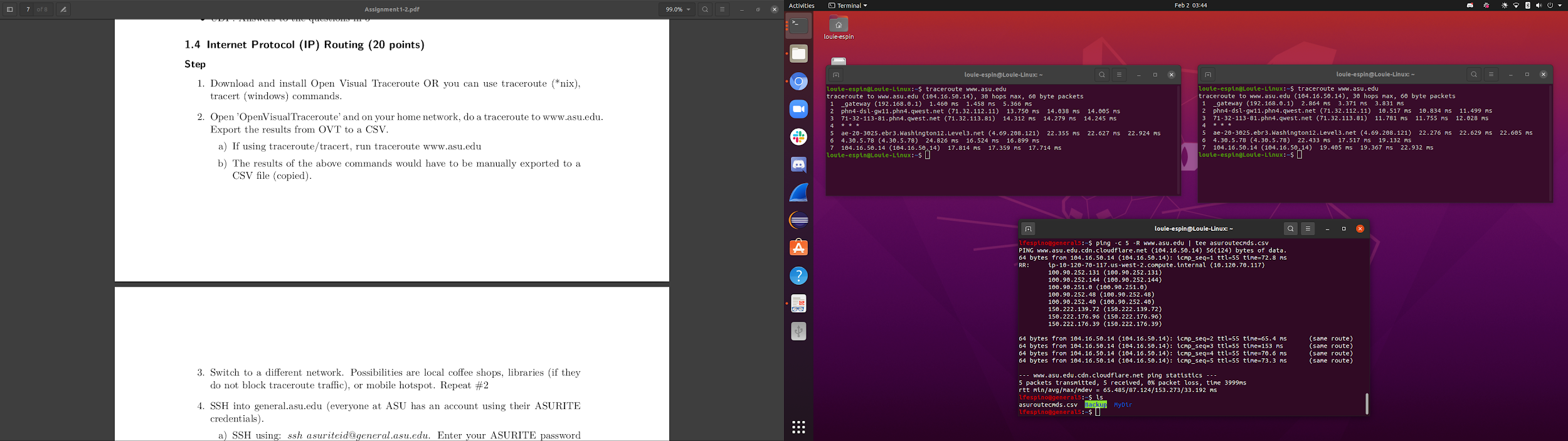
**3. How many total bytes went over the wire? How much overhead was there (percent of bytes not in the above 2 lines)?**

There were 2 packets in the UDP connection, amounting to 98 bytes. If “SER321 Rocks!” is equal to 14 bytes, then it is equivalent to around 14% of the total bytes being transferred.

**4. What is the difference in relative overhead between UDP and TCP and why? Specifically, what kind of information was exchanged in TCP that was not exchanged in UDP? Show the relative parts of the packet traces.**

UDP has much less overhead than TCP due to the fact that it does not have flow control or error correction. TCP exchanges information like acknowledgement and sequence numbers, while UPD does not.

**1.4 Internet Protocol (IP) Routing**

****

**1. Which is the fastest?**

Route 1 seems to be the fastest

**2. Which has the fewest hops?**

Routes 1 and 2

**3. Which runs its traffic through a bridge?**

Route SSH general.asu.edu