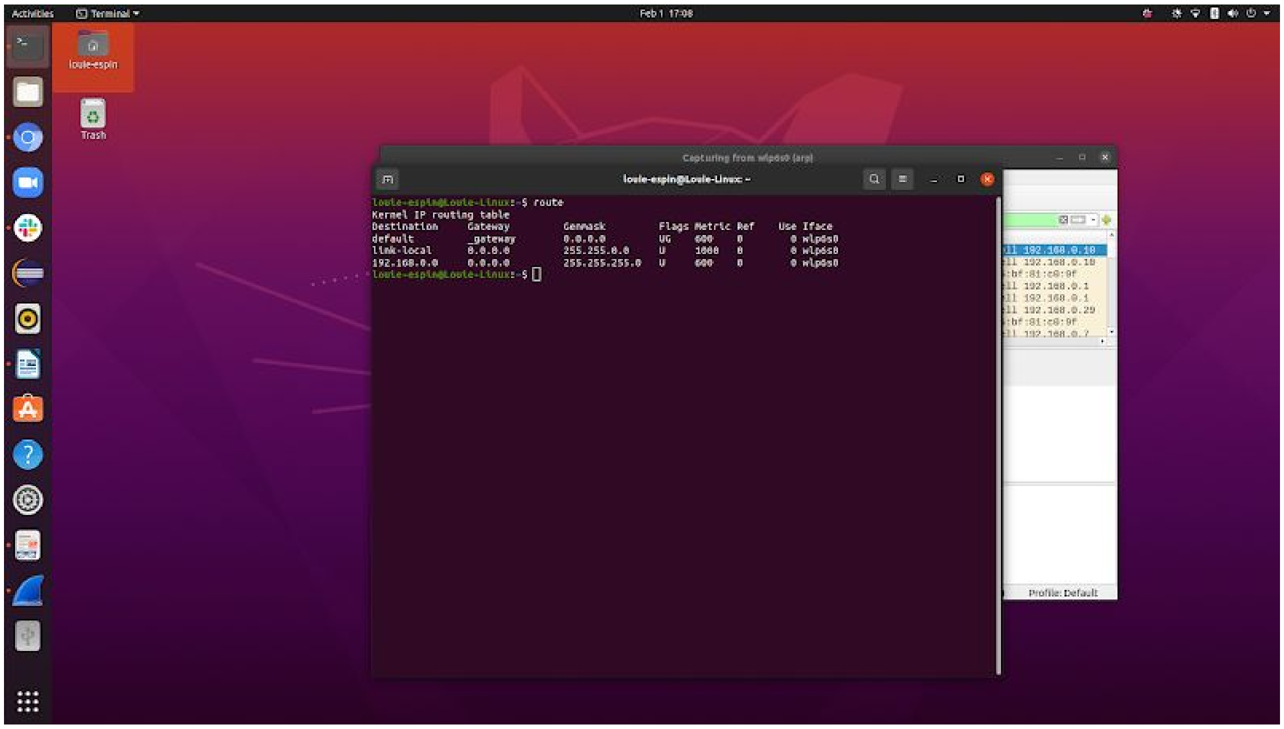
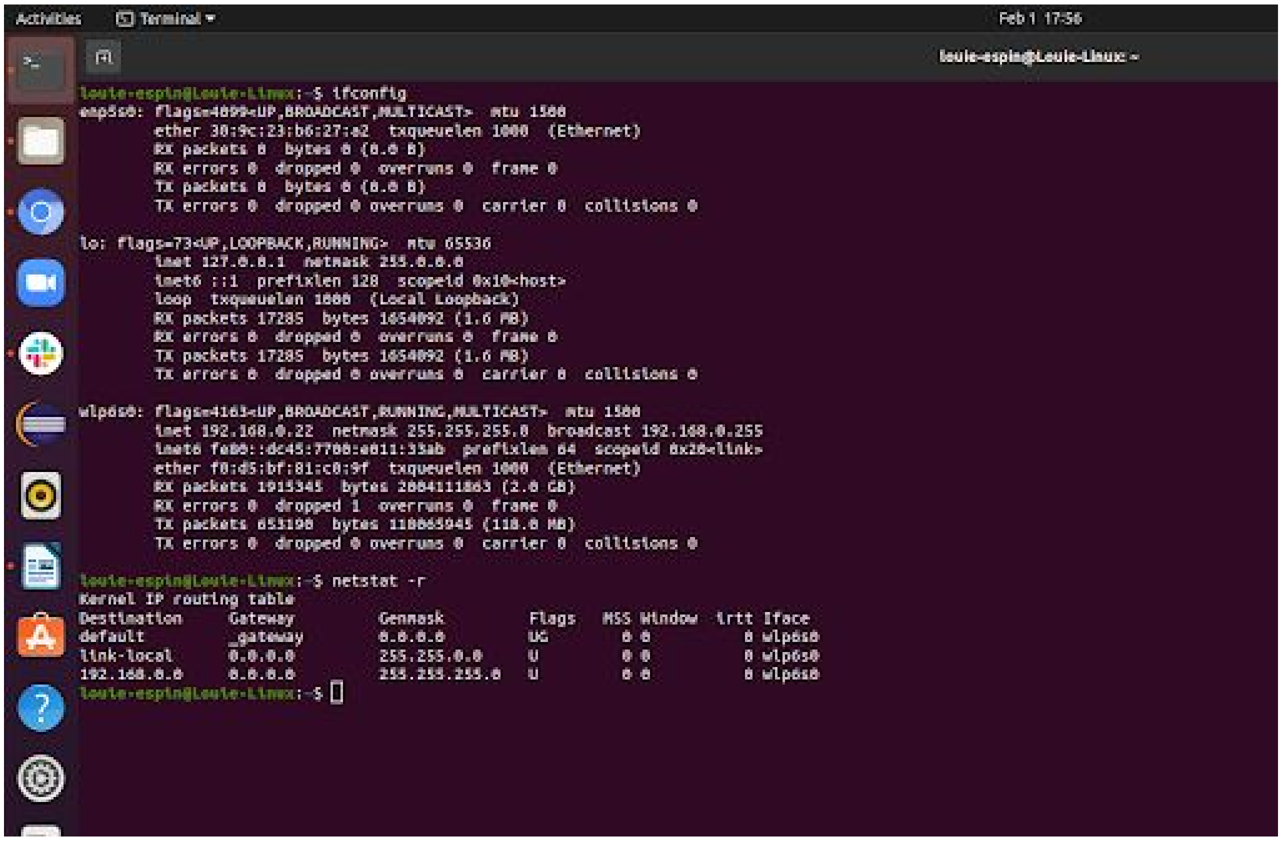
1.1 Explore the Data Link Layer with ARP

1. Capture a Trace

Steps 1 and 2: ifconfig and netstat



```
louie-espin@Louie-Linux:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    600    0        0 wlp6s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 wlp6s0
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0 wlp6s0
louie-espin@Louie-Linux:~$ 
```



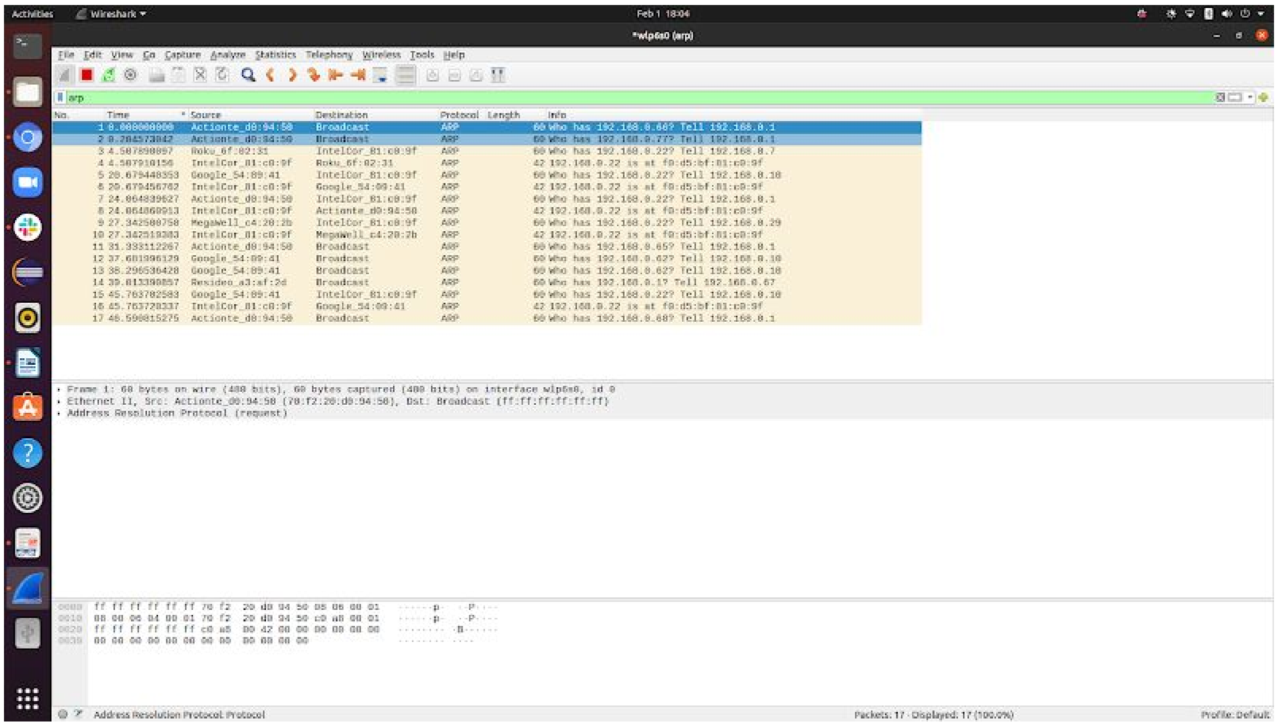
```
louie-espin@Louie-Linux:~$ ifconfig
enp5s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 30:9c:23:b6:27:a2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17285  bytes 1654092 (1.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17285  bytes 1654092 (1.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp6s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.22  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::dc45:7700:e011:33ab  prefixlen 64  scopeid 0x20<link>
        ether f0:d5:bf:81:c0:9f  txqueuelen 1000  (Ethernet)
        RX packets 1915345  bytes 2004111863 (2.0 GB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 653190  bytes 110065945 (110.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

louie-espin@Louie-Linux:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 wlp6s0
link-local      0.0.0.0         255.255.0.0     U         0 0          0 wlp6s0
192.168.0.0     0.0.0.0         255.255.255.0   U         0 0          0 wlp6s0
louie-espin@Louie-Linux:~$ 
```

Step 3: Launch Wireshark with arp filter


Activities
Wireshark
Feb 1 18:04
*wlp6s0 (arp)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp

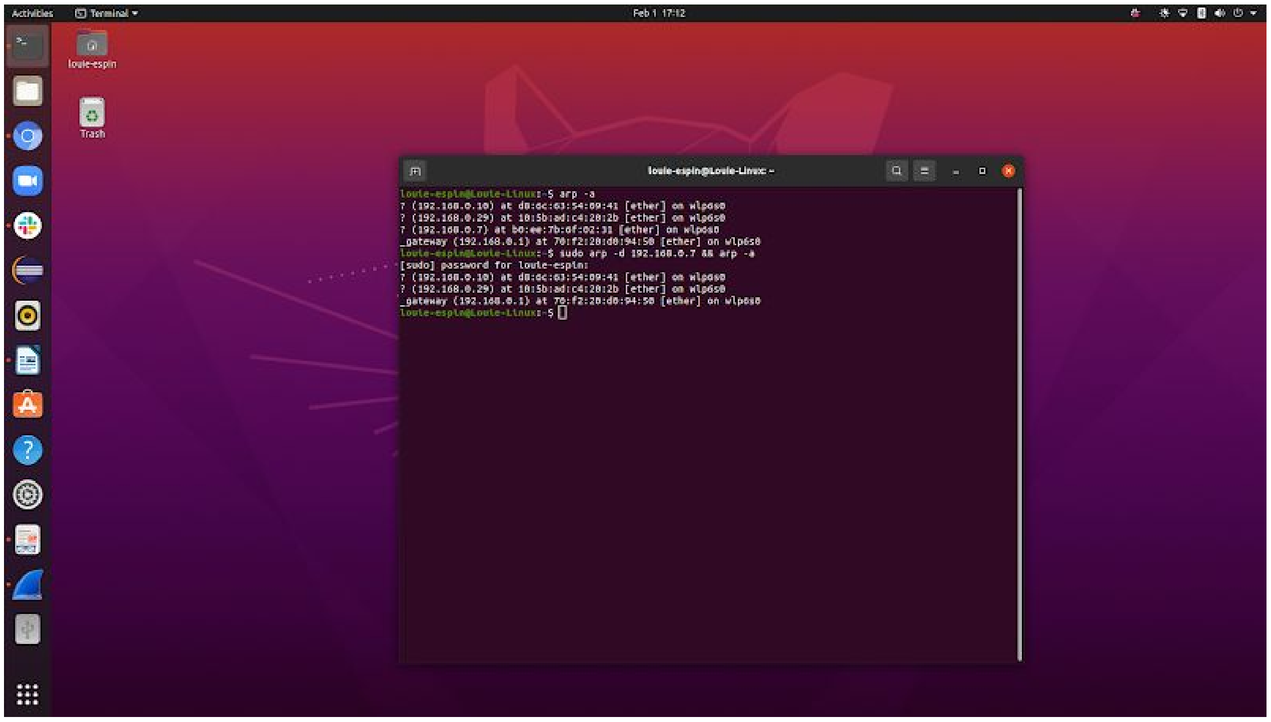
```
No.   Time          Source              Destination         Protocol Length  Info
    1 0.000000000   Actionte_d0:94:50   Broadcast           ARP       60 Who has 192.168.0.68? Tell 192.168.0.1
    2 0.204573042   Actionte_d0:94:50   Broadcast           ARP       60 Who has 192.168.0.77? Tell 192.168.0.1
    3 4.507890097   Roku_6f:02:31       IntelCor_81:c0:9f   ARP       60 Who has 192.168.0.22? Tell 192.168.0.7
    4 4.507910156   IntelCor_81:c0:9f   Roku_6f:02:31       ARP       42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    5 20.679448353  Google_54:09:41     IntelCor_81:c0:9f   ARP       60 Who has 192.168.0.22? Tell 192.168.0.10
    6 20.679456762  IntelCor_81:c0:9f   Google_54:09:41     ARP       42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    7 24.064839627  Actionte_d0:94:50   IntelCor_81:c0:9f   ARP       60 Who has 192.168.0.22? Tell 192.168.0.1
    8 24.064860913  IntelCor_81:c0:9f   Actionte_d0:94:50   ARP       42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    9 27.342500758  MegaWell_c4:20:2b   IntelCor_81:c0:9f   ARP       60 Who has 192.168.0.22? Tell 192.168.0.29
   10 27.342519383  IntelCor_81:c0:9f   MegaWell_c4:20:2b   ARP       42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   11 31.333112267  Actionte_d0:94:50   Broadcast           ARP       60 Who has 192.168.0.65? Tell 192.168.0.1
   12 37.601996129  Google_54:09:41     Broadcast           ARP       60 Who has 192.168.0.62? Tell 192.168.0.10
   13 38.296536428  Google_54:09:41     Broadcast           ARP       60 Who has 192.168.0.62? Tell 192.168.0.10
   14 39.013390057  Resideo_a3:af:2d    Broadcast           ARP       60 Who has 192.168.0.1? Tell 192.168.0.67
   15 45.763782583  Google_54:09:41     IntelCor_81:c0:9f   ARP       60 Who has 192.168.0.22? Tell 192.168.0.10
   16 45.763728337  IntelCor_81:c0:9f   Google_54:09:41     ARP       42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   17 46.590815275  Actionte_d0:94:50   Broadcast           ARP       60 Who has 192.168.0.68? Tell 192.168.0.1

› Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface wlp6s0, id 0
› Ethernet II, Src: Actionte_d0:94:50 (70:f2:20:d0:94:50), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 70 f2  20 d0 94 50 08 06 00 01   ······p· ·P····
0010  08 00 06 04 00 01 70 f2  20 d0 94 50 c0 a8 00 01   ······p· ·P····
0020  ff ff ff ff ff ff c0 a8  00 42 00 00 00 00 00 00   ········ ·B······
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····

Address Resolution Protocol: Protocol          Packets: 17 · Displayed: 17 (100.0%)          Profile: Default
```

Step 4: arp -a and arp -d

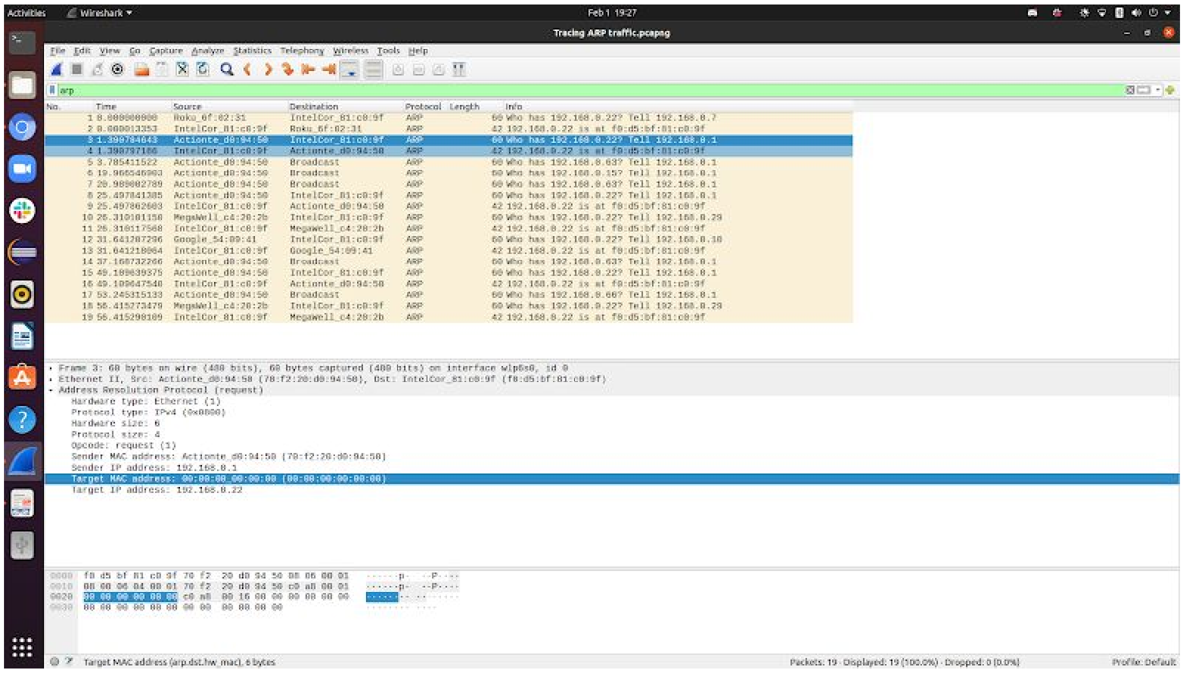


```
louie-espin@Louie-Linux:~$ arp -a
? (192.168.0.10) at d8:ec:63:54:09:41 [ether] on wlp6s0
? (192.168.0.29) at 10:5b:ad:c4:20:2b [ether] on wlp6s0
? (192.168.0.7) at b0:ee:7b:6f:02:31 [ether] on wlp6s0
_gateway (192.168.0.1) at 70:f2:20:d0:94:50 [ether] on wlp6s0
louie-espin@Louie-Linux:~$ sudo arp -d 192.168.0.7 && arp -a
[sudo] password for louie-espin:
? (192.168.0.10) at d8:ec:63:54:09:41 [ether] on wlp6s0
? (192.168.0.29) at 10:5b:ad:c4:20:2b [ether] on wlp6s0
_gateway (192.168.0.1) at 70:f2:20:d0:94:50 [ether] on wlp6s0
louie-espin@Louie-Linux:~$ 
```

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



Tracing ARP traffic.pcapng

arp

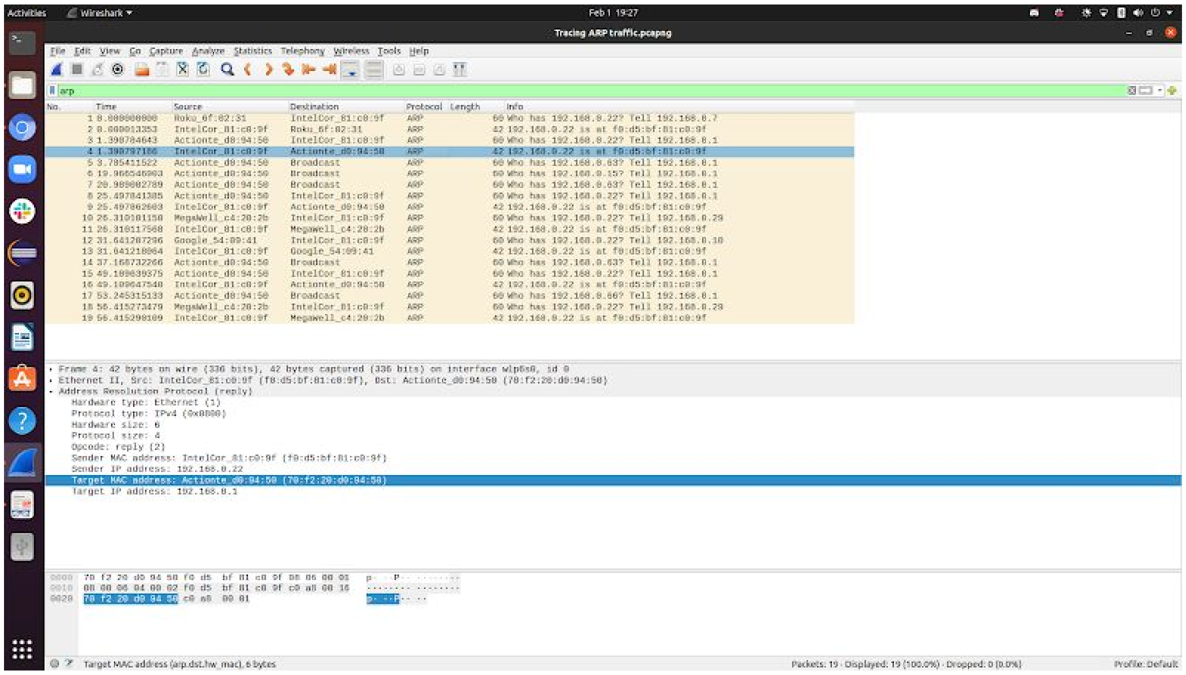
```
No.   Time            Source               Destination          Protocol  Length  Info
    1 0.000000000    Roku_6f:02:31        IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.7
    2 0.000013353    IntelCor_81:c0:9f    Roku_6f:02:31        ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    3 1.390784643    Actionte_d0:94:50    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.1
    4 1.390797186    IntelCor_81:c0:9f    Actionte_d0:94:50    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    5 3.785411522    Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.63? Tell 192.168.0.1
    6 19.966546903   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.15? Tell 192.168.0.1
    7 20.989002789   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.63? Tell 192.168.0.1
    8 25.497841385   Actionte_d0:94:50    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.1
    9 25.497862603   IntelCor_81:c0:9f    Actionte_d0:94:50    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   10 26.310101150   MegaWell_c4:20:2b    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.29
   11 26.310117568   IntelCor_81:c0:9f    Megawell_c4:20:2b    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   12 31.641207296   Google_54:09:41      IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.10
   13 31.641218064   IntelCor_81:c0:9f    Google_54:09:41      ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   14 37.168732266   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.63? Tell 192.168.0.1
   15 49.109639375   Actionte_d0:94:50    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.1
   16 49.109647540   IntelCor_81:c0:9f    Actionte_d0:94:50    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   17 53.245315133   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.66? Tell 192.168.0.1
   18 56.415273479   MegaWell_c4:20:2b    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.29
   19 56.415298109   IntelCor_81:c0:9f    Megawell_c4:20:2b    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
```

```
• Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface wlp6s0, id 0
• Ethernet II, Src: Actionte_d0:94:50 (70:f2:20:d0:94:50), Dst: IntelCor_81:c0:9f (f0:d5:bf:81:c0:9f)
• Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Actionte_d0:94:50 (70:f2:20:d0:94:50)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.22
```

```
0000  f0 d5 bf 81 c0 9f 70 f2  20 d0 94 50 08 06 00 01   ······p· ··P····
0010  08 00 06 04 00 01 70 f2  20 d0 94 50 c0 a8 00 01   ······p· ··P····
0020  00 00 00 00 00 00 c0 a8  00 16 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

Target MAC address (arp.dst.hw_mac), 6 bytes   Packets: 19 · Displayed: 19 (100.0%) · Dropped: 0 (0.0%)   Profile: Default

Step 2: The reply



Tracing ARP traffic.pcapng

arp

```
No.   Time            Source               Destination          Protocol  Length  Info
    1 0.000000000    Roku_6f:02:31        IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.7
    2 0.000013353    IntelCor_81:c0:9f    Roku_6f:02:31        ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    3 1.390784643    Actionte_d0:94:50    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.1
    4 1.390797186    IntelCor_81:c0:9f    Actionte_d0:94:50    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
    5 3.785411522    Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.63? Tell 192.168.0.1
    6 19.966546903   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.15? Tell 192.168.0.1
    7 20.989002789   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.63? Tell 192.168.0.1
    8 25.497841385   Actionte_d0:94:50    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.1
    9 25.497862603   IntelCor_81:c0:9f    Actionte_d0:94:50    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   10 26.310101150   MegaWell_c4:20:2b    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.29
   11 26.310117568   IntelCor_81:c0:9f    Megawell_c4:20:2b    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   12 31.641207296   Google_54:09:41      IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.10
   13 31.641218064   IntelCor_81:c0:9f    Google_54:09:41      ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   14 37.168732266   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.63? Tell 192.168.0.1
   15 49.109639375   Actionte_d0:94:50    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.1
   16 49.109647540   IntelCor_81:c0:9f    Actionte_d0:94:50    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
   17 53.245315133   Actionte_d0:94:50    Broadcast            ARP             60 Who has 192.168.0.66? Tell 192.168.0.1
   18 56.415273479   MegaWell_c4:20:2b    IntelCor_81:c0:9f    ARP             60 Who has 192.168.0.22? Tell 192.168.0.29
   19 56.415298109   IntelCor_81:c0:9f    Megawell_c4:20:2b    ARP             42 192.168.0.22 is at f0:d5:bf:81:c0:9f
```

```
• Frame 4: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp6s0, id 0
• Ethernet II, Src: IntelCor_81:c0:9f (f0:d5:bf:81:c0:9f), Dst: Actionte_d0:94:50 (70:f2:20:d0:94:50)
• Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_81:c0:9f (f0:d5:bf:81:c0:9f)
    Sender IP address: 192.168.0.22
    Target MAC address: Actionte_d0:94:50 (70:f2:20:d0:94:50)
    Target IP address: 192.168.0.1
```

```
0000  70 f2 20 d0 94 50 f0 d5  bf 81 c0 9f 08 06 00 01   p· ··P·· ········
0010  08 00 06 04 00 02 f0 d5  bf 81 c0 9f c0 a8 00 16   ········ ········
0020  70 f2 20 d0 94 50 c0 a8  00 01                     p· ··P·· ··
```

Target MAC address (arp.dst.hw_mac), 6 bytes   Packets: 19 · Displayed: 19 (100.0%) · Dropped: 0 (0.0%)   Profile: Default

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

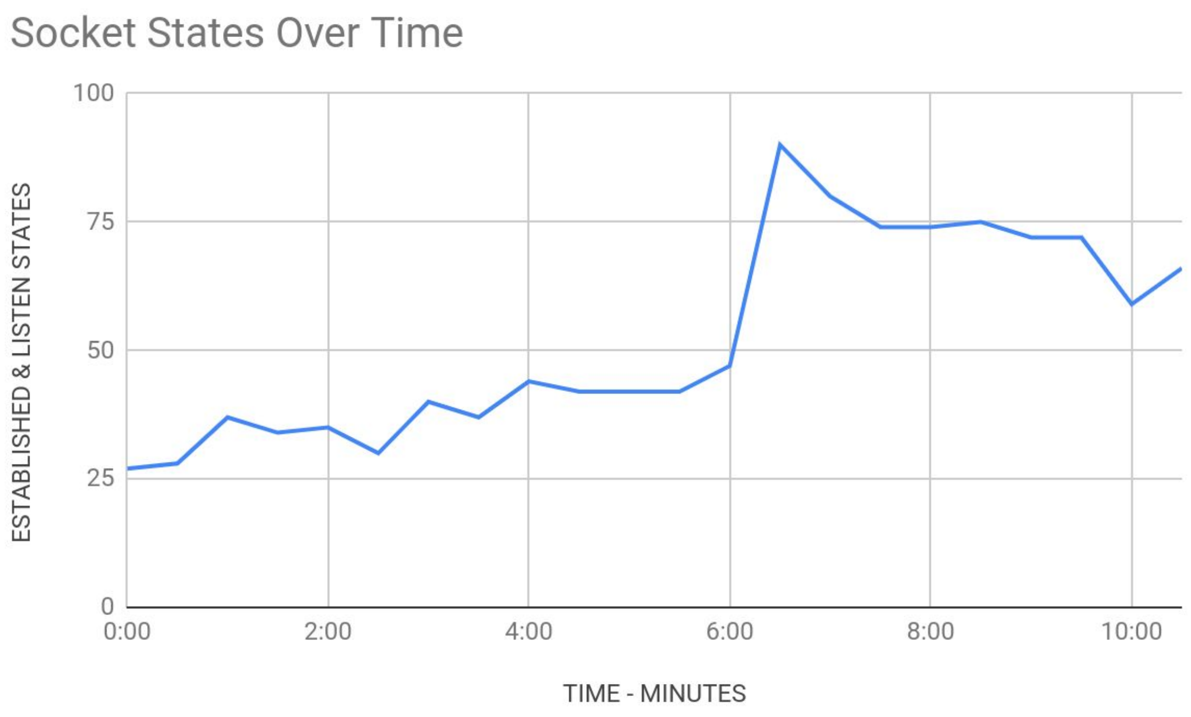
2. How large is the ARP header for a request? What about for a reply?
They are both 28 bytes

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"

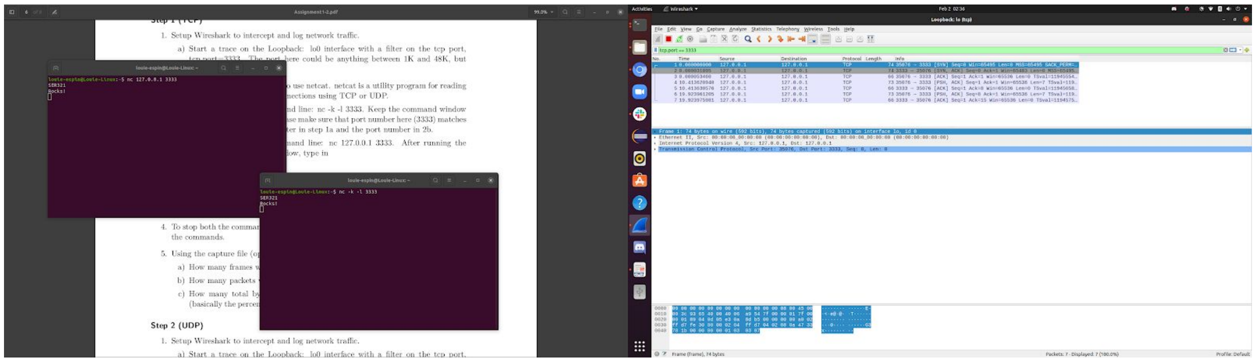


1.3 Sniffing TCP/UDP traffic

TCP: Wireshark capture

Wireshark capture added to github repository.

TCP: netcat commands



```
louie-espin@Louie-Linux:~$ nc 127.0.0.1 3333
SER321
Rocks!
```

```
louie-espin@Louie-Linux:~$ nc -k -l 3333
SER321
Rocks!
```

```
tcp.port == 3333
No. Time          Source     Destination Protocol Length Info
  1 0.000000000   127.0.0.1  127.0.0.1   TCP      74 35076 → 3333 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=…
  2 0.000031995   127.0.0.1  127.0.0.1   TCP      74 3333 → 35076 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495…
  3 0.000053460   127.0.0.1  127.0.0.1   TCP      66 35076 → 3333 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=11945554…
  4 10.413620940  127.0.0.1  127.0.0.1   TCP      73 35076 → 3333 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=7 TSval=119…
  5 10.413630576  127.0.0.1  127.0.0.1   TCP      66 3333 → 35076 [ACK] Seq=1 Ack=8 Win=65536 Len=0 TSval=11945658…
  6 19.923961205  127.0.0.1  127.0.0.1   TCP      73 35076 → 3333 [PSH, ACK] Seq=8 Ack=1 Win=65536 Len=7 TSval=119…
  7 19.923975981  127.0.0.1  127.0.0.1   TCP      66 3333 → 35076 [ACK] Seq=1 Ack=15 Win=65536 Len=0 TSval=1194575…

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 35076, Dst Port: 3333, Seq: 0, Len: 0
```

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.

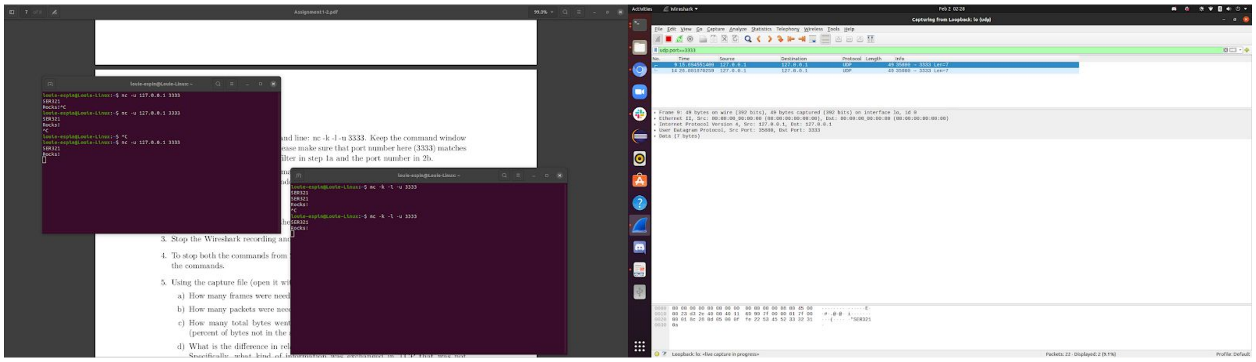
3. How many total bytes went over the wire? How much overhead was there (basically the percentage of traffic that was not needed to send SER321 Rocks!)?

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



```
louie-espin@Louie-Linux:~$ nc -u 127.0.0.1 3333
SER321
Rocks!^C
louie-espin@Louie-Linux:~$ nc -u 127.0.0.1 3333
SER321
Rocks!
^C
louie-espin@Louie-Linux:~$ ^C
louie-espin@Louie-Linux:~$ nc -u 127.0.0.1 3333
SER321
Rocks!
```

```
louie-espin@Louie-Linux:~$ nc -k -l -u 3333
SER321
SER321
Rocks!
^C
louie-espin@Louie-Linux:~$ nc -k -l -u 3333
SER321
Rocks!
```

```
udp.port==3333
No.  Time            Source     Destination  Protocol  Length  Info
   9 15.694551409    127.0.0.1  127.0.0.1    UDP       49      35880 → 3333 Len=7
  14 26.881878259    127.0.0.1  127.0.0.1    UDP       49      35880 → 3333 Len=7

▸ Frame 9: 49 bytes on wire (392 bits), 49 bytes captured (392 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▸ User Datagram Protocol, Src Port: 35880, Dst Port: 3333
▸ Data (7 bytes)
```

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.

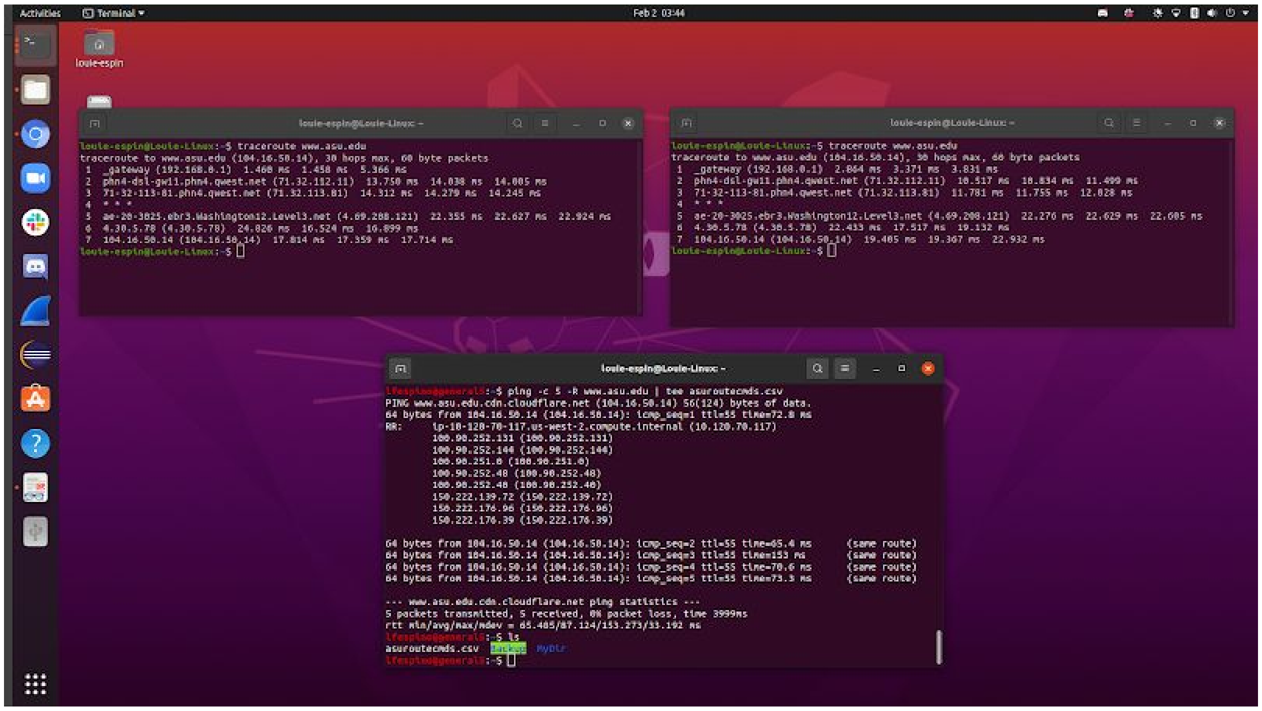
3. How many total bytes went over the wire? How much overhead was there (percent of bytes not in the above 2 lines)?

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



```
louie-espin@Louie-Linux:~$ traceroute www.asu.edu
traceroute to www.asu.edu (104.16.50.14), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  1.460 ms  1.458 ms  5.366 ms
 2  phn4-dsl-gw11.phn4.qwest.net (71.32.112.11)  13.750 ms  14.038 ms  14.005 ms
 3  71-32-113-81.phn4.qwest.net (71.32.113.81)  14.312 ms  14.279 ms  14.245 ms
 4  * * *
 5  ae-20-3025.ebr3.Washington12.Level3.net (4.69.208.121)  22.355 ms  22.627 ms  22.924 ms
 6  4.30.5.78 (4.30.5.78)  24.826 ms  16.524 ms  16.899 ms
 7  104.16.50.14 (104.16.50.14)  17.814 ms  17.359 ms  17.714 ms
louie-espin@Louie-Linux:~$
```

```
louie-espin@Louie-Linux:~$ traceroute www.asu.edu
traceroute to www.asu.edu (104.16.50.14), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  2.864 ms  3.371 ms  3.831 ms
 2  phn4-dsl-gw11.phn4.qwest.net (71.32.112.11)  10.517 ms  10.834 ms  11.499 ms
 3  71-32-113-81.phn4.qwest.net (71.32.113.81)  11.781 ms  11.755 ms  12.028 ms
 4  * * *
 5  ae-20-3025.ebr3.Washington12.Level3.net (4.69.208.121)  22.276 ms  22.629 ms  22.085 ms
 6  4.30.5.78 (4.30.5.78)  22.433 ms  17.517 ms  19.132 ms
 7  104.16.50.14 (104.16.50.14)  19.405 ms  19.367 ms  22.932 ms
louie-espin@Louie-Linux:~$
```

```
lfespino@general5:~$ ping -c 5 -R www.asu.edu | tee asuroutecmds.csv
PING www.asu.edu.cdn.cloudflare.net (104.16.50.14) 56(124) bytes of data.
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=1 ttl=55 time=72.8 ms
RR:     ip-10-120-70-117.us-west-2.compute.internal (10.120.70.117)
        100.90.252.131 (100.90.252.131)
        100.90.252.144 (100.90.252.144)
        100.90.251.0 (100.90.251.0)
        100.90.252.48 (100.90.252.48)
        100.90.252.48 (100.90.252.48)
        150.222.139.72 (150.222.139.72)
        150.222.176.96 (150.222.176.96)
        150.222.176.39 (150.222.176.39)

64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=2 ttl=55 time=65.4 ms      (same route)
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=3 ttl=55 time=153 ms       (same route)
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=4 ttl=55 time=70.6 ms      (same route)
64 bytes from 104.16.50.14 (104.16.50.14): icmp_seq=5 ttl=55 time=73.3 ms      (same route)

--- www.asu.edu.cdn.cloudflare.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 65.405/87.124/153.273/33.192 ms
lfespino@general5:~$ ls
asuroutecmds.csv  Backup  mydir
lfespino@general5:~$
```

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