

M4 Lab: Data Link Layer

CITA 220: DATA COMM & NETWORK TECH

Gonzalez, Marco A.
SUNY CANTON | 34 CORNELL DRIVE, CANTON, NEW YORK 13617

TABLE OF CONTENTS

1	Table of Figures	1
2	Table of Tables	2
3	Windows	2
3.1	MAC Address.....	2
3.1.1	Method 1 (ipconfig)	2
3.2	Method 2 (getmac)	3
3.3	ARP	3
4	MAC Address Structure.....	4
5	Linux.....	5
5.1	Frame Check Sequence	5
5.2	MAC Address.....	6
5.2.1	Method 1 (ifconfig)	6
5.2.2	Method 2 (ip)	7
5.3	ARP	7
5.3.1	Method 1 (arp).....	7
5.3.2	Method 2 (ip)	7
6	Capturing ARP Broadcasts Using Wireshark	8
6.1	Starting Wireshark	8
6.2	Changing the Network Settings	8
6.3	Starting and Stopping to Capture Frames.....	9
6.4	Examining ARP Frames.....	10

1 TABLE OF FIGURES

Figure 1. The ipconfig Command	2
Figure 2. Alternate ipconfig Method.....	3
Figure 3. The getmac Command	3
Figure 4. The arp Command.....	4
Figure 5. Windows Calculator in Programmer Mode	5
Figure 6. Creating a Data File	6
Figure 7. CRC Checksum.....	6
Figure 8. A Slight Different Data Changed the CRC Checksum	6

Figure 9. The ifconfig Command	7
Figure 10. The ip link Command	7
Figure 11. The arp Command.....	7
Figure 12. The ip neighbor Command.....	7
Figure 13. Starting Wireshark	8
Figure 14. Changing the Network Settings	9
Figure 15. Changing to Bridged Adapter.....	9
Figure 16. Starting to Capture Frames.....	9
Figure 17. Stopping to Capture Frames	10
Figure 18. Captured ARP Frames	10

2 TABLE OF TABLES

Table 1. MAC Address Structure	4
--------------------------------------	---

3 WINDOWS

This section reviews some Windows data link layer commands. Use a Command Prompt.

3.1 MAC ADDRESS

A MAC address is a **locally significant** device identifier. There are two common methods to obtain the MAC addresses.

3.1.1 Method 1 (ipconfig)

Execute the following command. The MAC address is displayed as **Physical Address**. See Figure 1.

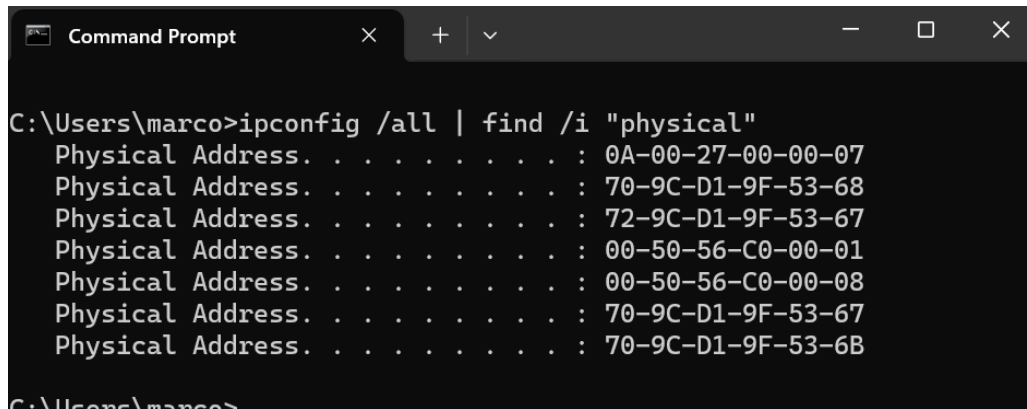
```

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix . : 
    Description . . . . . : Intel(R) Wi-Fi 6 AX1650w 160MHz Wireless Network Adapter (200D2W)
    Physical Address. . . . . : 70-9C-D1-9F-53-67
    Driver Version . . . . . : 
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::cd0c:fa2e:80:9244%21(Preferred)
    IPv4 Address. . . . . : 10.11.65.11(Preferred)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained. . . . . : Thursday, October 19, 2023 5:11:21 PM
    Lease Expires . . . . . : Wednesday, November 1, 2023 10:52:18 PM
    Default Gateway . . . . . : 10.11.65.1
    DHCP Server . . . . . : 172.17.112.2
    DHCPv6 IAID . . . . . : 359701713
    DHCPv6 Client DUID. . . . . : 00-01-00-01-2A-73-97-C0-70-9C-D1-9F-53-67
    DNS Servers . . . . . : 8.8.8.8
    . . . . . : 1.1.1.1
    NetBIOS over Tcpip. . . . . : Enabled
  
```

Figure 1. The ipconfig Command

Alternatively, the same information can be obtained by the following method. See Figure 2.



```

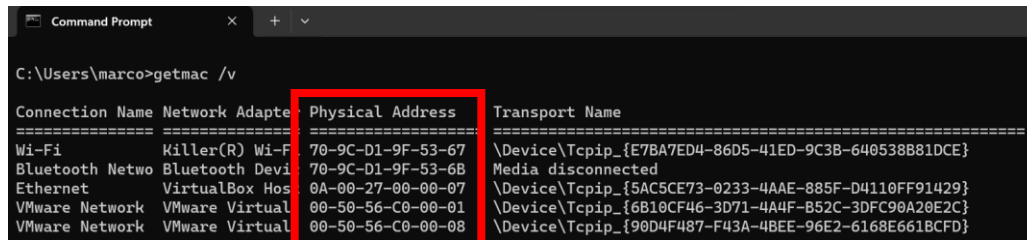
C:\Users\marco>ipconfig /all | find /i "physical"
Physical Address. . . . . : 0A-00-27-00-00-07
Physical Address. . . . . : 70-9C-D1-9F-53-68
Physical Address. . . . . : 72-9C-D1-9F-53-67
Physical Address. . . . . : 00-50-56-C0-00-01
Physical Address. . . . . : 00-50-56-C0-00-08
Physical Address. . . . . : 70-9C-D1-9F-53-67
Physical Address. . . . . : 70-9C-D1-9F-53-6B

```

Figure 2. Alternate ipconfig Method

3.2 METHOD 2 (GETMAC)

Execute the following command. See Figure 3.



```

C:\Users\marco>getmac /v

Connection Name Network Adapter Physical Address Transport Name
=====
Wi-Fi Killer(R) Wi-Fi 70-9C-D1-9F-53-67 \Device\Tcpip_{E7BA7ED4-86D5-41ED-9C3B-640538B81DCE}
Bluetooth Netwo Bluetooth Devi 70-9C-D1-9F-53-68 Media disconnected
Ethernet VirtualBox Hos 0A-00-27-00-00-07 \Device\Tcpip_{5AC5CE73-0233-4AAE-885F-D4110FF91429}
VMware Network VMware Virtual 00-50-56-C0-00-01 \Device\Tcpip_{6B10CF46-3D71-4A4F-B52C-3DFC90A20E2C}
VMware Network VMware Virtual 00-50-56-C0-00-08 \Device\Tcpip_{90D4F487-F43A-4BEE-96E2-6168E661BCFD}

```

Figure 3. The getmac Command

3.3 ARP

ARP (Address Resolution Protocol) is used to query a MAC address from a given IP address. Queried MAC addresses are temporarily stored in a table in the RAM as **cached data**. This table is called an **ARP table**. To view the current contents of the ARP table, execute the **arp** command. See Figure 4.

```

C:\Users\marco>arp -a

Interface: 192.168.56.1 --- 0x7
    Internet Address      Physical Address      Type
    192.168.56.255        ff-ff-ff-ff-ff-ff     static
    224.0.0.22            01-00-5e-00-00-16     static
    224.0.0.251          01-00-5e-00-00-fb     static
    224.0.0.252          01-00-5e-00-00-fc     static
    239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.29.1 --- 0xa
    Internet Address      Physical Address      Type
    192.168.29.254        00-50-56-e1-47-54     dynamic
    192.168.29.255        ff-ff-ff-ff-ff-ff     static
    224.0.0.22            01-00-5e-00-00-16     static
    224.0.0.251          01-00-5e-00-00-fb     static
    224.0.0.252          01-00-5e-00-00-fc     static
    239.255.255.250       01-00-5e-7f-ff-fa     static
    255.255.255.255       ff-ff-ff-ff-ff-ff     static

```

Figure 4. The arp Command

In this example, the device's MAC address at 192.168.29.254 (IPv4 address) is 00-50-56-e1-47-54. The type **dynamic** means that this entry was obtained via arp. You may sometimes see the type **static**. Static means that the entry was manually or automatically created by a system administrator or the operating system at boot time.

4 MAC ADDRESS STRUCTURE

Consider a MAC address E4-54-E8-A0-35-6B. See Table 1.

Table 1. MAC Address Structure

Part	OUI (Byte 1)	OUI (Byte 2)	OUI (Byte 3)	S/N (Byte 1)	S/N (Byte 2)	S/N (Byte 3)
MAC (Hex)	E4	54	E8	A0	35	6B
Decimal	228	84	232	160	53	107
Binary (Bits)	1110 0100	0101 0100	1110 1000	1010 0000	0011 0101	0110 1011
	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte
	1 Octet	1 Octet	1 Octet	1 Octet	1 Octet	1 Octet

Therefore, a MAC address consists of 48 bits, 6 bytes, or 6 octets. The first 24 bits (3 bytes or 3 octets) are called an OUI (organizationally unique identifier), which uniquely identifies the manufacturer of the network interface card (NIC). The second 24 bits (3 bytes or 3 octets) are used as the NIC serial number.

Note that a hexadecimal number can be converted into decimal, octal, and binary equivalent numbers using Windows Calculator in the Programmer mode. To switch to the Programmer mode, click the ≡ button and choose **Programmer**. See Figure 5.

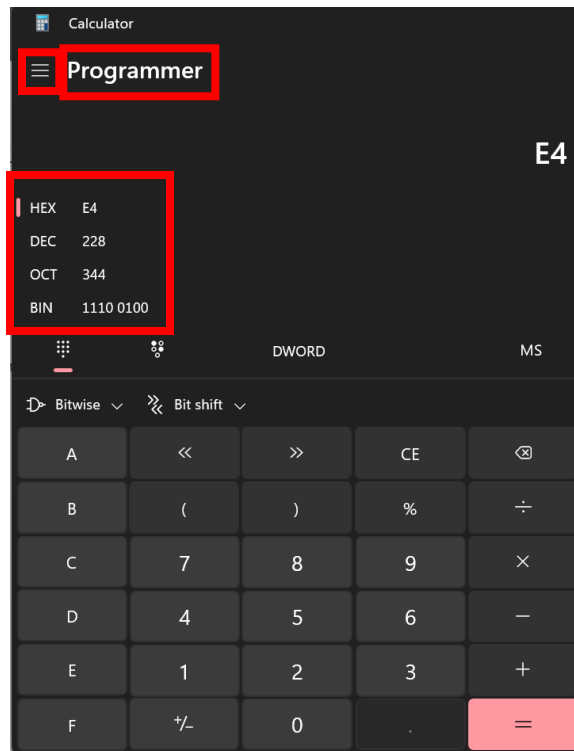


Figure 5. Windows Calculator in Programmer Mode

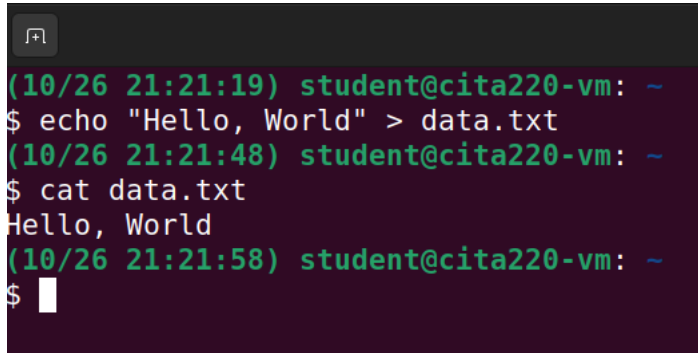
5 LINUX

This section reviews some Linux data link layer commands.

5.1 FRAME CHECK SEQUENCE

The data integrity of a frame is checked using a unique code in the frame trailer called a **frame check sequence** (FCS). This number is calculated using a mathematical formula called the **cyclic redundant check** (CRC). This formula calculates a unique number for the data. The following example shows how the CRC works.

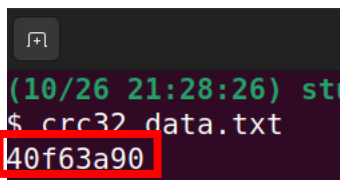
Create a data file named **data.txt** that contains *Hello, World*. See Figure 6.

A terminal window with a dark background and green text. It shows a user at a VM prompt creating a file named 'data.txt' with the text 'Hello, World' and then verifying its contents.

```
(10/26 21:21:19) student@cita220-vm: ~  
$ echo "Hello, World" > data.txt  
(10/26 21:21:48) student@cita220-vm: ~  
$ cat data.txt  
Hello, World  
(10/26 21:21:58) student@cita220-vm: ~  
$
```

Figure 6. Creating a Data File

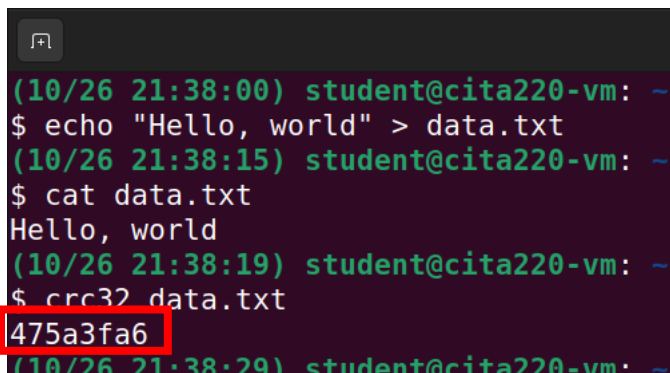
Run this data file through the CRC formula. Notice a hexadecimal number is returned. See Figure 7.

A terminal window showing the command to calculate the CRC32 checksum of 'data.txt'. The result '40f63a90' is highlighted with a red box.

```
(10/26 21:28:26) stu  
$ crc32 data.txt  
40f63a90
```

Figure 7. CRC Checksum

Next, change the contents of the data file. Notice the only difference is one letter, “w”. A different number is displayed when this data file is run through the CRC formula, although the data difference is subtle. See Figure 8.

A terminal window showing the user changing the content of 'data.txt' to 'Hello, world' and then calculating its CRC32 checksum. The new result '475a3fa6' is highlighted with a red box.

```
(10/26 21:38:00) student@cita220-vm: ~  
$ echo "Hello, world" > data.txt  
(10/26 21:38:15) student@cita220-vm: ~  
$ cat data.txt  
Hello, world  
(10/26 21:38:19) student@cita220-vm: ~  
$ crc32 data.txt  
475a3fa6  
(10/26 21:38:29) student@cita220-vm: ~
```

Figure 8. A Slight Different Data Changed the CRC Checksum

When the receiver receives a frame from the sender, it calculates the CRC checksum from the received data and compares it with the sender’s calculated checksum (FCS) that came with the frame’s data. If the checksums match, the receiver knows it has received the data intact.

5.2 MAC ADDRESS

There are two common methods to obtain IP addresses.

5.2.1 Method 1 (ifconfig)

The **ifconfig** command displays the MAC address as an **Ethernet** (ether) **address**. See Figure 9.

```

(10/27 17:46:11) student@cita220-vm: ~
$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::862b:1d5f:0e04:e378 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:b9:82:41 txqueuelen 1000 (Ethernet)
    RX packets 22702 bytes 33437333 (33.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5144 bytes 416576 (416.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figure 9. The ifconfig Command

5.2.2 Method 2 (ip)

The **ip** command used with the **link** subcommand displays the MAC address as a **link** (Ethernet address). See Figure 10.

```

(10/27 17:49:51) student@cita220-vm: ~
$ ip link
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:b9:82:41 brd ff:ff:ff:ff:ff:ff

```

Figure 10. The ip link Command

5.3 ARP

There are two common methods to obtain the MAC address.

5.3.1 Method 1 (arp)

The **arp** command is typically used the **-n** option. See Figure 11.

```

(10/27 17:52:20) student@cita220-vm: ~
$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.2                  ether    52:54:00:12:35:02    C                     enp0s3
(10/27 17:52:24) student@cita220-vm: ~

```

Figure 11. The arp Command

5.3.2 Method 2 (ip)

The **ip** command is used with a **neighbor** (n) subcommand. See Figure 12.

```

(10/27 17:55:07) student@cita220-vm: ~
$ ip n
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 STALE
(10/27 17:55:09) student@cita220-vm: ~

```

Figure 12. The ip neighbor Command

6 CAPTURING ARP BROADCASTS USING WIRESHARK

This section uses Wireshark to capture ARP broadcasts in the network. Wireshark is a well-known free, open-source software tool to capture frames that a network interface card (NIC) attached to the computer running Wireshark. It captures all frames regardless of where those frames are going. It then de-encapsulates those frames and displays the contents for analysis. Wireshark is classified as a **protocol analyzer** tool.

6.1 STARTING WIRESHARK

To start Wireshark, click the Wireshark icon. See Figure 13.

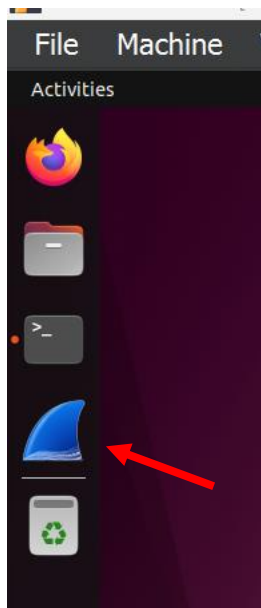


Figure 13. Starting Wireshark

6.2 CHANGING THE NETWORK SETTINGS

By default, the CITA220 resides in its private broadcast domain. Therefore, there are not many network frames to capture. More frames can be captured when placed on the same network the host operating system resides in. From the **Devices** menu, choose **Network > Network Settings...** . See Figure 14.

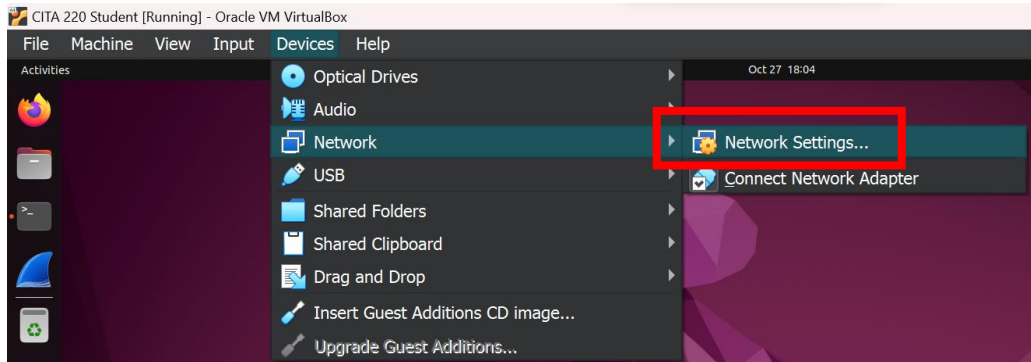


Figure 14. Changing the Network Settings

Change the **Attached to** from **NAT** to **Bridged Adapter** and click OK. Wait about 10 seconds. See Figure 15.

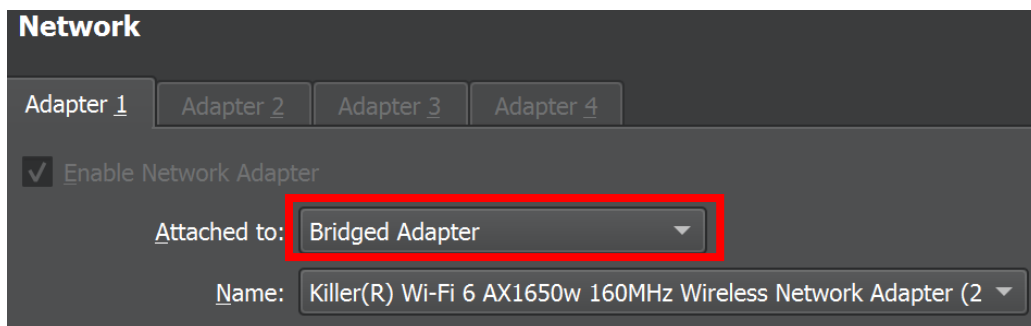


Figure 15. Changing to Bridged Adapter

6.3 STARTING AND STOPPING TO CAPTURE FRAMES

Make sure the **enps03** interface is selected. Click the blue **Start capturing packets** button to start capturing frames. See Figure 16.

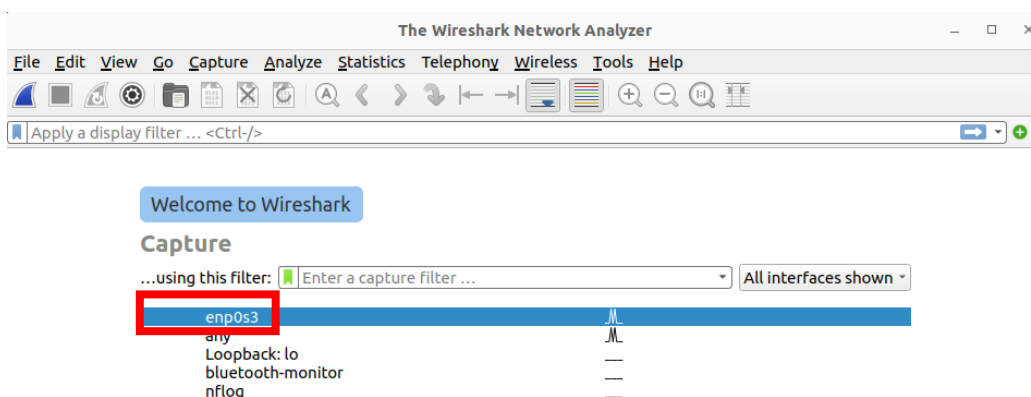


Figure 16. Starting to Capture Frames

The capturing starts immediately. Wait for 20 to 30 seconds. Then click the red **Stop capturing packets** button to stop capturing. See Figure 17.

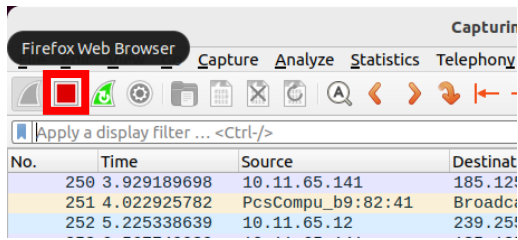


Figure 17. Stopping to Capture Frames

6.4 EXAMINING ARP FRAMES

Note that a large number of frames are captured within a short period. Search for ARP frames is very difficult. Wireshark supports a powerful display filter command set that allows the user to search various captured frames' information. Type **arp** and press Enter in the display filter bar. As you type, the background color becomes red. It changes to green after the command syntax becomes correct. Only the ARP frames are displayed. Select one of them and expand the **Ethernet II** section.

Note that Wireshark automatically translated the OUIs of the MAC addresses to the manufacturers' names. The destination MAC address is **ff:ff:ff:ff:ff:ff**. It is a special MAC address for a broadcast. Wireshark automatically replaces the MAC address with the word **Broadcast**. Also, note the message under the **Info** column. The "*Who has (receiver IP address)? Tell (sender IP address).*" Is the ARP broadcast message. See Figure 18.

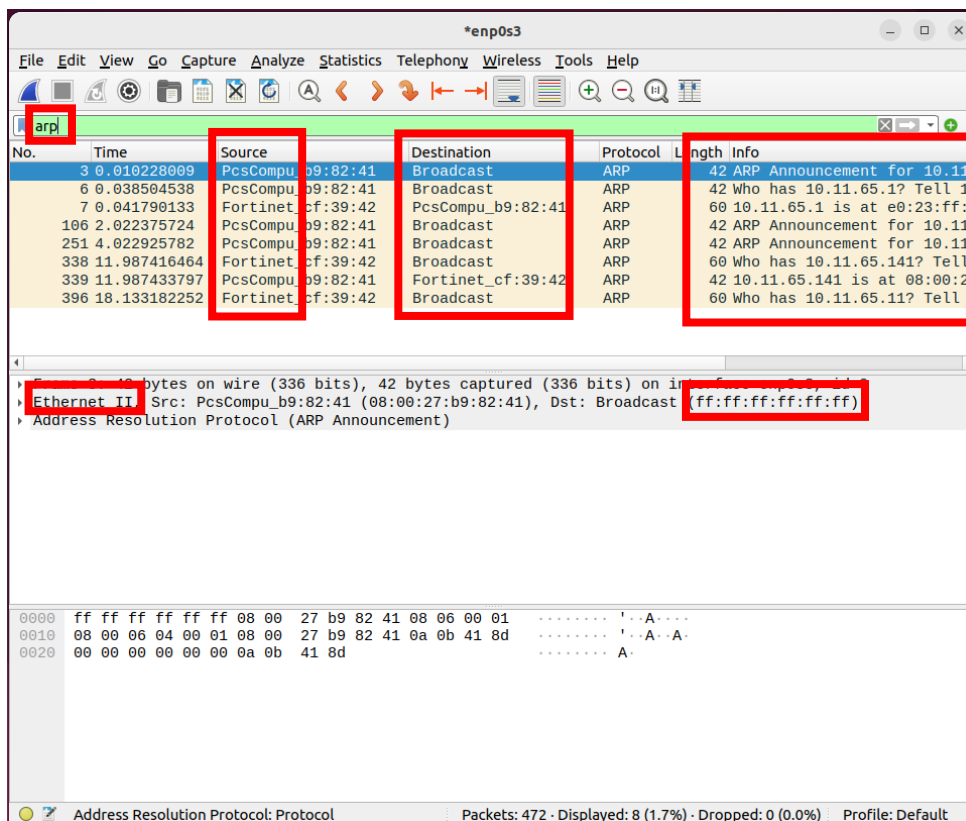


Figure 18. Captured ARP Frames