Lab 04 – ARP

Author: Raymond Ng
Course Number/Section: IS 3413-006
Date: September 23, 2022

INTRODUCTION

The purpose of this lab was to explore and analyze address resolution protocol (ARP) packets via Wireshark. Moreover, it allowed to user to further understand communication between internet protocols (IP) and media access controller (MAC) addresses as the pertain to the Data Link (Layer 2) and Network (Layer 3) layers.

PROCESS

Step 1

Per the instructions of this lab, I explored the use of `ipconfig` and `ipconfig /all` to understand the network interface cards (or network adapters) on my computer. For the purpose of this lab I was more interested in the Ethernet address of the main netowkr interface of my computer.

```
Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
   Physical Address. . . . . . . . . : 00-50-56-C0-00-01
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::30ed:c572:a0a9:999b%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.18.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 83906646
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2A-B2-E4-C8-BC-17-B8-13-62-67
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 1: Finding the computer's Ethernet address using command `ipconfig /all`.

Here, I executed `netstat -r` to find the local router (or default gateway) my computer uses to reach the rest of the the Internet. From observing the output, I can see my computer uses `192.168.4.1` as its default gateway.

```
C:\Users\rayng>netstat -r
===========================================================================
Interface List
  3...bc 17 b8 13 62 68 ......Microsoft Wi-Fi Direct Virtual Adapter
 10...be 17 b8 13 62 67 ......Microsoft Wi-Fi Direct Virtual Adapter #2
  7...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1
 13...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8
  6...bc 17 b8 13 62 67 ......Intel(R) Wi-Fi 6 AX201 160MHz
 15...bc 17 b8 13 62 6b ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.4.1    192.168.4.25     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      192.168.4.0    255.255.252.0         On-link      192.168.4.25    291
     192.168.4.25  255.255.255.255         On-link      192.168.4.25    291
    192.168.7.255  255.255.255.255         On-link      192.168.4.25    291
     192.168.18.0    255.255.255.0         On-link      192.168.18.1    291
     192.168.18.1  255.255.255.255         On-link      192.168.18.1    291
```

Figure 2: Finding the default gateway IP address using command `netstat -r`.

Here, I executed the `arp -a` command in the command terminal. [**Breakpoint 01**] Observing the outputs resulted, I see the Interfaces and their associative IP and MAC addresses. I know that IP addresses are associated with Layer 3, the Network Layer and MAC addresses (Ethernet) are associated with Layer 2, the Data layer. Within each interface the IP identified is communicating or "asking" the layer 2 address associated with the IP address [1].

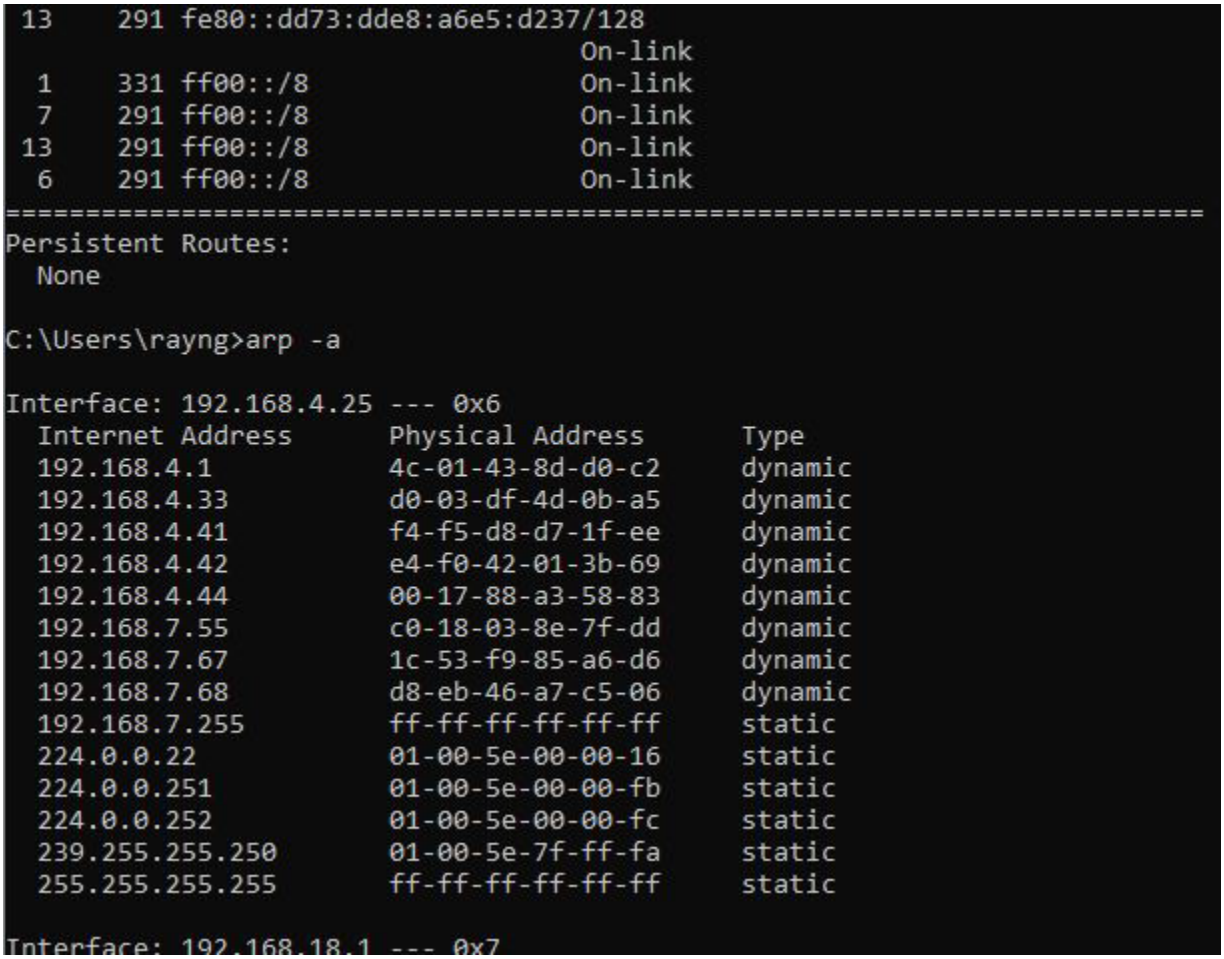

Figure 3: Using `arp -a` command in Windows.

Step 2 (Breakpoint2)

The personal laptop that I use for school does not have an Ethernet port, but I wanted to see if I could capture ARP packets with toher NICs that depicted activity in the Wireshark interface (*Figure 4*).

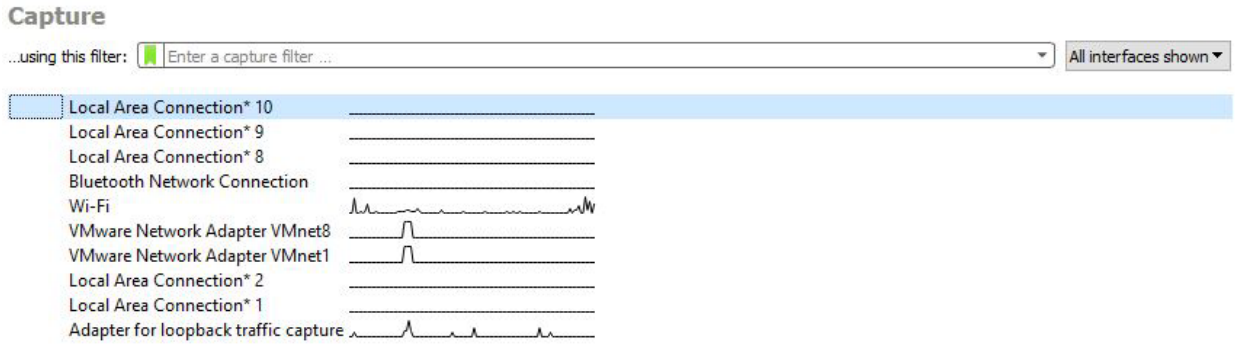

Figure 4: Wireshark interface depicting all NICs.

Beginning with the Wi-Fi NIC (*Figure 5*), I was able to capture a few ARP packets. When I observed the Source column I noticed that one of the smart devices in my home, `iRobot`, depicted.

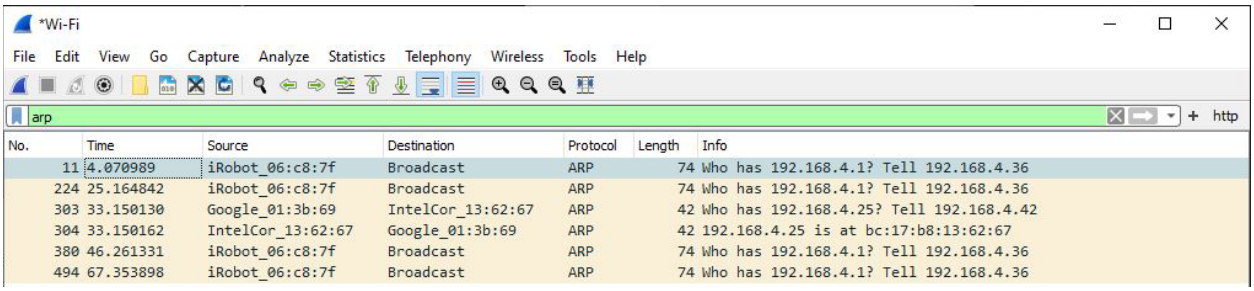

Figure 5: WiFi NIC Wireshark capture with "arp" filter.

Next, I decided to test the VMware Network Adapter VMnet1 because activity was detected on the Wireshark interface. No ARP packets were captured, however, I did observe Simple Service Discovery Protocols (SSDP). (*Figure 6*)

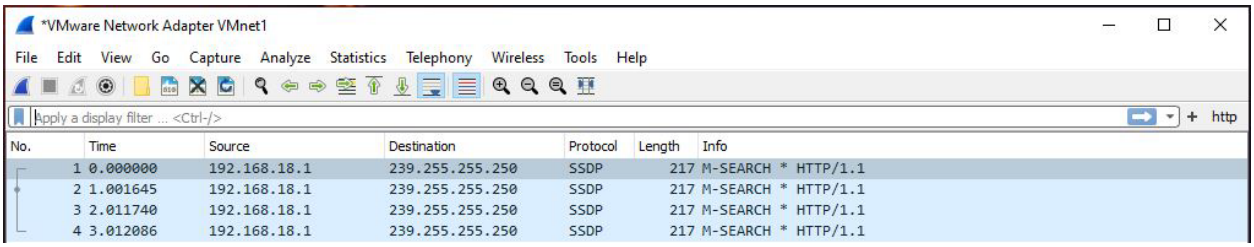

Figure 6: VMWare Network Adapter Wireshark capture.

For the purpose of this lab, I decided to use the `data-link.pcapng` file provided by the instructor to use for the analysis portion of this lab because it had several ARP packets and they were Ethernet captures. (*Figure 7*)

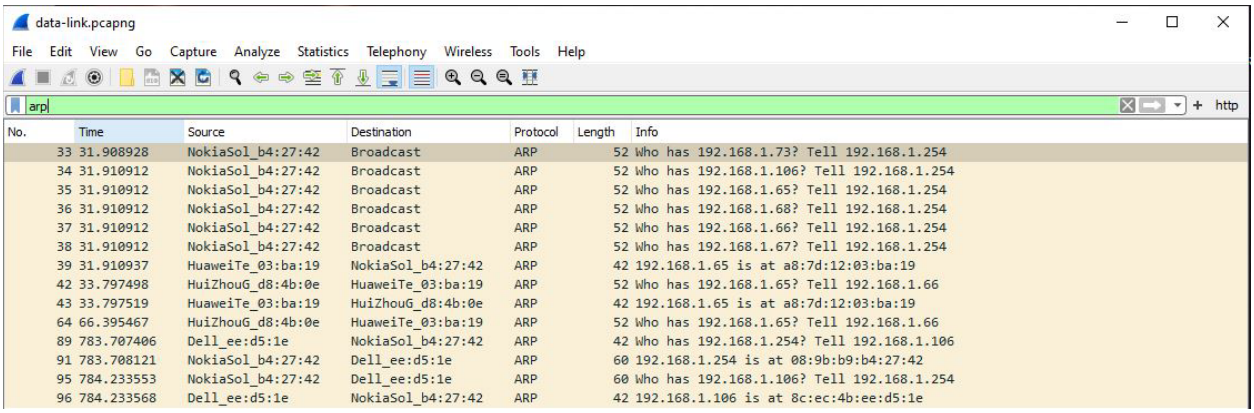

Figure 7: data-link.pcapng file provided by instructor, "arp" filter was applied.

Step 3 (Breakpoint 3)

I decided the analyze the ARP request and ARP reply of user device `NokiaSol_b4:27:42`.

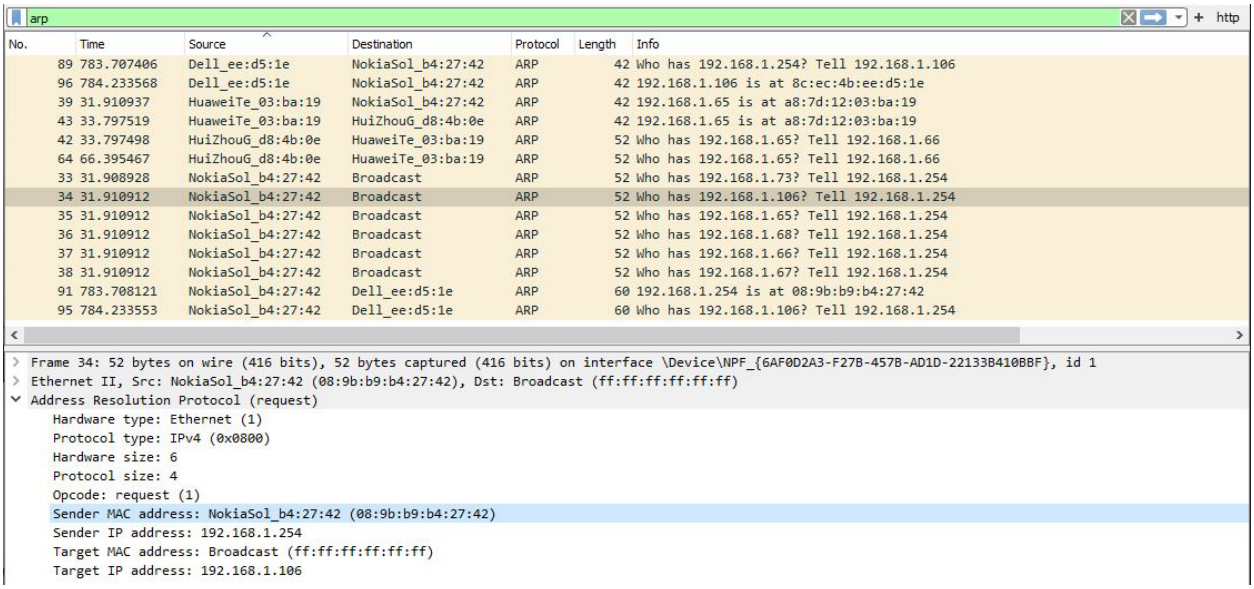

Figure 8: data-link.pcapng file filtered by "arp" in Wireshark, packet details pane depicts ARP payload of `NokiaSol_b4:27:42`.

The Request (Figure 8)

- Hardware Type: `Ethernet (1)`
- Protocol Type: `IPv4 (0x0800)`
- Sender MAC address: `NokiaSol_b4:27:42 (08:9b:b9:b4:27:42)`
- Sender IP address: `192.168.1.254`
- Target MAC address: `Broadcast (ff:ff:ff:ff:ff:ff)`

- Target IP: `192.168.1.106`
- Opcode: `request (1)`

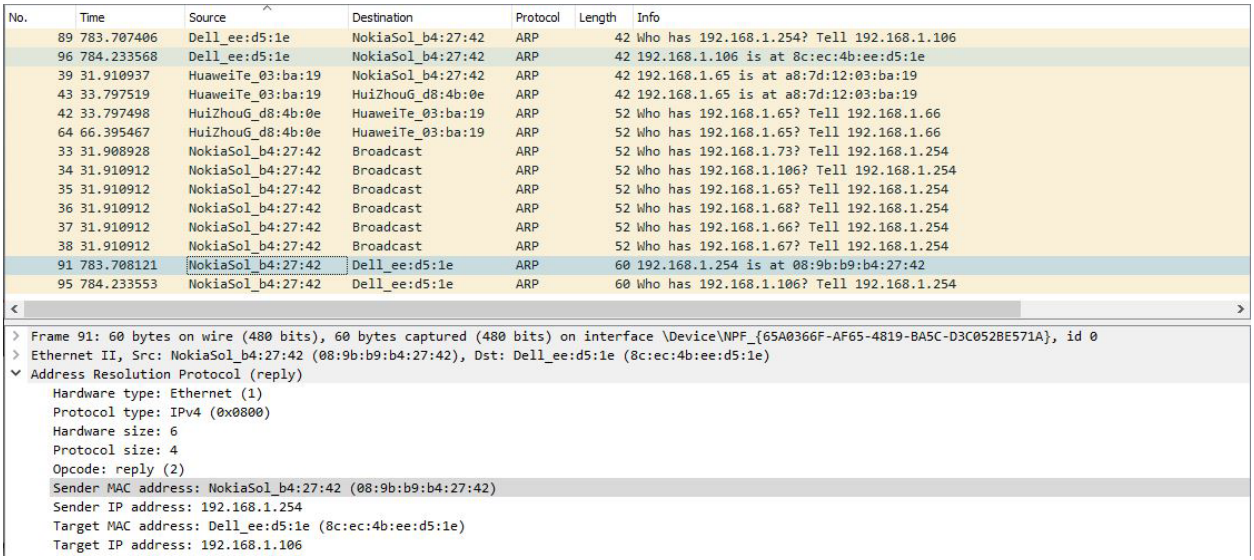

Figure 9: NokiaSol_b4:27:42 reply with payload of ARP packet depicting in packet details pane.

The Reply

- Hardware Type: `Ethernet (1)`
- Protocol Type: `IPv4 (0x0800)`
- Sender MAC address: `NokiaSol_b4:27:42 (08:9b:b9:b4:27:42)`
- Sender IP address: `192.168.1.254`
- Target MAC address: `Dell_ee:df:1e (8c:ec:4b:ee:d5:1e)`
- Target IP: `192.168.1.106`
- Opcode: `reply (2)`

User's device (`NokiaSol_b4:27:42`) made a request looking/searching for IP `192.168.1.106`. From my understanding, on every network there should only be one device using a specific address. `NokiaSol`'s request was sent as a Broadcast, as indicated in the Target MAC address, to everything in the network and the server who owns `192.168.1.106` responded back and included its Layer 2 information (its MAC address, its Ethernet address) so that `NokiaSol` can learn that information. In this case, device `Dell_ee:d5:1e` responded back confirming it had that IP and validated the Layer 2 information. Therefore, `NokiaSol` can send a packet with all the aforementioned including the source and destination layer to the address and this frame of data can be forwarded to the correct destination on the local network.

LIMITATIONS/CONCLUSION

As an introductory experiment for the novice user like myself, I thought the lab's difficulty was fairly simple. I do not think there were any limitations because everything was executed in a live environment versus a controlled environment, like on a virtual machine. Moreover, there were plenty of online resources to guide the user. One of the biggest takeaways from this lab was learning how to interpret

ARP packets via Wireshark, mainly understanding how to analyze an ARP request and ARP reply. Further, I developed a better understanding of how IP and MAC addresses work in relation to the Data Link and Network link respectively.

REFERENCES

[1] Barker, *YouTube* [Online]. “How Address Resolution Protocol (ARP) Works”, July 2, 2019. Available: https://www.youtube.com/watch?v=Cx7foWGm5fo [Accessed: 23-Sep-2022]

COLLABORATION

The entirety of this lab was executed independently by the author. No additional collaboration to report.