Lab 01: ARP

Dillon McDaniel, Adam Stammer

(Dated: February 11, 2020)

Abstract

Using a Windows 10 PC and a Raspberry Pi, we experimented with multiple network configurations and used ARP requests to gain a better understanding of the traffic and shape of the lab network. These experiments began with IPv4 configurations on both the same and different networks from our destination hosts, and moved on to similar experiments with IPv6.

I. PURPOSE

The purpose of this lab is to get an understanding of basic network topology as well as ARP protocols such as:

- Routing to a host on the same network
- Routing to a host on a different network
- Host Specific Routing
- Development and configuration of the ARP cache

Through hands on experimentation this lab will give us experience and a better understanding of these topics. This lab will also give valuable experience working in the lab that we will use in the future, and the systems that compromise said lab.

II. EQUIPMENT/SUPPLIES

Necessary equipment was supplied as components of the lab room itself. Used equipment includes the following:

- Windows 10 Desktop PC with Windows Powershell Administrative Privileges
- Raspberry Pi with root terminal access
- Network access for both systems with additional connected hosts as required by the experiments denoted below

III. METHODOLOGY

First, we ensured our Windows machine was correctly configured to be on the 199.17.161.0 network by running the ipconfig command. Once we did, we sent ICMP echo request to 3 different IP addresses. One of which was on the same logical and physical network, one that was on the same physical but different logical network, and one that was on a different network altogether. We got a response from all three with their routing entries logged in the ARP cache. The one on the same network as us gave us the MAC address of a Microsoft

Ethernet card while the off-network ones returned the MAC address of the Cisco router that was on the network, as all traffic was to be routed through that interface first.

There is at least one switch between our systems and the router that showed up in our ARP cache but the switch itself never showed in the cache. This is because the switch is a layer 2 device without a MAC address, at least for routing purposes.

After that we re-configured our device to connect to the 199.17.162.0 network. We then sent out echo requests to the same 3 addresses plus an additional off-network address. And same as before we got results showing the router from off-network address and one response from the Raspberry Pi that was connected to the network.

We then pinged a few nodes off WSU's network and got the router in the ARP cache. This was expected as all network traffic to these nodes would first be routed to the router of our network. When we pinged hosts that we believed to not exist, the ping eventually timed out and resulted in a "host unreachable" message.

While still connected to the 162 network we added a host specific route for a node on the 199.17.161.0 network. After the node was pinged, the ARP cache showed a direct connection, instead of routing through the router. We then added a host specific route for the entire 161.0 class C network, allowing us to directly route to any node on 161.0 network.

For the next section of the lab we turned on the IPv6 capabilities of our device. After we were properly configured, we pinged the Raspberry Pi using its IPv6 address and its link local address.

We then ran similar experiments on the Raspberry Pi to see if the operating system behaved similarly. We pinged the windows machine and examined both ARP caches to see that the Raspberry Pi appeared in the Windows ARP cache. We then pinged the Pi to see that the reverse was true as well.

IV. RAW RESULTS

- Pinging from the 199.17.161 net id
    1. 199.17.161.32, ARP cache: 199.17.161.32
    2. 199.17.161.24, ARP cache: 199.17.161.24
    3. 199.17.166.186, ARP cache: 199.17.161.1
    4. 199.17.162.20, ARP cache: 199.17.161.1
- Pinging from 199.17.162 net id
    1. 199.17.161.32, ARP cache: 199.17.162.1
    2. 199.17.161.24, ARP cache: 199.17.162.1
    3. 199.17.166.186, ARP cache: 199.17.162.1
    4. 199.17.162.20, ARP cache: 199.17.162.20
- Pinging from 199.17.162 net id with a host specific route to 199.17.161.132
    1. 199.17.161.32, ARP cache: 199.17.161.32
    2. 199.17.161.24, ARP cache: 199.17.162.1
- Pinging from 199.17.162 net id with a host specific route to net id 199.17.161.0
    1. 199.17.161.32, ARP cache: 199.17.161.32
    2. 199.17.161.24, ARP cache: 199.17.161.24
- IPv6 Interfaces
    1. 1:Loopback
    2. 17:Ethernet
- Pingin Windows machine from Raspberry Pi and vice versa on the same logical network
    1. Windows ARP cache when being pinged by Pi: 199.17.162.32
    2. Raspberry Pi ARP cache when being pinged by Windows: 199.17.162.174

- Piging Windows machine from Raspberry Pi and vice versa on different logical networks

    1. Windows ARP cache when being pinged by Pi: 199.17.161.1
    2. Raspberry Pi ARP cache when being pinged by Windows: 199.17.161.1

V. ANALYSIS

With the results shown in this lab we can determine several things about how the network might route traffic.

- If the source address and the destination address have the same net id, they will communicate directly, saving each other's MAC address in their ARP caches.
- If they do not share the same net id the host will send it to the default gateway of the network. It will then save that routing in its ARP cache.
- However, if the host has a host specific route for a particular address or net address it will always send packets on that route. That routing will then be listed in the ARP cache for that entry.
- When a host receives packets from a source address it will save the were it received the packets from as its default for routing to the source address.

These statements all assume the destination host is reachable.

VI. LESSONS AND OTHER OBSERVATIONS

Windows and Linux differ significantly in their display of the ARP cache and routing tables. It proved a noticeable hindrance in this lab's experiments and may foreshadow some of the challenges in the future.

We also noticed that proper configuration is key in the success of these experiments, especially when comparing results to others. This may also prove to be a challenge in the future, but now we know to stay vigilant.

Unexpected results were also very hard to diagnose due to the multiple layers of systems in which such results could form. It could be a configuration issue on our systems, or that of other network hosts, some of which are entirely out of our control. There may be unexpected interactions between our own systems and that of our peers. It can be difficult to solve a problem when you first have to find where it came from in the first place. This wasn't a significant issue in this lab, but we suspect it may grow as a concern as the labs become more complicated.