

ECE 5600 Project (Phase 2)

OBJECTIVE

1. Familiar with the structure of Ethernet frame.
2. Understand the ARP mechanism.
3. Know how to use Wireshark to analyze the captured frame, and locate the MAC addresses and IP addresses.
4. Construct the ARP cache.

BACKGROUND

Address resolution protocol (ARP) belongs to the MAC sublayer of the data-link layer. For the purposes of this project, it provides a single service to upper layers, namely to send a frame to a node on the local network whose protocol address (in our case, IP address) is known. Layers above ARP are blissfully unaware of what MAC addresses mean or that they even exist.

This is not as easy as it might seem. If ARP does not already know what MAC address goes with the specified IP address, it must broadcast an ARP request. Once the reply comes back, it must immediately send the frame. Care must be taken that in-coming packets are handled (at all protocol layers) while awaiting the ARP reply.

PRE-LAB READING

Chapter 5. P442~469

PROJECT PROCEDURE

1. Write a code loop whose job is to get packets from frameio and dispatch them to their appropriate protocol stack. We will only be using two protocols in this project: 0x800 (IP) and 0x806 (ARP). You may dispatch frames in any way you please (function call, message queue, etc.), but this loop will need to run as its own thread (I recommend you use pthread, you will need to add `-lpthread` to your compile line) **Hint:** see [example1.cpp](#) for details. You can use [example1.cpp](#) as a template to start your project.

2. Write codes to respond to ARP requests. When your ARP implementation receives an ARP request and its destination IP address is your computer's, your machine must immediately send an ARP reply. The ARP reply must use the correct destination MAC address (i.e. not broadcast). In wireshark, you will see an ARP reply from the machine and a duplicate one generated from your code.

3. Test your code by pinging your IP address from a different computer (Use arping). Verify with wireshark that your protocol stack has responded. You should see the ARP request, your reply (duplicated, one is from your

code, the other is from the machine itself), and ICMP frames on the wire. Copy & paste the ARP request, your ARP response to a file.

Attention: It will broadcast ARP request and you will observe two ARP replies: one is from machine automatically and the other is from your code.

4. Implement a cache mechanism to facilitate Layer 2 communication. Specifically, upon receiving any ARP frame (even not for you) in your code, cache its source MAC and IP address.

5. Now, write codes to send an ARP frame with known IP address . If the IP address is already in the cache, compose the frame and send it immediately.

Otherwise, send an ARP broadcast and get the ARP reply, cache the IP and MAC address pair. then send the frame now has known MAC addresses. (Recall that we need MAC address in layer 2 to communicate.) Again copy & paste your request and the other computer's response into the file.

Hints for this project can be found in canvas

REPORT REQUIREMENTS

1. Requirements for each procedure are listed in the hint file, please follow the instructions. Make sure your screenshots are clear in the reports otherwise please attach the file in your submissions.
2. Even if you work in groups, please submit your own lab report, the code can be same.

3. Project due; October 21, 2016