Hints for Project 2

Haijian Sun

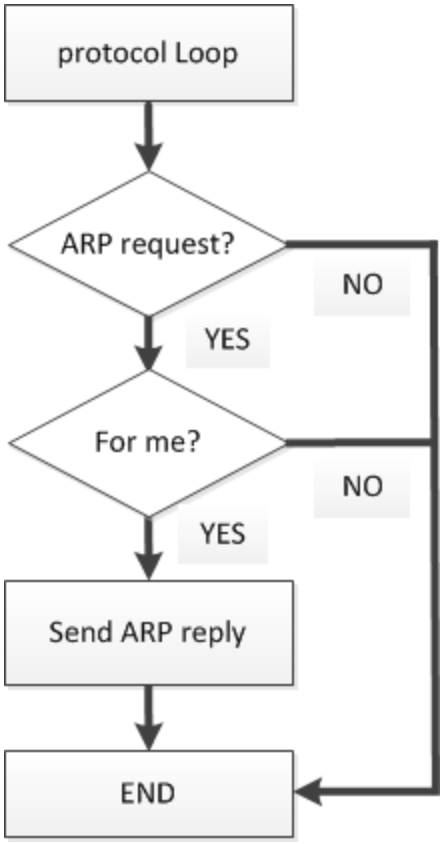
UtahState University

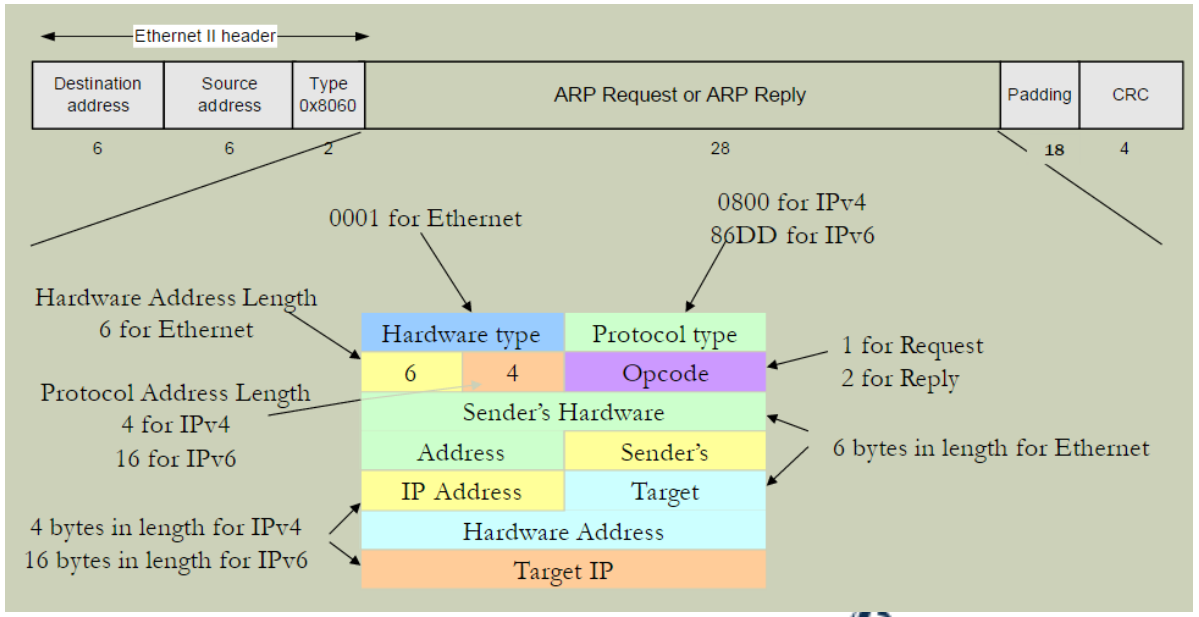
Procedure 1-3:

- Procedure 1 is already done in the example code (protocol loop function) , it checks the buf.prot to see if it is an IP or ARP protocol. Then dispatches the packet either to ip_protocol_loop or arp_protocol_loop function.
- What you need to do is checking the Opcode in the arp_protocol_loop function to see if the incoming frame is a request or reply. If it is a request, go on checking if its target IP address is the same as your machine's. If true, it means that the request is for your computer.
- Then, you need to compose an ARP reply frame, you can do it by looking at a regular reply frame made by the computer from wireshark. (arping your IP address from other machine and see the reply) Make sure you know what each field means.

Procedure 1-3

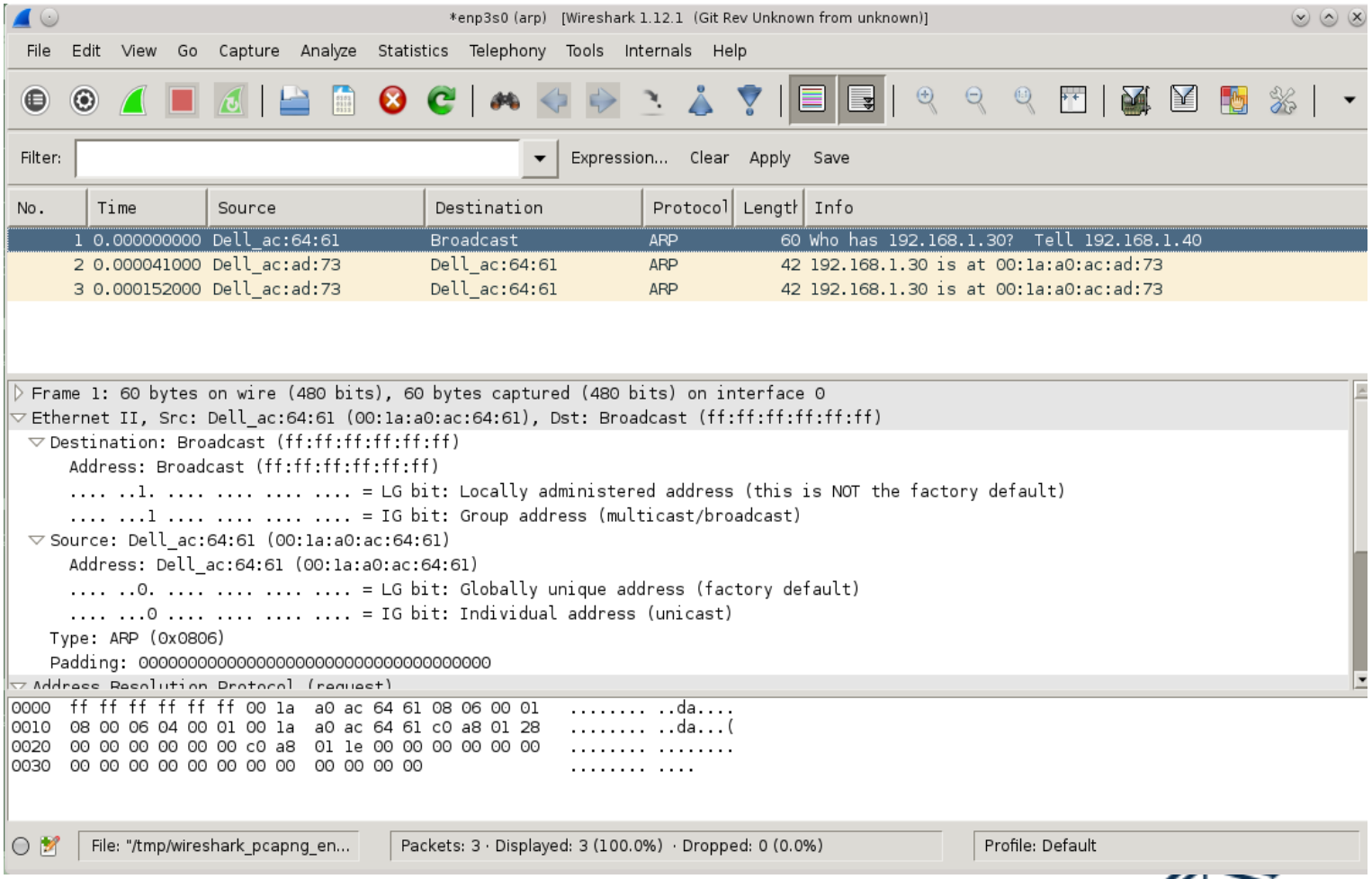
- Write codes to respond to an ARP request.
- A brief flow chart.





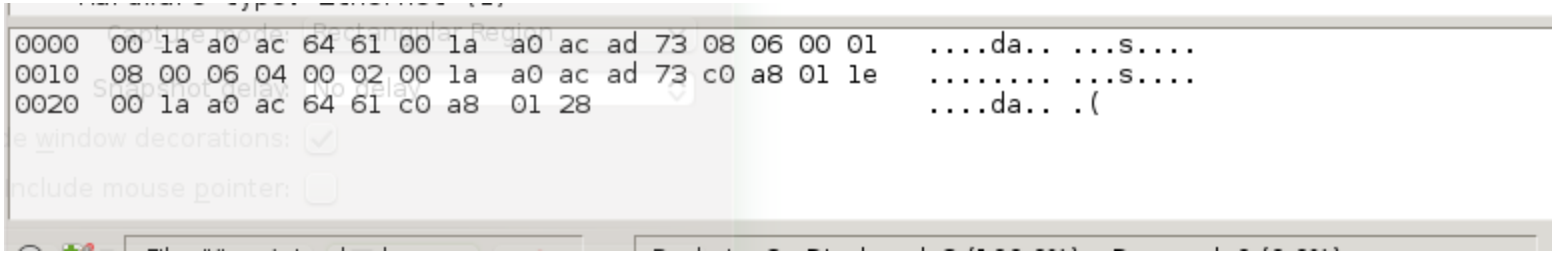
UtahState University

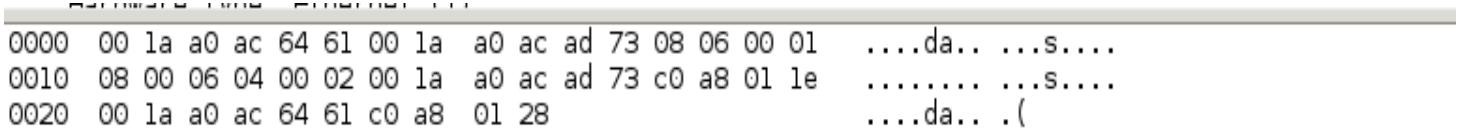
Screenshots:



Utah State University

Screenshots:



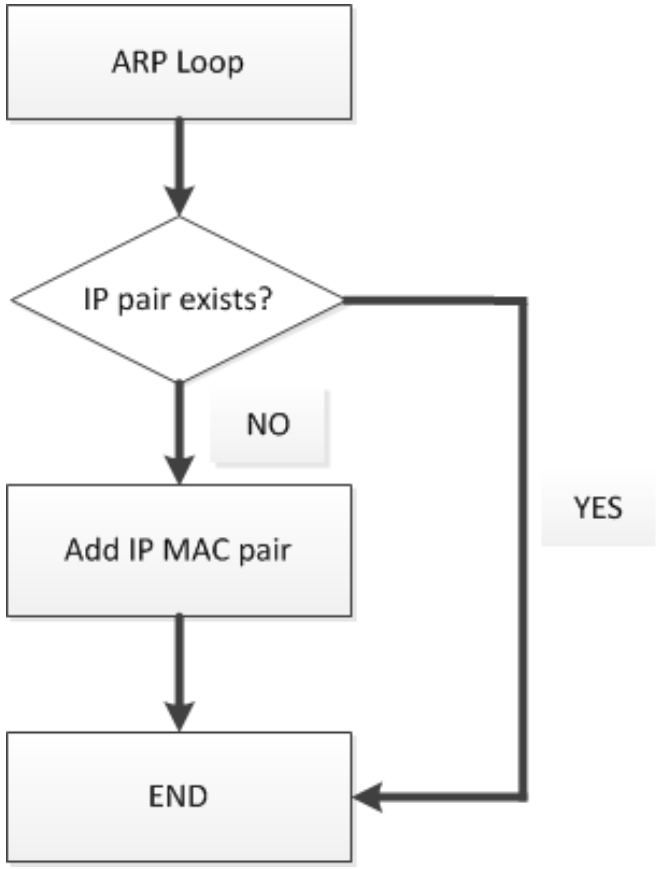


- Except for the screenshots, please also copy and paste the contents of your reply to the lab report.
- Also, pay attention of the broadcast frame, it can be useful later.

Utah State University

Procedure 4-5:

- First, set up a cache function to store the IP-MAC pair in the code.
- This cache function will store any ip-mac pair even the frame is not for you.



- One possible way is to use struct, like the ether_frame.
  - Struct ipmac

```
{
    octet ip[4];
    octet mac[6];
};
```

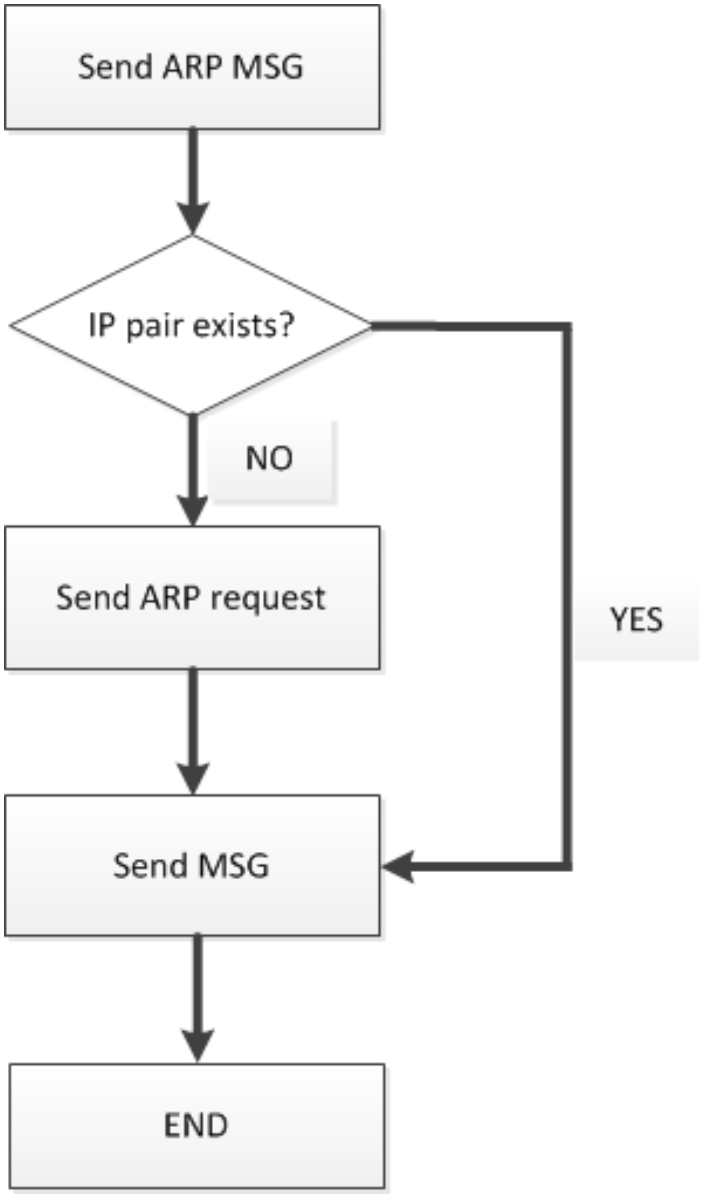
Procedure 4-5:

- Basically, what you need to do in procedure 4 is to send an ethernet frame. For consistency, you will send an ARP message to the destination with known IP address. (The IP address either comes from input from console or pre-defined in the code, depends on the implementation detail of your own codes).

- With only known IP address, you can't send a frame in layer 2 protocol. Because you need to know the corresponding MAC address, since we have IP-MAC cache, first, you should look up that table. If the pair exists, compose an ARP frame and send directly, if not, then send ARP broadcast first(again, check wireshark to see the details of a broadcast frame), when the target machine replies, cache the IP-MAC pair, then send the frame(So, next time ,if you send to the same IP address, you just need to check ip-mac table).

- For simplicity, the ARP msg can also be the ARP reply in procedure 3.

Procedure 4-5:



- In the report, show me two scenarios:
  - IP-MAC does not exist, your code will send an ARP broadcast, then the target machine replies, then your code add this pair, sends the ARP msg directly, the whole procedure should be highlighted in wireshark.
  - IP-MAC exists, ARP msg sends directly, also screenshots the wireshark.

Utah State University

Extra Credit part

- If you feel good about procedure 1-5, consider to add a timer about the ip-mac cache, which is also the mechanism in our computer. Specifically, when adding a new pair, set up a timer, when the time expires (say, 20 seconds), delete the pair.
- In the following page, I list one picture which can explain the procedure 4-5, and also the extra credit part.
  - Always highlight the important results!
  - If you have further concerns, please send me the email.

An Example

```
netlab30:/home/student/Documents/Pro2 # ./phase2
adding IP MAC pair: 192 168 1 30
adding IP MAC pair: 192 168 1 40
p 192 168 1 40
Try to send IP msg
IP is in cache
IP MSG sent
delete expires ip mac pair 192 168 1 30

adding IP MAC pair: 192 168 1 1
p 192 168 1 40
Try to send IP msg
IP is in cache
IP MSG sent
adding IP MAC pair: 192 168 1 30
ip mac pair already exists 192 168 1 40
ip mac pair already exists 192 168 1 1
ip mac pair already exists 192 168 1 30
ip mac pair already exists 192 168 1 1
p 192 168 1 40
Try to send IP msg
IP is in cache
IP MSG sent
delete expires ip mac pair 192 168 1 1

adding IP MAC pair: 192 168 1 1
p 192 168ip mac pair already exists 192 168 1 30
ip mac pair already exists 192 168 1 1
p 192 168 1 delete expires ip mac pair 192 168 1 30

delete expires ip mac pair 192 168 1 40

p 192 168 1 40
Try to send IP msg
IP is not in cache, send arp broadcast first
```

- When I run my code, it listens to every traffic and adds ip-mac pair if doesn't exist (if exists, it will output ip mac pair already exists).
- My code always waits for my input from console, p 192 168 1 40 is the command to send ARP msg (Not IP msg in the picture) to the destination 192.168.1.40.
- As you can see 192.168.1.40 is in cache, so it will compose an ARP frame with known mac address directly sent.
- If you try to send to 192.168.1.50, it's not in cache, so send ARP broadcast first, waits for the reply, then adds to the cache, and send ARP msg.
- This code has a timer for the cache, after certain time, 192.168.1.40 is deleted from cache, so if I still try to send msg to this address, it will send ARP broadcast first, waits for the reply, then adds to the cache again, and then send ARP msg.