# Instruction for project 3

Hi, all

Here is some guidance and hints for project 3, I will explain what you are supposed to do and how step by step. If you check from Wireshark, you can see what a “ping” command do is to get target MAC via ARP and then keep sending ICMP echo request (reply from target). We have finished ARP in project 2, in project 3 we will implement ICMP part, then you can simulate exactly what a “ping” command did.

Step 1

This step just require you to identify a ICMP packet. You can print out “ICMP packet received” after your code detecting a ICMP packet. To find a ICMP packet from Wireshark, still ping your PC from another one. Then locate a ICMP packet on Wireshark and check the explains of each component, you can find there is a byte denoting it is a ICMP and by checking that byte in your code you can detect the ICMP packet.

For grading, I need a screenshot of you terminal printing out something like “ICMP packet received”

Step 2

In project 2, the packet you send after ARP could have any content as long as ethernet header is correct. Now you need to make a IP header for the frame you want to send. Also referring Wireshark to see how the IP header constructed. Notice that when calculating the checksum in IP header, you only need to add the content of IP header together excluding Ethernet header and data.

Step 2 also need you to make a IP mask. Which means in your code it will check whether the target IP is in the lab, if it is, your code will send ARP to it and get the MAC; if not, your code will send ARP to router (192.168.1.1) and get its MAC, then enclose it to the ethernet header of the frame you want to send. So finally your code should work like this:

**1.Let you type in the target IP**
**2.Get MAC through ARP using your previous code, if the target in the lab, then the ARP request is send to target directly, otherwise the ARP request is send to the router.**
**3.After MAC acquired, construct IP header and send out the packet, the packet sent out should be like this**

**Target IP in lab**

Ethernet header: MAC of target
IP header: IP of target
Data
**Target IP not in the lab**

Ethernet header: MAC of router
IP header: IP of target
Data

For Data part you can put in anything to show this is the packet you made.

For grading, I will check at least two screen shot of Wireshark, one is for the case target IP is in the lab, the other one is for target IP not in the lab. Please show me the content of the packet in both case. Make sure it can be print out clearly on your report or you can copy past it in text in your report.

Step 3

This step require you to implement ICMP protocol for echo reply and request. You need to make ICMP request and reply and test them. ICMP has its own checksum and calculated after IP header. The sequence number increase by 1 for each request and reply pair, you can find it from Wireshark. **You need to show me your sequence number changing by each pair of echo**. To do this, show me at least two pair of request and reply and high light the difference. Or you can show me this in step 5. Checksum need to be recalculated for each request and reply. If the checksum is not calculated correctly, the packet will be report as error in Wireshark. You can program based on your code for step 2 and it should work in this way:

**1.Let you type in the target IP**
**2.Get MAC through ARP using your previous code, if the target in the lab, then the ARP request is send to target directly, otherwise the ARP request is send to the router.**
**3.After MAC acquired, keep sending ICMP request and get echo reply from your target.**

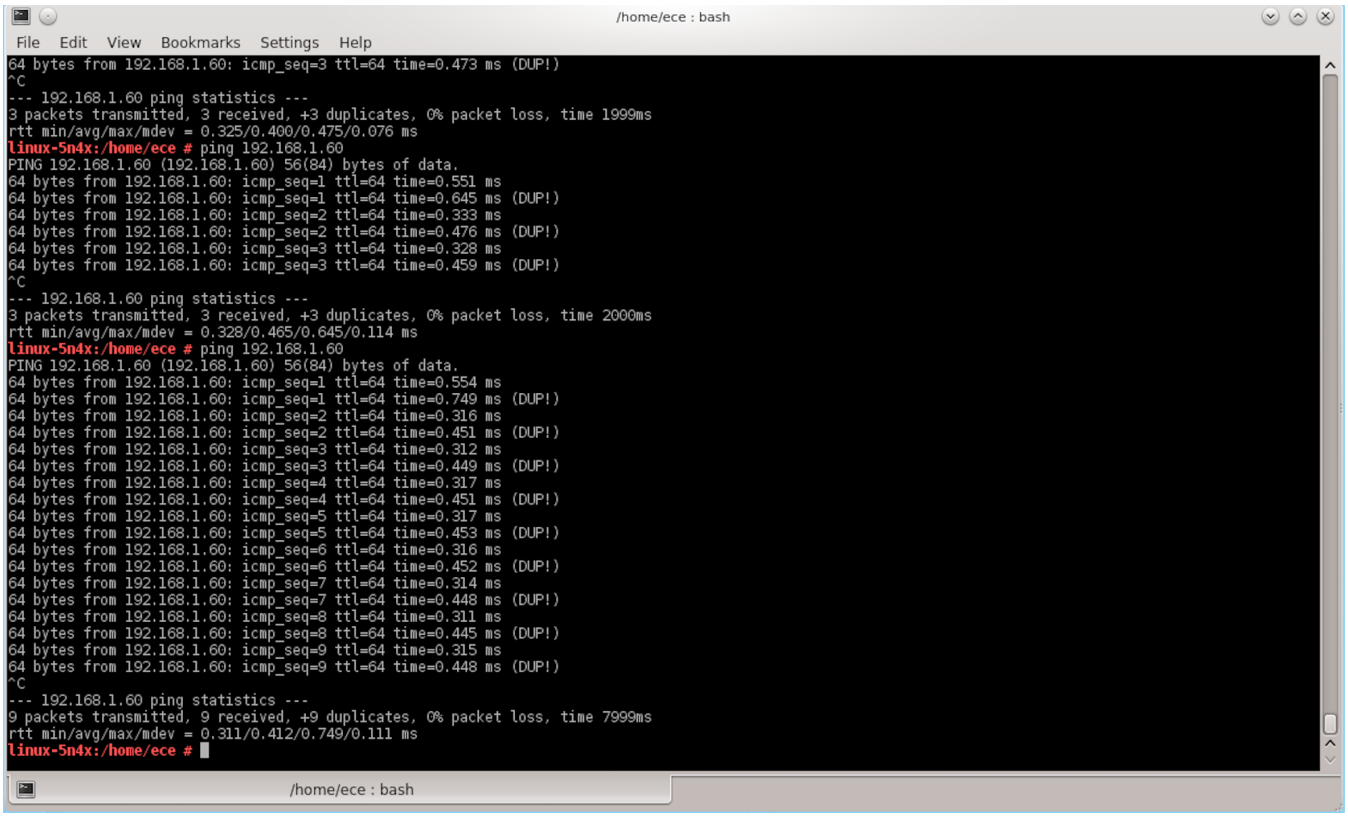
For grading, I will check screen shot of echo request and reply you made. Like in project 2, you should be able to receive two reply, one from your machine, the other one from your code. Show me the content of each packet, so the screen shots can be arranged in the way below. **In this step you don't need to consider the case that target is not in the lab**

Echo request you made
Reply by machine
Reply by your code

Step 4.

Test your ICMP echo reply, if your your code is correct, you can see this on the terminal of PC "ping"ing your PC. Messages end with (DUP!) means your reply is received along with the one send by machine just like the duplicated ARP reply you can see from Wireshark before.

For grading, I will check the screen shot like below or you can copy past it from terminal like said in document.




Step 5

Use your echo request to “ping” a IP address out of lab, of course you need to get router’s MAC through ARP at first. You can input “ping www.google.com” in terminal to get google’s IP address and set it as your target.

For grading, I will check the screen shot of at least two pair of request and reply, show me the content for each packet. If you didn’t show the changing of sequence number in step 3, you need to do it here. But if you did, then don’t need to do this again.