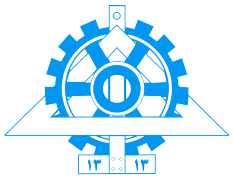
In the name of Allah

بسم الله الرحمن الرحیم

Linux and TCP/IP networking Laboratory Manual



University of Tehran
دانشگاه تهران

School of Electrical and Computer Engineering
دانشکده مهندسی برق و کامپیوتر

Computer Network Lab
آزمایشگاه شبکه‌های کامپیوتری

Dr. Ahmad Khonsari - احمد خونساری
a_khonsari@ut.ac.ir

Amir Haji Ali Khamseh'i - امیر حاجی علی خمسه‌ء
khamse@ut.ac.ir

Muhammad Borhani - محمد برهانی
Amirahmad Khordadi - امیر احمد خردادی
Sina Kashi pazha - سینا کاشی پزها
Mohammad Ali Shahsavand - محمد علی شاهسوند

March 11, 2019

۲۰ اسفند ۱۳۹۷

Network interface exercises

The following exercises use the single segment network topology shown in Fig. 1.3.

1 Exercise 1

Use the **ifconfig -a** command to display information about the network interfaces on your host. Find the IP address and the net mask of your machine.

Report

How many interfaces does the host have? List all the interfaces found, give their names, and explain their functions briefly.
What are the MTUs of the interfaces on your host?
Is network subnetted? What is the reasoning for your answer? What the experimental are the reasons for subnetting?

2 Exercise 2

While **tcpdump host** *your-host* is running in one command window, run **ping 127.0.0.1** from another command window.

Report

From the **ping** output, is the 127.0.0.1 interface on? Can you see any ICMP message sent from your host in the **tcpdump** output? Why?

3 Exercise 3

By using **netstat -in** command, collect the statistics from all the hosts on the network. Since we use the same login name and password, we can **telnet** to other workstations and run **netstat -in** there. [1]
Save the **netstat -in** outputs.
If you don't see a significant amount of output packets in the **netstat** output, the machine was probably restarted recently. You may do this experiment later, or use the following **sock** command to generate some network traffic:
sock -u -i -n200 *remote-host* **echo**

Report

Calculate the average collision rate over all the hosts for the set of statistics you collected in this exercise.

ARP exercises

In the following experiment, we shall examine the host ARP table and the ARP operation, including two interesting cases: proxy ARP and gratuitous ARP. You may need to ask the lab instructor for the MAC addresses of the host and router interfaces, and record these MAC addresses in Table A.1 and Table A.2 in the appendix. You need these MAC addresses for the exercises and lab report.

[1] After you are done with a remote host, you should exit the **telnet** session before you **telnet** to another remote host. Recursive **telnet** will generate unnecessary data in the **tcpdump** output and cause confusion.

4 Exercise 4

Use **arp -a** to see the entire ARP table. Observe that all the IP addresses displayed are on the same subnet.
If you find that all the remote hosts are in your host's ARP table, you need to delete a remote host (not your workstation) from the table, using,
arp -d *remote-host*. [2]
Save the ARP table for your lab report.
While **tcpdump -enx -w exe2.out** is running, **ping** a remote host that has no entry in your host ARP table. Then terminate the **tcpdump** program.
Next, run **wireshark -r exe2.out&** to load the **tcpdump** trace file.
Observe the first few lines of the packet trace to see how ARP is used to resolve an IP address.
Run **arp -a** to see a new line added in your host's ARP table. Save the new ARP table for your lab report.
Mark the ARP request packet and the ARP reply packet in the **wireshark** window. Then go to menu `File/Print ...` to print the marked packets for your lab report (See Exercise 6 of Chapter 1).

Report

From the saved **tcpdump** output, explain how ARP operates. Draw the format of a captured, ARP request and reply including each field and the value.
Your report should include the answers for the following questions.

- What is the target IP address in the ARP request?
- At the MAC layer, what is the destination Ethernet address of the frame carrying the ARP request?
- What is the `frame` type field in the Ethernet frame?
- Who sends the ARP reply?

5 Exercise 5

While **tcpdump host** *your-host* is running to capture traffic from your machine, execute **telnet 128.238.66.200**. Note there is no host with this IP address in the current configuration of the lab network.
Save the **tcpdump** output of the first few packets for the lab report.
After getting the necessary output, terminate the **telnet** session.

Report

From the saved **tcpdump** output, describe how the ARP timeout and retransmission were performed. How many attemps were made to resolve a non-existing IP address?

6 Exercise 6

The network topology for this proxy ARP exercise is shown in Fig. 2.9. We will divide the group into two subnets interconnected by a router. The IP addresses and network masks for the hosts are also given in Fig. 2.9. Change the IP address and network mask of your host accordingly (see Section 2.3.2). The IP addresses and network masks of the `Router4` interfaces are the same as their default settings. Note that the network mask of the hosts in the 128.238.65.0 network is 255.255.0.0.
Next we will enable the proxy ARP function on the ethernet1 interface of `Router4`.

1. **telnet** to `Router4` from `shakti`: **telnet 128.238.64.4**. The login password is `el537`. [3]

[2] If you deleted your workstation's IP address from the ARP table by mistake, you must add the entry back in the table. See the **arp** manual page to add. Note that, in order for your workstation to reply to the ARP requests, the ARP entry of your workstation must have the **P** flag in the ARP table.
[3] Check with your lab instructor for the password of the router you are using, which may be different from `el537`.

2. Log in to the router, type **enable** to enter the *Privileged EXEC* mode.[4] The password is again `el537`.
3. Enter the *Global Configuration* mode by typing **config term**.
4. Then type the following lines:
   - **interface ethernet 1**[5]
   - **ip proxy-arp**
   - `Ctrl-Z`
5. Type **exit** to terminate the **telnet** session.

Now `Router4`'s `ethernet1` interface can perform proxy ARP for the hosts in the 128.238.64.0 subnet.
Run **tcpdump -enx** on all the hosts.
Then let the hosts in the 128.238.65.0 subnet send UDP datagrams to the hosts in the 128.238.64.0 subnet. For example, on `guchi` type:
sock -i -u -n1 -w1000 *Host-in-64.0-subnet* **echo**
When you are done with all the hosts in the 128.238.64.0 subnet, save the tcpdump output for the lab report.
Run **arp -a** to display the new ARP table in your host. Save the ARP table for your lab report.
After the lab instructor restores the network into a single subnet (see Fig. 1.3), change the IP address and network mask of your host's interface back to their default values as in Fig. 1.3.
Exchange your data saved in this exercise with a student working in the other subnet.

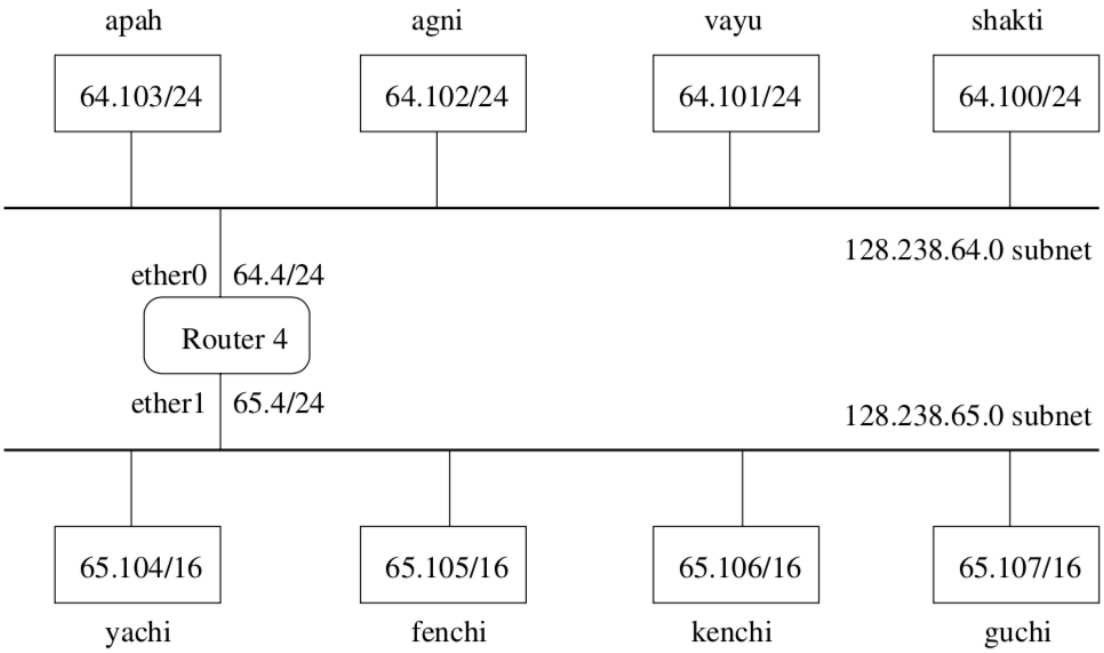


Figure 2.9. Network configuration for the proxy ARP experiment.

Report

Explain the operation of proxy ARP.
Why can a host in the 128.238.65.0 subnet reach a host in the 128.238.64.0 subnet, even though they have different subnet IDs?

[4] We will discuss bridge and router configuration in Chapter3.
[5] The name of the router interfaces may be different for various routers. You can find the names by typing **write term** in the *Privilege EXEC* mode.

What are the MAC addresses corresponding to hosts in the 128.238.64.0 subnet, in the ARP table of a host in the 128.238.65.0 subnet?
Give one advantage and one disadvantage of using proxy ARP.

7 Exercise 7

This exercise will be performed by all the students together. While **tcpdump -ex -w exe7.out** is running on all the hosts, reboot host `guchi`.
After `guchi` is started, terminate **tcpdump** and run **wireshark -r exe7.out &** to load the **tcpdump** trace. Print the the gratuitous ARP request for your lab report.

Report

What is the purpose of gratuitous ARP?
List the sender IP address, target IP address, sender MAC address, and target MAC address of the gratuitous ARP you saved.

Exercise with ICMP and Ping

8 Exercise 8

Use **ping -sv** *remote-host* to test whether the remote host is reachable, while running: **tcpdump -enx host** *your-host* **and** *remote-host*. Save the **tcpdump** and **ping** output for the future study on **ping**.

Report

What ICMP messages are used by **ping**?

9 Exercise 9

While running **tcpdump -x -s 70 host** *your-host* **and** *remote-host*, execute the following **sock** command to send a UDP datagram to the remote host: **sock -i -u -n1 -w1000** *remote-host* **88888**.
Save the **tcpdump** output for the lab report.

Report

Study the saved ICMP port unreachable error message (see Fig. 2.7). Why are the first 8 bytes of the original IP datagram payload included in the ICMP message?

10 Exercise 10

While **tcpdump** is running to capture the ICMP messages, **ping** a host with IP address 128.238.60.100. Save the **ping** output.

Report

Can you see any traffic sent on the network? Why? Explain what happened from the **ping** output.
List the different ICMP messages you captured in Exercises 8, 9, and 10 (if any). Give the values of the type and code fields.

Exercises with IP address and subnet mask

In this section, we will observe what happens when the same IP address is assigned to two different hosts. We will also set an incorrect subnet mask for hosts and see what are the consequences. For the next two exercises, we split the current single segment network into two segments, Group A and Group B as shown in Table 2.3, so that they will not interfere with each other.

Table 2.3. *Host IP addresses and network masks for exercise 11*

Group	Name	IP address	Subnet mask
Group A	`shakti`	128.238.66.100	255.255.255.0
	`vayu`	128.238.66.100	255.255.255.0
	`agni`	128.238.66.102	255.255.255.0
	`apah`	128.238.66.103	255.255.255.0
Group B	`yachi`	128.238.66.104	255.255.255.0
	`fenchi`	128.238.66.104	255.255.255.0
	`kenchi`	128.238.66.106	255.255.255.0
	`guchi`	128.238.66.107	255.255.255.0

11 Exercise 11

Change the IP address of your workstation as shown in Table 2.3.
Delete the entries for all hosts other than your own workstation from your workstation's ARP table.
Run **tcpdump -enx** on all the hosts. Then, do the following three experiments.

1. Execute **telnet** from one of two hosts with the duplicate IP address to a host with unique IP address (e.g. `shakti` –> `agni` in Group A and `yachi` –> `kenchi` in Group B).
   Now, from the other host with the duplicate IP address, execute **telnet** command to the same host (`vayu` –> `agni` or `fenchi` –> `kenchi`).
   Observe what happens and save the **tcpdump** output and the ARP tables in all the hosts in your group.
2. Execute **telnet 128.238.66.100** (or **128.238.66.104**) from `agni` (or `kenchi`). Which host provides the telnet connection? Why?
3. Execute **telnet 128.238.66.100** (or **128.238.66.104**)from `apah` (or `guchi`). Which host is connected to `apah` (or `guchi`)? Why?

Report

Explain what happened in the first case and why. Answer the questions for the second and third cases.

12 Exercise 12

Change the host IP addresses and the subnet masks as shown in Table 2.4. Since we still have two separate segments, Groups A and B can do the exercise independently. Note that two hosts in each group (`shakti` and `apah` in Group A, or `yachi` and `guchi` in Group B) are assigned an incorrect subnet mask.

Capture the packets with **tcpdump -e** for the following cases.

1. When `shakti` (`yachi`) **ping**s one of the hosts that have the correct subnet mask.
2. When `apah` (`guchi`) **ping**s one of the hosts that have the correct subnet mask.
   Now, copy the output displayed from the **ping** window in `apah` (`guchi`). Share the saved output message with other students.
3. When a host with the correct subnet mask **ping**s `shakti` (`yachi`).
4. When a host with the correct subnet mask **ping**s `apah` (`guchi`).

To avoid confusion, only one machine in each group should generate traffic in each case. Clearly, this exercise has to be performed as a team.

Table 2.4. *Host IP addresses and network masks for exercise 12*

Group	Name	IP address	Subnet mask
Group A	`shakti`	128.238.66.100	255.255.255.240
	`vayu`	128.238.66.101	255.255.255.0
	`agni`	128.238.66.102	255.255.255.0
	`apah`	128.238.66.120	255.255.255.240
Group B	`yachi`	128.238.66.104	255.255.255.240
	`fenchi`	128.238.66.105	255.255.255.0
	`kenchi`	128.238.66.106	255.255.255.0
	`guchi`	128.238.66.121	255.255.255.240

Report

Explain what happened in each case according to the **tcpdump** outputs saved. Explain why `apah` (or `guchi` in Group B) could not be reached from other hosts, whereas `shakti` (or `yachi` in Group B), which has the same incorrect subnet mask, could communicate with the other hosts.