بسم الله الرحمن الرحیم

In the name of Allah

The Web, DHCP, NTP and NAT[1]

Laboratory Manual

University of Tehran
School of Electrical and Computer Engineering

دانشگاه تهران
دانشکده‌ی مهندسی برق و کامپیوتر

Computer Network Lab

آزمایشگاه شبکه‌های کامپیوتری

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

Amir Haji Ali Khamseh'i
امیر حاجی‌علی‌خمسه‌ء
khamse@ut.ac.ir

Muhammad Borhani
محمد برهانی
m.borhani@ut.ac.ir

AmirAhmad Khordadi
امیراحمد خردادی
a.a.khordadi@ut.ac.ir

Sina Kashipazha
سینا کاشی‌پزها
sina_kashipazha@ut.ac.ir

Hadi Safari
هادی صفری
hadi.safari@ut.ac.ir

November 14, 2019

۲۳ آبان ۱۳۹۸

[1] S. Panwar, S. Mao, J.-dong Ryoo, and Y. Li, "The Web, DHCP, NTP and NAT," in TCP/IP Essentials: A Lab-Based Approach, Cambridge: Cambridge University Press, 2004, pp. 159–186.

Part I
HTTP Exercises

For the exercises in this section, the network topology is simple with two hosts are connected to a single network segment with IP addresses, i.e. from 128.238.66.100 to 128.238.66.101. Add first host from *http-server* and another hsot from *gui*.

1 HTTP Server

Examine the various configuration directives used and the corresponding settings[1].

Start the Apache server on your host by: `service apache2 start` . In order to check if the server is working properly, you may start a *Mozilla* web browser to download the test page at `http://remote-host/`.

Then, execute `pgrep apache2` to list the process IDs of the `apache2` processes started.[2] Save the output and the configuration file for the lab report.

Report

1. How many `apache2` processes were started? Which one was the master server, and which ones were the child servers? (*use `htop` and switch to* ***tree*** *to see process hierarchy*) Justify your answer using the `apache2.conf` file.
2. What is the purpose of initiating multiple apache2 processes?

2 HTTP Request

Execute `wireshark` to capture packets between your host and a remote host.

Login to the remote host's web server: `telnet remote-host 80` .

In the login console, type the following HTTP request line by line:

```
GET /index.html HTTP/1.0
From: netlab@your-host
User-Agent: HTTPTool/1.0
```

Note that you need to press the *Return* key to input the last line, which is blank. When the `telnet` process is terminated, save the output for your lab report.

Terminate `wireshark` and analyze the captured HTTP packets. Print and save the HTTP request and response.

Save the HTTP response's data part into a file, named `index.html`. Use *Mozilla* to view the file.

Report

1. Submit the HTTP request and response, including the start-lines and all the headers.

3 HTTP `Keep-Alive`

By default, Apache server supports persistent connections. Before this exercise, the lab instructor should check the `KeepAlive` directive in the server configuration file [3] to make sure it is turned on, as `KeepAlive On` .

[1] `/etc/apache2/apache2.conf`

[2] By `a2enmod mpm_prefork` you can enable Multi-Processing Module and change default spare process (config file: `/etc/apache2/mods-available/mpm_prefork.conf`).

[3] `/etc/apache2/apache2.conf`

Execute `tcpdump host your-host and remote-host` or run `wireshark` to capture packets between your host and a remote host.

Enter the URL `http://remote-host/try1.html`, to download the HTML file consisting a line of text, an embedded picture, and a hyperlink.

Disable Apache server persistent connections by change value of `KeepAlive` option to `Off` and restart `apache2` service[4].

Use `tcpdump host your-host and remote-host` or run `wireshark` and print the HTTP requests and responses for the lab report.

Use *Mozilla* to reload the `try1.html` file. Use *Ctrl+F5* to ignore cache.

Use `wireshark` and print the HTTP requests and responses for the lab report.

Report

1. When you browsed the `try1.html` file for the first time, how many HTTP requests were sent? Which files were requested? How many TCP connections were used?
2. Answer the above questions for when you browsed the `try1.html` file for the second time.
3. What is the purpose of using persistent connections?

4 HTTP Submit

Execute `wireshark` to capture packets between your host and a remote host.

Use *Mozilla* to download the `http://remote-host/try2.htm` file, which is an *HTML* form, from the remote host.

Fill a text string, e.g. the name of the host being used, into the text field in the form and click the submit button in the form.

When the server response is received, terminate `wireshark`.

Examine how LAMP (for php)[5] works, and identify the data string sent to the server. Save the HTTP request containing the data string for lab report.

Report

1. Submit the data string sent to the server.

Part II
DHCP Exercises

For the exercises in this section, the network topology is given in Figure 1.3, where all the hosts are connected to a single network segment whitout their default IP addresses.

[4] `service apache2 restart`

[5] or CGI (for perl)

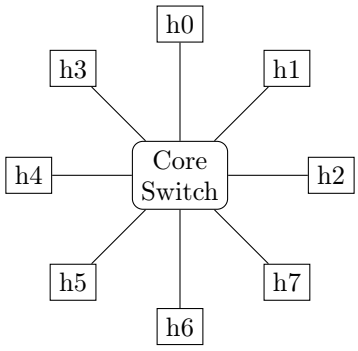

Figure 1: A single segment network (Figure 1.3)

5 DHCP Server

In this exercise, we use *h1* as the DHCP server, with a configuration file[6] shown in Table 8.3. Do the following:

1. Change MAC address of *h5* in server configuration file (Table 8.3) by your *h5* MAC address and save to *dhcpd.conf*.
2. Set IP of *h1* to 128.238.66.100
3. Start the DHCP server on *h1* in the foreground and working in the debugging mode: `dhcpd -d -f` [7].
4. Execute `tcpdump -exn -nn -s 100` or `wireshark` to capture the DHCP messages in the network segment.
5. Then do the following to enable DHCP for the Ethernet interface on *h2*.
   Run `ifconfig eth0 0` on `h2` to clear IP of *eth0*
   And run `dhclient eth0` .
   When *h0* is successfully reconfigured, execute `ifconfig -a` or `ip a` to display its network interface configurations and execute `netstat -rn` or `ip r` to display its routing table.
   Save the outputs for the lab report.
6. Then, repeat 5 for *h3*.
7. Repeat 5 for *h4*.
8. Repeat 5 for *h5*.

Terminate `wireshark` . Print out the DHCP messages for the lab report.

Save the DHCP server output on *h1* for the lab report.

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 128.238.66.255;
option routers 128.238.66.1;
#option domain-name-servers 128.238.2.38, 128.238.3.21;
#option domain-name "netlab.ut.ac.ir";

subnet 128.238.66.0 netmask 255.255.255.0 {
    range 128.238.66.111 128.238.66.112;
}
```

[6] `/etc/dhcp/dhcpd.conf`
[7] default

```
host h5 {
    hardware ethernet 08:00:20:79:e9:9f;
    fixed-address 128.238.66.110;
}
```

Code 1: A DHCP server configuration file (Table 8.3)

Report

1. Compare the DHCP operation captured by `wireshark` and that shown by the DHCP server output. Explain how DHCP works.
2. Did *h2* and *h3* successfully obtain a set of new parameters? Compare the `ifconfig` and `netstat` output with the parameters carried in the corresponding DHCP messages.
3. Answer the above question for *h4*. Explain why *h4* failed.
4. Answer the above question for *h5*. Explain why *h5* succeeded.

Part III

NTP exercises

Before proceeding to the next exercise, reboot the hosts and set network topology is given in Figure 1.3 and set hosts IP address from 128.238.66.100 to 128.238.66.107.

Table 1: The IP addresses of the hosts (Table 1.2)

Host	IP Address	Subnet Mask
h0	128.238.66.100	255.255.255.0
h1	128.238.66.101	255.255.255.0
h2	128.238.66.102	255.255.255.0
h3	128.238.66.103	255.255.255.0
h4	128.238.66.104	255.255.255.0
h5	128.238.66.105	255.255.255.0
h6	128.238.66.106	255.255.255.0
h7	128.238.66.107	255.255.255.0

6 NTP Local Date

Execute `date` to display the system time of your host. Display the manual page of `date`, and study its options and usages. Try the following `date` commands:

```
date --date='2 days ago'
date --date='3 months 2 days'
date --set='+3 minutes'
date -r file-name
```

You can choose any file in the current directory for the *file-name* parameter.

Report

1. Submit the `date` outputs you saved. Explain the use of the commands.

7 Remote Date

While `tcpdump -n -nn -ex host your-host and remote-host` or `wireshark` is running, execute `rdate -p remote-host` to display the system time of the remote machine. Repeat the above `rdate` command, but use the `-u` option. Save the `wireshark` outputs for the lab report.

Report

1. What port numbers were used by the remote machine? What port numbers were used by the local host?
2. How many bytes of data were returned by the remote time server, both in the UDP case and in the TCP case?
3. What TCP header options were used?

8 NTP Sync

In this exercise, we start the NTP server daemon on *h0* and use NTP to synchronize all the other hosts to *h0*. Study the NTP configuration file `/etc/ntp.conf` in *h0* and in your host. If you are using another machine, you can telnet to *h0* and display the `/etc/ntp.conf` file in the telnet window. Start the NTP server on *h0* by `service ntp start`. To determine the status of the NTP server, use `service ntp status`. Use `tcpdump -ex -n -nn host your-host and h0` or `wireshark` to capture packets between your host and *h0*. Execute `ntpdate -d -v 128.238.66.100` to synchronize other host to *h0*. Study the output of this command. Save the `ntpdate` and the `wireshark` outputs for the lab report.

Report

1. Which port does the NTP server use? Justify your answer using the `wireshark` output.

9 ** NTP Server

Keep the NTP server running on *h0*. Execute `tcpdump -exn -nn host your-host and 128.238.66.100` or `wireshark` to capture the NTP messages between *your-host* and *h0*.

Config `/etc/ntp.conf` in *other-host* and add `server 128.238.66.100` to config file.

Start the NTP clients on your host, by `service ntp start`.

Wait for several minutes. Then terminate the `tcpdump` program. Analyze the captured NTP packets. Print one of the NTP packets for the lab report.

Execute `ntpq -p` to get `ntp` server list.

Execute `ntptrace` to show the client/server relation of NTP.

Report

1. Submit the NTP packet captured. List the fields and their values.
2. What was the rate at which NTP queries were sent by the client?
3. Which stratum was your host in? Which stratum was the NTP server in?

Part IV

NAT Exercises

For the exercises in this section, we use a network setting as shown in Figure 8.7. The lower subnet is a private network where the hosts are assigned with the Class A addresses with the 10.0.0.0/8 prefix. The upper subnet represents the Internet. The hosts, i.e. *h0* and *h1* are assigned with public IP addresses. Router 1 is used as the stub router, which performs address or port translation for the private network.

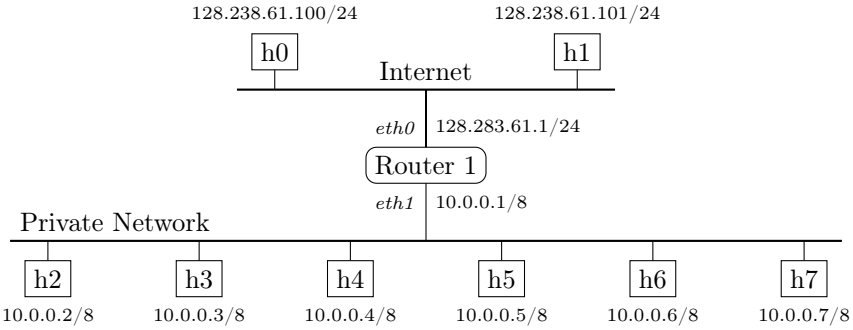

Figure 2: The network configuration for the NAT exercises (Figure 8.7)

10 NAT

Connect the hosts and Router 1 as shown in Figure 8.7. Then set the IP address and the network mask of your host as shown in the figure. In addition, you need to add a default route in your host's routing table, using the router interface on your subnet as the default router. One student should `telnet` to the router and configure the router as shown in Table 8.5. Note that there is a static translation that maps 10.0.0.7, or *h7*, to 128.238.61.104. Login to the router, execute `write term` to display the current router configuration. Execute `show ip nat translations` in the *Privileged EXEC* mode to display the translation table. Save both outputs for the lab report.

Report

1. How many entries were there in the translation table? Why?

```
config term
    ip nat pool mypool 128.238.61.102 128.238.61.103 netmask 255.255.255.0
    ip nat inside source list 8 pool mypool
    ip nat inside source static 10.0.0.7 128.238.61.104

    interface ethernet 0
        ip address 128.238.61.1 255.255.255.0
        no shut
        ip nat outside

    interface ethernet 1
        ip address 10.0.0.1 255.0.0.0
        no shut
        ip nat inside

    access-list 8 deny host 10.0.0.7
    access-list 8 permit 10.0.0.0 0.0.0.255
```

Code 2: Figure 8.7 (Table 8.5)]NAT Router Configuration in Figure 8.7 (Table 8.5)

11 NAT Visibility

Keep the login session to the router running. Execute `tcpdump -exn -nn` or `wireshark` on all the hosts.

Before any host in the private network send any packets out, `ping` an inside host (e.g. *h5*) from an outside host (e.g. *h1*). You may try to `ping` *10.0.0.5, 128.238.61.102, 128.238.61.103, or 128.238.61.104*. Can you `ping` these IP addresses?

Let an inside host send packets to an outside host, e.g. from *h5*, execute `ping` 128.238.61.100. Can you `ping` *h5* from an outside host now? Why? Which IP address should be used in the `ping` command in order to `ping` *h5*?

Execute `show ip nat translations` in the router login window to display the translation table. Save the output for the lab report.

Exchange the data you saved with a student in the other subnet.

Report

1. Answer the above questions. Use the saved translation table to justify your answers.
2. Compare the IP header of the ICMP query captured in the private network with that of the same ICMP query captured in the upper subnet, list their differences. Explain how NAT works.
3. In addition to the IP address, what else was changed in the ICMP query packet?

12 NAT Table

Keep the login session to the router running. Execute `tcpdump -enx -s 100 ip proto 1` or `wireshark` to capture ICMP messages.

Execute `socket -i -u -n1 128.238.61.101 8888` on *h2* to generate an ICMP port unreachable error.

Print the ICMP error message for the lab report.

Execute `show ip nat translations` in the router login window to display the translation table. Save the output for the lab report.

Exchange the data you saved with a student in the other subnet.

Report

1. Analyze the IP headers, the ICMP headers, and the ICMP payloads of the ICMP port unreachable errors captured in the private network and in the public network from the first experiment. Explain how ICMP error was handled by the NAT router.

13 NAT and PAT

Reboot the router to restore its default configuration. Then, configure the router to use PAT, as given in Table 8.6. Now all the hosts in the private network use the same IP address 128.238.61.1. However, note that there is a static translation that maps *h7*'s port 80 to 128.238.61.1 port 80.

Execute `tcpdump` on all the hosts.

Generate traffic between the inside and outside hosts. Examine the `tcpdump` output to see how PAT works.

Start the Apache web server on *h7*. Also, run `curl` on an outside host (e.g. *h0*), and with the URL `http://128.238.61.1`. Save the `tcpdump` output.

Use `show ip nat translations` to display and then save the translation table. Exchange the data you saved with a student in the other subnet.

Report

1. From the `wireshark` data, explain how PAT worked, both for a dynamic translation and a static translation.
2. With PAT, can you have two web servers in the private network? If not, why? If yes, explain how this can be done.

```
config term
    ip nat inside source list 8 interface ethernet 0 overload
    ip nat inside source static tcp 10.0.0.7 80 128.238.61.1 80

    interface ethernet 0
        ip address 128.238.61.1 255.255.255.0
        no shut
        ip nat outside
        exit

    interface ethernet 1
        ip address 10.0.0.1 255.0.0.0
        no shut
        ip nat inside
        exit

    access-list 8 deny host 10.0.0.7
    access-list 8 permit 10.0.0.0 0.0.0.255
```

Code 3: Figure 8.7 (Table 8.6)]PAT Router Configuration in Figure 8.7 (Table 8.6)

Part V
Socket Programming Exercises

14 Socket Programming

Examine the UDP socket programs `/home/netlab/code/UDPserver.c` and `/home/netlab/code/UDPclient.c` to learn how to write a UDP socket program. You can compile the C programs by using `gcc -o UDPserver UDPserver.c -lnsl` and `gcc -o UDPclient UDPclient.c -lnsl`. Compile output available at `/usr/local/bin/` directory.

Start `tcpdump host` *`remote-host`* or `wireshark` to capture packets from or to a remote host.

On the remote host, start the UDP server by `./UDPserver` *`server-port`*. Then, start the UDP client on your host by `./UDPclient` *`remote-host server-port a-message`*. You may execute the UDP client program on other hosts to connect to the same UDP server.

Terminate `tcpdump`, examine its output and compare the output with the UDP server and client outputs.

Repeat the above experiments, but now use the `TCPserver.c` and `TCPclient.c`.

15 Socket Options

Execute `man setsockopt` to display the various socket options and how to set them.

Examine the `netspy` and `netspyd` source code in Appendix C.2 of reference book to see how to create a multicast socket and how to set the TTL value for the packets.

16 FTP Socket Programming

This is an optional exercise on socket programming. Or, it can be assigned as a take-home project for extra credits. Note that familiarity with C (or C++) programming is required.

Problem

Examine the message exchanges of FTP. Write a FTP client program which takes a file name as input, and upload the file to a standard FTP server on a remote machine.

Hints

- First you need to set up the control connection to Port 21 of the remote machine, using a TCP socket.
- When the control connection is established, you need to exchange FTP commands with the remote FTP server, as given in Table 5.1 of reference book.
- You can first run `telnet remote-host 21`, then type `help` to list all the FTP commands. Also, you can try the commands out in the `telnet` window, e.g. use `USER netlab` to send the user ID and `PASS netlab` to send the password to the FTP server. To terminate the `telnet` session, type `QUIT`.
- In your program, these messages should be sent to the FTP server by calling the `send()` function of the local TCP socket.
- Also your program needs to parse the server responses (some examples are given in Table 5.2 of reference book) to find out the status of the previous FTP command.
- The FTP data connection should be established using the `PORT` command (see Chapter 5 of reference book).