In the name of Allah

بسم الله الرحمن الرحیم

Static and Dynamic routing 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

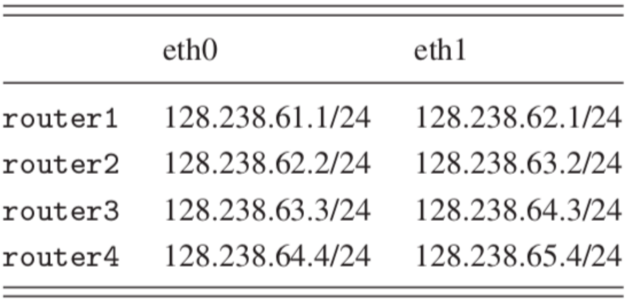
March 5, 2019

۱۴ اسفند ۱۳۹۷

A Simple Router Experiment

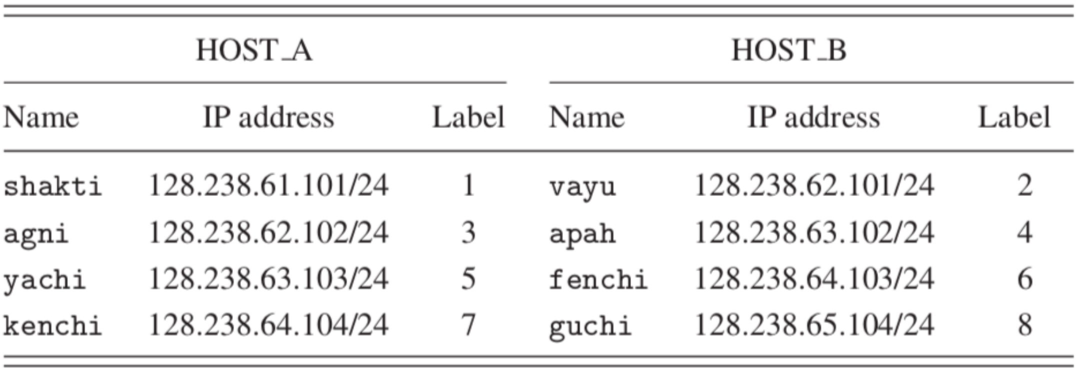
As in the previous lab, we will divide the students into four groups, each with two workstations, a router, and two hubs, which are to be connected as shown in Figure 4.10. The IP addresses of the routers and hosts are given in Table 4.2 and Table 4.3, respectively.

Table 4.2. *Router IP addresses for Fig. 4.10*



	eth0	eth1
router1	128.238.61.1/24	128.238.62.1/24
router2	128.238.62.2/24	128.238.63.2/24
router3	128.238.63.3/24	128.238.64.3/24
router4	128.238.64.4/24	128.238.65.4/24

Table 4.3. *Host IP addresses for Fig. 4.10*



HOST_A			HOST_B		
Name	IP address	Label	Name	IP address	Label
shakti	128.238.61.101/24	1	vayu	128.238.62.101/24	2
agni	128.238.62.102/24	3	apah	128.238.63.102/24	4
yachi	128.238.63.103/24	5	fenchi	128.238.64.103/24	6
kenchi	128.238.64.104/24	7	guchi	128.238.65.104/24	8

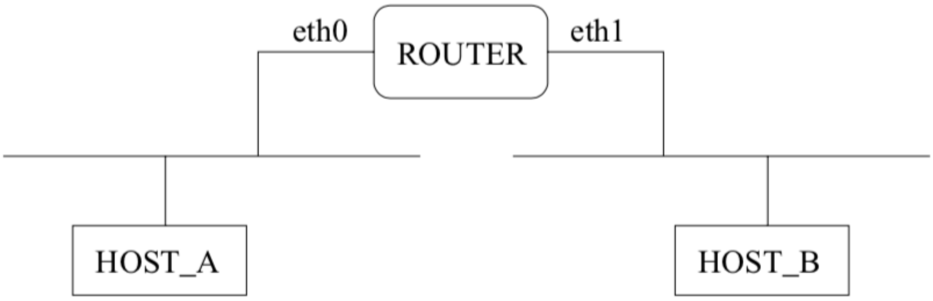


Figure 4.10. Simple router experiment.

1 Exercise 1

Configure the IP addresses of your workstations and the router as shown in Figure 4.10, Table 4.2 and Table 4.3.

Initially your host's routing table has no entry for the subnet on the other side of the router. In order to be connected, you need to add a routing entry for the other subnet in the routing table of your workstation (see

section 4.3.1).
Run **tcpdump -en** on your machine, and **tcpdump -en** on your partner's machine in the other subnet simultaneously:
tcpdump -en host *remote-host* and *your-machine*
Send **ping** messages continuously to your partner's machine:
ping -sv *remote-host*
After receiving the tenth echo reply, quit **ping** and save the **tcpdump** outputs from both machines. Also, copy the **ping** output.
During this exercise, don't run the **ping** program at the same time. For clean results, do your experiments in turn.

Lab Report

- When a packet was sent to a workstation in the other subnet, explain how the source and destination Ethernet addresses were changed.
- What are the source and destination addresses in the IP and Ethernet headers of a packet that went from your machine to the router?
- What are the source and destination addresses in the IP and Ethernet headers of a packet that went from the router to your partner's machine?
- Answer the above two questions, but now for the echo reply that was returned from your partner's machine.

Lab Report

- Use the **tcpdump** outputs from both machines to calculate the average delay that a packet experienced in the router. Note that the system times of the two machines might be different. Show all the steps and submit the **tcpdump** outputs with your report.
- Compare this value with the previous value in the case of the bridge. Which, a router or a bridge, is faster? Why?

RIP exercises

In this section, we will examine the operation of RIP. To enable the RIP routing process in a router, use the following commands in the *Global Configuration* mode.
`Router(config)#` **router rip**
`Router(config)#` **network** *network-number*,
where *network-number* could be 128.238.0.0. To remove the network, use:
`Router(config)#` **no network** *network-number*,
To shutdown the RIP process, use:
`Router(config)#` **no router rip**
Consider Figure 4.11 as our network topology for this section. Since the IP address of ethernet1 in router4 is the only interface which is different from the initial configuration in Appendix B, we will reboot all the four routers to restore their default configurations, and change the IP address on the ethernet1 in router4 only. Since our workstations started **routed** at boot-up time, no further action is needed to run RIP on the workstations. [1]

[1] The lab instructor should make sure that the IP-Forwarding function is enabled in each host.

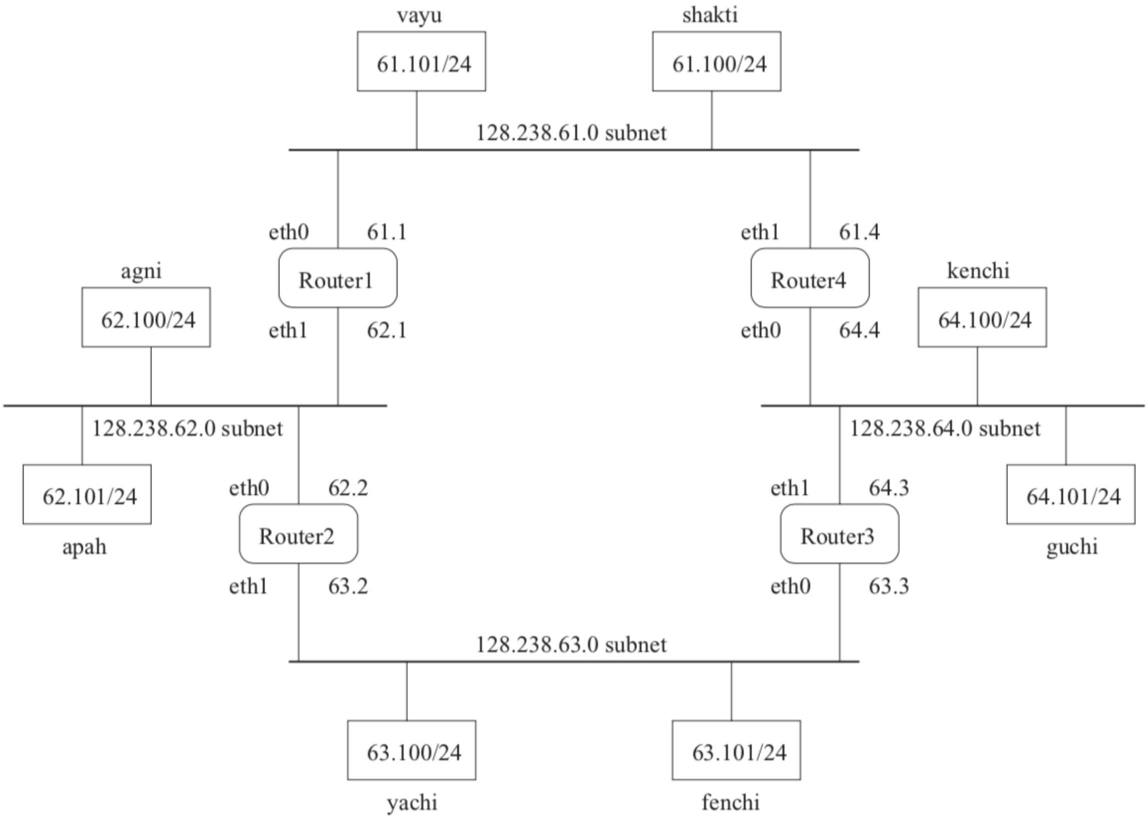


Figure 4.11. Network configuration of the RIP experiment.

2 Exercise 2

Connect the routers and hosts and change the IP addresses of the workstations and router4 as shown in Figure 4.11. Also, make sure that your workstation has no other routing entries than your own subnet and your loopback interface. For how to remove an entry from the host routing table, see Section 4.3.
Run the RIP process in each router. To avoid confusion, each router should be configured by only one person. After starting RIP in all the routers, test connections to other hosts by pinging them. Once you can successfully reach all the hosts, run the following command to capture the RIP messages sent on your subnet:

```
tcpdump -x -s 100 -c 4 -w exe2.out udp port 520
```

Save the routing table in your workstation. Note the number of hops needed to reach destinations other than in your own subnet.
Run **ethereal -r exe2.out &** to load the packet trace recorded by the above **tcpdump** command. Mark and print two different RIP messages captured in your subnet (see Exercise 6 of Chapter 1). Exchange the printed RIP messages with students in other groups. You need eight different RIP messages for your lab report.

Lab Report

- Explain why you can only get two different RIP messages in your subnet. Was a RIP packet forwarded by the routers? Why?
- Draw the format of one of the saved RIP response packets from your sub- net, including the IP and UDP headers and the RIP message (see Figs 0.13, 0.14, and 4.4). Identify each field, and express their values in decimal format.
- For the other seven RIP response packets collected, explain the contents of the RIP messages only, excluding IP and UDP headers.

- Draw the distance tables and the routing tables in the routers based on Figure 4.11, assuming that number of hops is used as the metric.
- Verify the routing tables using the RIP messages you captured.

3 Exercise 3

In this exercise we will examine how RIP responds to link failures. Send **ping** message continuously from `apah` to `yachi` and start **tcpdump** on `apah`. Let the two programs run during this exercise.
Disconnect the cable from the `ethernet0` port of `router2` from the hub in the 128.238.62.0 subnet, and type the **time** command to get the current time.
Observe the **ping** and **tcpdump** windows. When the connection is re-established, type the **time** command again. See how much time RIP takes to alter the routing table in your workstation to the new topology.
Once you can successfully reach other hosts, connect the cable to the original position. Again, measure the time that RIP takes to change your routing table.

Lab Report

- Compare this time with the previous value in the spanning tree experiment.
- Explain why it takes this time for RIP to react to the route change. Refer to Section 4.2.4 for RIP operation and default timer values.

Routing experiments with ICMP

4 Exercise 4

Eliminate the routing entries for subnets other than your own and the loopback interface. Save the routing table for your lab report.
Create a default routing entry using one of the routers directly connected to your workstation.
While **tcpdump -enx -s 100 ip proto 1** is running, send **ping** messages to a host that is three hops away through the default router.
After capturing an ICMP redirect message, save the **tcpdump** output, the **ping** output, and your workstation's routing table. You may need to **ping** the same host several times in order to get your routing table updated.

Lab Report

- Submit what you saved in Exercise 4.
- Identify every field in the ICMP redirect message (see Figure 4.2).
- Compare the original routing table with the new routing table. Explain the meaning of the flags of the new entry.

OSPF exercise

In order to enable OSPF in the routers, you need to create an OSPF routing process first. Then, define the range of IP addresses to be associated with the routing process and assign area IDs for these IP addresses, using the following commands:
`Router(config)#` **router ospf** *process-id*,
`Router(config)#` **network** *address wildcard-mask* **area** *area-id*
Process-id is a numeric value local to the router. It does not have to match *process-id*s on other routers. *address* is the network address of the interface on which the OSPF process runs (128.238.0.0 in our case). *wildcard-mask* helps reduce the number of configuration commands. 0 is a match and 1 is a "don't care" bit (0.0.255.255 in our case). *area-id* is the number of the area that the interfaces belong to (see Figure 4.7). It can be any integer

between 0 and $2^{32} - 1$ or can have an IP address form. Note that 0 is reserved for the backbone.
The above commands are required to configure OSPF, while other tasks (configuring interface parameters, configuring area parameters, etc.) are optional. For more information on other configuration tasks, refer to the router manual.
Consider Figure 4.13 for our OSPF experiment. The lab instructor should reboot the routers to restore their default configurations.

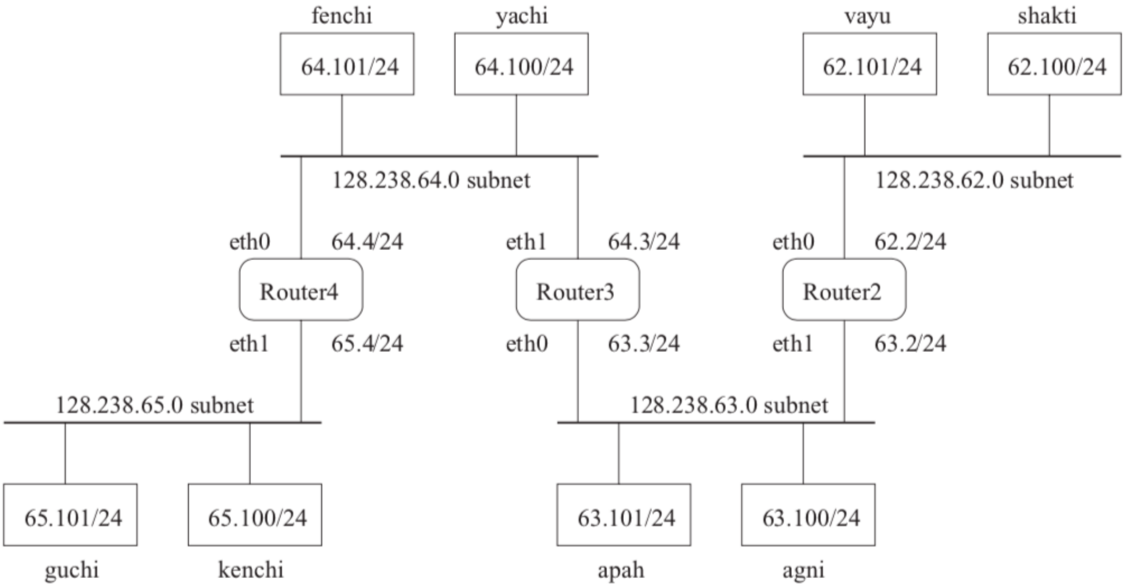


Figure 4.13. Network configuration for the OSPF exercise, the static routing exercise, and the **traceroute** exercise.

5 Exercise 5

After connecting the cables properly, change the host IP addresses as given in Figure 4.13 You need to remove the default route added in Exercise 4 from the host routing table. Note that the router interfaces are set as Fig. 4.13 by default.
Run the following command to capture any OSPF packets:
tcpdump -x -s 120 ip proto 89
Login to a directly connected router and start the OSPF process. Set the argument *area-id* to 1 for all the routers.
The workstations in our lab run **routed** (which uses RIP). The routing daemon supporting OSPF, **gated**, is not installed. In order to reach the routers and hosts in the other subnets, you need to add a default router in your host's routing table.
Examine the routing table in each router (see Section 4.3). When the routing table gets an entry for the network that is not directly connected, kill the **tcpdump** process and save the **tcpdump** output.
Collect the **tcpdump** outputs from other subnets. Study the various types of OSPF packets from the **tcpdump** outputs.
You can display OSPF information in a router using the following commands in the *Privileged EXEC* mode.

```
show ip ospf
show ip ospf database [router|network|summary|asb-summary|external|database-summary]
show ip ospf interface ethernet [0|1]
show ip ospf neighbor
show ip ospf virtual-links
```

Lab Report

- Draw the common header of a saved OSPF message, giving the decimal values of the header fields (see Fig. 4.8).
- Submit the routing tables you collected from the routers.

Static routing experiment

In this experiment, we reuse the network as shown in Figure 4.13.

6 Exercise 6

After checking the wiring, as shown in Figure 4.13, reboot the routers to restore their initial settings. Check the IP addresses of the workstations as shown in Figure 4.13.
Remove all the routing entries other than your own subnet and the loopback interface from your host routing table. Save the output of **netstat -rn** before building your workstation's routing table.
Examine Figure 4.13 and build your host's static routing table manually.
Telnet to a router that is directly connected to your workstation, and save its routing table before building any route. Save the routing table of the other router if you have one more router connected directly. You may not be able to **telnet** to a router that is not directly connected. In this case, copy the initial routing table of these routers from students in other subnets later.
Now configure the routing table in each router. See Section 4.3 for commands and syntax on manipulating router routing tables. Note that each router should be configured by one person only.
Use **ping** to test the connections. When you can reach all other subnets successfully,[2] save the routing tables in your workstation and all the routers for the lab report.

Lab Report

Submit the routing tables saved in this exercise.

Traceroute experiment

In this exercise, we use the same network and configuration of the previous exercises, and use **traceroute** to find a multi-hop path.

7 Exercise 7

Execute **tcpdump -enx -s 100 host** *your-host* **and** *remote-host* on your host, where *remote-host* is a workstation at least two hops away.
Then, execute **traceroute** *remote-host* to find the route from your host to the remote host.
Save the output of both **traceroute** and **tcpdump**.

Lab Report

Submit what you saved in this exercise.
From the **tcpdump** output, explain how the multi-hop route was found. Explain the sequence of the ICMP messages used.

[2] Even when the routing table in your workstation and all the routers are configured perfectly, you may not be able to **ping** a remote host, if the routing table in the remote host is incorrect. When you can get **ping** reply messages from all the interfaces of the routers successfully, your work is done for this exercise.