



University of Bamberg
Professorship for Computer Science

Foundations of Internet Communication
KTR-GIK-M
Laboratories Summer Term 2020

Assignment 2

Static Routing and DHCP

Marcel Großmann, Duy Le, Jopaul John, Markus Wolff

Issued: May 11, 2020

Deadline: May 24, 23:55, 2020

Prelab

1. Make yourself familiar with the **iproute2** framework and especially its command `ip route`.
2. Browse through the introduction material about the Vyatta software router (provided in the VC-course). We will use a fork of the Vyatta software - called VyOS.
3. Read the documentation about [Kathará Labs](#).
4. Make yourself familiar with the *netstat* command.
5. Make yourself familiar with the *brctl* command, which is used to configure a normal Linux PC as bridge.
6. Refresh your knowledge about IP networks (e.g. IP addresses, sub-netting, etc.) and make yourself familiar with the Linux network configuration commands and files, like *ifconfig*, *ip*, etc. (e.g. here <http://linux-ip.net/>)

1 The Netstat Command

To investigate the *netstat* command, you can either use any Linux machine, or you can create a Kathará node.

1. Display the information on the TCP and UDP ports that are currently in use.
2. Display the statistics of the various networking protocols.
3. Suppose you want to write a small application that needs the process id (PID) of a given application. In order to achieve this, use a command of your choice, e.g. *grep*, *sed* or *awk*, to filter the output of *netstat*. Your application should **only** deliver **the port number** of a particular application (e.g. *inetd* or *sshd*), identified by the PID, as parameter.

2 Static Routing

Every Linux PC or, in our case, every Kathará node with at least two network interfaces can be set up as an IP router. The configuration of a Linux PC as a simple IP router involves basically only two steps:

- modifying the configuration of Linux, so that IP forwarding is enabled,
- and configuring the routing table.

Moreover, we can also emulate router software like VyOS in Kathará by utilizing the image [unibaktr/vyos](#). However, emulating router software comes with a price and such a container needs to be executed in privileged mode. To create the Kathará lab, provide the required `lab.conf` file.

1. At first, plug in “the cables” as shown in Figure 1 below by writing Kathará’s `lab.conf` file.
2. Additionally, specify the Docker images, which should be used in your environment according to Table 1.
3. Then, configure all the interfaces as shown in Table 1. You may also use startup scripts to ease the configuration.
4. Start Wireshark on your host to capture the traffic on the collision domains (CDs). You can get the ID’s of the CDs with `docker network ls`.
5. Try to check the connectivity between all hosts. What is needed to provide global connectivity, i.e., all hosts can communicate with each other?
6. Start to manually add the routing entries wherever needed, to provide global connectivity.
7. Start to ping `web1` from `pc1`. Investigate what happens with the packets that get forwarded at `r1`. What do you observe?

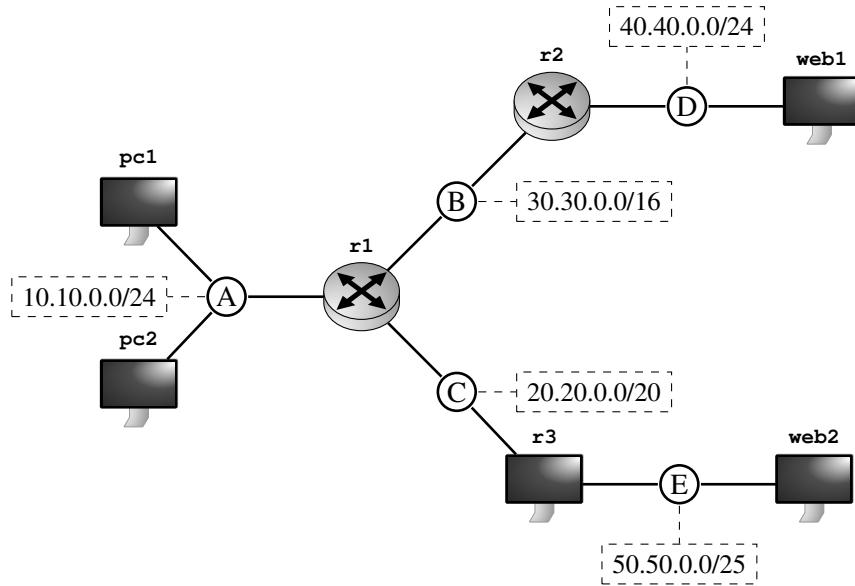


Figure 1: Experiment configuration

Name	Docker Image	IP Address	Interface	CD
pc1	unibaktr/alpine:busybox	10.10.0.10/24	eth0	A
pc2	unibaktr/alpine:busybox	10.10.0.11/24	eth0	A
r1	unibaktr/vyos	10.10.0.1/24	eth0	A
r1	unibaktr/vyos	30.30.0.1/16	eth1	B
r1	unibaktr/vyos	20.20.0.1/20	eth2	C
r2	unibaktr/vyos	30.30.0.2/16	eth0	B
r2	unibaktr/vyos	40.40.0.2/24	eth1	D
r3	alpine	20.20.0.3/20	eth0	C
r3	alpine	50.50.0.3/25	eth1	E
web1	alpine	40.40.0.100/24	eth0	D
web2	alpine	50.50.0.100/25	eth0	E

Table 1: Experiment configuration

[Hint:] To flush the routing cache, use the following command: `ip route flush cache`

3 DHCP - Dynamic Host Control Protocol

Now consider that `pc1` and `pc2` are in a local area network (LAN), where router `r1` provides the IP addresses via DHCP. vyos

1. First, configure a DHCP server on `r1`, which provides IP addresses in the range from 10.10.0.100/24 to 10.10.0.200/24. Additionally, let the server provide the default gateway to the attached machines.
2. From your host start a Wireshark capture on CD A.
3. Start the [udhcp client](#) on `pc1` and `pc2` to automatically gather IP addresses.
4. From `pc2` try to ping `web2` and ensure connectivity.
5. With the captured procedure, explain how DHCP is working.

Lab Report: Write a short summary of your observations in the experiment and the encountered pitfalls in the programming exercise, provide screenshots whenever possible to justify your statements. Include the commands and write a short explanation with screenshots.