TCP/IP NETWORKING

LAB EXERCISES (TP) 4
DYNAMIC ROUTING (OSPF) AND SDN BASICS

November 17th, 2022
Deadline: November 30th, 2022 at 23.59 PM

1 LAB ORGANIZATION AND INSTRUCTIONS

In this lab, you will learn how to configure the OSPF routing protocol, which typically runs inside the network of an autonomous system (AS). The protocol automatically sets up network routes that ensure shortest path between two points in the network with respect to a predefined metric (such as number of hops). You will configure a fully functional network using "Cisco-like" **emulated** routers. You will learn how to configure a network of routers. Optionally (bonus exercise), you can learn about software-defined networking (SDN) and study how it can be used to create forwarding rules on switches. You are strongly advised to do this part as it gives you a good idea of how enterprise networks are managed today.

For this lab, you will be using the same Mininet on the virtual machine provided on Moodle (see Lab 0 for installation instructions if need be).

The lab is meant to be done sequentially. Random access might give you different answers. We advise you to do a section in one sitting. Restarting Mininet within a section might make the analysis difficult. Finally, if you are copying commands into your terminal, make sure that the copied string is still a valid command: especially, be wary of the *underscore* and *newline* characters.

2 FRROUTING: SOFTWARE ROUTING SUITE

The Internet core is run by powerful routers that can handle large amounts of traffic and are built by companies such as Cisco, Huawei, or Juniper. Even in a large company, or in large university campuses (such as EPFL), these routers are present. They use proprietary operating systems (such as Cisco IOS, or JunOS) and can be accessed via control terminals tailored for network configuration, with commands that are quite different from those you may encounter in a UNIX/Linux console. In this lab and in lab 6, we will give you a flavour of router configuration.

Ideally, we would have liked to run Cisco IOS in a virtual environment. It is technically possible but not legal, as Cisco or Juniper do not allow their OSs to be run on a device other than their routers. Therefore, we will use FRR, a free software that implements and manages various IPv4 and IPv6 routing protocols. It accepts similar commands to the ones in Cisco's IOS.

2.1 What is FRR and how does it work?

FRR is a routing software suite that provides implementations of several routing protocols (namely OSPF, RIP, and BGP-4) for Unix platforms. The architecture of FRR is shown in Figure 1.

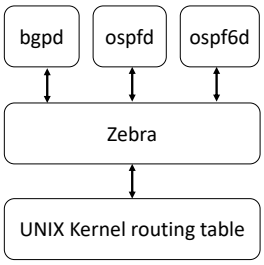

Figure 1: FRR Architecture

As depicted in the figure, it consists of a handful of processes that can be run in the background as daemons. Three FRR processes (daemons) are important for executing this lab:

- *zebra*: is used to manage the network interfaces of a machine (in our case, each router will run in the virtual machine). It allows you to configure them (using IPv4 and/or IPv6 addresses), to monitor their states, and it provides a more detailed view of the routing tables than the `route -n` command. In a way, *zebra* is a replacement for the Linux networking commands used during the first three labs (*i.e.,* `ifconfig`, `route`, etc.).
- *ospf*: handles OSPF version 2 implementation.
- *ospf6d*: handles OSPF routing for IPv6.

3 Scenario: A Game of Routers

With the impending attack of the White Walkers on Westeros, all the kingdoms must unite in the fight for the living. As the lands of men stretch from Winterfell in the North to Valyria down in the South, the wise maesters at the Citadel have realized that their communication network with carrier ravens will not be sufficient to communicate for large distances in short time.

After several sleepless nights in the Citadel, Maester Illyrio Pycell and his intern Samwell Tarly have designed the communication mechanism, called TCP/IP. Together, they setup routers and switches at strategic locations in Westeros as shown in Figure 2. There are a total of 5 routers r1-r5 placed at Winterfell, Braavos, Valyria, Casterly Rock, and King's Landing, respectively. Samwell has connected the routers through 7 switches as shown in the figure. He has also configured the routers with the IP addresses as shown in the figure. Note that all IP Addresses for router rx end in x.

Sam's configuration scripts are available in the lab folder. To be consistent with the paths in this document and in the scripts, please place the uncompressed folder directly in the Desktop of the virtual machine and be sure to name it **lab4** (case-sensitive). Run `lab4_network.py` as root from a terminal of your virtual machine, to recreate the established topology. See the output of the `net` command in the Mininet prompt and verify whether the connections in the created network correspond to the ones of Figure 2. Use the `pingall` command in the mininet prompt.

Now go to Moodle and answer questions for this Lab. You need to come back to this pdf for answering Part 4 (Bonus), which involves Fig. 3.

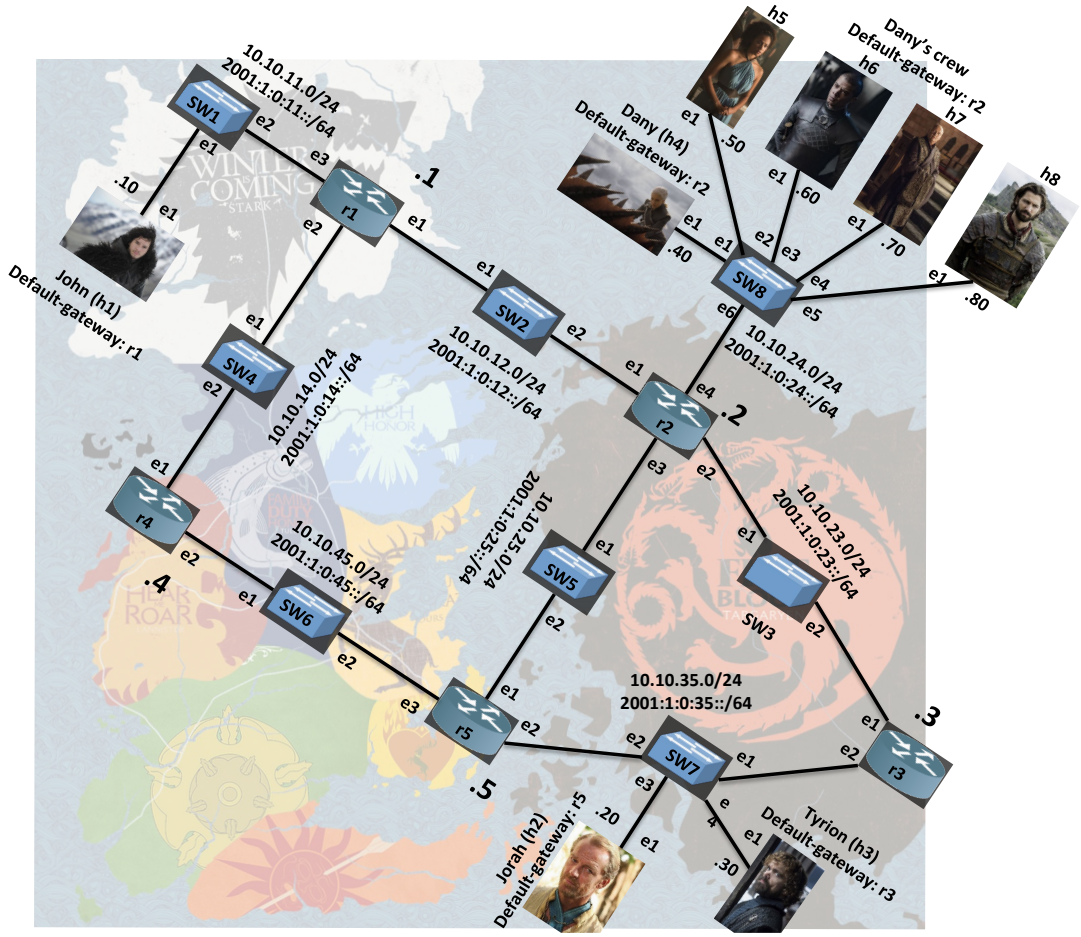

Figure 2: Networking in Westeros

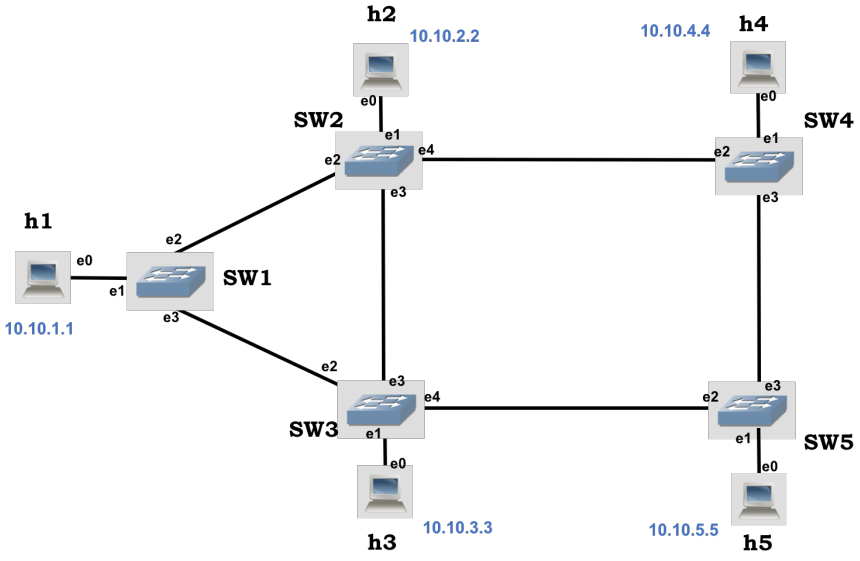

Figure 3: Lab 4 SDN topology

4 BONUS: SOFTWARE DEFINED NETWORKING (SDN)

For students who could not run openVSwitch switches in the previous labs (and were using LinuxBridges), please contact the TAs to borrow a Minix for this part.

Fig. 3 is useful to visualize and answer questions in Part 4 (Bonus) on SDN. All necessary information except this figure is on Moodle.