

IMT Atlantique
Département Informatique
Technopôle de Brest-Iroise - CS 83818
29238 Brest Cedex 3
URL: www.imt-atlantique.fr



UE PRIP
2022

Lab 3: Intradomain routing, the OSPF protocol

Version: 1.4

Report filled-in by:



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1. Objectives

- Understand the advantages of a dynamic routing protocol
- Understand the main functioning characteristics of a link state routing protocol (OSPF in this case)
- Get familiar with a routing operating system (Quagga in this case) and the commands needed to configure OSPF

2. Pre-Lab

- Read about OSPF. You can check some of the suggested bibliography [9],[8], or find information in the web. In particular, be sure to understand the following concepts:
 - OSPF Neighbor
 - OSPF Adjacency
 - Link
 - Link State Advertisements and the different types of them
 - RouterID
 - Designated router
 - OSPF area
 - Types of OSPF messages and their purposes

3. Introduction

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) that routes packets within a single autonomous system (AS). OSPF uses link state information to make routing decisions, calculating routes using the shortest-path-first (SPF) algorithm (also referred to as the Dijkstra algorithm). Each router running OSPF floods link state advertisements throughout the AS or area that contains information about that router's attached interfaces and routing metrics. Each router uses the information in these link state advertisements to calculate the least cost path to each network and create a routing table for the protocol [1].

3.1. Quagga

Quagga is a routing software package that provides several routing protocols such as RIP, OSPF and BGP. With Quagga installed on, a system will act as a dedicated router. These systems are able to exchange routing information using routing protocols and we can also configure them directly by command line which is similar to Cisco router configuration [2].

Quagga consists of several processes named daemons that work together to build a routing table.

Each daemon handles a corresponding protocol such as `ripd` for RIP protocol and `ospfd` for OSPF protocol. Meanwhile, `zebra` daemon is an IP routing manager which provides kernel routing table updates, interface lookups, and redistribution of routes between different routing protocols. Therefore, it is mandatory to enable zebra daemon along with other protocols.

Thanks to this multi-process architecture, each daemon has its own configuration file and terminal interface. When you configure a static route, it must be done in zebra configuration file, when you configure BGP network it must be done in bgpd configuration file, etc.

3.2. Configurations in Cisco router and Quagga

3.2.1. User Levels and Modes in Quagga

Quagga adopts, same as Cisco IOS, user access levels for configuration as a security feature. We have thus two different levels, the so called user EXEC level, and privileged EXEC level. User EXEC level only allows access to basic monitoring commands; privileged EXEC level allows access all router commands.

- User EXEC level: the first level when users access to a router. This level displays in the router as below:

```
hostname >
```

- Privileged EXEC level: to access this level, from user EXEC level, we type:

```
hostname > enable
hostname > <password> // password to access privileged EXEC level
hostname # // privileged EXEC level
hostname # disable // to disable this level and go back to user EXEC level
```

Router commands available from privileged level are organized in different *configuration modes*. Each configuration mode allows to configure a specific feature (interface, router, etc.).

- Global configuration mode: used to configure your system globally, or to enter specific configuration modes to configure specific elements such as interfaces or protocols. To access this mode, from privileged EXEC mode, we input:

```
hostname # configure terminal
hostname (config) # // Global configuration mode
hostname (config) # exit // exit global mode and back to privileged mode
```

- Interface mode: specifies the interface to be configured. Access this mode from global configuration mode:

```
hostname (config) # interface <name or number of interface>
hostname (config-if) #
```

- Router configuration mode: configure an IP routing protocol. Access this mode from global configuration mode

```
hostname (config) # router <ospf/bgp/rip...>
hostname (config-router) #
```

Tip: At any level, to display the list of available commands, type `?`. More details can be checked out in [3].

Routers can be configured either through the command line, either through configuration files. For both options, we have to respect the levels introduced above. We shall see in the following two subsections, examples of configuration files and of command line configurations.

3.2.2. Daemon configuration files

To configure `zebra` and `ospfd` daemons through configuration files, a configuration file must be created for each router. When triggering a daemon, the configuration file must be specified.

1. Zebra configuration file: below you can find an example of `zebra.conf` file.

```
hostname R1 //Name of zebra daemon in router
password en //Password to access EXEC level in router
enable password en //Password to access privileged EXEC level in router

interface r1-eth1 //Access to interface mode of r1-eth1
  ip address 192.168.1.1/24 //Assign IP address to r1-eth1

log file /tmp/R1-zebra.log //Generate log file R1-zebra.log in /tmp directory
```

2. OSPF daemon configuration file: below you can find an example of `ospfd.conf` file.

```
hostname R1 //Name of ospfd daemon in router
password en //Password to access EXEC level in router
enable password en //Password to access privileged EXEC level in router

router ospf //Access to router mode of OSPF routing protocol
  ospf router-id 1.1.1.1 //Set router id for router
  network 192.168.1.0/24 area 0 //Enable OSPF in interfaces which networks
  are attached to area 0 in the interface
  network 192.168.100.0/30 area 0 //and also advertise these networks to
  its neighbors, also identify
  network 192.168.102.0/30 area 0
  passive-interface r1-eth1 //Set interface r1-eth1 as passive, that will
  not listen and send OSPF packets in this interface

log file /tmp/R1-ospfd.log //Generate log file R1-ospfd.log in /tmp directory

debug ospf packet all send //Enable debugs to observe OSPF states, packets...
debug ospf packet all rcv
```

Please note the different passwords in the routers' daemons are configured as in the previous examples (i.e.: `en`).

3.2.3. Configuration through the command line interface

Quagga also allows to configure daemons through the command line interface (CLI). For that, once the daemon is running, we need to access to the CLI. To access `zebra` or `ospfd` daemon CLI in each system, from a terminal of the system, type: `telnet localhost zebra` for accessing a `zebra` CLI or `telnet localhost ospfd` for accessing an `ospfd` CLI.

Once there, we retrieve the different levels and modes we have already seen. Firstly, a password is required to access the user EXEC level, then another password to access the privileged EXEC level. Those passwords must be defined in configuration files in advance.

For instance, in the following you can see the sequence of commands when accessing to the `ospfd` daemon CLI on a router with hostname "R1":

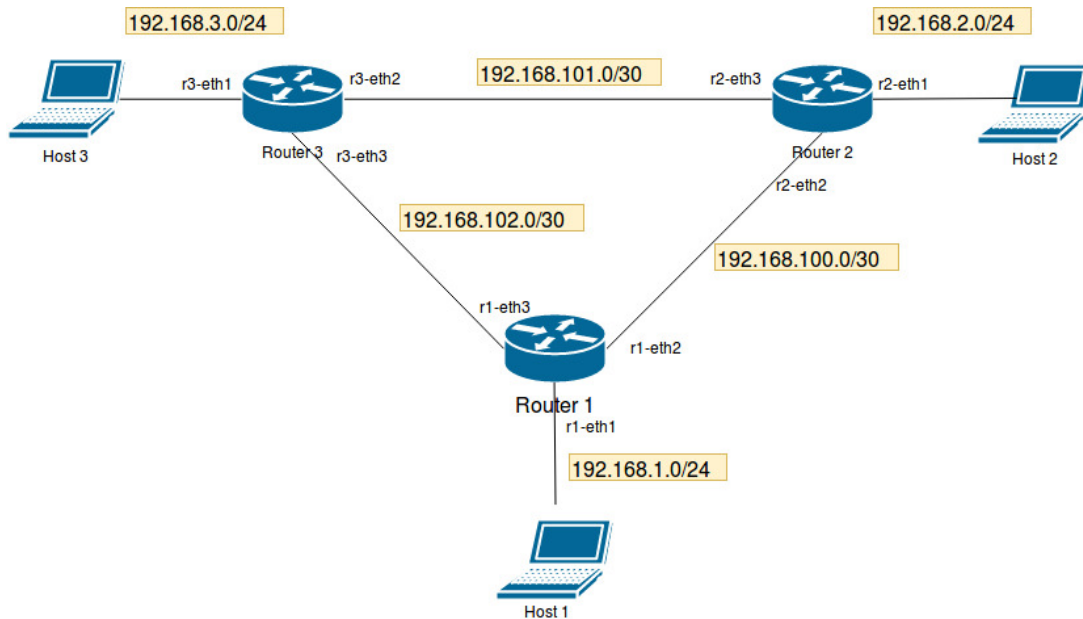


Figure 1: Topology 1

```
telnet localhost ospfd
Password: en // password to access user mode is 'en'
R1> enable // type "enable" to access privileged level
Password: en // password to access privileged level is 'en'
R1# // in privileged level
```

Similar to the commands in configuration files, after accessing zebra daemon, you can assign the IP address to a given interface, show the routing table, etc. About ospfd daemon, you can configure everything related to OSPF protocol, show OSPF information, database and more.

Following there is a list of useful commands you will be using throughout the lab to obtain information about the OSPF network. Please notice that these commands have to be input in the ospfd daemon CLI.

show ip ospf: information on a variety of general OSPF and area state and configuration information such as router ID, attached areas...

show ip ospf interface <interface_name>: shows state and configuration of OSPF the specified interface, or all interfaces if no interface is given.

show ip ospf neighbor: displays OSPF neighbor information on a per-interface basis such as router's priority, state, ip address.

show ip ospf database: displays lists of information related to the OSPF database.

show ip ospf route: display all routes distributed in OSPF area.

More details about configuration in Cisco router and Quagga are given in [4].

4. Hands On

In this lab, you are going to use Mininet to emulate a network. You will be provided all the scripts needed to emulate the networks, run and configure OSPF.

You will be working in the course's VM. Once on a VM terminal, check the presence of the `net_labs` directory with command `ls`. Go to the directory and **update its content**:

```
cd net_labs
```

```
git pull origin master
```

We will go through 3 exercises, you will not be asked to start from scratch any OSPF configuration, but you are asked to carefully study a working configuration, and to make some modifications. Exercise 4 is optional.

4.1. Exercise 1: OSPF routing protocol basic configuration and functioning

In this first exercise, we create a simple topology consisting of three routers running OSPF, see Fig 1.

4.1.1. Network set-up

1. Run the course VM and open a terminal
2. The working directory for this lab is `~/net_labs/ospf/`. Go to this directory and, if you have not done it yet, update its content through git: `git pull origin master`.
3. Execute the Python script to create topology 1. For that, type `sudo python3 run.py topology1`
4. Verify the configuration of all routers of the emulated system through the Mininet CLI. In particular, verify interface configurations (you can use `ip address` or the more portable but unmaintained `ifconfig`) and routing tables (`ip route` or the more portable but unmaintained `route -n`).
5. Write down the IP addresses of the routers in your printed copy of this lab assignment.

Question 4.1.

Do you have connectivity between all the systems? Why? Justify why you have or not have connectivity between certain systems.

You are going to run zebra and ospfd daemon at each of the routers by executing the already provided configuration files. For that, you are going to use a script we have already provided for this purpose (`config.mn`).

6. Run `zebra` and `ospfd` with the help of the provided script. For that, from the Mininet CLI, type:

```
source config.mn
```

7. Wait for about 1 minute for network convergence.
8. For each router, show (and understand) the routing table calculated by OSPF. For that you need to:
 - (a) Open a terminal on router i (from mininet CLI type `xterm Ri`).
 - (b) From the router's terminal you need to access to zebra daemon (`telnet localhost zebra`)
 - (c) Once in the zebra CLI you can type `show ip route` to show the routing table of R_i .
9. Run `pingall` in Mininet CLI to test connectivity

Question 4.2.

Do you have connectivity between all routers? between all hosts? Which is the content of the routing table of router 1? Explain its content.

Question 4.3.

According to the routing table of R1, which path is taken for going from h1 to h3? Why did OSPF choose such route?

4.1.2. Protocol functioning

1. We now want to capture packets in the very beginning of the routing protocol, so first of all, stop the emulation. To do so, type in the Mininet CLI `exit`. This will also stop the quagga daemons.
2. Run again the emulated topology (`sudo python3 run.py topology1`)
3. Run wireshark on each router. For that, you can type on the Mininet CLI `Ri wireshark &` for `i=1` to 3.
4. Start a capture at all interfaces of the router (for example `r1-eth2` and `r1-eth3` in router 1).
5. Start zebra and ospfd daemons by typing, on the Mininet CLI `source config.mn`

Question 4.4.

Observe in wireshark, which OSPF messages are routers exchanging among them? Give a detailed explanation, including the different messages that you see, the order of appearance, the protocol stack being used.

6. Run a ospfd terminal (`telnet localhost ospfd`) on the router of your choice and try the commands `show ip ospf database`, `show ip ospf database network`, `show ip ospf neighbor`, `show ip ospf route`.

Question 4.5.

What information does each of the commands provide?

Links costs

OSPF is a link state vector protocol based on link state (metric, cost) to calculate the shortest path (SPF) to a destination and create a routing table. In quagga, there are two ways to set a link's metric:

Method 1:

```
hostname (config) # router ospf
hostname (config-router) # auto-cost reference-bandwidth <1-4294967>
```

This sets the reference bandwidth for cost calculations, where this bandwidth is considered equivalent to an OSPF cost of 1, specified in Mbits/s. The default is 100Mbit/s (i.e. a link of bandwidth 100Mbit/s or higher will have a cost of 1, costs for lower bandwidth links will be scaled with reference to this cost). This configuration setting must be consistent across all routers within the OSPF domain. Cost will be calculated as reference bandwidth/ default cost (with Ethernet, default = 10).

Method 2:

```
hostname (config) # interface <name_of_interface>
hostname (config-if) # ip ospf cost <1-65535>
```

This way sets link cost for the specified interface. Notice that as given above, the configuration for OSPF protocol has to be done in ospfd daemon.

1. Make sure to have a wireshark capture running on r1-eth3
2. Using the second method, change the link cost of the interface r1-eth3 of router 1 to 100, and observe what happens.
3. Using again the second method change the link cost of the interface r3-eth3 of router 3 to 100.

Question 4.6.

Which type of message is sent to inform about the change? How does the routing table change? Compare the actual routing table for router 1 with that one of the answer to question 1.

In order to verify the path that is being taken by the packets, we can run a traceroute command from h1 to h3.

4. Run a traceroute command from h1 to 192.168.3.2. For that, you can type on Mininet's CLI `h1 traceroute 192.168.3.2` or `h1 tracecroute -n h3`. Another useful command is `ping` with the -R option.

Question 4.7.

What is the output of traceroute command?

Topology changes

We are now going to force a link to shut down and observe how the OSPF protocol reacts.

1. Start a wireshark capture at R3
2. Open a terminal in h1, and execute a `ping -R` to h3 (`ping -R` makes the echo packets record the route they take)
3. Disable the interface r2-eth3 by going to **interface mode of zebra daemon** in router 2 then type:
`hostname (config-if) # shutdown.`

Question 4.8.

Did packets get lost due to the link going down? Justify your answer, by explaining how OSPF reacts. To build your answer observe the capture, the output of the ping command, and the content of the routing tables.

We are now going to observe what happens when the link comes up back again.

4. Same as step 2
5. Enable interface r2-eth3: `hostname (config-if) # no shutdown.`

Question 4.9.

Did packets get lost due to the link going up? Justify your answer, by explaining how OSPF reacts. To build your answer observe the capture, the output of the ping command, and the content of the routing tables.

6. Stop the emulation (close all xterms, if any, and type `exit` on Mininet CLI)

From now you will use a script (`run_ospf.py`) that automates running ospf and zebra daemons in all routers. You do no longer need to use `config.mn` script.

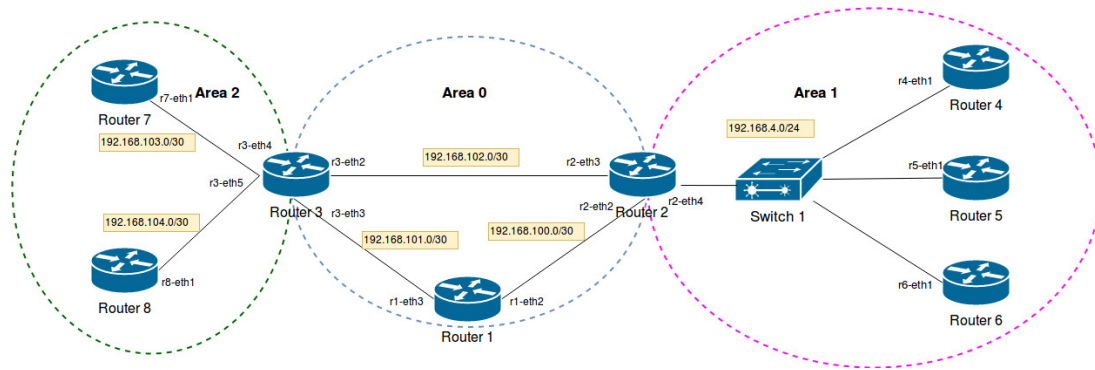


Figure 2: Topology 2

4.2. Exercise 2: OSPF Areas

As you have observed, each OSPF router is aware of all other routers and their links within the OSPF network. When the network grows large, the link state database that each router maintain also grows large. Besides, if there is any change in the network, the SPF recalculation could take significant time causing a negative effect on network's performance. To solve that situation, OSPF uses the concept of areas.

An area is a logical grouping of contiguous networks and routers. All routers in the same have the same information concerning the area, but they do not have detailed information about other areas. The main benefits of creating areas is that the size of the topology, the routing table on a router, and the number of Link State Advertisements (LSAs) transmitted are reduced, less time is required to run the SPF algorithm and routing updates are also reduced [5].

In this exercise, we are going to work with three OSPF areas. This topology is illustrated in Figure 2.

Question 4.10.

In the topology of Figure 2, which routers are backbone routers, which ones are internal routers, and which ones are area border routers (ABR)? What is the function of an ABR?

1. Run the working topology, by calling `sudo python3 run_ospf.py topology2`
2. Check the running configuration at some routers (e.g. R7 and R6) and their OSPF database. For that, on ospfd privileged level, type `show running-config`, and `show ip ospf database`, respectively.
3. You must have noticed that due to a configuration error, routers in area 2 have also been configured as being in area 0. We have thus a big area 0 with 5 routers.

Question 4.11.

Show the OSPF database in router 6 and 7, what is difference on the content of each database?

We will now correct the misconfiguration and configure routers 7,8 and 3 properly as in Figure 2. For that we need to update the configuration of area of each network on each router. For example, for router 3 you need to first disable the current configuration and then enable the new one, as follows:

```
ospfd-R3 (config) # router ospf
ospfd-R3 (config-router) # no network 192.168.103.0/30 area 0
ospfd-R3 (config-router) # network 192.168.103.0/30 area 2
ospfd-R3 (config-router) # no network 192.168.104.0/30 area 0
ospfd-R3 (config-router) # network 192.168.104.0/30 area 2
```

4. Go to configuration mode on ospfd for each concerned router (Routers 7,3 and 8)
5. Correct the area configurations on each of those routers, according to Figure 2 and to the previous example.

Question 4.12.

Once again, show the OSPF database in router 7. What is the difference with respect to its previous content, seen in question 11?

6. Look at the content of the OSPF routing table in R7. Verify that all destinations are reachable.
7. Try a pingall from Mininet CLI to verify connectivity in your network.

Question 4.13.

You should by now have an idea of what Link State Advertisements (LSAs) are. What are they? In the current OSPF database, which LSAs can we observe? Justify your answer explaining the database content of router 4.

4.3. Exercise 3: Designated Router and Backup Designated Router

In order to reduce the number of link state advertisement in an OSPF area, OSPF uses the concept of Designated Router (DR) and Backup Designated Router (BDR). An election process to choose DR and BDR happens in the Init phase during exchanging Hello packets [1]. Other routers establish a relationship named OSPF adjacency with DR/BDR, meanwhile, OSPF neighbor relationship is established among those non-DR/BDR routers. Therefore, all update information are sent to DR/BDR, and then the DR will be responsible to forward those update to others. With OSPF neighbors, they only exchange Hello packets

to see and understand each other. The BDR will replace the DR if it happens to be down or found as no longer connected.

An OSPF parameter called *priority* is used by OSPF in the process of DR selection. An interface on an OSPF router can be assigned a number of priority which varies from 1 to 255. By default, priority value is set to 1. If this value is set to 0, the router will not be eligible in the DR election process. The router with the highest priority will be elected as DR, and the second one will be BDR for the same segment. If priorities are tie, routers' ids (RID) will be considered.

In this exercise we will focus on Area 1 of the running topology (Figure 2) which is a multi access broadcast network to observe election process and how routers exchange their packets.

Question 4.14.

Using command `show ip ospf neighbor`, determine which router in the broadcast domain is DR, and which as BDR in area 1. Justify your answer.

We will now change something in the topology such as the cost of interface r2-eth4, as we have done in Section 4.1.2. We will then observe the exchanged packets.

8. Start a wireshark capture in R2
9. Change the cost of r2-eth4 interface

Question 4.15.

Observe the captured traffic. There are two destination IP addresses used (224.0.0.5 and 224.0.0.6). Explain how non-DR and non-BDR communicate with DR and BDR. Which type of OSPF messages do you see with each of the destinations? Between which routers are these messages exchanged?

5. Conclusion

Question 5.1.

OSPF is one of the most commonly used dynamic routing protocol within a single autonomous system. Compared to static route, what are the advantages and disadvantages of OSPF?

Question 5.2.

As we have seen in Section 4.2, OSPF uses area concept to scale well when network size increases. Explain why using areas makes the protocol scale well.

References

- [1] RFC2328 - OSPF version 2, [online] <https://tools.ietf.org/html/rfc2328>.
- [2] Quagga Documentation, [online] <https://www.nongnu.org/quagga/docs.html>.
- [3] Modes in Cisco router, [online] http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/configuration/guide/ffun_c/fcf001.html.
- [4] OSPF in Quagga, [online] <http://www.nongnu.org/quagga/docs/quagga.html#OSPFv2>.
- [5] RFC5185 - OSPF multi-area, [online] <https://tools.ietf.org/html/rfc5185#section-2.3>.
- [6] RFC3509 - OSPF area border router, [online] <https://tools.ietf.org/html/rfc3509>.
- [7] Mininet, An Instant Virtual Network on your Laptop (or other PC) [online] <https://mininet.org>
- [8] Computer Networks, Fifth Edition Andrew S. Tanenbaum, David J. Wetherall, Prentice Hall
- [9] Les réseaux, 4th edition, Guy Pujolle, EYROLLES