

# CSIS0234B Computer and Communication Networks (Class B)

## Reading for Tutorial 9

### Dynamic Routing with OSPF using Zebra

Creating routing tables manually is practical only if the network is small and seldom change in topology. In a larger or dynamic environment like the Internet, the tremendous amount of effort required makes it completely infeasible. Instead, the routers have to compute the routing table by themselves. Within an autonomous system of the Internet, an **Interior Gateway Protocol (IGP)** is used; among autonomous systems an **Exterior Gateway Protocol (EGP)** is used instead<sup>1</sup>. Commonly used routing protocols includes the Routing Information Protocol (RIP) version 1 and 2, RIPng (ng stands for “New Generation”, which supports IPv6), OSPF (Open Shortest Path First) protocol version 2 (for IPv4) and 3 (for IPv6), and BGP (“Border Gateway Protocol”) version 4. Apart from BGP, all are IGPs. We will focus on the OSPF algorithm, which uses the link state routing algorithm. It is a good idea to review the textbook or notes about how such algorithm works.

The Linux kernel does not implement any routing algorithm. Instead, a **routing server**, e.g., `routed` and `gated`, sends and receives information from the peer, and hence calculate the preferred routing table. The routing table is then installed into the kernel. In this tutorial, we get some experience on how dynamic routing behaves by configuring and looking at the operation of a routing server called GNU `zebra`<sup>2</sup>. Zebra is so versatile that all the routing protocols mentioned in the previous paragraph are implemented. The downside is that documentation is very thin, as it does not yet reach the 1.0 release (current release is 0.94).

#### 1. OSPF on broadcast networks

In the lecture we have examined the OSPF protocol in point-to-point networks. Its operation in broadcast networks like the Ethernet is similar. OSPF uses **one node to represent each broadcast network**. Link state routing is then used as if the network is completely point-to-point.

For this to work, one of the routers must act as the “node” representing the broadcast network. This responsibility is taken up by a **designated router** in the broadcast network: when a router starts up and finds that there is no adjacent router which is designated, or if the previous designated router (and “backup” designated router) fails, it tries to be a designated router. If multiple routers claim to be the designated router at the same time, an election takes place based on a priority chosen by the administrator (normally a more reliable router would be configured to be of a higher priority) and the router ID.

The other routers of the broadcast network are not directly “adjacent” to each other. They are only adjacent to the node representing the broadcast network, i.e., the designated router. So Hello packets and Link State Update packets are only forwarded to the designated router. In this way the amount of processing by each router in the network is reduced. In contrast, Hello packets and Link State Update packets of the designated router are **multicast** back to all the routers using the multicast address 224.0.0.5 (meaning “all OSPF routers within this LAN), thus reducing the network bandwidth consumption of the protocol.

---

<sup>1</sup>The most important difference between IGP and EGP is that while in IGP the primary concern is cost of routing, for EGP the primary concern is to find just any path (without a cycle) that the system administrator doesn't hate. It is because each AS may use a completely different metric, so there is no way to compare costs among ASes.

<sup>2</sup>Zebra is recently orphaned: the developers stopped working on it. But good open-source projects never die: another team has taken it up and continue development under the name “Quagga”. Expect newer releases under that name.

## 2. Zebra

Zebra is actually a suite of routing daemons rather than a single daemon. An administrator selects a set of routing protocols to use, and runs only the subset of routing daemons. E.g., to run only the OSPF protocol, one only needs to run the `ospfd` daemon. Such daemons do not install the found routes. Instead, the information is passed to a central daemon (called `zebra`), which combines the information obtained from different routing protocols and installs the routes.

### 2.1. Configuring Zebra

Before one sets up the daemons themselves, one must stop all interfaces. To avoid interfering with a working network configuration, `zebra` will not install routes on an interface which is already configured. So the complete network setup must be done by `zebra`. Luckily, the setup is not very complicated.

All these daemons are configured in the same way. One should start them (using “`service zebra start`”, “`service ospfd start`”, etc) after the configuration is done. The configuration files are stored in `/etc/zebra/xxx.conf`, where `xxx` is `zebra`, `ospfd`, etc. Once the service is running you can control the service and see the information about it by connecting to a TCP port for the daemon, i.e., `telnet localhost xxx`. If you’d like to shut down the server at a later time, just replace the `start` word of the command above by `stop`. If you restart the `zebra` server, you should also restart all the actual routing servers.

The main `zebra.conf` configuration file is responsible for setting up the hostnames, interfaces and IP addresses of the routers. All configuration files can also setup the password to use the (telnet-based) control interface. As an example,

```
hostname router01
password zebra
enable password secret
interface lo
    ip address 127.0.0.1/8
    no shutdown
interface eth0
    ip address 192.168.0.3/24
    multicast
    no shutdown
ip route 0.0.0.0/0 192.168.0.1
```

There are two passwords, the first one for the control interface (i.e., connected using “telnet”), and the second one for enabling commands which modifies the behaviour of the router. After that is the interface setup, and you will have one per interface available. Each will be assigned an address (including the number of bits for the network part). Then a “no shutdown” command will ensure the interface is up (in most cases, prefixing “no” in a command would undo the command). A “multicast” command enables multicasting over the network, which should always be done whenever you want to use OSPF (because it uses multicasting itself).

The last line shows how to add a “static route” for the router, i.e., a route that does not come from a routing daemon. It is possible to ask `zebra` to “redistribute” static routes of a router to the OSPF algorithm, as we will see later. However, a default route (one with 0-length network part like the above) is never redistributed. If desired you can break it into two parts so that such static routes can be distributed.

## 2.2. Configuring OSPF daemon

The `ospfd` daemon is configured in `ospfd.conf`. As in `zebra.conf`, you can use it to setup the password of the control interface. You should also use it to enable OSPF routing (by default it is disabled even after you execute the daemon, to be enabled in the control interface), and to tell which of the networks should OSPF be used, which must also specify what area your router belongs to. As an example,

```
password zebra
router ospf                ! enable OSPF routing
redistribute static        ! ask zebra to redistribute routes
network 160.80.40.20/26 area 0 ! use OSPF on the interface for this subnet,
                                ! which is in the backbone area
```

Note that for two routers in the same network to communicate routing information, they must both have the interface to that network setup for OSPF routing.

The configuration file can be used to modify many other parameters, including the “metric” of a network (i.e., cost to go through a network, by default 10), how network delay can be scaled to automatically compute a metric, the amount of time between Hello and Link State Update, authentication, etc. For details you can refer to the manual, which is available from the course web page, and can also be found in the file

`/usr/share/doc/zebra-0.93b/zebra.doc`

of the Redhat computers in our lab. Exactly the same commands can also be executed in the control interface of the OSPF daemon, to further configure `ospfd` when it is already running.

## 2.3. Showing OSPF daemon interface

Once OSPF is up and running, you can examine the current status of the router by using the control interface. Some of the more interesting commands are listed here:

- `show ip ospf`: show basic information about the router.
- `show ip ospf interface`: show information about all or a particular interface.
- `show ip ospf neighbor`: show information about all or a particular neighbour.
- `show ip ospf database`: show the link state database stored.
- `show ip ospf route`: show the routes installed by OSPF.