Open Shortest Path First - OSPF

OSPF maintains the view of the network topology conceptually as a directed graph. Each router represents a vertex in the graph. Each link between neighboring routers represents a unidirectional edge and each link has an associated weight (called cost) that is either automatically derived from its bandwidth or administratively assigned. Using the weighted topology graph, each router computes a shortest path tree (SPT) with itself as the root, and applies the results to build its forwarding table. The computation is generally referred to as SPF computation and the resultant tree as the SPF tree.

An LSA (link-state advertisement) is the fundamental quantum of information that OSPF routers exchange with each other. It seeds the graph building process on the node and triggers SPF computation. LSAs originated by a node are distributed to all the other nodes in the network through a mechanism called flooding. Flooding is done hop-by-hop. OSPF ensures reliability by using link state acknowledgement packets. The set of LSAs in a router’s memory is termed link-state database (LSDB), a representation of the network graph. Therefore, OSPF ensures a consistent view of LSDB on each node in the network in a distributed fashion (eventual consistency model); this is key to the protocol’s correctness.

This topic describes OSPFv2, which is a link-state routing protocol for IPv4. For IPv6 commands, refer to Open Shortest Path First v3 - OSPFv3

Contents

- Scalability and Areas
- Configure OSPFv2
  - Enable the OSPF and Zebra Daemons
  - Configure OSPF
  - Define Custom OSPF Parameters on the Interfaces
  - OSPF SPF Timer Defaults
  - Configure MD5 Authentication for OSPF Neighbors
- Scaling Tips
  - Summarization
  - Stub Areas
  - Multiple ospfd Instances
  - Auto-cost Reference Bandwidth
- Unnumbered Interfaces
- Apply a Route Map for Route Updates
- ECMP
- Topology Changes and OSPF Reconvergence
- Troubleshooting
- Related Information

Scalability and Areas

An increase in the number of nodes affects:

- Memory footprint to hold the entire network topology
- Flooding performance
- SPF computation efficiency

The OSPF protocol advocates hierarchy as a divide and conquer approach to achieve high scale. You can divide the topology into areas, resulting in a two-level hierarchy. Area 0 (or 0.0.0.0), called the backbone area, is the top level of the hierarchy. Packets traveling from one non-zero area to another must go through the backbone area. For example, you can divide the leaf-spine topology into the following areas:

- **Scalability and Areas**
- **Configure OSPFv2**
  - Enable the OSPF and Zebra Daemons
  - Configure OSPF
  - Define Custom OSPF Parameters on the Interfaces
  - OSPF SPF Timer Defaults
  - Configure MD5 Authentication for OSPF Neighbors
- **Scaling Tips**
  - Summarization
  - Stub Areas
  - Multiple ospfd Instances
  - Auto-cost Reference Bandwidth
- **Unnumbered Interfaces**
- **Apply a Route Map for Route Updates**
- **ECMP**
- **Topology Changes and OSPF Reconvergence**
- **Troubleshooting**
- **Related Information**
To configure OSPF, you need to:

- Enable the OSPF daemon
- Configure OSPF
- Define custom OSPF parameters on the interfaces

Enable the OSPF and Zebra Daemons

To enable OSPF, enable the `zebra` and `ospf` daemons, as described in Configuring FRRouting, then start the FRRouting service:

```bash
cumulus@switch:~$ sudo systemctl enable frr.service
cumulus@switch:~$ sudo systemctl start frr.service
```

Configure OSPF

Before you configure OSPF, you need to identify:

- Which router has the router ID
- With which device the router communicates
- What information to advertise (the prefix reachability)

To configure OSPF, specify the router ID, an IP subnet prefix, and an area address. All the interfaces on the router whose IP address matches the `network` subnet are put into the specified area. The OSPF process starts bringing up peering adjacency on those interfaces. It also advertises the interface IP addresses formatted into LSAs (of various types) to the neighbors for proper reachability.

The subnets can be as coarse as possible to cover the most number of interfaces on the router that should run OSPF.

```bash
cumulus@switch:~$ net add ospf router-id 0.0.0.1
cumulus@switch:~$ net add ospf network 10.0.0.0/16 area 0.0.0.0
cumulus@switch:~$ net add ospf network 192.0.2.0/16 area 0.0.0.1
```

You can configure OSPF per interface instead of using the `net add ospf network` command. However, you cannot use both configuration methods at the same time. Here is an example of configuring OSPF per interface:
There may be interfaces where it is undesirable to bring up OSPF adjacency. For example, in a data center topology, the host-facing interfaces do not need to run OSPF; however the corresponding IP addresses still need to be advertised to neighbors. In this scenario, you can use the `passive-interface` option.

```bash
cumulus@switch:~$ net add ospf passive-interface swp10
cumulus@switch:~$ net add ospf passive-interface swp11
```

As an alternative, you can use the `passive-interface default` command to put all interfaces as passive and selectively remove certain interfaces:

```bash
R3# configure terminal
R3(config)# router ospf
R3(config-router)# passive-interface default
R3(config-router)# no passive-interface swp1
```

To specify what information to advertise (the prefix reachability), you can use the redistribution method. Redistribution loads the database unnecessarily with type-5 LSAs. Only use this method to generate real external prefixes (for example, prefixes learned from BGP). Generate local prefixes using `network` and/or `passive-interface` statements.

```bash
cumulus@switch:~$ net add ospf redistribute connected
```

**Define Custom OSPF Parameters on the Interfaces**

You can define additional custom parameters for OSPF, such as the network type (point-to-point or broadcast) and the timer tuning, such as a hello interval.

```bash
cumulus@switch:~$ net add interface swp1 ospf network point-to-point
cumulus@switch:~$ net add interface swp1 ospf hello-interval 5
```

The OSPF configuration is saved in the `/etc/frr/frr.conf` file.

**OSPF SPF Timer Defaults**

OSPF uses the following timers to prevent consecutive SPFs from overburdening the CPU:

- 0 ms from initial event until SPF runs
- 50 ms between consecutive SPF runs (the number doubles with each SPF, until it reaches the value of C)
- 5000 ms maximum between SPFs

**Configure MD5 Authentication for OSPF Neighbors**

Simple text passwords have largely been deprecated in FRRouting, in favor of MD5 hash authentication.

To configure MD5 authentication:

1. Create a key and a key ID for MD5 with the `net add interface <interface> ospf message-digest-key <KEYID> md5 <KEY>` command.
**KEYID** is a value between 1 and 255 that represents the key used to create the message digest. This value must be consistent across all routers on a link.

**KEY** is a value that has an upper range of 16 characters (longer strings are truncated) and represents the actual message digest key that is associated with the given KEYID.

The following example command creates key ID 1 with the message digest key *thisisthekey*:

```
cumulus@switch:~$ net add interface swp1 ospf message-digest-key 1 md5 thisisthekey
```

You can remove existing MD5 authentication hashes with the `net del interface <interface> ospf message-digest-key <KEYID> md5 <KEY>` command.

---

2. **Enable authorization with the** `net add interface <interface> ospf authentication message-digest` command.

```
cumulus@switch:~$ net add interface swp1 ospf authentication
message-digest

cumulus@switch:~$ net pending

cumulus@switch:~$ net commit
```

These commands creates the following configuration in the `/etc/frr/frr.conf` file:

```
cumulus@switch:~$ sudo cat /etc/frr/frr.conf
...
interface swp1
  ip ospf area 0.0.0.0
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 thisisthekey
  ip ospf network point-to-point
...
```

---

**Scaling Tips**

Here are some tips for how to scale out OSPF.

**Summarization**

By default, an ABR creates a summary (type-3) LSA for each route in an area and advertises it in adjacent areas. Prefix range configuration optimizes this behavior by creating and advertising one summary LSA for multiple routes.

To configure a range:
cumulus@switch:~$ sudo vtysh
switch# configure terminal
switch(config)# router ospf
switch(config-router)# area 0.0.0.1 range 30.0.0.0/8
switch(config-router)# exit
switch(config)# exit
switch# write mem
switch# exit
cumulus@switch:~$

Summarize in the direction to the backbone. The backbone receives summarized routes and injects them to other areas already summarized.

Summarization can cause non-optimal forwarding of packets during failures. Here is an example scenario:

As shown in the diagram, the ABRs in the right non-zero area summarize the host prefixes as 10.1.0.0/16. When the link between R5 and R10 fails, R5 will send a worse metric for the summary route (metric for the summary route is the maximum of the metrics of intra-area routes that are covered by the summary route. Upon failure of the R5-R10 link, the metric for 10.1.2.0/24 goes higher at R5 as the path is R5-R9-R6-R10). As a result, other backbone routers shift traffic destined to 10.1.0.0/16 towards R6. This breaks ECMP and is an under-utilization of network capacity for traffic destined to 10.1.1.0/24.

Stub Areas

Nodes in an area receive and store intra-area routing information and summarized information about other areas from the ABRs. In particular, a good summarization practice about inter-area routes through prefix range configuration helps scale the routers and keeps the network stable.

Then there are external routes. External routes are the routes redistributed into OSPF from another protocol. They have an AS-wide flooding scope. In many cases, external link states make up a large percentage of the LSDB.

*Stub areas* alleviate this scaling problem. A stub area is an area that does not receive external route advertisements.

To configure a stub area:
cumulus@switch:~$ net add ospf area 0.0.0.1 stub

Stub areas still receive information about networks that belong to other areas of the same OSPF domain. Especially, if summarization is not configured (or is not comprehensive), the information can be overwhelming for the nodes. Totally stubby areas address this issue. Routers in totally stubby areas keep in their LSDB information about routing within their area, plus the default route.

To configure a totally stubby area:

cumulus@switch:~$ net add ospf area 0.0.0.1 stub no-summary

Here is a brief tabular summary of the area type differences:

<table>
<thead>
<tr>
<th>Type</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal non-zero area</td>
<td>LSA types 1, 2, 3, 4 area-scoped, type 5 externals, inter-area routes summarized</td>
</tr>
<tr>
<td>Stub area</td>
<td>LSA types 1, 2, 3, 4 area-scoped, No type 5 externals, inter-area routes summarized</td>
</tr>
<tr>
<td>Totally stubby area</td>
<td>LSA types 1, 2 area-scoped, default summary, No type 3, 4, 5 LSA types allowed</td>
</tr>
</tbody>
</table>

**Multiple ospfd Instances**

The ospfd daemon can have multiple independent processes.

- Multiple ospfd processes are only supported in the default VRF.
- You can run multiple ospfd instances with OSPFv2 only.
- FRRouting supports up to five instances and the instance ID must be a value between 1 and 65535.

To configure multi-instance OSPF:

1. Edit `/etc/frr/daemons` and add `ospfd_instances=instance1 instance2 ...` to the ospfd line, specifying an instance ID for each separate instance. For example, the following configuration has OSPF enabled with 2 ospfd instances, 11 and 22:

   ```
cumulus@switch:~$ cat /etc/frr/daemons
zebra=yes
bgpd=no
ospfd=yes ospfd_instances="11 22"
ospfd6d=no
ripd=no
ripngd=no
isisd=no
```

2. After you modify the daemons file, restart FRRouting:

   ```
cumulus@switch:~$ sudo systemctl restart frr.service
```

3. Configure each instance:
cumulus@switch:~$ net add interface swp1 ospf instance-id 11
cumulus@switch:~$ net add interface swp1 ospf area 0.0.0.0
cumulus@switch:~$ net add ospf router-id 1.1.1.1
cumulus@switch:~$ net add interface swp2 ospf instance-id 22
cumulus@switch:~$ net add interface swp2 ospf area 0.0.0.0
cumulus@switch:~$ net add ospf router-id 1.1.1.1

4. Confirm the configuration:

cumulus@switch:~$ net show configuration ospf

hostname zebra
log file /var/log/frr/zebra.log
username cumulus nopassword

service integrated-vtysh-config

interface eth0
  ipv6 nd suppress-ra
  link-detect

interface lo
  link-detect

interface swp1
  ip ospf 11 area 0.0.0.0
  link-detect

interface swp2
  ip ospf 22 area 0.0.0.0
  link-detect

interface swp45
  link-detect

interface swp46
  link-detect

interface swp47
  link-detect

interface swp48
  link-detect

interface swp49
  link-detect

interface swp50
  link-detect
interface swp51
    link-detect

interface swp52
    link-detect

interface vagrant
    link-detect

router ospf 11
    ospf router-id 1.1.1.1

router ospf 22
    ospf router-id 1.1.1.1

ip forwarding
ipv6 forwarding
5. Confirm that all the OSPF instances are running:

```
cumulus@switch:~$ ps -ax | grep ospf
21135 ? S<s 0:00 /usr/lib/frr/ospfd --daemon --A 127.0.0.1 -n
11
21139 ? S<s 0:00 /usr/lib/frr/ospfd --daemon --A 127.0.0.1 -n
22
21160 ? S<s 0:01 /usr/lib/frr/watchfrr --adz -r
/usr/sbin/servicebBfrrbBrestartbB%s -s
/usr/sbin/servicebBquaggabBstopbB%s -k
/usr/sbin/servicebBfrrbBstopbB%s -b bb -t 30 zebra ospfd-11 ospfd-22
pimd
22021 pts/3 S+ 0:00 grep ospf
```

Caveats

You can use the `redistribute ospf` option in your `frr.conf` file works with this so you can route between the instances. Specify the instance ID for the other OSPF instance. For example:

```
cumulus@switch:~$ cat /etc/frr/frr.conf
hostname zebra
log file /var/log/frr/zebra.log
username cumulus nopassword
!
service integrated-vtysh-config
!
...
!
router ospf 11
  ospf router-id 1.1.1.1
!
router ospf 22
  ospf router-id 1.1.1.1
  redistribute ospf 11
!
...
```

Don't specify a process ID unless you are using multi-instance OSPF.

If you disabled the `integrated` FRRouting configuration, you must create a separate `ospfd` configuration file for each instance. The `osp
cumulus@switch:~$ cat /etc/frr/ospfd-11.conf

hostname zebra
log file /var/log/frr/zebra.log
username cumulus nopassword

service integrated-vtysh-config

interface eth0
  ipv6 nd suppress-ra
  link-detect

interface lo
  link-detect

interface swp1
  ip ospf 11 area 0.0.0.0
  link-detect

interface swp2
  ip ospf 22 area 0.0.0.0
  link-detect

interface swp45
  link-detect

interface swp46
  link-detect

interface swp47
  link-detect

interface swp48
  link-detect

interface swp49
  link-detect

interface swp50
  link-detect

interface swp51
  link-detect

interface swp52
  link-detect

interface vagrant
  link-detect

router ospf 11
  ospf router-id 1.1.1.1
!
router ospf 22
  ospf router-id 1.1.1.1
!
ip forwarding
ipv6 forwarding
Auto-cost Reference Bandwidth

Auto-cost reference bandwidth provides the ability to dynamically calculate the OSPF interface cost to cater for higher speed links. You specify the bandwidth in Mbps and this value is used to calculate the link speed. The default value is 100000, for 100Gbps link speed. The cost of interfaces with link speeds lower than 100Gbps is higher.

It is a good idea to specify that the bandwidth setting should be a consistent value across all OSPF routers; otherwise routing loops can occur.

To configure the auto-cost reference bandwidth for 90Gbps, run the following commands:

```
cumulus@switch:~$ net add ospf auto-cost reference-bandwidth 90000
cumulus@switch:~$ net pending
cumulus@switch:~$ net commit
```

These commands create the following configuration in the /etc/frr/frr.conf file:

```
cumulus@switch:~$ cat /etc/frr/frr.conf
...
router ospf
 auto-cost reference-bandwidth 90000
...
```

Unnumbered Interfaces

Unnumbered interfaces are interfaces without unique IP addresses. In OSPFv2, configuring unnumbered interfaces reduces the links between routers into pure topological elements, which dramatically simplifies network configuration and reconfiguration. In addition, the routing database contains only the real networks, so the memory footprint is reduced and SPF is faster.

It is a good idea to specify that the bandwidth setting should be a consistent value across all OSPF routers; otherwise routing loops can occur.

Unnumbered is usable for point-to-point interfaces only.

If there is a network <network number>/<mask> area <area ID> command present in the FRRouting configuration, the ip ospf area <area ID> command is rejected with the error “Please remove network command first.” This prevents you from configuring other areas on some of the unnumbered interfaces. You can use either the network area command or the ospf area command in the configuration, but not both.

Unless the Ethernet media is intended to be used as a LAN with multiple connected routers, we recommend configuring the interface as point-to-point. It has the additional advantage of a simplified adjacency state machine; there is no need for DR/BDR election and LSA reflection. See RFC5309 for a more detailed discussion.
To configure an unnumbered interface, take the IP address of another interface (called the anchor) and use that as the IP address of the unnumbered interface:

```
cumulus@switch:~$ net add loopback lo ip address 192.0.2.1/32
cumulus@switch:~$ net add interface swp1 ip address 192.0.2.1/32
cumulus@switch:~$ net add interface swp2 ip address 192.0.2.1/32
```

These commands create the following configuration in the `/etc/network/interfaces` file:

```
auto lo
iface lo inet loopback
  address 192.0.2.1/32

auto swp1
iface swp1
  address 192.0.2.1/32

auto swp2
iface swp2
  address 192.0.2.1/32
```

To enable OSPF on an unnumbered interface:

```
cumulus@switch:~$ net add interface swp1 ospf area 0.0.0.1
```

### Apply a Route Map for Route Updates

To apply a route map to filter route updates from Zebra into the Linux kernel:

```
cumulus@switch:$ net add routing protocol ospf route-map <route-map-name>
```

### ECMP

During SPF computation for an area, if OSPF finds multiple paths with equal cost (metric), all those paths are used for forwarding. For example, in the reference topology diagram, R8 uses both R3 and R4 as next hops to reach a destination attached to R9.

### Topology Changes and OSPF Reconvergence

Topology changes usually occur due to one of four events:

1. Maintenance of a router node
2. Maintenance of a link
3. Failure of a router node
4. Failure of a link

For the maintenance events, operators typically raise the OSPF administrative weight of the link(s) to ensure that all traffic is diverted from the link or the node (referred to as holding out). The speed of reconvergence does not matter. Indeed, changing the OSPF cost causes LSAs to be reissued, but the links remain in service during the SPF computation process of all routers in the network.

For the failure events, traffic may be lost during reconvergence; that is, until SPF on all nodes computes an alternative path around the failed link.
or node to each of the destinations. The reconvergence depends on layer 1 failure detection capabilities and at the worst case DeadInterval OSPF timer.

**Example configuration for event 1, using vtysh:**

```bash
cumulus@switch:~$ sudo vtysh
switch# configure terminal
switch(config)# router ospf
switch(config-router)# max-metric router-lsa administrative
switch(config-router)# exit
switch(config)# exit
switch# write mem
switch# exit
cumulus@switch:~$
```

**Example configuration for event 2:**

```bash
cumulus@switch:~$ net add interface swp1 ospf cost 65535
```

**Troubleshooting**

- To debug neighbor states, run the `net show ospf neighbor` command.
- To capture OSPF packets, run the `sudo tcpdump -v -i swp1 ip proto ospf` command.
- To verify that the LSDB is synchronized across all routers in the network, run the `net show ospf database` command.
- To debug why an OSPF route is not being forwarded correctly, run the `net show route ospf` command. This command shows the outcome of the SPF computation downloaded to the forwarding table.

Debugging-OSPF lists all of the OSPF debug options.

**Related Information**

- Bidirectional forwarding detection (BFD) and OSPF
- [en.wikipedia.org/wiki/Open_Shortest_Path_First](en.wikipedia.org/wiki/Open_Shortest_Path_First)
- FRR OSPFv2
- RFC 2328 OSPFv2
- RFC 3101 OSPFv2 Not-So-Stubby Area (NSSA)