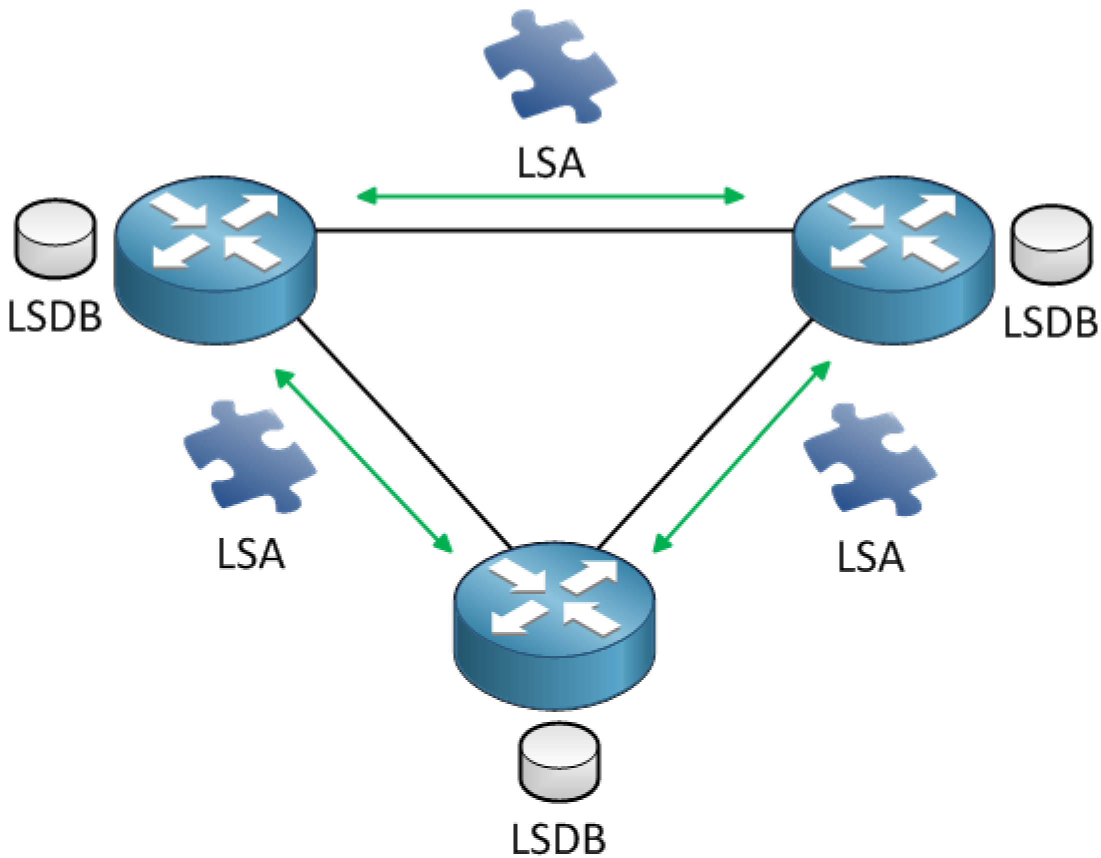
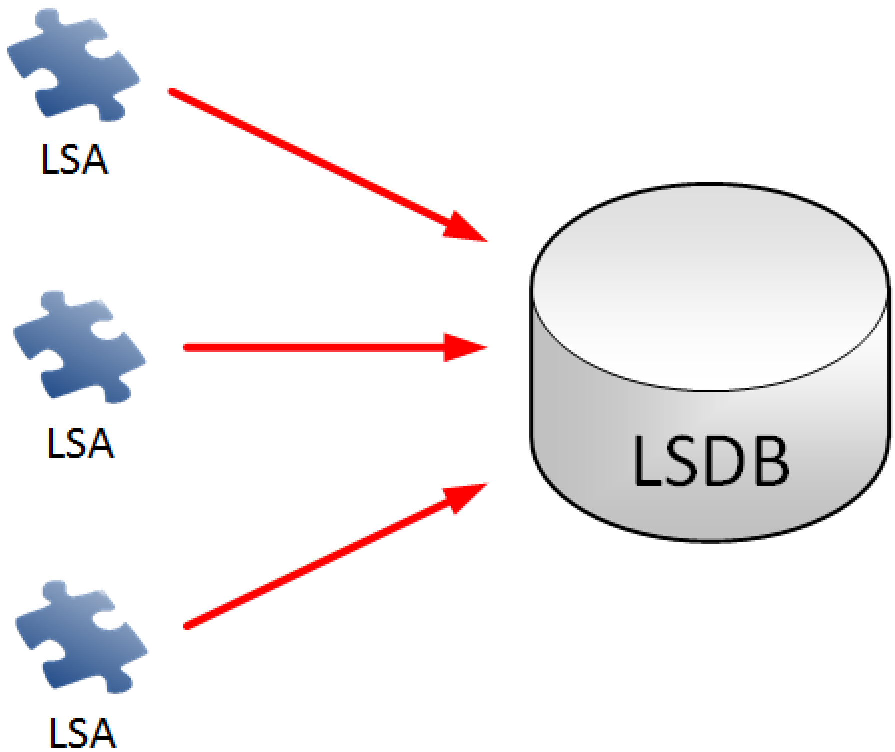
Introduction to OSPF

OSPF is a link-state routing protocol and it's one of the routing protocols you need to understand if you want to do the Cisco CCNA, CCNP or CCIE R&S exam(s).

Link-state routing protocols are like your navigation system, they have a complete map of the network. If you have a full map of the network you can just calculate the shortest path to all the different destinations out there. This is cool because if you know about all the different paths it's impossible to get a loop since you know everything! The downside is that this is more CPU intensive than a distance vector routing protocol. It's just like your navigation system…if you calculate a route from New York to Los Angeles it's going to take a bit longer than when you calculate a route from one street to another street in the same city.



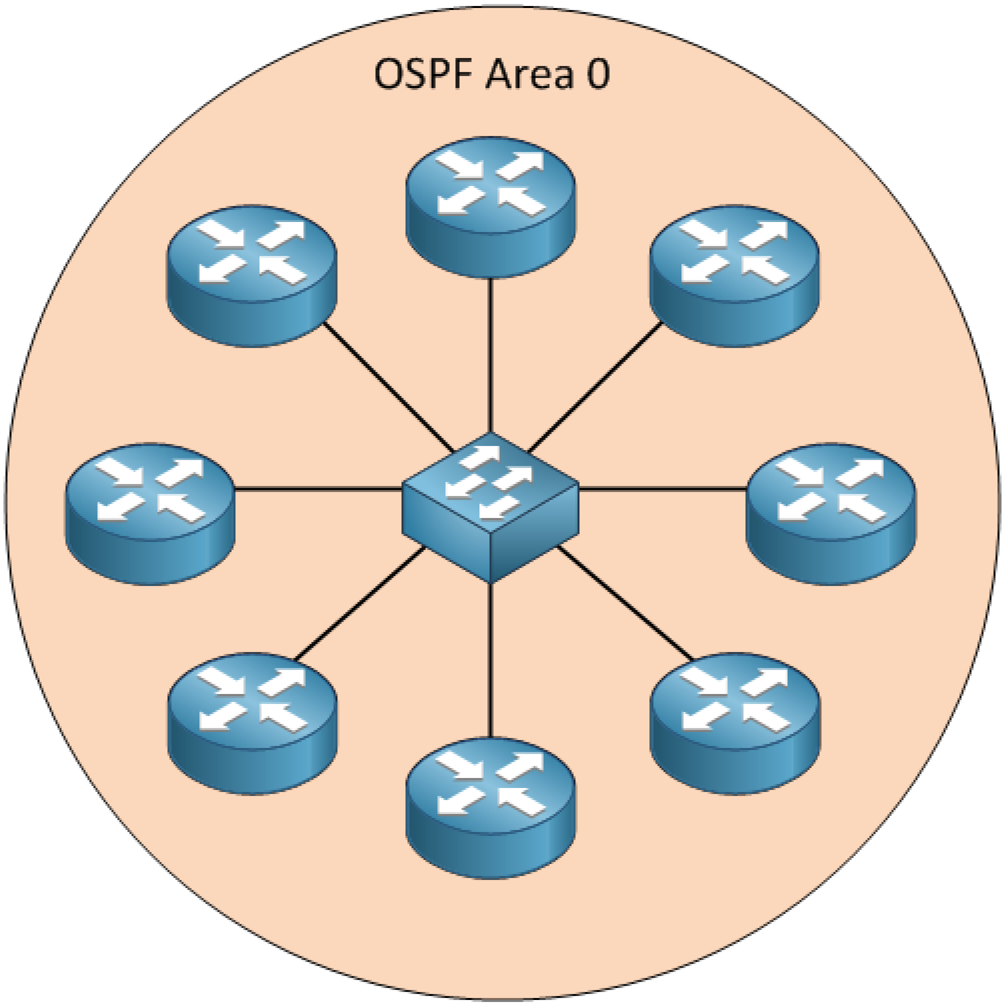
- Link: That's the interface of our router.
- State: Description of the interface and how it's connected to neighbor routers.



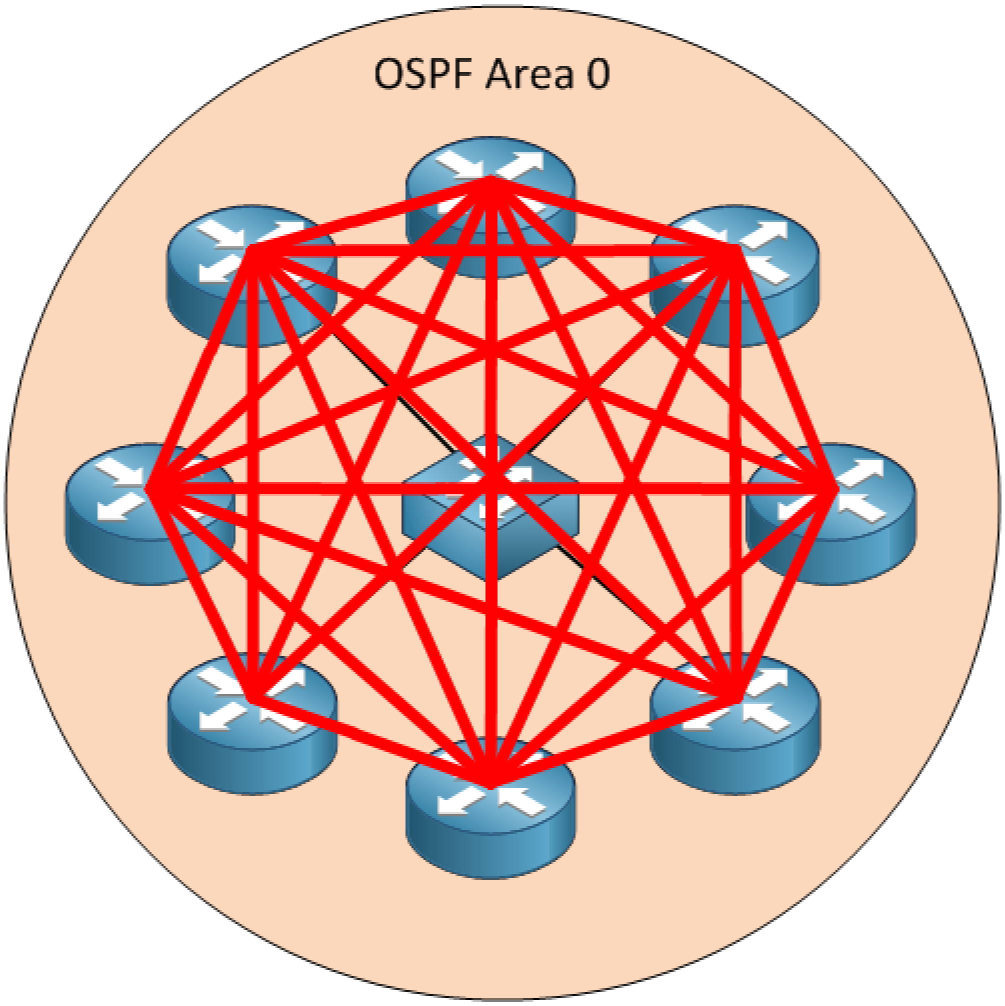
Link-state routing protocols operate by sending **link-state advertisements** (**LSA**) to all other link-state routers.

All the routers need to have these link-state advertisements so they can build their **link-state database** or **LSDB**. Basically all the link-state advertisements are a piece of the puzzle which builds the LSDB.

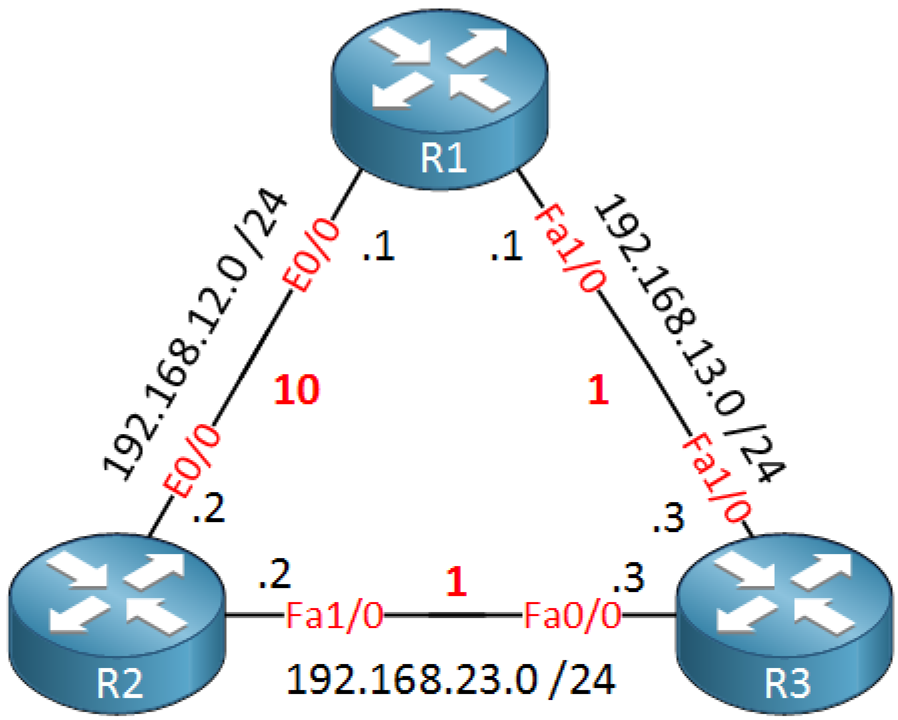
If you have a lot of OSPF routers it might not be very efficient that each OSPF router floods its LSAs to all other OSPF routers. Let me show you an example:

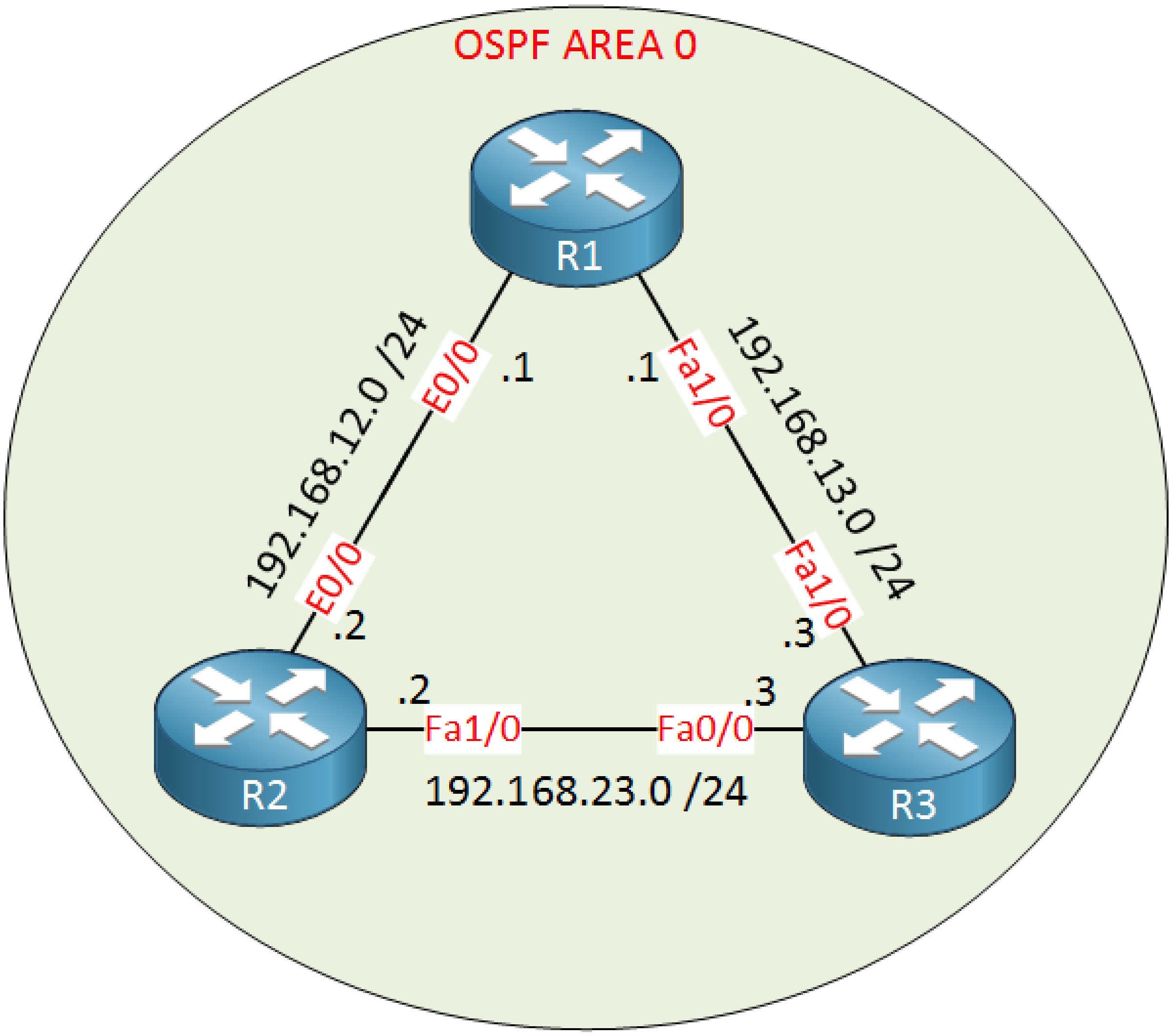


Above we have a network with 8 OSPF routers connected on a switch. Each of those routers is going to become OSPF neighbors with all of the other routers…sending hello packets, flooding LSAs and building the LSDB. This is what will happen:



Basic OSPF Configuration


R1
R2
R3
192.168.12.0 /24
192.168.13.0 /24
192.168.23.0 /24
E0/0
E0/0
Fa1/0
Fa1/0
Fa1/0
Fa0/0
.1
.1
.2
.2
.3
.3
10
1
1



This is the topology that we'll use. All routers are in OSPF Area 0. Note that the link between R2 and R1 is an Ethernet (10Mbit) link. All other links are FastEthernet (100Mbit) interfaces.

We'll start with the configuration between R2 and R3:

```
R2(config)#router ospf 1

R2(config-router)#network 192.168.23.0 0.0.0.255 area 0

R3(config)#router ospf 1

R3(config-router)#network 192.168.23.0 0.0.0.255 area 0
```

I need to use the router ospf command to get into the OSPF configuration. The number “1” is a process ID and you can choose any number you like. It doesn’t matter and if you want you can use a different number on each router.

The second step is to use the network command. It works similar to RIP but it is slightly different, let me break it down for you:

```
network 192.168.23.0 0.0.0.255
```

Just like RIP the network command does two things:

- Advertise the networks that fall within this range in OSPF.
- Activate OSPF on the interface(s) that fall within this range. This means that OSPF will send hello packets on the interface.

Behind 192.168.23.0 you can see it says 0.0.0.255. This is not a subnet mask but a **wildcard mask**. A wildcard mask is a **reverse subnet mask**. Let me give you an example:

Subnetmask	**255**	**255**	
	11111111	11111111	
Wildcardmask	**0**	**0**	
	00000000	00000000	

When I say reverse subnet mask I mean that the binary 1s and 0s of the wildcard mask are flipped compared to the subnet mask. A subnet mask of 255.255.255.0 is the same as wildcard mask 0.0.0.255. Don’t worry about this too much for now as I’ll explain wildcard masks to you when we talk about access-lists!

OSPF uses areas so you need to specify the area:

```
area 0
```

In our example we have configured single area OSPF. All routers belong to area 0.

After typing in my network command you’ll see this message in the console:

```
R3# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.23.2 on FastEthernet0/0 from LOADING to
FULL, Loading Done

R2# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.23.3 on FastEthernet1/0 from LOADING to
FULL, Loading Done
```

Great! It seems that R3 and R2 have become neighbors. There's another command we can use to verify that we have become neighbors:

```
R3#show ip ospf neighbor



Neighbor ID Pri State Dead Time Address Interface

192.168.23.2 1 FULL/BDR 00:00:36 192.168.23.2 FastEthernet0/0

R2#show ip ospf neighbor



Neighbor ID Pri State Dead Time Address Interface

192.168.23.3 1 FULL/DR 00:00:32 192.168.23.3 FastEthernet1/0
```

Show ip ospf neighbor is a great command to see if your router has OSPF neighbors. When the state is full you know that the routers have successfully become neighbors.

Each OSPF router has a router ID and we check it with the show ip protocols command:

```
R2#show ip protocols

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.23.2

R3#show ip protocols
```

```
Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.23.3
```

Above you see the router ID of R2 and R3. They used their highest active IP address as the router ID. Let's create a loopback on R2 to see if the router ID changes…

```
R2(config)#interface loopback 0

R2(config-if)#ip address 2.2.2.2 255.255.255.0
```

This is how you create a loopback interface. You can pick any number that you like it really doesn't matter.

```
R2#show ip protocols

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.23.2
```

The router ID still the same. We need to reset the OSPF process before the change will take effect, this is how you do it:

```
R2#clear ip ospf process

Reset ALL OSPF processes? [no]: yes
```

Use clear ip ospf process to reset OSPF. Let's see if there is a difference:

```
R2#show ip protocols
```

```
Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 2.2.2.2
```

We can also change the router ID manually. Let me demonstrate this on R3:

```
R3#show ip protocols

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 192.168.23.3
```

Right now it's 192.168.23.3…

```
R3(config-router)#router-id 3.3.3.3

Reload or use "clear ip ospf process" command, for this to take effect

R3#clear ip ospf process

Reset ALL OSPF processes? [no]: yes
```

The router is friendly enough to warn me to reload or clear the OSPF process. Let's verify our configuration:

```
R3#show ip protocols

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set
```

```
Router ID 3.3.3.3
```

As you can see above the router ID is now 3.3.3.3.

Changing the router ID isn't something you would normally do. IP addresses on your router have to be unique so your OSPF router ID will also be unique. Understanding how OSPF selects a router ID is something you have to understand for the exam however.

Right now we have an OSPF neighbor adjacency between R2 and R3. Let's configure our routers so that R2/R1 and R1/R3 also become OSPF neighbors:

```
R2(config)#router ospf 1

R2(config-router)#network 192.168.12.0 0.0.0.255 area 0

R1(config)#router ospf 1

R1(config-router)#network 192.168.12.0 0.0.0.255 area 0

R1(config-router)#network 192.168.13.0 0.0.0.255 area 0

R3(config)#router ospf 1

R3(config-router)#network 192.168.13.0 0.0.0.255 area 0
```

I'll advertise all networks in OSPF. Before we check the routing table it's a good idea to see if our routers have become OSPF neighbors:

```
R2#show ip ospf neighbor



Neighbor ID   Pri   State     Dead Time   Address        Interface

192.168.13.1    1   FULL/BDR  00:00:31    192.168.12.1   Ethernet0/0

3.3.3.3         1   FULL/DR   00:00:38    192.168.23.3   FastEthernet1/0

R1#show ip ospf neighbor
```

```
Neighbor ID    Pri   State      Dead Time   Address        Interface

3.3.3.3          1   FULL/BDR   00:00:33    192.168.13.3   FastEthernet1/0

2.2.2.2          1   FULL/DR    00:00:30    192.168.12.2   Ethernet0/0

R3#show ip ospf neighbor



Neighbor ID    Pri   State      Dead Time   Address          Interface

192.168.13.1     1   FULL/DR    00:00:37    192.168.13.1     FastEthernet1/0

2.2.2.2          1   FULL/BDR   00:00:30    192.168.23.2     FastEthernet0/0
```

Excellent our routers have become OSPF neighbors and the state is full which means they are done exchanging information. Let’s check the routing tables:

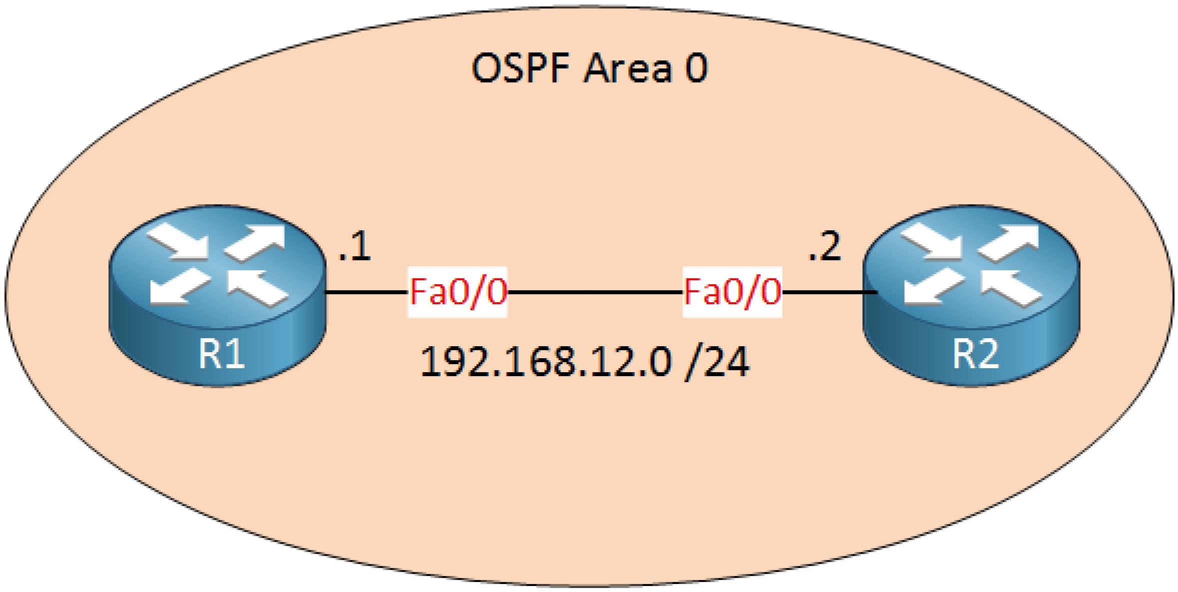
```
R2#show ip route ospf

O    192.168.13.0/24 [110/2] via 192.168.23.3, 00:09:45, FastEthernet1/0
```

OSPF Plain Text Authentication

All routing protocols can be protected by using authentication and OSPF is no exception. There are two options for authentication:

- Plain text authentication
- MD5 authentication

Each OSPF packet will be authenticated if you enable any form of authentication. In this lesson we’ll take a look at how to configure plain text authentication for OSPF. Here’s the topology that we’ll use:



Above you see the topology I'm going to use for authentication. Just two routers but we can use it to check all different methods of authentication.

```
R1(config)#router ospf 1

R1(config-router)#network 192.168.12.0 0.0.0.255 area 0

R2(config)#router ospf 1

R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
```

First I'll enable OSPF, nothing fancy here.

```
R1(config)#interface fastEthernet 0/0

R1(config-if)#ip ospf authentication

R1(config-if)#ip ospf authentication-key MYPASS

R2(config)#interface fastEthernet 0/0

R2(config-if)#ip ospf authentication

R2(config-if)#ip ospf authentication-key MYPASS
```

By using the **ip ospf authentication** command we enable plain text authentication on the interface level. I configured the password MYPASS by using the **ip ospf authentication-key** command.

```
R1(config)#router ospf 1

R1(config-router)#area 0 authentication
```

If you have a lot of interfaces you probably don't want to enable OSPF authentication for each interface. You can also enable area-wide authentication by using the **area authentication** command. In my example above I enabled authentication for area 0.

```
R1#show ip ospf interface fastEthernet 0/0

FastEthernet0/0 is up, line protocol is up

  Internet Address 192.168.12.1/24, Area 0

  Process ID 1, Router ID 192.168.12.1, Network Type BROADCAST, Cost: 1

  Transmit Delay is 1 sec, State BDR, Priority 1

  Designated Router (ID) 192.168.12.2, Interface address 192.168.12.2

  Backup Designated router (ID) 192.168.12.1, Interface address 192.168.12.1

  Flush timer for old DR LSA due in 00:01:49

  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

    oob-resync timeout 40

    Hello due in 00:00:01

  Supports Link-local Signaling (LLS)

  Index 1/1, flood queue length 0

  Next 0x0(0)/0x0(0)

  Last flood scan length is 1, maximum is 1

  Last flood scan time is 0 msec, maximum is 0 msec
```

```
Neighbor Count is 1, Adjacent neighbor count is 1

  Adjacent with neighbor 192.168.12.2  (Designated Router)

Suppress hello for 0 neighbor(s)

Simple password authentication enabled
```

If you use the **show ip ospf interface** command you can see OSPF information per interface. You can also check if authentication is enabled. You can see the neighbor count is 1 and simple password authentication is enabled.