OSPF Router ID

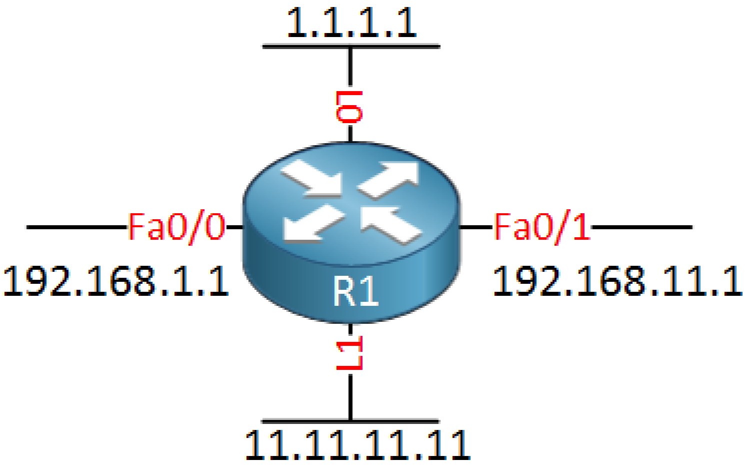
Each OSPF router selects a router ID (RID) that has to be unique on your network. OSPF stores the topology of the network in its LSDB (Link State Database) and each router is identified with its unique router ID , if you have duplicate router IDs then you will run into reachability issues.

Because of this, two OSPF routers with the same router ID will not become neighbors but you could still have duplicated router IDs in the network with routers that are not directly connected to each other.

OSPF uses the following criteria to select the router ID:

1. Manual configuration of the router ID.
2. Highest IP address on a loopback interface.
3. Highest IP address on a non-loopback interface.

Let’s see this in action. I will use the following router for this demonstration:



There are two physical interfaces and two loopback interfaces. All interfaces are active:

```
R1#show ip interface brief

Interface                  IP-Address      OK? Method Status                Protocol

FastEthernet0/0            192.168.1.1     YES manual up                    up

FastEthernet0/1            192.168.11.1    YES manual up                    up
```

```
Loopback0                  1.1.1.1         YES manual up                    up

Loopback1                  11.11.11.11     YES manual up                    up
```

Let’s start an OSPF process:

```
R1(config)#router ospf 1

R1(config-router)#exit
```

Now we can check what router ID it selected:

```
R1#show ip protocols | include Router ID

  Router ID 11.11.11.11
```

It selected 11.11.11.11 which is the highest IP address on our loopback interfaces. Let’s get rid of the loopbacks now:

```
R1(config)#no interface loopback 0

R1(config)#no interface loopback 1
```

Take a look again at the router ID:

```
R1#show ip protocols | include Router ID

  Router ID 11.11.11.11
```

It’s still the same, this is because the router ID selection is only done once. You have to reset the OSPF process before it will select another one:

```
R1#clear ip ospf process

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

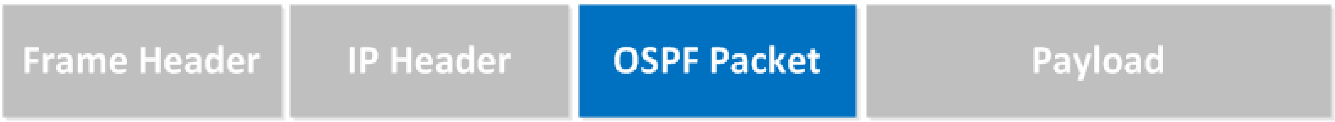
Let’s see if this makes any difference:

```
R1#show ip protocols | include Router ID

  Router ID 192.168.11.1
```

There we go, the router ID is now the highest IP address of our physical interfaces. If you want we can manually set the router ID. This will overrule everything:

OSPF Packets and Neighbor Discovery

In this lesson I'm going to show you the different packets OSPF uses and how neighbor discovery works.



OSPF uses its own protocol like EIGRP and doesn't use a transport protocol like TCP or UDP. If you would look at the IP packet in wireshark you can see that OSPF has **protocol ID 89** for all its packets.

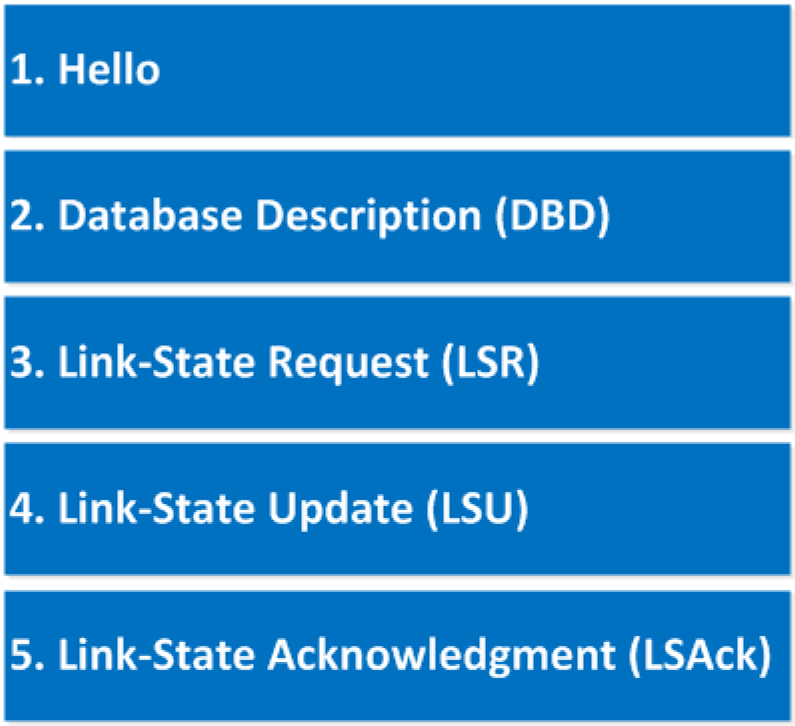
```
R2#debug ip ospf packet

OSPF packet debugging is on

OSPF: rcv. v:2 t:1 l:48 rid:1.1.1.1

      aid:0.0.0.0 chk:4D40 aut:0 auk: from FastEthernet0/0
```

If we use **debug ip ospf packet** we can look at the OSPF packet on our router. Let's look at the different fields we have:

- **V:2** stands for OSPF version 2. If you are running IPv6 you'll version 3.
- **T:1** stands for OSPF packet number 1 which is a hello packet. I'm going to show you the different packets in a bit.
- **L:48** is the packet length in bytes. This hello packet seems to be 48 bytes.
- **RID 1.1.1.1** is the Router ID.
- **AID** is the area ID in dotted decimal. You can write the area in decimal (area 0) or dotted decimal (area 0.0.0.0).
- **CHK 4D40** is the checksum of this OSPF packet so we can check if the packet is corrupt or not.
- **AUT:0** is the authentication type. You have 3 options:
  - 0 = no authentication

    - 1 = clear text
    - 2 = MD5
    - **AUK:** If you enable authentication you’ll see some information here.

Let’s continue by looking at the different OSPF packet types:



1. Hello

2. Database Description (DBD)

3. Link-State Request (LSR)

4. Link-State Update (LSU)

5. Link-State Acknowledgment (LSAck)

I’m throwing them right at you; here are all the **OSPF packet types** we have. In my **debug ip ospf packet** at the previous page you could see T:1 which stands for packet type 1. Here you see that it corresponds to an OSPF hello packet. What is the role of each OSPF packet?

- **Hello:** neighbor discovery, build neighbor adjacencies and maintain them.
- **DBD:** This packet is used to check if the LSDB between 2 routers is the same. The DBD is a **summary of the LSDB**.
- **LSR:** Requests specific link-state records from an OSPF neighbor.
- **LSU:** Sends specific link-state records that were requested. This packet is like an envelope with multiple LSAs in it.
- **LSAck:** OSPF is a reliable protocol so we have a packet to acknowledge the others.

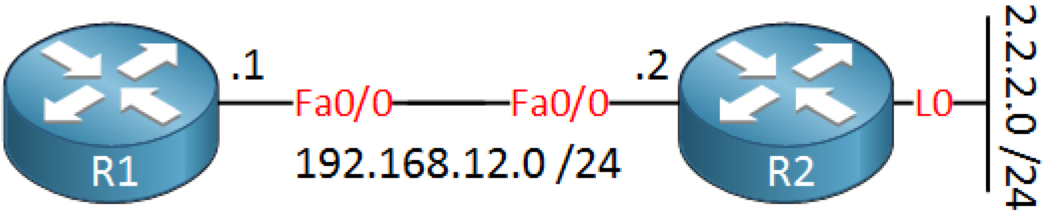
OSPF has to get through 7 states in order to become neighbors…here they are:

1. **Down**: no OSPF neighbors detected at this moment.
2. **Init:** Hello packet received.
3. **Two-way:** own router ID found in received hello packet.
4. **Exstart:** master and slave roles determined.
5. **Exchange:** database description packets (DBD) are sent.

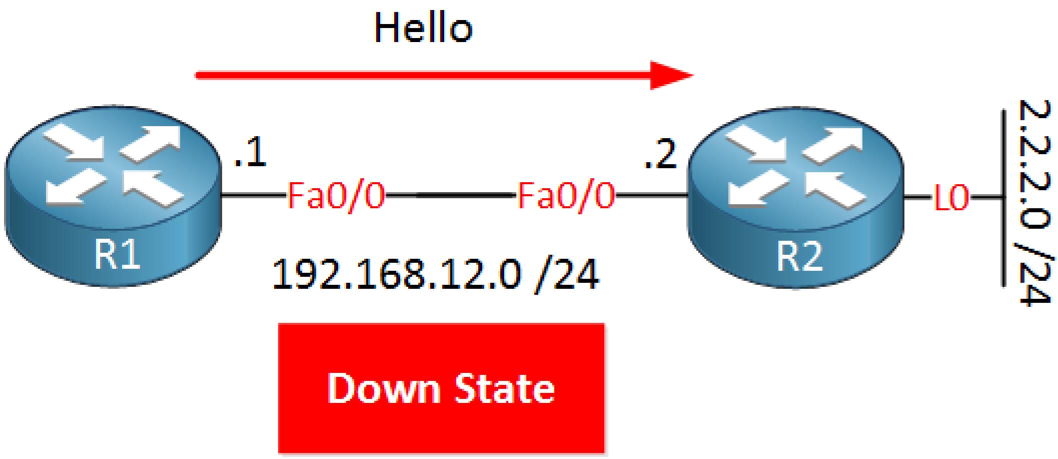
6. **Loading:** exchange of LSRs (Link state request) and LSUs (Link state update) packets.
7. **Full:** OSPF routers now have an adjacency.

Let's have a detailed look at this process! You can also see it in this packet capture:

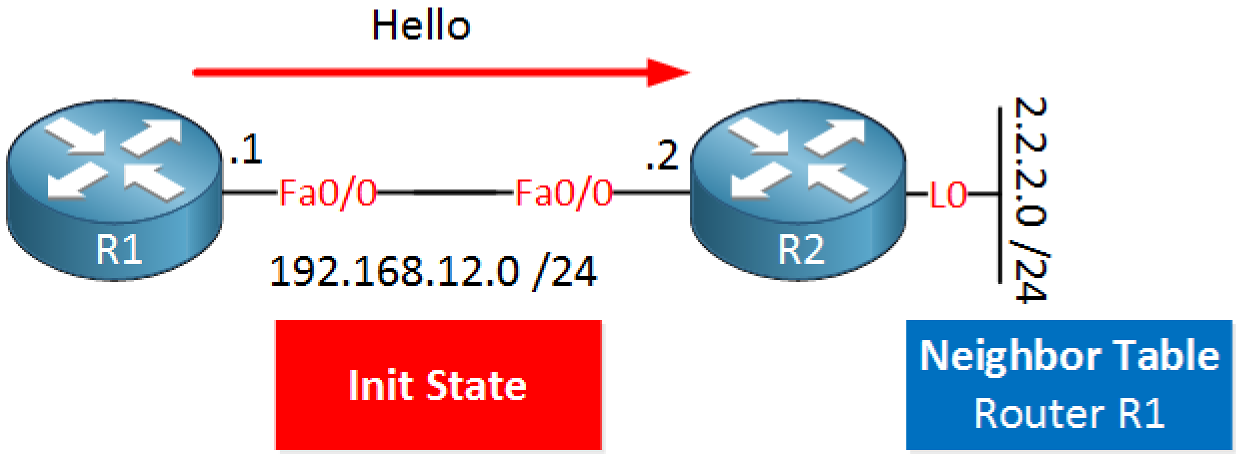
OSPF Neighbor Adjacency – Network Type Broadcast



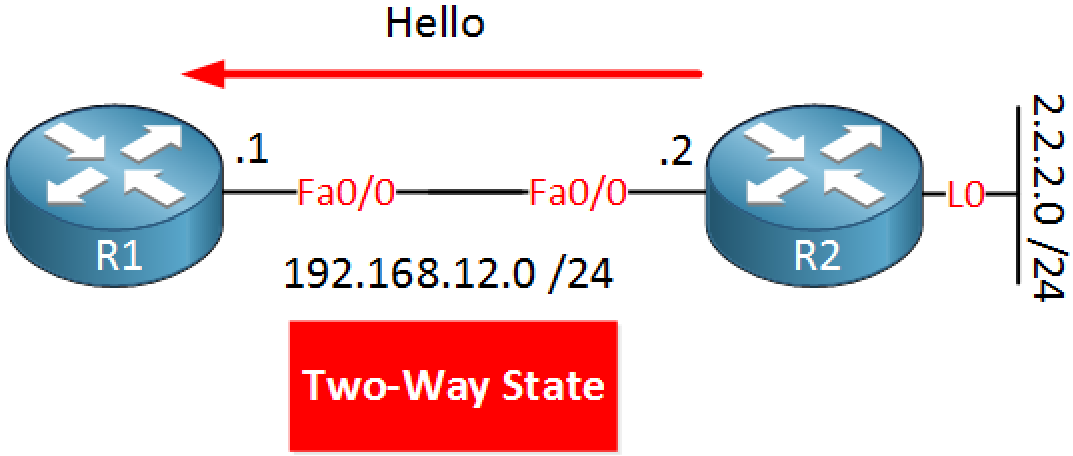
This is the topology I'm using. R1 and R2 are connected using a single link and we will see how R1 learns about the 2.2.2.0 /24 network.



As soon as I configure OSPF on R1 it will start sending hello packets. R1 has no clue about other OSPF routers at this moment so it's in the **down state**. The hello packet will be sent to the **multicast address 224.0.0.5.**



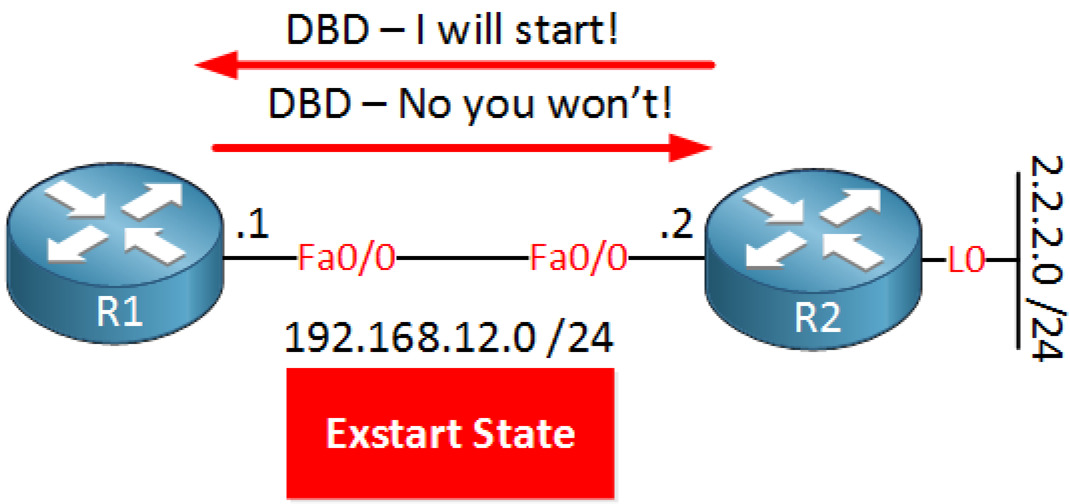
R2 receives the hello packet and will put an entry for R1 in the **OSPF neighbor table.** We are now in the **init state.**



R2 has to respond to R1 with a hello packet. This packet is not sent using multicast but with **unicast** and in the neighbor field it will include **all OSPF neighbors** that R2 has. R1 will see **its own name** in the neighbor field in this hello packet.

R1 will receive this hello packet and sees its own router ID. We are now in the **two-way state**.

I have to take a pause here. If the link we are using is a **multi-access network** OSPF has to elect a **DR (Designated Router)** and **BDR (Backup Designated Router)**. This has to happen before we can continue with the rest of the process. In another lesson I'm going to teach you DR/BDR…for now just hold the thought that the DR/BDR election happens right after the two-way state ok?

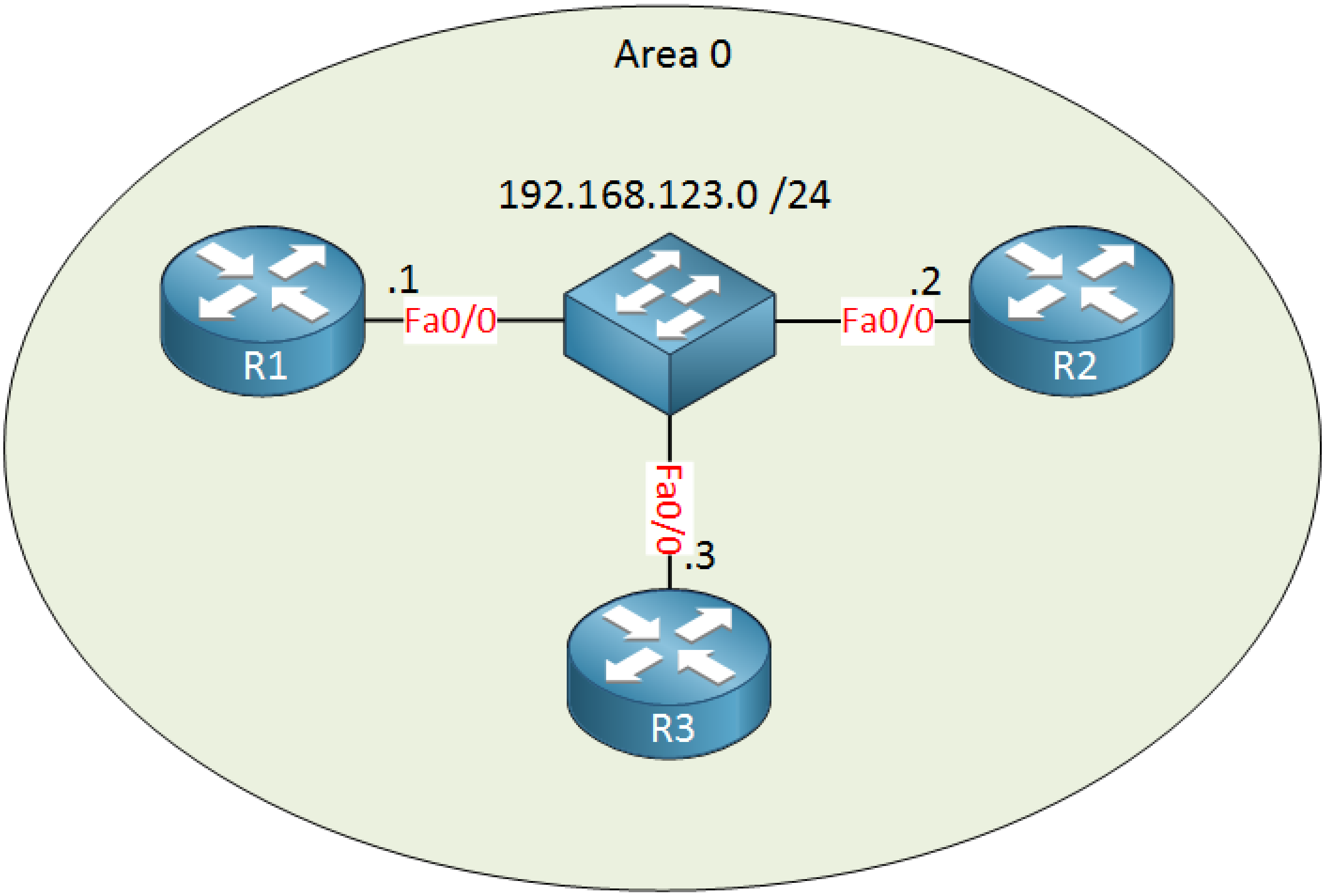


Our next stop is the **exstart state**. Our routers are ready to sync their LSDB. At this step we have to select a **master** and **slave** role. The router with the highest router ID will become the master. R2 has the highest router ID and will become the master.

OSPF DR/BDR Election explained

OSPF uses a DR (Designated Router) and BDR (Backup Designated Router) on each multi-access network. A multi-access network is a segment where we have more than two routers. OSPF figures this out by looking at the interface type. For example, an Ethernet interface is considered a multi-access network, and a serial interface is considered a point-to-point interface.

Most CCNA students think that this DR/BDR election is done per area but this is **incorrect**. I'll show you how the election is done and how you can influence it. This is the topology we'll use:



Here’s an example of a network with 3 OSPF routers on a FastEthernet network. They are connected to the same switch (multi-access network) so there will be a DR/BDR election. OSPF has been configured so all routers have become OSPF neighbors, let’s take a look:

```
R1#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address         Interface

192.168.123.2     1   FULL/BDR        00:00:32    192.168.123.2 FastEthernet0/0

192.168.123.3     1   FULL/DR         00:00:31    192.168.123.3 FastEthernet0/0
```

From R1 perspective, R2 is the BDR and R3 is the DR.

```
R3#show ip ospf neighbor
```

```


Neighbor ID     Pri   State         Dead Time   Address         Interface

192.168.123.1   1    FULL/DROTHER  00:00:36    192.168.123.1 FastEthernet0/0

192.168.123.2   1    FULL/BDR      00:00:39    192.168.123.2 FastEthernet0/0
```

When a router is not the DR or BDR it's called a **DROTHER**. I have no idea if we have to pronounce it like "BROTHER with a D" or "DR-OTHER" ☺ Here we can see that R1 is a DROTHER.

```
R2#show ip ospf neighbor



Neighbor ID     Pri   State         Dead Time   Address         Interface

192.168.123.1   1    FULL/DROTHER  00:00:31    192.168.123.1 FastEthernet0/0

192.168.123.3   1    FULL/DR       00:00:32    192.168.123.3 FastEthernet0/0
```

And R2 (the BDR) sees the DR and DROTHER.

Of course we can change which router becomes the DR/BDR by playing with the priority. Let's turn R1 in the DR:

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

R1(config-if)#ip ospf priority 200
```

You change the priority if you like by using the **ip ospf priority** command:

- The default priority is 1.
- A priority of 0 means you will never be elected as DR or BDR.
- You need to use **clear ip ospf process** before this change takes effect.

```
R1#show ip ospf neighbor
```

```
Neighbor ID     Pri   State        Dead Time   Address         Interface

192.168.123.2   1   FULL/BDR       00:00:31    192.168.123.2 FastEthernet0/0

192.168.123.3   1   FULL/DR        00:00:32    192.168.123.3 FastEthernet0/0
```

As you can see R3 is still the DR, we need to reset the OSPF neighbor adjacencies so that we'll elect the new DR and BDR.

```
R3#clear ip ospf process

Reset ALL OSPF processes? [no]: yes

R2#clear ip ospf process

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

I'll reset all the OPSF neighbor adjacencies.

```
R1#show ip ospf neighbor



Neighbor ID     Pri   State        Dead Time   Address         Interface

192.168.123.2   1   FULL/DROTHER   00:00:36    192.168.123.2 FastEthernet0/0

192.168.123.3   1   FULL/BDR       00:00:30    192.168.123.3 FastEthernet0/0
```

Now you can see R1 is the DR because the other routers are DROTHER and BDR.

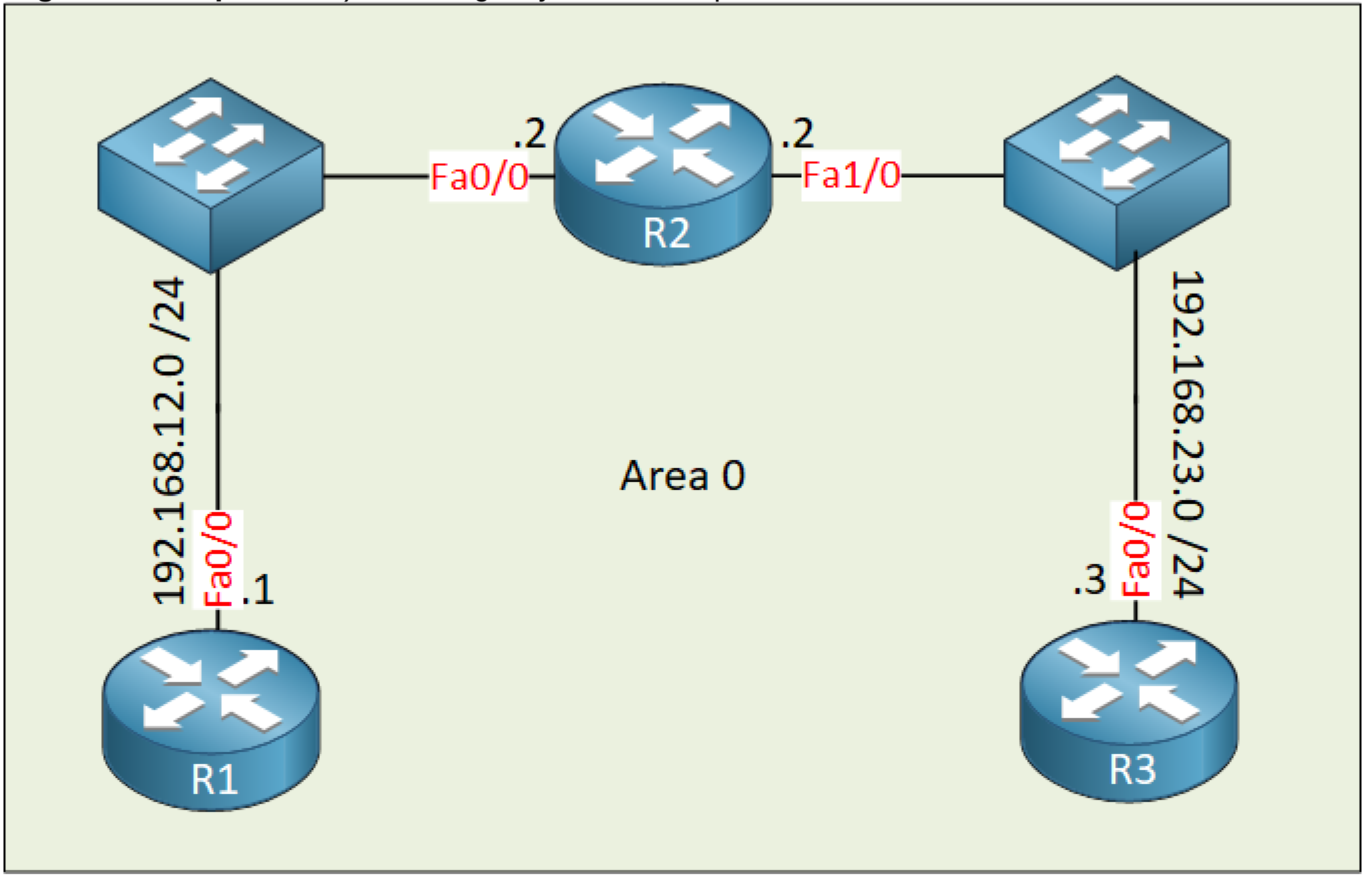
```
R3#show ip ospf neighbor

Neighbor ID     Pri  State         Dead Time   Address         Interface

192.168.123.1   200  FULL/DR       00:00:30    192.168.123.1 FastEthernet0/0

192.168.123.2   1    FULL/DROTHER  00:00:31    192.168.123.2 FastEthernet0/0
```

Or we can confirm it from R3, you'll see that R1 is the DR and that the priority is 200.

- Configurations
- R1
- R2
- R3

Want to take a look for yourself? Here you will find the startup configuration of each device.

Something you need to be aware of is that the **DR/BDR election is per multi-access segment…not per area!**). Let me give you an example:



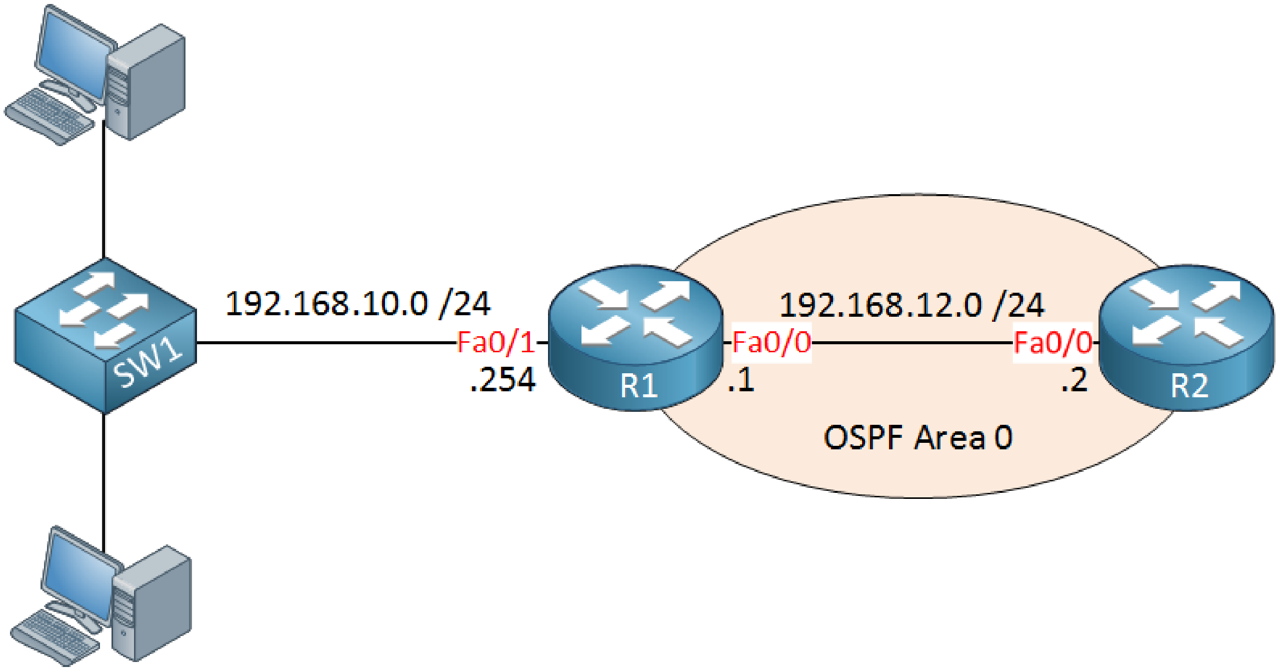
In the example above we have 2 multi-access segments. Between R2 and R1, and between R2 and R3. For each segment, there will be a DR/BDR election.

OSPF Passive Interface

When you use the network command in OSPF, two things will happen:

- All interfaces that have a network that falls within the range of the network command will be advertised in OSPF.
- OSPF hello packets are sent on these interfaces.

Sometimes it's undesirable to send OSPF hello packets on certain interfaces. Take a look at the image below:



R1 and R2 are configured for OSPF. R1 is connected to network 192.168.10 /24 which has some computers connected to a switch. R1 wants to advertise this network to R2.

Once we use the network command to advertise 192.168.10.0 /24 in OSPF, R1 will also send OSPF hello packets towards the switch. This is a bad idea, first of all because there are no routers on this network but it's also a security risk. If someone on the computer starts an application that replies with OSPF hello packets then R1 will try to become neighbors. An attacker could advertise fake routes using this technique.

To prevent this from happening, we can use the **passive-interface** command. This command tells OSPF not to send hello packets on certain interfaces. Let's see how it works…

Configuration

Here’s the OSPF configuration of R1 and R2:

```
R1(config)#router ospf 1

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

R1(config-router)#network 192.168.10.0 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
```

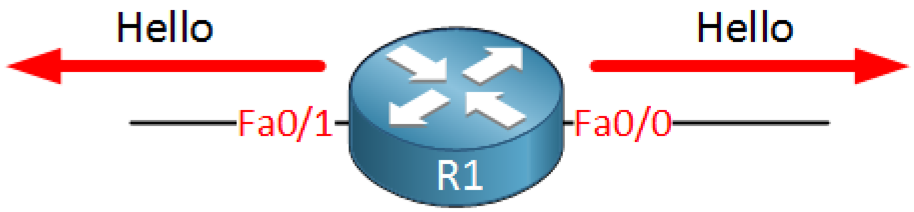
With the above configuration, R2 will learn network 192.168.10.0 /24:

```
R2#show ip route ospf

O    192.168.10.0/24 [110/20] via 192.168.12.1, 00:03:21, FastEthernet0/0
```

This is great but a side-effect of this configuration is that R1 will send hello packets on its FastEthernet 0/1 interface. We can see this with a debug:

```
R1#debug ip ospf hello

OSPF hello events debugging is on



OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/1 from 192.168.10.254



OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 192.168.12.1
```

Above you can see that hello packets are sent in both directions.



Let’s fix this. We will configure OSPF to stop the hello packets towards the switch:

```
R1(config)#router ospf 1

R1(config-router)#passive-interface FastEthernet 0/1
```

You only have to use the **passive-interface** command under the OSPF process. You can verify our work with the following command: