

Troubleshooting OSPF Neighbor Adjacency

OSPF is unlike EIGRP a link-state protocol but what they share in common is that both routing protocols establish a neighbor adjacency before exchanging routing information. In the case of OSPF we are exchanging LSAs (Link State Advertisement) in order to build the LSDB (Link State Database). The best information from the LSDB will be copied to the routing table.

In this lesson we'll troubleshoot the OSPF neighbor adjacency. Once we have a working OSPF neighbor adjacency we can look at other issues like [missing routes](#).

When looking at the OSPF neighbor adjacency, this is what we like to see:

```
R1#
```

```
%OSPF-5-ADJCHG: Process 1, Nbr 192.168.12.2 on FastEthernet0/0 from LOADING to FULL, Loading Done
```

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

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.12.2	1	FULL/DR	00:00:39	192.168.12.2	FastEthernet0/0

Typically we want to see “full” here when we look at our neighbors.

There is one exception to this rule: When we use a DR/BDR then two “normal” (non-DR or non-BDR) OSPF routers will not form a full adjacency with each other. They will remain in the 2-way state.

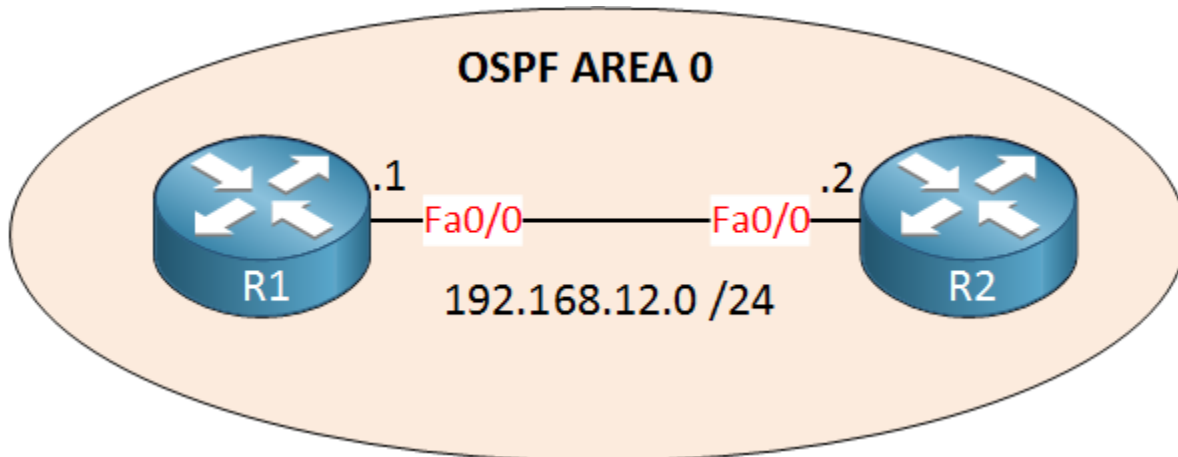
When the OSPF neighbor adjacency is not in the full state then it's in one of the other states:

- There's no OSPF neighbor at all.
- It's stuck in ATTEMPT.
- It's stuck in INIT.
- It's stuck in 2-WAY.
- It's stuck in EXSTART/EXCHANGE.
- It's stuck in LOADING.

Let's take a look at some of the different issues why we can't form an OSPF neighbor adjacency.

OSPF Network Command

Let's start with our first scenario. We have two OSPF routers:



Unfortunately they don't become neighbors:

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

As you can see we don't have any OSPF neighbors...so what could be wrong? Let's take a look:

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

```
%OSPF: OSPF not enabled on FastEthernet0/0
```

```
R2#show ip ospf interface fastEthernet 0/0
```

```
FastEthernet0/0 is up, line protocol is up
```

```
Internet Address 192.168.12.2/24, Area 0
```

```
Process ID 1, Router ID 192.168.12.2, Network Type BROADCAST, Cost: 1
```

I could just look at the running-configuration and see what's wrong but I want to show you some other useful OSPF commands. First I'll use the **show ip ospf interface** command.

We can see that OSPF is not enabled on the FastEthernet 0/0 interface of R1 but it's running on R2. Now let's take a look at the configuration of R1:

```
R1#show run | section ospf

router ospf 1

  log-adjacency-changes

  network 192.168.21.0 0.0.0.255 area 0
```

Someone made a mistake with the network command and typed in the wrong network address...silly mistake but stuff like this happens. Let's fix it:

```
R1(config)#router ospf 1

R1(config-router)#no network 192.168.21.0 0.0.0.255 area 0

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

Configuring the correct network address and wildcard should do the job. Let's see if it helped:

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

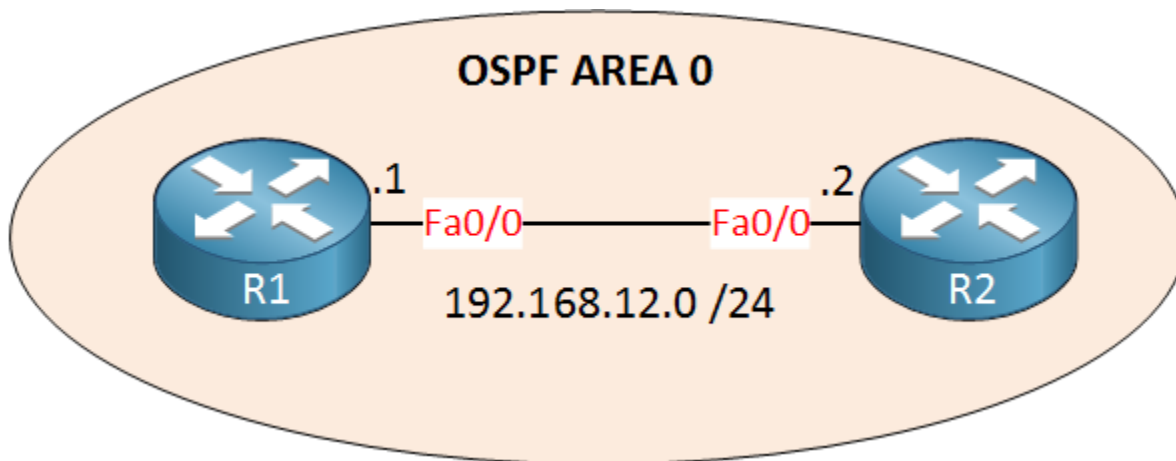
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.12.2	1	FULL/DR	00:00:31	192.168.12.2	FastEthernet0/0

Problem solved, we now have a working OSPF neighbor adjacency. This was an easy one to get you started...

Lesson learned: Make sure you have the configured the correct network address, wildcard bits and area.

OSPF Passive Interface

Let's look at the next issue, same two routers but another problem.



Let's check it out:

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

As you can see there is no OSPF neighbor. Let's take a look at the OSPF specific interface information:

```
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 WAITING, Priority 1

No designated router on this network

No backup designated router on this network

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

oob-resync timeout 40

No Hellos (Passive interface)

```
R2#show ip ospf interface fastEthernet 0/0
```

FastEthernet0/0 is up, line protocol is up

```
Internet Address 192.168.12.2/24, Area 0

Process ID 1, Router ID 192.168.12.2, Network Type BROADCAST, Cost: 1

Transmit Delay is 1 sec, State DR, Priority 1

Designated Router (ID) 192.168.12.2, Interface address 192.168.12.2

No backup designated router on this network

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

  oob-resync timeout 40

  Hello due in 00:00:01
```

OSPF has been enabled on the interface of both routers so we know that the correct network type was used. However if you look closely at R1 you can see that it says “No Hellos (Passive Interface)”. If you configure passive interface then the network on the interface will still be advertised but it won’t send any OSPF hello packets. This way it’s impossible to form an OSPF neighbor adjacency. Take a look at the configuration of R1:

```
R1#show run | section ospf

router ospf 1

  log-adjacency-changes

  passive-interface FastEthernet0/0

  network 192.168.12.0 0.0.0.255 area 0
```

There’s our problem. Let’s fix it:

```
R1(config)#router ospf 1

R1(config-router)#no passive-interface fastEthernet 0/0
```

Let’s get rid of the passive interface. Now we have a neighbor adjacency:

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

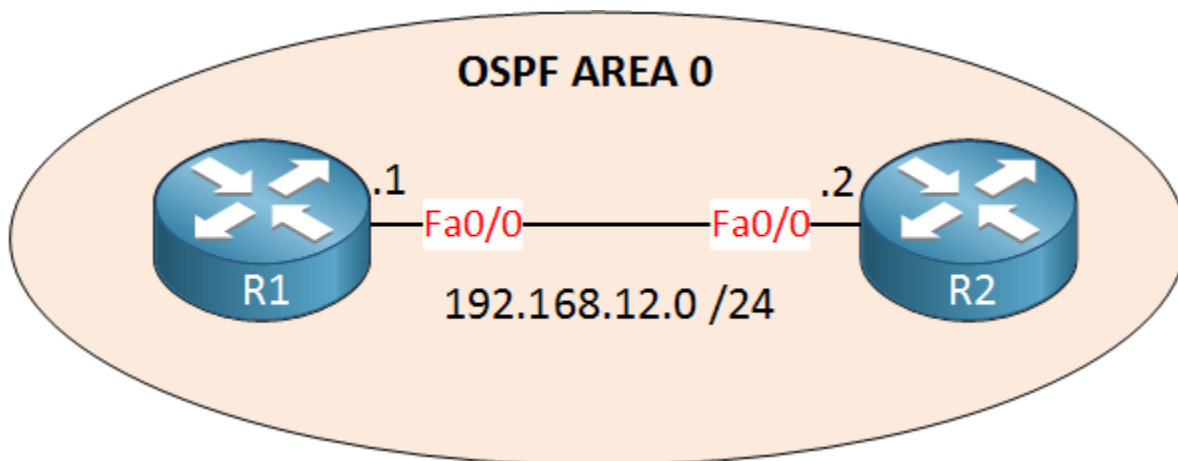
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.12.2	1	FULL/DR	00:00:31	192.168.12.2	FastEthernet0/0

Now we have a working OSPF neighbor adjacency...problem solved!

Lesson learned: Make sure OSPF is sending hello packets on an interface because otherwise you won't be able to become neighbors.

OSPF Multicast Filtering

Next scenario, same two routers but different issue:



Check out the neighbor adjacency:

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

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.12.2	1	INIT/DROTHER	00:00:31	192.168.12.2	FastEthernet0/0

```
R2#show ip ospf neighbor
```

Interesting...R1 is showing our OSPF neighbor to be in the INIT state while R2 is showing nothing. Let's look at the interfaces:

```
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
```

```
R2#show ip ospf interface fastEthernet 0/0
```

```
FastEthernet0/0 is up, line protocol is up
```

```
Internet Address 192.168.12.2/24, Area 0
```

```
Process ID 1, Router ID 192.168.12.2, Network Type BROADCAST, Cost: 1
```

OSPF has been configured properly on both interfaces as we can see in the example above.

Since R1 is showing the INIT state we can draw the conclusion that it's receiving something from R2. R2 isn't showing anything so it's probably not receiving anything from R1. OSPF uses hello packets to establish the OSPF neighbor adjacency and these are sent using the 224.0.0.5 multicast address. Let's see if this address is reachable:

```
R1#ping 224.0.0.5
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 224.0.0.5, timeout is 2 seconds:
```

```
.
```

```
R2#ping 224.0.0.5
```

```
Type escape sequence to abort.
```

Sending 1, 100-byte ICMP Echos to 224.0.0.5, timeout is 2 seconds:

.

It's a good idea to check if we can ping the multicast address that OSPF uses for the hello packets. We can see the R1 and R2 both don't receive a response. What about unicast traffic?

R1#ping 192.168.12.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms

R2#ping 192.168.12.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

Pinging the IP addresses of each other is no problem however. So what could cause an issue with sending and receiving OSPF multicast traffic? How about an access-list?

R1#show ip interface fastEthernet 0/0 | include access list

Outgoing access list is not set

Inbound access list is not set


```
R2#show ip interface fastEthernet 0/0 | include access list
```

```
Outgoing access list is not set
```

```
Inbound access list is BLOCKSTUFF
```

We are onto something. R2 has an inbound access-list called BLOCKSTUFF. Let's take a closer look:

```
R2#show access-lists
```

```
Extended IP access list BLOCKSTUFF
```

```
10 permit tcp any any
```

```
20 permit udp any any
```

```
30 permit icmp any any (45 matches)
```

The access-list is only permitting TCP,UDP and ICMP traffic. OSPF doesn't use TCP or UDP and it's being dropped by this access-list because of the implicit deny any. You don't see it but there's always a deny any at the bottom of an access-list. Let's change the access-list so that OSPF traffic is allowed:

```
R2(config)#ip access-list extended BLOCKSTUFF
```

```
R2(config-ext-nacl)#5 permit ospf any any
```

```
R2#show access-lists
```

```
Extended IP access list BLOCKSTUFF
```

```
5 permit ospf any any (12 matches)
```

```
10 permit tcp any any
```

```
20 permit udp any any
```

```
30 permit icmp any any (45 matches)
```

Above you can see that OSPF traffic is matching the access-list. Did this fix the issue?

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

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.12.2	1	FULL/DR	00:00:37	192.168.12.2	FastEthernet0/0

Problem solved, it's now showing FULL. Let's try to ping that multicast address one more time:

```
R1#ping 224.0.0.5
```

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.0.0.5, timeout is 2 seconds:

Reply to request 0 from 192.168.12.2, 16 ms

```
R2#ping 224.0.0.5
```

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.0.0.5, timeout is 2 seconds:

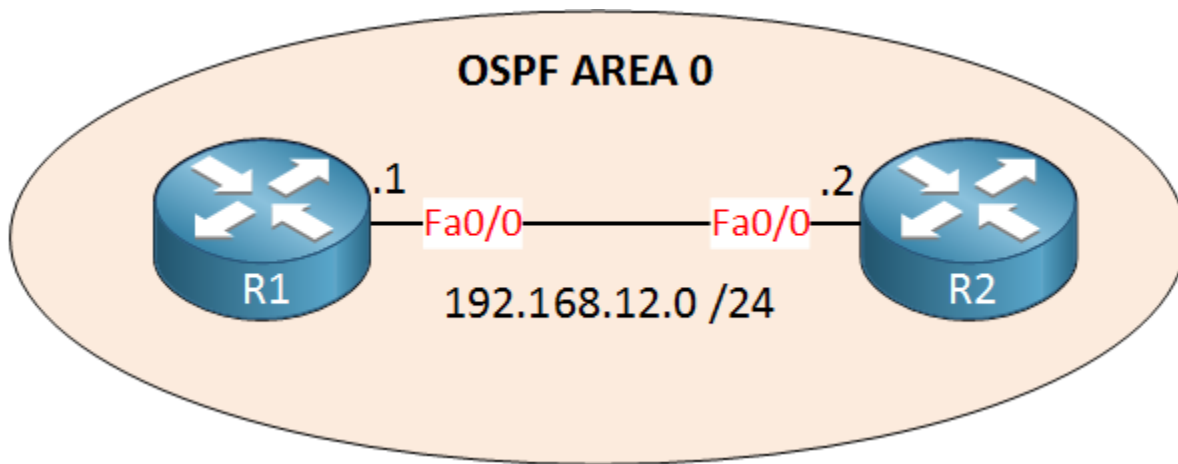
Reply to request 0 from 192.168.12.1, 4 ms

These pings are now working. This is a nice and quick method to check if multicast traffic is blocked or not. Anyway our problem is solved!

Lesson learned: Don't block OSPF multicast addresses 224.0.0.5 and 224.0.0.6.

OSPF Subnet Mask

Once more R1 and R2 are having issues:



Same scenario different issue:

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

```
R2#show ip ospf neighbor
```

There are no OSPF neighbors but we do see that OSPF has been enabled on the interface. Let's see if the multicast addresses are reachable:

```
R1#ping 224.0.0.5
```

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.0.0.5, timeout is 2 seconds:

Reply to request 0 from 192.168.12.2, 16 ms

```
R2#ping 224.0.0.5
```

Type escape sequence to abort.

```
Sending 1, 100-byte ICMP Echos to 224.0.0.5, timeout is 2 seconds:
```

```
Reply to request 0 from 192.168.12.1, 4 ms
```

I can ping the multicast addresses so that's no problem. This might be a good moment to enable a debug to see what is going on:

```
R1#debug ip ospf hello
```

```
OSPF hello events debugging is on
```

This is a very useful debug to see what is going on behind the scenes. Let's reset OSPF:

```
R1#clear ip ospf process
```

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

I'll reset the OSPF process to speed things up. Keep in mind you can also reset just 1 OSPF neighbor which is a better idea if this was a production network. Here's what you will see:

```
R1#
```

```
OSPF: Mismatched hello parameters from 192.168.12.2
```

```
OSPF: Dead R 40 C 40, Hello R 10 C 10 Mask R 255.255.255.128 C 255.255.255.0
```

Now we have something to work with. R1 says it received a hello packet but we have mismatched hello parameters. The R stands for what we received and the C stands for what we have configured.

You can see that there is a mismatch in the subnet mask. R1 is configured with subnet mask 255.255.255.0 while R2 has subnet mask 255.255.255.128. OSPF will only compare the subnet mask when you are using the broadcast network type. You can also spot this error if you look at the OSPF information per interface:

```
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
```

```
R2#show ip ospf interface fastEthernet 0/0
```

```
FastEthernet0/0 is up, line protocol is up
```

```
Internet Address 192.168.12.2/25, Area 0
```

```
Process ID 1, Router ID 192.168.12.2, Network Type BROADCAST, Cost: 1
```

I can use the **show ip ospf interface** command to check the network type, you can see it's broadcast. It also shows the incorrect subnet mask. Of course you could also find this in the running configuration:

```
R1#show run interface fastEthernet 0/0
```

```
Building configuration...
```

```
Current configuration : 97 bytes
```

```
!
```

```
interface FastEthernet0/0
```

```
ip address 192.168.12.1 255.255.255.0
```

```
duplex auto
```

```
speed auto
```

```
R2#show run interface fastEthernet 0/0
```

```
Building configuration...
```

```
Current configuration : 130 bytes
```

```
!  
  
interface FastEthernet0/0  
  
ip address 192.168.12.2 255.255.255.128  
  
duplex auto  
  
speed auto
```

Let's change the subnet mask on R2 so that it matches with R1:

```
R2(config)#interface fastEthernet 0/0  
  
R2(config-if)#ip address 192.168.12.2 255.255.255.0
```

That should be it. Here's what we see on the console:

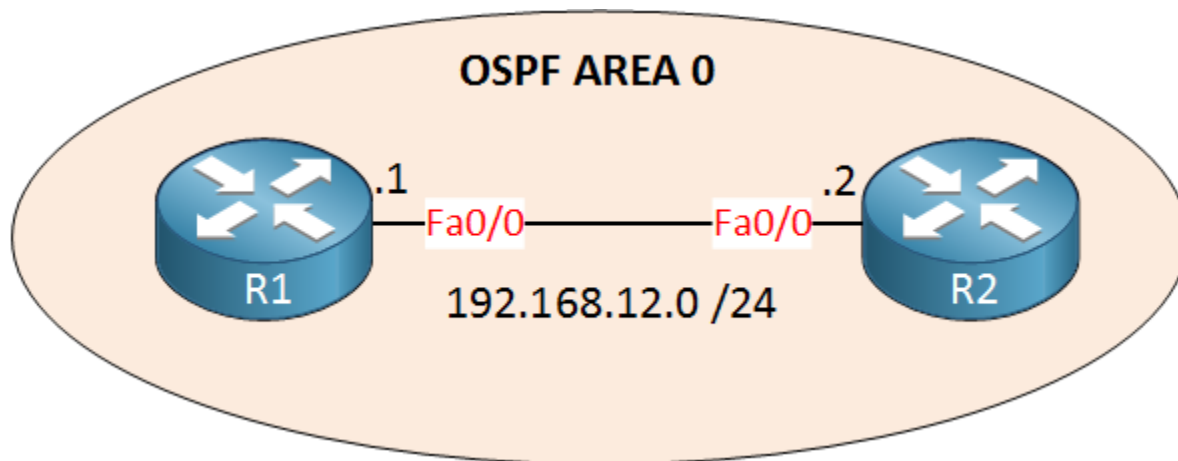
```
R1#  
  
%OSPF-5-ADJCHG: Process 1, Nbr 192.168.12.2 on FastEthernet0/0 from LOADING to FULL,  
Loading Done  
  
R2#  
  
%OSPF-5-ADJCHG: Process 1, Nbr 192.168.12.1 on FastEthernet0/0 from LOADING to FULL,  
Loading Done
```

And as you can see we now have a working OSPF neighbor adjacency.

Lesson learned: Make sure you use the same subnet mask on routers that are directly connected to each other.

OSPF Hello & Dead Interval

The next issue is also related to the hello packet. We'll use the same two routers:



I'm going to jump straight to the debug portion:

```
R1#debug ip ospf hello

OSPF hello events debugging is on

R1#

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

OSPF: Rcv hello from 192.168.12.2 area 0 from FastEthernet0/0 192.168.12.2

OSPF: Mismatched hello parameters from 192.168.12.2

OSPF: Dead R 11 C 24, Hello R 10 C 6 Mask R 255.255.255.0 C 255.255.255.0
```

This issue is similar to the last scenario. There are a number of things that have to match in the hello packet in order to become OSPF neighbors. The dead interval on R1 is configured for 24 seconds and on R2 for 11 seconds. The hello interval is configured for 10 seconds on R2 and 6 seconds on R1. Let's make sure these are the same:

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

R1(config-if)#ip ospf hello-interval 10

R1(config-if)#ip ospf dead-interval 11
```

We need to change this on the interface. A few seconds later the OSPF neighbor adjacency will appear:

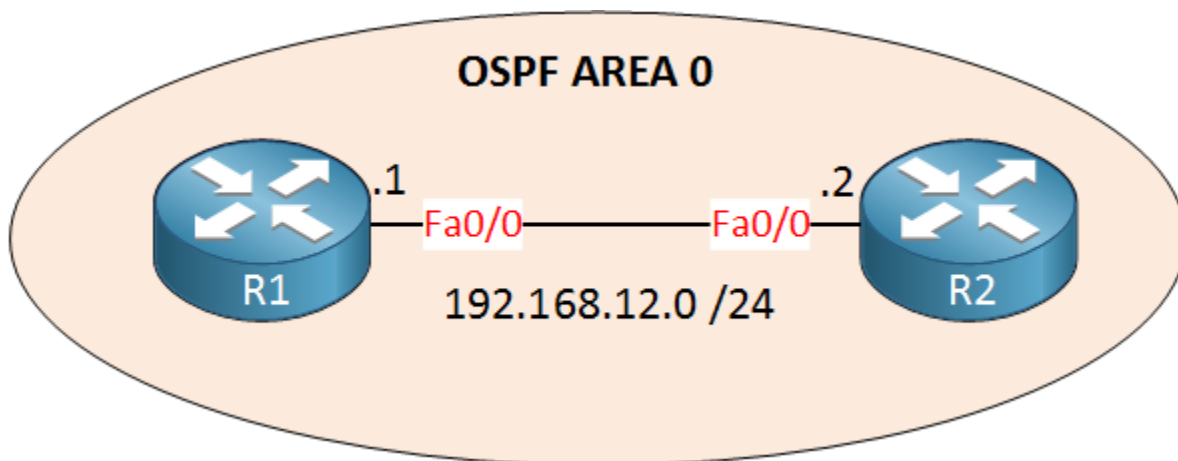
```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.12.2	1	FULL/DR	00:00:04	192.168.12.2	FastEthernet0/0

This fixes our problem; we now have a working OSPF neighbor adjacency.

Lesson learned: Make sure all required parameters in the hello packets match.

OSPF Authentication



Something else that messes up the OSPF neighbor adjacency is authentication.

OSPF offers 3 methods for authentication: