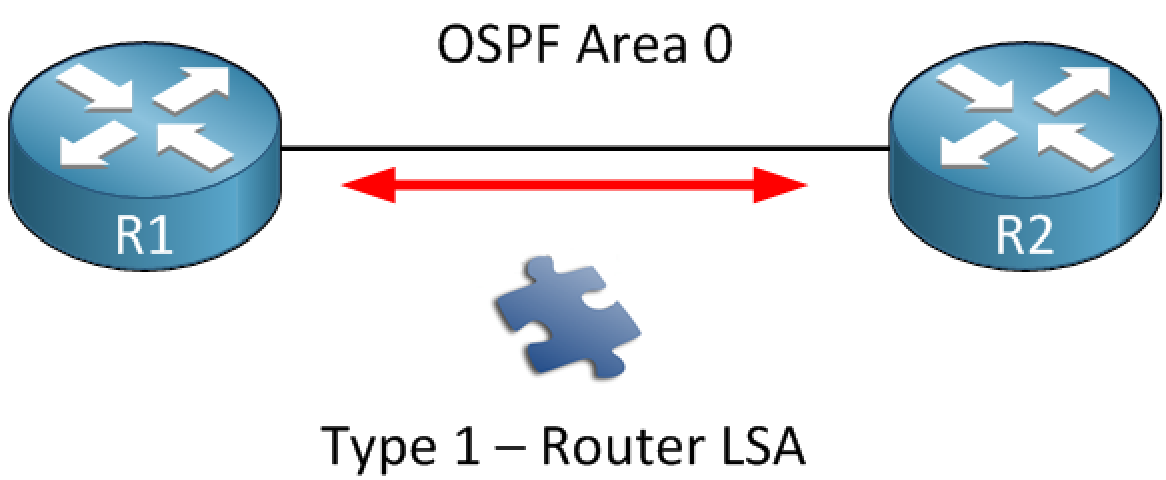
OSPF LSA Types Explained

OSPF uses a LSDB (link state database) and fills this with LSAs (link state advertisement). Instead of using 1 LSA packet OSPF has many different types of LSAs and in this lesson I'm going to show all of them to you. Let's start with an overview:

- LSA Type 1: Router LSA
- LSA Type 2: Network LSA
- LSA Type 3: Summary LSA
- LSA Type 4: Summary ASBR LSA
- LSA Type 5: Autonomous system external LSA
- LSA Type 6: Multicast OSPF LSA
- LSA Type 7: Not-so-stubby area LSA
- LSA Type 8: External attribute LSA for BGP

For many people it helps to visualize things in order to understand and remember. I like to visualize OSPF LSAs as jigsaw puzzle pieces. One jigsaw is nothing but all of them together give us the total picture…for OSPF this is the LSDB.

Here's the first LSA Type:



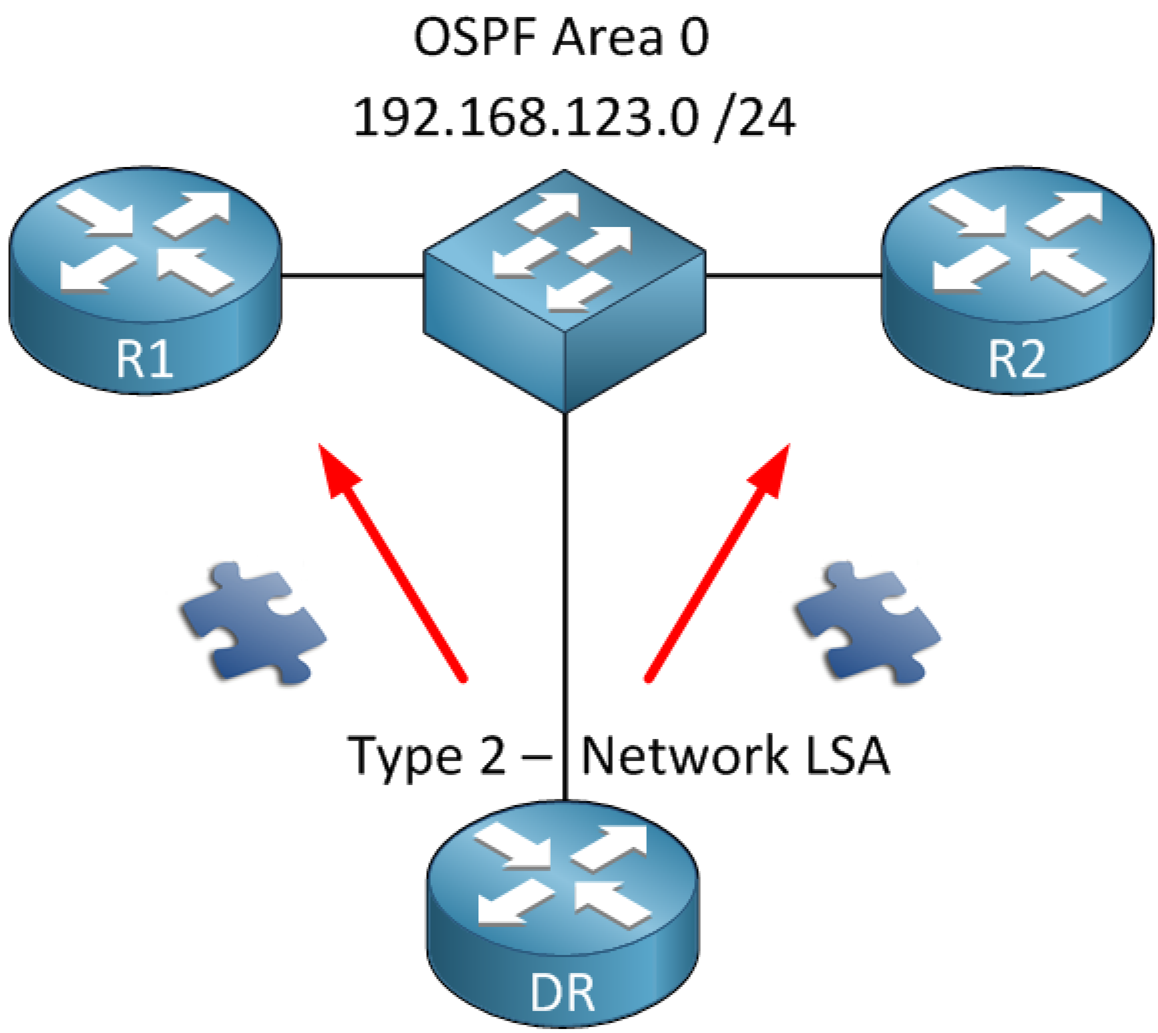
Each router within the area will flood a **type 1 router LSA** within the area. In this LSA you will find a list with all the directly connected links of this router. How do we identify a link?

- The IP prefix on an interface.
- The link type. There are 4 different link types:

Link Type	Description
1	Point-to-point connection to another router.
2	Connection to transit network.
3	Connection to stub network.
4	Virtual Link

Don't worry too much about the link types for now, we will see them later. Keep in mind that the router LSA **always stays within the area**.

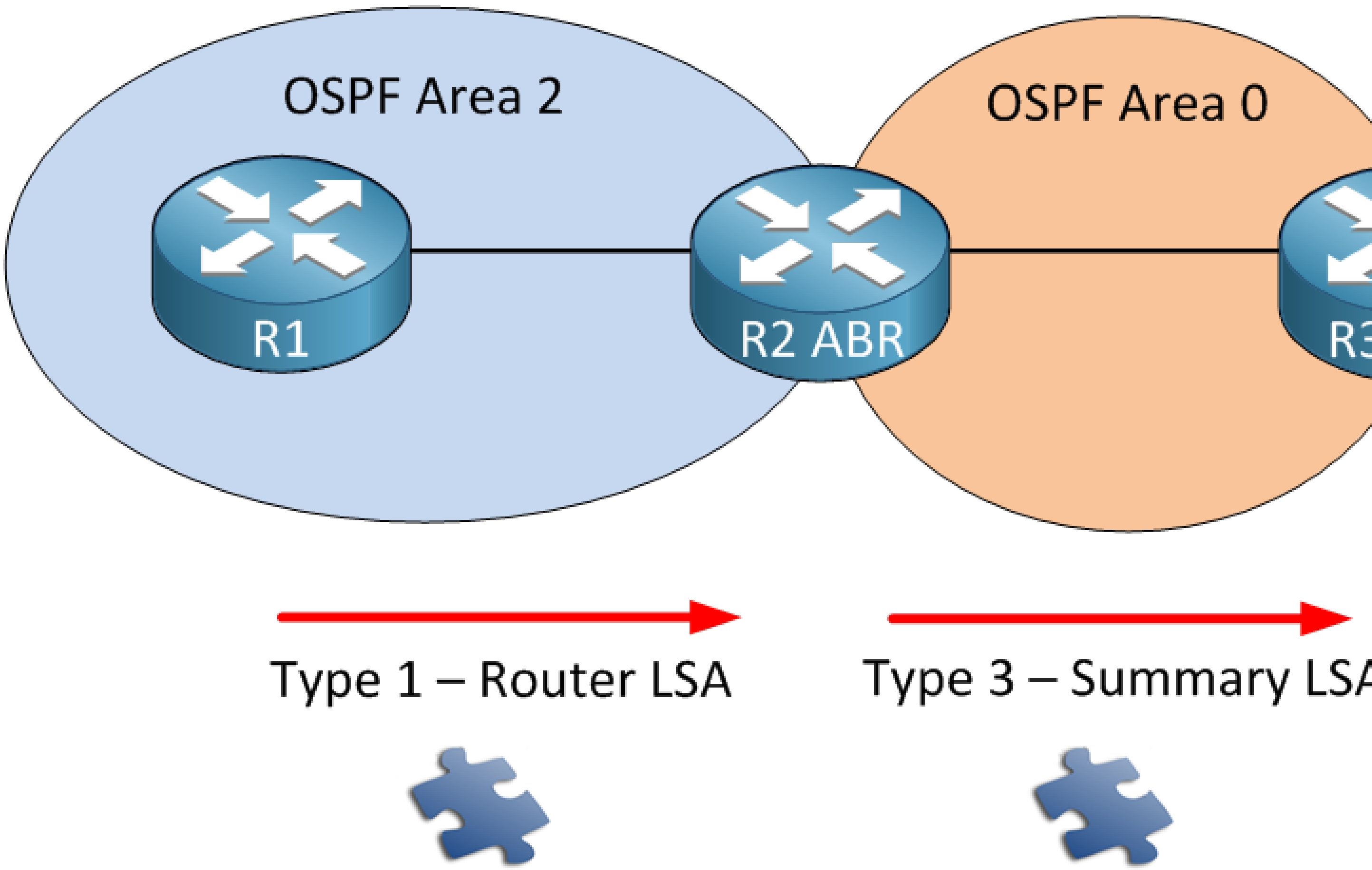
The second LSA type (network LSA) is created for multi-access networks:



The **network LSA** or **type 2** is created for each multi-access network. Remember the OSPF network types? The broadcast and non-broadcast network types require a DR/BDR. If this is the case you will see these network LSAs being generated by the DR. In this LSA we will find all the routers that are connected to the multi-access network, the DR and of course the prefix and subnet mask.

In my example above we will find R1, R2 and the DR in the network LSA. We will also see the prefix 192.168.123.0 /24 in this LSA. Last thing to mention: the network LSA always **stays within the area**.

Let's look at the third LSA type:

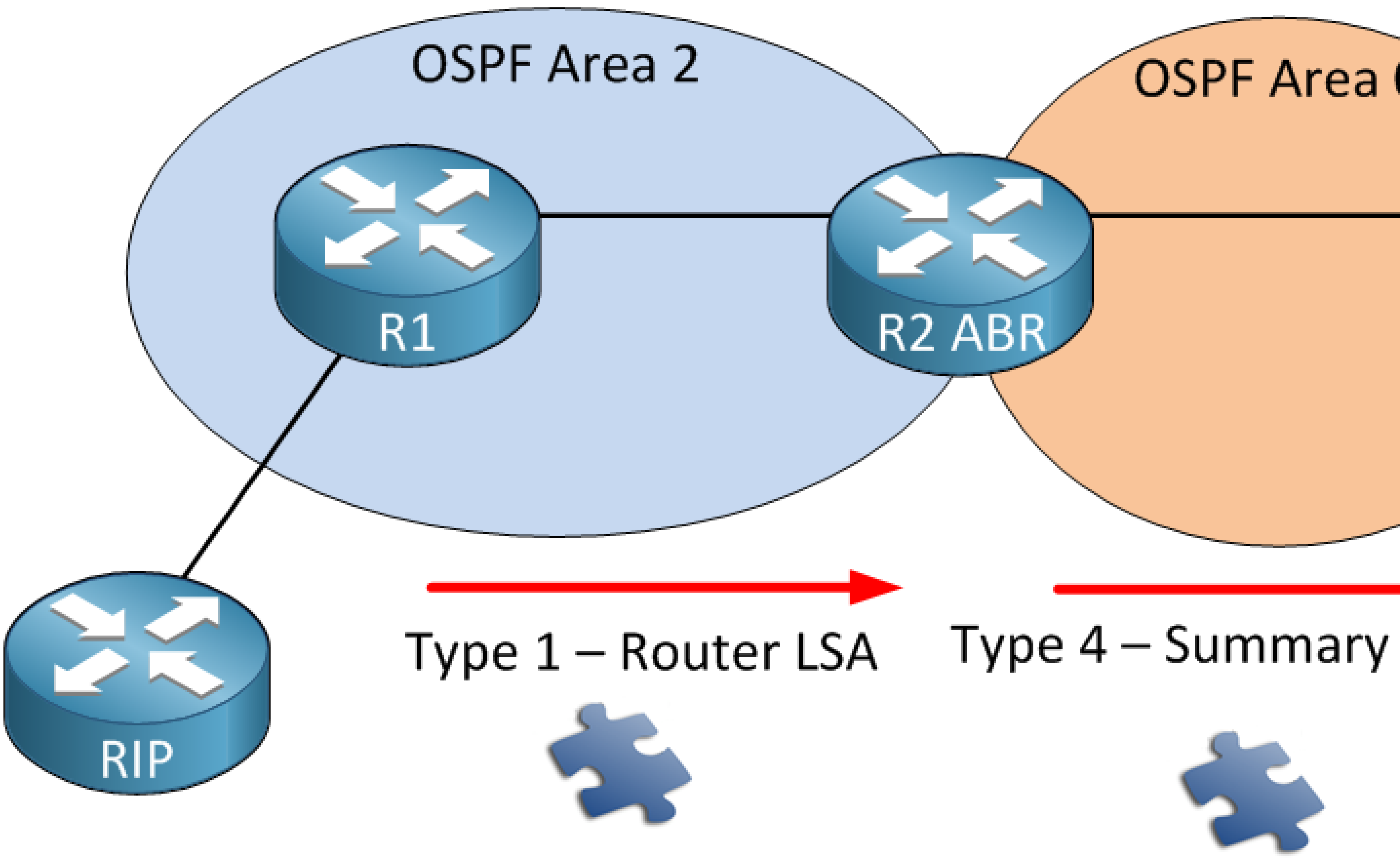


Type 1 router LSAs **always stay within the area**. OSPF however works with multiple areas and you probably want full connectivity within all of the areas. R1 is flooding a router LSA within the area so R2 will store this in its LSDB. R3 and R4 also need to know about the networks in Area 2.

R2 is going to create a **Type 3 summary LSA** and flood it into area 0. This LSA will flood into all the other areas of our OSPF network. This way all the routers in other areas will know about the prefixes from **other areas.**

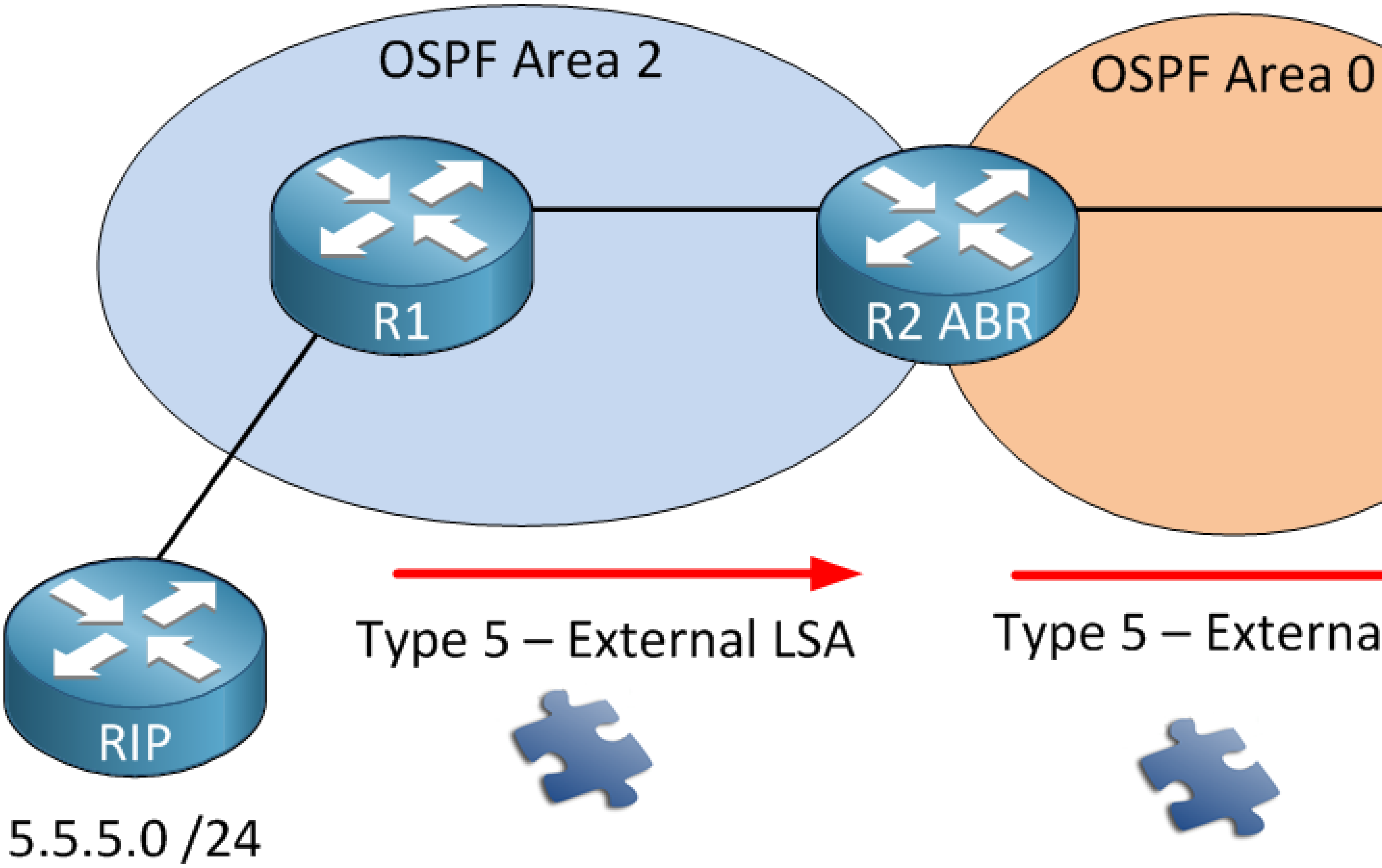
The name "summary" LSA is very misleading. By default OSPF is **not going to summarize** anything for you. There is however a command that let you summarize inter-area routes. Take a look at my OSPF summarization lessonif you are interested. If you are looking at the routing table of an OSPF router and see some **O IA** entries you are looking at LSA type 3 summary LSAs. Those are your inter-area prefixes!

Time for the fourth LSA type:



In this example we have R1 that is redistributing information from the RIP router into OSPF. This makes R1 an **ASBR (Autonomous System Border Router).** What happens is that R1 will flip a bit in the router LSA to identify itself as an ASBR. When R2 who is an ABR receives this router LSA it will create a **type 4 summary ASBR LSA** and flood it into area 0. This LSA will also be flooded in all other areas and is required so all OSPF routers know where to find the ASBR.

What about LSA type 5? Let's check it out:

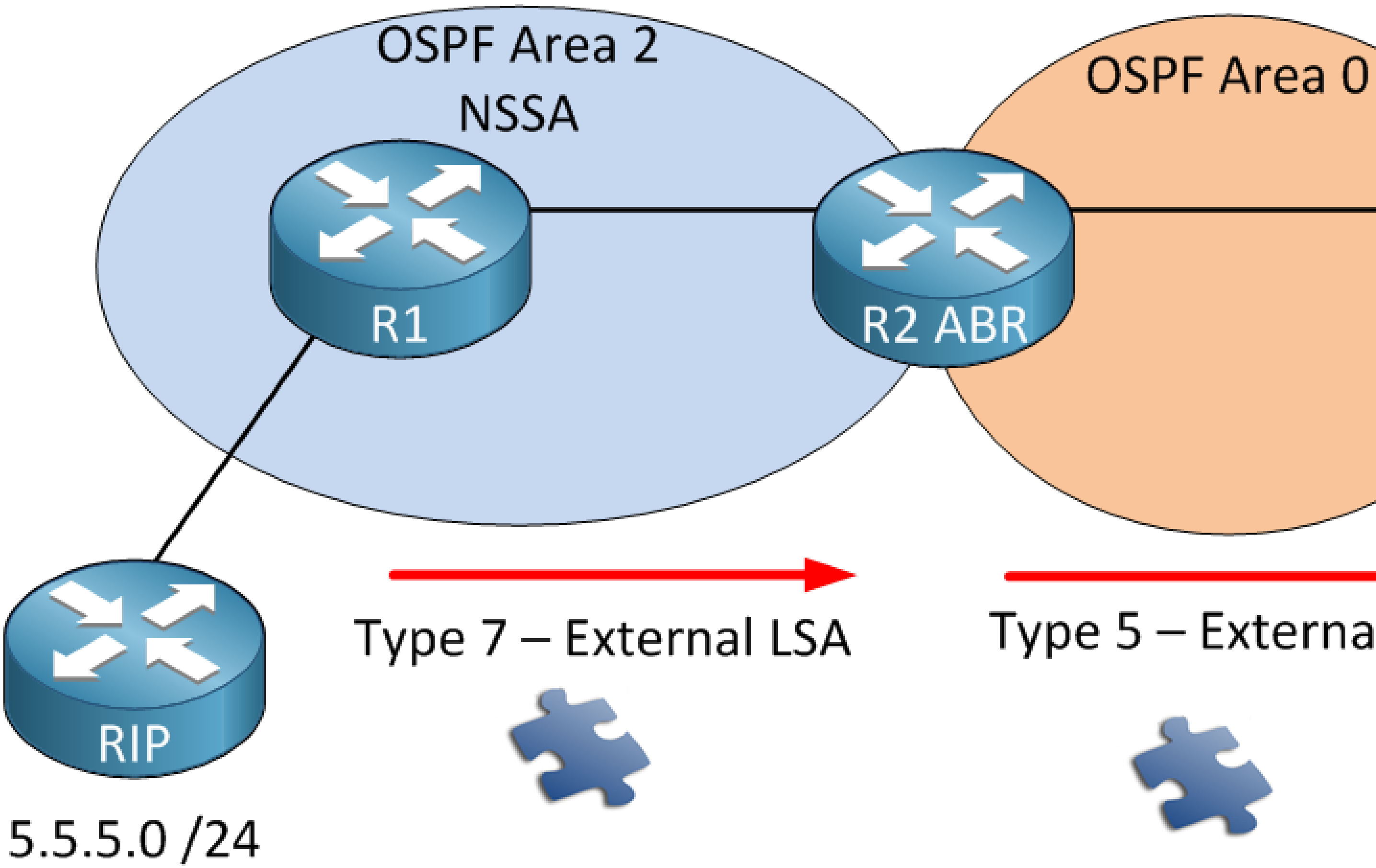


Same topology but I've added a prefix (5.5.5.0 /24) at our RIP router. This prefix will be redistributed into OSPF. R1 (our ASBR) will take care of this and create a **type 5 external LSA** for this. Don't forget we still need type 4 summary ASBR LSA to locate R1. If you ever tried redistribution with OSPF you might have seen **O E1 or E2** entries. Those are the external prefixes and our type 5 LSAs.

What about OSPF LSA type 6? **Type 6 multicast ospf LSA** I can skip because it's not being used. It's not even supported by Cisco. We use PIM (Protocol Independent Multicast) for multicast configurations.

If you are studying the LSA types for CCNA R&S then you don't have to worry about LSA type 7. These are used for a special area type called NSSA that is only covered in the CCNP ROUTE material.

Let's look at the last LSA type, number 7:



Last LSA type…promised! NSSA areas do not allow type 5 external LSAs. In my picture R1 is still our ASBR redistributing information from RIP into OSPF.

Since type 5 is not allowed we have to think of something else. That's why we have a **type 7 external LSA** that carries the exact same information but is not blocked within the NSSA area. R2 will translate this type 7 into a type 5 and flood it into the other areas.

Let me summarize the LSA types for you:

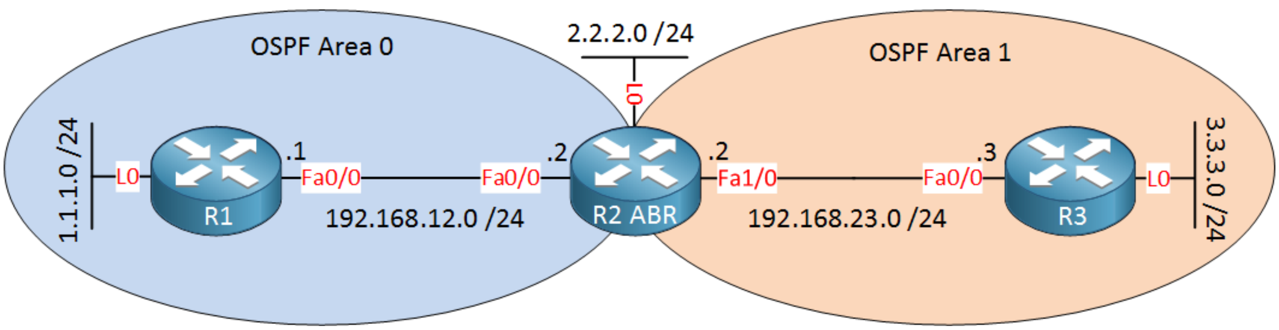
- **Type 1 – Router LSA:** The Router LSA is generated by each router for each area it is located. In the link-state ID you will find the originating router's ID.
- **Type 2 – Network LSA:** Network LSAs are generated by the DR. The link-state ID will be the interface IP address of the DR.
- **Type 3 – Summary LSA:** The summary LSA is created by the ABR and flooded into other areas.
- **Type 4 – Summary ASBR LSA:** Other routers need to know where to find the ASBR. This is why the ABR will generate a summary ASBR LSA which will include the router ID of the ASBR in the link-state ID field.
- **Type 5 – External LSA:** also known as autonomous system external LSA: The external LSAs are generated by the ASBR.
- **Type 6 – Multicast LSA:** Not supported and not used.

- **Type 7 – External LSA:** also known as not-so-stubby-area (NSSA) LSA: As you can see area 2 is a NSSA (not-so-stubby-area) which doesn't allow external LSAs (type 5). To overcome this issue we are generating type 7 LSAs instead.

The only thing left to do is take a look at these LSAs in action…time to configure some routers!

Verification

We can see all the OSPF types in the LSDB, to demonstrate this I will use the following topology:



It's a simple setup with 3 routers and 2 areas. I've added a couple of loopbacks so we have prefixes to look at. Here's the configuration:

```
R1(config)#router ospf 1

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

R1(config-router)#network 1.1.1.0 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

R2(config-router)#network 192.168.23.0 0.0.0.255 area 1

R3(config)#router ospf 1

R3(config-router)#network 192.168.23.0 0.0.0.255 area 1

R3(config-router)#network 3.3.3.0 0.0.0.255 area 1
```

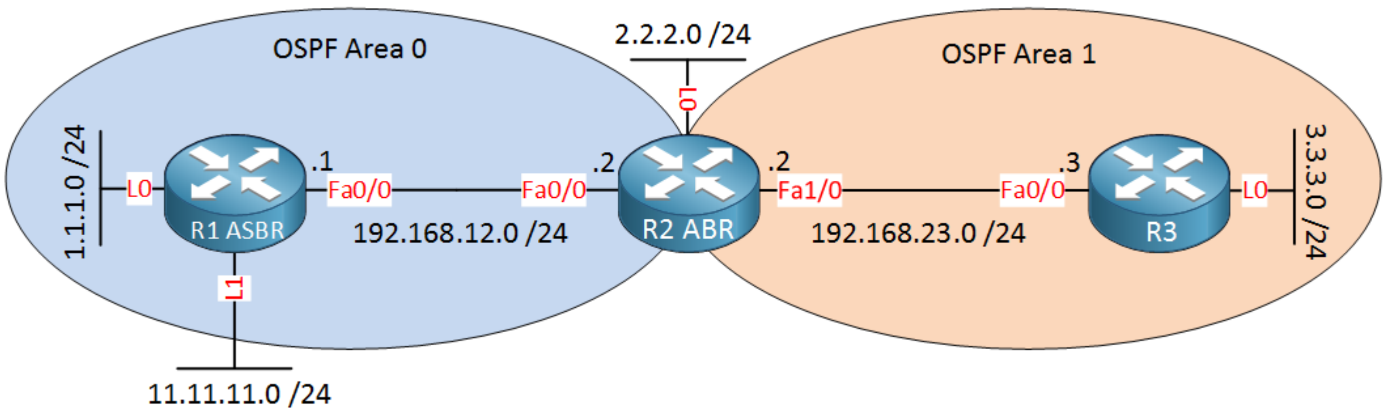
Let's start by looking at the LSDB of R1:

```
R1#show ip ospf database

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count

1.1.1.1         1.1.1.1         30          0x80000003 0x004CD9 2

2.2.2.2         2.2.2.2         31          0x80000002 0x0048E9 1

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum

192.168.12.2    2.2.2.2         31          0x80000001 0x008F1F

                Summary Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum

3.3.3.3         2.2.2.2         17          0x80000001 0x00D650

192.168.23.0    2.2.2.2         66          0x80000001 0x00A70C
```

By using the show ip ospf database we can look at the LSDB and we can see the type 1 router LSAs, type 2 network LSAs and the type 3 summary LSAs here. What else do we find here?

- Link ID: This is what identifies each LSA.
- ADV router: the router that is advertising this LSA.
- Age: The maximum age counter in seconds. The maximum is 3600 seconds or 1 hour.
- Seq#: Here you see the sequence number which starts at 0x80000001 and will increase by 1 for each update.
- Checksum: There is a checksum for each LSA.
- Link count: This will show the total number of directly connected links and is only used for the router LSA.

So that's LSA type 1,2 and 3. To show you number 4 and 5 I have to make some changes:



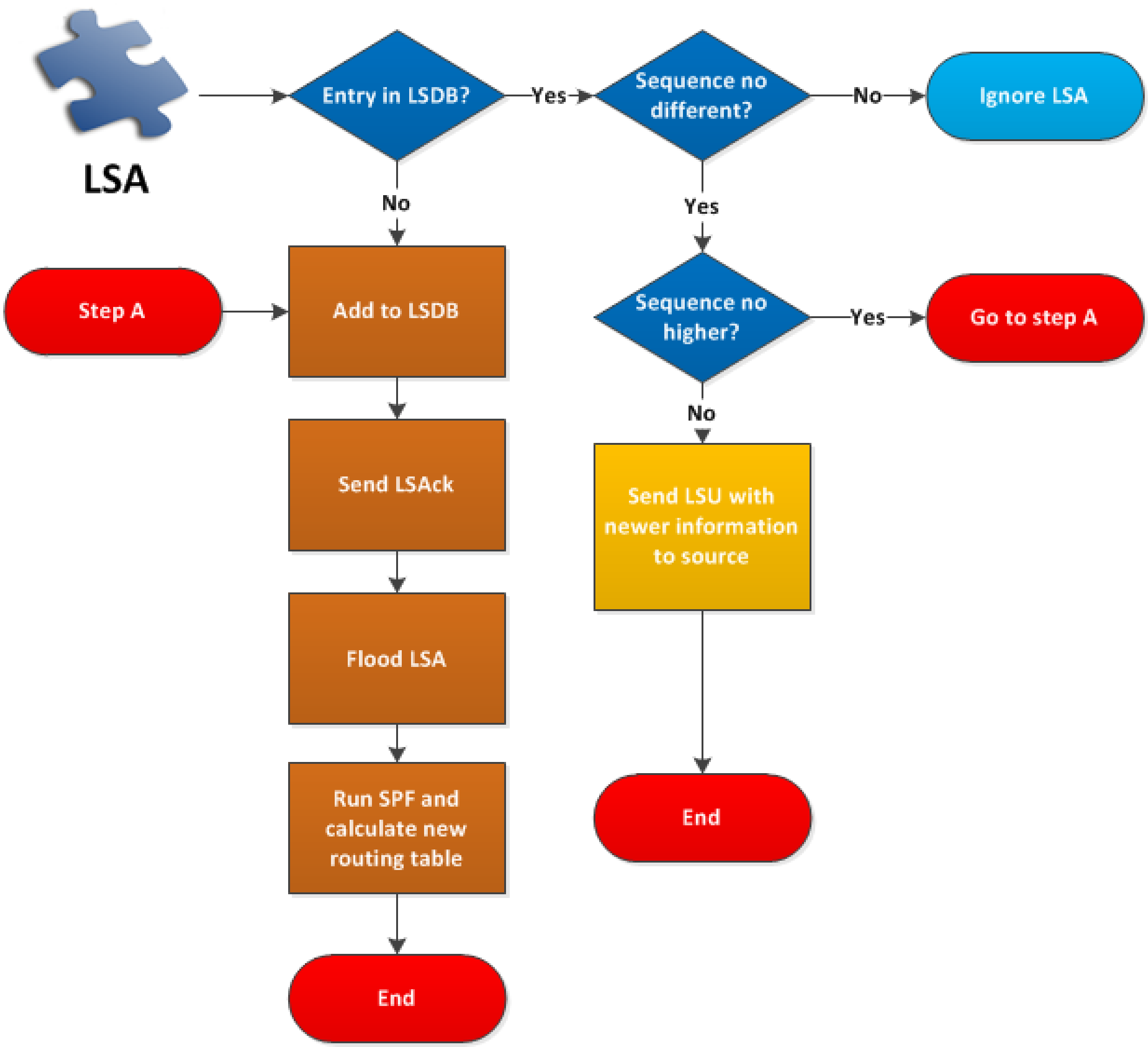
To accomplish this I will redistribute something on R1 into OSPF.

```
R1(config)#interface loopback 1

R1(config-if)#ip address 11.11.11.11 255.255.255.0

R1(config-if)#exit

R1(config)#router ospf 1

R1(config-router)#redistribute connected subnets
```

I created an additional loopback interface and configured an IP address. Then I'm telling OSPF to redistribute the directly connected interfaces into OSPF. Let's look at the LSDB of R2 and R3:

OSPF LSAs and LSDB flooding Tutorial

How exactly does OSPF fill the LSDB? Let's zoom in on the operation of how OSPF keeps its link-state database up-to-date:



Each LSA has an **aging timer** which carries the **link-state age field.** By default each OSPF LSA is only valid for **30 minutes.** If the LSA expires then the router that created the LSA will resend the LSA and increase the **sequence number**.

Let's walk through this flowchart together. In this example a new LSA is arriving at the router and OSPF has to decide what to do with it:

1. If the LSA isn't already in the LSDB it will be added and a LSAck (acknowledgement) will be sent to the OSPF neighbor. The LSA will be flooded to all other OSPF neighbors and we have to run SPF to update our routing table.
2. If the LSA is already in the LSDB and the sequence number is the same then we will ignore the LSA.
3. If the LSA is already in the LSDB and the sequence number is different then we have to take action:
    1. If the sequence number is higher it means this information is newer and we have to add it to our LSDB.
    2. If the sequence number is lower it means our OSPF neighbor has an old LSA and we should help them. We will send a **LSU** (**Link state update)** including the newer LSA to our OSPF neighbor. The LSU is an envelope that can carry multiple LSAs in it.

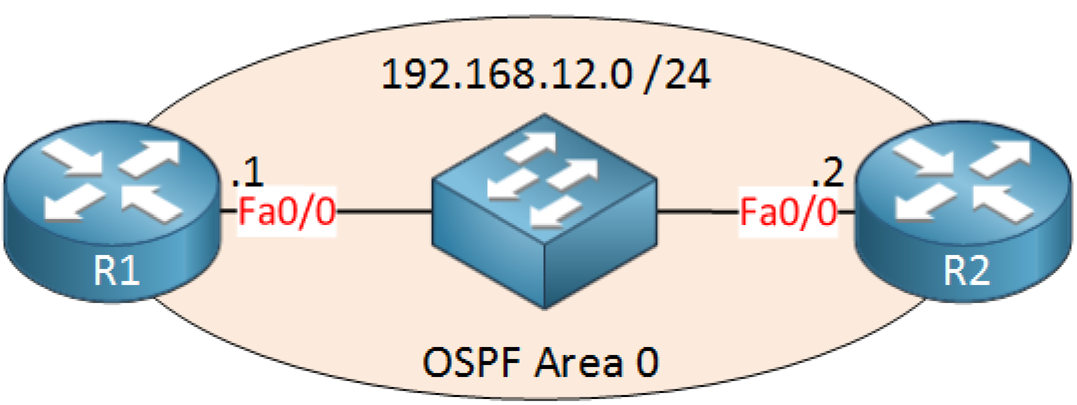
OSPF Hello and Dead Interval

OSPF uses hello packets and two timers to check if a neighbor is still alive or not:

- Hello interval: this defines how often we send the hello packet.
- Dead interval: this defines how long we should wait for hello packets before we declare the neighbor dead.

The hello and dead interval values can be different depending on the OSPF network type. On Ethernet interfaces you will see a 10 second hello interval and a 40 second dead interval.

Let's take a look at an example so we can see this in action. Here's the topology I will use:



We'll use two routers with a switch in between.

Configuration

Let's enable OSPF:

```
R1 & R2#

(config)#router ospf 1

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

Let's take a look at the default hello and dead interval:

```
R1#show ip ospf interface FastEthernet 0/0 | include intervals

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

The hello and dead interval can be different for each interface. Above you can see that the hello interval is 10 seconds and the dead interval is 40 seconds. Let's try if this is true:

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

R1(config-if)#shutdown
```

After shutting the interface on R1 you will see the following message:

```
R1#

Aug 30 17:57:05.519: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.12.2 on FastEthernet0/0
from FULL to DOWN, Neighbor Down: Interface down or detached
```

R1 will know that R2 is unreachable since its interface went down. Now take a look at R2:

```
R2#

Aug 30 17:57:40.863: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.12.1 on FastEthernet0/0
from FULL to DOWN, Neighbor Down: Dead timer expired
```

R2 is telling us that the dead timer has expired. This took a bit longer. The interface on R1 went down at 17:57:05 and R2's dead timer expired at 17:57:40…that's close to 40 seconds.

Let's activate the interface again:

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

R1(config-if)#no shutdown
```

40 seconds is a long time…R2 will keep sending traffic to R1 while the dead interval is expiring. To speed up this process we can play with the timers. Here's an example:

```
R1 & R2

(config)#interface FastEthernet 0/0

(config-if)#ip ospf hello-interval 1

(config-if)#ip ospf dead-interval 3
```

You can use these two commands to change the hello and dead interval. We'll send a hello packet every second and the dead interval is 3 seconds. Let's verify this:

```
R1#show ip ospf interface FastEthernet 0/0 | include intervals

  Timer intervals configured, Hello 1, Dead 3, Wait 3, Retransmit 5
```

Reducing the dead interval from 40 to 3 seconds is a big improvement but we can do even better: