

OSPF Neighbor States Explained With Example

OSPF routers go through the seven states, called Down, Attempt/Init, Two ways, Exstart, Exchange, Loading and full while building adjacency with other OSPF speaking routers. In this we will see these states in easy language with examples. Along with these states, also explanation of few other terminologies used in this process.

OSPF Neighborship states

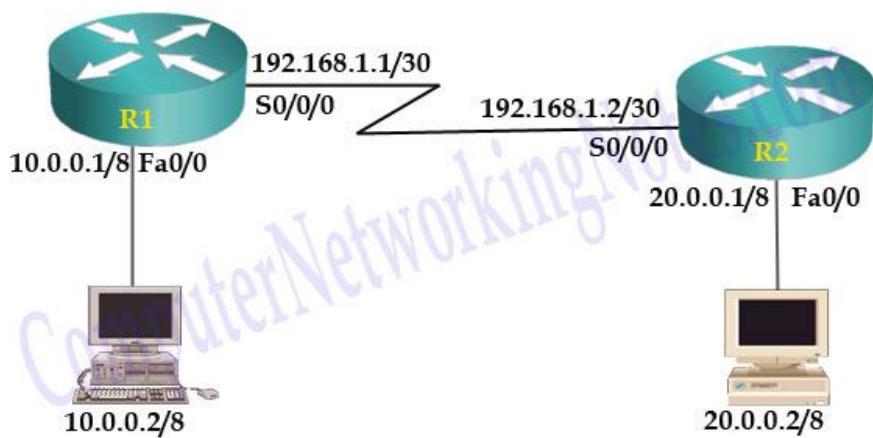
OSPF routers go through the seven states while building neighborship with other routers.

- Down state
- Attempt/Init state
- Two ways state
- Exstart state
- Exchange state
- Loading state
- Full state

Let's understand these states with a simple example. Assume that our network has two routers running OSPF routing protocol. Routers are connected with each other via serial link. We just turned on both routers simultaneously.

Down state

At this point both routers have no information about each other. R1 does not know which protocol is running on R2. Vice versa R2 have no clue about R1. In this stage OSPF learns about the local interfaces which are configured to run the OSPF instance.



In down state routers prepares themselves for neighborship process. In this state routers choose RID (Router ID). RID plays a big role in OSPF process. Before we move in next state let's understand what is RID.

RID

RID is a unique identifier of Router in OSPF network. It must be unique within the autonomous system. Routers identify each other through the RID in AS.

How do routers choose RID?

An OSPF router looks in three places for RID:-

1. Manual configuration
2. Loopback interface IP configuration
3. Active interfaces IP configuration

Manual configuration

Because RID plays a significant role in network, OSPF allows us to configure it manually. RID is 32 bit long. IP address is also 32 bit in length. We can use IP address as a RID. This gives us more flexibility over RID. For example we can use a simple and sequential IP scheme such as 1.1.1.1 for R1, 1.1.1.2 for R2, 1.1.1.3 for R3, 1.1.1.4 for R4, 1.1.1.5 for R5 and so on.

We can assign RID from OSPF sub command mode.

```
Router(config)#router ospf 1  
Router(config-router)#router-id ip_address
```

If we have assigned RID manually, OSPF will not look in next two options. Suppose we did not assign it through the command. In this situation OSPF will look in next option to find the RID.

Loopback interface IP configuration

If loopback interface is configured, OSPF will choose its IP address as RID. If multiple loopback interfaces are configured, highest IP address will be chosen from all loopback interfaces configuration.

If loopback interface is not configured, OSPF will look in next and last possible place to choose the RID.

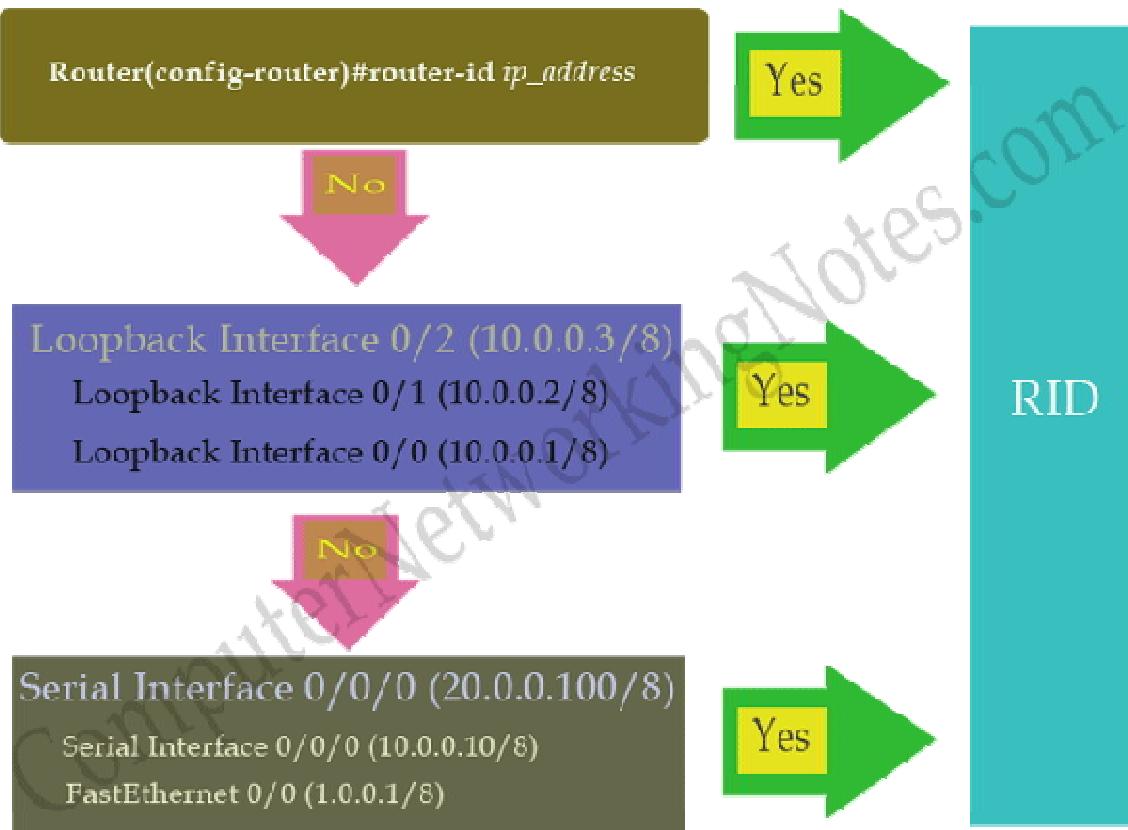
Active interface IP configuration

OSPF will choose the highest IP address from all operational IP interfaces. We should not let the OSPF to use this option. This option does not provide a fix RID which is very necessary for network stability.

This option has several reasons which may force OSPF to recalculate the RID such as Interface which IP address is chosen may go down or for troubleshooting we may enable / disable the interfaces.

Key points

- OSPF will follow the sequence (Manual configuration => Loopback interface => Active interface) of options while selecting RID. If RID is found, it will not look in next option.
- OSPF will choose IP address only from operational IP interface. Operational means interface should be listed as line is up and line protocol is up in the output of show ip interface brief command.
- When multiple IP addresses are available, OSPF will always pick highest IP address for RID.
- For network stability we should always set RID from either **router-id** command or by using loopback interfaces.
- By default Router chooses OSPF RID when it initialized. Once RID is selected it will use that RID until next reboot.
- OSPF will not consider any change in RID which we make after initialization. We have two options to implement new RID. Either reboots the router or clear the OPSF process with clear ip ospf process command.
- If OSPF fails to select the RID, it will halt the OSPF process. We cannot use OSPF process without RID.



In down state router do following

- Choose RID and initialize the OSPF process
- Run OSPF instance on local interfaces which are configured through the network command such as **R1(config-router)#network 10.0.0.0 0.0.255.255 area 0.**
- Collect necessary information for Hello packet such RID and configuration values which are required to build the neighborship.

Attempt/Init state

Neighborhood building process starts from this state. R1 multicasts first hello packet so other routers in network can learn about the existence of R1 as an OSPF router. This hello packet contains Router ID and some essential configuration values such as area ID, hello interval, hold down timer, stub flag and MTU. Essential configuration values must be same on routers who want to build an OSPF neighborship.

In previous part of this article I explained essential configuration values in detail with example. For this tutorial I assume that these values match on both routers. If essential configuration values match, R2 will add R1 in his neighbor Table.



In Init state routers do following

- R1 will generate a hello packet with RID and essential configuration values and send it out from all active interfaces.
- The hello packets are sent to the **multicast address 224.0.0.5**.
- R2 will receive this packet.
- R2 will read RID from packet and look in neighbor table for existing entry.
- If match found, R2 would skip neighborship building process and reset the dead interval timer for that entry.
- If OSPF does not find a match in neighbor table, it will consider R1 (sender router) as a possible OSPF neighbor and start neighborship building process.
- R2 will match its essential configuration values with values listed in packet.
- If all necessary configuration values match, R2 will add R1 in its neighbor table.

At this moment R1 has no idea about R2. R1 will learn about R2 when it will respond.

Before we enter in third state, let's have a quick look on attempt state.

Attempt

In Non-broadcast multi-access environment such as Frame Relay and X.25, OSPF uses Attempt state instead of Init state. OSPF uses this state only if neighbors are statically configured with **neighbor** command. In this situation, it does not have to discover them dynamically. As it already knows the neighbors, it will use unicast instead of multicast in this state.

Once neighborship is built, OSPF uses hello packets as keep alive. If a router does not receive a hello packet from any particular neighbor in dead interval, it will change its state to down from full. After changing the state it will make an effort to contact the neighbor by sending Hello packets. This effort is made in Attempt state.

Basically Both Init and Attempt states describe similar situation where one router has sent a hello packet and waiting for response.

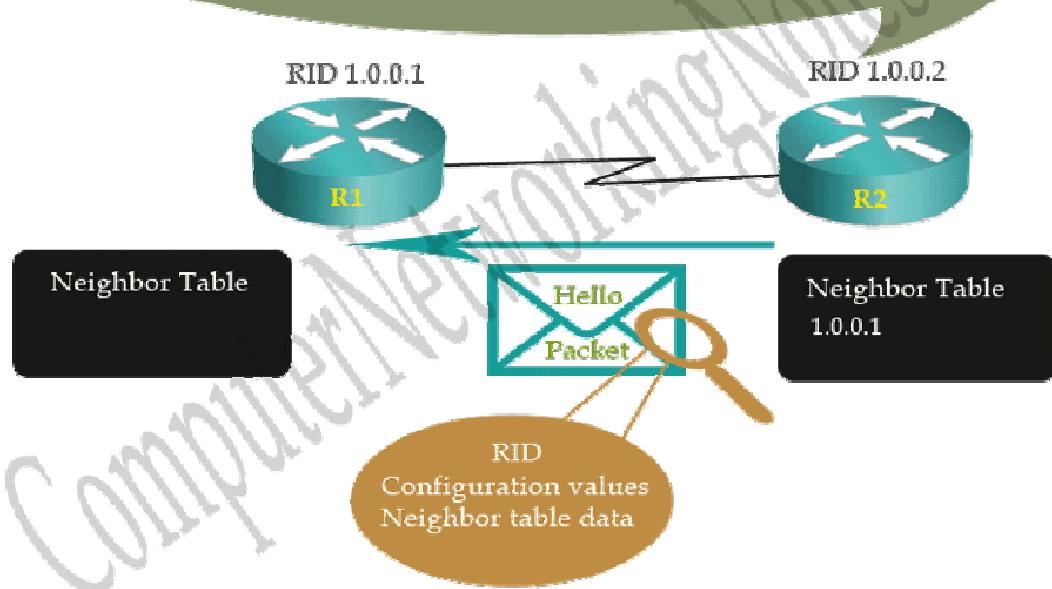
Two ways state

If essential configuration values match, R2 will add R1 in neighbor table and reply with its hello packet. As R2 knows the exact address of R1, it will use unicast for reply. Beside RID and configuration values, this packet also contains the R2's neighbor table data. As we know R2 has already added R1 in its neighbor table. So when R1 will see R2's neighbor table data, R1 would also see its name in this data. This will assure R1 that R2 has accepted its neighborship request.

At this point:-

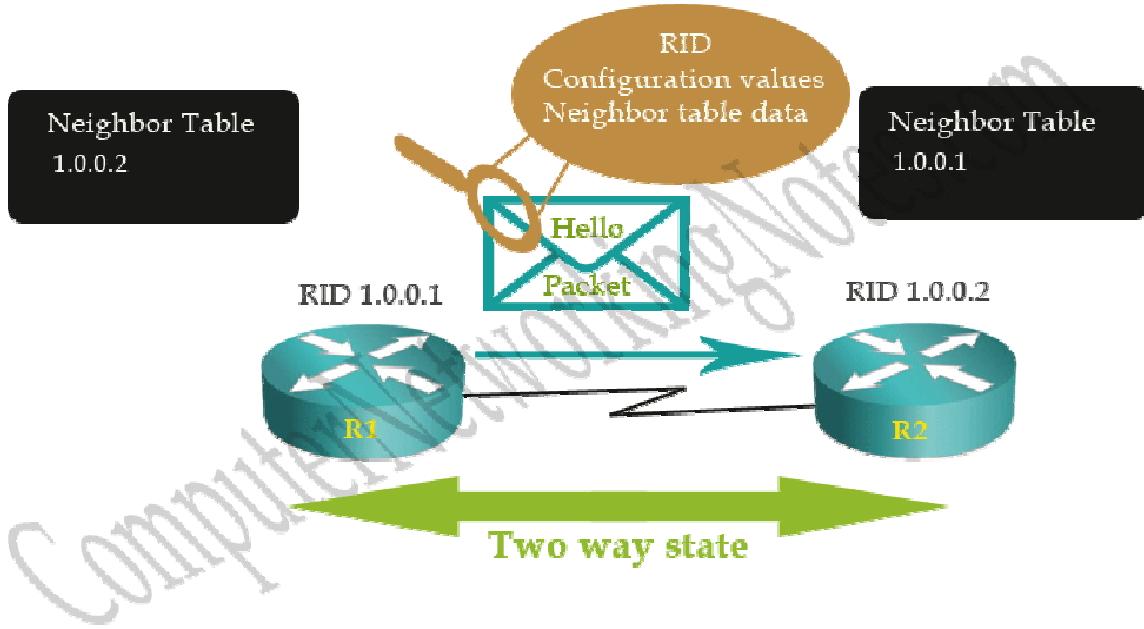
- R2 has checked all essential configuration values listed in hello packet which it received from R1.
- R2 is ready to build neighborship with these parameters.
- R2 has added R1 in its neighbor table.
- To continue the neighborship process, R2 has replied with its hello packet.
- R1 has received a reply from neighbor, with its own RID listed in R2's neighbor table.

Hey buddy, I got your neighborship request.
 Your neighborship parameters meet with my neighborship requirement.
 You are my neighbor now. Am I eligible to be your neighbor?
 Please check my neighborship parameters and
 let me know can we build neighborship?



Now it is R1's turn to take action on R2's reply. This reply would be based on hello packet which it received from R2. As we know that this hello packet contains one additional field; Neighbor table data field which indicates that this is not a regular neighbor discovery hello packet. This packet is a reply of its own request.

- R1 will take following actions:-
- It will read RID from hello packet and look in its neighbor table for existing entry.
- If a match for RID found in neighbor table, it would reset the dead interval timer for that entry.
- If a match is not found in neighbor table, it would read the essential configuration values from packet.
- It will match configuration values with its own values. If values match, it will add R2's RID in neighbor table.
- If packet contains neighbor table data with its own RID, it will consider that as request to enter in two way state.
- R1 will reply with a hello packet which contains its neighbor table data.
- This packet is a confirmation of two ways state.



Fine, our routers are neighbor now. They are ready to exchange the routing information.

Before we understand how routers will exchange routing information, we need to understand the types of network. OSPF uses different types of exchange process for different types of network.

Point to point network

It is a Cisco specific network type. It connects a single pair of routers. HDLC and PPP are example of point to point network type. In this type of network:-

- All routers form full adjacencies with each other.
- Hello packets are sent using a multicast address 224.0.0.5
- No DR and BDR are required.
- All routers are considered as **AllSPFRouters**.

I will explain the terms adjacencies, DR, BDR and AllSPFRouters shortly.

Broadcast Networks

Broadcast networks are capable in connecting more than two devices. Ethernet and FDDI are the example of broadcast type network. In this type of network:-

- A single transmitted packet can be received by all attached devices.
- DR and BDR are required.
- All routers form full adjacencies only with DR and BDR.
- Routers use a multicast address 224.0.0.6 to update the DR.
- DR uses a multicast address 224.0.0.5 to update the all routers.

NMBA

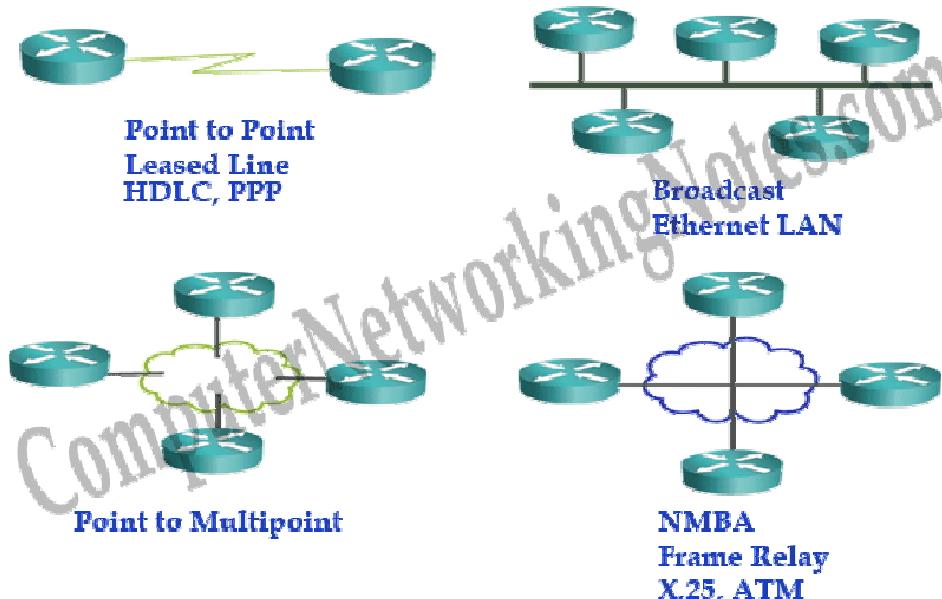
Non-broadcast Multi-access networks are also capable in connecting more than two devices. But they do not have broadcast capability. X.25 and Frame Relay are the example of NMBA type network. In this type of network:-

- As network does not have broadcast capability, dynamic network discovery will not be possible.
- OSPF neighbors must have to define statically.
- All OSPF packets are unicast.
- DR and BDR are required.

Point to multipoint

Point to multipoint is a special implementation of NMBA network where networks are configured as a collection of point to point links. In this type of network:-

- Network must be configured statically.
- No DR and BDR are selected in this type of network.
- OSPF packets are multicast.



We can divide these networks in two types:

1. Networks which need DR and BDR such as broadcast and NBMA
2. Networks which do not need DR and BDR such as point to point and point to multipoint

So what does DR and BDR actually do? Why do we need them in our network?

DR and BDR

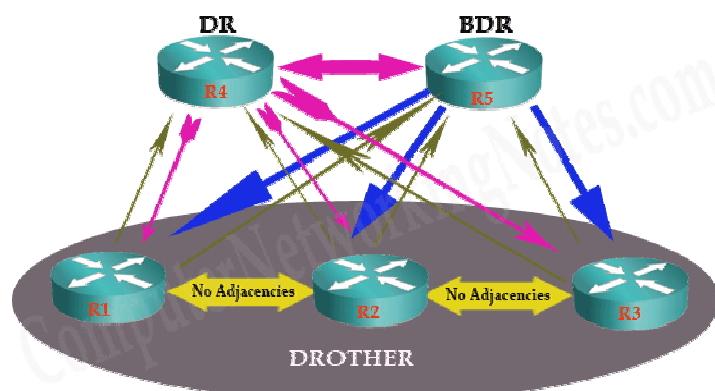
OSPF routers in a network which need DR (Designated router) and BDR (Backup designated router) do not share routing information directly with all each other's. To minimize the routing information exchange, they select one router as designated router (DR) and one other router as backup designated router (BDR). Remaining routers are known as DROTHERs.

All DROTHERs share routing information with DR. DR will share this information back to all DROTHERs. BDR is a backup router. In case DR is down, BDR will immediately take place the DR and would elect new BDR for itself.

Main reason behind this mechanism is that routers have a central point for routing information exchange. Thus they need not to update each other's. A DROTHER only need to update the central point (DR) and other DROTHERs will receive this update from DR.

Practically this will cut the numbers of routing information exchange from $O(n*n)$ to $O(n)$ where n is the number of routers in a multi-access segment.

For example following figure illustrates a simple OSPF network. In this network R4 is selected as DR and R5 is selected as BDR. DROTHERs (R1, R2 and R3) will share routing information with R4 (DR) and R5 (BDR), but they will not share routing information with each other. Later DR will share this information back to all DROTHERs.



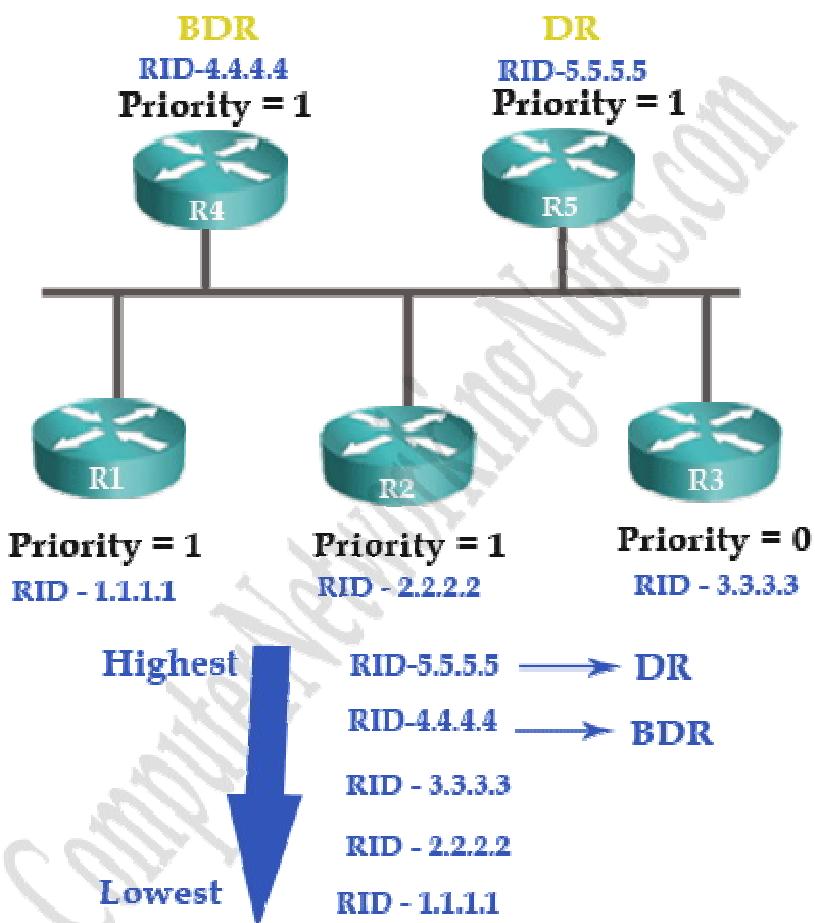
DR and BDR Election process

OSPF uses priority value to select DR and BDR. OSPF router with the highest priority becomes DR. Router with second highest priority becomes BDR. If there is a tie, router with the highest RID will be chosen.

Priority value is 8 bit in length. Default priority value is 1. We can set any value from range 0 to 255. We can change it from Interface Sub-configuration mode with *ip ospf priority* command.

We can force any router to become DR (Highest) or BDR (Second highest) by changing its priority value. If we set priority value to 0, it will never become DR or BDR.

For example following figure illustrates a simple OSPF network. In this network we have five routers. We do not want that R3 becomes DR or BDR. So we changed its default priority value to 0. Now let's see how these routers select DR and BDR.



Condition 1:- Use the highest priority value

This condition says "Arrange all routes in high to low order and pick the highest for DR and second highest for BDR". If we arrange our routers in high to lower order, R3 will stand at last. Remaining routers have equal priority value. So at the end of this condition we have a tie between four routers.

Condition 2:- If there is a tie use the highest RID

This condition says "If there is a tie, use RID value to choose". In our network we have a tie between four routers, so our routers will use RID to elect the DR and BDR. Arranging routers in high to low order will give us the DR and BDR.

As we know that there are two types of network; networks which do not require DR and BDR for exchange process and networks which require DR and BDR for exchange process.

In first type all routers will exchange routing information with each other's. In second type DROTHERs will exchange routing information with DR and BDR.

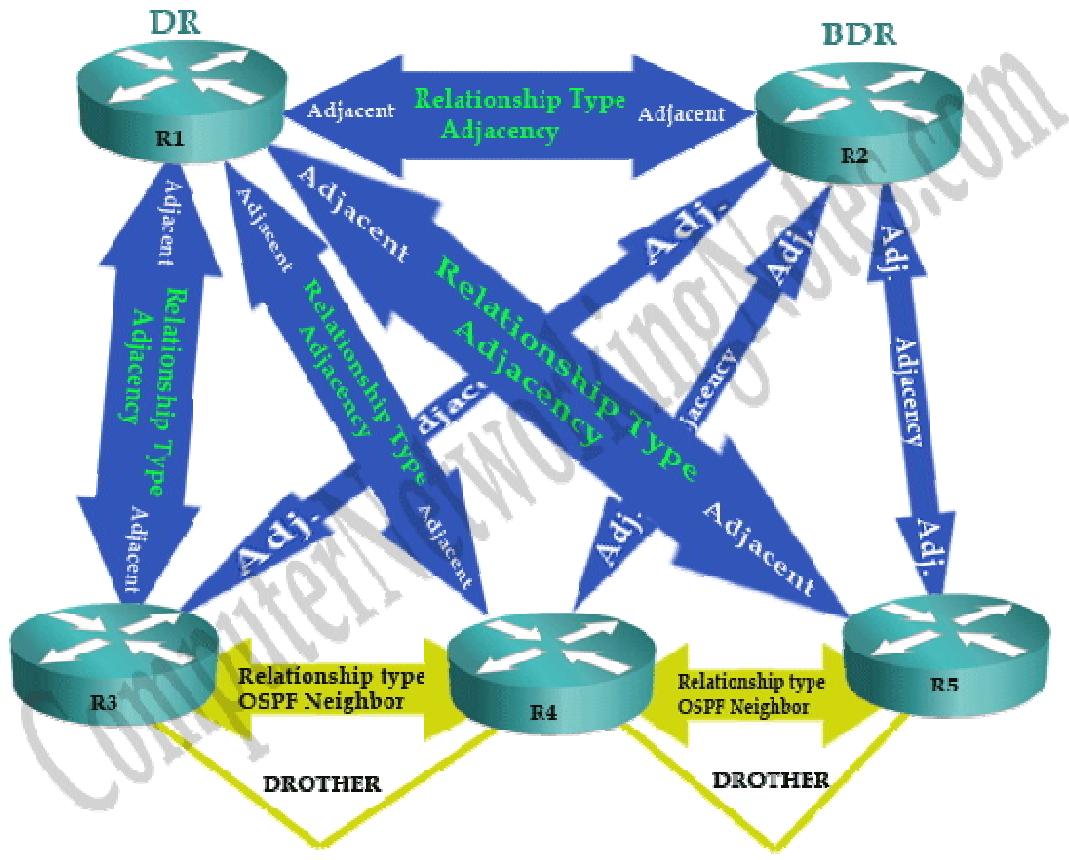
Routers which will exchange routing information are known as adjacent. Relationship between two adjacent is known as adjacency. This terminology is associated with interfaces.

A router which has two interfaces can be **adjacent** in one interface and **DROTHER** in other interface.

For example following figure illustrates an OSPF running NBMA network. In this network;

R3 will build adjacency with R1, so in this relationship they will be considered as **Adjacent**.

R3 will not build adjacency with R4, so in this relationship they will be considered only **DROTHER**.

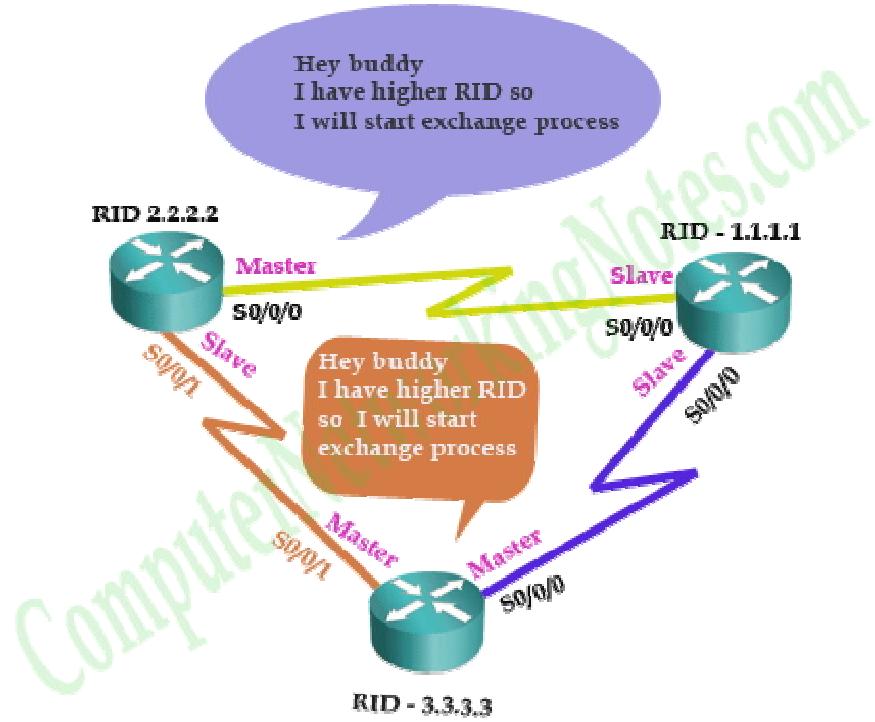


In a network which doesn't require DR and BDR, all routers will be considered as **Adjacent** and relationship between them will be considered as **Adjacency**.

Only adjacent routers will enter in next states to build the adjacency.

Exstart state

Routers who decided to build adjacency will form a master / slave relationship. In each adjacency router who has higher RID will become master and other will become slave. Do not mix Master / Slave relationship with DR/ BDR/ DROTHER relationship. Both terms look similar but have different meaning. DR/ BDR/ DROTHER relationship is built in a segment and have a wider meaning while Master / Slave relationship is built between two interfaces which need to exchange routing information. Master / Slave relationship has limited purpose. It is used to decide the Router who will start exchange process. Always Master starts exchange process.



Once routers settle down on Master/Slave, they will establish the initial sequence numbers which will be used in routing information exchange process. Sequence numbers insure that routers get most accurate information.

Exchange state

In exchange state, Master and slave decide how much information needs to be exchanged. A router that has more than one interface may learn same network information from different sources. An OSPF router is smart enough to filter the updates before receiving it. It will ask only for the updates which it does not have. In this state, routers will filter the updates which need to be exchanged.

Before we learn how routes will filter this information, let's understand few relative terms.

LSA and LSDB are explained in the first part of this tutorial. To maintain the flow of this article I am including the summary of these terms here again.

LSA

Link state advertisement (LSA) is a data packet which contains link-state and routing information. OSPF uses it share and learn network information.

LSDB

Every OSPF router maintains a Link state database (LSDB). LSDB is collection of all LSAs received by a router. Every LSA has a sequence number. OSPF stores LSA in LADB with this sequence number.

DBDs

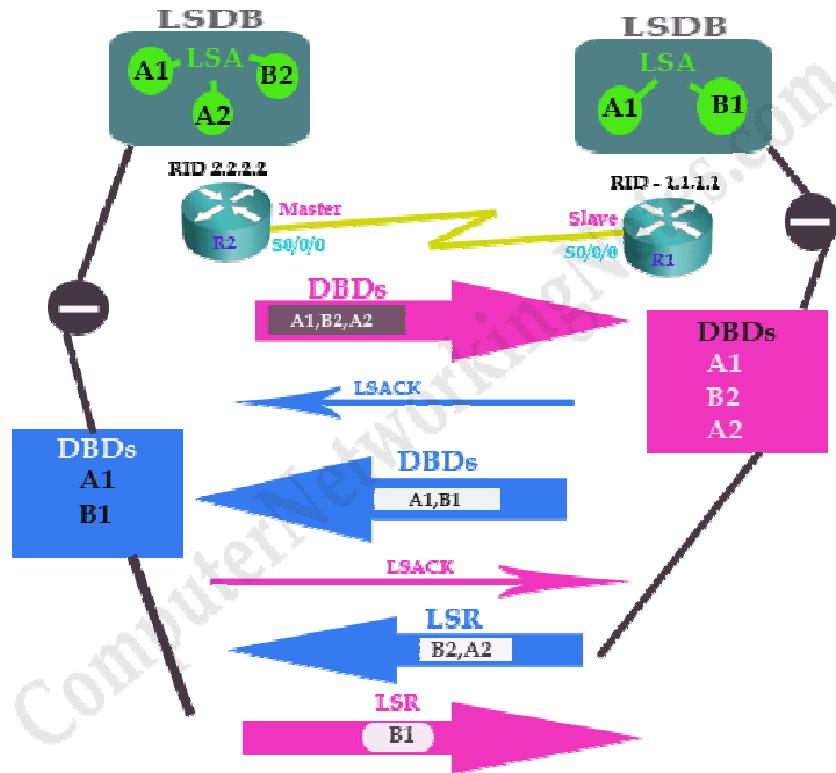
Database description packets (also referred as DDPs) contain the list of LSA. This list includes link state type, cost of link, ID of advertising router and sequence number of link. Make sure you understand this term correctly. It is only a list of all LSAs from its respective database. It does not include full LSAs.

In this state, routers exchanges DBDs. Through DBDs routers can learn which LSAs they already have. For example in following network R1 has A1, A2 and B2 LSAs in its LADB. So it will send a list of these LSAs to R2. This list is a DBDs. R2 will send an acknowledgment of receiving the list with LSACK signal. Same as R2 will send its DBDs to R1 and R1 would acknowledge that with its LSACK single.

LSR

Upon receiving DBDs, routers will compare it with their own LADB. Thus they will learn what they need to order. For example R1 received a check list (DBDs) of A1 and B1. When it will compare this list with its own LSA database (LADB), it will learn that it already has A1. So it does not need to order this LSA again. But it does not have B1, so it needs to order for this LSA. After a complete comparison, both routers will prepare a list of LSAs which they do not have in their own LADB. This list is known as LSR (Link State Request).

What other have (DBDs) – What I have (LADB) = What I need to order (LSR)



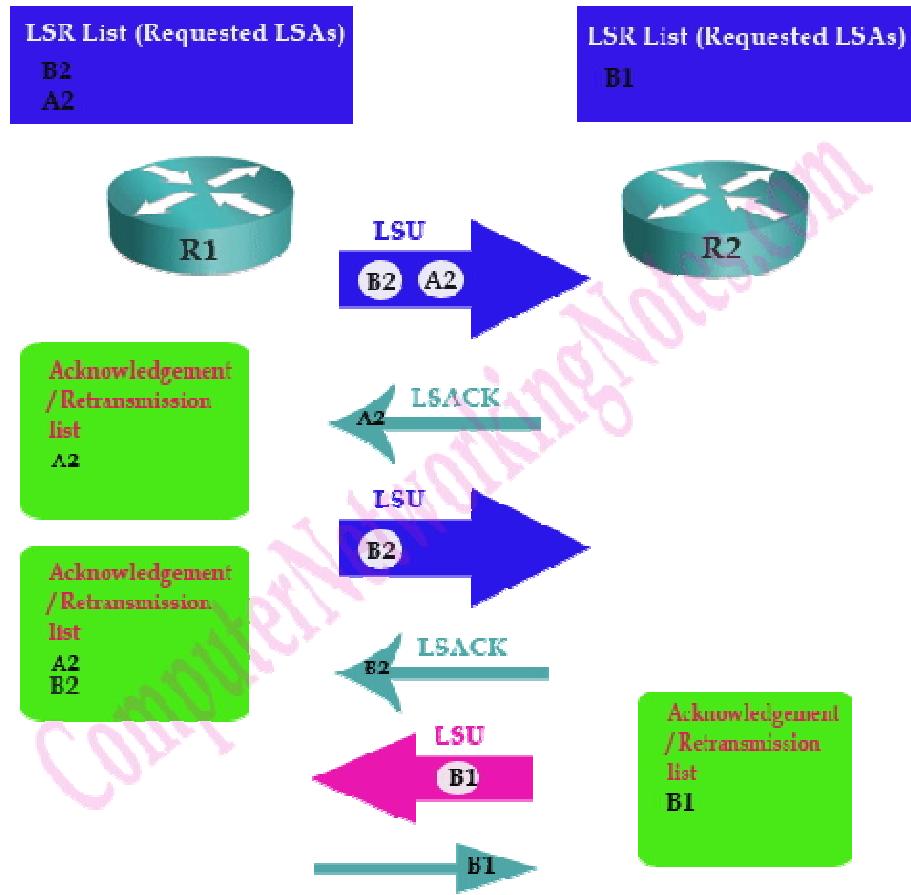
At the end of this state both routers have a list of LSAs which need to be exchanged.

Loading state

In this state actual routing information is exchanged. Routers exchange LSAs from LSR list.

Routers will use LSU (Link state update) to exchange the LSAs. Each LSA contains routing information about a particular link. Routers also maintain a retransmission list to make sure that every sent LSA is acknowledged.

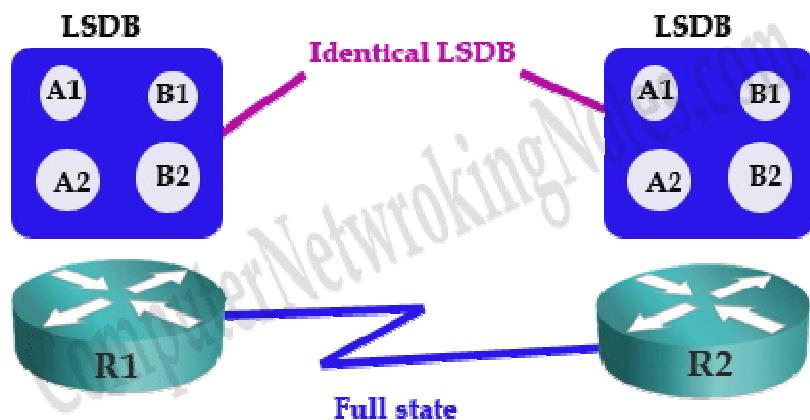
For example following figure illustrates loading state of above example. R1 sent a LSU which contain two LSAs but it received acknowledgement of only one, so it had to resend lost LSA again.



This exchange process will continue till router has any unsent LSA in LSR list.

Full state

Full state indicates that both routers have exchanged all LSAs from LSR list. Now they have identical LSDB.



Adjacent routers remain in this state for life time. This state also referred as adjacency. If any change occurs in network, routers will go through this process again.

Maintaining adjacency

- Routers will send hello messages in hello interval.
- If a router does not receive hello message from neighbor in dead interval, it will declare that neighbor as dead.
- Once a neighbor is dead, router will flood this change to other connected neighbors.
- Beside this if router detect any change in network or receive any update, it will flood that change.
- A LSA has a default lifetime of 30 minutes. Any unchanged LSAs must be reflooded in every 30 minutes.

That's all for this part. In next part, I will explain configuration part of OSPF.

To keep this tutorial simple, I used terms *neighbor* and *adjacencies* synonymously. Technically both terms are related but have different meanings especially in OSPF. Neighboring routers are defined in RFC 2328.

Neighboring routers are the routers that have interfaces in common network.

Adjacency is a relationship formed between neighboring routers for the purpose of exchanging routing information. Not every pair of neighboring routers becomes adjacent.