

Why Doesn't OSPF Form Adjacency on a PRI, BRI, or Dialer Interface?

Document ID: 13691

Contents

Introduction

Prerequisites

Requirements

Components Used

Conventions

The Problem

The Solution

Related Information

Introduction

This Tech Note explains an issue with the formation of OSPF adjacency when the dialer interfaces are configured as point-to-point links.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

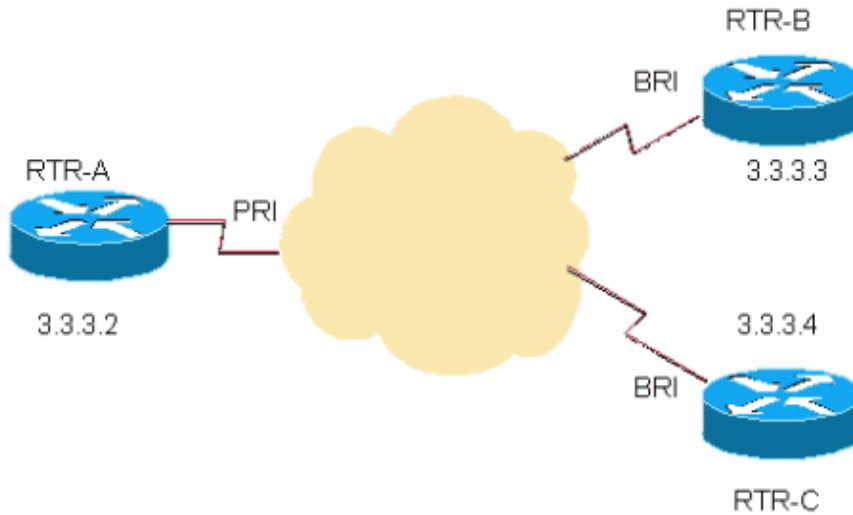
This document is not restricted to specific software and hardware versions.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

The Problem

The OSPF network-type on Primary Rate Interface (PRI), Basic Rate Interface (BRI), and dialer interfaces is point-to-point, which means that an interface can't form adjacency with more than one neighbor. A common problem when a PRI, BRI, or dialer interfaces try to form an OSPF adjacency is the neighbor gets stuck in the xstart/exchange process. Let's look at an example.



Using the **show ip ospf neighbor** command, we can see the neighbor state is stuck in "EXSTART".

```
RTR-A# show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address      Interface
3.3.3.3          1    EXSTART/ -      00:00:37   3.3.3.3     Serial6/0:23
3.3.3.4          1    EXSTART/ -      00:00:39   3.3.3.4     Serial6/0:23

RTR-B# show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address      Interface
3.3.3.2          1    EXSTART/ -      00:00:36   3.3.3.2     BRI0

RTR-C# show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address      Interface
3.3.3.2          1    EXSTART/ -      00:00:35   3.3.3.2     BRI0
```

RTR-Bs configuration shows the network-type is point-to-point:

```
RTR-B# show ip ospf interface bri0
BRI0 is up, line protocol is up (spoofing)
Internet Address 3.3.3.3/24, Area 2
Process ID 1, Router ID 3.3.3.3, Network Type POINT_TO_POINT, Cost: 1562
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:06
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 0
Suppress hello for 0 neighbor(s)
```

We can debug this situation using the **debug ip ospf adj** command. Let's look at some sample output taken while running this command on RTR-B in the figure above:

```
1: Send DBD to 3.3.3.2 on BRI0 seq 0xB41 opt 0x42 flag 0x7 len 32
2: Rcv DBD from 3.3.3.2 on BRI0 seq 0x1D06 opt 0x42 flag 0x7 len 32
   mtu 1500 state EXSTART
3: First DBD and we are not SLAVE
4: Rcv DBD from 3.3.3.2 on BRI0 seq 0xB41 opt 0x42 flag 0x2 len 92 mtu
   1500 state EXSTART
5: NBR Negotiation Done. We are the MASTER
6: Send DBD to 3.3.3.2 on BRI0 seq 0xB42 opt 0x42 flag 0x3 len 92
7: Database request to 3.3.3.2
8: sent LS REQ packet to 3.3.3.2, length 12
9: Rcv DBD from 3.3.3.2 on BRI0 seq 0x250 opt 0x42 flag 0x7 len 32
```

```
mtu 1500 state EXCHANGE
10: EXCHANGE - inconsistent in MASTER/SLAVE
11: Bad seq received from 3.3.3.2 on BRI0
12: Send DBD to 3.3.3.2 on BRI0 seq 0x2441 opt 0x42 flag 0x7 len 32
13: Rcv DBD from 3.3.3.2 on BRI0 seq 0x152C opt 0x42 flag 0x2 len 92
    mtu 1500 state EXSTART
14: Unrecognized dbd for EXSTART
15: Rcv DBD from 3.3.3.2 on BRI0 seq 0xB42 opt 0x42 flag 0x0 len 32
    mtu 1500 state EXSTART
16: Unrecognized dbd for EXSTART
```

Lines 1 – 3: RTR–B sends the first DBD to 3.3.3.2 (RTR–A) with seq 0xB41 and receives the first DBD from 3.3.3.2 (RTR–A) with seq# 0x1D06. Neighbor negotiation is still not complete.

Lines 4 – 6: RTR–B receives a reply from 3.3.3.2 (RTR–A) indicating that RTR–A received RTR–B's first DBD. Since RTR–B has the higher router ID, RTR–A elects itself slave. After receiving the acknowledgement from RTR–A, RTR–B declares itself master and sends the first DBD with data in it. Note the sequence number, which is 0xB42. Since RTR–B is the master, only it can increment the sequence number.

Line 7: RTR–B requests data from RTR–A since RTR–A indicated it has more data to send (flag set to 0x2 in last DBD received from RTR–A).

Line 8: RTR–B sends a link–state request packet to 3.3.3.2 (RTR–A). This is an OSPF packet type 3. This packet is usually sent to the neighbor's IP address. In this case, the neighbor's IP address is its router ID.

Lines 9 – 11: RTR–B receives a reply from slave (RTR–A) with a completely different sequence number and a flag of 0x7, which is the init flag. This DBD was intended for another router (most likely RTR–C), but RTR–B incorrectly received it. RTR–B declares there is a discrepancy because a flag of 0x7 means the slave has changed its status to master by setting the MS (Master/Slave) bit during the adjacency exchange. RTR–B also complains about the sequence number because it's out of order. The slave should always follow the master's sequence number.

Line 12: RTR–B re–initializes the adjacency by sending the first DBD to 3.3.3.2 to re–elect master and slave.

Lines 13 – 14: RTR–B receives a DBD from 3.3.3.2 (RTR–A), indicating that it's a slave, without recognizing RTR–B's sequence number. RTR–B declares that it doesn't recognize this DBD since the master and slave negotiation is not complete yet. This DBD packet was intended for another router.

Line 15: RTR–B receives a reply from 3.3.3.2 (RTR–A) for the old DBD, but it's too late because RTR–B has already re–initialized the adjacency process.

Line 16: RTR–B fails to recognize this DBD because it is for an "old" adjacency, which RTR–B has already torn down.

This process will repeat endlessly.

The Solution

According to section 8.1 of RFC 2328 [↗](#), OSPF sends a multicast packet for a point–to–point network–type even after the interface achieves the 2–way state. Since RTR–A is trying to form adjacencies with both RTR–B and RTR–C, RTR–B receives DBD packets meant for RTR–C and RTR–C receives DBD packets meant for RTR–B.

To solve this problem, change the network type on all routers to point–to–multipoint. This changes the

behavior of OSPF to send unicast packets after the 2-way state. Now RTR-B receives only packets destined for itself and RTR-C receives packets destined for itself. Changing the network-type in this way ensures that the OSPF router will form adjacency on a PRI, BRI, or dialer interface.

To change the network-type, enter the following configuration commands, ending each line by pressing ENTER. We'll change RTR-B as an example.

```
RTR-B# configure terminal
RTR-B(config)# int bri 0
RTR-B(config-if)# ip ospf network point-to-multipoint
RTR-B(config-if)# end
```

Now if we look at the **show** commands for RTR-B, we can verify that the network-type is point-to-multipoint and the state is full.

```
RTR-B# show ip ospf interface bri0
BRI0 is up, line protocol is up (spoofing)
  Internet Address 3.3.3.3/24, Area 2
  Process ID 1, Router ID 3.3.3.3, Network Type POINT_TO_MULTIPOINT, Cost: 1562
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:16
  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 172.16.141.10
  Suppress hello for 0 neighbor(s)
```

```
RTR-B# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.141.10	1	FULL/ -	00:01:36	3.3.3.2	BRI0

Related Information

- [Configuring BRI-to-BRI Dialup with DDR Dialer Maps](#)
- [OSPF Support Page](#)
- [Technical Support – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Aug 10, 2005

Document ID: 13691
