

排除阻滞在Exstart/Exchange状态的开放最短路径优先邻居故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[Exstart 状态](#)

[交换状态](#)

[邻居停滞在准备/交换状态](#)

[解决方案](#)

[相关信息](#)

简介

本文档介绍如何对开放最短路径优先(OSPF)邻居停滞在Exstart和Exchange状态的情况进行故障排除。

先决条件

要求

建议用户熟悉基本的OSPF操作和配置，尤其是OSPF[邻居状态](#)。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Cisco 2503 路由器
- 在两台^路由器上运行的Cisco IOS®软件版本12.2(24a)

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

规则

有关文件规则的更多信息请参见“Cisco技术提示规则”。

背景信息

形成邻接关系的OSPF状态为Down、Init、2-way、Exstart、Exchange、Loading和Full。导致开放

最短路径优先 (OSPF) 邻居停滞在 exstart/exchange 状态的原因有许多种。本文档重点介绍导致 Exstart/Exchange状态的OSPF邻居之间的MTU不匹配。有关如何排除OSPF故障的详细信息，请参阅[排除OSPF故障](#)。

Exstart 状态

当两个 OSPF 邻居路由器建立双向通信并完成 DR/BDR 选择 (在多路访问网络上) 之后，路由器将转换到 exstart 状态。在此状态下，相邻路由器建立主/从关系，并确定交换DBD数据包时使用的初始数据库描述符(DBD)序列号。

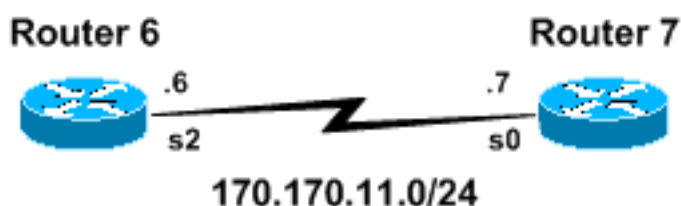
交换状态

一旦 Primary/Subordinate 已协商关系 (具有最高路由器ID的路由器成为主路由器) ，相邻路由器会转换到Exchange状态。在此状态下，路由器将交换 DBD 数据包，这些数据包描述了其整个链路状态数据库。路由器还会发送链路状态请求数据包，该数据包请求来自邻居的最新链路状态通告(LSA)。

虽然 OSPF 邻居在正常的 OSPF 邻接关系构建过程中会转入 exstart/exchange 状态，但是 OSPF 邻居停滞在此状态是不正常的。下一节介绍OSPF邻居停滞在此状态的最常见原因。有关不同 OSPF状态的详细信息，请参阅[OSPF邻居状态](#)。

邻居停滞在准备/交换状态

当您尝试在Cisco路由器和其他供应商路由器之间运行OSPF时，问题最常出现。当的最大传输单位 (MTU)设置发生于 neighboring 路由器接口不匹配。如果MTU较高的路由器发送的数据包大于相邻路由器上设置的MTU，则相邻路由器会忽略该数据包。当出现此问题时， show ip ospf neighbor 命令显示的输出类似于下图。



Router 6和Router7通过帧中继连接

本节介绍此问题的实际再现。

图中的路由器6和路由器7通过帧中继连接，并且路由器6已配置了5条静态路由，这些路由已重分发到OSPF。路由器 6 上串行接口的默认 MTU 为 1500，而路由器 7 上串行接口的 MTU 为 1450。表中显示了每台路由器的配置 (仅显示必要的配置信息) ：

路由器 6 配置

```
interface Serial2
!--- MTU is set to its default value of 1500.
no ip address
no ip directed-broadcast
encapsulation frame-relay
no ip mroute-cache
```

路由器 7 配置

```
interface Serial0
mtu 1450
no ip address
no ip directed-broadcast
encapsulation frame-relay
frame-relay lmi-type ANSI
!
interface Serial0.6 point-to-p
```

```

frame-relay lmi-type ansi
!
interface Serial2.7 point-to-point
 ip address 10.170.10.6 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 101
!
router ospf 7
 redistribute static subnets
 network 10.170.10.0 0.0.0.255 area 0
!
ip route 192.168.0.10 255.255.255.0 Null0
ip route 192.168.10.10 255.255.255.0 Null0
ip route 192.168.10.0 255.255.255.0 Null0
ip route 192.168.37.10 255.255.255.0 Null0
ip route 192.168.38.10 255.255.255.0 Null0

```

```

 ip address 172.16.7.11
 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 11
!
router ospf 7
 network 172.16.11.6 0.0.0.255
 0

```

每个路由器的 **show ip ospf neighbor** 命令输出为：

```
router-6# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
172.16.7.11	1	EXCHANGE/ -	00:00:36	172.16.7.11	Serial2.7

```
router-6#
```

```
router-7# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.170.10.6	1	EXSTART/ -	00:00:33	10.170.10.6	Serial0.6

```
router-7#
```

当路由器 6 在 exchange 状态下发送大于 1450 字节 (路由器 7 的 MTU) 的 DBD 数据包时，将会发生问题。请使用 `debug ip packet` 和 `debug ip ospf adj` 命令查看 OSPF 邻接过程。路由器 6 和 7 从步骤 1 到 14 的输出依次为：

1. 路由器 6 的 debug 输出：

```
<<<ROUTER 6 IS SENDING HELLOS BUT HEARS NOTHING, STATE OF NEIGHBOR IS DOWN>>>
00:03:53: OSPF: 172.16.7.11 address 172.16.7.11 on Serial2.7 is dead 00:03:53: OSPF:
172.16.7.11 address 172.16.7.11 on Serial2.7 is dead, state DOWN
```

2. 路由器 7 的 debug 输出：

```
<<<OSPF NOT ENABLED ON ROUTER7 YET>>>
```

3. 路由器 6 的 debug 输出：

```
<<<ROUTER 6 SENDING HELLOS>>>
00:03:53: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7), len 64, sending
broad/multicast, proto=89 00:04:03: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7), Len
64, sending broad/multicast, proto=89
```

4. 路由器 7 的 debug 输出：

```
<<<OSPF NOT ENABLED ON ROUTER7 YET>>>
```

5. 路由器 7 的 debug 输出：

```
<<<OSPF ENABLED ON ROUTER 7, BEGINS SENDING HELLOS AND BUILDING A ROUTER LSA>>>
00:17:44: IP: s=172.16.7.11 (local), d=224.0.0.5 (Serial0.6), Len 64, sending
broad/multicast, proto=89 00:17:44: OSPF: Build router LSA for area 0, router ID
172.16.7.11, seq 0x80000001
```

6. 路由器 6 的 debug 输出：

```
<<<RECEIVE HELLO FROM ROUTER7>>>
00:04:04: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5, Len 64, rcvd 0, proto=89 00:04:04:
OSPF: Rcv hello from 172.16.7.11 area 0 from Serial2.7 172.16.7.11 00:04:04: OSPF: End of
hello processing
```

7. 路由器 6 的 debug 输出：

```
<<<ROUTER 6 SEND HELLO WITH ROUTER7 ROUTERID IN THE HELLO PACKET>>>
```

```
00:04:13: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7), Len 68, sending
broad/multicast, proto=89
```

8. 路由器 7 的 debug 输出：

```
<<<ROUTER 7 RECEIVES HELLO FROM ROUTER6 CHANGES STATE TO 2WAY>>>
00:17:53: IP: s=10.170.10.6 (Serial0.6), d=224.0.0.5, Len 68, rcvd 0, proto=89 00:17:53:
OSPF: Rcv hello from 10.170.10.6 area 0 from Serial0.6 10.170.10.6 00:17:53: OSPF: 2 Way
Communication to 10.170.10.6 on Serial0.6, state 2WAY
```

9. 路由器 7 的 debug 输出：

```
<<<ROUTER 7 SENDS INITIAL DBD PACKET WITH SEQ# 0x13FD>>>
00:17:53: OSPF: Send DBD to 10.170.10.6 on Serial0.6 seq 0x13FD opt 0x2 flag 0x7 Len 32
00:17:53: IP: s=172.16.7.11 (local), d=224.0.0.5 (Serial0.6), Len 52, sending
broad/multicast, proto=89 00:17:53: OSPF: End of hello processing
```

10. 路由器 6 的 debug 输出：

```
<<<ROUTER 6 RECEIVES ROUTER7'S INITIAL DBD PACKET CHANGES STATE TO 2-WAY>>>
00:04:13: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5, Len 52, rcvd 0, proto=89 00:04:13:
OSPF: Rcv DBD from 172.16.7.11 on Serial2.7 seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450
state INIT 00:04:13: OSPF: 2 Way Communication to 172.16.7.11 on Serial2.7, state 2WAY
```

11. 路由器 6 的 debug 输出：

```
<<<ROUTER 6 SENDS DBD PACKET TO ROUTER 7 (PRIMARY/SUBORDINATE NEGOTIATION - ROUTER 6 IS
SUBORDINATE)>>>

00:04:13: OSPF: Send DBD to 172.16.7.11 on Serial2.7
seq 0xE44 opt 0x2 flag 0x7 Len 32
00:04:13: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7),
Len 52, sending broad/multicast, proto=89
00:04:13: OSPF: NBR Negotiation Done. We are the SLAVE
```

12. 路由器 7 的 debug 输出：

```
<<<RECEIVE ROUTER 6'S INITIAL DBD PACKET (MTU MISMATCH IS RECOGNIZED)>>>
00:17:53: IP: s=10.170.10.6 (Serial0.6), d=224.0.0.5, Len 52, rcvd 0, proto=89 00:17:53:
OSPF: Rcv DBD from 10.170.10.6 on Serial0.6 seq 0xE44 opt 0x2 flag 0x7 Len 32 mtu 1500
state EXSTART 00:17:53: OSPF: Nbr 10.170.10.6 has larger interface MTU
```

13. 路由器 6 的 debug 输出：

```
<<<SINCE ROUTER 6 IS SUBORDINATE SEND DBD PACKET WITH LSA HEADERS, SAME SEQ# (0x13FD) TO
ACK ROUTER 7'S DBD. (NOTE SIZE OF PKT)>>>

00:04:13: OSPF: Send DBD to 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x2 Len 1472
00:04:13: IP: s=10.170.10.6 (local), d=224.0.0.5 (Serial2.7),
Len 1492, sending broad/multicast, proto=89
```

14. 路由器 7 的 debug 输出：

```
<<<NEVER RECEIVE ACK TO ROUTER7'S INITIAL DBD, RETRANSMIT>>>
00:17:54: IP: s=172.16.7.11 (local), d=224.0.0.5 (Serial0.6), Len 68, sending
broad/multicast, proto=89 00:18:03: OSPF: Send DBD to 10.170.10.6 on Serial0.6 seq 0x13FD
opt 0x2 flag 0x7 Len 32 00:18:03: OSPF: Retransmitting DBD to 10.170.10.6 on Serial0.6 [1]
```

此时，路由器6继续尝试确认来自路由器7的初始DBD数据包。

```
00:04:13: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5,
Len 68, rcvd 0, proto=89
00:04:13: OSPF: Rcv hello from 172.16.7.11 area 0 from
Serial2.7 172.16.7.11
00:04:13: OSPF: End of hello processing
```

```
00:04:18: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:04:18: OSPF: Rcv DBD from 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450 state EXCHANGE
```

```
00:04:18: OSPF: Send DBD to 172.16.7.11 on Serial2.7
seq 0x13FD opt 0x2 flag 0x2 Len 1472
00:04:18: IP: s=10.170.10.6 (local), d=224.0.0.5
```

```
(Serial2.7), Len 1492, sending broad/multicast, proto=89
```

```
00:04:23: IP: s=10.170.10.6 (local), d=224.0.0.5  
(Serial2.7), Len 68, sending broad/multicast, proto=89
```

```
00:04:23: IP: s=172.16.7.11 (Serial2.7), d=224.0.0.5,  
Len 52, rcvd 0, proto=89
```

```
00:04:23: OSPF: Rcv DBD from 172.16.7.11 on Serial2.7  
seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450 state EXCHANGE
```

由于来自路由器 7 的 DBD 数据包对于路由器 7 MTU 而言太大，因此，路由器 7 从未得到路由器 6 的确认。路由器 7 将重复地重新传输 DBD 数据包。

```
0:17:58: IP: s=172.16.7.11 (local), d=224.0.0.5  
(Serial0.6), Len 52, sending broad/multicast, proto=89
```

```
00:18:03: OSPF: Send DBD to 10.170.10.6 on Serial0.6  
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:  
Retransmitting DBD to 10.170.10.6 on Serial0.6 [2]
```

```
00:18:03: IP: s=172.16.7.11 (local), d=224.0.0.5  
(Serial0.6), Len 52, sending broad/multicast, proto=89
```

```
00:18:03: IP: s=10.170.10.6 (Serial0.6), d=224.0.0.5,  
Len 68, rcvd 0, proto=89  
00:18:03: OSPF: Rcv hello from 10.170.10.6 area 0 from  
Serial0.6 10.170.10.6  
00:18:03: OSPF: End of hello processing
```

```
00:18:04: IP: s=172.16.7.11 (local), d=224.0.0.5  
(Serial0.6), Len 68, sending broad/multicast, proto=89
```

```
00:18:03: OSPF: Send DBD to 10.170.10.6 on Serial0.6  
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:  
Retransmitting DBD to 10.170.10.6 on Serial0.6 [3]
```

```
00:18:08: IP: s=172.16.7.11 (local), d=224.0.0.5  
(Serial0.6), Len 52, sending broad/multicast, proto=89  
router-7#
```

由于路由器 6 的 MTU 较大，因此，它将继续接受来自路由器 7 的 DBD 数据包，以尝试对其进行确认，从而保持在 EXCHANGE 状态。

由于路由器 7 的 MTU 较小，因此，它将忽略来自路由器 6 的确认以及一同出现的 DBD 数据包，继续重新传输初始的 DBD 数据包，从而保持在 EXSTART 状态。

在步骤 9 和 11 中，路由器 7 和路由器 6 发送其带有标志 0x7 的第一个 DBD 数据包，作为主/从协商的一部分。在 Primary/Subordinate 确定，由于路由器 ID 较高，路由器 7 被选为主路由器。步骤 13 和 14 中的标志清楚地显示，路由器 7 是主路由器（标志 0x7），路由器 6 是从属路由器（标志 0x2）。

在步骤 10 中，路由器 6 收到路由器 7 的初始 DBD 数据包，并转换到 2-way 状态。

在步骤 12 中，路由器 7 收到路由器 6 的初始 DBD 数据包，并识别出 MTU 不匹配。（由于路由器 6 在 DBD 数据包的接口 MTU 字段中说明了其接口 MTU，因此，路由器 7 可识别出 MTU 不匹配。）路由器 7 拒绝了路由器 6 的初始 DBD。路由器 7 将重新传输初始 DBD 数据包。

步骤 13 显示 Router 6, subordinate，采用路由器 7 的序列号，并发送包含其 LSA 报头的第二个 DBD 数据包，这将增加数据包的大小。但是由于此 DBD 数据包大于路由器 7 的 MTU，因此，路由器 7 从未收到此 DBD 数据包。

在步骤 13 之后，路由器 7 继续将初始 DBD 数据包重新传输到路由器 6，而路由器 6 继续发送遵循主序

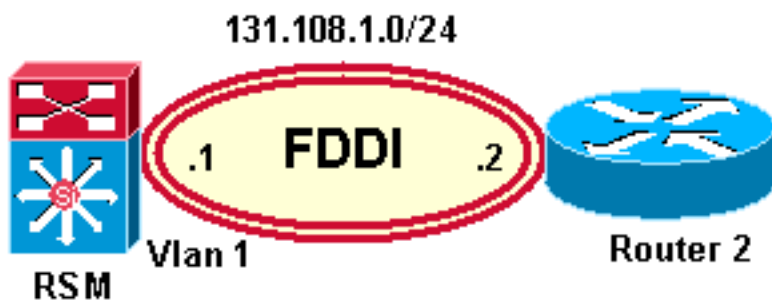
列号的DBD数据包。此循环无限期地继续下去，从而阻止任一路由器退出 exstart/exchange 状态。

解决方案

由于问题是由 MTU 不匹配导致的，因此，解决方案是更改任一路由器的 MTU，以便与邻居路由器的 MTU 匹配。

注意： Cisco IOS 软件 12.0(3) 版引入了对接口 MTU 不匹配进行检测的功能。此检测涉及在 DBD数据包中通告接口MTU的OSPF，它符合[OSPF RFC 2178](#)附录G.9。当路由器收到通告的 DBD数据包时，路由器会忽略该DBD数据包，并且邻居状态仍处于Exstart状态。这将阻碍邻接的形成。要解决此问题，请确保链路两端上的 MTU 相同。

在Cisco IOS软件12.01(3)中，`ip ospf mtu-ignore`还引入了接口配置命令来关闭MTU不匹配检测；但是，仅在极少数情况下需要用到此命令，如下图所示：



光纤分布式数据接口(FDDI)端口

上图显示了Cisco Catalyst 5000上的光纤分布式数据接口(FDDI)端口，带有路由交换模块(RSM)，该模块连接到路由器2上的FDDI接口。RSM上的虚拟LAN(VLAN)是虚拟以太网接口，MTU为1500，路由器2上的FDDI接口的MTU为4500。在交换机的 FDDI 端口上收到数据包时，数据包将会进入背板，并且在交换器自身内部将会发生 FDDI 到以太网的转换/分段。这是一个有效的设置，但使用MTU不匹配检测功能时，路由器和RSM之间不会形成OSPF邻接关系。由于FDDI和以太网MTU不同，`ip ospf mtu-ignore`命令在RSM的VLAN接口上非常有用，可停止MTU不匹配的OSPF检测并形成邻接关系。

请注意，MTU不匹配虽然最常见，但并非OSPF邻居停滞在Exstart/Exchange状态的唯一原因。此问题通常是由于无法成功交换 DBD 数据包而导致的。但是，根源可能在于以下原因之一：

- MTU 不匹配
- 单播中断。在Exstart状态下，路由器向邻居发送单播数据包，以选择Primary和Subordinate。这是正常的，但如果拥有点对点链路，就会发送多播数据包。以下为可能的原因：在高度冗余的网络中，异步传输模式(ATM)或帧中继环境中的错误虚电路(VC)映射。MTU问题，这意味着路由器只能对特定长度的数据包执行ping操作。访问列表会阻止单播数据包。NAT在路由器上运行并转换单播数据包。
- PRI 和 BRI/拨号器之间的邻居。
- 两台路由器的路由器ID相同（配置错误）。

此外，[OSPF RFC 2328](#)第10.3节指出，针对以下任何事件（任何事件都可能由内部软件问题引起）启动Exstart/Exchange进程：

- 序列号不匹配。意外的 DD 序列号.意外地设置了“l”位.选项字段与 DBD 数据包中收到的上一个

选项字段不同.

- BadLSReq邻居在交换过程中发送了无法识别的 LSA。邻居在交换过程中请求的 LSA 无法找到

当OSPF未形成邻居时，请考虑前面提到的因素，例如物理介质和网络硬件，以便排除故障。

相关信息

- [OSPF 邻居状态](#)
- [Cisco - 了解 OSPF 邻居问题](#)
- [技术支持 - Cisco Systems](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。