为什么 OSPF 邻居停滞在 Exstart/Exchange 状态？

目录

简介

形成邻接的 OSPF 状态有 Down、Init、Attempt、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 OSPF 邻居之间的 MTU 不匹配导致停滞在 exstart/exchange 状态。有关 OSPF 故障排除的详细信息，请参阅 OSPF 故障排除。

先决条件

要求

阅读本文档之前应首先熟悉 OSPF 的基本操作和配置，尤其是有关 OSPF 邻居状态的部分。

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
- 运行在两路由器的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。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 Cisco 技术提示规则。

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，相邻的两个路由器将建立主从关系，并确定交换 DBD 数据包时所使用的初始数据库描述符 (DBD) 序列号。

交换状态

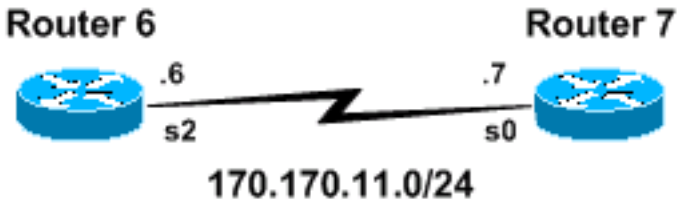
对主从关系进行协商之后（路由器 ID 最大的路由器充当主设备），两个邻居路由器将转换到 exchange 状态。在此状态下，路由器将交换 DBD 数据包，这些数据包描述了其整个链路状态数据库。路由器也发送链路状态请求数据包，请求从邻居的更加最近的Link State Advertisement (LSA)。

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。以下为 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

尝试在 Cisco 路由器和另一供应商的路由器之间运行 OSPF 时，最有可能发生问题。当相邻路由器接口的最大传输单元(MTU)设置不配比，问题发生。如果 MTU 较大的路由器向邻居路由器发送了一个数据包，该数据包的大小大于邻居路由器上设置的 MTU，则邻居路由器会忽略该数据包。发生此问题时，**show ip ospf neighbor** 命令的输出将类似于下文所介绍的形式。

本部分提供了对此问题的实际重现。



以上拓扑中的路由器 6 和路由器 7 通过帧中继进行连接，并且路由器 6 配置了重分配到 OSPF 的 5 个静态路由。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。每个路由器的配置均显示在下表中（仅显示必要的配置信息）：

路由器 6 配置	路由器 7 配置
interface Serial2 *!--- MTU sets to it's default value of 1500.* no ip address no ip directed-broadcast encapsulation frame-relay no ip mroute-cache frame-relay lmi-type ansi ! interface Serial2.7 point-to-point ip address 170.170.11.6 255.255.255.0 no ip directed-broadcast frame-relay interface-dlci 101 ! router ospf 7 redistribute static subnets network 170.170.11.0 0.0.0.255 area 0 ip	! interface Serial0 mtu 1450 no ip address no ip directed-broadcast encapsulation frame-relay frame-relay lmi-type ANSI !

route 192.79.34.0 255.255.255.0 Null0 ip route 192.79.35.0 255.255.255.0 Null0 ip route 192.79.36.0 255.255.255.0 Null0 ip route 192.79.37.0 255.255.255.0 Null0 ip route 192.79.38.0 255.255.255.0 Null0	interface Serial0.6 point-to-point ip address 170.170.11.7 255.255.255.0 no ip directed-broadcast frame-relay interface-dlci 110! ! router ospf 7 network 170.170.11.0 0.0.0.255 area 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 170.170.11.7 1
EXCHANGE/ - 00:00:36 170.170.11.7 Serial2.7 router-6# router-7# show ip ospf neighbor Neighbor
ID Pri State Dead Time Address Interface 170.170.11.6 1 EXSTART/ - 00:00:33 170.170.11.6
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。请在每个路由器上使用 **debug ip packet** 和 debug ip ospf adj 命令，以便查看 OSPF 邻接过程的发生。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：
   ```
   ***ROUTER6 IS SENDING HELLOS BUT HEARS NOTHING,
   STATE OF NEIGHBOR IS DOWN
   00:03:53: OSPF: 170.170.11.7 address 170.170.11.7 on
   Serial2.7 is dead
   00:03:53: OSPF: 170.170.11.7 address 170.170.11.7 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：
   ```
   ***ROUTER6 SENDING HELLOS
   00:03:53: IP: s=170.170.11.6 (local), d=224.0.0.5
   (Serial2.7), len 64, sending broad/multicast, proto=89
   00:04:03: IP: s=170.170.11.6 (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 ROUTER7, BEGINS SENDING
   HELLOS AND BUILDING A ROUTER LSA
   00:17:44: IP: s=170.170.11.7 (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 170.170.11.7, seq 0x80000001
   ```
6. 路由器 6 的 debug 输出：
   ```
   ***RECEIVE HELLO FROM ROUTER7
   00:04:04: IP: s=170.170.11.7 (Serial2.7), d=224.0.0.5,
   Len 64, rcvd 0, proto=89
   00:04:04: OSPF: Rcv hello from 170.170.11.7 area 0 from
   Serial2.7 170.170.11.7
   00:04:04: OSPF: End of hello processing
   ```
7. 路由器 6 的 debug 输出：
   ```
   ***ROUTER6 SEND HELLO WITH ROUTER7 ROUTERID
   IN THE HELLO PACKET
   00:04:13: IP: s=170.170.11.6 (local), d=224.0.0.5
   (Serial2.7), Len 68, sending broad/multicast, proto=89
   ```
8. 路由器 7 的 debug 输出：
   ```
   ***ROUTER7 RECEIVES HELLO FROM ROUTER6 CHANGES
   STATE TO 2WAY
   ```

```
00:17:53: IP: s=170.170.11.6 (Serial0.6), d=224.0.0.5,
Len 68, rcvd 0, proto=89
00:17:53: OSPF: Rcv hello from 170.170.11.6 area 0 from
Serial0.6 170.170.11.6
00:17:53: OSPF: 2 Way Communication to 170.170.11.6 on
Serial0.6, state 2WAY
```

9. 路由器 7 的 debug 输出：`***ROUTER7 SENDS INITIAL DBD PACKET WITH SEQ# 0x13FD`

```
00:17:53: OSPF: Send DBD to 170.170.11.6 on Serial0.6
seq 0x13FD opt 0x2 flag 0x7 Len 32
00:17:53: IP: s=170.170.11.7 (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：`***ROUTER6 RECEIVES ROUTER7'S INITIAL DBD PACKET CHANGES STATE TO 2-WAY`

```
00:04:13: IP: s=170.170.11.7 (Serial2.7), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:04:13: OSPF: Rcv DBD from 170.170.11.7 on Serial2.7
seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450 state INIT
00:04:13: OSPF: 2 Way Communication to 170.170.11.7 on
Serial2.7, state 2WAY
```

11. 路由器 6 的 debug 输出：`***ROUTER6 SENDS DBD PACKET TO ROUTER7 (MASTER/SLAVE NEGOTIATION - ROUTER6 IS SLAVE)`

```
00:04:13: OSPF: Send DBD to 170.170.11.7 on Serial2.7
seq 0xE44 opt 0x2 flag 0x7 Len 32
00:04:13: IP: s=170.170.11.6 (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 ROUTER6'S INITIAL DBD PACKET (MTU MISMATCH IS RECOGNIZED)`

```
00:17:53: IP: s=170.170.11.6 (Serial0.6), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:17:53: OSPF: Rcv DBD from 170.170.11.6 on Serial0.6
seq 0xE44 opt 0x2 flag 0x7 Len 32 mtu 1500 state EXSTART
00:17:53: OSPF: Nbr 170.170.11.6 has larger interface MTU
```

13. 路由器 6 的 debug 输出：`***SINCE ROUTER6 IS SLAVE SEND DBD PACKET WITH LSA HEADERS, SAME SEQ# (0x13FD) TO ACK ROUTER7'S DBD. (NOTE SIZE OF PKT)`

```
00:04:13: OSPF: Send DBD to 170.170.11.7 on Serial2.7
seq 0x13FD opt 0x2 flag 0x2 Len 1472
00:04:13: IP: s=170.170.11.6 (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=170.170.11.7 (local), d=224.0.0.5
(Serial0.6), Len 68, sending broad/multicast, proto=89
00:18:03: OSPF: Send DBD to 170.170.11.6 on Serial0.6
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:
Retransmitting DBD to 170.170.11.6 on Serial0.6 [1]
```

此时，路由器 6 继续不断尝试确认来自路由器 7 的初始 DBD 数据包。

```
00:04:13: IP: s=170.170.11.7 (Serial2.7), d=224.0.0.5,
Len 68, rcvd 0, proto=89
00:04:13: OSPF: Rcv hello from 170.170.11.7 area 0 from
Serial2.7 170.170.11.7
00:04:13: OSPF: End of hello processing

00:04:18: IP: s=170.170.11.7 (Serial2.7), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:04:18: OSPF: Rcv DBD from 170.170.11.7 on Serial2.7
seq 0x13FD opt 0x2 flag 0x7 Len 32 mtu 1450 state EXCHANGE
```

```
00:04:18: OSPF: Send DBD to 170.170.11.7 on Serial2.7
seq 0x13FD opt 0x2 flag 0x2 Len 1472
00:04:18: IP: s=170.170.11.6 (local), d=224.0.0.5
(Serial2.7), Len 1492, sending broad/multicast, proto=89

00:04:23: IP: s=170.170.11.6 (local), d=224.0.0.5
(Serial2.7), Len 68, sending broad/multicast, proto=89

00:04:23: IP: s=170.170.11.7 (Serial2.7), d=224.0.0.5,
Len 52, rcvd 0, proto=89
00:04:23: OSPF: Rcv DBD from 170.170.11.7 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=170.170.11.7 (local), d=224.0.0.5
(Serial0.6), Len 52, sending broad/multicast, proto=89
00:18:03: OSPF: Send DBD to 170.170.11.6 on Serial0.6
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:
Retransmitting DBD to 170.170.11.6 on Serial0.6 [2]

00:18:03: IP: s=170.170.11.7 (local), d=224.0.0.5
(Serial0.6), Len 52, sending broad/multicast, proto=89
00:18:03: IP: s=170.170.11.6 (Serial0.6), d=224.0.0.5,
Len 68, rcvd 0, proto=89
00:18:03: OSPF: Rcv hello from 170.170.11.6 area 0 from
Serial0.6 170.170.11.6
00:18:03: OSPF: End of hello processing

00:18:04: IP: s=170.170.11.7 (local), d=224.0.0.5
(Serial0.6), Len 68, sending broad/multicast, proto=89

00:18:03: OSPF: Send DBD to 170.170.11.6 on Serial0.6
seq 0x13FD opt 0x2 flag 0x7 Len 32 00:18:03: OSPF:
Retransmitting DBD to 170.170.11.6 on Serial0.6 [3]

00:18:08: IP: s=170.170.11.7 (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。

请看以上某些步骤。在步骤 9 和步骤 11 中，路由器 7 和路由器 6 将发送各自的第一个 DBD 数据包，这些数据包带有标志 0x7 设置作为主从协商的一部分。确定主从关系之后，由于路由器 7 的路由器 ID 较大，因此，它被选为主设备。步骤 13 和步骤 14 中的标志清楚地显示出路由器 7 为主设备（标志为 0x7），路由器 6 为从设备（标志为 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 显示了作为从设备的路由器 6 采用了路由器 7 的序列号，并发送了包含自身 LSA 报头的第二个 DBD 数据包，而此 LSA 报头还会增加数据包的大小。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。

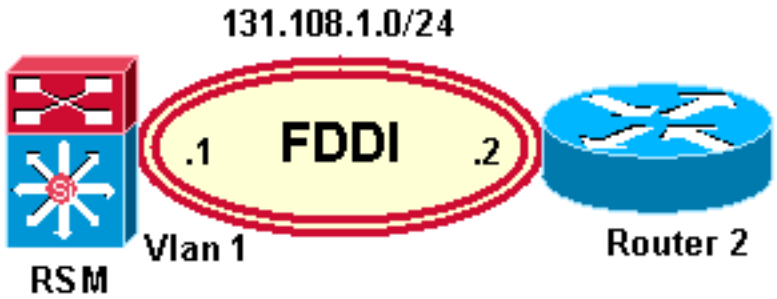
在步骤 13 之后，路由器 7 继续将初始 DBD 数据包重新传输到路由器 6，而路由器 6 则继续发送跟随主设备序列号的 DBD 数据包。此循环无限期地继续下去，从而阻止任一路由器退出 exstart/exchange 状态。

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。请注意，Cisco IOS 不支持更改 LAN 接口（如以太网或令牌环）上的物理 MTU。如果在 LAN 介质上的 Cisco 路由器和另一供应商路由器之间发生问题，请调整另一供应商路由器上的 MTU。

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。此检测包括 OSPF 在 DBD 数据包中通告接口 MTU，这与 OSPF RFC 2178 附录 G.9 中的内容一致。 如果 DBD 数据包通告 MTU 大于路由器可接收的 MTU，则当路由器收到此 DBD 数据包时会将其忽略，并且邻居保持在 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) 中还引入了 ip ospf mtu-ignore 接口配置命令，此命令可关闭对 MTU 不匹配的检测；但是，仅在极少数情况下需要用到此命令，如下图所示：



上述图表显示一台思科Catalyst 5000的光纤分布式数据接口(FDDI)端口用路由交换模块(RSM)连接对在Router2的一个FDDI接口。RSM 上的虚拟 LAN (VLAN) 是 MTU 为 1500 的虚拟以太网接口，而路由器 2 上 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。这是有效的设置，但由于使用了 MTU 不匹配检测功能，OSPF 邻接将不会在路由器和 RSM 之间形成。由于 FDDI 和以太网 MTU 不同，因此，该 **ip ospf mtu-ignore** 命令对于 RSM 的 VLAN 接口十分有用，该命令可使 OSPF 停止检测 MTU 不匹配并形成邻接。

尤其值得注意的是，虽然 MTU 不匹配是最常见的原因，但它并不是 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，路由器将会向邻居发送单播数据包，以选择主从设备。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：映射在异步传输模式 (ATM)或帧中继环境的错误的虚拟电路在极为冗长的网络。MTU 问题，即路由器仅可 ping 特定长度的数据包。访问列表阻止了单播数据包。NAT 正在路由器上运行，并且正在转换单播数据包。
- PRI 和 BRI/拨号器之间的邻居。
- 两个路由器具有相同的路由器 ID（错误配置）。

此外，OSPF RFC 2328 第 10.3 部分中指明，针对以下任一事件都会启动 exstart/exchange 过程

（而其中任一事件都可能由内部软件问题引起）：

- SequenceNumberMismatch意外的 DD 序列号意外地设置了“I”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

如果 OSPF 无法形成邻接，请考虑上述因素（例如物理介质和网络硬件），以便对问题进行故障排除。

相关信息

- OSPF 邻居状态
- Cisco - 了解 OSPF 邻居问题
- 为什么 show ip ospf neighbor 命令显示邻居阻塞在初始状态？
- 为什么 show ip ospf neighbor 命令显示邻居阻塞在双向状态？
- OSPF 支持页
- 技术支持 - Cisco Systems