

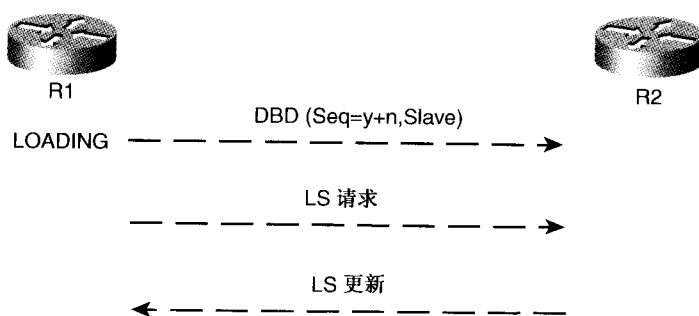


图 8-35 OSPF 调入状态

8.5.8 OSPF Full 状态

这个状态表示在 OSPF 邻居间已经交换了完整的信息。在图 8-36 中, R1 和 R2 已经交换了它们整个数据库信息并且都处于 Full 状态中。

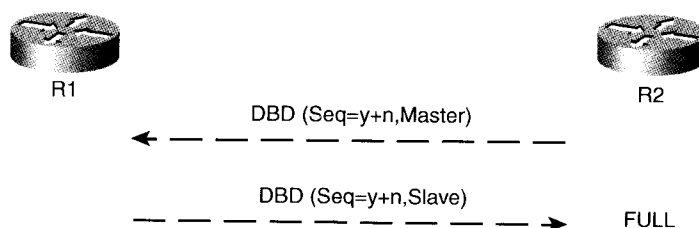


图 8-36 OSPF Full 状态

8.6 小 结

OSPF 是一种链路状态协议。OSPF 有 5 种分组——Hello、DBD、链路状态请求、链路状态更新和链路状态确认。根据邻接体的状态来使用这些分组。存在多种类型的 LSA, 它们是路由器 LSA、网络 LSA、总 LSA、SBR 汇总 LSA、外部 LSA 和 NSSA LSA。OSPF 有多种区域类型, 它们是普通区域、端区、完全端区、NSSA 区域和完全 NSSA 区域。可根据网络需求来使用这些区域。最受限的区域形式是完全端区, 只依靠它从 ABR 接收到的缺省汇总路由来进行路由。

OSPF 可以在多种介质类型选项中工作——多点接入、点到点、NBMA 和请求线路。在非完全网状互连 NBMA 环境中, 强烈推荐使用的网络类型是点到多点类型。点到多点非广播网络仅在介质不支持组播功能时才有用。在这种网络类型中没有 DR 和 BDR 的选举。

OSPF 邻接体关系在形成以前要经历多个阶段。邻接体的最后阶段是 Full 状态, 它表示邻居间已经交换了一个完整的数据库。在广播介质中, 只与 DR 和 BDR 形成邻接体关系。所有其它邻居达到 2-way 状态。这是为了减少邻接体数目以使网段中存在更少的泛洪流量。

8.7 复 习 题

1. OSPF 中有多少类型的分组？
2. 哪一种 LSA 有一个叫做转发地址的字段？
3. 哪一种 LSA 不允许在完全端区中使用？
4. OSPF 路由器的组播地址是什么？
5. 哪个 OSPF 协议分组被用于选举主/从设备？
6. OSPF 协议的哪个分组实现了 LSA 泛洪？
7. 在一个 LSA 被宣布达到 MAXAGED 状态之前的以秒为单位的计时时限是多少？
8. 一个通用 LSA 头有多少个字节长？

第 9 章

OSPF 协议疑难解析

本章涵盖下列关键主题：

- OSPF 邻居关系疑难解析。
- OSPF 路由通告疑难解析。
- OSPF 路由安装疑难解析。
- OSPF 中路由重发布疑难解析。
- OSPF 中路由汇总疑难解析。
- CPUHOG 问题疑难解析。
- OSPF 中按需拨号路由（DDR）问题疑难解析。
- SPF 计算问题和路由抖动问题疑难解析。
- 常见 OSPF 出错消息。

本章讨论 OSPF 的常见问题同时给出如何解决这些问题的办法。与 RIP 和 IGRP 比较而言，OSPF 是一个复杂的协议。有时，问题可以被相当容易地解决并且只需要少量的配置改变。而其它一些时候，问题可能非常复杂，需要更多的帮助。

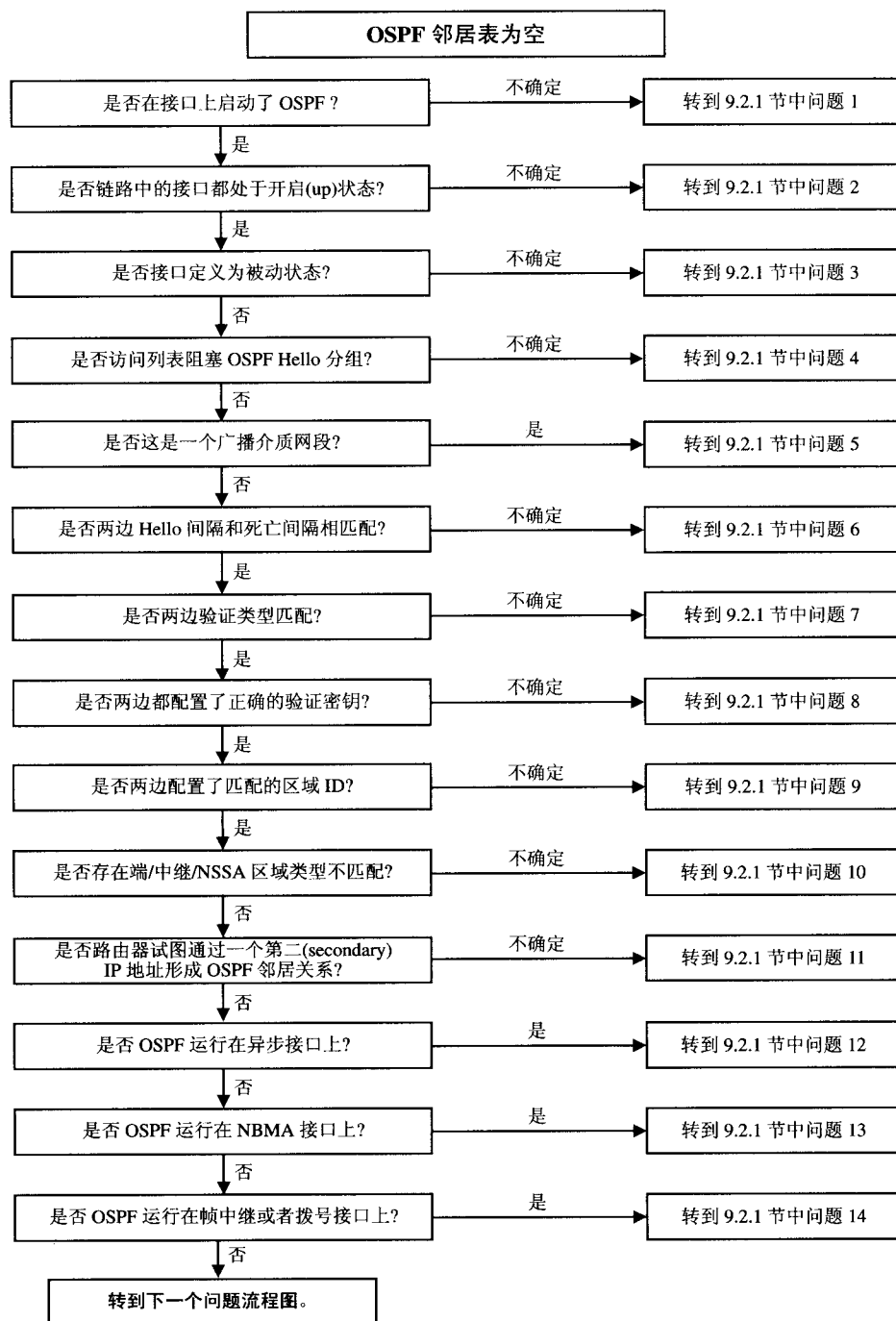
本章讨论几种类型的问题。这些例子是多年来从实际的用户网络环境中收集来的。

一些问题需要开启调试功能。OSPF 中的调试通常不会需要占用很多的 CPU 时间，除非这个问题影响到了整个 OSPF 网络。例如，如果 OSPF 邻居没有建立起来，则使用 `debug ip ospf adj` 命令不会占用很多 CPU 时间，除非 300 个邻居同时出现了问题。

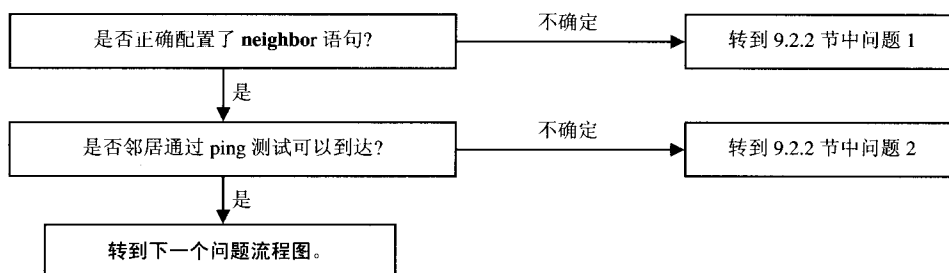
本章中使用流程图后面跟着如何解决常见 OSPF 问题的文档这样一种方法来进行 OSPF 疑难解析。

9.1 解决 OSPF 常见问题的流程图

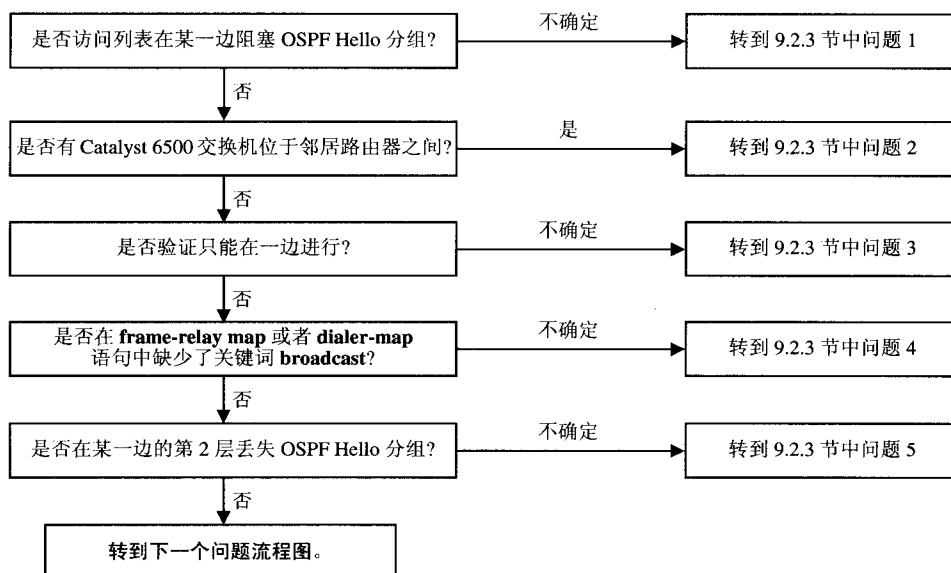
9.1.1 OSPF 邻居关系疑难解析



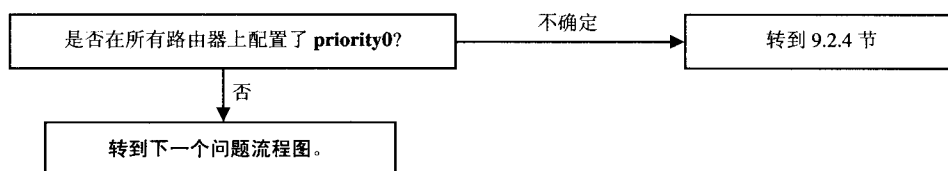
OSPF 邻居停滞于(Stuck in)ATTEMPT 状态



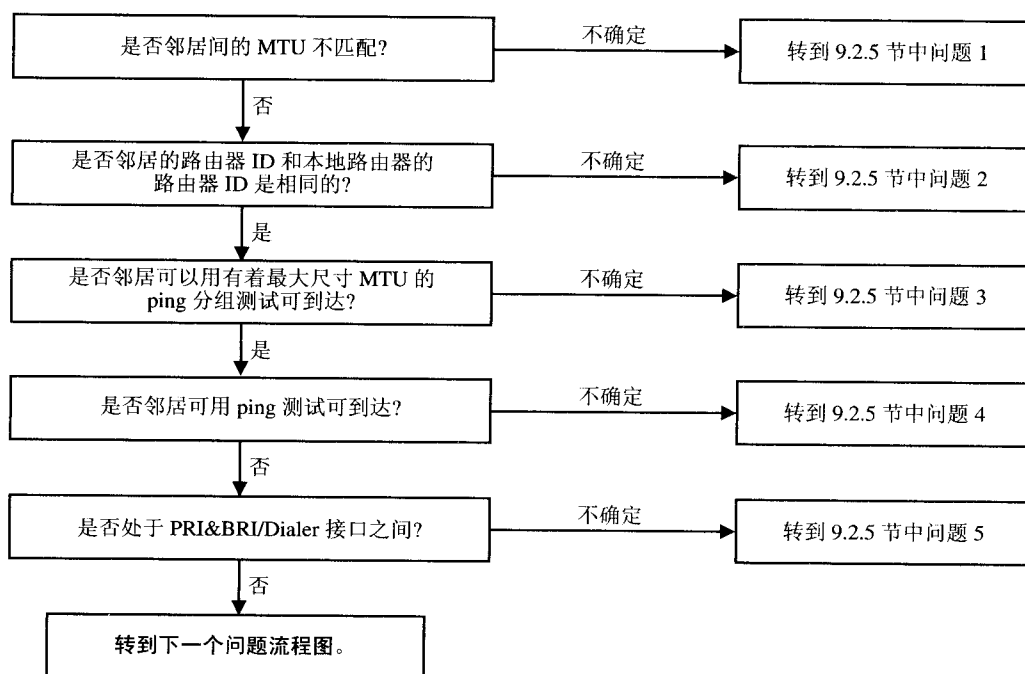
OSPF 邻居停滞于(Stuck in)INIT 状态



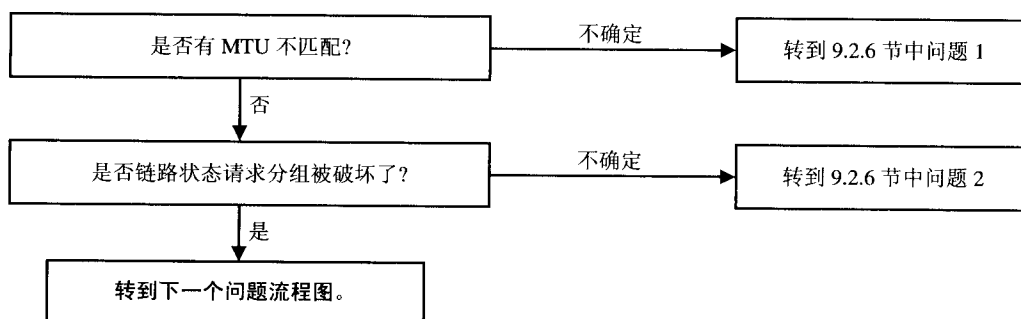
OSPF 邻居停滞于 2-way 状态



OSPF 邻居停滞于 EXSTART/EXCHANGE 状态

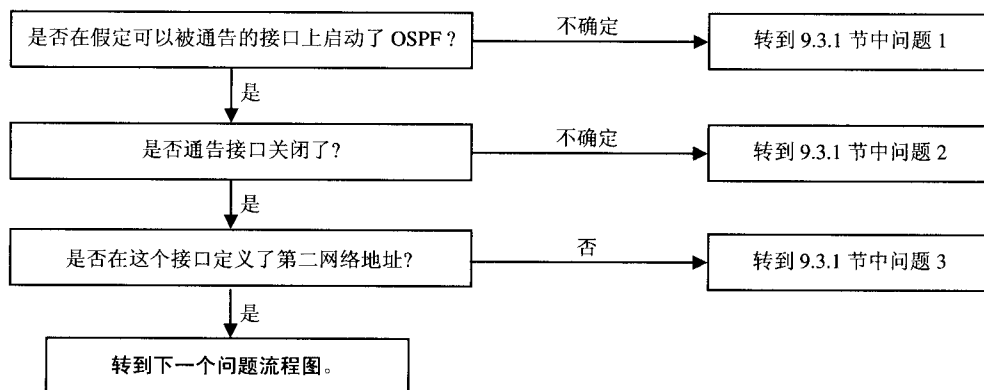


OSPF 邻居停滞于 LOADING 状态

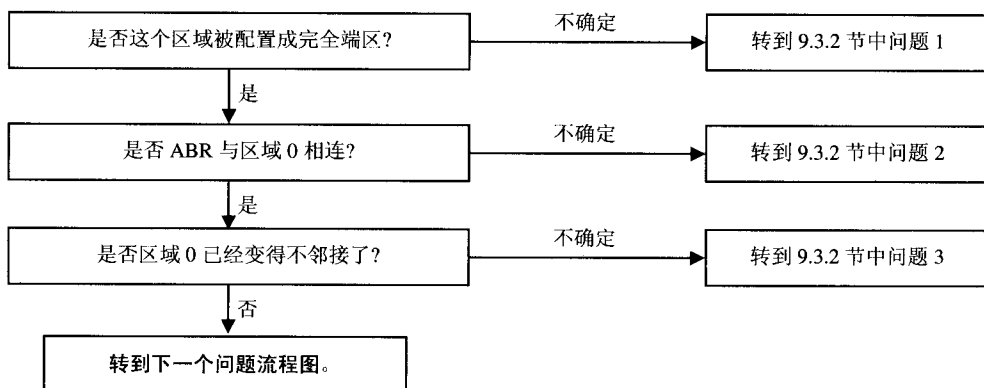


9.1.2 OSPF 路由通告疑难解析

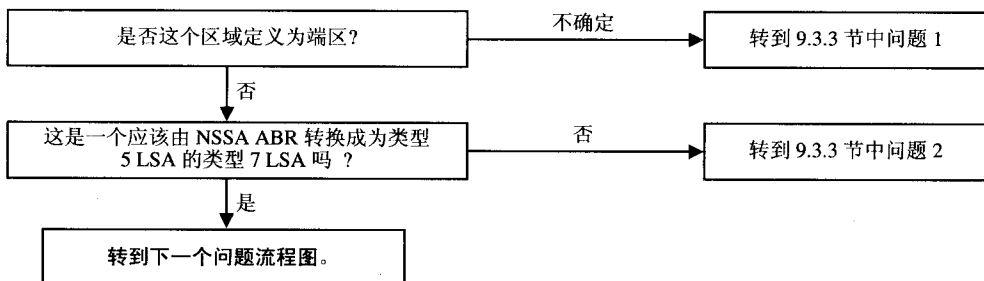
OSPF 邻居不通告路由



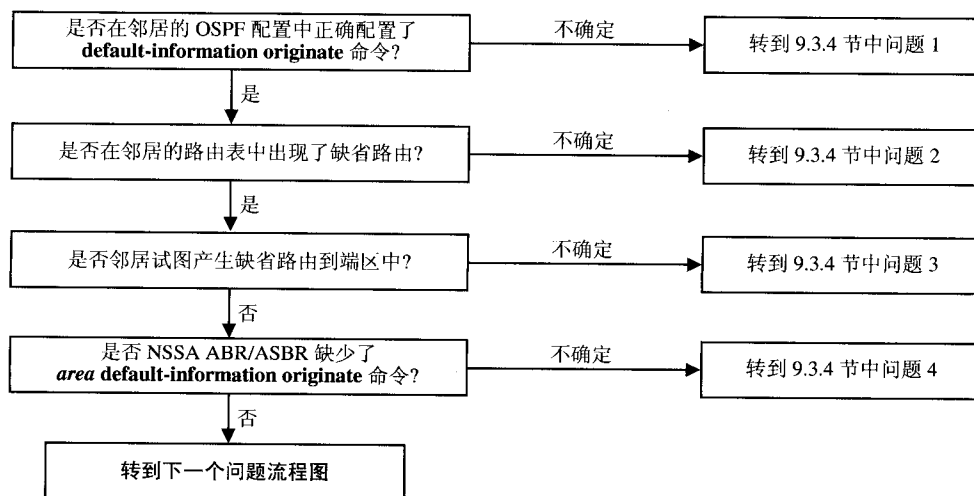
OSPF 邻居(ABR)不通告汇总路由



OSPF 邻居不通告外部路由

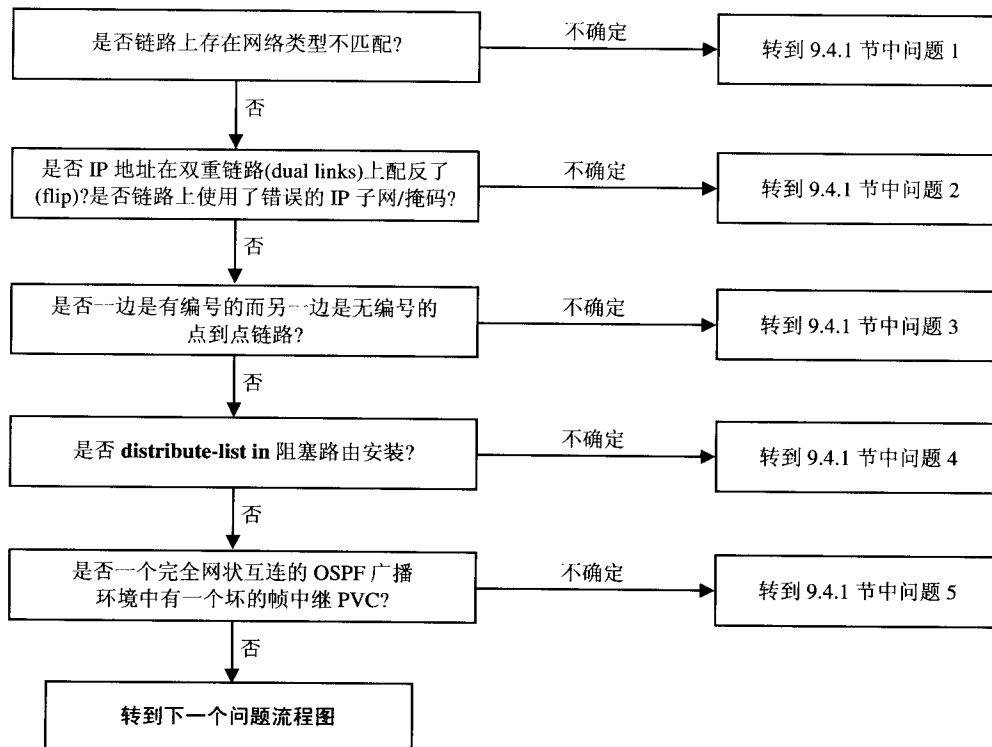


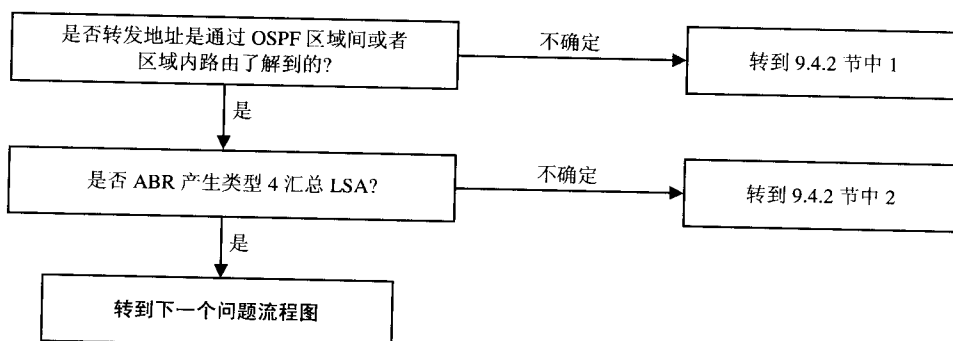
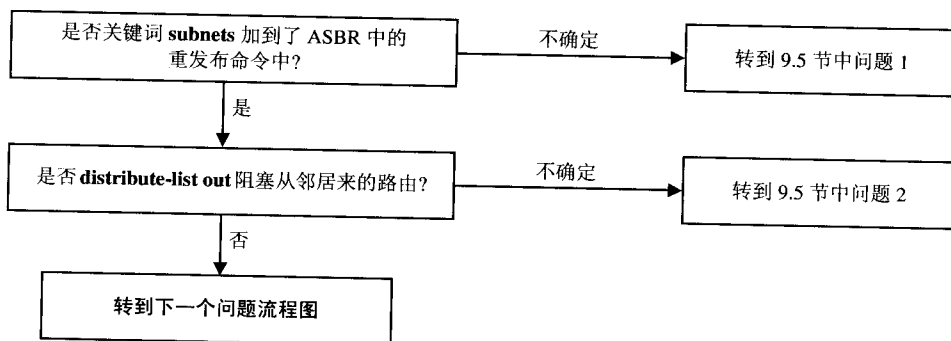
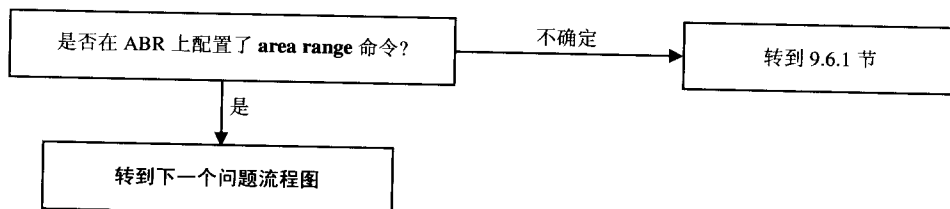
OSPF 邻居不通告缺省路由

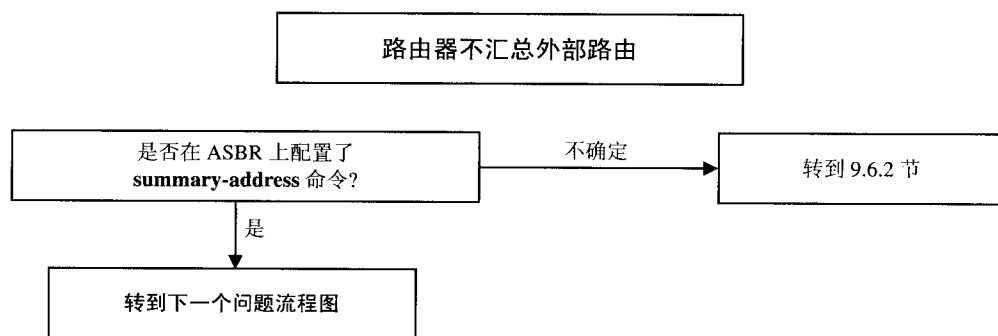


9.1.3 OSPF 路由安装疑难解析

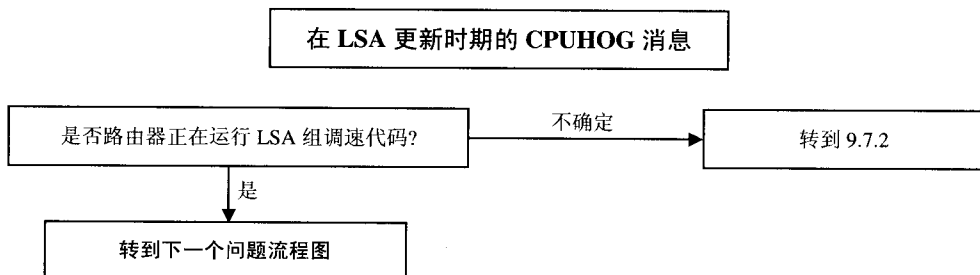
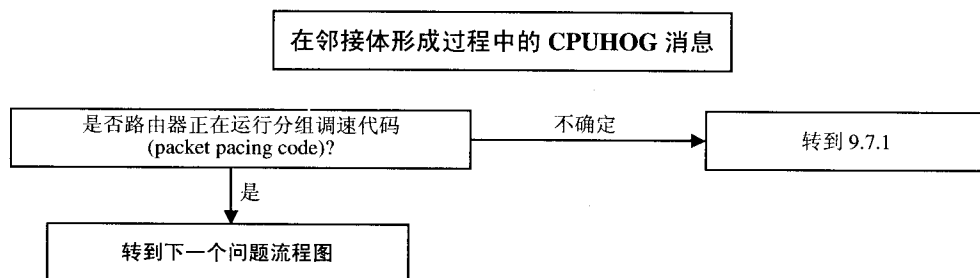
OSPF 没有在路由表中安装任何路由



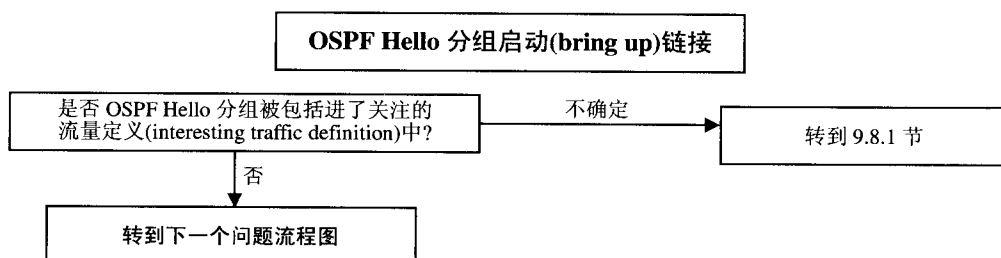
OSPF 没有在路由表中安装外部路由**9.1.4 OSPF 中路由重发布疑难解析****OSPF 邻居不通告外部路由****9.1.5 OSPF 中路由汇总疑难解析****路由器不汇总域间路由**

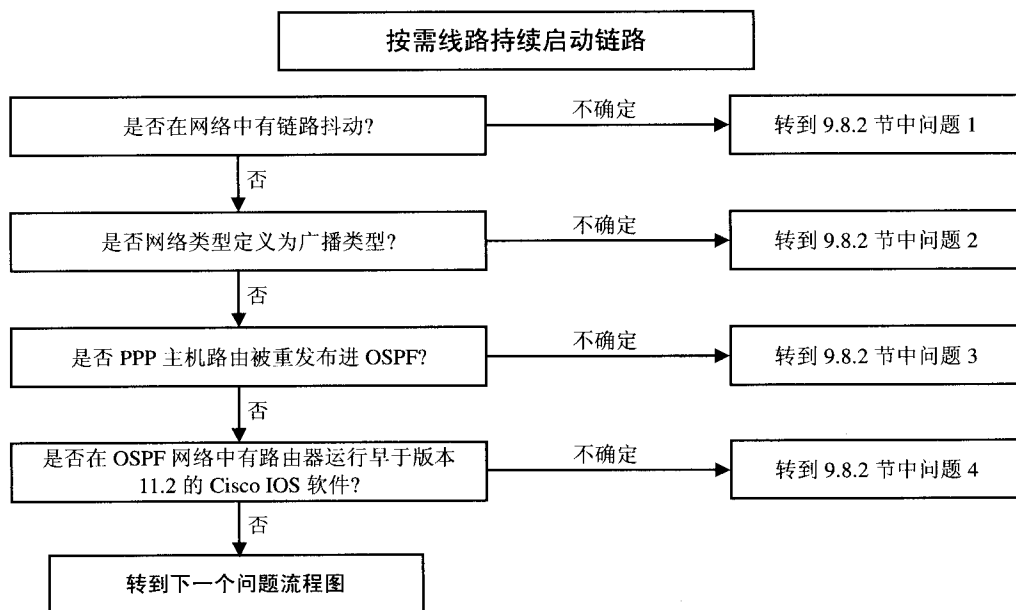


9.1.6 “CPUHOG” 问题疑难解析

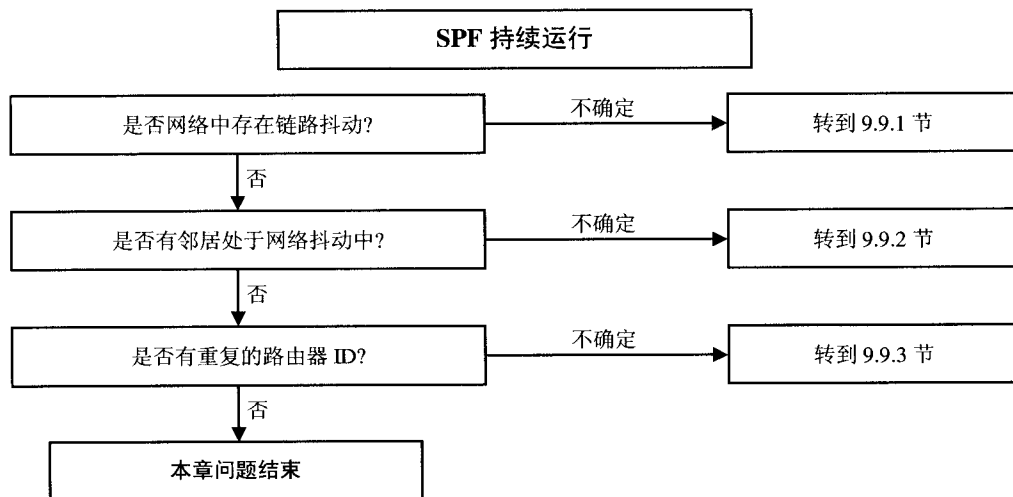


9.1.7 OSPF 中按需拨号路由 (DDR) 问题疑难解析





9.1.8 SPF 计算问题和路由抖动问题疑难解析



9.2 OSPF 邻居关系疑难解析

本节讨论与建立 OSPF 邻居关系相关的问题。OSPF 邻居关系问题可以是任何类型。有时，邻居表为空（也就是，OSPF 邻居甚至可能看不到彼此之间发送的 Hello 分组）。有时，这个问题是邻居停滞于一个特定的状态。回忆第 8 章“理解 OSPF 协议”知道 OSPF 邻居的通常状态是 FULL 状态。如果不是 FULL 状态而是其它状态并且持续很长一段时间，这就表

明存在问题。

以本节开头是因为这是使用 OSPF 协议最重要的一步。如果没有邻居关系建立起来或者邻居被停滞于一个不是 FULL 的状态中, OSPF 就不会在路由表中装入任何路由。因此, 在 OSPF 中非常重要是要确信邻居启动了。

OSPF 邻居关系问题可能是以下这些类型中的任何一种:

- OSPF 邻居表为空。
- OSPF 邻居停滞于 ATTEMPT 状态。
- OSPF 邻居停滞于 INIT 状态。
- OSPF 邻居停滞于 2-WAY 状态。
- OSPF 邻居停滞于 EXSTART / EXCHANGE 状态。
- OSPF 邻居停滞于 LOADING 状态。

处于这个列表中提到的任何一种状态并不表明出现问题, 但如果一个邻居停滞于这些状态中的一个并且持续很长时间, 这就成为一个问题而且必须纠正; 否则, OSPF 将不会正常工作。

9.2.1 问题: OSPF 邻居表为空

这是 OSPF 邻居关系中最常见的问题。最常见的原因不是与错误配置有关就是与缺少配置有关。如果邻居表为空, 将不能继续形成 OSPF 邻居关系。

最常见的引起这个问题的可能的原因如下:

- 在接口上没有启动 OSPF。
- 网络第 1 或第 2 层故障。
- 这个接口在 OSPF 中被定义为被动的。
- 一个访问列表在两边阻止 OSPF Hello 分组。
- 在一条广播链路上存在一对子网号 / 掩码不匹配。
- Hello 间隔 / 死亡间隔不匹配。
- 验证类型 (纯文本或者 MD5) 不匹配。
- 一个验证密钥不匹配。
- 一个区域 ID 不匹配
- 端 / 中继 / NSSA 区域选项不匹配。
- 一个 OSPF 邻接体存在有第二 IP 地址。
- 一个 OSPF 邻接体存在于一个异步接口上。
- 在 NBMA (帧中继、X.25、SMDS, 等等) 上没有定义网络类型或邻居。
- 在两边的 **Frame-relay map / dialer map** 语句中都缺少关键字 **broadcast**。

图 9-1 显示了两个路由器彼此之间运行 OSPF 协议。Show ip ospf neighbor 命令的输出显示了一个空表。在正常情况下, 这个输出显示 OSPF 邻居的状态。这个图应用于本节描述的大多数原因之中。

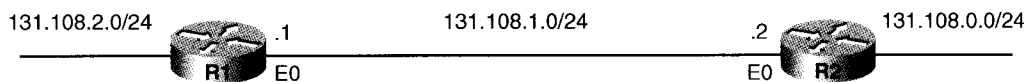


图 9-1 导致空 OSPF 邻居表问题的 OSPF 网络拓扑图

例 9-1 显示了 **show ip ospf neighbor** 命令的输出，其中出现了空邻居表。

例 9-1 **show ip ospf neighbor** 命令输出中有一个空邻居表

```
R2#show ip ospf neighbor
R2#
```

1. OSPF 邻居表为空——原因：没有在这个接口上启动 OSPF

可以基于每个接口启动 OSPF。为了在所有接口上启动 OSPF，在 **router ospf** 命令下使用一个 **network** 命令并且包含带有通配掩码（wildcard mask）的网络地址。当在 OSPF 中定义 **network** 语句时，应该小心地检查通配掩码以看清它所覆盖的地址范围。图 9-2 显示了为了解决基于这个原因的问题而遵循的流程图。

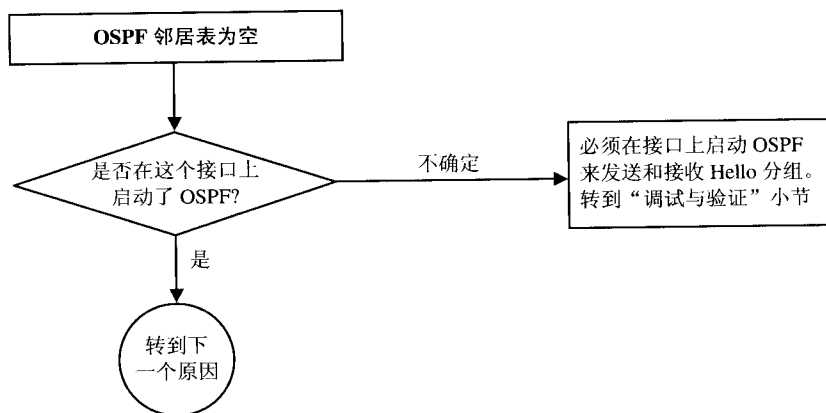


图 9-2 解决问题流程图

(1) 调试与验证

例 9-2 显示了路由器 R2 的配置。这个配置显示了在 **network** 语句下配置了错误的网络掩码使得仅仅把回环接口 0 包括进了区域 0 中。在 OSPF 中决定 **network** 语句的方式就与定义访问列表的方式一样。这里主要的意思是要包括一个区域中地址的范围。在例 9-2 中，带有通配掩码 0.0.0.255 的地址 131.108.0.0 的 **network** 语句将不会包括 131.108.1.2；它仅仅覆盖从 131.108.0.0 到 131.108.0.255 的范围，就像通配掩码所表示的一样。

例 9-2 R2 配置了错误的掩码

```
R2#
interface Loopback0
ip address 131.108.0.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router ospf 1
network 131.108.0.0 0.0.0.255 area 0
!
```

例 9-3 显示了路由器 R2 的配置。在 R2 的以太网接口上没有启动 OSPF。

例 9-3 在 R2 的以太网接口 Ethernet 0 上没有启动 OSPF

```
R2#show ip ospf interface Ethernet 0
Ethernet0 is up, line protocol is up
  OSPF not enabled on this interface
```

(2) 解决方法

有时，配置显示了正确的掩码而 OSPF 邻居表仍然显示为空。这是一个很少见的例子。在 OSPF 里网络配置更改过程中，对 OSPF 配置的剪切和粘贴都可能引发这个问题。因此，你总是应该注视对指定接口的命令 **show ip ospf interface** 的输出并注意是否在那个接口启动了 OSPF。这种类型的问题可以通过重新输入 **network** 语句来纠正。

如果没有在这个接口启动 OSPF，则接口就不能发送或接收 OSPF Hello 分组。为了纠正这个问题，改变网络掩码以使它包含以太网接口地址。

例 9-4 显示解决这个问题之后的新配置。在这个例子中，通配掩码是 0.0.255.255，这表示它覆盖了从 131.108.0.0 到 131.108.255.255 的地址范围。

例 9-4 改进 R2 上的配置来包括正确的网络掩码

```
R2#
router ospf 1
network 131.108.0.0 0.0.255.255 area 0
```

例 9-5 显示了在应用了正确的网络掩码之后命令 **show ip ospf neighbor** 的输出。

例 9-5 **show ip ospf neighbor** 命令的输出确认了在配置了正确的网络掩码之后 OSPF 启动了

```
R2#show ip ospf neighbor

Neighbor ID    Pri  State      Dead Time   Address        Interface
131.108.2.1    1    FULL/DR    00:00:38    131.108.1.1    Ethernet0
```

从 Cisco IOS 软件版本 12.0 开始，如果在接口上没有启动 OSPF，则命令 **show ip ospf interface** 的输出不显示任何东西。

2. OSPF 邻居表为空——原因：第 1/2 层故障

OSPF 在第 2 层之上的第 3 层中运行。如果发生了第 2 层故障，则 OSPF 就不能发送任何 Hello 分组了。OSPF 不能形成邻居关系的原因之一就是第 1 层或第 2 层故障。如果发生了第 1 层或第 2 层故障，那么这并不是一个与 OSPF 直接相关的问题。

图 9-3 显示了解决这个问题的流程图。

(1) 调试与验证

例 9-6 显示了在以太网接口 Ethernet 0 上运行命令 **show ip ospf interface** 的输出，其中显示了线路协议没有启动。

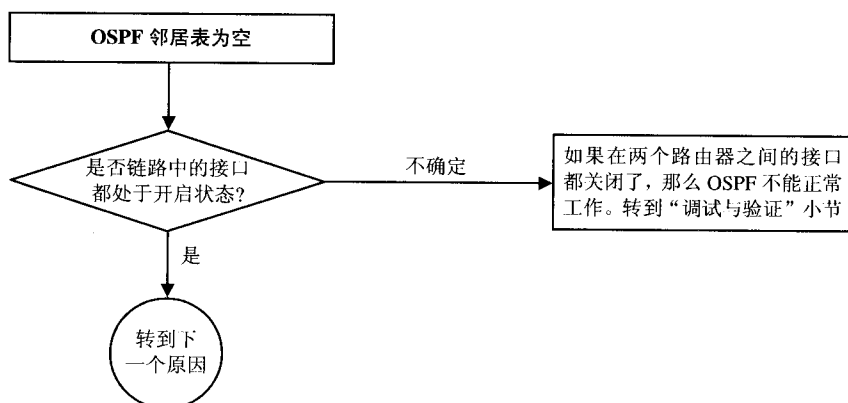


图 9-3 解决问题流程图

例 9-6 show ip ospf interface 命令输出表明线路协议没有启动

```

R2#show ip ospf interface Ethernet 0
Ethernet0 is up, line protocol is down
  Internet Address 131.108.1.2/24, Area 0
  Process ID 1, Router ID 131.108.1.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DOWN, Priority 1
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  
```

(2) 解决方法

第1层或第2层可能因为多个原因而出现故障。下面的列表包括了一些最常见的原因, 要对这些原因进行检查以决定接口是否关闭或者线路协议是否没有启动。

- 电缆没插入。
- 电缆松了。
- 电缆损坏了。
- 收发器损坏了。
- 端口损坏了。
- 接口卡损坏了。
- 电信的一条 WAN 链路的第2层出现了问题。
- 在背对背串行连接中缺少了 **clock** 语句。

为了解决这个问题, 首先通过检查上面提到的条件来解决第2层的问题。例 9-7 显示了解决了第2层问题后在接口 Ethernet 0 上运行 **show ip ospf interface** 命令的输出。

例 9-7 确认第2层完好

```

R2#show ip ospf interface Ethernet 0
Ethernet0 is up, line protocol is up
  Internet Address 131.108.1.2/24, Area 4
  Process ID 1, Router ID 131.108.1.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State BDR, Priority 1
  
```

(待续)

```

Designated Router (ID) 131.108.2.1, Interface address 131.108.1.1
Backup Designated router (ID) 131.108.1.2, Interface address 131.108.1.2
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:07
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 131.108.1.1 (Designated Router)
Suppress hello for 0 neighbor(s)

```

例 9-8 显示了 **show ip ospf neighbor** 的输出，其中显示 OSPF 邻接体状态为 FULL。

例 9-8 确认 OSPF 邻接体状态

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:39	131.108.1.1	Ethernet0

3. OSPF 邻居表为空——原因：接口在 OSPF 中被定义为被动的

当一个接口在路由器上的 OSPF 协议中被定义为被动的时候，它禁止 OSPF Hello 分组。这表示 OSPF 在这样的接口上不发送或接收任何 Hello 分组。因此，没有邻接体关系可以形成。图 9-4 显示了解决这个问题的一个流程图。

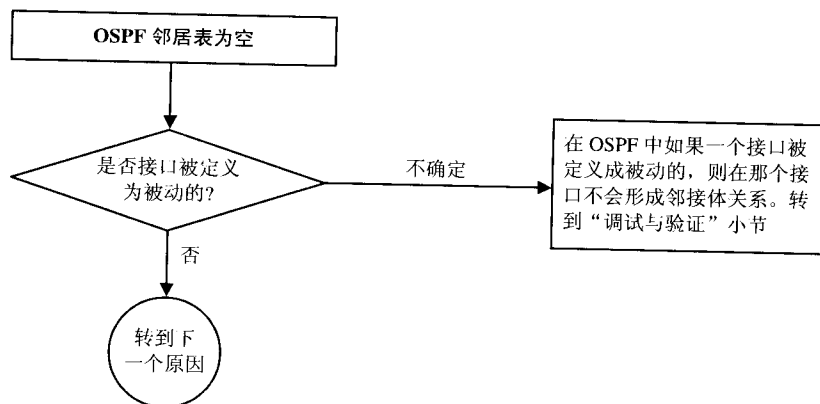


图 9-4 解决问题流程图

(1) 调试与验证

例 9-9 显示了在路由器 R2 的接口 Ethernet 0 上运行 **show ip ospf interface** 命令的输出。命令显示这个接口被定义成被动。

例 9-9 确定一个接口是否被定义成被动

```

R2#show ip ospf interface Ethernet 0
Ethernet0 is up, line protocol is up
Internet Address 131.108.1.2/24, Area 0
Process ID 1, Router ID 131.108.1.2, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State DR, Priority 1

```

(待续)

```

Designated Router (ID) 131.108.1.2, Interface address 131.108.1.2
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
No Hellos (Passive interface)
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)

```

例 9-10 显示了路由器 R2 的配置。这个配置显示 R2 的接口 Ethernet 0 被定义为被动。

例 9-10 R2 的接口 Ethernet 0 被定义为被动

```

R2#
interface Loopback0
ip address 131.108.0.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router ospf 1
  passive-interface Ethernet0
network 131.108.0.0 0.0.255.255 area 0

```

(2) 解决方法

为了解决这个问题，在 OSPF 的配置中去掉 **passive-interface** 命令。有时，有意键入这个命令以使路由器不能参与到那一个网段中任何 OSPF 过程中。当你不想在一个接口上建立任何邻居关系但是又想通告那个接口时也会键入这个命令。

有时，目的是在那个接口上不发送任何路由但是要接收所有的路由，就如在 RIP 和 IGRP 中所做的。记住，在 RIP 或 IGRP 中定义一个被动接口与在 OSPF 或 EIGRP 中定义一个被动接口有着不同的含义。当 RIP 或 IGRP 被定义为被动的，RIP 和 IGRP 将不在那个接口上发送任何路由更新分组，但是将在那个接口接收所有的路由更新分组。在 OSPF 中，一个被动的接口表示“不在这个接口发送或接收 OSPF Hello 分组”。因此，要想在一个接口上阻止路由器发送任何路由但是接收所有路由而将那个接口配置为被动的，这种做法是错误的。

例 9-11 显示了路由器 R2 的新的配置，命令 **passive-interface** 在配置中被去掉了。

例 9-11 在一个路由器接口配置中去掉被动接口定义

```

router ospf 1
no passive-interface Ethernet0
network 131.108.0.0 0.0.255.255 area 0

```

例 9-12 显示了在去掉 **passive-interface** 命令后 OSPF 正在形成邻接体关系。

例 9-12 确认新接口定义解决了问题

```

R2#show ip ospf neighbor

```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:37	131.108.1.1	Ethernet0

(待续)

4. OSPF 邻居表为空——原因：访问列表在两边阻止 OSPF Hello 分组

OSPF 以组播地址 224.0.0.5 发送它的 Hello 分组。所有启动了 OSPF 的接口监听这个地址。为了安全估量在接口级实现一个访问列表是很常见的事情。确信在这一情况下要在访问列表中置入 OSPF 组播 Hello 分组地址允许其运行；否则，访问列表将不自觉地阻止 OSPF 组播地址并阻止 OSPF 在那个接口形成邻居关系。

这一情形仅仅在当访问列表在两边的路由器都阻止 Hello 分组的时候发生。如果仅仅是一边阻止 OSPF Hello 分组，命令 **show ip ospf neighbor** 的输出将表明邻居停滞于 INIT 状态。这种情况在本章稍后讨论。

图 9-5 显示了解决这个问题所遵循的流程图。

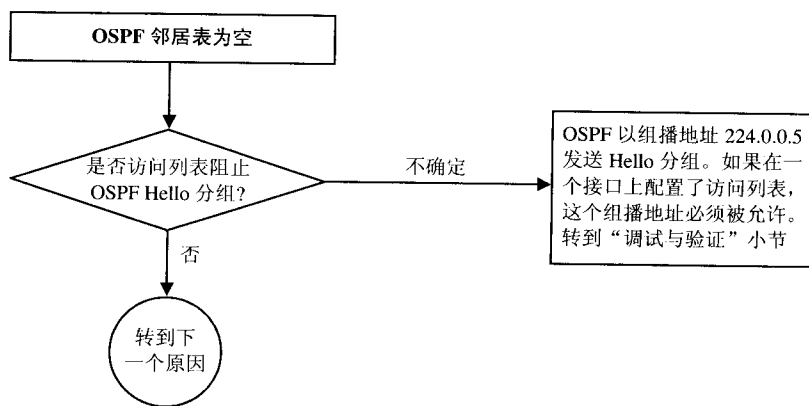


图 9-5 解决问题流程图

(1) 调试与验证

例 9-13 显示了路由器 R1 和 R2 的配置，它显示访问列表只允许 TCP 和 UDP 流量进入。入访问列表仅仅检查那个接口上进入的流量。因为有一个明确的 **deny** 在每个访问列表的尾部，所以这个访问列表将阻止 OSPF 组播地址 224.0.0.5。例 9-13 中的访问列表 101 仅被定义用来进行调试的目的。这个访问列表检查源地址为从 131.108.1.0 到 131.108.1.255 范围的发往 OSPF 组播地址 224.0.0.5 的 IP 分组。

例 9-13 R1 和 R2 的访问列表配置

```

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
ip access-group 100 in
!
access-list 100 permit tcp any any
access-list 100 permit udp any any

access-list 101 permit ip 131.108.1.0 0.0.0.255 host 224.0.0.5

R2#
interface Ethernet0
  
```

(待续)

```

ip address 131.108.1.2 255.255.255.0
ip access-group 100 in
!
access-list 100 permit tcp any any
access-list 100 permit udp any any

access-list 101 permit ip 131.108.1.0 0.0.0.255 host 224.0.0.5

```

例 9-14 显示了命令 **debug ip packet 101 detail** 的输出。这个调试仅仅在以太网段捕捉 OSPF Hello 分组。调试显示从路由器 R1 发出的 OSPF Hello 分组被 R2 拒绝了。

例 9-14 debug 显示 OSPF 组播分组被拒绝了

```

R2#debug ip packet 101 detail
IP packet debugging is on (detailed) for access list 101
IP: s=131.108.1.2 (Ethernet0), d=224.0.0.5, len 68, access denied, proto=89

```

(2) 解决方法

为了解决这个问题，必须重新配置访问列表以允许 OSPF 组播 Hello 分组。例 9-15 显示了解决这一问题的配置。在这个配置中，OSPF 组播 Hello 分组被允许。

例 9-15 配置访问列表来允许 OSPF 组播地址

```

interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip access-group 100 in
!
access-list 100 permit tcp any any
access-list 100 permit udp any any
access-list 100 permit ip any host 224.0.0.5

```

类似地，在另一边改变访问列表，确信 OSPF Hello 分组在访问列表中被允许。例 9-16 显示解决了这个问题之后 OSPF 邻居处于 FULL 状态。

例 9-16 确认重新配置访问列表解决了问题

```

R2#show ip ospf neighbor

```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:37	131.108.1.1	Ethernet0

5. OSPF 邻居为空——原因：在广播链路上存在子网号/掩码不匹配

OSPF 在除了点到点和虚链路之外的所有介质上执行子网号和掩码的检查，在 OSPF RFC 2328 的 10.5 节中有详细描述。为这种情形的目的，介质是以太网并且以太网上的网络类型是广播类型。网络掩码在 Hello 分组中得到通告。在无编号的点到点链路和虚链路的情形中，网络掩码字段为 0.0.0.0。如果子网掩码在以太网链路上各不相同，OSPF 将不会在那条链路上形成一个邻居关系。

(待续)

图 9-6 显示了解决这个问题应遵循的流程图。

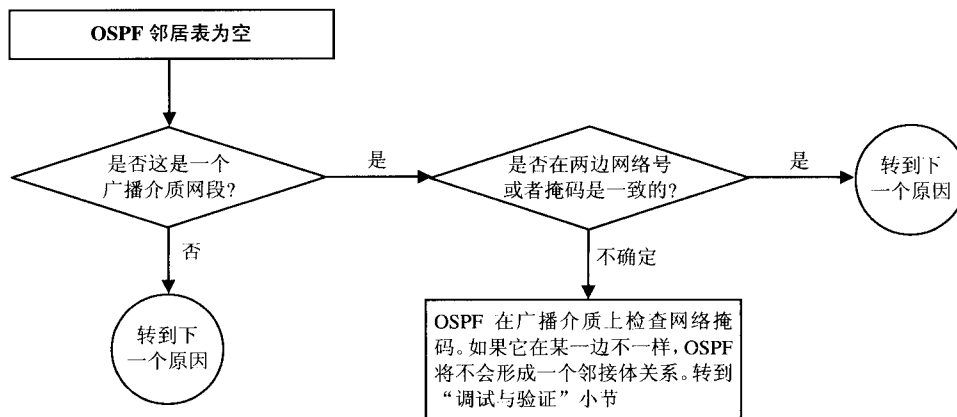


图 9-6 解决问题流程图

(1) 调试与验证

例 9-17 显示了命令 **debug ip ospf adj** 的输出。这个调试显示有一个不匹配的 Hello 参数。邻居子网掩码为 255.255.255.252 而路由器 R2 的子网掩码为 255.255.255.0。

例 9-17 命令 debug ip ospf adj 的输出表明有一个不匹配的 Hello 参数

```

R2##debug ip ospf adj
OSPF adjacency events debugging is on
R2#
OSPF: Rcv hello from 131.108.2.1 area 4 from Ethernet0 131.108.1.1
OSPF: Mismatched hello parameters from 131.108.2.1
Dead R 40 C 40, Hello R 10 C 10 Mask R 255.255.255.248 C 255.255.255.0
  
```

字母 R 表示“邻居配置”而字母 C 表示“这个路由器的配置”。在不同子网号的情况下调试信息将会为：

```
OSPF: Rcv pkt from 131.108.1.1, Ethernet0, area 0.0.0.1 : src not on the same network
```

例 9-18 显示了路由器 R1 和 R2 的配置，它显示了两边路由器的以太网有着不同的子网掩码。

例 9-18 有着不同子网掩码的 R1 和 R2 的配置

```

R2#
interface Ethernet0
ip address 131.108.1.2 255.255.255.0

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.248.0
  
```

(2) 解决方法

要解决这个问题，可以改变邻居（R1）的子网掩码以匹配路由器 R2 的子网掩码，或者

改变 R2 的子网掩码以匹配邻居的子网掩码。假设这里改变 R1 的子网掩码为 255.255.255.0 来匹配 R2。

例 9-19 显示修正了子网网络 / 掩码之后, 邻接体状态为 FULL。

例 9-19 确认 R1 和 R2 的子网掩码现在匹配了

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:31	131.108.1.1	Ethernet0

6. OSPF 邻居表为空——原因: Hello 间隔/死亡间隔不匹配

OSPF 邻居周期性地交换 Hello 分组来形成和维持邻居关系。OSPF 在 Hello 分组中通告路由器的 Hello 间隔和死亡间隔。这些间隔必须与邻居的匹配; 否则, 将不会形成邻接体关系。

图 9-7 显示了解决这个问题应遵循的流程图。

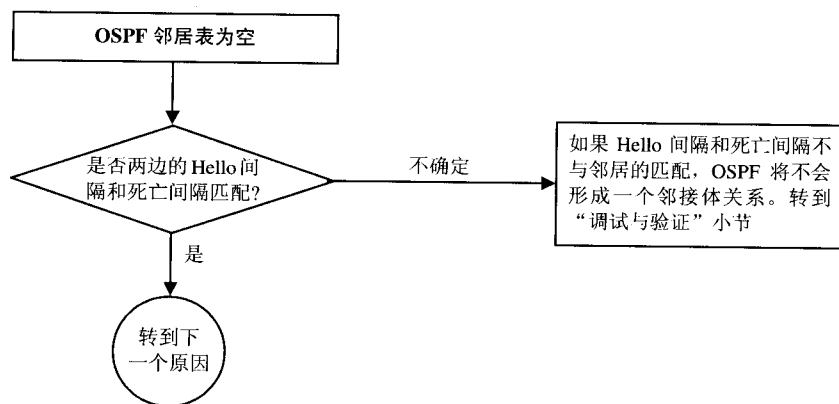


图 9-7 解决问题流程图

(1) 调试与验证

例 9-20 显示了 `debug ip ospf adj` 的输出, 其中表示邻居的 Hello 间隔与路由器 R2 的不匹配。

例 9-20 确认 OSPF 邻居间 Hello 间隔匹配

```
R2#debug ip ospf adj
```

```
OSPF adjacency events debugging is on
```

```
R2#
```

```
OSPF: Rcv hello from 131.108.2.1 area 4 from Ethernet0 131.108.1.1
```

```
OSPF: Mismatched hello parameters from 131.108.2.1
```

```
Dead R 40 C 40, Hello R 15 C 10 Mask R 255.255.255.0 C 255.255.255.0
```

例 9-21 显示了路由器 R1 和 R2 的配置。在 R1 中, Hello 间隔被配置成 15 秒。在 R2 中, Hello 间隔缺省为 10 秒。

例 9-21 R1 和 R2 的 Hello 间隔配置

```

R2#
interface Ethernet0

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.248.0
ip ospf hello-interval 15

```

(2) 解决方法

这个例子显示了为多个 OSPF 邻居配置的 Hello 间隔不匹配时出现的一个问题。当 OSPF 邻居间死亡间隔不匹配的时候会发生同样的问题。在这两种情形中，解决方法是改变 OSPF 邻居间的 Hello 间隔 / 死亡间隔使得它们一致。除非有一些特殊的原因使其偏离缺省设置，否则应该保持 Hello 间隔和死亡间隔的缺省值。

在例 9-22 中，在 R1 上的配置被改变以使得它使用以太网上 Hello 间隔的缺省值，即 10 秒。去掉 Hello 间隔值将 Hello 间隔值改变回到它的缺省值。

例 9-22 改变 Hello 间隔为其缺省值

```

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.248.0
no ip ospf hello-interval 15

```

例 9-23 显示在修改了 Hello 间隔和死亡间隔之后，OSPF 形成了一个邻接体关系。

例 9-23 确认在 Hello 间隔和死亡间隔匹配了之后，OSPF 形成了邻居关系

```

R2#show ip ospf neighbor

Neighbor ID    Pri   State           Dead Time   Address        Interface
131.108.2.1    1     FULL/DR         00:00:32   131.108.1.1   Ethernet0

```

7. OSPF 邻居表为空——原因：验证类型不匹配

OSPF 使用两种类型的验证，即纯文本（类型 1）和 MD5（类型 2）。类型 0 叫做无效验证（null authentication）。如果在某一边使用了纯文本验证类型，则另一边也必须使用纯文本验证。除非两边就使用同一种类型验证达成一致，否则 OSPF 将不会形成一个邻接体关系。

在一种情况下，某一边配置了纯文本或 MD5 验证但是另一边没有配置任何类型的验证。这种情况导致了一个 OSPF 邻居停滞于 INIT 状态的情形，在本章后面讨论这个问题。

图 9-8 显示了解决这个问题应遵循的流程图。

(1) 调试与验证

例 9-24 显示了 `debug ip ospf adj` 的输出，表明 R2 的邻居配置了 MD5 验证而 R2 配置了纯文本验证。

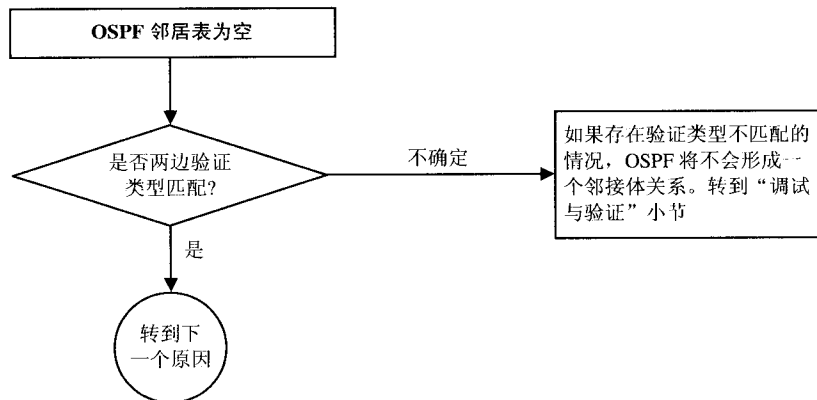


图 9-8 解决问题流程图

例 9-24 调试显示了不匹配的验证类型

```

R2#debug ip ospf adj
OSPF adjacency events debugging is on
R2#
OSPF: Rcv pkt from 131.108.1.1, Ethernet0 : Mismatch Authentication type. Input packet
specified type 2, we use type 1
  
```

例 9-25 显示了路由器 R1 和 R2 的配置, 表明 R2 使用纯文本验证而 R1 使用 MD5 验证。

例 9-25 R1 和 R2 的验证类型配置

```

R2#
router ospf 1
 area 0 authentication
 network 131.108.0.0 0.0.255.255 area 0

R1#
router ospf 1
 area 0 authentication message-digest
 network 131.108.0.0 0.0.255.255 area 0
  
```

(2) 解决方法

要解决这个问题, 需确信两边使用同样的验证类型。使 9-26 显示使用了一致的验证类型后, OSPF 形成了邻接体关系, 就如 FULL 状态表示的。

例 9-26 确认在 OSPF 邻居间的验证类型现在一致了

```

R2#show ip ospf neighbor

Neighbor ID    Pri   State   Dead Time   Address        Interface
131.108.2.1    1     FULL/DR 00:00:32    131.108.1.1    Ethernet0
  
```

8. OSPF 邻居表为空——原因: 验证密钥不匹配

当可以使用验证时, 必须在接口上配置验证密钥。以前验证是基于每区域被支持的, 但

在使用了 RFC 2328 中的规范之后, 验证现在基于每端口被支持了。这一特性已在 Cisco IOS 软件版本 12.0.8 及更高版本中实现。

如果在一边使用了验证而另一边没有使用, OSPF 会对不匹配的验证类型报错。有时, 验证密钥在两边都正确配置了, 但是命令 **debug ip ospf adj** 仍然显示出一个不匹配的验证类型。在这种情形中, **authentication-key** 必须被再次键入因为有这样一种可能性, 即在配置验证密钥时错误键入了一个空格。因为空格字符不显示在配置中, 所以这一部分很难确定。

另一种可能出错的事情是, 在 R1 这一边配置了纯文本密钥, 而在 R2 那边尽管验证类型是纯文本, 但却配置了 MD5 密钥。在这种情况下, MD5 密钥完全被 R2 忽略了因为没有在这个路由器上使用 MD5 验证类型。这与在接口上没有配置任何纯文本密钥是等价的。要获取更多关于验证的信息, 请参考第 8 章。

图 9-9 显示了解决这个问题应遵循的流程图。

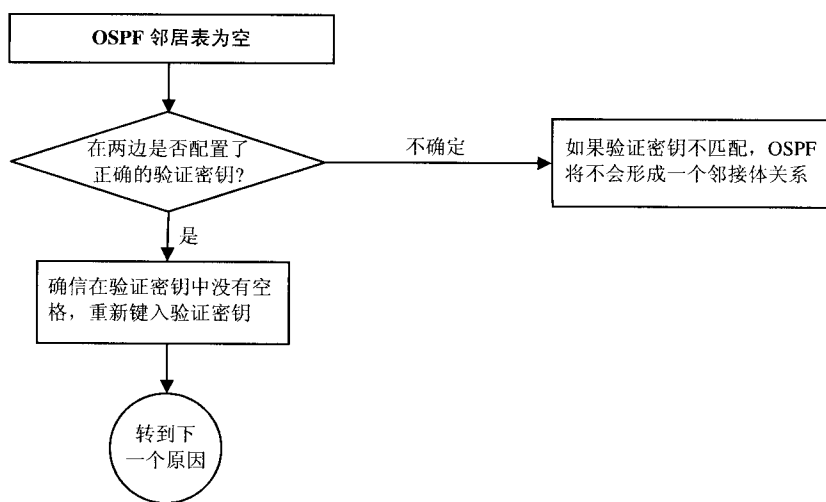


图 9-9 解决问题流程图

(1) 调试与验证

例 9-27 显示了 **debug ip ospf adj** 的输出, 其中表明存在一个验证密钥的不匹配。

例 9-27 检测到一个验证密钥不匹配

```

R2#debug ip ospf adj
OSPF adjacency events debugging is on
R2#
OSPF: Rcv pkt from 131.108.1.1, Ethernet0 : Mismatch Authentication Key - Clear Text
  
```

例 9-28 显示了 R1 和 R2 的配置。注意 R2 没有配置任何验证密钥, 然而 R1 配置了一个验证密钥因而导致了这个问题。

例 9-28 R1 和 R2 的配置

```

R2#
interface Ethernet0
  
```

(待续)

```
ip address 131.108.1.2 255.255.255.0
```

```
R1#
interface Ethernet0
ip address 131.108.1.1 255.255.248.0
ip ospf authentication-key Cisco
```

(2) 解决方法

要解决这个问题，需确信两边有同样类型的验证密钥。如果问题依然存在，重新键入验证密钥；存在这样一种可能，即在验证密钥的前面或后面加入了一个空格字符。

例 9-29 显示了改正这个问题后 **show ip ospf neighbor** 的输出。

例 9-29 确认在使用一致的验证密钥后 OSPF 邻居完好了

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:32	131.108.1.1	Ethernet0

9. OSPF 邻居表为空——原因：区域 ID 不匹配

OSPF 在 Hello 分组中发送区域信息。如果两边不能就它们是否是同一个普通区域的成员达成一致，那么就不会形成 OSPF 邻接体关系。区域信息是 OSPF 协议头的一部分。

图 9-10 显示了解决这个问题应遵循的流程图。

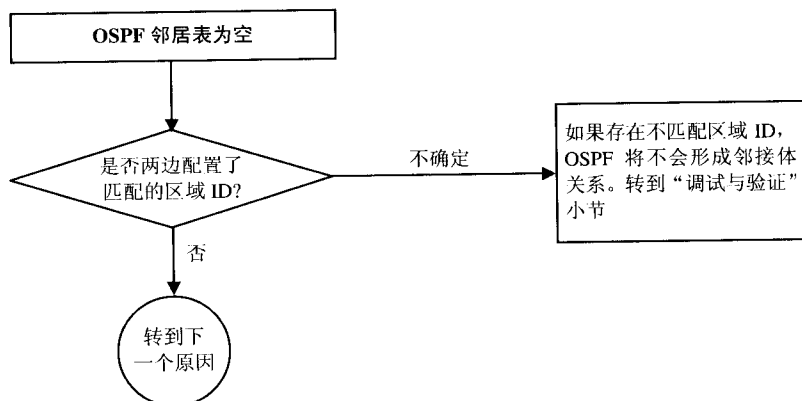


图 9-10 解决问题流程图

(1) 调试与验证

例 9-30 显示了 R2 和 R1 的配置。参考图 9-1，如果 R1 的以太网接口被包括在区域 0 中而 R2 的以太网接口被包括在区域 1 中，这就会引起区域 ID 不匹配。

例 9-30 R1 和 R2 上的区域配置

```
R2#
interface Ethernet0
```

(待续)

(待续)

```

ip address 131.108.1.2 255.255.255.0
!
router ospf 1
network 131.108.0.0 0.0.255.255 area 1

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router ospf 1
network 131.108.0.0 0.0.255.255 area 0

```

例 9-31 显示了 R1 上命令 **debug ip ospf adj** 的输出,表明 R1 从 R2 收到一个 OSPF 分组,这个 OSPF 分组头部含有一个区域 0.0.0.1。这证明另一边配置了区域 0.0.0.1 而不是区域 0。在这个情形中没有必要去检查另一边的配置。

例 9-31 确定 OSPF 邻居区域配置

```

R1#debug ip ospf adj
OSPF adjacency events debugging is on
R1#
OSPF: Rcv pkt from 131.108.1.2, Ethernet0, area 0.0.0.0
      mismatch area 0.0.0.1 in the header

```

例 9-32 显示了 R2 的控制台日志。这个日志显示 R1 收到了一个 OSPF 分组,其中 OSPF 头中有区域 0.0.0.0。因为这个路由器配置区域 0,所以它在控制台日志的等级接收这个消息。如果邻居路由器 R1 配置了其它区域,那么要找出区域不匹配的唯一方法就是打开 **debug ip ospf adj** 命令,如同在 R1 的情形中一样。

例 9-32 R2 的控制台日志显示了区域不匹配

```

R2#show log
%OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from backbone area must be
      virtual-link but not found from 131.108.1.1, Ethernet0

```

(2) 解决方法

要解决这个问题,需在链路上配置一样的区域。例 9-33 显示了 R1 的配置被改变了,因此 R1 的区域 ID 现在和 R2 的匹配了。

例 9-33 R1 上改正的配置

```

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router ospf 1
no network 131.108.0.0 0.0.255.255 area 0
network 131.108.0.0 0.0.255.255 area 1

```

例 9-34 显示了解决问题之后，OSPF 形成了一个邻接体关系。

例 9-34 确认在解决了区域 ID 不匹配问题之后 OSPF 的邻居关系

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:32	131.108.1.1	Ethernet0

10. OSPF 邻居表为空——原因：端/中继/NSSA 区域选项不匹配

当 OSPF 和一个邻居交换 Hello 分组的时候，在 Hello 分组中交换的一个东西就是一个可选的用 8 比特表示的特性。选项字段之一用于 E 比特，它是 OSPF 端区标记。当 E 比特设置为 0 时，与路由器相关联的区域为一个端区，因而在这个区域中不允许外部 LSA。

如果一边 E 比特设为 0 而另一边不是这样，则 OSPF 邻接体不会形成。这叫作一个可选特性不匹配（optional capability mismatch）。一边说它可以允许外部路由，而另一边说它不允许外部路由，因此 OSPF 邻居关系不能形成。

图 9-11 显示了解决这个问题应遵循的流程图。

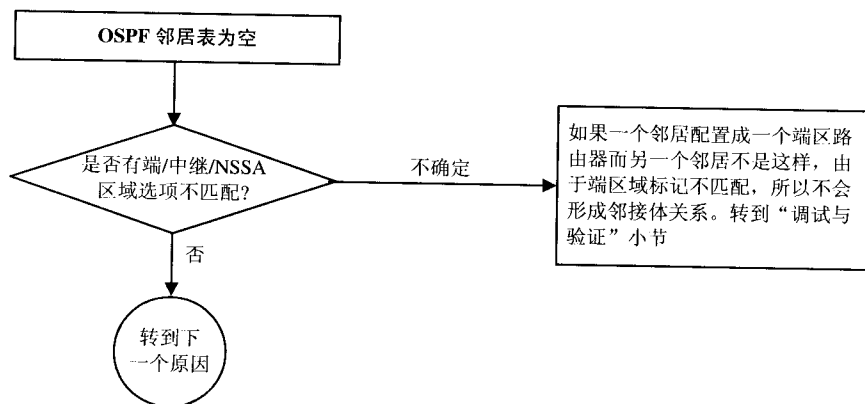


图 9-11 解决问题流程图

(1) 调试与验证

例 9-35 显示了路由器 R1 和 R2 的配置。R2 的配置显示区域 1 配置成一个端区，但是 R1 的区域 1 被配置成了一个标准区域。

例 9-35 R1 和 R2 的区域配置

```
R2#
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router ospf 1
area 1 stub
```

(待续)

```

network 131.108.0.0 0.0.255.255 area 1

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router ospf 1
network 131.108.0.0 0.0.255.255 area 1

```

例 9-36 显示了 R1 上 **debug ip ospf adj** 的输出。这个调试显示问题是一个端 / 中继区域选项不匹配。

例 9-36 debug ip ospf adj 命令输出确定一个端 / 中继区域选项比特不匹配

```

R1#debug ip ospf adj
OSPF adjacency events debugging is on
R1#
OSPF: Rcv hello from 131.108.0.1 area 1 from Ethernet0 131.108.1.2
OSPF: Hello from 131.108.1.2 with mismatched Stub/Transit area option bit

```

(2) 解决方法

要解决这个问题，需确信两边已经就一样的区域类型达成了一致。这个例子仅提及到了端区类型，但如果一边配置成端区而另一边配置成一个 OSPF NSSA 区域也会发生类似的问题。另一个情况是一边配置成 NSSA 区域而另一边配置成一个普通区域。在所有的情况下，不论何时只要存在一个区域类型不匹配，OSPF 邻接体将不会形成。

例 9-37 显示了一个 NSSA 区域不匹配的情形中 **debug ip ospf adj** 的输出。

例 9-37 debug ip ospf adj 命令输出确定一个 NSSA 类型比特不匹配

```

R1#debug ip ospf adj
OSPF adjacency events debugging is on
R1#
OSPF: Rcv hello from 131.108.0.1 area 1 from Ethernet0 131.108.1.2
OSPF: Hello from 131.108.1.2 with mismatched NSSA option bit

```

例 9-38 显示了解决这个问题后 R1 上配置的变化。现在 R1 也是端区的一部分了。

例 9-38 R1 上配置的改变解决了问题

```

R1#
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router ospf 1
area 1 stub
network 131.108.0.0 0.0.255.255 area 1

```

例 9-39 显示了解决这个问题后 **show ip ospf neighbor** 的输出。

例 9-39 确认在解决了端区 / NSSA 区域不匹配问题后 OSPF 邻居完好

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:32	131.108.1.1	Ethernet0

11. OSPF 邻居表为空——原因：在第二 IP 地址上的 OSPF 邻接体关系

这是一个很常见的问题，一个客户可能在一个局域网段有一个 C 类地址。当客户用完分配的地址空间后，他得到另一个 C 类地址并将这个新地址作为一个第二地址分配给同一个接口。什么都运行得很好直到两个路由器必须交换 OSPF Hello 分组 / 更新分组的时候，此时一个路由器分配了主 IP 地址但是另一边路由器使用了第二 IP 地址，如同在图 9-12 网络中的描述一样。这两个路由器通过一个第 2 层交换机相连。

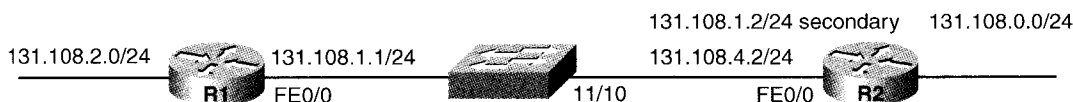


图 9-12 两个路由器通过一个交换机相连，一边的主接口 IP 地址与另一边的第二 IP 地址完全相同

图 9-13 显示了解决这个问题应遵循的流程图。

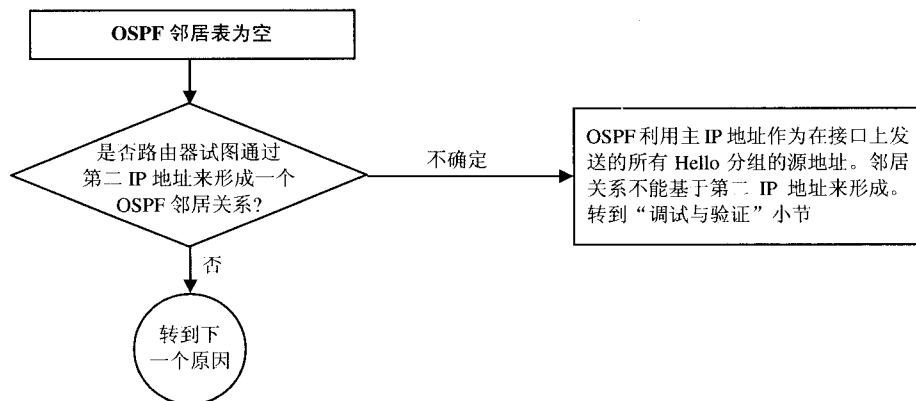


图 9-13 解决问题流程图

(1) 调试与验证

例 9-40 显示了 R1 和 R2 的配置。配置表明 R2 在它的以太网接口上配置了一个主地址和第二地址，并且在 R1 上用于主地址的子网号在 R2 上用于第二地址。

例 9-40 R2 的 Fast Ethernet0 / 0 接口第二地址配置匹配 R1 的 Ethernet 0 接口主地址配置

```
R2#
interface FastEthernet0/0
ip address 131.108.1.2 255.255.255.0 secondary
```

(待续)

```
ip address 131.108.4.2 255.255.255.0
```

```
R1#
```

```
interface Ethernet0
```

```
ip address 131.108.1.1 255.255.255.0
```

例 9-41 显示了 **debug ip ospf adj** 的输出。这个输出和在一个子网号不匹配的情形中的调试输出恰好是一样的。这是因为，当 R1 从 R2 收到一个 Hello 分组时，源地址将是 131.108.4.2，这个地址与它所连接的接口属于一个不同的子网。因此，R1 将会报错。

例 9-41 debug ip ospf adj 命令的输出表明有一个 IP 地址冲突

```
R2#debug ip ospf adj
```

```
OSPF adjacency events debugging is on
```

```
R2#
```

```
OSPF: Rcv pkt from 131.108.1.1, FastEthernet0/0, area 0.0.0.1 : src not on the same network
```

(2) 解决方法

对这类问题的解决方法是在 R1 上创建子接口。这种做法仅在带有第二地址的接口是快速以太网或吉比特以太网并且通过一个第 2 层交换机相连的情况下才可行。在第 2 层交换机是 Cisco 交换机的情况下，可以通过一条交换机间链路（ISL），或者如果是其它厂家的交换机的情况下，采用 dot1Q 封装来达到这一目的。ISL 或者 dot1Q 封装用于在两个分离的 VLAN 之间实现路由。与 R2 的快速以太网接口相连的交换机端口配置为主干端口（trunk port），因此所有在 VLAN1 和 VLAN2 之间的流量将会流经路由器 R2 并且路由器 R2 将在这两个 VLAN 之间执行路由。

例 9-42 显示 R2 的配置和使用了一个 ISL 主干的 Cisco 交换机，因而它可以在 R2 上创建子接口。

例 9-42 在 R2 上创建子接口

```
R2#
```

```
interface FastEthernet0/0
```

```
no ip address
```

```
full-duplex
```

```
!
```

```
interface FastEthernet0/0.1
```

```
encapsulation isl 2
```

```
ip address 131.108.1.2 255.255.255.0
```

```
interface FastEthernet0/0.2
```

```
encapsulation isl 1
```

```
ip address 131.108.4.2 255.255.255.0
```

```
cat-5k-1> (enable) set trunk 11/10 on
```

```
Port(s) 11/10 trunk mode set to on
```

例 9-42 显示了如何创建子接口和如何在 Cisco Catalyst 交换机上运行 VLAN 主干的例子。R1 的快速以太网接口包括在 VLAN2 中，并且 R2 的子接口 FE 0/0.1 也被加入到了 VLAN2 中。连接 R2 的交换机端口 11/10 也为主干启用了，因而它将携带 VLAN1 和 VLAN2 的流量。类似的配置也可在 802.1Q 封装中实现。图 9-14 显示了作出这一变化后的逻辑图。FE 0/0.1

是一个使用 ISL 封装的子接口。

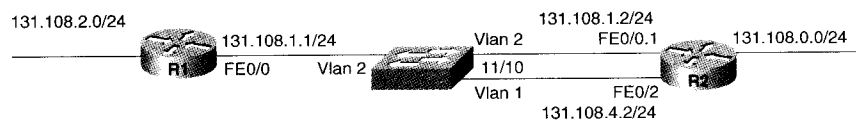


图 9-14 使用了 ISL 封装后的子接口逻辑图

在作出这个变化之后，R2 将在 FE 0/0.1 上与 R1 形成一个邻居关系。其它子接口，如 VLAN1 中的 FE 0/0.2，将与 VLAN1 中的其它路由器形成一个邻居关系。

FE 0/0.1 和 FE 0/0.2 是逻辑子接口。这表示这些子接口都是连接到交换机端口 11/10 的一个物理接口 (FE 0/0) 的子集。例 9-43 显示了作出这一变化后 show ip ospf neighbor 的输出。

例 9-43 在创建子接口后 OSPF 形成邻居关系

R2#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
131.108.2.1	1	FULL/DR	00:00:32	131.108.1.1	FastEthernet0/0.1

12. OSPF 邻居表为空——原因：OSPF 邻接体位于异步接口之上

当 OSPF 在两个路由器的异步接口之间运行时你必须启用异步缺省路由或者启动动态路由。当启用异步缺省路由 (async default routing) 时，路由器总是在一个异步接口发送分组。在交互式异步连接的情况下，使用者不得不键入 **ppp** 来建立 PPP 会话，这时可使用命令 **async dynamic routing**，但这时使用者必须键入 **ppp / routing** 来在异步接口启用路由。如果没有能力来做这件事情将会引起 OSPF 在异步链路上不形成任何邻接体关系。

图 9-15 显示了一个网络图，其中有两个路由器在它们的异步接口间运行 OSPF。



图 9-15 OSPF 在异步接口间运行

图 9-16 显示了解决这个问题应遵循的流程图。

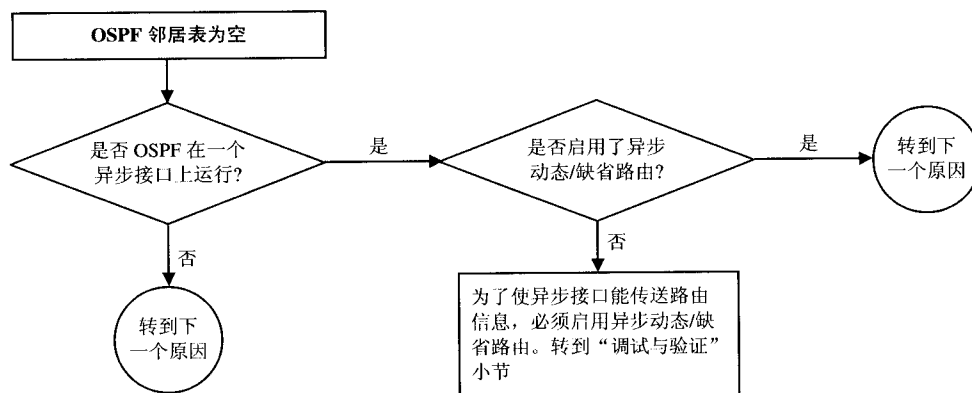


图 9-16 解决问题流程图

(1) 调试与验证

例 9-44 显示了 R1 和 R2 的配置。这个配置显示了在接口配置中缺少了异步缺省路由。

例 9-44 确认在 R1 和 R2 的异步接口中缺少异步缺省路由

```
R1#
interface Async1
  description ASYNC LINE TO R2
  ip address 131.108.1.1 255.255.255.0
  encapsulation ppp
  async mode dedicated
  dialer in-band
  dialer map ip 131.108.1.2 name Router2 broadcast
  dialer-group 1
  ppp authentication chap
```

```
R2#
interface Async1
  description ASYNC LINE TO R1
  ip address 131.108.1.2 255.255.255.0
  encapsulation ppp
  async mode dedicated
  dialer in-band
  dialer map ip 131.108.1.1 name Router2 broadcast
  dialer-group 1
  ppp authentication chap
```

(2) 解决方法

在这个例子中, 使用 **async default routing** 或者 **async dynamic routing** 来解决这个问题。

例 9-45 显示了使用 **async default routing** 之后 R1 和 R2 的配置。

例 9-45 配置 R1 和 R2 使用 **async default routing**

```
R1#
interface Async1
  description ASYNC LINE TO R2
  ip address 131.108.1.1 255.255.255.0
  encapsulation ppp
  async default routing
  async mode dedicated
  dialer in-band
  dialer map ip 131.108.1.2 name Router2 broadcast
  dialer-group 1
  ppp authentication chap
```

```
R2#
interface Async1
  description ASYNC LINE TO R1
  ip address 131.108.1.2 255.255.255.0
```

(待续)