

第 6 章

理解 EIGRP 协议

本章涵盖下列有关 EIGRP 协议的关键主题：

- 度量。
- EIGRP 邻居关系。
- 扩散更新算法 DUAL。
- DUAL 有限状态机。
- EIGRP 可靠传输协议。
- EIGRP 分组格式。
- EIGRP 行为。
- EIGRP 汇总。
- EIGRP 查询过程。
- 缺省路由和 EIGRP。
- EIGRP 中的非等价代价负载均衡。

随着网络规模的日益扩大，可以看到传统的距离向量路由协议，如 IGRP 和 RIP，不再能够满足网络的需要。IGRP 和 RIP 存在的一些最严重的可扩展性问题如下：

- 完全的定期路由更新消耗了带宽——RIP 每 30 秒将其整个路由表发送一次，而 IGRP 则每 90 秒将其整个路由表发送一次，这消耗了极大的带宽。
- RIP 跳数限制为 15——这个限制使得 RIP 协议在当今的网络中成为了一个不可扩展的协议，因为大多数中等规模的网络都包含 15 台以上的路由器。
- 不支持 VLSM 和非连续网络——这也阻碍了 RIP 和 IGRP 扩展自身以适应大型网络的能力。因为这个因素，路由汇总也不被支持。
- 收敛时间长——因为 RIP 和 IGRP 定期发送路由更新，所以如果一个网络的一部分变得不可用，则网

络的其他部分要花相当的时间才能知道这部分已经不能使用。

- 不是百分之百的无环路由——RIP 和 IGRP 不保存拓扑表，所以它们没有一种机制来保证其路由表能达到百分之百的无环。

由于 IGRP 和 RIP 的这些缺点，Cisco 开发出了 IGRP 的一种增强版本，它不仅解决了 IGRP 和 RIP 的所有这些问题，还是一种足以跟上当今网络规模扩大速度的路由协议。这一增强的版本就称为增强的内部网关路由协议（Enhanced Interior Gateway Protocol，EIGRP）。

EIGRP 既不是一种传统的距离向量路由协议，也不是一种链路状态协议——它是由这两类路由协议混合而成的一种协议。像距离向量协议那样，EIGRP 从它的相邻路由器那里得到更新信息；也像链路状态协议那样，IGRP 保存着一张被发送路由的拓扑表，并使用扩散更新机制（Difussing Update Algorithm，DUAL）选择一个无环路由的路径。网络的收敛时间就是网络上所有路由器都就网络变化达成一致所用的时间。收敛时间越短，路由器就能更快地适应网络拓扑结构的变化。不像传统的距离向量协议，EIGRP 有很快的收敛时间，而且不用发送完全的定期性路由更新；也不像链路状态协议，EIGRP 并不知道整个网络看上去是怎样的，只能靠相邻路由器公布的信息。因为 EIGRP 同时具有距离向量协议和链路状态协议的特性，所以 Cisco 将 EIGRP 定级为一种高级距离向量协议。

EIGRP 的优点包含如下一些方面：

- 100%的无环路由——如果整个网络都被包含在一个自治系统中，EIGRP 将保证提供一张百分之百无环路由转发表。
- 配置简单——EIGRP 的配置极其简单，在基本的层次上与 IGRP 和 RIP 相同。
- 快速收敛——EIGRP 的收敛时间比 RIP 和 IGRP 要快得多。
- 增量更新——在一个 EIGRP 网络中，除非网络发生变化，否则是不会交换路由更新的。同样，除非对路由表作了修改，否则也不会发送整个路由表。这样就节省了 CPU 的资源，非常高效。
- 组播地址的使用——IGRP 和 RIP 使用广播地址 255.255.255.255 来发送它们的分组。这意味着在同一网段上的每一台设备都会收到更新信息。而 EIGRP 在组播地址 224.0.0.10 上发送它的分组，这将保证只有启用了 EIGRP 的设备才会接收 EIGRP 分组。
- 更好地利用带宽——EIGRP 从 EIGRP 分组即将发出的接口上获得带宽参数。参数值是基于特定的接口指派的。例如，在缺省的情况下，所有串行端口都有 1544kbps 的带宽。不过，这个带宽参数是可配置的。EIGRP 最多可以使用 50%的接口带宽来承载 EIGRP 分组，这就保证了 EIGRP 分组不会在主要的网络收敛过程中“饿死”被路由的数据分组。RIP 和 IGRP 没有这一特性，所以大量的 RIP 和 IGRP 更新分组就可能阻止常规分组的通过。
- 支持 VLSM 和非连续网络——不像 RIP 和 IGRP，EIGRP 支持 VLSM 和非连续网络。这使得 EIGRP 可以在现代网络中实现，并使其自身具有了更好的网络可扩展性。

6.1 度 量

EIGRP 和 RIP 使用相同的公式计算它们的度量，然而，EIGRP 的度量可以通过把 IGRP

的度量乘

EIGR

而 IC

缺省

EIGR

公式

EIGR

在其更新

字段。这

某一个目的

不像

EIGRP 过

有路由器

如以太网、

接口上以及

当或低于

一次。

除了发

一个时间

个时间，路

经超过了保

味着在局域

网接口的 H

间都是可以

● 接

保

● 接

● 接

立

例 6-1

Route	
IP-EIGRP	
H	Ad
1	5.
0	19

的度量乘以 256 来获得。换句话说：

$$\text{EIGRP 度量} = \text{IGRP 度量} \times 256$$

而 IGRP 的度量计算如公式 6-1 所示。

缺省情况下，K1 和 K3 的 K 值为 0，因此，EIGRP 的度量可以简化如下：

$$\text{EIGRP 度量} = [(10^7 / \text{路径上的较低带宽}) + (\text{所有延迟的总和})] \times 256$$

公式 6-1 IGRP 的度量计算

EIGRP 与 IGRP 的度量的不同之处是一个因子 256，这是因为 Metric 字段的原因。IGRP 在其更新分组中只使用 24 位的 Metric 字段，而 EIGRP 在其更新分组中使用 32 位的 Metric 字段。这相差的 8 位就需要 IGRP 度量乘以 256 才能得到 EIGRP 度量。例如，如果 IGRP 到某一个目的网络的度量是 8586，则 EIGRP 的度量是 $8586 \times 256 = 2,198,016$ 。

6.2 EIGRP 的邻居关系

不像 IGRP，EIGRP 必须在发送更新信息之前建立邻居关系。当在路由器上配置了一个 EIGRP 过程后，该路由器就开始在组播地址 224.0.0.10 上交换 EIGRP 的 Hello 分组。当所有路由器都收到了彼此的 Hello 分组后，它们之间形成邻居关系。在局域网的广播介质上，如以太网、令牌环或 FDDI，Hello 分组每隔 5 秒发送一次。在广域网中带宽高于 T1 的多点接口上以及点对点的子接口上，Hello 也是每隔 5 秒被发送一次。而广域网中带宽与 T1 相当或低于 T1 的多点接口被认为是低带宽接口，在这些接口上 Hello 分组每隔 60 秒被发送一次。

除了发送 Hello 的时间，还有一个概念叫做保持时间 (hold time)。保持时间告诉路由器一个时间长度，这个时间是当 Hello 分组没有被收到时，路由器将等待的最长时间，过了这个时间，路由器将重置某个相邻路由器。换句话说，如果在接收到一个 Hello 分组之前就已经超过了保持时间的话，邻居关系就会被重置。缺省的保持时间是 Hello 时间的三倍。这意味着在局域网广播介质中，如果 Hello 时间是 5 秒，那么保持时间就是 15 秒，而慢速的广域网接口的 Hello 时间是 60 秒，因而缺省的保持时间是 180 秒。要记住，Hello 时间和保持时间都是可以配置的。在 EIGRP 路由器着手建立邻居关系之前，还必须满足一些条件：

- 接收方路由器将 Hello 分组的源地址与收到该分组的接口的 IP 地址进行比较，以确保它们属于同一子网。
- 接收方路由器将源路由器的 K 常数值与自己的相比较，以确保它们是匹配的。
- 接收方路由器必须与源路由器相同的自治系统内。例 6-1 显示了在邻居关系被建立之后 `show ip eigrp neighbor` 命令的输出。

例 6-1 `show ip eigrp neighbor` 命令的输出

```
Router_1#show ip eigrp neighbor
IP-EIGRP neighbors for process 1
H  Address      Interface    Hold Uptime   SRTT   RTO    Q    Seq
                          (sec)      (ms)  Cnt  Num
1  5.5.5.4       Et0          11 00:00:22   1    4500   0    3
0  192.168.9.5   Et1          10 00:00:23  372   2232   0    2
```

输出表格的表头解释如下:

- **H**——相邻路由器的清单, 以它们被学习的先后顺序排列。
- **Address**——相邻路由器的 IP 地址。
- **Interface**——相邻路由器通过它学习的接口。
- **Hold**——为相邻路由器设置的保持时间计时器。如果计时器归 0, 邻居关系就断开。
- **Uptime**——跟踪相邻路由器建立时间长度的计时器。
- 平滑路径往返时间 (**Smooth Round Trip Time, SRTT**)——发送和接收某一可靠 EIGRP 分组的平均时间。
- 往返超时时间 (**Round Trip Timeout, RTO**)——如果未收到某个可靠分组的确认信息, 路由器在重传该分组前将要等待的时间。
- **Q Count**——等待发送到相邻路由器的 EIGRP 分组数量。
- **序列号**——从相邻路由器那里接收到的最近一个可靠的 EIGRP 分组的序列号。这个序列号是为了确保从相邻路由器收到的分组按顺序排列。

6.3 扩散更新算法

DUAL 机制是 EIGRP 操作的核心。这一机制跟踪从某一相邻路由器公布的所有路由, 并选择一条到达目的地的无环路径。在详细讨论 DUAL 之前, 必须弄清几个术语和概念:

- **可行距离 (feasible distance, FD)**——指沿着到达目的地的路径的最小度量。图 6-1 显示了从路由器 A 的角度看, 它的所有相邻路由器到达网络 7 的可行距离的计算。

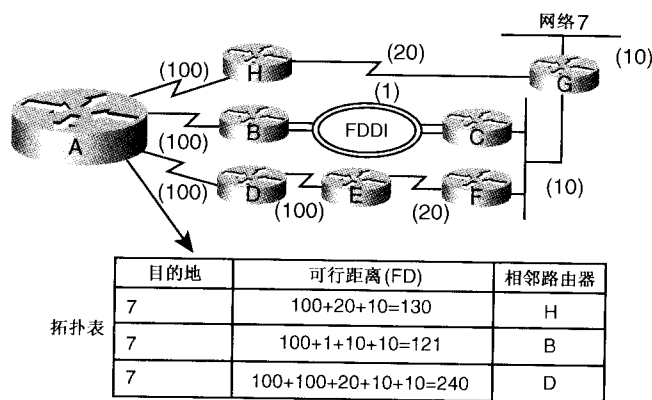


图 6-1 可行距离的计算

- **报告距离 (reported distance, RD)**——有时也称为公布距离, 它是就目的地而言的度量, 由上游相邻节点公布。换句话说, 报告距离是相邻路由器到目的地的度量。图 6-2 显示了路由器 A 的相邻路由器到网络 7 的报告距离的计算。
- **可行性条件 (Feasibility condition, FC)**——是一种报告距离 (RD) 小于可行距离 (FD) 的条件。换句话说, 当相邻路由器到达目的地的度量比本地路由器的度量小的时候, 就满足可行性条件的要求。这个条件很重要, 可以保证一条路径是无环的。

- **EIGRP 后继**——后继是满足可行性条件 (FC) 并具有到目的地的最低度量的一个相邻路由器。后继被用作下一跳来转发分组到目的网络。

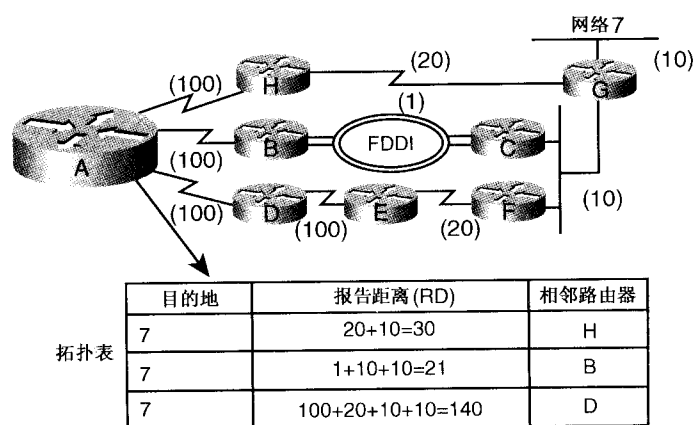


图 6-2 报告距离的计算

- **可行后继**——指满足可行性条件 (FC) 但没有被选作后继的一个相邻路由器。当主路由丢失时，可行后继可以被认为是一个潜在的备份路由。

图 6-3 说明了后继和可行后继的概念。

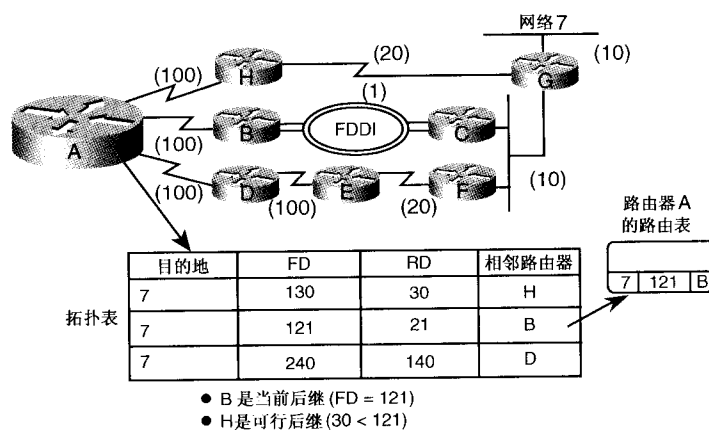


图 6-3 后继和可行后继的图例

路由器 B 被选作后继，因为在路由器 A 的所有相邻路由器当中，路由器 B 有到网络 7 的最低可行距离（度量为 121）。要选择一个可行后继，路由器 A 要看它的相邻路由器中哪一个的报告距离 (RD) 比它后继的可行距离小。在这种情况下，路由器 H 的报告距离是 30，因此路由器 H 被选作可行后继。路由器 D 既不是后继也不是可行后继，因为它的报告距离是 140，比 121 要大，这样就不能满足可行性条件。

- **被动路由**——EIGRP 中的被动路由表明路由器有一个到目的地的合法后继并且 EIGRP 工作正常。
- **主动路由**——EIGRP 中的主动路由表明路由器已经丢失了它的后继，它没有任何可用的可行后继，并且当前该路由器正主动地搜索替代的路由以实现收敛。

6.4 DUAL 有限状态机

当 EIGRP 失去它的后继或主路由时，它就立即查看拓扑表以确定是否有可用的可行后继，试图重新收敛。如果有一个可用的可行后继，EIGRP 就立即将其提升为后继，并将这一变化通知它的相邻路由器，而这个可行后继就变成了 EIGRP 转发分组到目的地的下一跳。EIGRP 进行本地收敛的过程称作本地计算，这个过程中不涉及其他路由器。这同样也节省了 CPU 资源，因为所有的可行后继在主路由失败之前都已经被选择了（参看图 6-3）。如果主路由（路由器 D）因为某种原因变得不可用，预先选择的可行后继路由器 H 就立即取而代之成为主路由。

现在，如果主路由失效且没有可用的可行后继，该路由器就进入扩散计算状态。在扩散计算状态中，路由器向其所有的相邻路由器发送查询分组，寻找丢失的路由，且该路由器将进入活动状态。如果相邻的路由器有关于被丢失路由的信息，它们就会回复进行查询的路由器。如果相邻的路由器没有关于被丢失路由的信息，它们就会向所有相邻路由器发送查询。如果相邻路由器没有可替换的路由，也没有相邻的其他路由器，它就向请求路由器发回一个回复分组，并把度量设置为无穷大，这也表示没有可替换的路由。请求路由器等待从相邻路由器那里发回的所有回复，然后选择其回复分组中包含的度量最低的相邻路由器作为发送分组的下一跳。

参考图 6-3，如果主后继路由器 B 不可用，并且其可行后继路由器 H 也不可用，路由器 A 就向路由器 D 发送一个查询，寻找网络 7。在这种情况下，路由器 D 只简单地以一个到网络 7 的合法度量回复这次查询，然后路由器 A 就会收敛，使用路由器 D 作为到网络 7 的下一跳。

总结 DUAL 的操作，它首先选择一个后继作为主路径，同时基于可行性条件选择一个可行后继作为它的备份路径。如果后继变为不可用，可行后继就被用作主路由。如果没有给出可行后继，路由器就查询所有相邻路由器，并基于查询的回复计算一个新的后继。因此，在一个 EIGRP 网络中，查询机制是获得快速收敛的唯一途径。

由 Jeff Doyle 编写的 Cisco 版《TCP/IP 路由技术》（第一卷）第 8 章提供了一个关于 EIGRP DUAL 机制的精彩的详细描述。

6.5 EIGRP 的可靠传输协议

存在五种 EIGRP 分组，可进一步分为可靠分组和不可靠分组。可靠的 EIGRP 分组如下：

- **更新**——更新分组包括发往某个相邻的 EIGRP 路由器的 EIGRP 路由更新。
- **查询**——当某个路由不可用且路由器需要询问路由状态以便快速收敛时，查询分组就被发往相邻的路由器。
- **回复**——查询的回复分组包含被查询路由的状态。

不可靠的 EIGRP 分组如下：

- **Hello**——Hello 分组用于通过链路建立 EIGRP 的邻居关系。
- **确认**——确认分组确保 EIGRP 分组的可靠传递。

所有 EIGRP 分组都是自动的，所以 EIGRP 可靠传输协议表，当一个路由器发回一个确认分组保留发送，除非收到确认分组重发多次。在一个组如先前所讨论的，分组进行了确认，EIGRP 分组例如，如就不会再在该组在假设路由器那么路由器 1 由器 3 上。EIGRP 发给没有对前组的相邻节点，路由器重发该分组就会重置邻居关系。

图 6-4 显示三元组。TLV 参数 TLV、IP 内

所有 EIGRP 分组都通过 EIGRP 组播地址 224.0.0.10 发送, 而每台启用了 EIGRP 的设备都自动的侦听地址 224.0.0.10。因为这是一个组播地址且很多设备会同时收到 EIGRP 分组, 所以 EIGRP 需要自己的传输协议来确保 EIGRP 分组传递的可靠性, 这个协议就是 EIGRP 可靠传输协议 (Reliable Transport Protocol, RTP)。路由器为每一台相邻路由器保存一张发送列表, 当一个可靠的 EIGRP 分组被发送到某个相邻路由器上时, 发送路由器就期望该相邻路由器发回一个确认分组, 说明可靠的 EIGRP 分组已经被接收。EIGRP RTP 为每一个未确认的分组保留发送窗口, 因此, 在下一个可靠的分组被发出前, 前一个可靠分组都必须得到确认。除非收到确认, 否则路由器会重发未确认的分组。如果没有收到确认, EIGRP RTP 会将同一分组重发多达 16 次。如果在 16 次重发后仍然没有收到确认, EIGRP 就将重置邻居关系。

在一个组播局域网中, 发送某一组播更新时如果发送窗口大小为 1, 就会造成问题。正如先前所讨论的那样, 对于可靠的组播流置, 下一组播分组必须在所有接收端都对前一组播分组进行了确认后才能被发送。如果一个组播局域网中有一个或多个相邻的 EIGRP 路由器确认 EIGRP 分组很慢或者失败, 那么其他所有相邻路由器都会受到影响。

例如, 如果在一个以太网段上有三台路由器, 路由器 1 发送一个组播 EIGRP 更新, 它就不会再在该以太网上发送另外的组播 EIGRP 分组, 直到收到了其他两台路由器的确认。现在假设路由器 2 成功地向路由器 1 发送了确认, 而路由器 3 在发送确认时发生了某种问题。那么路由器 1 就会停止发送任何 EIGRP 分组, 而路由器 2 也因此受到影响, 尽管问题出在路由器 3 上。EIGRP RTP 会避免这一问题的出现, 它将未确认的 EIGRP 分组作为单播分组重发给没有对前一个 EIGRP 分组进行确认的相邻路由器, 并继续向那些已经确认了 EIGRP 分组的相邻节点发送 EIGRP 组播分组。EIGRP 会以单播分组的形式, 向未确认 EIGRP 分组的路由器重发该分组 16 次。如果在进行了 16 次重试后, 该相邻路由器仍然没有进行确认, EIGRP 就会重置邻居关系, 而整个过程结束。16 次重试的超时时间通常会持续 50 至 80 秒。

6.6 EIGRP 分组格式

图 6-4 显示了 EIGRP 分组头。注意跟在自治系统号后面的是 Type/Length/Value (TLV) 三元组。TLV 三元组承载路由表项, 同时也提供 DUAL 进程管理字段。一般的 TLV 有 EIGRP 参数 TLV、IP 内部路由 TLV 和 IP 外部路由 TLV。

0	8	16	24	31
版本	操作码	校验和		
标志				
顺序号				
确认信息				
自治系统号				
TLVs				

图 6-4 EIGRP 分组头

EIGRP 分组参数描述如下:

- **版本**——指定不同版本的 EIGRP。EIGRP 版本 2 的实现是从 Cisco IOS 版本 10.3(11)、11.0(8) 和 11.0(3) 开始的。EIGRP 版本 2 是最新的版本, 它包含 EIGRP 稳定性和可扩展性的许多增强和改进。
- **操作码**——指定所包含的 EIGRP 分组的类型。opcode 1 是指更新分组; opcode 3 是指查询; opcode 4 是指回复; 而 opcode 5 是指 EIGRP 的 Hello 分组。
- **校验和**——用作常规的 IP 校验和, 它是基于除 IP 头外的整个 EIGRP 分组计算的。
- **标志**——现在这一参数只涉及两个标志。一个标志是指初始化信息或新的邻居关系, 另一个标志是指 EIGRP RTP 的条件接收。
- **序列号**——指定 EIGRP RTP 使用的序列号。
- **确认信息**——用于确认收到了一个可靠的 EIGRP 分组。
- **自治系统号**——指定用于表示 EIGRP 网络范围的编号。

一种最常见的 EIGRP TLV 是 EIGRP 参数 TLV, 如图 6-5 所示。它包含建立邻居关系所需要的参数。常数 K 的值包含在这个 TLV 中, 保持时间也包含在其中。两台路由器之间的 K 值必须一致才能建立邻居关系。

0	8	16	24	31
类型 = 0x0001				
K1	K2	K3	K4	
K5	保留	保持时间		

图 6-5 EIGRP 参数 TLV

图 6-6 和图 6-7 显示了另外两种常见的 EIGRP TLV——分别是 IP 内部路由 TLV 和 IP 外部路由 TLV。EIGRP 内部路由是从与接收方路由器相同的自治系统号产生的路由。而 EIGRP 外部路由是那些将要被重发布到 EIGRP 自治系统中的路由。

EIGRP IP 内部路由 TLV 包含这些信息:

- **下一跳**——分组将要被转发到的下一跳的 IP 地址。
- **延迟**——这是路由度量的延迟参数。这个延迟值是到达目的网络的路径上的所有接口的延迟总和。

0	8	16	24	31
类型 = 0x0102		长度		
下一跳				
延迟				
带宽				
MTU			跳数	
可靠性	负载	保留		
TLVs				
前缀长度	目的地址			

图 6-6 EIGRP 内部路由 TLV

- 带宽
- 径上
- MTU
- 跳数
- 可靠
- 靠性
- 负载
- 轻,
- 前缀
- EIGRP IP
- 源路
- 源自
- 治系
- 外部
- 外部
- 类型

不像 IGMP
着子网掩码。
VLSM 的更

- **带宽**——路由度量的带宽参数。这个带宽是从接口得到的，它是到达目的网络的路径上的接口的最低带宽。
- **MTU**——路由度量的接口 MTU 参数。
- **跳数**——到达目的网络的跳数。
- **可靠性**——接口的可靠性，是 1 到 255 之间的某一个值。可靠性为 1 表示接口的可靠性为 1/255，而 255 表示接口是百分之百可靠的。
- **负载**——接口的负载，是 1 到 255 之间的一个值。负载值为 1 表示接口的负载非常轻，而负载值为 255 表示接口已经高度饱和。
- **前缀长度**——目的网络的子网掩码。

EIGRP IP 外部路由 TLV 还包含更多的路由信息：

- **源路由器**——产生外部 EIGRP 路由的路由器的路由器 ID。
- **源自治系统**——某路由的 EIGRP 自治系统号，此时该路由还没有被重发布到这个自治系统中。
- **外部协议度量**——被重发布到 EIGRP 之前路由的度量。
- **外部协议 ID**——产生将被重发布到 EIGRP 的路由的路由协议类型。这些路由协议的类型可以是 BGP、OSPF、RIP、IGRP 等等。

0	8	16	24	31
类型 = 0x0103		长度		
下一跳				
源路由器				
源自主系统号				
随意标记				
外部协议度量				
保留		外部协议 ID	标志	
延迟				
带宽				
MTU			跳数	
可靠性	负载	保留		
前缀长度	目的地址			

图 6-7 EIGRP 外部路由 TLV

6.7 EIGRP 行为

不像 IGRP，EIGRP 是一种高级的距离向量协议，当一个更新分组被发出时，它还承载着子网掩码。因此，EIGRP 支持非连续网络和变长子网掩码（VLSM）。关于非连续网络和 VLSM 的更多讨论，请参看第 2 章“理解 RIP 协议”。图 6-8 显示了说明 EIGRP 支持非连续

网络的网络图。

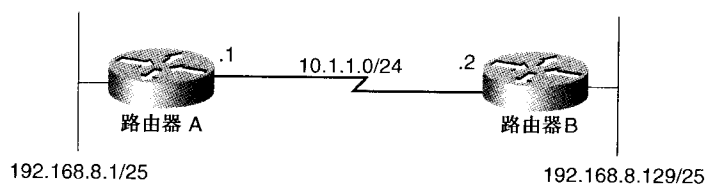


图 6-8 EIGRP 支持非连续网络的示例

图 6-8 显示了通过串行端口连接的两台路由器。路由器 B 有一个网络 192.168.8.128/25 需要通过网络 10.1.1.0/24 发送到路由器 A。在缺省情况下，因为 EIGRP 是一种传统的路由协议，所以路由器 B 将自动汇总穿过主网络边界的路由。因此，路由器 B 会发送 192.168.8.0/24 到路由器 A，而路由器 A 会忽略这次路由发送。要使 EIGRP 支持非连续网络，就必须在 router eigrp 命令下面设置 no auto-summary 命令。路由器 B 有了 no auto-summary 命令，就会发送路由 192.168.8.128/25 到路由器 A，接着路由器 A 就为该路由准备一个路由条目，非连续网络的问题就被解决了。

6.8 EIGRP 汇总

在 EIGRP 中发生两种类型的汇总——自动汇总和手动汇总。与 IGRP 和 RIP 一样，自动汇总是 EIGRP 的缺省行为。基本上，当某路由器发送出一个路由更新，它就会在路由跨过主网边界公布时自动将该路由汇总到它的自然的主网。图 6-9 显示了一个自动汇总的示例。在图 6-9 中，路由器 R1 需要通过主网络 192.168.2.0 发送一个关于网络 132.168.1.0 的更新信息到路由器 R2。然后 R1 会汇总这一更新信息到它的有类的网络 132.168.0.0 并将其发送到 R2。自动汇总的问题在于网络不能被设计为非连续的。

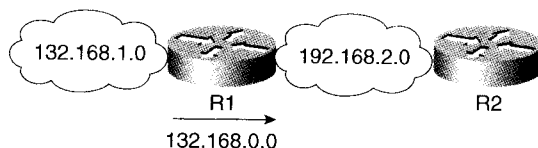


图 6-9 自动汇总的示例

在 EIGRP 中，手动汇总可以在网络中任何一台路由器中逐个接口地通用界面上进行配置。EIGRP 手动汇总的命令是 ip summary address eigrp autonomous-system-number address mask。不像 OSPF 只能在一台区域边界路由器（Area Border Router, ABR）或一台自治系统边界路由器（Autonomous System Border Router, ASBR）上进行汇总，使用 EIGRP，汇总可以在网络中任一接口和任一路由器上完成。当在接口上设置了手动汇总以后，路由器就会立即建立一个管理距离为 5 的空路由表。这是为了防止产生汇总地址的路由环。最后，当最后一个特定的汇总路由通过后，汇总路由就会被删除。例 6-2 显示了在图 6-10 所示的网络中，EIGRP 手动汇总的配置。

例 6-2 配置 EIGRP 手动汇总

```
interface s0
ip address 192.168.11.1 255.255.255.252
ip summary-address eigrp 1 192.168.8.0 255.255.252.0
```

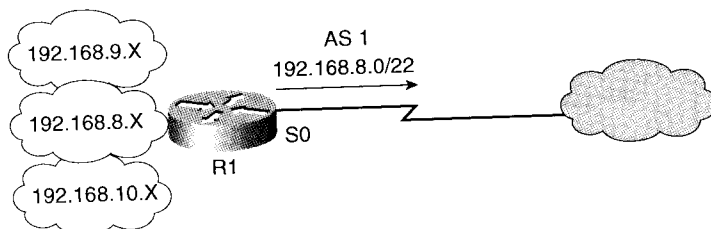


图 6-10 EIGRP 手动汇总的例

例 6-2 展示了图 6-10 中 R1 如何把地址 192.168.8.0/24、192.168.9.0/24 和 192.168.10.0/24 汇总到一个更新信息 192.168.8.0/22 中。EIGRP 中的汇总减小了路由表的大小和更新的数量。但同时它也限制了查询的范围，而这对于使一个大型网络更加稳定和更易于伸缩来说是很关键的。

6.9 EIGRP 查询过程

虽然 EIGRP 是一种高级的距离向量协议，其收敛时间很短，但一台 EIGRP 路由器仍然要依靠它的相邻路由器来公布路由器信息。要实现快速收敛，EIGRP 就不能像 IGRP 那样使用刷新计时器。为了快速收敛，EIGRP 需要主动地搜索丢失的路由。这一过程就称作查询过程，它在前几节中曾简要地讨论过。在查询过程中，当主路由丢失且没有可行的可行后继时，查询就被发送。在这一阶段，路由被称为是处于激活状态。

查询将被发送到所有相邻路由器上，并且被发送到除了到后继的接口以外的所有接口上。如果相邻路由器没有丢失的路由信息，就会有更多的查询发往相邻路由器的相邻路由器，直至到达查询的边界。查询边界要么由网络的末端构成，要么是发布列表的边界，或者是汇总边界。发布列表和汇总边界由设置了发布列表和汇总的路由器定义。当查询被发送后，路由器就必须等待相邻路由器发回的所有答复，然后才能计算后继信息。如果有任何相邻路由器未能够在三分钟之内发回回复，路由就叫做被粘滞于活动状态（Stuck in Active, SIA）了，那么未作出答复的相邻路由器的邻居关系就会被重置。第 7 章“EIGRP 协议疑难解析”，强调了 SIA 问题，并非常详细地叙述了如何进行诊断。

6.10 缺省路由和 EIGRP

不像 IGRP，EIGRP 把路由 0.0.0.0/0 识别为缺省路由，并允许该路由被重发布到 EIGRP 域作为缺省路由。EIGRP 也是用它自己的方法使用 ip default-network 命令传播缺省路由，正如 IGRP 那样。

ip default-network 命令的工作方式与其在 IGRP 下的工作方式完全相同。

ip default-network 命令指定一个主网络地址并将其标志为缺省网络。这个主网络可以由一个静态路由定义的直接连接，也可以是由动态路由发现的。图 6-11 说明了 ip default-network 命令是如何工作的。

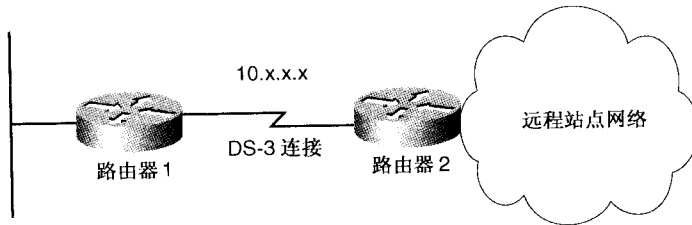


图 6-11 传播一个 IGRP 的缺省路由

在图 6-11 中，路由器 1 通过一个 DS-3 链路连到远程站点上。路由器 1 现在需要一个到路由器 2 以及到远程站点网络中所有路由器的缺省路由。在 IGRP 中，到 0.0.0.0 的路由是不能被识别为一个缺省路由的，因而相应地，路由器 1 必须配置 ip default-network 192.168.1.0 来将路由 192.168.1.0 标志为一个缺省路由。路由器 1 将发送出所有 192.168.1.0 的路由更新，并将其标志为缺省路由。当远程站点网络上的路由器收到 192.168.1.0 的更新时，它们就会将其标志为缺省路由，并将到 192.168.1.0 的路由作为最后解决办法的网关来装入。

6.11 EIGRP 中的非等价负载平衡

EIGRP 和 IGRP 使用相同的公式计算它们的度量，并且当处理非等价负载平衡时也具有同样的操作。与 IGRP 一样，EIGRP 也会安装多达六条的并行等价路径实现负载平衡。并且 EIGRP 也使用相同的 variance 命令，就像 IGRP 实现非等价负载平衡一样。

考虑如图 6-12 所示的网络。

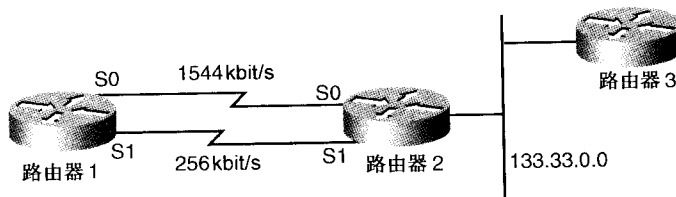


图 6-12 非等价负载平衡示例

记住这些多路径操作规则：

- 被利用作为替换通路的相邻路由必须距离目的地更近（也就是说，对于给定的目的地，它必须公布比本地路由器更小的度量）。后退和前进都是不可能的。
- 相邻路由公布的度量必须比本地路由器度量的变化量小。变化量 = 变化量因子 × 本地度量。

当路由器 1 计算它到路由器 3 的 EIGRP 度量时，通过 1544bit/s 链路的度量如下：

$$\text{EIGRP 度量} = 256(6476 + 2100) = 2,195,456$$

通过 256 kbit/s

没有非等价
正如例 6-3 的输

例 6-3 sh
一个非最优路由

```
Router_1#
Routing e
Known v
Redistr
Advertis
Last up
Routing
* 192.168
Rou
Total del
Reliabili
Loadin
```

使用 EIGRP

度量可能与某一
多重路由的度量

在例 6-3 中
对于在路由表中
很简单，只需需
出显示路由器

例 6-4 E

```
Router_1#
Routing e
Known v
Redistr
Advertis
Last up
Routing
* 192.168
Ro
Tot
Tot
Reliabil
Lo
10.1
Ro
To
Re
Lo
```

通过 256 kbit/s 链路的度量如下：

$$\text{EIGRP 度量} = 256(39,062 + 2100) = 10,537,472$$

没有非等价负载平衡，EIGRP 就不会简单地选择 1544bit/s 的链路转发分组到路由器 3，正如例 6-3 的输出所显示的那样。

例 6-3 show ip route 命令的输出表明在没有非等价负载平衡的情况下路由器 1 选择了一个非最优路由

```
Router_1#show ip route 133.33.0.0
Routing entry for 133.33.0.0/16
  Known via "eigrp 1", distance 90, metric 2195456
  Redistributing via eigrp 1
  Advertised by eigrp 1 (self originated)
  Last update from 192.168.6.2 on Serial0, 00:00:20 ago
  Routing Descriptor Blocks:
  * 192.168.6.2, from 192.168.6.2, 00:00:20 ago, via Serial0
    Route metric is 2195456, traffic share count is 1
  Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 0
```

使用 EIGRP 的非等价负载平衡特性需要使用 **variance** 命令。变化量是一个乘数，其中度量可能与某一个路由的最低度量不同。变化量的值必须是整数值，缺省的值 1，意思是多重路由的度量必须等于负载平衡。

在例 6-3 中，通过 256kbit/s 链路的度量是通过 1544kbit/s 链路的度量的 4.8 倍，因此，对于在路由表中考虑 256kbit/s 链路的情况，必须在路由器 1 中配置一个变化量 5。这一配置很简单，只需简单地在 **router eigrp** 命令下加上 **variance 5**。例 6-4 中 **show ip route** 命令的输出显示路由器 1 在其路由表中装入了这两种链路。

例 6-4 EIGRP 中非等价负载平衡的输出

```
Router_1#show ip route 133.33.0.0
Routing entry for 133.33.0.0/16
  Known via "eigrp 1", distance 90, metric 2195456
  Redistributing via eigrp 1
  Advertised by eigrp 1 (self originated)
  Last update from 10.1.1.2 on Serial1, 00:01:02 ago
  Routing Descriptor Blocks:
  * 192.168.6.2, from 192.168.6.2, 00:01:02 ago, via Serial0
    Route metric is 2195456, traffic share count is 5
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 0
  10.1.1.2, from 10.1.1.2, 00:01:02 ago, via Serial1
    Route metric is 10537472, traffic share count is 1
    Total delay is 21000 microseconds, minimum bandwidth is 256 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 0
```

在例 6-4 中, 通过 Serial 0 的路由的流量共享数为 5, 而通过 Serial 1 的流量共享数为 1。这表明路由器在 Serial 1 上每发送一个分组, 就会在 Serial 0 上发送 5 个分组。

6.12 小 结

EIGRP 和 IGRP 在某些方面很相似, 但在其他方面却又存在区别。EIGRP 和 IGRP 使用相同的公式来计算到目的网络的度量, 也使用相同的技术实现负载平衡。然而, EIGRP 保存一张网络的拓扑表, 并使用 DUAL 机制选择一条无环路由路径。EIGRP 还使用后继和可行后继的概念以及查询过程实现快速收敛。EIGRP 在发出路由更新的同时也会携带子网掩码信息, 这就使 EIGRP 能够支持非连续网络和 VLSM, 从而使 EIGRP 成为一种可扩展的路由协议, 适应当今的网络需求。表 6-1 显示了 IGRP 与 EIGRP 的总结性比较。

表 6-1

IGRP 与 EIGRP 的比较情况表

IGRP	EIGRP
度量计算如下: IGRP 度量= [K1 * 带宽 + (K2 * 带宽) + K3 * 延迟] * K5 (256—负载) (可靠性+K4)	度量计算如下: IGRP 度量 × 256
不支持 VLSM 和非连线网络	支持 VLSM 和非连续网络
不保持邻居关系	在邻居表中保持邻居关系
易于形成路由环	保存网络拓扑表并使用 DUAL 机制来选择
具有长收敛时间	因为有可行后继和查询过程, 所以收敛时间短
不重发丢失的 IGRP 更新分组	有可靠的传输机制来重发丢失的 EIGRP 更新
不支持手动汇总和无类路由聚合	支持手动汇总和无类路由聚合
不将 0.0.0.0/0 理解为缺省路由	将 0.0.0.0/0 理解为缺省路由

6.13 复 习 题

1. IGRP 与 EIGRP 的度量计算有什么不同?
2. 什么是 EIGRP 查询? 它是用来干什么的?
3. 术语“主动路由”的含义是什么?
4. 什么是可行后继?
5. EIGRP 的组播地址是什么?
6. 什么是可行性条件?
7. 什么叫粘滞于活动状态?

第 7 章

EIGRP 协议 疑难解析

本章涵盖下列关键主题：

- EIGRP 邻居关系疑难解析。
- EIGRP 路由公布疑难解析。
- EIGRP 路由安装疑难解析。
- EIGRP 路由抖动疑难解析。
- EIGRP 路由汇总疑难解析。
- EIGRP 路由重发布疑难解析。
- EIGRP 拨号备份疑难解析。
- EIGRP 错误消息。

本章讨论 EIGRP 中的一些常见问题以及解决这些问题的方法。在必要的地方给出了调试、配置命令和有用的 show 命令。

注意：调试工具可能会占用大量的 CPU 资源并且可能会对网络产生不利的影响。因此，除非 Cisco 技术支持中心（Technical Assistance Center, TAC）指出使用调试工具，在一个产品化的网络中不推荐使用它们。

有时候，同一个问题可能有多个原因。因此，如果一个情景没有修正这个网络问题，一般要检查其他的情景。

7.1 EIGRP 邻居关系疑难解析

本节讨论 EIGRP 邻居关系相关问题的疑难解析方法。导致 EIGRP 邻居关系出现问题的最常见的原因有：

- 单向链路。
- 非公共子网、主地址和二级地址不匹配。

- 不匹配的掩码。
- K 值不匹配。
- 不匹配的 AS 号。
- 被粘滞于活动状态 (stuck in active)。
- 第 2 层的问题。
- 访问列表拒绝组播分组。
- 手动更改 (汇总路由器、度量改变、路由过滤)。

图 7-1 给出了解决 EIGRP 邻居关系常见问题的流程图。

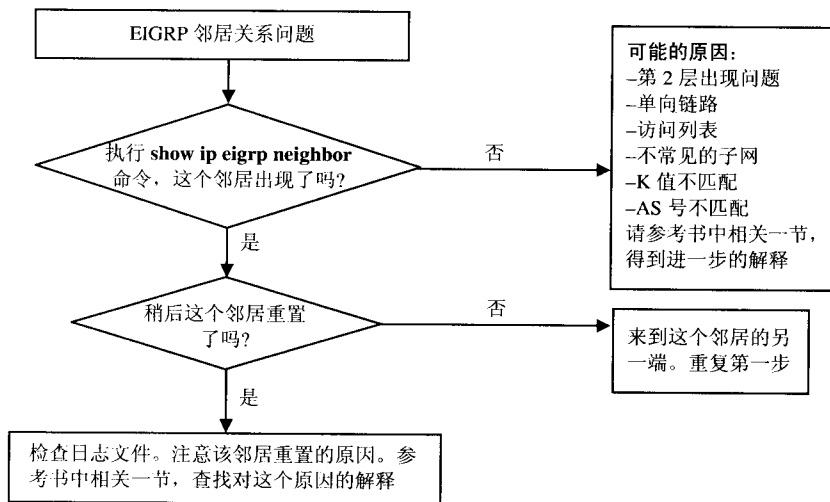


图 7-1 解决 EIGRP 邻居关系常见问题的流程图

7.1.1 查阅 EIGRP 日志获悉邻居改变

每当 EIGRP 重置它的邻居关系时, 都会在日志中记录重置的原因。在早期的 Cisco IOS 中, 启用这一特性的配置是必需的。在 **router eigrp** 命令下配置命令 **eigrp log-neighbor-change**。在 Cisco IOS 版本 12.1.3 及以后的版本中, **eigrp log-neighbor-change** 命令变成这个路由器的缺省设置。关于 EIGRP 邻居日志的一个例子如下所示:

```
%DUAL -5 -NBRCHANGE: IP -EIGRP EIGRP AS number: Neighbor neighbor IP address is down:
reason for neighbor down.
```

表 7-1 用文件记录了可以在 EIGRP 日志中找到的邻居关系的变化情况及其含义和基于日志消息修正这个问题所需要的动作。

表 7-1 在 EIGRP 日志中用文档证明的邻居关系的变化

日志消息	含 义	解决问题的动作
NEW ADJACENCY	表示已经建立了一个新的邻居	不需要任何动作
PEER RESTARTED	表示另一个邻居启动该邻居关系的重置操作。得到这条消息的路由器不是重置该邻居关系的那个路由器	在得到这条消息的路由器上不需要任何动作。搜集另一个邻居上的 EIGRP 邻居日志信息

图 7-2 当

续表

日志消息	含 义	解决问题的动作
HOLD TIME EXPIRED	表示在这个保持时间的范围内, 路由器还没有收到来自该邻居的任何 EIGRP 分组	因为这是一种分组丢失问题, 所以检查是否属于在第 2 层出现问题。使用图 7-2 中显示的流程图来解决
RETRY EXCEEDED	LIMIT 表示 EIGRP 没有收到来自该邻居的对 EIGRP 可靠分组的确认分组, 以及 EIGRP 已经尝试重新传输这个可靠分组 16 次, 但没有一次成功	使用图 7-3 中显示的流程图解决这个问题
ROUTE CHANGED	FILTER 表示因为在该路由过滤器中有一个改动 (在 router eigrp 命令下执行命令 distribute-list), 所以该 EIGRP 邻居正在重置	不需要任何动作。这是 EIGRP 中的正常行为, EIGRP 需要在路由过滤器改变的时候重置这个邻居, 并且需要在邻居之间重新同步 EIGRP 拓扑表
INTERFACE CHANGED	DELAY 表示因为在这个接口上的延迟参数中有一个手动配置改动, 所以该 EIGRP 邻居正在重置	不需要任何动作。这是 EIGRP 中的正常行为, EIGRP 需要在延迟参数改变的时候重置这个邻居
INTERFACE BANDWIDTH CHANGED		不需要任何动作。这是 EIGRP 中的正常行为, EIGRP 需要在带宽参数改变的时候重置这个邻居
STUCK IN ACTIVE	表示因为 EIGRP 被粘滞于活动状态, 所以该 EIGRP 邻居正在重置。该邻居被重置是由于被粘滞于活动状态的结果	从被粘滞于活动状态的角度解决问题。参考“EIGRP 邻居问题——原因: 被粘滞于活动状态”一节

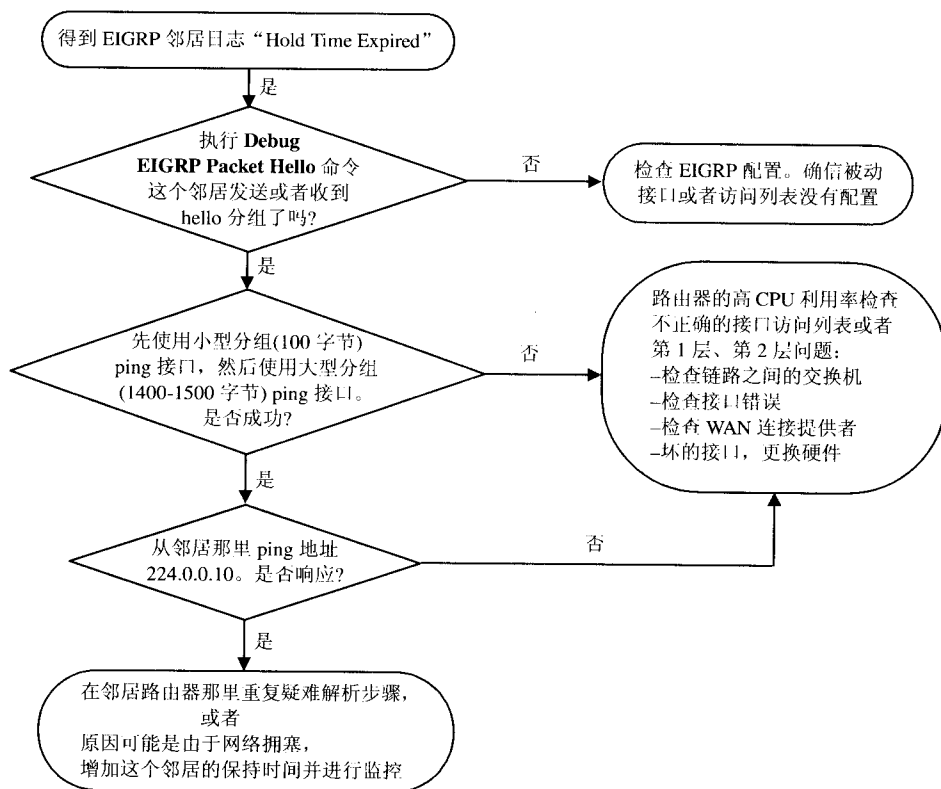


图 7-2 当得到邻居日志消息“HOLD TIME EXPIRED”的时候解决 EIGRP 邻居关系问题的流程图

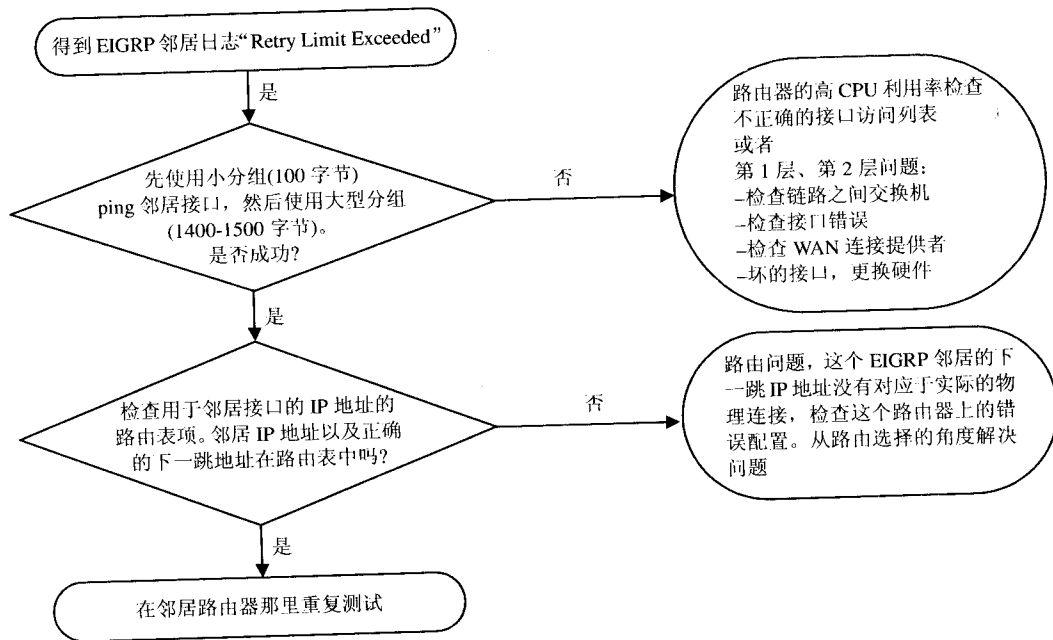


图 7-3 当得到邻居日志消息“RETRY LIMIT EXCEEDED”时疑难解析 EIGRP 邻居关系的流程图

7.1.2 EIGRP 邻居问题——原因：单向链路

有时候，由于 WAN 连接产生的问题会导致 EIGRP 具有一个单向的邻居关系。一般情况下，一个单向的邻居关系是由这些邻居之间的单向连接引起的。单向连接的原因一般是在第 2 层出现了问题。例如，链路可能经历许多 CRC 错误、交换机问题或者大型分组或小型分组带来的一次 ping 测试失败。这种情况下，需要请负责这个链路状况的工作组来检查该链路的完整性。有时候，只是一个错误配置的访问列表就导致 EIGRP 形成一个单向的邻居关系。图 7-4 示范了由于单向链路导致的一个 EIGRP 问题的例子。



图 7-4 容易遭受由于单向链路造成的 EIGRP 邻居问题的网络拓扑

在图 7-4 中，路由器 RTR A 和 RTR B 通过一个 WAN 连接被连接在一起。从 RTR A 到 RTR B 的电路是好的，但是从 RTR B 到 RTR A 的电路被破坏。在 RTR A 上执行命令 `show ip eigrp neighbor` 得到的结果不会显示任何东西，因为 RTR B 的 EIGRP 的 hello 分组无法把结果送到 RTR A。例 7-1 显示了 RTR B 上 `show ip eigrp neighbor` 命令的输出。

例 7-1 在 R

```

RtrB#show ip eigrp neighbor
IP-EIGRP Neighbor Table
H Address      H Pri  H Metric  S
1 10.88.18.1
  
```

RTR B 把 RTR A 的 hello 分组视为问题。从 `show ip eigrp neighbor` 命令的输出中，RTR B 的 hello 分组计时器的值（RTO）计时器的值是一个重要的线索，暗示着问题。

- 平滑往返时间：这个邻居的往返时间（RTO）值。
- 重传超时：这个邻居的 hello 分组重传超时时间。
- Q 计数：这个邻居的 hello 分组重传次数。

参考例 7-1，

这一点表明该链路已经尝试了 16 次来重新发送 hello 分组，但 EXCEEDED，然后链路没有完成的，不是组有问题，这个

问题的解决是链路的提供者，被修复之后，这个 WAN 链路已经

例 7-2 sh

```

RtrB#show ip eigrp neighbor
IP-EIGRP Neighbor Table
H Address      H Pri  H Metric  S
1 10.88.18.1
  
```

注意，现在

7.1.3

很多情况下，造成这个 EIGRP 邻居关系起麻烦，就会

IP-EIGRP

例 7-1 在 RTR B 上 show ip eigrp neighbor 命令的输出

```
RtrB#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H Address Interface Hold Uptime SRTT RTO Q Seq
                          (sec) (ms) Cnt Num
1 10.88.18.2 S0 14 00:00:15 0 5000 4 0
```

RTR B 把 RTR A 作为一个邻居显示，因为 RTR A 的 EIGRP hello 分组到达 RTR B 没有问题。从 show 命令的输出可以看出：SRTT 的值是 0 毫秒；重传超时（retransmission timeout, RTO）计时器的值是 5000 毫秒；Q 计数器的值是 4，而且没有递减。这三个数字提供了最重要的线索，暗示这是一个单向链路问题。下面是 SRTT、RTO 和 Q 计数器的含义：

- 平滑往返时间（smooth round-trip time, SRTT）——表示一个 EIGRP 分组被发送到这个邻居以及该本地路由器收到这个分组的一个确认分组所花费的毫秒数。
- 重传超时（retransmission timeout, RTO），以毫秒为单位——表示在从重传队列向一个邻居重新传输一个分组之前软件等待的时间数。
- Q 计数器——表示软件等待发送的 EIGRP 分组（Update、Query 和 Reply）的数目。

参考例 7-1，SRTT 计时器是 0 的事实表明没有收到任何确认分组。Q 计数器没有递减，这一点表明该路由器正在试图发送 EIGRP 分组，但是没有收到任何确认分组。RTR B 会重试 16 次来重新发送这个分组；最后，RTR B 会重置这个邻居关系，使用其日志指明 RETRY LIMIT EXCEEDED，然后这一过程再次启动。还有，要记住同一个分组的这 16 次重传是使用单播完成的，不是组播。因此，RETRY LIMIT EXCEEDED 消息指明在这个链路上传输组播的分组有问题，这个问题最有可能是一个在第 1 层或者第 2 层出现的问题。

问题的解决方法是从第 2 层的角度来解析问题。在这个例子中，需要打电话给这个 WAN 链路的提供者，以查明为什么从 RTR B 到 RTR A 的电路被破坏。当 RTR B 到 RTR A 的链路被修复之后，这个问题就会解决。在例 7-2 中，show ip eigrp neighbors 的输出显示了这个 WAN 链路已经修复之后的邻居关系。

例 7-2 show ip eigrp neighbor 命令的输出证实问题已解决

```
RtrB#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H Address Interface Hold Uptime SRTT RTO Q Seq
                          (sec) (ms) Cnt Num
1 10.88.18.2 S0 14 01:26:30 149 894 0 291
```

注意，现在 Q 计数器这一列是 0，SRTT 和 RTO 有了有效的数值。

7.1.3 EIGRP 邻居问题——原因：非公共子网

很多情况下，因为邻居不是在同一个子网中，所以 EIGRP 不会建立邻居关系。一般情况下，造成这个问题的原因是路由器配置错误。EIGRP 在建立邻居关系时由于非公共子网而引起麻烦，就会出现下面的错误消息：

```
IP-EIGRP: Neighbor ip address not on common subnet for interface
```

图 7-5 显示了当“Neighbor not on common subnet”错误出现在路由器上的时候解决这一问题的流程图。

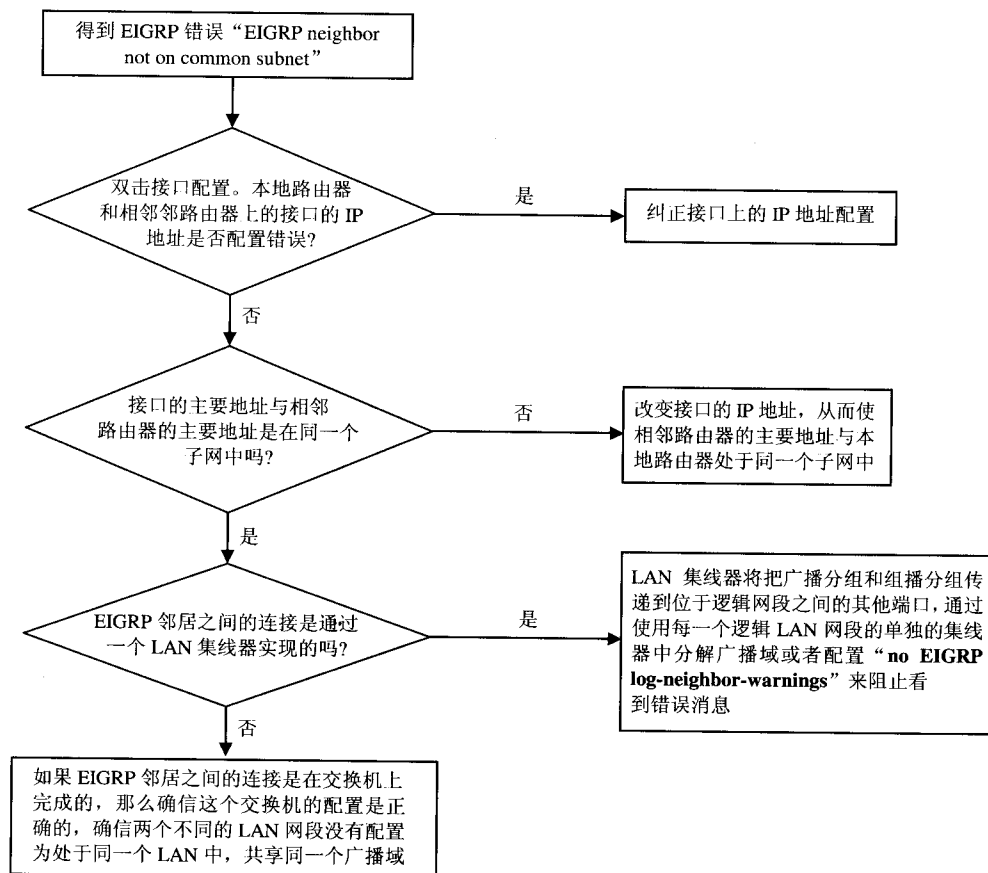


图 7-5 问题解决流程图

根据图 7-5 中的疑难解析流程图, 得到“EIGRP neighbor not on common subnet”错误消息的原因有三个, 如下所示:

- IP 地址在接口上被错误配置。
- 相邻接口的主 IP 地址和二级 IP 地址不匹配。
- 位于这一 EIGRP 邻居连接之间的交换机或集线器被错误配置或者正在向其他端口泄露组播分组。

1. 接口上 IP 地址的错误配置

有时候, 与其他 EIGRP 邻居不是位于一个公共子网上的一个 EIGRP 邻居正是由于错误配置了接口上的 IP 地址而造成的结果。例如, 网络管理员可能把 IP 地址 192.168.3.1 255.255.255.252 误输入成 192.168.3.11 255.255.255.252, 从而导致 EIGRP 抱怨该邻居不是在一个子网上。

2. 相邻接口的主 IP 地址和二级 IP 地址不匹配

正如在第 6 章“理解 EIGRP 协议”中所提到的, EIGRP 从接口的主地址发送 hello 分组。

如果路由器上的形成邻居关系, 种情景。

主 IP 地址
二级 IP 地址

图

在图 7-6 中配置了一个 50.1.1.1。决定于哪一个路由器 C 收到来自路由器 A 或者路由器 B 都不会形成邻居关系。这个例子的网络而言,

3. EIGRP 邻居分组

如果位于 EIGRP 邻居这些 EIGRP 邻居同 LAN 网段的 EIGRP 传递广播分组和组播分组, 那么位于 LAN 网段的解决方法是通过在配置下面配置 no

如果是一个确信这个交换机不是这样配置的: 广播域。

7.1.4 EIGRP

有时候, 接口这种情景的网络

如果路由器上的主网络地址被用于第二台路由器上的二级网络地址，反之亦然，那么就不会形成邻居关系，而且这些路由器会抱怨这一邻居不是在一个公共的子网上。图 7-6 示范了这种情景。

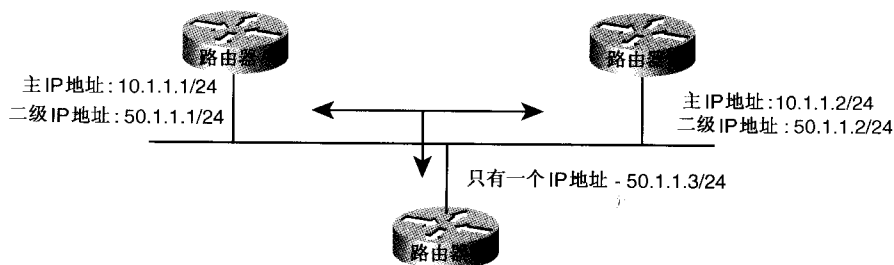


图 7-6 容易遭受由于主 IP 地址和二级 IP 地址不匹配产生的 EIGRP 邻居问题的网络拓扑

在图 7-6 中，路由器 A 和路由器 B 在 10.1.1.0/24 网络内有一个主地址，而路由器 C 配置了一个 50.1.1.0/24 的地址范围。当路由器 A 或者路由器 B 发送 EIGRP hello 分组时，取决于哪一个路由器发送了 hello 分组，hello 分组的源地址将是 10.1.1.1 或者 10.1.1.2。当路由器 C 收到来自路由器 A 或者路由器 B 的 hello 分组时，它注意到源地址是来自 10.1.1.0 网络。因为路由器 C 在接口上配置了一个 50.1.1.3 的 IP 地址，所以路由器 C 不会处理来自路由器 A 或者路由器 B 的 hello 分组，原因是它们来自不同的网络。因此，路由器 C 与 A 或 B 都不会形成邻居关系。

这个例子的解决方法是把该网段上的所有 IP 地址都匹配到主网络地址空间。对于图 7-6 中的网络而言，需要把路由器 C 配置成位于 10.1.1.0/24 的主网络地址空间中。

3. EIGRP 邻居连接之间的交换机或集线器被错误配置或者正在向其他端口泄露组播分组

如果位于 EIGRP 邻居之间的接口上的 IP 地址配置是正确的，那么你或许希望检查连接这些 EIGRP 邻居的交换机或者集线器上的配置。如果一个单独的 LAN 集线器连接了来自不同 LAN 网段的 EIGRP 邻居，那么这一集线器就会向位于两个逻辑 LAN 网段之间的其他端口传递广播分组和组播分组。因此，如果一个集线器连接了位于不同 LAN 网段的所有 LAN 设备，那么位于 LAN 网段 1 中的邻居就会看到来自 LAN 网段 2 的组播 EIGRP hello 分组。解决方法是通过在每个 LAN 网段都使用一个集线器来分解这一广播域，或者简单地在 EIGRP 配置下面配置 `no eigrp log-neighbor-warnings` 阻止看到这一错误消息。

如果是一个 LAN 交换机连接这些 LAN 设备，那么你或许希望检查这个交换机的配置。确信这个交换机不是这样配置的：不同的 LAN 网段驻留在同一个 VLAN 中。确信这个交换机是这样配置的：每一个 LAN 网段都有它自己的广播域，没有与其他 LAN 网段共享它的广播域。

7.1.4 EIGRP 邻居问题——原因：不匹配的掩码

有时候，接口子网掩码的一个简单的配置错误也会导致 EIGRP 邻居问题。图 7-7 示范了这种情景的网络图。

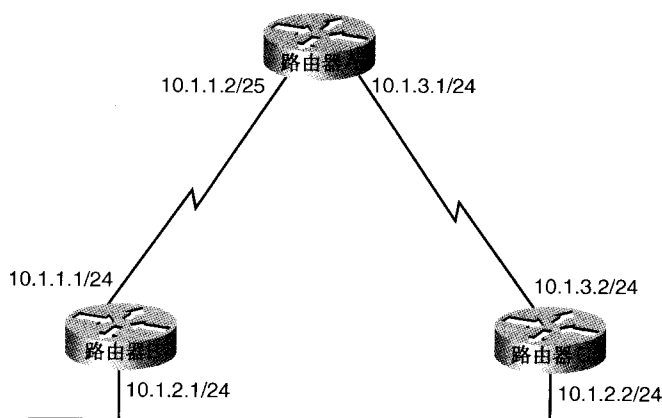


图 7-7 容易遭受由于不匹配掩码产生的 EIGRP 邻居问题的网络拓扑

例 7-3 显示了路由器 A、B 和 C 的配置。

例 7-3 用于图 7-7 中网络的路由器 A、B 和 C 的配置

```
Router A#interface serial 0
ip address 10.1.1.2 255.255.255.128
interface serial 1
ip address 10.1.3.1 255.255.255.0
```

```
Router B#interface serial 0
ip address 10.1.1.1 255.255.255.0
interface ethernet 0
ip address 10.1.2.1 255.255.255.0
```

```
Router C#interface ethernet 0
ip address 10.1.2.2 255.255.255.0
interface serial 0
ip address 10.1.3.2 255.255.255.0
```

注意路由器 A 和 B 的串行接口上不匹配的掩码。路由器 A 有一个 255.255.255.128 的掩码，而路由器 B 在 Serial 0 上有一个 255.255.255.0 的掩码。最初，在路由器 A 和 B 之间建立 EIGRP 邻居没有问题，因为 10.1.1.1 和 10.1.1.2 在同一个子网中。当建立了邻居关系后，路由器 A 和 B 开始交换 EIGRP 拓扑表，并且基于这张 EIGRP 拓扑表建立路由的时候，问题出现了，如例 7-4 中所示。

例 7-4 路由器 B 和 C 的路由表

```
Router B#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
```

(待续)

*
P

Gateway

C 10.1.

D 10.1.

Router

Codes:

Gateway

D 10.1.1

D 10.1.1

当路由器

器 B 的目的地

的时候，因为路

ACK 分组，而不

/25 路由覆盖/2

分组时，它查看

个 ACK 分组转

到 B，再从 B

来自路由器 A

新分组，在 16

这个问题

255.255.255.0。

7.1.5

对于 EIGRP

考第 6 章，其中

有接口的带宽和

(比如负载和可

被用在计算中，

些 K 值必须是相

子。

对于图 7-8

K4 的 K 值全部

RTR A 和 RTR

```

* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set
C 10.1.1.0/24 Serial 0
D 10.1.1.0/25 10.1.2.2
Router c#show ip route eigrp
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set
D 10.1.1.0/24 10.1.2.1
D 10.1.1.0/25 10.1.3.1

```

当路由器 B 向路由器 A 发送一次 EIGRP 更新时, 路由器 A 通过一个带有发送到路由器 B 的目的地址是 10.1.1.1 的 EIGRP 确认分组响应这次更新。当路由器 B 接收到这个分组的时候, 因为路由器 B 有一个来自路由器 C 的更详细的路由, 所以它向路由器 C 转发这个 ACK 分组, 而不是处理它。路由器 B 有一个更详细的路由: 10.1.1.0/25, 接下来转发到 10.1.2.2./25 路由覆盖/24 路由, 因为/25 路由比/24 路由更详细。当路由器 C 收到来自路由器 B 的 ACK 分组时, 它查看其路由表寻找 10.1.1.1 表项, 这张路由表指向路由器 A。路由器 C 于是把这个 ACK 分组转发回路由器 A。这样就产生一个路由环。发送到 10.1.1.1 的分组从路由器 A 到 B, 再从 B 到 C, 又从 C 返回到路由器 A, 形成了一个环路。结果, 路由器 B 不会处理来自路由器 A 的 ACK 分组; 路由器 B 会认为路由器 A 从来没有用 ACK 分组确认这一更新分组, 在 16 次重试之后, 路由器 B 会重置这一邻居关系。

这个问题的解决方法是: 把路由器 A 的 Serial 0 接口上的子网掩码配置成正确的 255.255.255.0。

7.1.5 EIGRP 邻居问题——原因: 不匹配的 K 值

对于 EIGRP 建立邻居关系而言, 操纵该 EIGRP 度量的 K 常量值必须是相同的。请参考第 6 章, 其中有对 K 值的解释。在 EIGRP 的度量计算中, K 值的缺省设置是这样的: 只有接口的带宽和延迟被用于计算 EIGRP 度量。网络管理员可能经常想利用其他的接口参数 (比如负载和可靠性) 来确定 EIGRP 度量。因此, 这些 K 值被改动。因为只有带宽和延迟被用在计算中, 缺省情况下剩余的 K 值都被设置为值 0。但是, 对所有的路由器而言, 这些 K 值必须是相同的, 否则 EIGRP 就不会建立邻居关系。图 7-8 显示了这种情况的一个例子。

对于图 7-8 中的网络而言, K1 是带宽, K3 是延迟。网络管理员把 RTR B 从 K1 到 K4 的 K 值全部改成 1, 而 RTR A 保留 K1 和 K3 的缺省值, 仍然为 1。在这个例子中, RTR A 和 RTR B 不会形成 EIGRP 邻居关系, 因为 K 值不匹配。例 7-5 显示了 RTR B 的

配置。



图 7-8 容易遭受由于不匹配的 K 值产生的 EIGRP 邻居问题的网络拓扑

例 7-5 图 7-8 中 RTR B 的配置

```
RTR B#router eigrp 1
network xxxx
metric weights 0 1 1 1 1 0
```

RTR B 的配置包括额外的 **metric weights** 命令。第一个数字是服务类型（type of service, ToS）编号，因为它不被支持，所以它设置为值 0。ToS 后面的五个数字是从 K1 到 K5 的值。

疑难解析这个问题需要仔细研究路由器的配置。解决方法是把所有相邻路由器上的 K 值都改成同样的值。本例中，在路由器 A 上改变其 K 值以匹配路由器 B 的 K 值就会解决问题，如例 7-6 中所示。

例 7-6 在路由器 A 上配置 K 值以匹配路由器 B

```
RTR A#router eigrp 1
network xxxx
metric weights 0 1 1 1 1 0
```

7.1.6 EIGRP 邻居问题——原因：不匹配的 AS 号

对于在不同自治系统中的邻居而言，EIGRP 不会形成任何邻居关系。如果这些 AS 号不匹配，那么不会形成任何邻接关系。一般情况下，这个问题是由于路由器配置错误造成的。图 7-9 示范了这样一个问题。

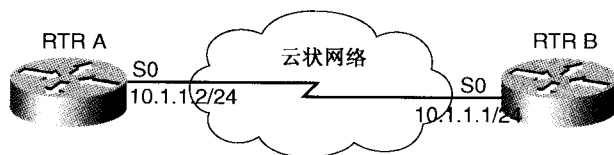


图 7-9 由于不匹配的 AS 号而产生 EIGRP 邻居问题的网络

在图 7-9 中显示的网络中，RTR A 和 RTR B 位于 EIGRP AS 1 中，正确的网络编号已经配置；但是，在 RTR A 和 RTR B 之间没有 EIGRP 邻居关系形成。首先在例 7-7 中检查 RTR A 和 RTR B 的配置。

例 7-7 图

```
RTR B#show
interface
IP address
router eigrp
network 10
```

```
RTR A#show
Interface
IP address
router eigrp
network 10
```

你应该立刻
而 RTR A 的 Se
有 EIGRP 邻居
EIGRP AS 号即
AS 1 中。

例 7-8 使用

```
RTR A#router
network 10
```

```
RTR B#router
network 10
```

7.1.7 E

有时候，EIGRP
是：

```
%DUAL -3 -3
up
这一小节讨论
```

1. 回顾 EIGRP

为了解决 EIGRP
第 6 章，其中有
EIGRP 是一
协议来告知协议
息。EIGRP 保留
法使用时，EIGRP
如果主要路由无
快速收敛的唯一
那么它又向它的

例 7-7 图 7-9 中 RTR A 和 RTR B 的配置

```
RTR B#show running-config
interface serial 0
IP address 10.1.1.1 255.255.255.0
router eigrp 11
network 10.0.0.0
```

```
RTR A#show running-config
Interface serial 0
IP address 10.01.1.2 255.255.255.0
router eigrp 1
network 10.0.0.0
```

你应该立刻注意到配置错误了。RTR B 的 Serial 0 接口被配置为在 EIGRP AS 11 中，而 RTR A 的 Serial 0 被配置为在 EIGRP AS 1 中。因为跨越链路的 AS 号不匹配，所以没有 EIGRP 邻居关系会形成。为了解决这一问题，只要把这两个路由器配置为具有相同的 EIGRP AS 号即可，如例 7-8 所示。在这个例子中，这两个路由器都将被配置为在 EIGRP AS 1 中。

例 7-8 使用相同的 EIGRP AS 号配置这两个路由器

```
RTR A#router eigrp 1
network 10.0.0.0
```

```
RTR B#router eigrp 1
network 10.0.0.0
```

7.1.7 EIGRP 邻居问题——原因：被粘滞于活动状态

有时候，EIGRP 会由于一个“被粘滞于活动状态”条件而重置邻居关系。错误消息是：

```
%DUAL -3 -SIA: Route network mask stuck-in-active state in IP-EIGRP AS. Cleaning up
```

这一小节讨论疑难解析 EIGRP 被粘滞于活动状态错误的方法。

1. 回顾 EIGRP DUAL 过程

为了解决 EIGRP 被粘滞于活动状态的错误，需要理解 EIGRP 中的 DUAL 过程。请参考第 6 章，其中有针对 DUAL 过程的全面讲述，在这里我们简单的回顾一下。

EIGRP 是一个高级的距离向量协议；它没有使用 LSA 泛洪（比如 OSPF）或者链路状态协议来告知协议关于网络的全景。EIGRP 仅仅依赖其邻居获取关于网络可达性和可用性的信息。EIGRP 保留一张称为可行后继者（feasible successor）的备份路由的列表。当主要路由无法使用时，EIGRP 立即把它的可行后继者用作其备份路由。这样做缩短了收敛时间。现在，如果主要路由无法使用，而且没有可行的后继者，那么路由就处于活跃状态。这时，EIGRP 快速收敛的唯一方法是向其邻居查询不可用路由的情况。如果邻居不知道这个路由的状态，那么它又向它的邻居询问，依此类推，直到到达该网络的边界。如果发生下面情况之一，查

询结束:

- 所有的查询都被所有的邻居回答。
- 到达网络的末尾。
- 丢失的路由是这些邻居不能识别的。

问题是, 如果没有查询界线, 那么 EIGRP 有可能会向网络中的所有路由器询问丢失的路由。当 EIGRP 开始查询其邻居的时候, 一个被粘滞于活动状态计时器启动。缺省情况下, 这个计时器是三分钟。如果在三分钟之内 EIGRP 没有收到来自其所有邻居的查询响应, 那么 EIGRP 宣称这个路由被粘滞于活动状态, 并且重置没有响应这一查询的邻居关系。图 7-10 示范了当一个路由丢失的时候 EIGRP 的查询过程。

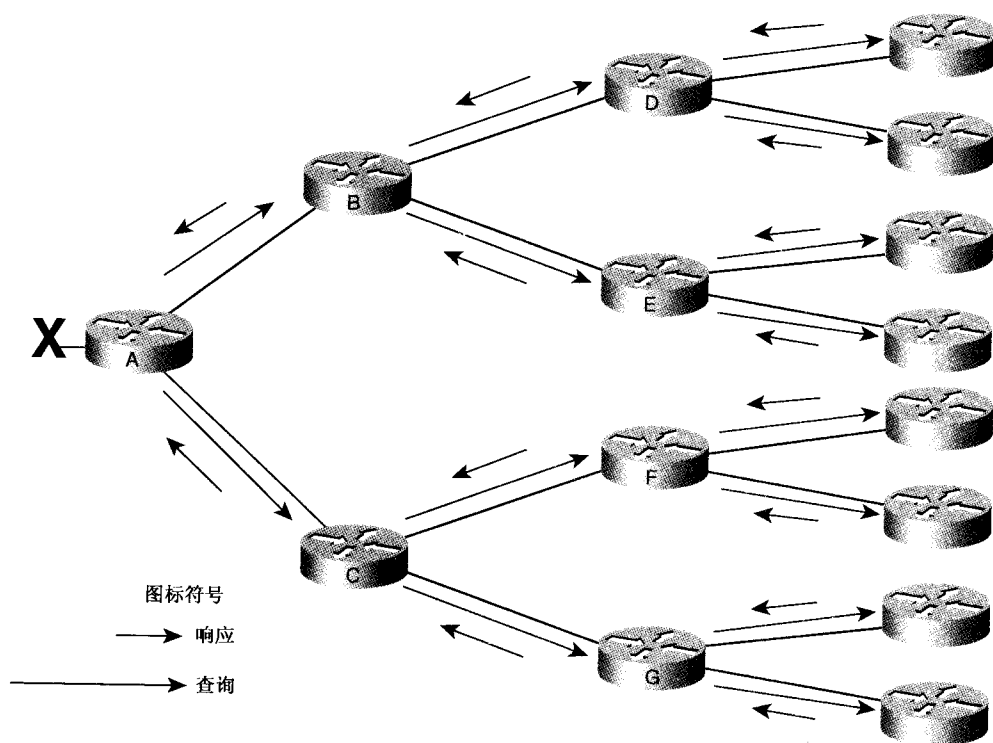


图 7-10 当路由丢失时 EIGRP 的查询过程

在图 7-10 中, 路由器 A 丢失了其 Ethernet 接口。因为它没有一个可行的后继者, 所以这一路由变成活跃的, 路由器 A 询问其邻居: 路由器 B 和路由器 C。现在, 路由器 B 不知道如何到达这一丢失的网络, 因此它询问其邻居: 路由器 D 和路由器 E。类似地, 路由器 C 询问其邻居: 路由器 F 和路由器 G。因为路由器 D、E、F 和 G 也不知道如何到达这一丢失的网络, 所以它们查询下游的邻居。这时, 到达了网络的边界, 而边界路由器没有其他任何邻居可以查询。于是边界路由器返回来响应路由器 D、E、F 和 G。这些路由器又响应路由器 B 和 C, 最后返回到路由器 A。然后这一查询过程结束。图 7-10 显示了这一 EIGRP 查询过程的级联效应 (cascade effect), 其中查询是从最初的路由器进行到网络的边界, 然后返回到最初的路由器。

2. 使用 show

为了解决 EIGRP

- 这个路由
- 这个路由

确定这个路由

能是由于链路抖动
一个断开的拨号通
得多的任务——而
粘滞于活动状态:

- 坏链路或
- 路由器资
- 查询范围
- 过多的冗

缺省情况下,

在三分钟内收到查
于活动状态的难度
获这一活跃路由查

疑难解析 EIGRP

令。这条命令显示
响应、哪些邻居没
且可以通过转发到
了 show ip eigrp to

例 7-9 show

```
Router#show
IP-EIGRP Top
A 20.2.1.0/2
1 replies,
via 10.1.
via 10.
Remaining r
Via 10.
```

正如例 7-9 中输

query-origin 是 Succ
它收到来自 10.1.3.1
知道路由 20.2.1.0.s
根据例 7-9 的输出,

为了进一步进

show ip eigrp top
Remaining replies:部
的另一个输出。

2. 使用 show ip eigrp topology active 命令确定活跃路由/被粘滞于活动状态路由

为了解决 EIGRP 被粘滞于活动状态的问题，必须回答两个问题：

- 这个路由为什么是活跃的？
- 这个路由为什么被粘滞于活动状态？

确定这个路由为什么是活跃的不是一个困难的任务。有时候，路由连续处于活跃状态可能是由于链路抖动。或者，如果这个路由是一个主机路由（/32 路由），那么有可能它是来自一个断开的拨号连接。但是，试图确定为什么活跃路由变成了被粘滞于活动状态是一个艰难得多的任务——而且学起来也更重要。一般情况下，活跃路由是由于下面原因之一变成了被粘滞于活动状态：

- 坏链路或者拥塞链路。
- 路由器资源少，比如路由器上的少量内存或者高 CPU 利用率。
- 查询范围长。
- 过多的冗余。

缺省情况下，被粘滞于活动状态计时器仅仅是三分钟。换句话说，如果 EIGRP 邻居没有在三分钟内收到查询的一个响应，那么邻居就会被重置。这样增加了疑难解析 EIGRP 被粘滞于活动状态的难度，因为每次当一个活跃路由被粘滞于活动状态时，只有三分钟的时间来捕获这一活跃路由查询路径并有希望找到其原因。

疑难解析 EIGRP 被粘滞于活动状态错误需要的工具是 **show ip eigrp topology active** 命令。这条命令显示了目前有哪些路由是活跃的；这些路由已经活跃多长时间；以及哪些邻居响应、哪些邻居没有响应这个查询。根据其输出，可以确定哪些邻居没有响应这个查询，而且可以通过转发到那些没有响应的邻居来追踪查询路径，从而找出查询的状态。例 7-9 显示了 **show ip eigrp topology active** 命令的样本输出。

例 7-9 show ip eigrp topology active 命令的样本输出

```
Router#show ip eigrp topology active
IP-EIGRP Topology Table for AS(1)/ID(10.1.4.2)
A 20.2.1.0/24, 1 successors, FD is Inaccessible, Q
  1 replies, active 00:01:43, query-origin: Successor Origin
    via 10.1.3.1 (Infinity/Infinity), Serial1/0
    via 10.1.4.1 (Infinity/Infinity), Serial1/1, serno 146
  Remaining replies:
    Via 10.1.5.2, r, Serial1/2
```

正如例 7-9 中输出所展示的，20.2.1.0 的路由处于活跃状态，而且已经活跃了 1 分 43 秒。query-origin 是 Successor Origin，这意味着这个路由的后继者向该路由器发送了查询。这时，它收到来自 10.1.3.1 的响应和 10.1.4.1 的响应；该响应是无穷大，这意味着这两个路由器也不知道路由 20.2.1.0。show ip eigrp topology active 命令最重要的输出是 Remaining replies: 部分。根据例 7-9 的输出，路由器表明来自接口 Serial 1/2 的邻居 10.1.5.2 没有响应这个查询。

为了进一步进行疑难解析，必须使用 Telnet 登录到 10.1.5.2 路由器并使用同一条命令 **show ip eigrp topology active** 查看其 EIGRP 活跃路由的状态。有时候，这个路由器没有在 Remaining replies: 部分列出没有响应查询的邻居。例 7-10 显示了 **show ip eigrp topology active** 的另一个输出。

例 7-10 show ip eigrp topology active 命令的另一个样本输出

```
Router#show ip eigrp topology active
IP-EIGRP Topology Table for AS(110)/ID(175.62.8.1)
A 11.11.11.0/24, 1 successors, FD is Inaccessible
  1 replies, active 00:02:06, query-origin: Successor Origin
    via 1.1.1.2 (Infinity/Infinity), r, Serial1/0, serno 171
    via 10.1.1.2 (Infinity/Infinity), Serial1/1, serno 173
```

在例 7-10 中, 与例 7-9 的输出的唯一不同是没有包含响应这个路由器的邻居列表。但是, 这并不表示所有的邻居都响应了这些查询。在例 7-10 中, 邻居 1.1.1.2 有一个紧接着地址 1.1.1.2 的 r。这同样表示这个邻居没有响应查询。换句话说, 路由器有两种方法来表示没有响应查询的邻居。一种方法是把它们列在 Remaining replies: 部分的下面; 另一种方法是让一个 r 紧接着邻居的接口 IP 地址。当使用 show ip eigrp topology active 命令的时候, 路由器可以使用这些方法的任意组合来表示仍然没有响应查询的邻居, 如例 7-11 中所示。

例 7-11 show ip eigrp topology active 命令的输出显示仍然没有响应这些查询的邻居的一个组合表示

```
Router#show ip eigrp topology active
IP-EIGRP Topology Table for AS(110)/ID(175.62.8.1)
A 11.11.11.0/24, 1 successors, FD is Inaccessible
  1 replies, active 00:02:06, query-origin: Successor Origin
    via 1.1.1.2 (Infinity/Infinity), r, Serial1/0, serno 171
    via 10.1.1.2 (Infinity/Infinity), Serial1/1, serno 173
  Remaining replies:
    via 10.1.5.2, r, Serial1/2
```

在例 7-11 中, 没有响应这些查询的邻居是 1.1.1.2 和 10.1.5.2。没有响应的邻居中只有一个 10.1.5.2 被列在 Remaining replies: 部分的下面; 另一个没有响应的邻居 1.1.1.2 和另一个响应的邻居列在了一起。总而言之, 在执行 show ip eigrp topology active 命令时, 要查找的最重要的部分是那些没有响应该查询的邻居。为了查找这样的邻居, 需要查找那些使 r 紧接着它们的接口 IP 地址的邻居。

3. 解决被粘滞于活动状态问题的方法学

解决 EIGRP 被粘滞于活动状态问题的方法和 show ip eigrp topology active 命令只有在这一问题正在发生的时候才有用。当被粘滞于活动状态事件结束, 网络稳定的时候, 再回溯这一问题并且找出其原因即使不是不可能的, 也是非常困难的。

图 7-11 显示了解决 EIGRP 被粘滞于活动状态问题的流程图。

考虑图 7-12 中显示的网络, 查看解决 EIGRP 被粘滞于活动状态问题的一个例子。

在图 7-12 中, 路由器 A 有一个到刚刚断开的网络 20.2.1.0/24 的以太网接口。路由器 A 没有一个可行的后继者作为可以切换的备份路由。路由器 A 没有别的选择, 只能把 20.2.1.0/24 路由置为活跃状态并查询其邻居路由器 B。注意在路由器 A 中 show ip eigrp topology active 命令的输出。20.2.1.0/24 路由已经活跃了 1 分 12 秒, 来自 Serial 0 的 10.1.1.2 作为没有响应的邻居被列出, 也就是路由器 B。下一步是使用 Telnet 登录到路由器 B, 查看在路由器 B 中该活跃路由的状态。图 7-13 通过执行命令 show ip eigrp topology active 显示了路由器 B 中该活跃路由的状态。

否
检查是否因为高
低内存或者路
坏的链路或者

确信该路由器
和低 CPU 利
CPU 利用率和
修正 EIGRP

Router A
IP-EIGRP
A 20.2.1
1 re
Rema

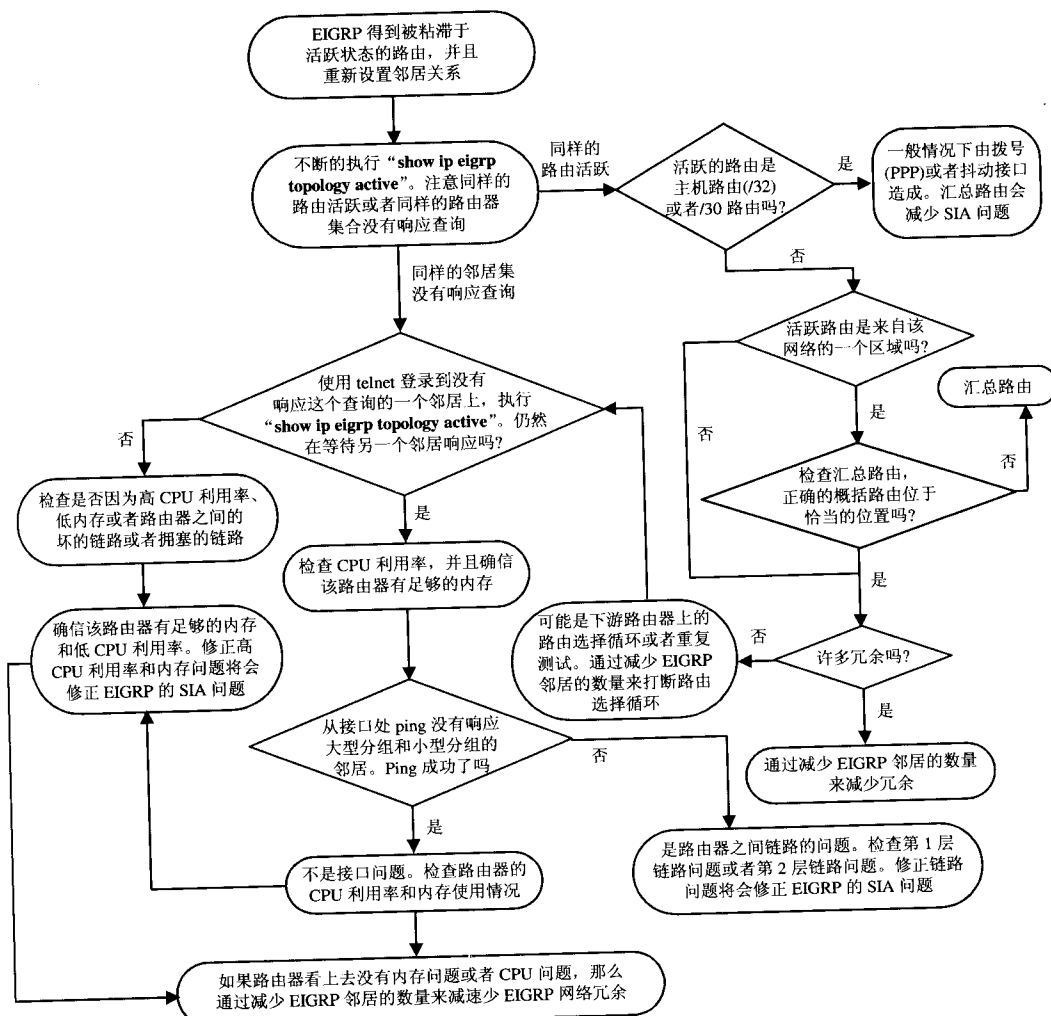


图 7-11 解决 EIGRP 被粘滞于活动状态问题的流程图

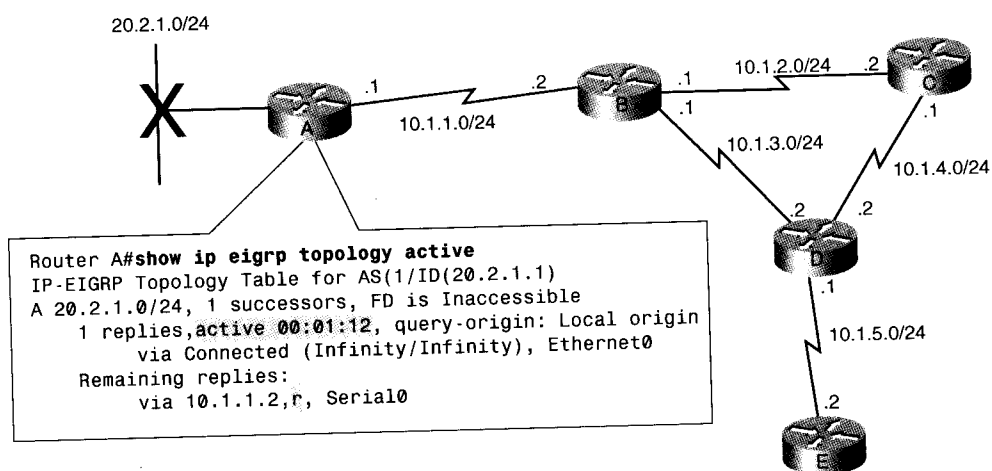


图 7-12 EIGRP 被粘滞于活动状态疑难解析例子的网络拓扑

在图 7-13 中, 路由器 B 上的命令 **show ip eigrp topology active** 表明路由 20.2.1.0/24 在路由器 B 中同样处于活跃状态, 而且它已经活跃了 1 分 23 秒。最重要的是, 对于路由 20.2.1.0/24, 路由器 B 不能向路由器 A 响应, 因为路由器 B 仍然在等待来自 Serial 1/2 的 IP 地址为 10.1.3.2 的邻居 (路由器 D) 对这个查询的响应。下一步是转到路由器 D 以查看活跃路由 20.2.1.0/24 的状态, 并查看为什么路由器 D 没有响应这个查询。图 7-14 显示了在路由器 D 上 **show ip eigrp topology active** 的输出。

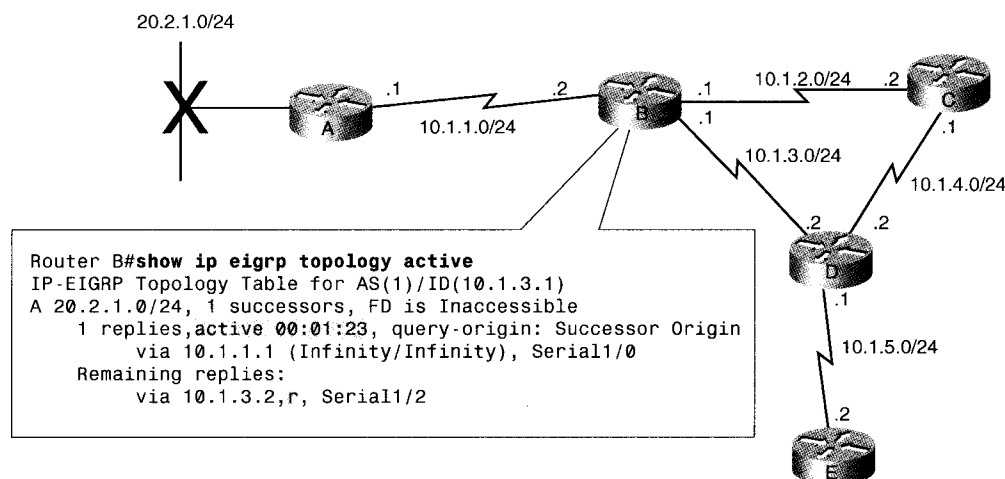


图 7-13 疑难解析 EIGRP 被粘滞于活动状态例子的路由器 B 上的活跃路由状态

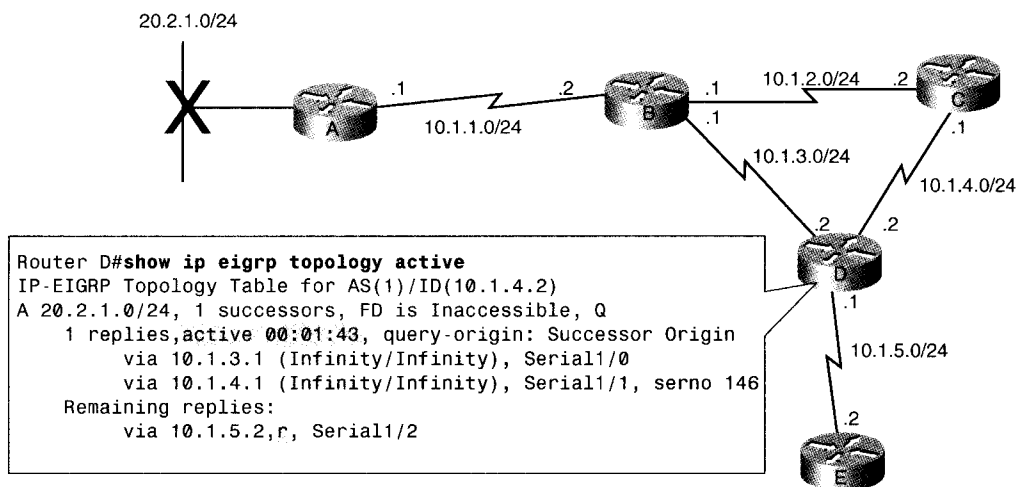


图 7-14 疑难解析 EIGRP 被粘滞于活动状态例子的路由器 D 上的活跃路由状态

路由器 D 也把路由 20.2.1.0/24 置为了活跃状态, 它已经处于活跃状态 1 分 43 秒。路由器 D 不能回答路由器 B 的查询, 因为路由器 D 正在等待来自 Serial 1/2 的 IP 地址为 10.1.5.2 的路由器 (路由器 E) 对这个查询的响应。于是再转到路由器 E 查看活跃路由 20.2.1.0/24 的状态以及为什么路由器 E 没有响应这个查询。图 7-15 显示了在路由器 E 上该活跃路由的状态。

20.2.1.0

X

Router
IP-EIG

show ip e

器 E 而言, 注

路由 20.2.1.0/

对于路由 20.2

状态。那么发

为了概括

1 对于路

2 路由器

3 路由器

4 最后,

何路由是活跃

根据这一

由器之间需要

查看路由

这两个路由器

了解这些邻居

例 7-12

RTRD#sh

IP-EIGR

H Addr

2 10.

1 10.

0 10.

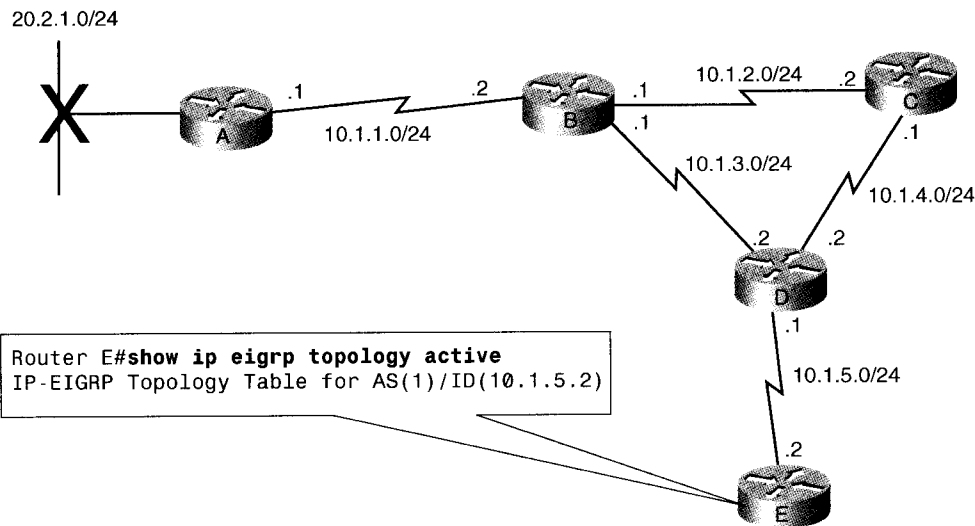


图 7-15 疑难解析 EIGRP 被粘滞于活动状态例子的路由器 E 上的活跃路由状态

show ip eigrp topology active 命令的输出没有显示路由器 E 的任何东西。这表明就路由器 E 而言，没有路由处于活跃状态。现在应该使用 Telnet 返回到路由器 D，再次检查对于路由 20.2.1.0/24 而言，这个路由器是否仍然处于活跃状态。使用 Telnet 返回到路由器 D 显示，对于路由 20.2.1.0/24，路由器 D 仍然处于活跃状态，但是路由器 E 没有任何路由处于活跃状态。那么发生了什么事？

为了概括到目前为止发生了哪些事情，在下面列出发生的事件链：

- 1 对于路由 20.2.1.0/24，路由器 A 是活跃的，而且正在等待路由器 B 响应这个查询。
- 2 路由器 B 不能响应，因为它正在等待路由器 D 的查询响应。
- 3 路由器 D 不能响应，因为它正在等待路由器 E 响应这个查询。

4 最后，在路由器 E 中的 **show ip eigrp topology active** 命令显示路由器 E 认为没有任何路由是活跃的，而返回到路由器 D 显示路由 20.2.1.0/24 仍然处于活跃状态。

根据这一系列事件可以看出在路由器 D 和路由器 E 之间明显有一个分歧。在这两个路由器之间需要更多的调查。

查看路由器 D 和路由器 E 的路由器 CPU 利用率和内存使用情况，没有显示任何问题。这两个路由器的 CPU 利用率和可获得的内存都是正常的。需要查看路由器 D 的邻居列表以了解这些邻居是否有问题。例 7-12 显示了路由器 D 的 EIGRP 邻居列表。

例 7-12 路由器 D 的 EIGRP 邻居列表

```

RTRD#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
  
```

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
				(sec)	(ms)		Cnt	Num
2	10.1.5.2	Se1/2	13	00:00:14	0	5000	1	0
1	10.1.3.1	Se1/0	13	01:22:54	227	1362	0	385
0	10.1.4.1	Se1/1	10	0124:08	182	1140	0	171

从例 7-12 中注意到在路由器 D 中使用 EIGRP 向 IP 地址为 10.1.5.2 的邻居（路由器 E）发送一个可靠的分组时有问题。Q 计数器是 1，连续几次成功执行 **show ip eigrp topology active** 命令显示 Q 计数器没有递减。

RTO 计数器处于它的最大值 5000 毫秒。这表明路由器 D 正在试图向路由器 E 发送一个可靠的分组，但是路由器 E 绝不会向路由器 D 确认该可靠分组。因为路由器 E 看上去没有高 CPU 利用率或者内存的问题，所以应该测试路由器 D 和路由器 E 之间链路的可靠性。现在从路由器 D 那里向 IP 地址 10.1.5.2（路由器 E 的串行接口）发送五个 ping 分组，以查看发生了什么事情。例 7-13 显示了该 ping 测试的结果。

例 7-13 从路由器 D 到路由器 E 的 ping 测试的结果

```
Router D#ping 10.1.5.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

例 7-13 中的 ping 测试显示成功率是百分之 0。这表明路由器 D 和路由器 E 之间存在链路问题。这一链路能够传递组播分组来建立 EIGRP 邻居关系，但是它在传送单播分组时有问题。因此，链路问题是这个例子中 EIGRP 被粘滞于活动状态问题的根本原因。解决这一 EIGRP 被粘滞于活动状态的方法是逐跳地追踪查询路径，找出在每一跳上活跃路由的状态。

对于解决 EIGRP 被粘滞于活动状态问题而言，上述过程是典型的疑难解析方法学。

有时候，逐跳追踪查询路径会导致循环，或者仅仅是因为有太多没有响应查询的邻居。在这种情况下，可以通过削减冗余来简化和降低 EIGRP 拓扑的复杂性。EIGRP 拓扑越简单，疑难解析 EIGRP 被粘滞于活动状态问题就越简单。

预防 EIGRP 被粘滞于活动状态问题的根本解决方法是无论何时，只要有可能就手动地汇总这些路由，以及拥有一个层次型的网络设计。EIGRP 汇总的网络越多，当一个主要的收敛行为发生时 EIGRP 必须做的工作就越少。因此，这样做减少了要发送的查询数目，最终减少了 EIGRP 被粘滞于活动状态错误的发生。图 7-16 显示了一个糟糕的网络设计的例子，它不能扩展为一个大型的 EIGRP 网络。

在图 7-16 中，每一个核心路由器都代表了这一完整网络的一个区域，并显示出在 IP 地址规划中没有层次性。Core 1 路由器正在把路由 1.1.1.0、3.3.4.0、1.1.2.0 和 2.2.3.0 插入到核心网络中。这些地址非常分散以致于手动汇总是不可能的。其他的核心路由器也正在经历同样的问题。Core 3 路由器和 Core 4 路由器不能把任何路由汇总到核心网络中。结果，如果 3.3.3.0 网络的 Ethernet 链路持续发生抖动，那么查询会一直行进到 Core 3 路由器，然后还会在 Core 1 区域和 Core 4 区域中被看到。最终，查询会经过内部网络中的所有路由器；这会急剧增加发生 EIGRP 被粘滞于活动状态问题的可能。最好的做法是重新制定一个 IP 地址规划。一个区域应该只采用一块 IP 地址；通过这种方法，这些核心路由器将能够把路由汇总到核心网络中，从而使核心网络中有一张简化的路由表：路由器和查询仅仅包含在一个区域中。图 7-17 显示了一个改进的并且可扩展性更好的 EIGRP 网络设计。



比较图 7-1

只采用 1.0.0.0 区域只采用 3.0 网络中。如果 Eth Core 3 区域中 EIGRP 网络而

有时候，E

从例 7-12 中注意到在路由器 D 中使用 EIGRP 向 IP 地址为 10.1.5.2 的邻居 (路由器 E) 发送一个可靠的分组时有问题。Q 计数器是 1, 连续几次成功执行 **show ip eigrp topology active** 命令显示 Q 计数器没有递减。

RTO 计数器处于它的最大值 5000 毫秒。这表明路由器 D 正在试图向路由器 E 发送一个可靠的分组, 但是路由器 E 绝不会向路由器 D 确认该可靠分组。因为路由器 E 看上去没有高 CPU 利用率或者内存的问题, 所以应该测试路由器 D 和路由器 E 之间链路的可靠性。现在从路由器 D 那里向 IP 地址 10.1.5.2 (路由器 E 的串行接口) 发送五个 ping 分组, 以查看发生了什么事情。例 7-13 显示了该 ping 测试的结果。

例 7-13 从路由器 D 到路由器 E 的 ping 测试的结果

```
Router D#ping 10.1.5.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

例 7-13 中的 ping 测试显示成功率是百分之 0。这表明路由器 D 和路由器 E 之间存在链路问题。这一链路能够传递组播分组来建立 EIGRP 邻居关系, 但是它在传送单播分组时有问题。因此, 链路问题是这个例子中 EIGRP 被粘滞于活动状态问题的根本原因。解决这一 EIGRP 被粘滞于活动状态的方法是逐跳地追踪查询路径, 找出在每一跳上活跃路由的状态。

对于解决 EIGRP 被粘滞于活动状态问题而言, 上述过程是典型的疑难解析方法学。

有时候, 逐跳追踪查询路径会导致循环, 或者仅仅是因为有太多没有响应查询的邻居。在这种情况下, 可以通过削减冗余来简化和降低 EIGRP 拓扑的复杂性。EIGRP 拓扑越简单, 疑难解析 EIGRP 被粘滞于活动状态问题就越简单。

预防 EIGRP 被粘滞于活动状态问题的根本解决方法是无论何时, 只要有可能就手动地汇总这些路由, 以及拥有一个层次型的网络设计。EIGRP 汇总的网络越多, 当一个主要的收敛行为发生时 EIGRP 必须做的工作就越少。因此, 这样做减少了要发送的查询数目, 最终减少了 EIGRP 被粘滞于活动状态错误的发生。图 7-16 显示了一个糟糕的网络设计的例子, 它不能扩展为一个大型的 EIGRP 网络。

在图 7-16 中, 每一个核心路由器都代表了这一完整网络的一个区域, 并显示出在 IP 地址规划中没有层次性。Core 1 路由器正在把路由 1.1.1.0、3.3.4.0、1.1.2.0 和 2.2.3.0 插入到核心网络中。这些地址非常分散以致于手动汇总是不可能的。其他的核心路由器也正在经历同样的问题。Core 3 路由器和 Core 4 路由器不能把任何路由汇总到核心网络中。结果, 如果 3.3.3.0 网络的 Ethernet 链路持续发生抖动, 那么查询会一直行进到 Core 3 路由器, 然后还会在 Core 1 区域和 Core 4 区域中被看到。最终, 查询会经过内部网络中的所有路由器; 这会急剧增加发生 EIGRP 被粘滞于活动状态问题的可能。最好的做法是重新制定一个 IP 地址规划。一个区域应该只采用一块 IP 地址; 通过这种方法, 这些核心路由器将能够把路由汇总到核心网络中, 从而使核心网络中有一张简化的路由表: 路由器和查询仅仅包含在一个区域中。图 7-17 显示了一个改进的并且可扩展性更好的 EIGRP 网络设计。



比较图 7-1

只采用 1.0.0.0 区域只采用 3.0 络中。如果 Eth Core 3 区域中 EIGRP 网络而

有时候, E

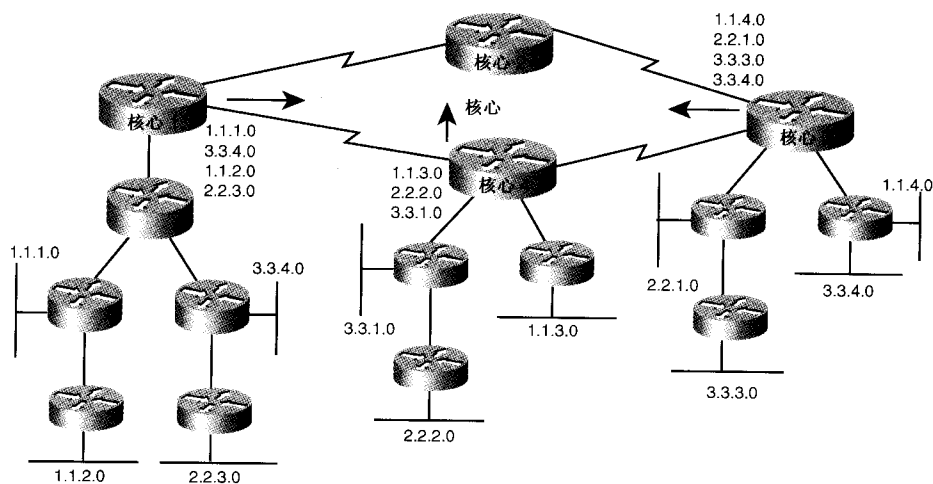


图 7-16 一个不可扩展的 EIGRP 网络的例子

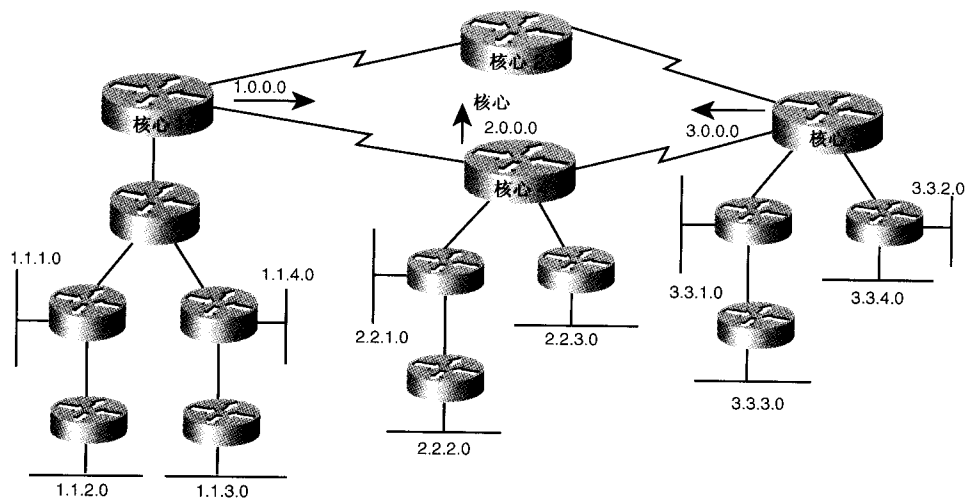


图 7-17 针对图 7-16 中网络进行的可扩展 EIGRP 网络设计的改进

比较图 7-16 和图 7-17，可以看出在图 7-17 中呈现的网络结构更好。Core 1 路由器区域只采用 1.0.0.0 的 IP 地址块，Core 4 路由器区域只采用 2.0.0.0 的 IP 地址块，而 Core 3 路由器区域只采用 3.0.0.0 的 IP 地址块。这使得这三个核心路由器可以把它们的路由汇总到核心网络中。如果 Ethernet 网络 3.3.3.0 在 Core 3 区域中发生抖动，那么这个查询将被限制到只在 Core 3 区域中进行，而不会行进到整个网络而影响网络中的其他路由器。对于一个大规模的 EIGRP 网络而言，汇总和层次性是最好的设计实践。

7.2 EIGRP 路由公布疑难解析

有时候，EIGRP 有路由公布的问题。本节讨论疑难解析 EIGRP 路由公布问题的方法，

这些方法可以分类为如下所示：

- 当网络管理员认为 EIGRP 应该向邻居公布路由时，EIGRP 没有这样做。
- 当网络管理员认为 EIGRP 不应该向邻居公布路由时，EIGRP 这样做了。
- EIGRP 使用网络管理员不理解的度量公布路由。

7.2.1 当网络管理员认为 EIGRP 应该向邻居公布路由时，EIGRP 没有这样做

本节讨论解决与 EIGRP 没有向邻居公布路由有关问题的方法。图 7-18 给出了如何疑难解析这一问题的流程图。

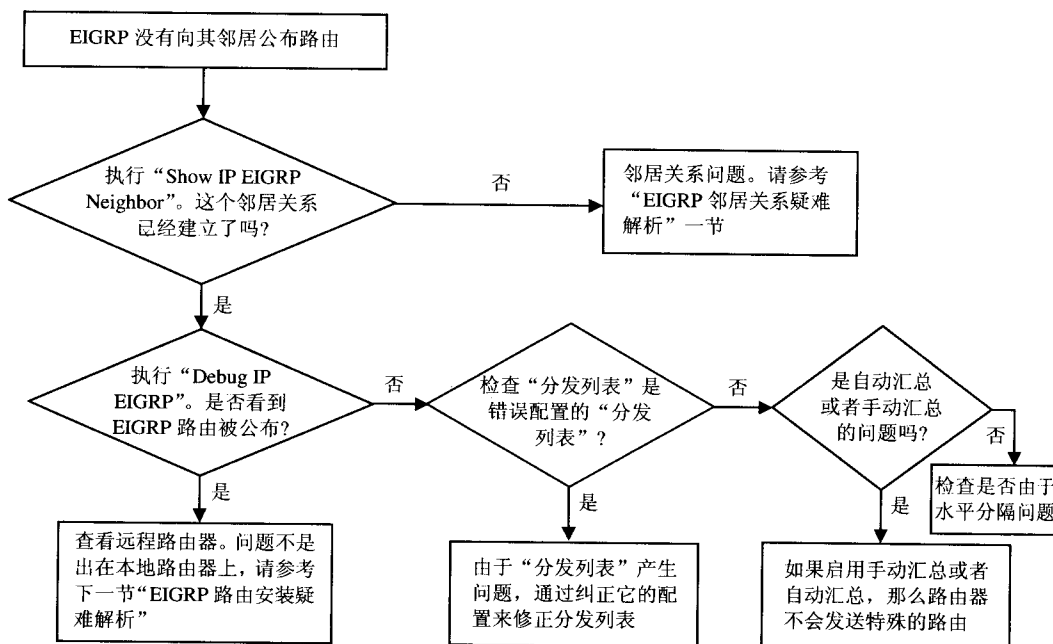


图 7-18 与 EIGRP 没有向其邻居公布路由相关的问题的疑难解析流程图

1. EIGRP 没有向邻居公布路由——原因：分发列表

图 7-19 显示了一个网络，其中 EIGRP 由于一个分发列表问题而没有向其邻居公布路由。例 7-14 显示了这个网络中的路由器 A 和路由器 B 的配置。

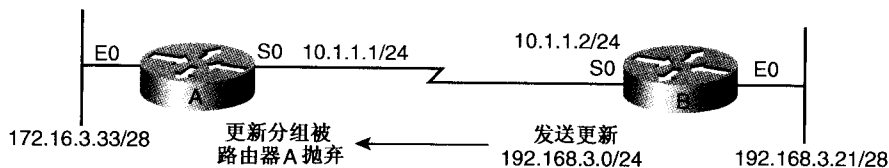


图 7-19 由于一个配置错误的分发列表，EIGRP 网络没有向其邻居公布路由

例 7-14

```

Router A
ip
interf
ip
router
ne
ne

```

```

Router B
ip
interf
ip
router
ne
ne
di
acc

```

问题在于
示了路由器 B

例 7-15

```

Router_B
IP-EIGRP

```

正如例 7-14 再次检查例 7-14 地配置了网络。没有包含在 p... 这一问题... 问列表改成允

2. EIGRP

使用图 7-19 动汇总或者通

例 7-16

```

Router A
ip a
interfac
ip a
router e
netw

```

例 7-14 图 7-19 中的路由器 A 和路由器 B 的配置

```
Router A# interface ethernet 0
      ip address 172.16.3.1 255.255.255.0
interface serial 0
      ip address 10.1.1.1 255.255.255.0
router eigrp 1
  network 172.16.0.0
  network 10.0.0.0
```

```
Router B# interface ethernet 0
      ip address 192.168.3.17 255.255.255.240
interface serial 0
      ip address 10.1.1.2 255.255.255.0
router eigrp 1
  network 192.168.3.0
  network 10.0.0.0
  distribute-list 1 out
  access-list 1 permit 192.168.3.160 0.0.0.15
```

问题在于路由器 A 没有收到来自路由器 B 的关于网络 192.168.3.16 的路由。例 7-15 显示了路由器 B 上的调试输出。

例 7-15 路由器 B 上 debug ip eigrp 命令的输出

```
Router B# debug ip eigrp

IP-EIGRP: 192.168.3.16/28 - denied by distribute list
```

正如例 7-15 中的输出所展示的，路由器 B 由于分发列表的配置而不会公布 192.168.3.16。再次检查例 7-14 中的配置，可以看到这一分发列表被关联到 access-list 1，而 access-list 1 错误地配置了网络号。access-list 1 应该允许 192.168.3.16 而不是 192.168.3.160。因为 192.168.3.16 没有包含在 **permit** 语句中，所以访问列表中隐式的 **deny** 命令阻止了网络 192.168.3.16 被公布。

这一问题的解决方法是把 access-list 1 改成允许 192.168.3.16 而不是 192.168.3.160。把访问列表改成允许 192.168.3.16 修正了这一问题。

2. EIGRP 没有向邻居公布路由——原因：非连续网络

使用图 7-19 中的网络图，EIGRP 没有公布该网络的另一个问题可能是在接口上配置的手动汇总或者通过一个主要网络边界的自动汇总，如例 7-16 中所示。

例 7-16 图 7-19 中的路由器 A 和路由器 B 的配置

```
Router A# interface ethernet 0
      ip address 192.168.3.33 255.255.255.240
interface serial 0
      ip address 10.1.1.1 255.255.255.0
router eigrp 1
  network 192.168.3.0
```

(待续)

, EIGRP

了如何疑难

检查是否由于
水平分隔问题

总或者
路由器
路由

邻居公布路由。

21/28

```

network 10.0.0.0

Router B# interface ethernet 0
ip address 192.168.3.21 255.255.255.240
interface serial 0
ip address 10.1.1.2 255.255.255.0
router eigrp 1
network 192.168.3.0
network 10.0.0.0

```

问题是路由器 A 没有收到来自路由器 B 的网络 192.168.3.16 的路由。例 7-17 显示了路由器 B 上的调试输出。

例 7-17 debug ip eigrp 命令在路由器 B 上的输出

```

Router B# debug ip eigrp

IP-EIGRP: 192.168.3.16/28 - don't advertise out Serial0
IP-EIGRP: 192.168.3.0/24 - do advertise out Serial0

```

根据调试信息，路由器 B 显示它没有公布 192.168.3.16/28 网络；而是仅仅向路由器 A 公布了主网 192.168.3.0/24。检查例 7-16 中路由器 A 和路由器 B 的配置，配置显示这两个路由器有一个非连续网络。路由器 A 在其 Ethernet 中有网络 192.168.3.32/28，而路由器 B 在其 Ethernet 中有另一个网络 192.168.3.16/28，它们被网络 10.1.1.0/24 分开。因此，当路由器 B 通过主网边界 10.1.1.0 公布网络 192.168.3.16/28 时，它只向路由器 A 公布了主网 192.168.3.0/24，而不是公布网络 192.168.3.16/28。当路由器 A 收到主网 192.168.3.0/24 时，它就没有把这一网络安装到拓扑表中，因为它已经在其 Ethernet 接口上有 192.168.3.0 网络了。

有两种解决这一非连续网络问题的方法。一种方法是在 **router eigrp** 下配置命令 **no auto-summary**。这条命令告诉 EIGRP 不要自动汇总主网边界。结果，路由器 B 的配置将如例 7-18 中所示。

例 7-18 在路由器 B 上禁用自动汇总以阻止非连续网络

```

Router B# router EIGRP 1
network 192.168.3.0
network 10.0.0.0
no auto-summary

```

第二种解决方法是把链路两端的串行接口的 IP 地址都改成 192.168.3.0 子网。作为一个例子，该串行 IP 地址可以采用 192.168.3.65/28 和 192.168.3.66/28。使用这种方法，路由器 B 就不会自动汇总这一路由，因为它没有通过一个主网边界。

3. EIGRP 没有向邻居公布路由——原因：水平分割问题

EIGRP 有它自己的 **split-horizon** 命令。这条命令配置在接口的下面，如下所示：

```
[no] ip split-horizon eigrp autonomous-system
```

关掉 IP 的水平分割并不能关掉 EIGRP 的水平分割。图 7-20 显示了一个容易出现水平分割问题的 EIGRP 网络。

例 7-19 显示

例 7-19 图

```

Router A#
ip add
interface
ip add
router eigrp
network

```

```

Router B#
ip add
interface
ip add
router eigrp
network

```

```

Router C#
ip add
interface
ip add
router eigrp
network

```

图 7-19 中显

中图 7-20 中的 h
hub 路由器使用
Ethernet 网络 1
172.16.3.0/24 的

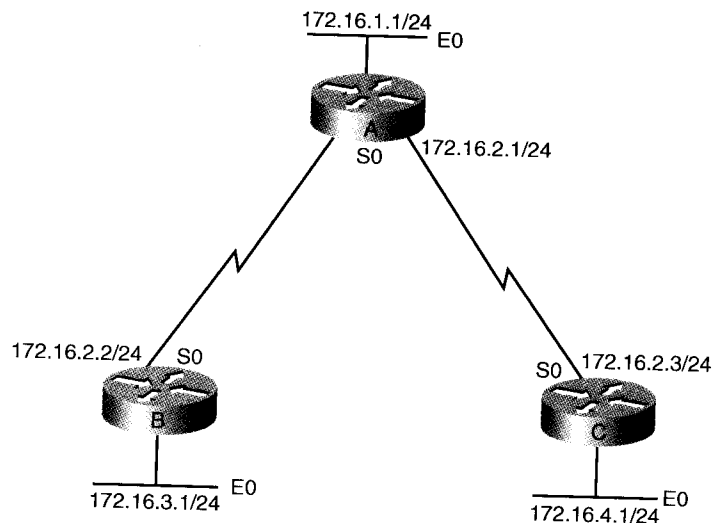


图 7-20 容易出现 EIGRP 水平分割问题的 EIGRP 网络

例 7-19 显示了图 7-20 中的 hub-and-spoke 网络中的路由器 A、B 和 C 的配置。

例 7-19 图 7-20 中的路由器 A、B 和 C 的配置

```
Router A# interface ethernet 0
      ip address 172.16.1.1 255.255.255.0
interface serial 0
      ip address 172.16.2.1 255.255.255.0
router eigrp 1
  network 172.16.0.0
```

```
Router B# interface ethernet 0
      ip address 172.16.3.1 255.255.255.0
interface serial 0
      ip address 172.16.2.2 255.255.255.0
router eigrp 1
  network 172.16.0.0
```

```
Router C# interface ethernet 0
      ip address 172.16.4.1 255.255.255.0
interface serial 0
      ip address 172.16.2.3 255.255.255.0
router eigrp 1
  network 172.16.0.0
```

图 7-19 中显示的一个常见网络环境是帧中继 (Frame Relay) 的 hub-and-spoke 设计，其中图 7-20 中的 hub 路由器 (路由器 A) 没有为每个远程的 spoke 站点都配置了子接口。结果，hub 路由器使用一个主接口连接这两个 spoke 站点。问题是路由器 B 没有收到路由器 C 的 Ethernet 网络 172.16.4.0/24 的路由，而路由器 C 没有收到路由器 B 的 Ethernet 网络 172.16.3.0/24 的路由。这一问题似乎出在 hub 站点上。hub 站点看到了所有的路由，但是它

没有从路由器 B 向路由器 C 传递这些路由，反之亦然。例 7-20 显示了该 hub 路由器（路由器 A）上的 debug 输出。

例 7-20 debug ip eigrp 命令在路由器 A 上的输出

```
Router A# debug ip eigrp
IP-EIGRP: 172.16.1.0/24 - do advertise out Serial0
IP-EIGRP: Processing incoming UPDATE packet
IP-EIGRP: Int 172.16.3.0/24
IP-EIGRP: Int 172.16.4.0/24
```

根据这一调试输出，可以看到 hub 路由器仅仅公布了 Serial 0 上的 172.16.1.0/24 路由。这一 hub 路由器收到了从路由器 B 到路由器 C 的 172.16.3.0/24 和 172.16.4.0/24 接口的路由。问题是 hub 路由器没有发送 Serial 0 上的所有路由。参考例 7-19 中的路由器 A、B 和 C 的配置，可以看到它们的串行接口全部在同一个子网中，但是它们没有在物理上被连接起来。因此，hub 路由器收到了从路由器 B 到路由器 C 的来自 Serial 0 的路由，但是不会在 Serial 0 上重新公布这些路由。这遵循了水平分割规则（路由信息一定不能离开那个通过其收到这一信息的路由器接口）。

为了解决 EIGRP 的水平分割问题，最容易的修正方法是关掉 EIGRP 的水平分割。例 7-21 显示了禁用水平分割的正确配置改动。

例 7-21 在 hub 路由器上禁用水平分割

```
Router A# interface ethernet 0
ip address 172.16.1.1 255.255.255.0
interface serial 0
ip address 172.16.2.1 255.255.255.0
no IP split-horizon EIGRP 1
router EIGRP 1
network 172.16.0.0
```

例 7-22 显示了这一配置改动之后路由器 A 上的调试输出。

例 7-22 验证禁用水平分割纠正了这个问题

```
Router A# debug ip eigrp
IP-EIGRP:172.16.1.0/24 - do advertise out Serial0
IP-EIGRP: 172.16.3.0/24 - do advertise out Serial0
IP-EIGRP: 172.16.4.0/24 - do advertise out Serial0
IP-EIGRP: Processing incoming UPDATE packet
IP-EIGRP: Int 172.16.3.0/24
IP-EIGRP: Int 172.16.4.0/24
```

现在 spoke 路由器 B 和 C 可以看到这些路由了。水平分割问题的另一种修正方法是配置 hub 路由器上的子接口，为每一个子接口都指派不同的 IP 地址子网。记住：对串行子接口的支持只对 WAN PVC 类型的连接才有效，比如 ATM 和帧中继。例 7-23 显示了为避免 EIGRP 的水平分割问题的这种设置的配置。

例 7-23

```
Router
ip
inter
de
ip ad
inter
de
ip
router
ne
```

```
Router
ip
inter
ip
router
ne
```

```
Router
ip
inter
ip
router
ne
```

当在路的连接。到现在是通过在 172.16.5.在两个不同和路由器 C

例 7-24

```
Router
IP-EIG
IP-EIG
IP-EIG
IP-EIG
IP-EIG
IP-EIG
```

通过路此的 LAN 挂

7.2.2 公布了

有时候

例 7-23 使用不同的 IP 地址子网配置子接口解决 EIGRP 的水平分割问题

```
Router A# interface ethernet 0
      ip address 172.16.1.1 255.255.255.0
interface serial 0.1 point-to-point
      description connection to router B
      ip address 172.16.2.1 255.255.255.0
interface serial 0.2 point-to-point
      description connection to router C
      ip address 172.16.5.1 255.255.255.0
router eigrp 1
  network 172.16.0.0
```

```
Router B# interface ethernet 0
      ip address 172.16.3.1 255.255.255.0
interface serial 0
      ip address 172.16.2.2 255.255.255.0
router eigrp 1
  network 172.16.0.0
```

```
Router C# interface ethernet 0
      ip address 172.16.4.1 255.255.255.0
interface serial 0
      ip address 172.16.5.2 255.255.255.0
router eigrp 1
  network 172.16.0.0
```

当在路由器 A 中配置子接口的时候,从逻辑上说,这隔离开了到路由器 B 和路由器 C 的连接。到路由器 B 和 C 的每个连接都有它们自己的网络。例如,从路由器 A 到 B 的连接现在是通过在 172.16.2.0/24 网络上的连接 Serial 0.1,而从路由器 A 到 C 的连接现在是通过在 172.16.5.0/24 网络上的连接 Serial 0.2。因为路由器 A 有两个到路由器 B 和路由器 C 的、在两个不同逻辑接口上的逻辑连接,所以没有应用水平分割规则,路由器 A 将向路由器 B 和路由器 C 公布所有路由,如例 7-24 中所示。

例 7-24 验证使用不同的子网配置子接口解决了水平分割问题

```
Router A# debug ip eigrp
IP-EIGRP: 172.16.1.0/24 - do advertise out Serial0.1
IP-EIGRP: 172.16.4.0/24 - do advertise out Serial0.1
IP-EIGRP: 172.16.5.0/24 - do advertise out Serial0.1
IP-EIGRP: 172.16.1.0/24 - do advertise out Serial0.2
IP-EIGRP: 172.16.2.0/24 - do advertise out Serial0.2
IP-EIGRP: 172.16.3.0/24 - do advertise out Serial0.2
```

通过路由器 A 向远程的路由器公布所有的路由,路由器 B 和路由器 C 现在可以到达彼此的 LAN 接口。

7.2.2 当网络管理员认为 EIGRP 不应该向邻居公布路由时, EIGRP 公布了

有时候, EIGRP 会向其邻居公布不期望的路由。请参见图 7-21 中疑难解析 EIGRP 不期

望的路由公布的流程图。

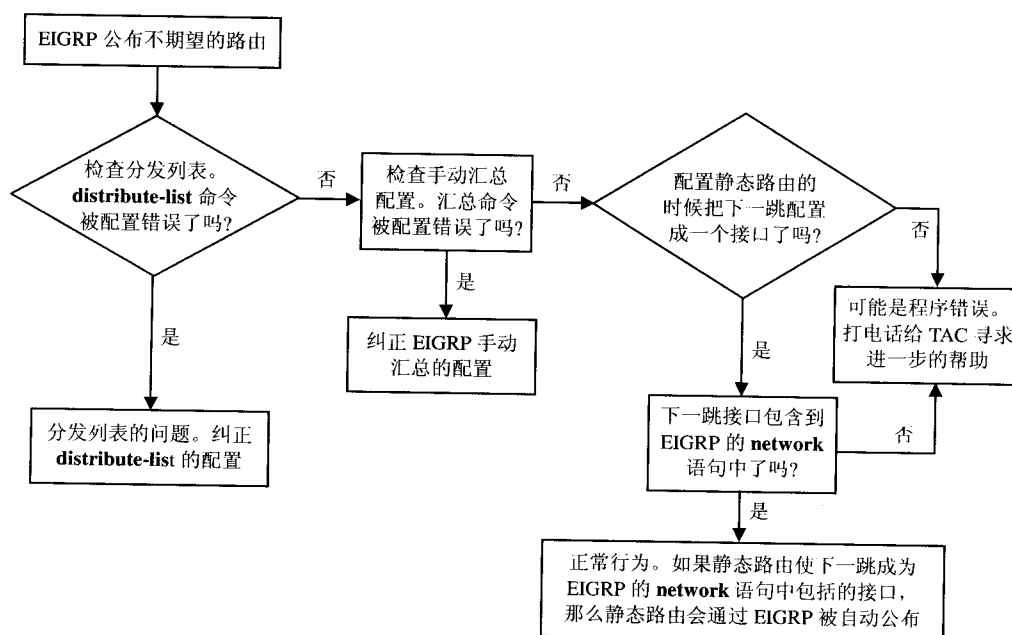


图 7-21 疑难解析 EIGRP 不期望的路由公布的流程图

参考图 7-19 中关于这个例子的网络图。例 7-25 显示了路由器 A 和路由器 B 的配置。

例 7-25 图 7-19 中示例的路由器 A 和路由器 B 的配置

```

Router A# interface ethernet 0
    ip address 172.16.3.1 255.255.255.0
interface serial 0
    ip address 10.1.1.1 255.255.255.0
router eigrp 1
    network 172.16.0.0
    network 10.0.0.0

Router B# interface ethernet 0
    ip address 192.168.130.1 255.255.255.0
interface serial 0
    ip address 10.1.1.2 255.255.255.0
router eigrp 1
    network 192.168.130.0
    network 10.0.0.0
ip route 192.168.1.0 255.255.255.0 ethernet 0
ip route 192.168.2.0 255.255.255.0 ethernet 0
ip route 192.168.3.0 255.255.255.0 ethernet 0
ip route 192.168.4.0 255.255.255.0 ethernet 0
:
ip route 192.168.127.0 255.255.255.0 ethernet 0
  
```

问题
那么路由
要的路由
口而配置
口。这些 E
因为 Ether
属于这一
态路由。

解决这
这些静态路
布这些静态
例 7-2
路由。

例 7-2

```

Router
ip
interface
ip
router
ne
ne
di
ip rou
ip rou
ip rou
ip rou
:
ip rou
access-
access-
  
```

这个分
192.168.127.0
送本例中不需
有向其他的 E

例 7-27

```

Router
IP-EIGR
IP-EIGR
IP-EIGR
  
```

问题是, 如果不在路由器 B 中的 **router eigrp** 命令下插入 **redistribute static** 命令, 那么路由器 B 会自动向路由器 A 重新分发所有已配置的 127 个静态路由。这将导致不必要的路由被非有意地向整个网络公布。造成这个问题的原因是这些静态路由是根据出口接口而配置的。在这种情况下, 路由器会认为所有静态路由都被直接连接到了 Ethernet 0 接口。这些 Ethernet 接口也通过 **network 192.168.130.0** 命令被包含在路由器 EIGRP 过程中。因为 Ethernet 0 被认为运行了 EIGRP, 所以所有通过静态路由连接到它的网络也都被认为属于这一 EIGRP 过程。于是即使没有配置 **redistribute static**, 路由器也公布所有这些静态路由。

解决这个问题的方法是配置一个阻止该路由器公布所有静态路由的分发列表, 或者是把这些静态路由改成引用下一跳的 IP 地址, 而不是一个接口。通过这种方法, 路由器就不会公布这些静态路由, 也不会使用不必要的路由来淹没整个网络。

例 7-26 显示了在路由器 B 上配置的分发列表, 它用于阻止发送不需要的重发布的静态路由。

例 7-26 路由器 B 上的配置, 通过配置分发列表阻止发送不需要的静态路由

```
Router B# interface ethernet 0
      ip address 192.168.130.1 255.255.255.0
!
interface serial 0
      ip address 10.1.1.2 255.255.255.0
router eigrp 1
      network 192.168.130.0
      network 10.0.0.0
      distribute-list 1 out
ip route 192.168.1.0 255.255.255.0 ethernet 0
ip route 192.168.2.0 255.255.255.0 ethernet 0
ip route 192.168.3.0 255.255.255.0 ethernet 0
ip route 192.168.4.0 255.255.255.0 ethernet 0
:
ip route 192.168.127.0 255.255.255.0 ethernet 0
access-list 1 deny 192.168.0.0 0.0.127.255
access-list 1 permit any
```

这个分发列表被关联到 access-list 1, 而 access-list 1 禁止发送从 192.168.0.0/24 到 192.168.127.0/24 范围内的任何路由, 并允许发送其他任何路由。这样, 分发列表就制止了发送本例中不需要的重发布的静态路由。例 7-27 中在路由器 B 上的调试输出表明该路由器没有向其他的 EIGRP 邻居发送这些静态路由, 这是因为配置了分发列表。

例 7-27 在路由器 B 上验证由于配置了一张分发列表所以没有发送静态路由

```
Router B# debug ip eigrp
IP-EIGRP: 192.168.1.0/24 - denied by distribute list
IP-EIGRP: 192.168.2.0/24 - denied by distribute list
IP-EIGRP: 192.168.3.0/24 - denied by distribute list
```

(待续)

```

IP-EIGRP: 192.168.4.0/24 - denied by distribute list
IP-EIGRP: 192.168.5.0/24 - denied by distribute list
IP-EIGRP: 192.168.6.0/24 - denied by distribute list

```

```

:

```

```

IP-EIGRP: 192.168.127.0/24 - denied by distribute list

```

解决这一问题的另一种方法是重新定义这些静态路由，使得该静态路由的下一跳是一个 IP 地址，而不是一个接口。例 7-28 显示了为修正这一问题而在路由器 B 中对静态路由配置的改动。

例 7-28 路由器 B 上的配置，通过使下一跳是一个 IP 地址而不是一个接口重新配置静态路由来阻止发送不需要的静态路由

```

Router B# interface ethernet 0
      ip address 192.168.130.1 255.255.255.0
!interface serial 0
      ip address 10.1.1.2 255.255.255.0
router eigrp 1
      network 192.168.130.0
      network 10.0.0.0
      distribute-list 1 out
ip route 192.168.1.0 255.255.255.0 192.168.130.2
ip route 192.168.2.0 255.255.255.0 192.168.130.2
ip route 192.168.3.0 255.255.255.0 192.168.130.2
ip route 192.168.4.0 255.255.255.0 192.168.130.2
:
ip route 192.168.127.0 255.255.255.0 192.168.130.2

```

7.2.3 EIGRP 使用了不期望的度量公布路由

EIGRP 不但可能向其邻居公布不期望的路由，还可能向其邻居公布不期望的度量。EIGRP 度量是 EIGRP 选择路由的基础，EIGRP 选择到目的网络具有最低 EIGRP 度量的路由。在路由器上发送或者接收一个不期望的 EIGRP 度量可能改变到目的网络的路由选择。最后的结果可能是非最优路由。图 7-22 给出了解决这个问题的流程图。

接下来的案例研究是无意中创建的一个偏移量列表（offset-list）的例子，它导致路由器按照一种非最优的方式路由分组。**offset-list** 命令向路由度量添加一个偏移量。这是管理某些路由的路由度量的一种方法，通过这种方式，改变某种特定路由协议选择路由的条件。图 7-23 示范了一个不期望的度量值问题的网络设置。

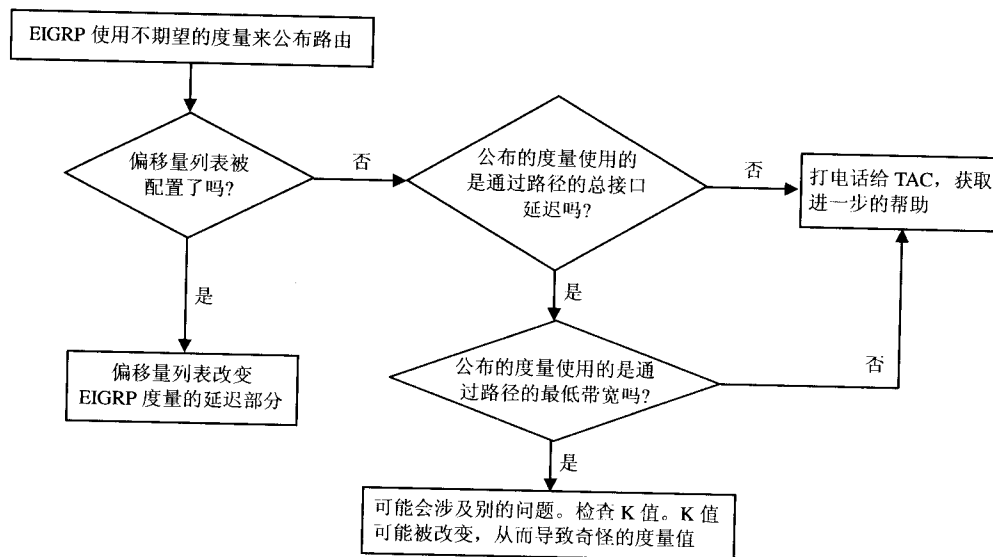


图 7-22 疑难解析 EIGRP 使用不期望的度量值公布路由的流程图

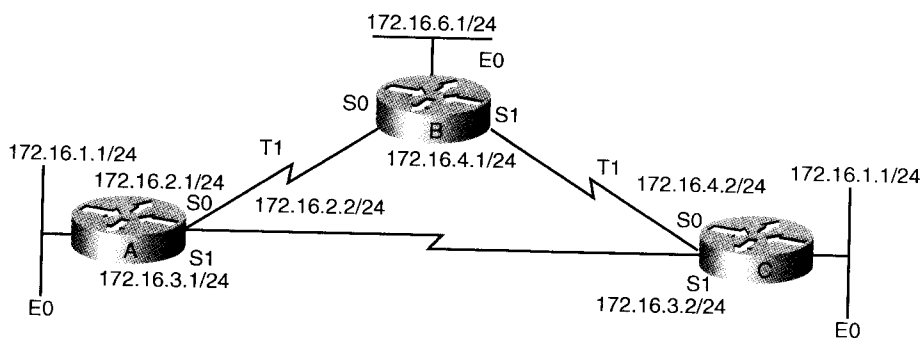


图 7-23 由于不期望的度量值而容易出现 EIGRP 公布问题的 EIGRP 网络

例 7-29 显示了在图 7-23 中显示的那个 EIGRP 网络中的路由器配置。

例 7-29 图 7-23 中路由器 A、B 和 C 的配置

```

Router A# interface ethernet 0
    ip address 172.16.1.1 255.255.255.0
interface serial 0
    ip address 172.16.2.1 255.255.255.0
interface serial 1
    ip address 172.16.3.1 255.255.255.0
router eigrp 1
    network 172.16.0.0
  
```

```

Router B# interface ethernet 0
    ip address 172.16.6.1 255.255.255.0
interface serial 0
    ip address 172.16.2.2 255.255.255.0
  
```

(待续)

```

interface serial 1
  ip address 172.16.4.1 255.255.255.0
router eigrp 1
  network 172.16.0.0

Router C# interface ethernet 0
  ip address 172.16.5.1 255.255.255.0
interface serial 0
  ip address 172.16.4.2 255.255.255.0
interface serial 1
  ip address 172.16.3.2 255.255.255.0
router eigrp 1
  network 172.16.0.0
  offset-list 1 out 600000 serial 1
access-list 1 permit 172.16.0.0 0.0.255.255

```

问题是路由器 A 没有采用到路由器 C 的直接路径到达路由器 C 的 Ethernet 网络 172.16.5.0/24。相反，路由器 A 采用了到路由器 B 然后到路由器 C 的路径。这就引入了额外的一跳。例 7-30 显示了路由器 A 上 172.16.5.0 255.255.255.0 的路由表和 EIGRP 拓扑表。

例 7-30 show ip route 和 show ip eigrp topology 命令的输出表明路由器 A 正在采用的到达路由器 C 的 172.16.5.0/24 Ethernet 网络的路由

```

Router_A#show ip route 172.16.5.0
Routing entry for 172.16.5.0/24
  Known via "eigrp 1", distance 90, metric 2707456, type internal
  Redistributing via eigrp 1
  Last update from 172.16.2.2 on Serial0, 01:08:13 ago
  Routing Descriptor Blocks:
    *172.16.2.2, from 172.16.2.2, 01:08:13 ago, via Serial0
      Route metric is 2707456, traffic share count is 1
      Total delay is 41000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2

Router_A#show ip eigrp topology 172.16.5.0 255.255.255.0 IP-EIGRP topology
entry for 172.16.5.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2707456
Routing Descriptor Blocks:
172.16.2.2 (Serial0), from 172.16.2.2, Send flag is 0x0
  Composite metric is (2707456 /2195456), Route is Internal
  Vector metric:
  Minimum bandwidth is 1544 Kbit
  Total delay is 41000 microseconds
  Reliability is 255/255
  Load is 1/255
  Minimum MTU is 1500

```

(待续)

```

Hop
172.16
Com
Vec
Min
Tot
Rel
Loa
Min
Hop

```

例 7-30 显示了比路由器 C 路径更多的延迟值的延迟值。对列表配置。

路由器 C 成问题的原因而导致路由器解决方法中所所示配置路

例 7-31

```

Router C
Router_C
Router_C

```

例 7-32 显示了表。

例 7-32 采用最优路由

```

Router_A
Routing
Known
Redis
Last
Routi
*172
Rou
Tot
Rel
Loa
Router A

```

```

Hop count is 2
172.16.3.2 (Serial1), from 172.16.3.2, Send flag is 0x0
Composite metric is (2795456 /281600), Route is Internal
Vector metric:
Minimum bandwidth is 1544 Kbit
Total delay is 44437 microseconds
Reliability is 255/255
Load is 1/255
Minimum MTU is 1500
Hop count is 1

```

例 7-30 表明路由器 A 选择路由器 B 作为到路由器 C 的下一跳, 因为路由器 B 有一个比路由器 C 更好的度量。仔细检查该拓扑表表明到路由器 C 的路径有比到路由器 B 的路径更多的延迟, 但是所有的链路都是 T1 链路。而路由器 C 的接口配置没有任何手动配置的延迟值。对路由器 C 中配置的更详细的检查揭示了路由器 C 在 **router eigrp** 下的偏移量列表配置。

路由器 C 的偏移量列表向 Serial 1 中的输出路由添加了一个 600,000 的度量。这才是造成问题的原因。当路由器 C 向路由器 A 发送路由时, 那些添加的偏移量增加了延迟值, 从而导致路由器 A 更愿意选择来自路由器 B 的路由。

解决方法是删除在路由器 C 上配置的偏移量列表。要删除该偏移量列表, 请按照例 7-31 中所示配置路由器 C。

例 7-31 从路由器 C 的配置中删除偏移量列表

```

Router C# config term
Router_C(config)#router eigrp 1
Router_C(config-router)#no offset-list 1 out 600000 serial 1

```

例 7-32 显示了删除路由器 C 上配置的偏移量列表之后的路由器 A 中的路由表和拓扑表。

例 7-32 **show ip route** 和 **show ip eigrp topology** 命令的输出验证了路由器 A 现在正在采用最优路由到达路由器 C 的 172.16.5.0/24 Ethernet 网络

```

Router_A#show ip route 172.16.5.0
Routing entry for 172.16.5.0/24
  Known via "eigrp 1", distance 90, metric 2195456, type internal
  Redistributing via eigrp 1
  Last update from 172.16.3.2 on Serial1, 00:08:23 ago
  Routing Descriptor Blocks:
    *172.16.3.2, from 172.16.3.2, 00:08:23 ago, via Serial1
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1

Router_A# show ip eigrp topology 172.16.5.0 255.255.255.0

```

(待续)

```

IP-EIGRP topology entry for 172.16.5.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2195456
Routing Descriptor Blocks:
172.16.3.2 (Serial1), from 172.16.3.2, Send flag is 0x0
  Composite metric is (2195456/281600), Route is Internal
  Vector metric:
    Minimum bandwidth is 1544 Kbit
    Total delay is 21000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1

172.16.2.2 (Serial1), from 172.16.2.2, Send flag is 0x0
  Composite metric is (2707456/2195456), Route is Internal
  Vector metric:
    Minimum bandwidth is 1544 Kbit
    Total delay is 41000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 2

```

例 7-32 中的输出显示 172.16.3.2 作为到路由器 C 的下一跳，这是到 172.16.5.0/24 网络的最优路径。还有，比较例 7-30 中和例 7-32 中显示的拓扑表，来自邻居 172.16.3.2 的 EIGRP 度量已经从 2,795,456 减少到 2,195,456。度量 600,000 的减少是删除这一偏移量列表的结果。正如这一案例研究所示，重要的是当异常行为发生时要认真检查配置。当使用 Cisco 的 TAC 打开一个案例时，要确保无论何时，只要有可能就提供路由器配置。

7.3 EIGRP 路由安装疑难解析

前一节讨论了当 EIGRP 向其邻居公布路由时 EIGRP 路由器出现的问题。本节讨论如何解决 EIGRP 没有在路由表中安装路由时出现的问题。引起这一问题的最常见原因如下所示：

- 自动配置或者手动配置的汇总。
- 更高的管理距离。
- 重复的路由器 ID。

接下来的几小节将详细讨论这一问题的原因以及如何解决它们。针对全面的疑难解析方法，图 7-24 显示了解决 EIGRP 路由安装问题的流程图。

7.3.1 EIGRP 没有安装路由——原因：自动汇总或者手动汇总

当 EIGRP 不能在其路由表中安装路由的时候，首先要检查的就是拓扑表。图 7-25 显示了用于这一案例研究的网络设置。

否
致电 TAC
进一步的

例 7-33 显



图 7-24 解决 EIGRP 路由安装问题的流程图

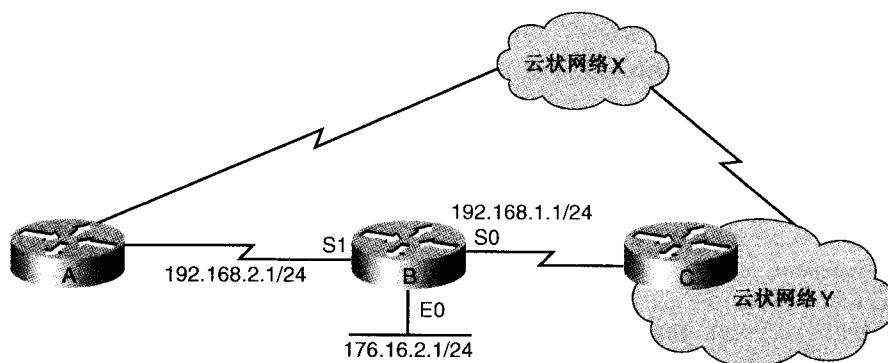


图 7-25 容易出现路由安装问题的 EIGRP 网络

例 7-33 显示了路由器 B 的配置。

例 7-33 图 7-25 中路由器 B 的配置

```
Router B# interface ethernet 0
      ip address 172.16.2.1 255.255.255.0
interface serial 0
      192.168.1.1 255.255.255.0
interface serial 1
      192.168.2.1 255.255.255.0
router eigrp 1
      network 172.16.0.0
      network 192.168.1.0
      network 192.168.2.0
```

图中云状网络 X 和 Y 是在 172.16.x.x 空间中的网络。问题是路由器 C 把所有的 172.16.x.x 网络都汇总到一个汇总路由 172.16.0.0/16 中了，并把它发送给了路由器 B。路由器 B 没有在其路由表中安装这些路由，如例 7-34 中所示。

例 7-34 路由器 B 的路由表

```
Router B# show ip route 172.16.0.0

Routing entry for 172.16.0.0/16
  Routing Descriptor Blocks:
    * directly connected, via Null 0
```

路由器 B 的路由表显示该路由被直接连接到 Null 0 而不是从路由器 C 中获得。路由器 B 中的拓扑表则表明它是从路由器 C 得到这些路由的，但是把这个路由作为已连接的路由安装，因为 Null 0 路由有一个值为 5 的距离，这是一个 EIGRP 汇总路由。路由器 B 的配置说明 EIGRP 由于自动汇总而汇总了 172.16.0.0/16 路由。每次发生自动汇总或者手动汇总的时候，EIGRP 都把汇总路由按照其下一跳是到 Null 0 的情况安装。这是一种用于 EIGRP 汇总路由的循环预防机制。在本案例研究中，下面是确切发生的事情：EIGRP 没有安装来自其汇总范围内的其邻居的一个路由。

基于这个原因，这一问题的解决方法更多的是一个设计问题。此网络中的两个位置一定没有向另一个发送同样的汇总路由。本例中，需要在路由器 B 上配置 **no auto-summary** 命令，从而允许路由器 B 接受来自路由器 C 的汇总路由。例 7-35 显示了为修正这一问题而在路由器 B 上做的配置改动。

例 7-35 为修正在图 7-25 中显示的问题而在路由器 B 上做的配置改动

```
Router B# interface ethernet 0
      ip address 172.16.2.1 255.255.255.0
interface serial 0
      192.168.1.1 255.255.255.0
interface serial 1
      192.168.2.1 255.255.255.0
router eigrp 1
      network 172.16.0.0
```

(待续)

随着路
路由器 C

例 7-3

```
Router B# show ip route 172.16.0.0
Routing entry for 172.16.0.0/16
  Routing Descriptor Blocks:
    * directly connected, via Null 0
```

7.3.2

请参考

150.150.0.0/16
到 150.150.0.0/16
路由距离(150)
安装 RIP 路由

例 7-37

```
Router B# show ip route 172.16.0.0
Routing entry for 172.16.0.0/16
  Routing Descriptor Blocks:
    * directly connected, via Null 0
```

```
network 192.168.1.0
network 192.168.2.0
no auto-summary
```

随着路由器 B 中配置的改动, 例 7-36 中显示的用于路由器 B 的路由表现在显示了来自路由器 C 的 172.16.0.0/16 的汇总路由。

例 7-36 路由器 B 的路由表现在显示了来自路由器 C 的汇总路由

```
Router_B#show ip route 172.16.0.0 255.255.0.0
Routing entry for 172.16.0.0/16
  Known via "eigrp 1", distance 90, metric 2195456, type internal
  Redistributing via eigrp 1
  Last update from 192.168.1.2 on Serial0, 00:16:24 ago
  Routing Descriptor Blocks:
    *192.168.1.2, from 192.168.1.2, 00:16:24 ago, via Serial0
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

7.3.2 EIGRP 没有安装路由——原因：更高的管理距离

请参考图 7-25 中的网络拓扑。如果云状网络 Y 向路由器 B 发送外部 EIGRP 路由 150.150.0.0/16, 而路由器 B 正在运行 RIP 和 EIGRP, 但是正在从路由器 A 的 RIP 域那里得到 150.150.0.0/16 路由, 那么类似问题的另一个变体就会发生。因为 RIP 有一个比外部 EIGRP 路由距离(170)更低的管理距离(120), 所以路由器 B 仅仅为来自路由器 A 的 150.150.0.0/16 安装 RIP 路由。例 7-37 显示了路由器 B 的 EIGRP 拓扑表。

例 7-37 路由器 B 用于 150.150.0.0/16 的 EIGRP 拓扑表

```
Router B# show ip eigrp topology 150.150.0.0 255.255.0.0

IP-EIGRP topology entry for 150.150.0.0/16
State is Passive, Query origin flag is 1, 0 Successor(s), FD is 4294967295
Routing Descriptor Blocks:
  192.168.1.2 (Serial0), from 192.168.1.2, Send flag is 0x0
  Composite metric is (2707456/2195456), Route is External
    Vector metric:
      Minimum bandwidth is 1544 Kbit
      Total delay is 41000 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 3
    External data:
      Originating router is 155.155.155.1
```

(待续)

(待续)

```
AS number of routes is 0
External protocol is OSPF, external metric is 64
Administrator tag is 0
```

EIGRP 拓扑表表明可行距离 (feasible distance, FD) 是不可访问的 (4294967295); 这一路由是一个外部路由, 它已经从 OSPF 中被重发布了。这表示路由器 B 正在接收来自路由器 C 的路由 150.150.0.0/16, 但是把该 FD 设置为不可访问的, 因为路由器 B 没有在这一路由表中使用 EIGRP 路由。事实上, 在路由器 B 中的路由表表项是一个用于 150.150.0.0/16 的 RIP 路由, 正如例 7-38 中所显示的。换句话说, 当在 EIGRP 拓扑表中该 FD 不可访问的时候, 路由器没有在其路由表中使用那个 EIGRP 路由。一般情况下, 这一路由会被另一个有更低管理距离的路由协议覆盖。

例 7-38 路由器 B 的路由表把 150.150.0.0/16 路由作为一个 RIP 路由显示

```
Router_B#show ip route 150.150.0.0
Routing entry for 150.150.0.0/16
  Known via "rip", distance 120, metric 5
  Redistributing via rip
  Last update from 192.168.2.2 on Serial1, 00:00:24 ago
  Routing Descriptor Blocks:
    *192.168.2.2, from 192.168.2.2, 00:00:24 ago, via Serial1
      Route metric is 5, traffic share count is 1
```

为了修正这个问题, 必须改变路由协议的管理距离, 从而使外部的 EIGRP 路由被优先选择。因此, 需要使用 distance 命令操纵路由协议的管理距离。例 7-39 显示了路由器 B 用于修正这一问题的配置改动。

例 7-39 为了修正由于更高管理距离造成的路由安装问题而在路由器 B 上所做的配置改动

```
Router B# interface ethernet 0
  ip address 172.16.2.1 255.255.255.0
interface serial 0
  192.168.1.1 255.255.255.0
interface serial 1
  192.168.2.1 255.255.255.0
router eigrp 1
  network 172.16.0.0
  network 192.168.1.0
  network 192.168.2.0
router rip
  network 172.16.0.0
  network 192.168.2.0
  distance 180 192.168.2.2 255.255.255.255
```

对于任何来自 192.168.2.2 的更新, 在例 7-39 中显示的 distance 命令都将其 RIP 管理距离设置为 180。这使得来自路由器 C 的外部 EIGRP 路由 (管理距离是 170) 可以比 RIP 路由优先选择。例 7-40 显示了这一结果。

例 7-40 路由器 B 的路由表现在显示来自路由器 C 的汇总路由

```

Router_B#show ip route 150.150.0.0
Routing entry for 150.150.0.0/16
  Known via "eigrp 1", distance 90, metric 2195456, type internal
  Redistributing via eigrp 1
  Last update from 192.168.1.2 on Serial0, 00:26:14 ago
  Routing Descriptor Blocks:
    *192.168.1.2, from 192.168.1.2, 00:26:14 ago, via Serial0
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1

```

7.3.3 EIGRP 没有安装路由——原因：重复的路由器 ID

经常会因为重复的路由器 ID 问题而导致 EIGRP 不安装路由。EIGRP 不像 OSPF 那样广泛地使用路由器 ID。EIGRP 仅仅在外部路由上使用路由器 ID 的概念以阻止循环。EIGRP 基于路由器上环回接口的最高 IP 地址选择路由器 ID。如果这个路由器没有任何环回接口，那么所有接口的最高活跃 IP 地址被选作 EIGRP 的路由器 ID。图 7-26 显示了针对 EIGRP 路由器 ID 的一个案例研究的网络设置。

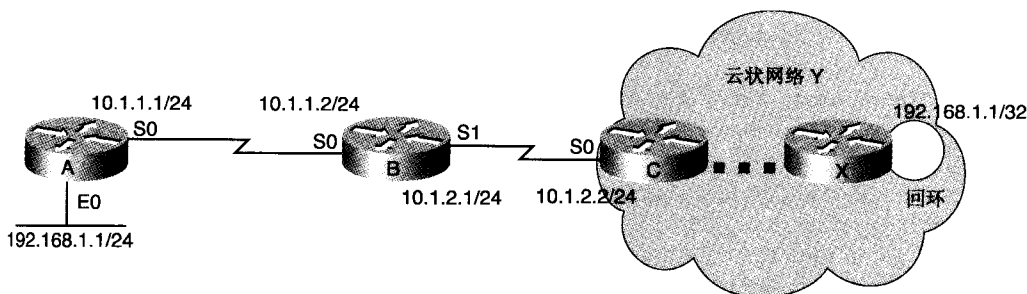


图 7-26 容易出现由于重复的路由器 ID 造成 EIGRP 没有安装路由的 EIGRP 网络

例 7-41 显示了引起这一问题相关配置。

例 7-41 图 7-26 中的路由器 A、B、C 和 X 的配置

```

Router A# interface ethernet 0
  ip address 192.168.1.1 255.255.255.0
interface serial 0
  ip address 10.1.1.1 255.255.255.0

```

```

Router B# interface serial 0
  IP address 10.1.1.2 255.255.255.0
interface serial 1

```

(待续)

IP address 10.1.2.1 255.255.255.0
Router C# interface serial 0 ip address 10.1.2.2 255.255.255.0
Router X# interface loopback 0 ip address 192.168.1.1 255.255.255.255

路由器 X 正在把来自 OSPF 的路由 150.150.0.0/16 重发布到 EIGRP 中, 并经过几跳向路由器 C 发送该路由。路由器 C 收到这个路由, 并把它作为 EIGRP 外部路由发送给路由器 B。路由器 B 把这个路由安装在其路由表中, 并把它发送给路由器 A。例 7-42 中的调试输出验证了路由器 B 是如何把该路由发送给路由器 A 的。

例 7-42 路由器 B 上 debug ip eigrp 命令的输出

Router B# debug ip eigrp
IP-EIGRP: 150.150.0.0/16 - do advertise out serial 0

问题是路由器 A 没有把路由 150.150.0.0/16 安装到路由表中。事实上, 路由器 A 没有在其拓扑表中显示路由 150.150.0.0/16。返回到路由器 B, 这一路由在其路由表中, 拓扑表如例 7-43 中所示。

例 7-43 路由器 B 上用于 150.150.0.0/16 的 EIGRP 拓扑表

Router B# show ip eigrp topology 150.150.0.0 255.255.0.0
IP-EIGRP topology entry for 150.150.0.0/16
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 3757056
Routing Descriptor Blocks:
10.1.2.2 (Serial1), from 10.1.2.2, Send flag is 0x0
Composite metric is (3757056/3245056), Route is External
Vector metric:
Minimum bandwidth is 1544 Kbit
Total delay is 82000 microseconds
Reliability is 255/255
Load is 1/255
Minimum MTU is 1500
Hop count is 7
External data:
Originating router is 192.168.1.1
AS number of routes is 0
External protocol is OSPF, external metric is 64
Administrator tag is 0

路由器 B 显示它正在从路由器 C 那里获取路由。通过检查外部数据网段, 注意到产生路由的路由器是 192.168.1.1, 它有七跳的距离。首先产生路由 150.150.0.0/16 的协议是 OSPF, 而度量是 64。注意产生路由的路由器是 192.168.1.1。返回来检查例 7-41 中路由器 A 的配置, 注意到路由器 A 也有一个在 Ethernet 0 上配置的 IP 地址 192.168.1.1, 它是这个路由器上最高的 IP 地址。这一证据充分说明 EIGRP 中的一个重复的路由器 ID 正是造成路由器 A 不安装

路由的原因。
收到来自路
器。在这种情
器 ID。路由
者, 并且从其
由环, 路由表
表中。

路由器
新分组中携带
有任何问题。

重复的路
器 A 中的 Et
IP 地址。把路

路由器
中所示。

例 7-44

Router
IP addr

例 7-45 拓扑表

Router_A
Routing
Known
Redis
Last
Routi
*10.1
Rout
Tota
Reli
Load

Router A
IP-EIGRP
State is
Routing
10.1.1.2
Composit
Vec
Min
Tot

路由的原因。因为路由器 X 和路由器 A 有同一个路由器 ID (192.168.1.1)，所以当路由器 A 收到来自路由器 B 的路由时，它检查这一路由的外部数据网段以了解谁是产生路由的路由器。在这种情况下，路由器 A 把 192.168.1.1 看作是产生路由的路由器，这是它自己的路由器 ID。路由器 A 没有把这一路由放到其拓扑表中，因为它认为自己是这一路由的原始生产者，并且从其他邻居那里重新收到这一路由，所以一定是一个循环。因此，为了阻止出现路由环，路由器 A 没有把路由 150.150.0.0/16 放到拓扑表中。所以，这一路由没有出现在路由表中。

路由器 A 不会安装任何起源于路由器 X 的外部路由，因为外部路由在它们的 EIGRP 更新分组中携带有它的路由器 ID。路由器 A 将安装来自路由器 X 的内部 EIGRP 路由，这没有任何问题。重复的路由器 ID 问题仅仅针对外部路由才发生。

重复的路由器 ID 问题的解决方法是改变路由器 X 的环回接口的 IP 地址，或者改变路由器 A 中的 Ethernet 0 的 IP 地址。经验规则是：决不要在一个网络中的两个位置上配置相同的 IP 地址。把路由器 X 的环回 IP 地址改成 192.168.9.1/32 可以修正这个问题（请参见例 7-44）。

路由器 X 中 IP 地址的改动的结果是路由器 A 中安装了路由 150.150.0.0/16，如例 7-45 中所示。

例 7-44 为了避免重复的路由器 ID 问题在路由器 X 中做的环回 IP 地址改动

```
Router X#interface Loopback 0
IP address 192.168.9.1 255.255.255.255
```

例 7-45 为了验证问题已修复，路由器 A 上关于 150.150.0.0/16 的路由表和 EIGRP 拓扑表

```
Router_A#show ip route 150.150.0.0
Routing entry for 150.150.0.0/16
  Known via "eigrp 1", distance 170, metric 4269056, type external
  Redistributing via eigrp 1
  Last update from 10.1.1.2 on Serial0, 00:06:14 ago
  Routing Descriptor Blocks:
    *10.1.1.2, from 10.1.1.2, 00:06:14 ago, via Serial0
      Route metric is 4269056, traffic share count is 1
      Total delay is 102000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 8
```

```
Router A# show ip eigrp topology 150.150.0.0 255.255.0.0
IP-EIGRP topology entry for 150.150.0.0/16
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 4269056
Routing Descriptor Blocks:
  10.1.1.2 (Serial0), from 10.1.1.2, Send flag is 0x0
  Composite metric is (4269056/3757056), Route is External
    Vector metric:
      Minimum bandwidth is 1544 Kbit
      Total delay is 102000 microseconds
```

(待续)

```

Reliability is 255/255
Load is 1/255
Minimum MTU is 1500
Hop count is 8
External data:
  Originating router is 192.168.9.1
  AS number of routes is 0
  External protocol is OSPF, external metric is 64
  Administrator tag is 0

```

7.4 EIGRP 路由抖动疑难解析

本节讨论如何疑难解析连续的 EIGRP 路由抖动。解决这一问题最重要的工具是 **show ip eigrp event** 命令。这条命令揭示了哪个邻居正在发送更新以及更新所使用的度量。请参见图 7-27 中解决 EIGRP 路由抖动问题的流程图。

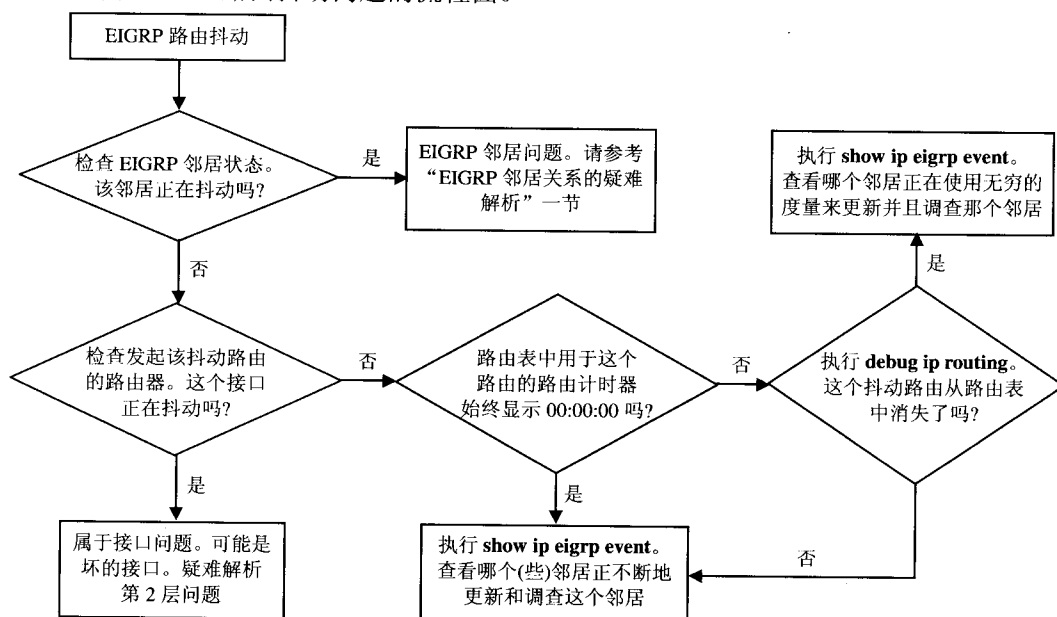


图 7-27 解决 EIGRP 路由抖动问题的流程图

解决 EIGRP 路由抖动问题时，在从路由表中消失的路由与路由表中显示 00:00:00 的路由计时器之间存在一个分歧，正如在例 7-46 中高亮显示的内容。

例 7-46 显示更新计时器始终处于 00:00:00 的路由表例子

```

Router A# show ip route 150.150.0.0

Routing entry for 150.150.0.0/16
Known via "eigrp 1", distance 90, metric 304128, type internal
Last update from 10.1.1.2 on Ethernet 0, 00:00:00 ago

```

当路由表中删除路由并重新更新。发送更新该路由器从它的证该路由器正在例 7-47 展示了在

例 7-47 del

```

Router B# d
RT: add 150
RT: delete

```

这一调试信息器而言是不可抵抗例如，如果希望只表，如例 7-48 中所

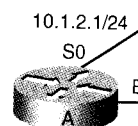
例 7-48 使用

```

Router B#deb
access-list
access-list

```

正如前面已经命令。缺省情况下有 500 行，这仅仅的极少内容，其中发送更新所使用的考虑图 7-28 中



192.168.1.1/24

当路由表中的路由计时器始终显示 00:00:00 的时候, 并不一定表示该路由器正在不断地删除路由并重新安装它。它仅仅表示这个路由器的一个邻居不断地向这一路由器发送此路由更新。发送更新路由的邻居不一定是到这一路由的最好路径, 不过是一条可能的路径。因为该路由器从它的一个邻居那里得到一个更新, 所以它就简单地刷新这一计时器。为了真正验证该路由器正在把这一路由从路由表中删除并重新安装它, 需要使用 **debug ip routing** 命令。例 7-47 展示了在路由器 B 上这条命令的输出。

例 7-47 debug ip routing 命令的输出验证一个路由是否正在被安装

```
Router B# debug ip routing
RT: add 150.150.0.0/16 via 10.1.1.2, eigrp metric [90/304128]
RT: delete route to 150.150.0.0 via 10.1.1.2, eigrp metric [90/304128]
```

这一调试信息显示该路由表删除和安装的所有路由, 虽然调试的输出可能对这些路由器而言是不可抵抗的。也可以把访问列表用于调试以使其输出只显示那些正被讨论的路由。例如, 如果希望只在路由表中的路由 192.168.1.0/24 上执行该调试, 那么就可以使用访问列表, 如例 7-48 中所配置的。

例 7-48 使用访问列表限制 debug ip routing 信息

```
Router B# debug ip routing 1
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 deny any
```

正如前面已经提到的, 用于疑难解析 EIGRP 路由抖动的最好工具是 **show ip eigrp event** 命令。缺省情况下, 该路由器保留一个关于所有 EIGRP 事件的日志。但是, 该日志的长度只有 500 行, 这仅仅涉及 EIGRP 事件的几百毫秒。**show ip eigrp event** 命令提供了 EIGRP 事件的极少内容, 其中包含那些正在使用已识别的路由向该路由器发送更新的邻居以及这些邻居发送更新所使用的度量。

考虑图 7-28 中显示的网络。

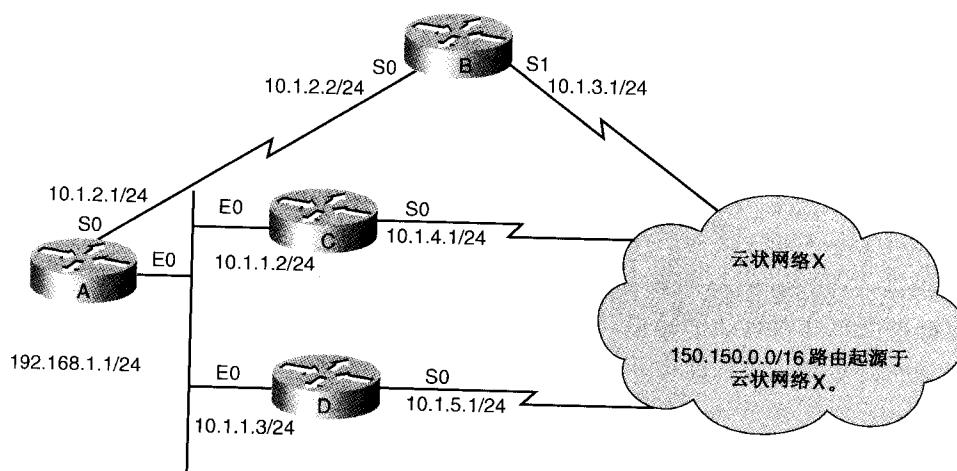


图 7-28 容易产生 EIGRP 路由抖动的 EIGRP 网络

在图 7-28 中, 云状网络 X 中的路由 150.150.0.0/16 从路由器 B、C 和 D 传递到路由器 A。路由器 A 选择路由器 C 作为到网络 150.150.0.0/16 的下一跳, 并把路由器 B 和路由器 D 设置为网络 150.150.0.0/16 的可行后继者。例 7-49 显示了这四个路由器的相关配置。

例 7-49 图 7-28 中的路由器 A、B、C 和 D 的配置

Router A# interface ethernet 0 ip address 10.1.1.1 255.255.255.0 interface serial 0 ip address 10.1.2.1 255.255.255.0
Router B# interface serial 0 ip address 10.1.2.2 255.255.255.0 interface serial 1 ip address 10.1.3.1 255.255.255.0
Router C# interface ethernet 0 ip address 10.1.1.2 255.255.255.0 interface serial 0 ip address 10.1.4.1 255.255.255.0
Router D# interface ethernet 0 ip address 10.1.1.3 255.255.255.0 interface serial 0 ip address 10.1.5.1 255.255.255.0

问题发生在路由器 A 上, 其用于路由表中路由 150.150.0.0/16 的路由计时器始终是 00:00:00。通过检查路由器 C (这一路由的下一跳), 可以看到这一路由是稳定的, 并没有抖动。路由器 A 的邻居关系也是稳定的, 路由器 A 上的接口也是稳定的, 没有任何接口抖动的迹象。于是下一步就检查 EIGRP 中的事件日志, 查看哪个邻居正在不断地向路由器 A 发送更新路由 150.150.0.0/16。例 7-50 显示了路由器 A 上 EIGRP 事件日志中的相关信息。

例 7-50 路由器 A 上 show ip eigrp event 命令的输出

Router A# show ip eigrp event
20:47:13.2 Rcv update dest/nh: 150.150.0.0/16 10.1.1.3
20:47:13.2 Metric set: 150.150.0.0/16 4872198
20:47:13.2 Rcv update dest/nh: 150.150.0.0/16 10.1.1.3
20:47:13.2 Metric set: 150.150.0.0/16 4872198

在事件日志中存在其他的输出, 但是这里只显示了最重要的几行。为了确信该路由器正在不断地执行更新, 必须连续几次执行 **show ip eigrp event** 命令。检查位于该输出左边的计时器是否在不断地变化。如果该计时器正在不断地变化, 那么表明 EIGRP 过程正在不断计算。从上而下阅读 EIGRP 事件日志, 最近的事件位于列表的顶部, 最早的事件位于列表的底部。例 7-50 中的事件日志显示路由器 A 不断地从 10.1.1.3 (路由器 D) 那里

得到路
中的路
计时器
掉任何分

根据
路由器
是在路由
环导致路
跳地到每

另一
维持了从
如果

是路由器
这些可能
ISO 版本
TAC 以确

本例
上, 而是
刷新其计
ISO 软件
连续执行
实了 EIGRP

例 7-5

Route
Route
Known
Receiv
Last
Route
*10
R
T
R
L

在一个
题的几个武
汇总问题的

和 D 传递到路
并把路由器 B
四个路由器的

得到路由 150.150.0.0/16 的更新。注意更新路由器 A 的下一跳路由器没有重置路由器 A 中的路由计时器。任何针对一个路由更新路由器的可行后继者都会重置该路由器上的路由计时器。因此, 这些路由计时器被重置, 但是该路由被保留在路由表中以使路由器不会丢掉任何分组。

根据 EIGRP 事件日志, 正是路由器 D 不断地向路由器 A 发送更新信息。下一步行进到路由器 D, 以调查为什么它使用更新信息不断更新路由器 A。更新不断发生的一个可能原因是在路由器 D 中有一个针对 150.150.0.0/16 路由、包括网络 X 中其他路由器的路由环, 路由环导致路由被相互发送。如果在网络中发生一个路由环, 那么就需要一张目前的网络图来逐跳地到每个路由器追踪这个路由环。

另一种可能性是在路由器 A 的 Ethernet 0 上的 LAN 交换机可能有一个生成树问题, 它维持了从路由器 D 到路由器 A 的分组循环。

如果在这一网络中没有路由环, 在这个交换机上也没有生成树问题, 那么另一种可能就是路由器 D 可能碰到了 EIGRP 程序错误, 使它没有任何原因地不断向路由器 A 发送更新。这些可能的程序错误之一是 CSCdt15109, 它使路由器不断地发送没有变化的更新信息。Cisco ISO 版本 12.1.7 及其后续版本将针对这一问题修正程序错误; 但是, 我们始终推荐咨询 Cisco TAC 以确定这一问题是否真是由于软件程序错误而引起的。

本例中, 路由器 D 碰到了前面提及的那个软件程序错误。注意, 问题不是在路由器 A 上, 而是在路由器 D 上。路由器 D 不断地向路由器 A 发送更新信息, 而路由器 A 不断地刷新其计时器。路由器 A 仅仅是由路由器 D 引起的问题的一个结果。在路由器 D 上的 Cisco ISO 软件更新之后, 路由器 A 就停止刷新其路由表计时器, 正如例 7-51 中所展示的。还有, 连续执行 **show ip eigrp event** 数次也会显示这一事件表上的那些计时器并没有改变。这也证实了 EIGRP 过程是稳定的, 没有从其邻居那里收到不必要的更新。

例 7-51 路由器 A 上路由表的输出证实该问题已被修正

```
Router_A#show ip route 150.150.0.0
Routing entry for 150.150.0.0/16
  Known via "eigrp 1", distance 90, metric 4269056, type internal
  Redistributing via eigrp 1
  Last update from 10.1.1.2 on ethernet 0, 00:03:18 ago
  Routing Descriptor Blocks:
    *10.1.1.2, from 10.1.1.2, 00:03:18 ago, via ethernet0
      Route metric is 4269056, traffic share count is 1
      Total delay is 102000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 4
```

且计时器始终是
稳定的, 并没
的, 没有任何接
正在不断地向路
事件日志中的相

了确信该路由器
位于该输出左边
EIGRP 过程正在
最早的事件位
路由器 D) 那里

7.5 EIGRP 路由汇总疑难解析

在一个设计良好的 EIGRP 网络中, 汇总是非常重要的。汇总是预防被粘滞于活动状态问题的几个武器之一。大多数汇总问题都是路由器配置错误的结果。图 7-29 给出了解决 EIGRP 汇总问题的流程图。

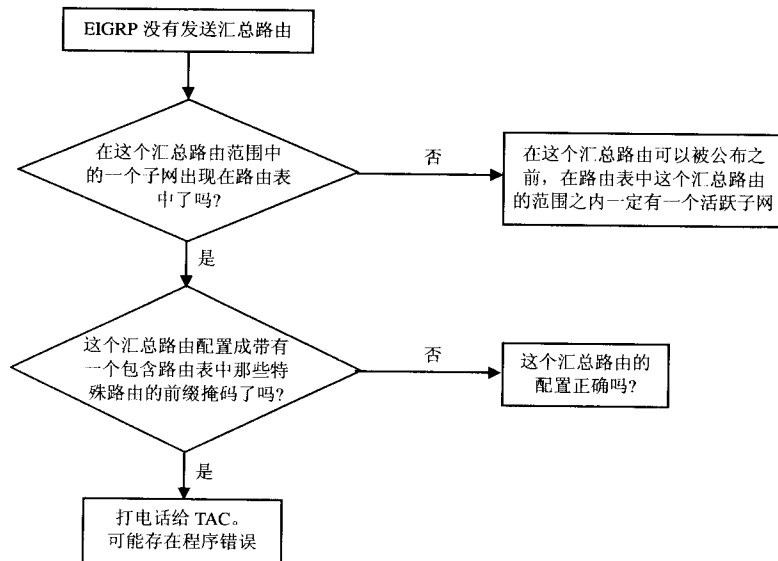


图 7-29 解决 EIGRP 汇总路由问题的流程图

7.5.1 EIGRP 汇总路由问题——原因：汇总路由的子网在路由表中不存在

考虑图 7-30 中显示的情况，路由器 A 被配置为向路由器 B 发送其 Ethernet 0 接口上的一个汇总路由 172.16.80.0 255.255.240.0。例 7-52 显示了路由器 A 的配置。但是，下一跳路由器没有看到这一路由，而且 172.16.80.0 255.255.240.0 路由没有在路由器 A 的拓扑表中。例 7-53 显示了它的路由表的一个快照。

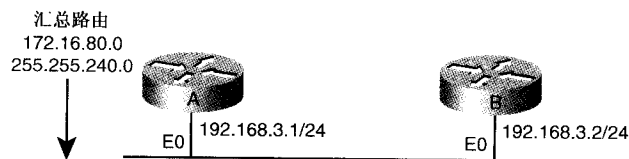


图 7-30 用于案例研究的 EIGRP 汇总路由问题的网络图

例 7-52 在图 7-30 中显示的例子中的路由器 A 的配置

```
Router_A#interface ethernet 0
ip address 192.168.3.1 255.255.255.0
ip summary-address EIGRP 1 172.16.80.0 255.255.240.0
interface Serial 0
ip address 192.168.1.2 255.255.255.0
interface Serial 1
ip address 192.168.2.2 255.255.255.0
router EIGRP 1
network 192.168.1.0
network 192.168.2.0
network 192.168.3.0
```

例 7-53 路由表

```
Router A#
C 192.168.1.0/24
C 192.168.2.0/24
C 192.168.3.0/24
D 172.16.80.0/20
D 172.16.80.0/20
D 172.16.80.0/20
D 172.16.80.0/20
D 172.16.80.0/20
D 172.16.80.0/20
D 172.16.80.0/20
D 172.16.80.0/20
```

在例 7-52

255.255.240.0 把

172.16.80.0 到 17

172.16.80.0 到 17

表中，那么路由

这一问题的解

地址 172.16.81.1

例 7-54 显示了在

例 7-54 为

```
Router_A#in
ip address
interface B
ip address
ip Summary-
interface S
ip address
Interface S
ip address
router EIGR
network 172
network 192
network 192
network 192
```

在这一配置改
总路由，如例 7-5

例 7-55 路由

```
Router A# sh
C 192.168.1.0/24
C 192.168.2.0/24
```

例 7-53 路由表快照

```
Router A# show ip route
C    192.168.1.0/24 is directly connected, Serial 0
C    192.168.2.0/24 is directly connected, Serial 1
C    192.168.3.0/24 is directly connected, Ethernet 0
D    172.16.99.0/24 [90/409600] via 192.168.1.1, Serial 0
D    172.16.97.0/24 [90/409600] via 192.168.1.1, Serial 0
D    172.16.79.0/24 [90/409600] via 192.168.1.1, Serial 0
D    172.16.70.0/24 [90/409600] via 192.168.1.1, Serial 0
D    172.16.103.0/24 [90/409600] via 192.168.1.1, Serial 0
D    172.16.76.0/24 [90/409600] via 192.168.1.1, Serial 0
D    172.16.98.0/24 [90/409600] via 192.168.1.1, Serial 0
```

在例 7-52 中显示的配置中, 通过使用命令 **ip summary-address eigrp 172.16.80.0 255.255.240.0** 把这个汇总路由配置成 172.16.80.0 255.255.240.0。这一汇总路由包含了范围从 172.16.80.0 到 172.16.95.255 的网络地址。根据例 7-53 中显示的路由表, 注意到没有路由在 172.16.80.0 到 172.16.95.255 之间。因此, 如果这一配置过的汇总路由的子网没有出现在路由表中, 那么路由器就不会生成汇总路由。

这一问题的解决方法是配置一个位于 172.16.80.0 255.255.240.0 范围内的接口。可以使用地址 172.16.81.1 255.255.255.0 配置一个环回接口, 从而产生在 Ethernet 0 上配置的汇总路由。例 7-54 显示了在路由器 A 中改动后的配置, 它将修正这一手动汇总问题。

例 7-54 为了修正手动汇总问题路由器 A 改动后的配置

```
Router_A#interface loopback 0
ip address 172.16.81.1 255.255.255.0
interface Ethernet 0
ip address 192.168.3.1 255.255.255.0
ip summary-address eigrp 1 172.16.80.0 255.255.240.0
interface Serial 0
ip address 192.168.1.2 255.255.255.0
Interface Serial 1
ip address 192.168.2.2 255.255.255.0
router EIGRP 1
network 172.16.0.0
network 192.168.1.0
network 192.168.2.0
network 192.168.3.0
```

在这一配置改动之后, 路由器 A 上的路由表显示了 172.16.80.0 255.255.240.0 的手动汇总路由, 如例 7-55 中所示。

例 7-55 路由器 A 在配置改动之后的路由表快照验证问题已被修复

```
Router A# show ip route
C    192.168.1.0/24 is directly connected, Serial 0
C    192.168.2.0/24 is directly connected, Serial 1
```

(待续)

```

C 192.168.3.0/24 is directly connected, Ethernet 0
C 172.16.81.1/24 is directly connected, Loopback 0
D 172.16.99.0/24 [90/409600] via 192.168.1.1, Serial 0
D 172.16.97.0/24 [90/409600] via 192.168.1.1, Serial 0
D 172.16.79.0/24 [90/409600] via 192.168.1.1, Serial 0
D 172.16.70.0/24 [90/409600] via 192.168.1.1, Serial 0
D 172.16.103.0/24 [90/409600] via 192.168.1.1, Serial 0
D 172.16.76.0/24 [90/409600] via 192.168.1.1, Serial 0
D 172.16.80.0/20 is a summary, 00:03:24, Null 0
D 172.16.98.0/24 [90/409600] via 192.168.1.1, Serial 0

```

7.5.2 EIGRP 汇总路由问题——原因：太多的汇总

另一个 EIGRP 汇总路由问题起源于当汇总路由包含了比所存在的子网更多的子网时。图 7-31 显示了针对这一案例研究可以引用的网络图。

如图 7-31 中所示，路由器 B 被连接到带有 172.16.1.0/24 到 172.16.15.0/24 网络的云状网络。路由器 B 把那些网络汇总到一个大型汇总路由 172.16.0.0/16 中，并把它发送给路由器 A。路由器 A 被连接到核心网络，路由器 A 向路由器 B 发送一个缺省路由 0.0.0.0 0.0.0.0。当核心网络中的一个设备试图到达 172.16.40.0/24 网络（这在该网络中是不存在的）时，问题出现了。当核心网络中的这个设备试图使用 **ping** 或者 **traceroute** 到达 172.16.40.0 网络时，这些分组就在路由器 A 和路由器 B 之间反复循环。

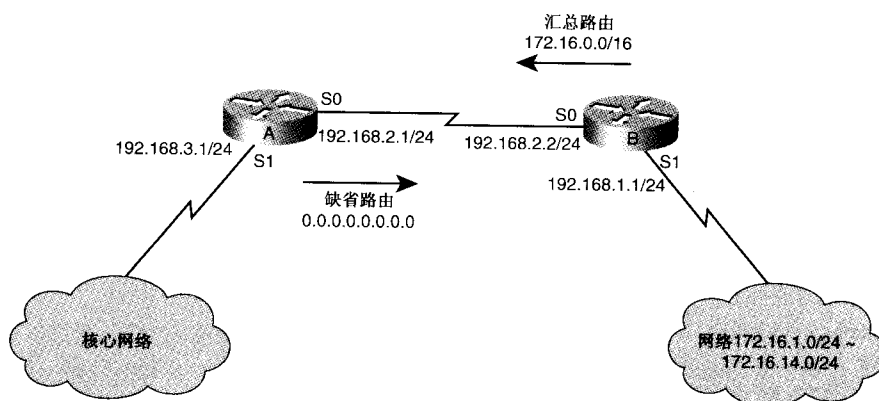


图 7-31 EIGRP 网络图——太多的 IP 地址汇总

例 7-56 显示了路由器 A 用于 172.16.40.0 的路由表。

例 7-56 路由器 A 用于 172.16.40.0 的路由表

```

Router A# show ip route 172.16.40.0

Routing entry for 172.16.0.0/16
  Known via "EIGRP 1", distance 90, metric 409600, type internal
  Last update from 192.168.2.2 on Serial0, 00:20:25 ago
  Routing Descriptor Blocks:

```

(待续)

```
* 192.168.2.2 from 192.168.2.2, 00:20:25 ago, via Serial 0
Route metric is 409600, traffic share count is 1
Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 1
```

路由器 A 的路由表表项显示了来自路由器 B 的汇总路由 172.16.0.0/16。因此，路由器 A 向路由器 B 转发这一分组。但是，因为路由器 B 没有用于 172.16.40.0 的路由，所以路由器 B 把这一分组直接发送回路由器 A；路由器 B 只有指回路由器 A 的缺省路由。对于在 172.16.0.0/16 范围中的任何不存在的网络而言，都会导致路由器 A 和路由器 B 之间出现路由环。

这个问题更大程度上是一个设计问题。主要问题是路由器 B 的汇总太广泛了，包含了不存在的子网。而路由器 A 也向路由器 B 发送一个更一般的汇总路由（缺省路由）。解决方法是让路由器 B 仅仅发送包含从 172.16.1.0 到 172.16.15.0 网络的汇总路由。换句话说，不是发送 172.16.0.0/16 汇总路由，相反，路由器 B 可以向路由器 A 发送 172.16.0.0 255.255.240.0 汇总路由。因此，当路由器 A 试图检查路由表查找 172.16.40.0/24 表项的时候，这一路由表仅仅返回 % Network not in table 消息，并丢掉分组，而不是把它发送给路由器 B，从而就结束了路由环。

7.6 EIGRP 重发布问题疑难解析

在许多实例中，当把其他路由协议重发布到 EIGRP 时会产生问题。图 7-32 显示了解决 EIGRP 重发布问题的流程图。

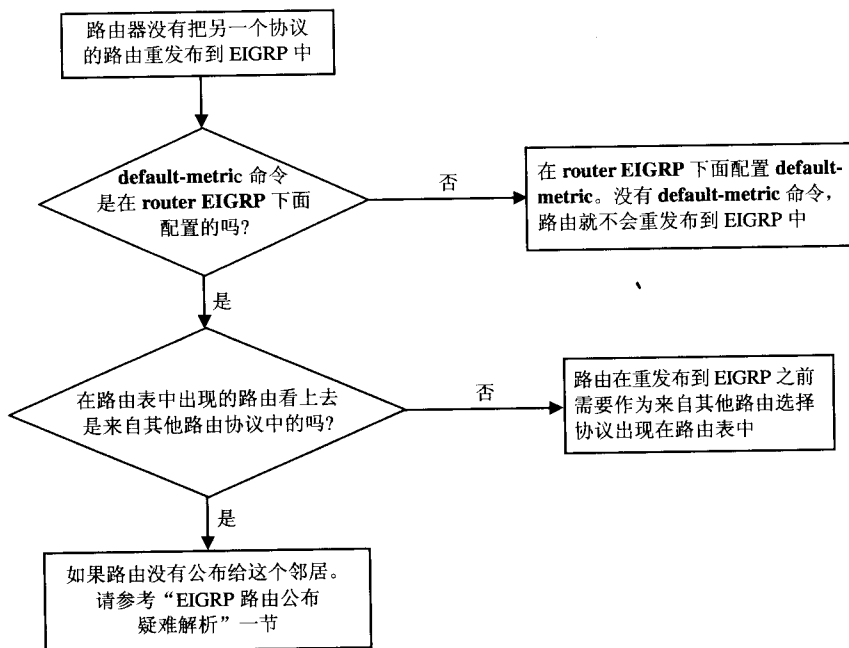


图 7-32 解决 EIGRP 重发布问题的流程图

考虑图 7-33 中的网络图，其中路由器是位于三个路由协议 RIP、OSPF 和 EIGRP 之间的边界路由器。

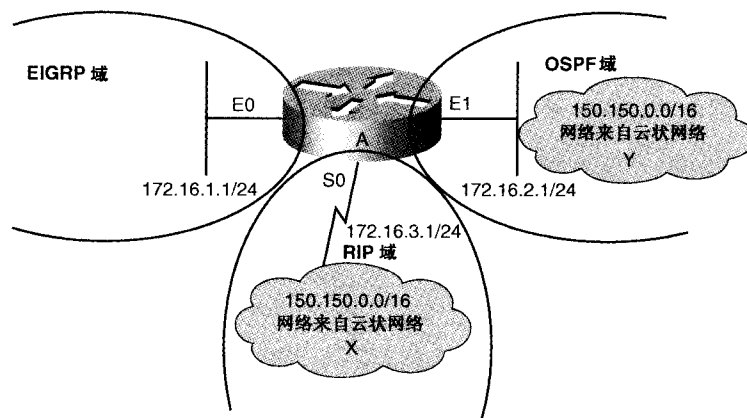


图 7-33 容易产生 EIGRP 重发布问题的网络

例 7-57 显示了路由器 A 的配置。

例 7-57 图 7-33 中路由器 A 的配置

```
Router A# interface ethernet 0
      ip address 172.16.1.1 255.255.255.0
interface ethernet 1
      ip address 172.16.2.1 255.255.255.0
interface serial 0
      ip address 172.16.3.1 255.255.255.0
router ospf 1
      network 172.16.0.0 0.0.255.255 area 0
router rip
      network 172.16.0.0
      passive-interface ethernet 1
router eigrp 1
      network 172.16.0.0
      redistribute rip
      default-metric 10000 100 255 1 1500
```

路由器 A 希望把 RIP 域中的所有路由都重发布到 EIGRP 域中。问题是网络 150.150.0.0/16 没有被重发布到 EIGRP 域中。

参考图 7-33，可以看到 150.150.0.0/16 网络在 RIP 域和 OSPF 域中出现了。在路由被重发布到 EIGRP 中之前，首先必须在 EIGRP 拓扑表中。检查例 7-58 中路由器 A 上网络 150.150.0.0/16 的 EIGRP 拓扑表。

例 7-58 用于 150.150.0.0/16 的 EIGRP 拓扑表

```
Router A# show ip eigrp topology 150.150.0.0 255.255.0.0
% Route not in topology table
```

正如输出所示, 路由 150.150.0.0/16 甚至不在 EIGRP 拓扑表中。例 7-59 显示了用于 150.150.0.0/16 网络的路由表。

例 7-59 用于 150.150.0.0/16 的路由表

```
Router A# show ip route 150.150.0.0 255.255.0.0

Routing entry for 150.150.0.0/16
  Known via "OSPF 1", distance 110, metric 186
  Redistributing via OSPF 1
  Last update from 172.16.2.2 on Ethernet 1
  Routing Descriptor Blocks:
  * 172.16.2.2, from 172.16.2.2, 00:10:23 ago, via Ethernet 1
    Route metric is 186, traffic share count is 1
```

例 7-59 中的输出显示了 150.150.0.0/16 路由作为一个 OSPF 路由出现, 而不是一个 RIP 路由。这是为什么该路由没有被重发布到 EIGRP 中的原因。在 RIP 路由被重发布到 EIGRP 中之前, 路由器查看其路由表, 并把所有的 RIP 路由重发布到 EIGRP 中。如例 7-59 中所示, 路由器既从 OSPF 又从 RIP 中获取对 150.150.0.0/16 路由的更新。因为 OSPF 有比 RIP 更低的管理距离, 所以路由器安装 OSPF 路由。因此, 如果这一路由作为一个 OSPF 路由出现, 那么路由器就不会把这个路由重发布到 EIGRP 中。换句话说, 路由器只把在路由表中出现的 RIP 路由重发布到 EIGRP 域中。

为了解决这一问题, 必须使路由器 A 安装 RIP 路由而不是 OSPF 路由。完成这件事情的一种方法是在 OSPF 下配置一个分发列表, 使其不安装路由 150.150.0.0/16, 如例 7-60 中所示。

例 7-60 在 OSPF 下配置一个分发列表从而不安装路由 150.150.0.0/16

```
router OSPF 1
  network 172.16.0.0 0.0.255.255 area 0
  distribute-list 1 out
  access-list 1 deny 150.150.0.0 0.0.255.255
  access-list 1 permit any
```

配置好分发列表之后, 路由器 A 用于 150.150.0.0/16 的路由表现在将显示例 7-61 中的结果。

例 7-61 在配置例 7-60 中的分配列表之后 150.150.0.0/16 的路由表

```
Router A# show ip route 150.150.0.0 255.255.0.0

Routing entry for 150.150.0.0/16
  Known via "RIP", distance 120, metric 4
  Redistributing via RIP
  Last update from 172.16.3.2 on Serial 0
  Routing Descriptor Blocks:
  * 172.16.3.2, from 172.16.3.2, 00:00:23 ago, via Serial 0
    Route metric is 4, traffic share count is 1
```

因为在路由器 A 中的路由表把 150.150.0.0/16 路由作为一个 RIP 路由显示, 所以重发布到 EIGRP 了, 路由器 A 中的 EIGRP 拓扑表现在显示例 7-62 中的结果。

例 7-62 在配置例 7-60 中的分配列表之后 150.150.0.0/16 的 EIGRP 拓扑表

```
Router A# show ip eigrp topology 150.150.0.0 255.255.0.0
IP-EIGRP topology entry for 150.150.0.0/16
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 281600
Routing Descriptor Blocks:
0.0.0.0, from RIP, Send flag is 0x0
Composite metric is (281600/0), Route is External
  Vector metric:
    Minimum bandwidth is 10000 Kbit
    Total delay is 1000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 0
  External data:
    Originating router is 172.16.3.1 (this system)
    AS number of routes is 0
    External protocol is RIP, external metric is 4
    Administrator tag is 0
```

该拓扑表表明路由 150.150.0.0/16 正在重发布到 EIGRP 中，使用的外部路由协议是 RIP。产生路由的路由器是 172.16.3.1，它是路由器 A。

考虑另一种情况，其中网络设置在图 7-34 中显示。在 OSPF 域中的路由不能被重发布到 EIGRP 域中。

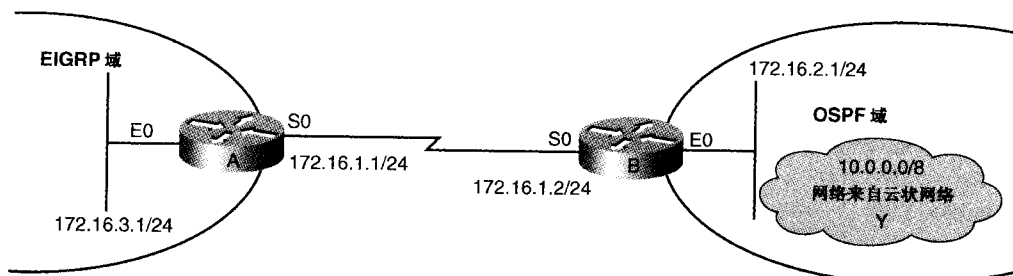


图 7-34 用于 OSPF 到 EIGRP 路由重发布问题的案例研究的网络设置

根据图 7-34 中显示的设置，路由器 B 正在从 OSPF 重发布到 EIGRP。10.0.0.0/8 网络来自 OSPF 域，正在被路由器 B 重发布到 EIGRP 域中。但是，路由器 A 绝不会在其路由表中看到 10.0.0.0/8 路由。例 7-63 显示了路由器 A 和路由器 B 的设置，例 7-64 显示了路由器 A 和路由器 B 中路由 10.0.0.0/8 的路由表。

例 7-63 用于图 7-34 中网络设置的路由器 A 和 B 的配置

```
Router A# interface ethernet 0
ip address 172.16.3.1 255.255.255.0
interface serial 0
ip address 172.16.1.1 255.255.255.0
```

(待续)

```

router eigrp 1
  network 172.16.0.0

Router B# interface ethernet 0
  ip address 172.16.2.1 255.255.255.0
interface serial 0
  ip address 172.16.1.2 255.255.255.0
router ospf 1
  network 172.16.0.0 0.0.255.255 area 0
router eigrp 1
  network 172.16.0.0
  redistribute ospf 1

```

例 7-64 在路由器 A 和 B 中用于 10.0.0.0/8 路由的路由表和 EIGRP 拓扑表

```

Router_A#show ip route 10.0.0.0 255.0.0.0
% Network not in table

Router_A# show ip eigrp topology 10.0.0.0 255.0.0.0
% Route not in topology table

Router_B# show ip route 10.0.0.0 255.0.0.0
Routing entry for 10.0.0.0/8
  Known via "OSPF 1", distance 110, metric 206
  Redistributing via OSPF 1
  Last update from 172.16.2.2 on Ethernet 0
  Routing Descriptor Blocks:
  * 172.16.2.2, from 172.16.2.2, 00:18:13 ago, via Ethernet 0
    Route metric is 206, traffic share count is 1
Router_B# show ip eigrp topology 10.0.0.0 255.0.0.0
% Route not in topology table.

```

根据例 7-64 的输出, 注意路由器 B 在其路由表中把路由 10.0.0.0/24 作为一个 OSPF 路由, 但是路由器 A 没有用于 10.0.0.0/8 的路由表表项。还有, 路由器 B 上的 EIGRP 拓扑表甚至没有用于路由 10.0.0.0/8 的表项。从此可以推断出在路由器 B 中从 OSPF 到 EIGRP 的重发布没有工作。

通过检查路由器 B 中的配置, 注意到虽然 **redistribute ospf 1** 命令在 EIGRP 下配置了, 但是没有 **default-metric** 命令的配置。当在不同的路由协议之间重发布时, 必须配置 **default-metric** 命令。当一个路由协议正在被重发布到另一个协议中时, 路由器没有把路由度量从一种路由协议转换到另一种协议的能力。**default-metric** 命令用于使网络管理员可以在路由重发布过程中手动地初始化路由度量。修正这一问题的方法是: 在路由器 B 中的 EIGRP 下配置一个缺省的度量。例 7-65 显示了路由器 B 修正后的配置。

例 7-65 纠正后的路由器 B 配置以修正图 7-34 中显示的重发布问题

```

Router B# interface ethernet 0
  ip address 172.16.2.1 255.255.255.0
interface serial 0

```

(待续)

```

ip address 172.16.1.2 255.255.255.0
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
router eigrp 1
 network 172.16.0.0
 redistribute ospf 1
 default-metric 10000 100 255 1 1500

```

根据例 7-65, 配置的缺省度量是 **default-metric 10000 100 255 1 1500**。10000 是以千位每秒为单位的带宽。100 是以 10 毫秒为单位的接口延迟。255 是接口可靠性, 其中 255 代表百分之百的可靠。1 是接口负载, 其中 255 代表百分之百的负载。最后一个数字 1500 是接口的 MTU。因为 10.0.0.0/8 路由来自路由器 B 的 Ethernet 接口, 所以我们设置的是匹配该 Ethernet 接口的缺省度量——也就是说, 10,000 千位每秒的带宽, 1000 毫秒的延迟, 百分之 100 的可靠性, 接口负载的 1/255, 以及 1500 字节的 MTU。记住: 路由器将接收来自缺省度量设置的任意值。路由器甚至会接收 1 1 1 1 1 的缺省度量值。但是, 使用与网络拓扑最匹配的缺省度量值将使路由器可以做出一个更好的路由决定。现在根据路由器 B 上恰当使用的正确配置, 例 7-66 显示了路由器 A 中用于路由 10.0.0.0/8 的路由表。

例 7-66 路由器 A 上用于路由 10.0.0.0/8 的路由表和路由器 B 中的 EIGRP 拓扑表证实问题已被修复

```

Router_A#show ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  Known via "eigrp 1", distance 170, metric 2195456, type external
  Redistributing via eigrp 1
  Last update from 172.16.1.2 on Serial0, 00:16:37 ago
  Routing Descriptor Blocks:
  *172.16.1.2, from 172.16.1.2, 00:16:37 ago, via Serial0
    Route metric is 2195456, traffic share count is 1
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1

```

```

Router B# show ip eigrp topology 10.0.0.0 255.0.0.0
IP-EIGRP topology entry for 10.0.0.0/8
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 281600
Routing Descriptor Blocks:
0.0.0.0, from Redistributed, Send flag is 0x0
Composite metric is (281600/0), Route is External
  Vector metric:
    Minimum bandwidth is 10000 Kbit
    Total delay is 1000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 0
  External data:
    Originating router is 172.16.2.1 (this system)
    AS number of routes is 1
    External protocol is OSPF, external metric is 206
    Administrator tag is 0

```

根据例 7-66, 可以看到路由器 A 把路由 10.0.0.0/8 作为 EIGRP 外部路由, 而路由器 B 有用于路由 10.0.0.0/8 的 EIGRP 拓扑表项。路由 10.0.0.0/8 现在已成功地从 OSPF 重发布到 EIGRP 中。

7.7 EIGRP 拨号备份问题疑难解析

拨号备份是远程接入路由器上的一个常见设置。当主链路失败的时候, 拨号备份提供另一种网络连接的方式。本节讨论 EIGRP 拨号备份问题, 主要是主链路复原后路由器没有断开拨号接口。请参见图 7-35 中解决 EIGRP 拨号备份问题的流程图。

图 7-36 显示了用于针对 EIGRP 拨号备份问题的案例研究的网络设置。

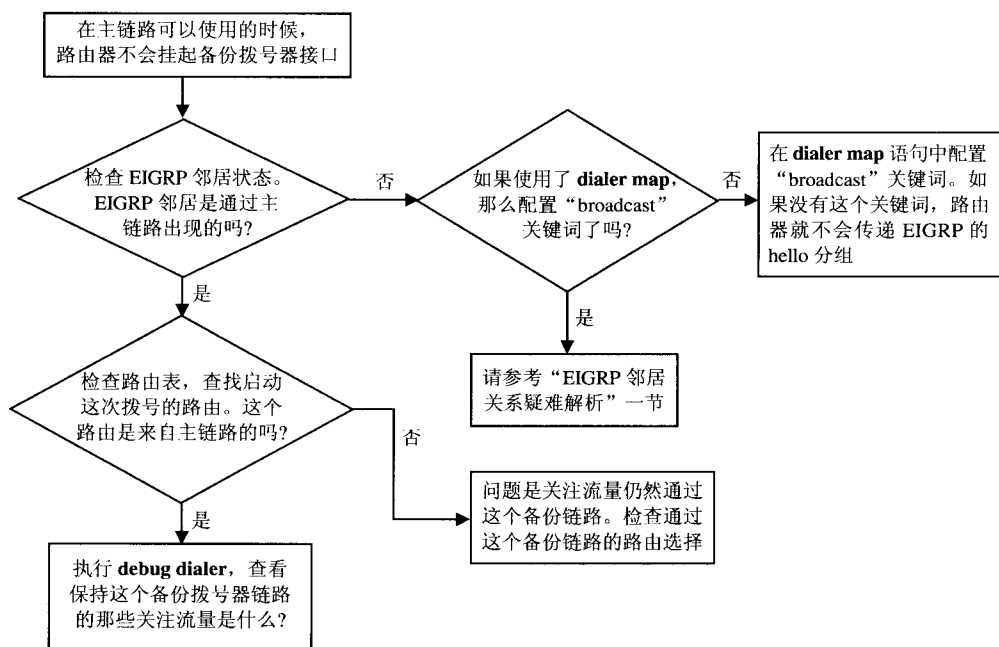


图 7-35 解决 EIGRP 拨号备份问题的流程图

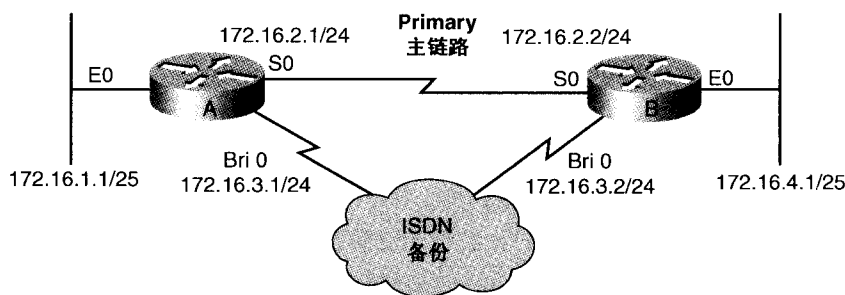


图 7-36 容易产生 EIGRP 拨号备份问题的网络

如图 7-36 所示, 路由器 A 和路由器 B 通过一条作为主链路的 T1 线路连接。如果该主链

路失败的话, ISDN 备份就起备份链路的作用。例 7-67 显示了路由器 A 和路由器 B 的配置。

例 7-67 图 7-36 中的路由器 A 和路由器 B 的配置

```
Router A# isdn switch-type basic-5ess
interface ethernet 0
    ip address 172.16.1.1 255.255.255.128
interface serial 0
    ip address 172.16.2.1 255.255.255.0

interface bri 0
    ip address 172.16.3.1 255.255.255.0
    encapsulation ppp
    dialer map ip 172.16.3.2 name Router B broadcast 1234567
    ppp authentication chap
dialer-group 1
router EIGRP 1
    network 172.16.0.0
access-list 101 deny eigrp any any
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
ip route 172.16.4.0 255.255.255.128 172.16.3.2 200
```

```
Router B# isdn switch-type basic-5ess
interface ethernet 0
    ip address 172.16.4.1 255.255.255.128
interface serial 0
    ip address 172.16.2.2 255.255.255.0
interface bri 0
    ip address 172.16.3.2 255.255.255.0
    encapsulation ppp
    dialer map IP 172.16.3.1 name Router_A broadcast 3456789
    ppp authentication chap
dialer-group 1
router eigrp 1
    network 172.16.0.0
access-list 101 deny eigrp any any
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
ip route 172.16.1.0 255.255.255.128 172.16.3.1 200
```

根据配置, 备份是通过位于配置末尾的浮动静态路由完成的。当主接口 (Serial 0) 停止时, 主要的 EIGRP 路由离开, 浮动静态路由被安装到使用 BRI 端口的路由表中。拨号列表被关联到 access-list 101, 它使用除 EIGRP 的 hello 分组之外的任何 IP 分组来启动拨号。这不会因为 EIGRP 的 hello 分组而导致 BRI 链路不停地拨号。

在这一情景中, 当主链路不可用时, 由于浮动静态路由, BRI 链路变为可用并传递流量。网络管理员试图修正这一链路问题; 在这样做的时候, 网络管理员重启路由器 B。当路由器 B 重启后, 主链路也变为可用了。问题是即使当主链路已经变为可用, BRI 链路也仍然出现, 而且流量仍然通过 BRI 端口传递。

在路由器 A 上, 必须验证用于这些关注流量的路由表表项是正确的。例 7-68 显示了在

路由器 A 上 **show ip route 172.16.4.0** 命令的输出。

例 7-68 172.16.4.0 的路由表

```
Router A# show ip route 172.16.4.0

Routing entry for 172.16.4.0/25
  Known via "static", distance 200, metric 0
  Routing Descriptor Blocks:
    * 172.16.3.2
      Route metric is 0, traffic share count is 1
```

例 7-68 中的输出表明路由器 A 仍然正向路由器 B 的 Ethernet 网络安装浮动静态路由。下一步是确信在主接口上路由器 A 和路由器 B 之间正确建立 EIGRP 邻居关系。可以使用 **show ip eigrp neighbor** 命令验证这一点，如例 7-69 中所示。

例 7-69 验证路由器 A 和路由器 B 之间的一个 EIGRP 邻居关系

```
Router A# show ip eigrp neighbor

IP-EIGRP neighbors for process 1
H   Address      Interface    Hold    Uptime    SRTT    RTO    Q    Seq
      (sec)      (ms)
0   172.16.2.2    S0           12      00:10:23   21      200    0    23
1   172.16.3.2    BRI0         12      00:10:23   40      240    0    50

Router B# show ip eigrp neighbor

IP-EIGRP neighbors for process 1
H   Address      Interface    Hold    Uptime    SRTT    RTO    Q    Seq
      (sec)      (ms)
0   172.16.2.1    S0           12      00:10:30   21      200    0    24
1   172.16.3.1    BRI0         12      00:10:30   40      240    0    51
```

从这两个路由器来看，邻居关系看上去是好的。路由器 A 和路由器 B 都显示邻居关系已建立，没有任何问题。下一步是检查路由器 B 上的配置，以确信所有内容都配置正确。例 7-70 显示了路由器 B 在重启之后的配置。

例 7-70 路由器 B 在重启之后的配置

```
Router B# interface ethernet 0
      ip address 172.16.4.1 255.255.255.0
interface serial 0
      ip address 172.16.2.2 255.255.255.0
interface bri 0
      ip address 172.16.3.2 255.255.255.0
      encapsulation PPP
      dialer map IP 172.16.3.1 name Router A broadcast xxx
      ppp authentication chap
dialer-group 1
```

(待续)

```

router eigrp 1
  network 172.16.0.0
access-list 101 deny eigrp any any
access-list 101 permit IP any any
dialer-list 1 protocol IP list 101
ip route 172.16.1.0 255.255.255.128 172.16.3.1 200

```

现在要注意路由器 B 的 Ethernet 0 的配置, IP 地址是 172.16.4.1 255.255.255.0; 掩码已经从/25 改成/24。这是这一问题的原因。当路由器 B 向路由器 A 公布它的 Ethernet 0 路由的时候, 它向路由器 A 公布 172.16.4.0/24 路由, 而路由器 A 仍然安装了浮动静态路由 172.16.4.0/25。路由表包含/25 路由, 因为它有一个更长的子网掩码。错误的掩码出现, 因为当网络管理员重启路由器 B 的时候, 路由器 B 使用了它以前存储的旧的配置, 在网络管理员把 Ethernet 0 的旧子网掩码改成/25 之前, 它是/24。当制定这一改动的时候, 网络管理员没有保存配置。

这一问题的解决方法是把路由器 B 中的 IP 地址子网掩码改成/25。例 7-71 显示了路由器 B 的 Ethernet 0 接口的配置。

例 7-71 正确配置用于路由器 B 的 Ethernet 0 接口的子网掩码

```

Router B# interface ethernet 0
ip address 172.16.4.1 255.255.255.128

```

现在这一改动导致路由器 B 向路由器 A 发送一个 EIGRP 更新 172.16.4.0/25, 导致路由器 A 使用 EIGRP 路由而不是那个浮动静态路由。例 7-72 显示了路由器 A 的路由表现在的情况。

例 7-72 在例 7-71 中配置改动之后路由器 A 上用于 172.16.4.0 的路由表

```

Router A# show ip route 172.16.4.0

Routing entry for 172.16.4.0/25
Known via "EIGRP 1", distance 90, metric 2195456, type internal
  Redistributing via eigrp 1
  Last update from 172.16.2.2 on Serial 0, 00:10:30 ago
  Routing Descriptor Blocks:
    * 172.16.2.2, from 172.16.2.2, 00:10:30 ago, via Serial 0
      Route metric is 2195456, traffic share count is 1
  Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1

```

流量停止流向 BRI 0 接口, 开始流向主链路。于是 BRI 接口停止, 重新变为备份模式。

7.8 EIGRP 错误消息

在日志中发生的一些 EIGRP 错误消息迷惑了许多网络管理员。本节讨论最常出现的一些 EIGRP 错误以及隐藏在這些 EIGRP 错误消息后面的含义:

- **DUAL-3-SIA**——这条消息表示主要路由断开，而且没有可以使用的可行后继者。路由器已经向其邻居发送了查询分组，在三分钟之后还没有听到来自一个特定邻居的响应。路由状态现在是被粘滞于活动状态。在“EIGRP 邻居关系疑难解析”一节有关于这个错误更详细的讨论。
- **Neighbor not on common subnet**——这条消息表示路由器已经收到来自一个邻居的 hello 分组，该邻居与这一路由器不在同一个子网中。关于这个错误更详细的讨论也可以在“EIGRP 邻居关系疑难解析”一节中找到。
- **DUAL-3-BADCOUNT**——badcount 表示 EIGRP 认为对于一个特定的网络，它知道的路由比实际存在的路由还要多。典型情况下（不是始终），可以看到它与 DUAL-3-SIA 结合在一起，但是 EIGRP 认为仅有这一个错误不会引起任何问题。
- **Unequal, <route>, dndb=<metric>, query=<metric>**——这条消息仅仅是提供信息的。它指出路由器在发送查询分组时具有的度量与它在接收响应时具有的度量不匹配。
- **DUAL-3-INTERNAL: IP-EIGRP Internal Error**——这条消息表明有一个 EIGRP 内部错误。但是，路由器被编码为完全可以从这个内部错误中恢复。这个 EIGRP 内部错误是由软件问题引起的，不应该影响路由器的操作。这一行为的目的是向 TAC 报告这个错误，使那些专家解码这一反馈消息。让他们识别这一程序错误编号并据此升级 Cisco IOS。
- **IP-EIGRP: Callback: callback_routes**——在某个时候，EIGRP 试图向接收方安装路由，但是失败了，最常见的原因是由于存在一个具有更好的管理距离的路由。当发生这种情况时，EIGRP 把其路由注册为备份路由。当更好的路由从路由表中删除时，通过 callback_routes 重新调用 EIGRP，从而使它可以试图重新安装那些它保存在其拓扑表中的路由。
- **Error EIGRP: DDB not configured on interface**——这条消息表示当路由器的接口收到一个 EIGRP 的 hello 分组并且路由器开始把这一分组与一个用于那个接口的 DDB（DUAL 描述符块，DUAL descriptor block）联系在一起的时候，它没有找到相匹配的 DDB。这表示路由器正在其没有配置 EIGRP 的接口上接收 hello 分组。
- **Poison squashed**——路由器为了响应一个更新而把某个拓扑表表项作为一个毒性逆转装入（该路由器设置了毒性逆转）。当路由器正在构建包含毒性逆转的分组时，该路由器认识到不需要发送它。例如，如果路由器从其邻居那里收到了询问那个路由的一个查询，那么它通常被应用毒性逆转。

7.9 小 结

本章讨论了解决各种 EIGRP 问题的方法。针对每一种问题给出的流程图指出了正确的疑难解析方向。当在路由器上执行调试时，要记住任何调试都有可能使该路由器无法承受，必须在路由器 CPU 利用率较低时再执行调试，最好是在维护期间。正如在本章中所指出的，大量的疑难解析都可以只通过执行 show 命令完成。需要花时间理解本章介绍的各种 show 命令输出的细节。这样，当发生问题的时候，就可以快速敏捷地识别出问题原因所在并修正它。

第 8 章

理解 OSPF 协议

本章涵盖下列有关 OSPF 协议的关键主题：

- OSPF 分组细节。
- OSPF LSA 细节。
- OSPF 区域。
- OSPF 介质类型。
- OSPF 邻接体。

OSPF 是一种设计用于大型复杂网络的链路状态内部网关协议。作为一项 IETF 的标准，OSPF 广泛应用于许多大型网络中。它的发展始于 1987 年，到 1991 年，随着 RFC1247 的发布，OSPF 协议版本 2 被确定下来了。其目的是为了拥有一个比 RIP 协议更有效和更具扩展性的链路状态协议。RFC 2328（1998 年 4 月）是 OSPF 协议版本 2 的最近修订。

OSPF 直接运行于 IP 协议之上并且使用协议号 89，就如同 TCP 协议直接运行于 IP 协议之上并且使用协议号 6 一样。OSPF 没有使用任何传输协议（例如 TCP）来提供可靠性保障。OSPF 协议自身有一个传输的可靠机制。

OSPF 是一个无类路由协议，它支持可变长子网掩码（VLSM）和非连续网络。OSPF 使用组播地址 224.0.0.5（所有的 SPF 路由器）和 224.0.0.6（指定路由器[DR]和备份指定路由器[BDR]）来发送 Hello 和路由更新分组。OSPF 也提供了两种类型的认证（authentication）——普通的文本和消息摘要算法 5（MD5）。

OSPF 使用 Dijkstra 算法作为路由表计算过程的一部分。Dijkstra 算法产生最短路径树（SPT）。每一个路由器用一种易于理解的方式——链路状态通告（LSA）来向邻居表述自己和自己的链路。基于来自最短路径树的信息，OSPF 可以画出网络拓扑结构。

使用 OSPF 协议的每个路由器与其它路由器交换关于它的代价、链路类型和网络信息的各种信息。正如本章后面所讨论的, 这种多步骤过程叫做链路状态通告 (LSA) 交换。

8.1 OSPF 分组细节

OSPF 有 5 种分别用于不同目的的分组。表 8-1 列出了不同的 OSPF 分组类型同时描述了它们的功能。

表 8-1

OSPF 分组类型

类 型	描 述	功 能
1	Hello	用于发现邻居和形成 DR / BDR 关系并且交换邻居性能参数
2	数据库描述	用于为数据库交换过程选举主 / 从设备, 并为数据库交换交换 LSA 头和选择第一个序列号
3	链路状态请求	用于在 DBD 交换过程中请求一个特定的 LSA
4	链路状态更新	用于向通过链路请求分组请求特定 LSA 的邻居发送整个的 LSA。这个分组也用于泛洪
5	链路状态确认	用于确认链路状态更新分组的接收

所有 OSPF 分组类型都有一个通用的 20 字节 OSPF 协议头。图 8-1 显示了通用的 OSPF 协议头格式

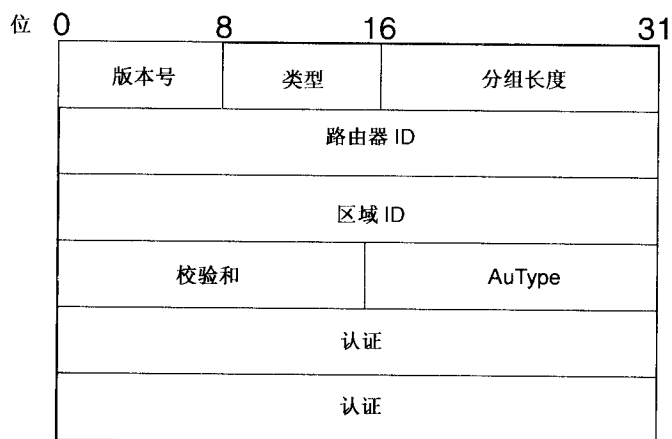


图 8-1 通用 OSPF 协议头格式

下面的列表描述了 OSPF 协议头中的字段。

- **版本号**——这个字段表明了 OSPF 的当前版本号。最近的版本是 2。版本 1 和版本 2 不兼容。
- **类型**——这个字段表明附加在分组头后面的是 OSPF 分组 5 种类型中的哪一种。
- **分组长度**——这个字段包含了整个 OSPF 分组的长度, 这一长度包括了 OSPF 头在内。
- **路由器 ID**——这个字段包含有 4 个字节的 IP 地址。路由器 ID 用于在自治系统中唯一地标识该路由器。对一个 Cisco 路由器来说, 这个字段包含此路由器的最大的 (highest) IP 地址。如果配置了回环地址 (loopback), 那么最大的回环地址就成为了路由器 ID。在路由器 ID 被选择了之后, 它将一直不改变, 除非路由器重启、被选择作为路由器 ID 的接口关闭, 或者是这个接口上的 IP 地址已经被删除或取代。
- **区域 ID**——这个字段指定分组所属的区域号。这也是一个 4 字节编号。为了形成邻

居关系，它的值在两方必须是相同的。这个字段的书写有两种方式：区域 1 或者区域 0.0.0.1。这两种书写方式没有区别。

- **校验和**——这个字段包含有除了对数据损坏的认证部分之外的整个 OSPF 分组的校验。
- **AuType**——这个字段包含有认证的类型码：
 - 0 表示没有认证（null authentication）。
 - 1 表示认证类型为普通文本。
 - 2 表示认证类型为 MD5。
- **认证**——如果是普通文本认证则这个 64 比特的字段包含有认证密钥。如果是消息摘要认证，则这个 64 比特的认证字段被重新定义为一些其它的参数。参见附录 D，RFC 2328，了解关于 MD5 认证方案的更多细节。

8.1.1 Hello 分组

Hello 分组是 OSPF 中的第一种分组类型。图 8-2 说明了 Hello 分组的格式。Hello 分组用于在两个路由器之间形成一个邻居关系。在包含有广播 / 非广播介质（media）的环境中，Hello 分组用于选举指定（DR）和备份指定（BDR）路由器。在广播介质中，Hello 分组目的地址是 224.0.0.5。在非广播介质中，目的地址是单播地址。

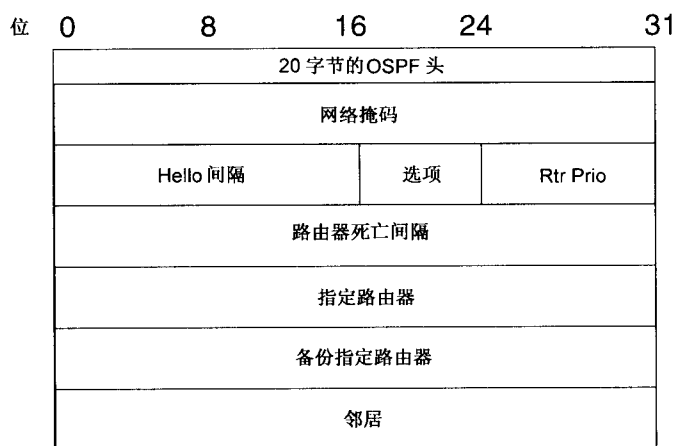


图 8-2 Hello 分组格式

下面的列表描述了 Hello 分组中的各字段：

- **网络掩码**——表示 OSPF 运行接口的网络掩码。这个网络掩码只在广播介质中才被检查。
- **Hello 间隔**——表示发送两个 Hello 分组之间相隔的时间秒数。这个间隔对于两个正尝试形成一个邻接体关系的路由器来说必须是相同的。Hello 间隔在广播介质和点到点介质中都是 10 秒，而在其它介质中是 30 秒。
- **选项**——表示路由器支持的可选特性。选项字段有着如下格式：

*	O	DC	EA	N/P	MC	E	T
---	---	----	----	-----	----	---	---

- O 比特用于不透明的 LSA（opaque LSA），在 RFC 2370 中提及。
- DC 用于请求线路（demand circuit）的性能，在 RFC 1793 中提及。
- EA 是外部特征。

- N/P 用于非完全端区 (not-so-stubby area, NSSA), 在 RFC 1587 中提及。
- MC 指定组播 OSPF。
- E, 当设置时, 表示在这个区域中允许出现外部 LSA。
- T 比特用于 ToS 特性 (通常设置为 0)。

可选项字段的第一个比特被预留到将来使用。Cisco 路由器也没有使用 EA 和 MC 比特。

- **Rtr Pri**——用于路由器的优先级。缺省时, 这个值被设置为 1。这个字段在选举 DR 和 BDR 的时候扮演着重要角色。高的优先级增加了这个路由器变成 DR 的机会。优先级为 0 表示这个路由器将不参与 DR 的选举。
- **路由器死亡间隔**——表示在一个邻居被宣布死亡之前以秒为单位的时间数目。缺省时, 死亡间隔是 Hello 间隔的 4 倍。
- **指定路由器**——列出指定路由器的 IP 地址。如果没有 DR, 则这个字段值为 0.0.0.0。DR 是通过 Hello 协议选举出来的。有着最高优先级的路由器变成 DR。如果优先级相等, 有着最大路由器 ID 的路由器成为 DR。DR 的目的是在多点接入介质 (multiaccess media) 中减少泛洪的数量。DR 使用组播来减少泛洪的数量。所有的路由器将它们的链路状态数据库向 DR 泛洪, 同时 DR 又将这些信息反过来泛洪给这个网段中其它的路由器。在点到点或者点到多点网段中没有 DR 或者 BDR 存在。
- **备份指定路由器**——标识 BDR 和列出 BDR 接口的 IP 地址。如果 BDR 不存在, 这个字段值为 0.0.0.0。BDR 也是通过 Hello 协议选举出来的。BDR 的目的是作为 DR 的备份, 在 DR 死亡的时候作一个平滑的转换。BDR 在泛洪中保持被动。
- **邻居**——包含每个邻居的路由器 ID, 从这些邻居收到了 Hello 分组。

8.1.2 数据库描述分组

OSPF 分组的第 2 种类型, 数据库描述 (DBD) 分组, 大多用于数据库交换期间。第一个 DBD 分组用于选举主从关系和配置由主设备选举的最初的序列号。有着最大路由器 ID 的路由器变成主设备并且对数据库同步进行初始化。主路由器发送序列号, 从路由器进行确认。在主从路由器选举出来之后, 数据库同步过程开始了, 在这个过程中, 所有 LSA 的分组头都在邻居之间进行交换。图 8-3 说明了 DBD 分组格式。

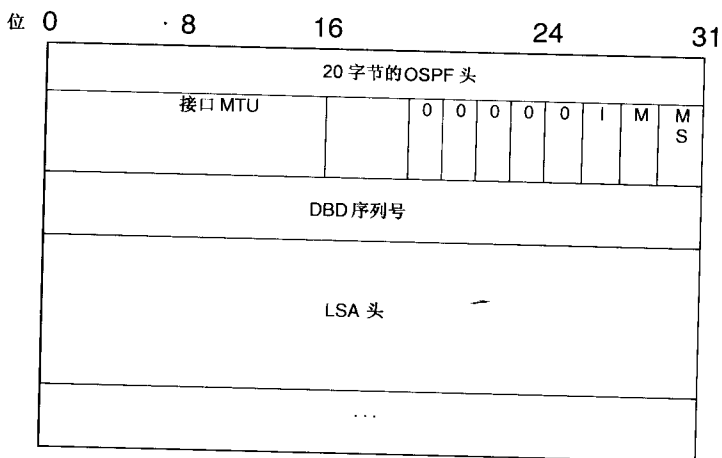


图 8-3 数据库描述分组格式

下面的列表描述了 DBD 分组中的字段：

- **接口 MTU**——这个字段包含有以字节为单位、可以通过相关接口传送的最大数据的大小。这个选项从 RFC 2178 起开始被添加。当通过虚链路发送分组时这个字段必须设置为 0。
- **选项**——这个字段的选项在上一节有关 Hello 分组的格式中已经被讨论过了。
- **I 比特**——当设置为 1 时，表示这是 DBD 交换中的第一个分组。
- **M 比特**——当设置为 1 时，表示后面将有更多的分组过来。
- **MS 比特**——用于主从设备。当这个比特设置为 1 时，表示路由器在 DBD 交换过程中是主设备，如果这个比特被设置为 0，表示路由器是从设备。
- **DBD 序列号**——这个字段包含一个由主设备设置的唯一的值。这个序列号在数据库交换过程中使用。只有主设备才能增加序列号。
- **LSA 头**——这个字段由一个链路状态数据库头列表组成。

8.1.3 链路状态请求分组

OSPF 分组的第 3 种类型，即链路状态请求分组，它在部分路由数据库丢失或者过时的情况下被发送。链路状态请求分组用于取回那些已丢失的数据库信息的精确部分。链路状态分组也用于在 DBD 交换完成之后请求得到在 DBD 交换中见到过的 LSA。图 8-4 说明了链路状态请求分组格式。

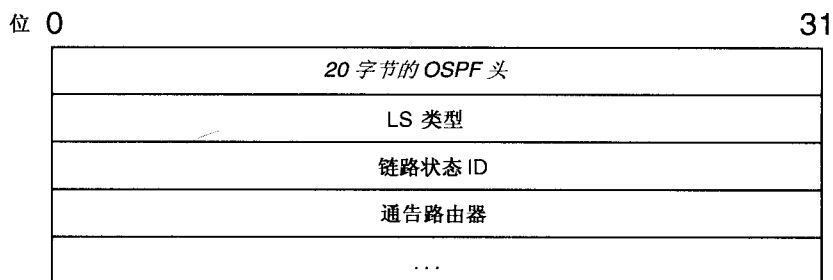


图 8-4 链路状态请求分组格式

下面的列表描述了链路状态请求分组中的字段：

- **LS 类型**——标识何种类型的 LSA 被请求。
- **链路状态 ID**——代表特定 LSA 的链路状态 ID。链路状态 ID 在本章稍后讨论。
- **通告路由器**——包含最初产生这个 LSA 的路由器的路由器 ID。

8.1.4 链路状态更新分组

OSPF 分组第 4 种类型，即链路状态更新分组，它实现泛洪。多个 LSA 被包含在单个分组中。链路状态更新分组也被发送来应答链路状态请求分组。被泛洪的 LSA 在 LSA 确认分组中被确认。如果一个 LSA 没有被确认，它将每隔一个重传间隔（缺省为 5 秒）被重传一次。图 8-5 说明了链路状态更新分组格式。

如图 8-5 所示, 这个分组包含了 LSA 的数量 (例如 10 个或者 20 个 LSA), 后面跟着这些 LSA 自身。



图 8-5 链路状态更新分组格式

8.1.5 链路状态确认分组

最后一种 OSPF 分组类型, 即链路状态确认分组, 用于对每个 LSA 进行确认。这个分组被发送用于应答链路状态更新分组。多个 LSA 可以在单个链路状态确认分组中被确认。这个分组承担链路状态更新分组可靠传输的责任。图 8-6 说明了链路状态确认分组的格式。

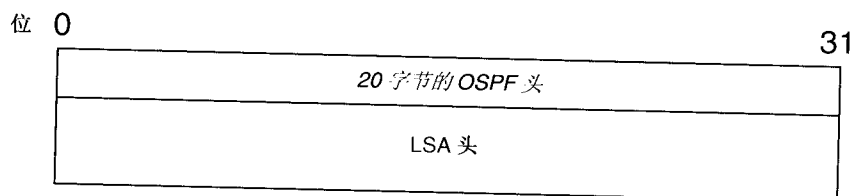


图 8-6 链路状态确认分组格式

链路状态确认分组以组播形式发送。如果路由器状态是 DR 或者 BDR, 确认将被发送到 OSPF 路由器组播地址 224.0.0.5。如果路由器状态不是 DR 或 BDR, 确认将被发送到所有 DR 路由器组播地址 224.0.0.6。

8.2 OSPF LSA 细节

存在多种类型的 LSA。本节讨论列于表 8-2 中的 9 种 LSA 类型。

表 8-2

LSA 的类型

类 型	LSA	功 能
1	路由器	定义通向邻居的链路的状态和代价以及与点到点链路相关的 IP 前缀
2	网络	定义归属于这个网段的路由器数量。它给出了在此网段子网掩码的信息
3	网络汇总 (Summary network)	描述那些在一个区域外但是同时又处于 OSPF 域内的目的地。一个区域的汇总被泛洪到其它区域, 反之亦然
4	自治系统边界路由器汇总 (Summary ASBR)	描述 ASBR 的信息。在一个单一区域 (single area), 将不存在 LSA 类型 4 汇总

续表

类 型	LSA	功 能
5	外部 (external)	定义到 OSPF 域之外目的地的路由, 每个子网由一个单一的外部 LSA 表示
6*	组成员	
7	NSSA	定义到外部目的地的路由, 但是以被称作类型 7 的单独的格式存在
8*	没有使用	
9-11*	不透明	

*类型 6 用于标识 OSPF 组播 (MOSPF) 中的组成员, 它没有被 Cisco 实现。类型 8 没有使用, 类型 9-11 被用于不透明 LSA, 不透明 LSA 不是用于路由计算而是用于 MPLS 流量工程, 这超出了本章讨论范围。关于不透明 LSA 的更多信息可以在 RFC 2370 中找到。

每个 LSA 都有一个 20 字节长的通用 LSA 头, 它的格式在图 8-7 中描述。

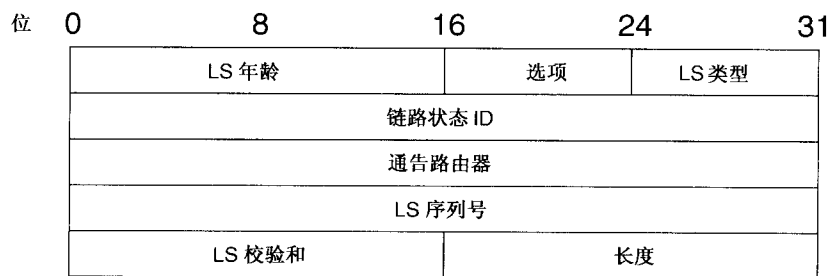


图 8-7 通用 LSA 头格式

下面的列表描述了 LSA 头中的字段。

- **LS 年龄**——给出以秒为单位的从 LSA 产生起的时间。LSA 最大的年龄是 3600 秒; 刷新时间为 1800 秒。如果 LSA 年龄达到了 3600 秒, 则 LSA 必须从数据库中清除出去。
- **选项**——在早先“Hello 分组”一节中讨论过了。
- **LS 类型**——表示 LSA 类型, 它们中的一些列于表 8-2 中。
- **链路状态 ID**——标识由 LSA 描述的网络部分。这个字段遵照 LS 类型而变化。
- **通告路由器**——表示产生 LSA 的路由器的路由器 ID。
- **LS 序列号**——检测老的或重复的 LSA。每个后继的实例都被给予一个后继的序列号。最大的序列号表示为 0x7FFFFFFF。第一个序列号总是 0x80000001。序列号 0x80000000 被保留。
- **LS 校验和**——完成对 LSA 的检验, 不包括 LS 年龄字段。一个 LSA 可能在泛洪过程中或者存储于内存中的时候被损坏, 所以这个校验和是需要的。这个字段的值不能为 0, 因为 0 表示校验和没有被处理。校验和的处理是在 LSA 产生或者收到 LSA 的时候进行的。它也发生在每个校验年龄间隔, 这个间隔缺省为 10 分钟。
- **长度**——包括 20 字节头长度在内的 LSA 长度。

8.2.1 路由器 LSA

路由器 LSA 是由每个路由器为它所属的每个区域产生的。这些分组描述了路由器到区域

连接链路的状态并且只在一个特定的区域内被泛洪。所有在一个区域内的路由器链路必须在一个单一的 LSA 中描述。

路由器 LSA 通过特定区域泛洪；可是，这个 LSA 的泛洪被限制在一个区域中。一个路由器的路由器 LSA 不能存在于区域之外；否则，OSPF 中的每个单一路由器将不得不承运巨大数量的细节信息。这些细节存在于一个区域之内。路由器表明它是否是一个区域边界路由器（ABR）、自治系统边界路由器（ASBR）或者是一个虚链路的终点。

图 8-8 显示了路由器 LSA 的分组格式。

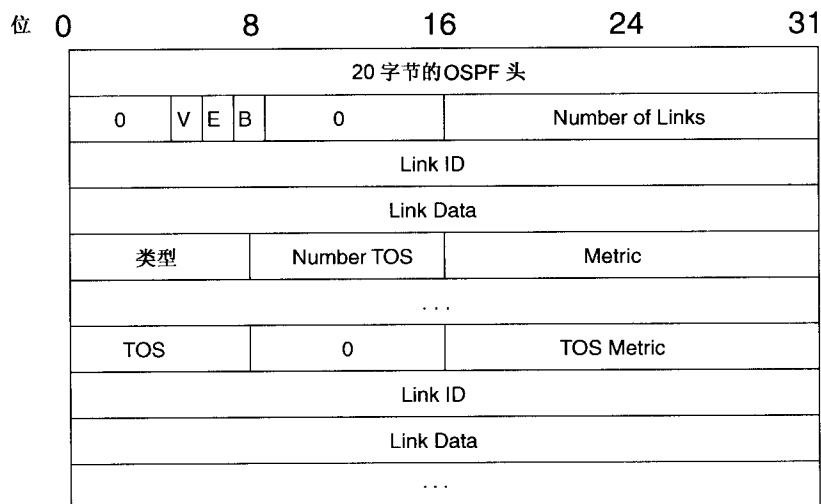


图 8-8 路由器 LSA 分组格式

下面的列表描述了路由器 LSA 分组中的字段。

- **V 比特**——这个比特用于确定它是否是一个虚链路终点。
- **E 比特**——这个比特用于确定这个路由器是不是一个自治系统边界路由器（ASBR）。
- **B 比特**——这个比特用于确定这个路由器是否是一个区域边界路由器（ABR）。
- **链路数量**——它含有路由器链路的数量。注意对一个区域来说，这个路由器 LSA 在一个单一的 LSA 中包含有这个区域中所有的路由器链路。
- **链路 ID、链路数据和类型**——类型字段表示 4 种路由器链路类型。另外两个字段，链路 ID 和链路数据，依据网络类型表示 4 字节 IP 地址值。在这里一个要注意的事情是这儿可能有两种类型的点到点链路：有编号的（numbered）和无编号的（unnumbered）。如果有编号的点到点链路，则链路数据字段含有与邻居相连的接口地址。如果是无编号链路，则链路数据字段含有 MIBII Ifindex 值，它是一个与每个接口相关的唯一的值。它的值通常从 0 开始，如同在 0.0.0.17 中一样。表 8-3 列出了链路 ID 和链路数据字段的所有可能值。
- **ToS 和 ToS 度量（Metric）**——这些字段代表服务类型并且通常设置为 0。
- **度量**——这个字段包含有一个特定链路的 OSPF 代价。计算 OSPF 代价的公式是 $10^8 / \text{链路带宽}$ 。例如，一个快速以太网接口的度量将是 1。度量由接口带宽直接决定，而接口带宽是可配置的。这个用于度量计算的公式可以用两种方法来实施。第一种方法是在接口上使用 `ip ospf cost cost` 命令其中 cost 是链路的 OSPF 代价。第二种方法是

在 **router ospf** 配置下使用 **auto-cost reference-band widthreference-bandwidth** 命令。
这个参考带宽 (**reference-bandwidth**) 实际改变了在度量计算公式中 10^8 这个值。

表 8-3

不同的路由器链路类型

类 型	描 述	链路 ID	链路数据
1	点到点有编号的	邻居的路由器 ID	接口 IP 地址
1	点到点无编号的	邻居的路由器 ID	MIBII Ifindex 值
3	中继 (transit)	DR 的 IP 地址	接口 IP 地址
4	端 (stub)	IP 网络号	子网掩码
5	虚链路	邻居的路由器 ID	接口 IP 地址

路由器 LSA 例子

例 8-1 显示了 Cisco 路由器上的一个路由器 LSA 的输出。

例 8-1 路由器 LSA 的输出

```
RouterB#show ip ospf database router 141.108.1.21
LS age: 1362
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 141.108.1.21
Advertising Router: 141.108.1.21
LS Seq Number: 80000085
Checksum: 0xE914
Length: 60
Area Border Router
Number of Links: 3
  Link connected to: another Router (point-to-point)
    (Link ID) Neighboring Router ID: 141.108.1.3
    (Link Data) Router Interface address: 141.108.1.2
    Number of TOS metrics: 0
    TOS 0 Metrics: 64

  Link connected to: another Router (point-to-point)
    (Link ID) Neighboring Router ID: 141.108.3.1
    (Link Data) Router Interface address: 141.108.1.2
    Number of TOS metrics: 0
    TOS 0 Metrics: 64

  Link connected to: a Stub Network
    (Link ID) Network/subnet number: 141.108.1.2
    (Link Data) Network Mask: 255.255.255.255
    Number of TOS metrics: 0
    TOS 0 Metrics: 0
```

例 8-1 的输出显示出了 3 种链路。在这个输出中一些应注意的重要事情 (图中标出的高亮部分) 如下:

- 一般情况下, LS 年龄字段应小于 1800。
- 在一个路由器 LSA 的情形中, 链路状态 ID 字段和通告路由器字段应该有相同的值, 如例 8-1 中所示。

- 这个路由器是一个 ABR 并且有三条路由器链路。

对每一条点到点链路，都有一个端链路用以提供链路的子网掩码。在这个例子中，有两条点到点链路和一条端链路与这两条点到点链路相关联，因为网络类型是点到多点的。因而，如果这儿有 300 条点到点链路，路由器将产生 300 条点到点链路和 300 条端链路来访问与每个点到点链路相关联的子网。在这个例子中点到多点网络类型是更好的选择，有两个原因：

- 每个点到多点网络只需要一个子网。
- 路由器 LSA 的大小被减半，因为将只会有一条端链路去访问一个点到点网络上的子网。这条链路通常是一个主机地址。

如果按这些信息画一幅网络拓扑图，你将能看到一个 OSPF 网络的一小部分，如图 8-9 所示。

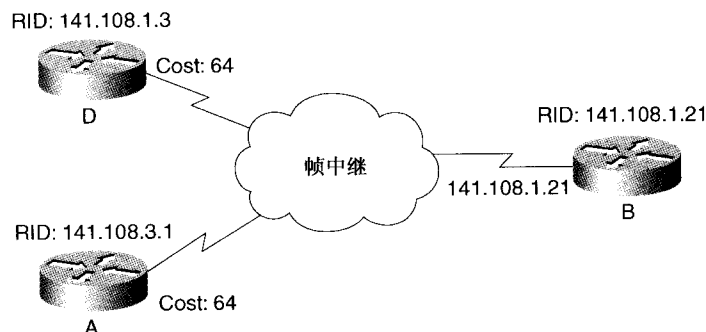


图 8-9 根据包含在路由器 LSA 中的信息而画出的网络拓扑

8.2.2 网络 LSA

DR 产生网络 LSA。如果没有 DR 存在（例如，在点到点或点到多点网络中），那将会没有网络 LSA。网络 LSA 描述所有属于此网络的路由器。这个 LSA 在包含这个网络的区域中被泛洪，就像路由器 LSA 一样。图 8-10 显示了网络 LSA 的分组格式。

位	0	8	16	24	31
	20 字节的 OSPF 头				
	网络掩码				
	附加路由器				
	...				

图 8-10 网络 LSA 分组格式

网络 LSA 有两个重要部分：

- **网络掩码**——这个字段表示与中继链路相关的网络掩码。
- **附属路由器**——这个字段包含有与这个中继链路相关的每个路由器的路由器 ID。指定路由器也将它自己列入到附属路由器中。

网络 LSA 例子

例 8-2 显示了一个 Cisco 路由器的网络 LSA 输出。

例 8-2 网络 LSA 输出

```

RouterA#show ip ospf database network 141.108.1.1
Routing Bit Set on this LSA
LS age: 1169
Options: (No TOS-capability, DC)
LS Type: Network Links
Link State ID: 141.108.1.1 (address of Designated Router)
Advertising Router: 141.108.3.1
LS Seq Number: 80000002
Checksum: 0xC76E
Length: 36
Network Mask: /29
Attached Router: 141.108.3.1
Attached Router: 141.108.1.21
Attached Router: 141.108.1.3

```

例 8-2 中输出的最后三行说明有三台路由器归属于这条中继链路。这条中继链路的网络掩码是/29。这里有两个重要的事情要记住：

- 链路状态 ID 字段总是包含 DR 的 IP 地址。
- 通告路由器字段总是包含 DR 的路由器 ID。

你可以同样地根据包含在网络 LSA 中的信息画一幅网络拓扑来显示附属的路由器号和链路的网络掩码。

图 8-11 显示了根据输出的网络 LSA 中的信息而画出的网络拓扑

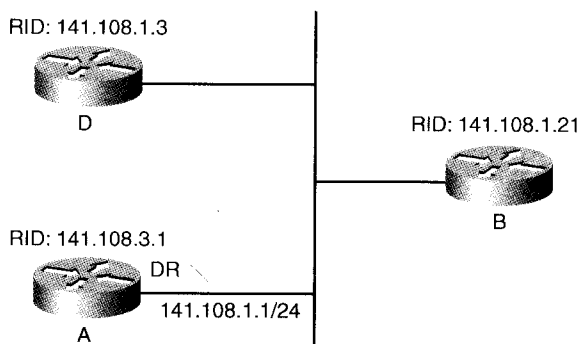


图 8-11 根据包含在路由器 LSA 中的信息而画出的网络拓扑

8.2.3 汇总 LSA

汇总 LSA (summary LSA) 描述那些在区域之外，但仍然位于自治系统中的目的地。当提供不只一个区域并且配置了区域 0 的时候，汇总 LSA 产生了。汇总 LSA 的目的是向区域外发送已精简的拓扑信息。如果没有一个区域层次结构，它将很难在一个单一区域扩展如此巨大的拓扑信息，这个 LSA 并不携带任何拓扑信息，而只携带一个 IP 前缀。这个 LSA 由 ABR 产生，如下所示：

- 从一个非骨干区域 (nonbackbone) 到一个骨干区域 (backbone), 产生汇总 LSA 是为了:

- 直连路由 (connected routes)。
- 区域内部路由 (intra-area routes)。

注意: 仅仅区域内部路由才被广播进骨干区域以避免循环。如果有任何来自于非骨干区域的区域间路由则表示骨干区域是非邻接体的。非邻接体的骨干区域在 OSPF 网络中是不允许的。

- 从一个骨干区域到一个非骨干区域产生汇总 LSA 是为了:

- 直连路由。
- 区域内部路由。
- 区域间路由 (interarea routes)。

存在两种类型的汇总 LSA:

- 类型 3——用于有关网络的信息。
- 类型 4——用于有关 ASBR 的信息。

图 8-12 显示了汇总 LSA 的分组格式。

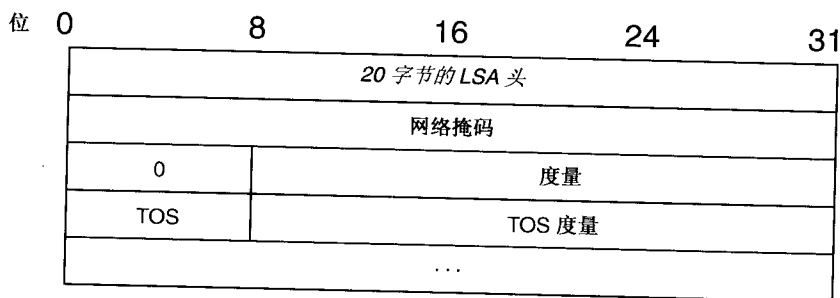


图 8-12 汇总 LSA 分组格式

下面的列表描述了汇总 LSA 分组中的字段:

- 网络掩码——对于类型 3 的汇总 LSA, 这个字段包含有与网络相关的网络掩码。对于类型 4 的汇总 LSA, 这个字段必须是 0。
- 度量——这个字段表示网络的代价。
- ToS 和 ToS 度量——这些字段通常设置为 0

类型 3 和类型 4 汇总 LSA 都使用同样的分组格式。关于类型 3 和类型 4 汇总 LSA 要记住的最重要的事情如下:

- 类型 3 中网络掩码包含有网络的子网掩码值。
- 在类型 4 LSA 中网络掩码字段必须是 0.0.0.0。
- 在类型 3 LSA 中, 链路状态 ID 字段必须有网络号。
- 在类型 4 LSA 中, 链路状态 ID 字段必须具有 ASBR 的路由器 ID。
- 通告路由器字段必须包含有产生汇总 LSA 的 ABR 的路由器 ID。这对类型 3 和类型 4 来说都是适用的。

有一个汇总 LSA 的特殊例子——当一个端区的 ABR 产生一个缺省路由汇总的情况。在这种情况下, 链路状态 ID 字段和网络掩码都必须是 0.0.0.0。

汇总 LSA 例子

例 8-3 显示了一个 Cisco 路由器的一个汇总 LSA 输出。

例 8-3 网络汇总 LSA 输出

```
RouterB#show ip ospf database summary 9.9.9.0
LS age: 1261
Options: (No TOS-capability, DC)
LS Type: Summary Links(Network)
Link State ID: 9.9.9.0 (summary Network Number)
Advertising Router: 141.108.1.21
LS Seq Number: 80000001
Checksum: 0xC542
Length: 28
Network Mask: /24
TOS: 0 Metric: 10
```

这里的链路状态 ID 字段是网络号 9.9.9.0，网络掩码是/24。在类型 3 汇总 LSA 中的链路状态 ID 字段总包含着产生这个汇总 LSA 的网络号和网络掩码。这里的汇总 LSA 是为 9.9.9.0 / 24 而产生的，如图 8-13 中所示。

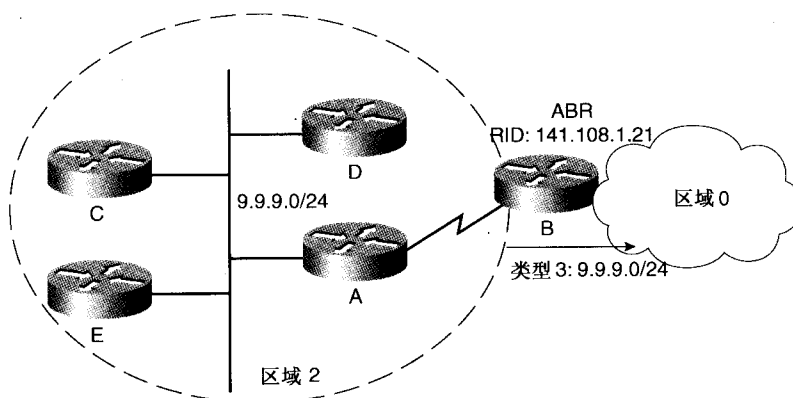


图 8-13 ABR 路由器在何处产生汇总 LSA 的网络图示

例 8-4 显示了 ASBR 汇总 LSA 的输出。

例 8-4 ASBR 汇总 LSA 的输出

```
RouterB#show ip ospf database asbr-summary 141.108.1.21
LS age: 1183
Options: (No TOS-capability, No DC)
LS Type: Summary Links(AS Boundary Router)
Link State ID: 141.108.1.21 (AS Boundary Router address)
Advertising Router: 141.108.1.1
LS Seq Number: 80000001
Checksum: 0x57E4
Length: 28
Network Mask: /0
TOS: 0 Metric: 14
```

例 8-4 中的输出说明这个汇总 LSA 是类型 4。网络掩码是 0，链路状态 ID 是 ASBR 的路由器 ID。在类型 4 的例子中，链路状态 ID 总是 ASBR 的路由器 ID。网络掩码字段总是 0，因为这是关于一个路由器（ASBR），而不是关于一个网络的信息。图 8-14 显示了基于例 8-4 中所显示的输出的网络图示。

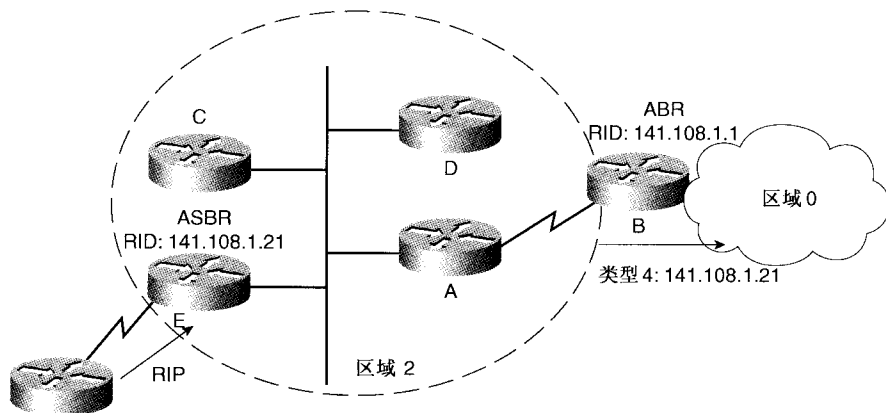


图 8-14 ABR 在何处产生类型 4 汇总 LSA 的网络图示

例 8-5 显示了 ASBR 汇总 LSA 的缺省输出。

例 8-5 汇总 LSA 的缺省输出

```
RouterB#show ip ospf database summary 0.0.0.0
LS age: 6
Options: (No TOS-capability, DC)
LS Type: Summary Links(Network)
Link State ID: 0.0.0.0 (summary Network Number)
Advertising Router: 141.108.1.21
LS Seq Number: 80000001
Checksum: 0xCE5F
Length: 28
Network Mask: /0
TOS: 0 Metric: 1
```

在例 8-5 中的输出显示链路状态 ID 和网络掩码都是 0.0.0.0。由于这是关于一条缺省路由的信息，所以它的链路状态 ID 必须是 0.0.0.0，网络掩码也必须是 0.0.0.0。这两条信息表示缺省路由是 0.0.0.0/0。这个缺省汇总将会出现在一个端区的情况下，如图 8-15 所示。

8.2.4 外部 LSA

外部 LSA 定义到自治系统外面的目的地址的路由。在域范围内，缺省路由也可以被作为一个外部路由引入（injected）。外部 LSA 通过除端区外的 OSPF 区域泛洪，为了要在路由表中建立一个外部 LSA，有两个关键的事情要发生：

- 进行计算的路由器必须通过区域内部或区域间路由发现 ASBR。这意味着在多区域情况下必须有一个为这个 ASBR 产生的路由器 LSA 或者一个为这个 ASBR 产生的类型 4 LSA。
- 转发地址必须通过一个区域内或区域间路由而获知。

图 8

下面

● 网

● E

类

路

● 转

为

是

发

—

—

—

—

● 外

通常

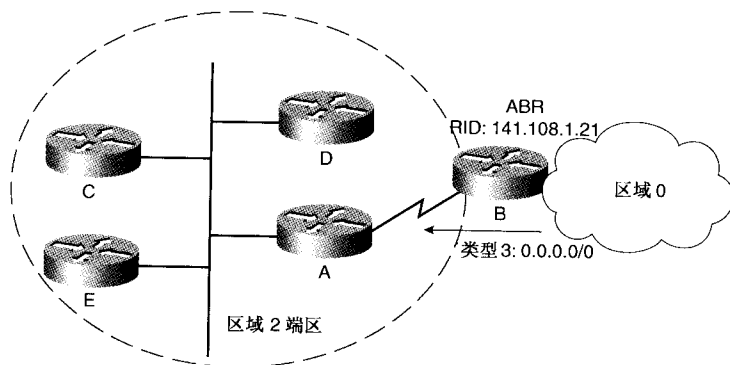


图 8-15 ABR 在何处产生一个缺省汇总 LSA 的图示

图 8-16 显示了外部 LSA 的分组格式。

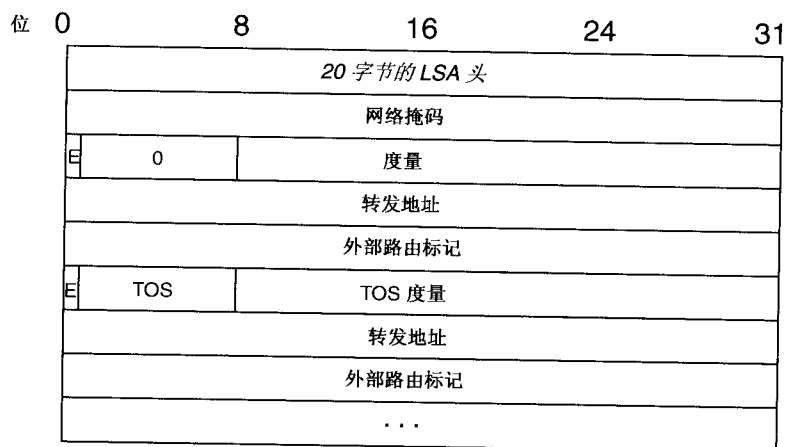


图 8-16 外部 LSA 分组格式

下面的列表描述了外部 LSA 分组中的字段。

- **网络掩码**——指定外部网络的网络掩码。
- **E 比特**——指定外部类型。如果设置了该字段，它就是一个外部类型 2，否则，它就是类型 1。在类型和外部类型之间的区别是类型 1 的度量和 OSPF 度量相似而且代价值每跳都改变；而在类型 2 中，外部度量不会改变。度量在同一 OSPF 域中都是一样的。
- **转发地址**——表示前往被通告网络的数据流量应该被转发的地址。如果这个值被设为 0.0.0.0，这表示流量必须被转发到 ASBR。在某些情况下，转发地址将不为 0，这是为了避免非优化路由 (suboptimal routing)。下面的列表描述了将产生一个非零转发地址的事件。
 - OSPF 在 ASBR 的下一跳接口上可用。
 - ASBR 的下一跳接口对 OSPF 来说是非被动 (nonpassive) 的。
 - ASBR 的下一跳接口的网络类型不是点到点或者点到多点类型。
 - ASBR 的下一跳接口地址是处于 OSPF 网络范围内的。
- **外部路由标记**——OSPF 没有使用它。

通常任何生产商都不使用 ToS 和 ToS 度量字段。

外部 LSA 例子

例 8-6 显示了 Cisco 路由器上的外部 LSA 的输出。

例 8-6 外部 LSA 的输出

```
RouterE#show ip ospf database external 10.10.10.0
LS age: 954
Options: (No TOS-capability, DC)
LS Type: AS External Link
Link State ID: 10.10.10.0 (External Network Number)
Advertising Router: 141.108.1.21
LS Seq Number: 80000003
Checksum: 0x97D8
Length: 36
Network Mask: /24
Metric Type: 2 (Larger than any link state path)
TOS: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0
```

在例 8-6 中的输出显示了网络 10.10.10.0/24 的一个外部 LSA。这是个类型 2 外部 LSA。这儿有一些重要的事情要记住：

- 链路状态 ID 字段表示外部网络号。
- 通告路由器字段包含了 ASBR 的路由器 ID。
- 度量类型：2 表示度量值（在这个例子中，度量值是 20）在通过整个 OSPF 域的过程中保持不变。
- 一个 0.0.0.0 转发地址表示流量应该直接转发给 ASBR。
- 通向非零转发地址的路由必须通过一个区域内部或者区域间路由而被了解；否则外部路由不会在路由表中被引入。

图 8-17 显示了一个网络，其中一个类型 5 的 LSA 由路由器 E (ASBR) 产生。RIP 被重分布 (redistributed) 进路由器 E，因此路由器 E 为每个 RIP 子网产生一个类型 5 LSA。这些类型 5 LSA 通过 OSPF 域传播。

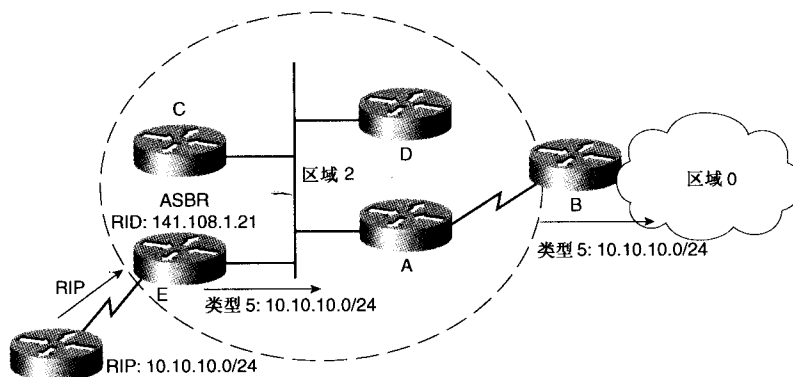


图 8-17 ASBR 在何处为一个通过 RIP 学习到的路由产生类型 5 LSA 的图示

8.3 OSPF 区域

OSPF 在一个区域中提供了两个等级的层次。一个区域是一个 32 比特的编号，它可以以 IP 地址的格式如“区域 0.0.0.0”来定义或者以十进制数字格式如“区域 0”来定义。区域 0 是一个骨干区域，如果配置了一个以上的区域，它就是必须存在的。所有的区域必须与区域 0 相连；否则，需要虚链路，如图 8-18 所示。

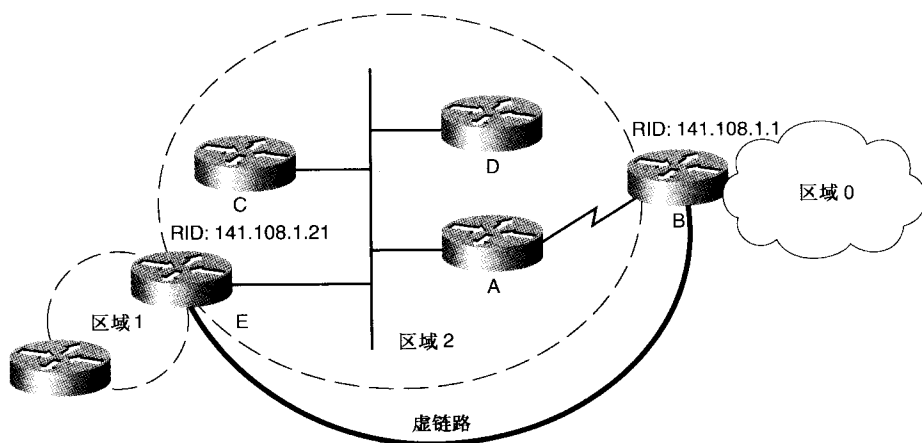


图 8-18 在一个不与主干相连的区域上使用一条虚链路

例 8-7 显示了在路由器 E 和路由器 B 之间使用一条虚链路所需要的配置。区域 2 是位于路由器 E 和 B 之间的中继区域。路由 E 要使用路由器 B 的路由器 ID 形成一条虚链路，反之亦然。建议你使用一个回环 IP 地址作为一个路由器 ID，因为回环链路总是开启的；因此，虚链接总能保持连通。

例 8-7 在路由器 E 和 B 之间配置虚链路

```
RouterE#
router ospf 1
area 2 virtual-link 141.108.1.1

RouterB#
router ospf area 2 virtual-link 141.108.1.21
area 2 virtual-link 141.108.1.21
```

虚链路其自身不是一件坏事情。一个不好的设计将包含一个不与区域 0 相连接的区域，如图 8-18 所示，这时就用一条虚链路来修补。虚链路在一些情况下非常有用。图 8-19 显示了一个例子，其中虚链路可以被用作一个备份或者用于冗余考虑——当路由器 A 和 B 之间的链路断开时，区域 3 的连通性不会被破坏。同样，如果路由器 C 和 D 之间的链路断开了，骨干区域通过虚链路仍然保持邻接。

例 8-8 显示了路由器 A、B、C 和 D 的配置。路由器 A 和 D 形成了一条位于彼此之间通过中继区域 2 的虚链路，而路由器 C 和 D 形成了一条虚链路，这条虚链路通过位于它们之间

的中继区域 1。路由器 A 和 B 之间的虚链路用于备份区域 3 的连通性；路由器 C 和 D 之间的虚链路用于在路由器 E 和 F 之间链路失效的时候作为区域 0 的备份。

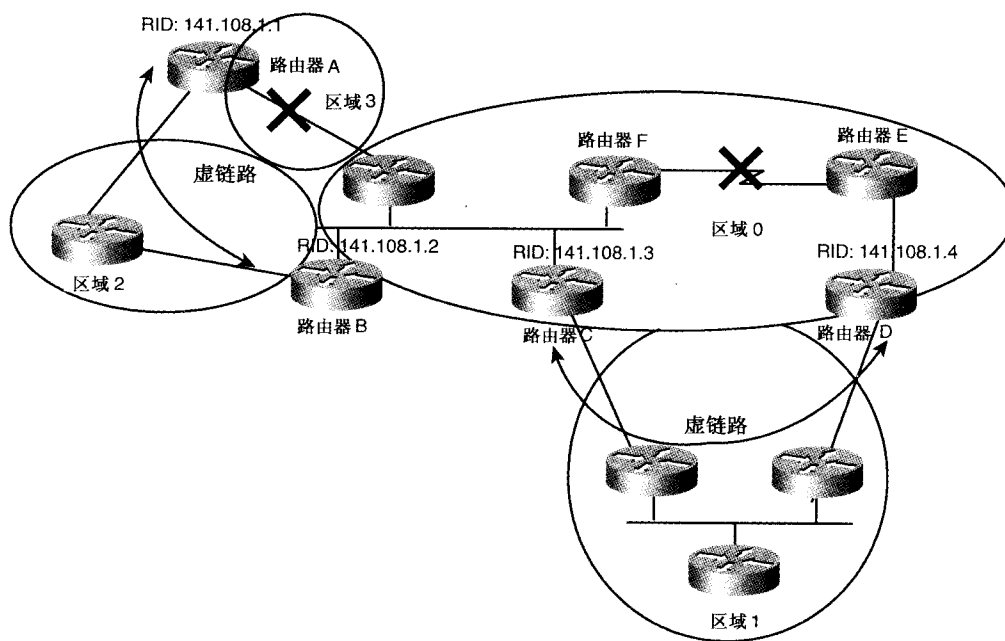


图 8-19 使用一条虚链路作为备份

例 8-8 在路由器 A、B、C 和 D 之间配置虚链路

```
RouterA#
router ospf 1
area 2 virtual-link 141.108.1.2
```

```
RouterB#
router ospf 1
area 2 virtual-link 141.108.1.1
```

```
RouterC#
router ospf 1
area 1 virtual-link 141.108.1.4
```

```
RouterD#
router ospf 1
area 1 virtual-link 141.108.1.3
```

图 8-20 显示了一个其它例子，其中一条虚链路可以用于路由优化。如果路由器 B 和 C 之间的链路被放置到区域 1 中，则区域 0 将受困于非优化路由，如果那条链路被放置到区域 0 中，那么区域 1 将受困于非优化路由。因此，解决的办法是将那条链路放置到区域 1 中，然后在路由器 B 和 C 之间配置一条虚链路，这样，它将同时承运主干区域和区域 1 的流量。

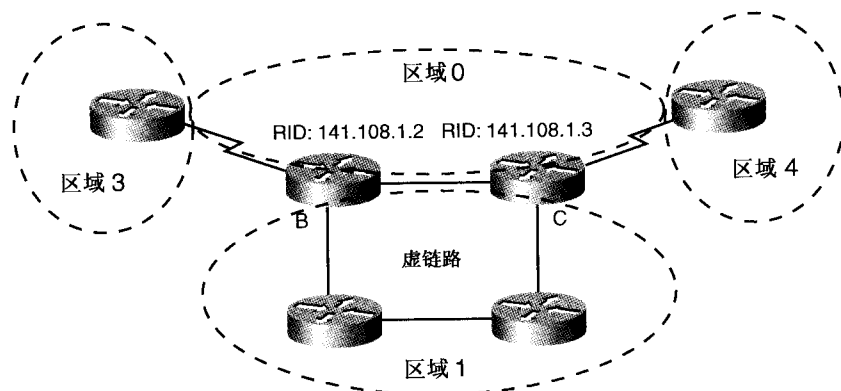


图 8-20 使用一条虚链路用于路径优化

例 8-9 显示了在路由器 B 和 C 之间形成一条虚链路所需的配置，这条虚链路需要用于路径优化。

例 8-9 配置路由器 B 和 C 来形成一条用于路径优化的虚链路

```
RouterB#
router ospf 1
area 1 virtual-link 141.108.1.3
```

```
RouterC#
router ospf 1
area 1 virtual-link 141.108.1.2
```

OSPF 有几种类型的区域，它们可以根据网络的需求来定义。

- 普通区域 (normal area)。
- 端区 (stub area)。
- 完全端区 (totally stubby area)。
- 非完全端区 (not-so-stubby area, NSSA)。
- 完全 NSSA 区域 (totally not-so-stubby area)。

下面几小节详细讨论各种不同的 OSPF 区域。

8.3.1 普通区域

当区域被缺省定义时，它被认为是普通或常规区域。普通区域有如下特性：

- 从其它区域来的汇总 LSA 被引入。
- 外部 LSA 被引入。
- 外部缺省 LSA 可以被引入。

在图 8-21 中，区域 1 和区域 2 是普通区域。IGRP 路由被重分布到区域 1 中，RIP 路由被重分布到区域 2 中。

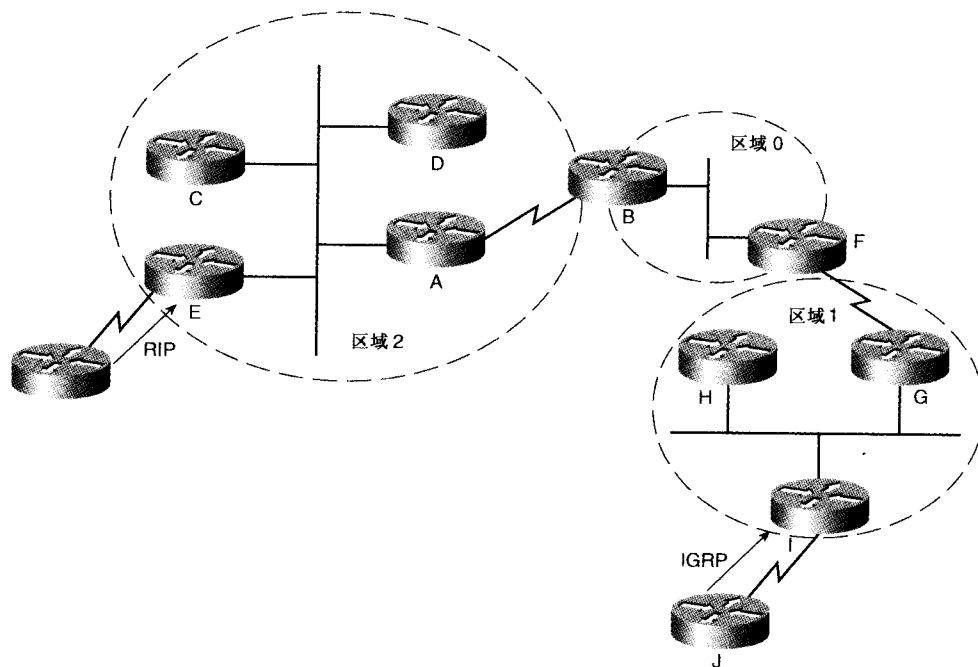


图 8-21 OSPF 普通区域例子

8.3.2 端区域

在端区域中，外部 LSA 是不允许的。回想在 OSPF Hello 分组中的选项字段，在选项字段中有一个比特是 E 比特。在端区的情形中，E 比特是空的，表明区域不能够引入任何外部 LSA。

端区有如下特性：

- 从其它区域来的汇总 LSA 被引入。
- 缺省路由作为一个路由汇总被引入。
- 外部 LSA 不被引入。

在图 8-22 中，区域 1 被定义为一个端区域。在路由器 I、H 或者 G 上不会发生路由重分布，因为端区域不允许类型 5 的 LSA。同样在路由器 E 上作为外部 OSPF 而引入的 RIP 路由被路由器 F 阻塞；可是，区域 1 仍然接收由路由器 F（ABR）产生的区域 2 的路由汇总。一个缺省汇总 LSA 也将被 ABR（路由器 F）引入到区域 1 中。这表示如果路由器 I、H 或者 G 需要发送一个分组给外部目的地，它们将总是把这个分组发送给最近的 ABR，在这个例子中 ABR 就是路由器 F。

例 8-10 显示了需要将区域 1 设置成端区所需要的配置。这个配置必须在区域 1 中所有的路由器上进行。

例 8-10 配置区域 1 作为一个端区

```
RouterF#
router ospf 1
area 1 stub
```

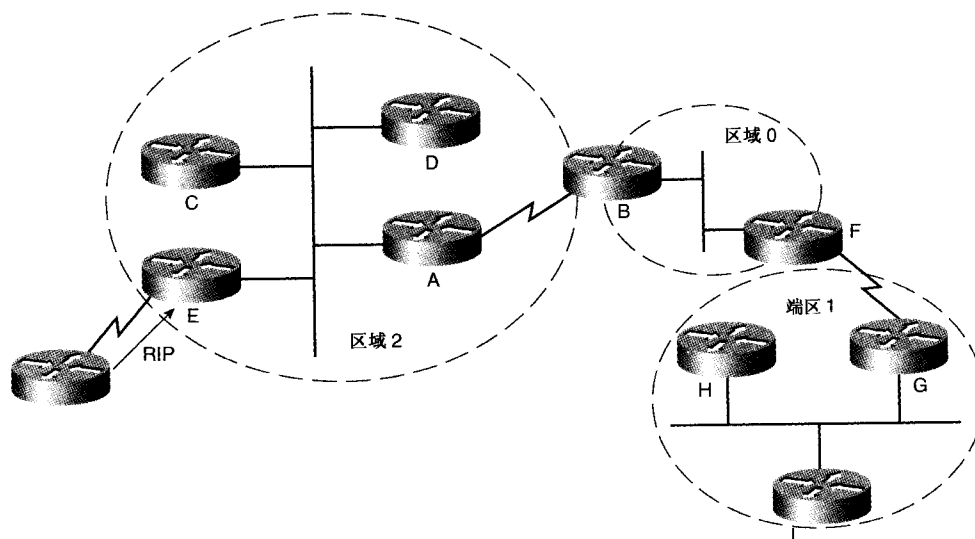


图 8-22 端区例子

8.3.3 完全端区域

完全端区域是区域中最受限的形式。在这种类型区域中的路由器仅仅依靠从 ABR 来的一个缺省路由汇总的引入。没有其它的外部或汇总信息被包含到路由表中。这是对端区的一个扩展，所以对这种区域来说，端区的所有特性仍是有效的。这个区域有以下特性：

- 不允许汇总 LSA。
- 不允许外部 LSA。
- 缺省路由作为一个路由汇总被引入。

在图 8-22 中，区域 1 将不接收任何路由汇总或任何外部路由。区域 1 拥有的唯一的路由是发往区域 1 的区域内部（在路由表中以“O”来标记）路由和由 ABR 引入的缺省路由，它以“OIA”来标记。

例 8-11 显示了为了将区域 1 设置为一个完全端区而需要在 ABR 上进行的配置。请注意端区和完全端区之间的差异在于没有汇总 LSA 被产生进入完全端区。由于汇总 LSA 的产生仅发生在 ABR 上，所以需要发生的配置改变仅仅在 ABR 之上。所有其它配置了一个端选项的路由器无需任何配置的改变。关键字 no-summary 表示避免在区域 1 中发送任何汇总 LSA。

例 8-11 配置 ABR（路由器 F）使区域 1 成为完全端区

```
RouterF#
router ospf 1
area 1 stub no-summary
```

8.3.4 非完全端区域

这也是端区的一个扩展。设想在图 8-22 中区域 1 被定义为一个端区域而且存在把一

条 IGRP 路由重分布进那个区域的要求。如果区域 1 被定义为端区，这将不可能发生。为了将 IGRP 路由重分布进区域 1，区域 1 必须被改变为 NSSA。当区域 1 变成 NSSA 的时候，它将允许路由重分布，这样 IGRP 路由可以作为类型 7 LSA 重分布进这个 NSSA 区域。

NSSA 是为了从端区引入外部路由到 OSPF 域而创建的。在 NSSA 中，当 ASBR 引入一条路由进入 AS 时，它产生一个类型 7 的 LSA。这个 ABR 将这个 LSA 转换成一个类型 5 的 LSA，它被传播到自治系统其余的部分中。类型 7 LSA 泛洪的范围是在 NSSA 区域内。

NSSA 从 Cisco IOS 软件版本 11.2 起开始支持。NSSA 有以下特性：

- 类型 7 LSA 在一个 NSSA 区域内携带外部信息。
- 类型 7 LSA 在 NSSA 的 ABR 上被转换成类型 5 LSA。
- 不允许外部 LSA。
- 汇总 LSA 被引入。

因为这是端区的一个扩展，所以 RIP 路由没有被作为 OSPF 外部路由引入区域 1；但是 IGRP 路由被转换成了类型 7 LSA。

例 8-12 显示了一个 NSSA 区域配置的例子。关键字 `nssa` 必须在属于区域 1 的一部分的所有路由器上键入，如例 8-12 所示。

例 8-12 在 NSSA 区域中的所有路由器上配置一个 NSSA

```
RouterF#  
router ospf 1  
area 1 nssa
```

1. 类型 7 LSA

类型 7 LSA 的分组格式和类型 5 LSA 的非常相似。它们的三个主要不同如下所示：

- 类型字段值为 7 而不是 5，表明它是类型 7 LSA。
- 转发地址计算如下：

如果这个路由有下一跳地址（对于直连路由是不正确的），试着使用它。这仅是在如果这条路由是一条 OSPF 内部路由的情况下才可能。在类型 5 转发地址选择中解释的所有东西对类型 7 来说也还是正确的。如果在类型 5 转发地址选择标准中解释的任何情况不再正确，则下一跳将不会被用作转发地址。下面两条规则应用于这种情况中：

——在宣布 LSA 的区域中使用回环地址中的一个（如果它开启了并且 OSPF 正在运行）。

——如果没有配置回环地址，则在那个区域中使用第一个接口的地址。

- P 比特在后面解释。

2. NSSA LSA 例子

例 8-13 显示了图 8-23 中 NSSA LSA 的输出。路由器 I 就是重分布 IGRP 到 OSPF 中的 NSSA ASBR。

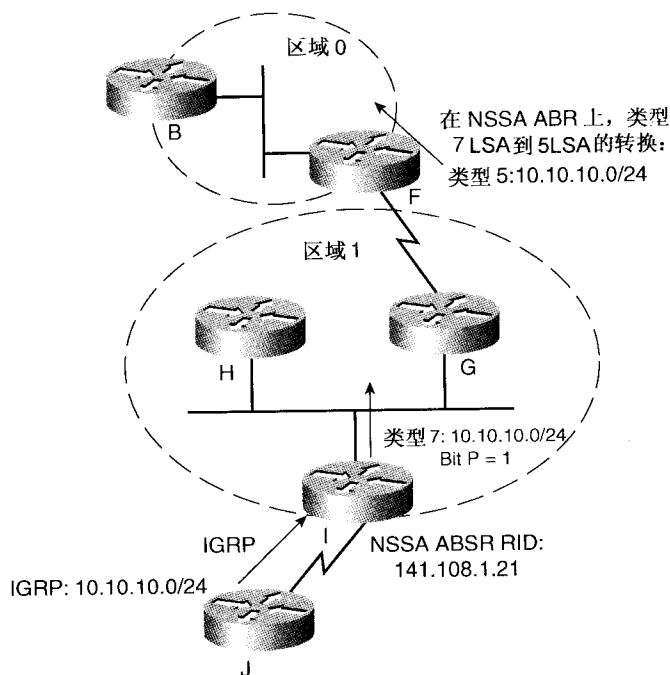


图 8-23 类型 7 LSA 在何处产生的网络图示

类型 7 LSA 被产生进入区域 1，接着被 NSSA ABR 转换成类型 5 LSA，这个 NSSA ABR 就是路由器 F。

例 8-13 NSSA LSA 输出

```
RouterI#show ip ospf database nssa-external 10.10.10.0
LS age: 36
Options: (No TOS-capability, Type 7/5 translation, DC)
LS Type: AS External Link
Link State ID: 10.10.10.0 (External Network Number)
Advertising Router: 141.108.1.21
LS Seq Number: 80000001
Checksum: 0x4309
Length: 36
Network Mask: /24
Metric Type: 2 (Larger than any link state path)
TOS: 0
Metric: 20
Forward Address: 141.108.1.21
External Route Tag: 0
```

NSSA LSA 的输出和外部 LSA 的输出类似，但是在这个输出中有一些关于 P 比特的事情要记住：

- P 比特用于告诉 NSSA ABR 是否要将类型 7 LSA 转换成类型 5 LSA。这个比特已经在选项字段中提到，这个选项字段在前面“Hello 分组”一节中讨论过。
- 没有类型 7 / 5 的转换表示 P 比特 = 0

- 类型 7 / 5 转换表示 P 比特=1
- 如果 P 比特=0, 那么 NSSA ABR 必须不把这个 LSA 转换成类型 5 LSA。这种情况在 NSSA ASBR 也是一个 NSSA ABR 的时候发生。
- 如果 P 比特=1, NSSA ABR (如果多个 NSSA ABR 存在, 则是有着最小路由器 ID 的那一个) 必须将这个类型 7 LSA 转换成一个类型 5 LSA。

P 代表传播 (propagation) 这个词。基本上, 这个比特用于传播控制。ABR 基于这个比特的值来做决定。

3. NSSA 配置例子

例 8-14 显示了定义一个 NSSA 区域的配置例子。这个配置必须在区域 1 中所有的路由器上出现, 如图 8-23 所示。

例 8-14 配置一个 NSSA

```
RouterF#
router ospf 1
area 1 nssa
```

在图 8-23 中定义区域 1 作为一个 NSSA 之后, 它将有如下特性:

- 在区域 1 中不允许类型 5 LSA。这表示在区域 1 中不允许 RIP 路由。
- 所有 IGRP 路由被作为类型 7 路由重新分发。这种类型 7 路由只能存在于 NSSA 中。
- 所有类型 7 LSA 被 NSSA ABR 转换成类型 5 LSA, 并且作为类型 5 LSA 漏进 OSPF 域。

4. 完全 NSSA 区域

这种类型的区域是 NSSA 的一个扩展。如果只有一个出口点存在的话, 这就是 NSSA 区域类型的最佳推荐形式。在图 8-23 中, 如果区域 1 被定义为一个完全 NSSA, 则下面这些就是真的:

- 不允许 RIR 路由引入区域 1, 因为这些路由是外部路由。
- 不允许从其它区域来的汇总 LSA 引入区域 1, 因为根据一个完全 NSSA 的定义这是不允许的。
- 缺省的汇总 LSA 将由 ABR, 即路由器 F 产生。

完全 NSSA 有如下特性:

- 不允许汇总 LSA。
- 不允许外部 LSA。
- 缺省路由被作为一个路由汇总来引入。
- 在 NSSA ABR 上类型 7 LSA 被转换成类型 5 LSA 的形式。

例 8-15 显示了在 NSSA ABR 上所需的配置。就如在完全端区的情形中一样, no-summary 命令仅仅在 ABR 上需要, 因为汇总 LSA 是在 ABR 上产生的。

例 8-15 在 NSSA ABR (路由器 F) 上配置完全 NSSA 区域

```
RouterF#
router ospf 1
area 1 nssa no-summary
```

5. 在 NSSA 中过滤

在某些情况下，不需要将外部路由作为类型 7 路由引入 NSSA。这种情况通常发生在一个 ASBR 同时也是 NSSA ABR 的时候。

当在这种情形中发生了路由重分布的时候，路由器产生类型 5 LSA 和类型 7 LSA。在图 8-24 中，区域 1 使用了 **no-redistribution** 选项配置的

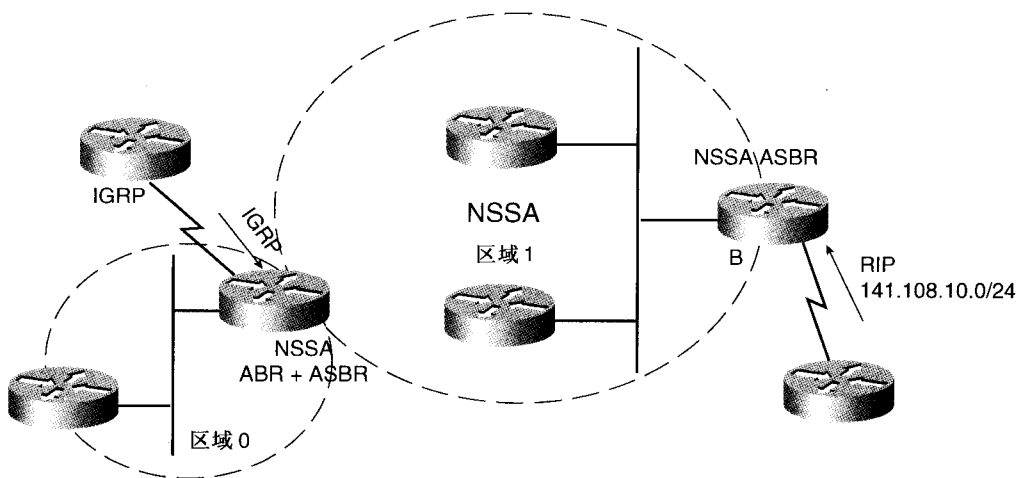


图 8-24 可以使用 NSSA 过滤的情形

这表示所有 IGRP 路由被重分布进了区域 0，但是不会有为区域 1 产生的类型 7 LSA。例 8-16 显示了阻止 NSSA ABR 路由器 A 为 IGRP 路由产生类型 7 LSA 的配置。

例 8-16 在 NSSA 中用来过滤类型 7 LSA 的配置

```
RouterA#
router ospf 1
area 1 nssa no-redistribution
```

在 NSSA ABR（它也是一个 ASBR）上配置 **no-redistribution** 命令。

过滤的另一个例子是当你需要阻止类型 7 LSA 在 NSSA 之外转换的时候发生。换句话说，你想控制哪个类型 7 LSA 转换成类型 5 LSA。例如，图 8-24 显示了一个通过 RIP 学习的路由 141.108.10.0/24 被引入 OSPF NSSA 区域 1。你不想让这条路由漏入 OSPF 区域的其它部分。例 8-17 显示了将阻止 RIP 路由转换成类型 5 LSA 的配置。这个配置在路由器 A 或 B 上都能使用。

例 8-17 控制类型 7 到类型 5 的转换的配置

```
RouterA#
router ospf 1
summary-address 141.108.10.0 255.255.255.0 not-advertise
```

这个 **summary-address** 配置产生一个类型 7 LSA，它不会被 NSSA ABR 转换成一个类型 5 LSA。

6. NSSA 中的缺省路由

有两种方式在一个 NSSA 中拥有一条缺省路由:

- 当你配置一个区域作为 NSSA 区域的时候, 这个 NSSA ABR 缺省时不产生缺省路由汇总。
- 假设是端区或者 NSSA 或者完全端区的情况下, 这个 NSSA ABR 产生一个缺省路由汇总。

(1) 缺省路由汇总

通过定义一个区域作为 NSSA 区域或者完全端区, 这个 NSSA ABR 产生一条缺省路由。如前面提到的, 如果这个 NSSA 区域没有定义为完全端区, 则 NSSA ABR 将不会产生一条缺省汇总路由。例 8-18 显示了如何通过配置 NSSA 区域来在 NSSA 区域或者完全端区中发送缺省汇总路由, 这是通过在 NSSA ABR 上应用 **no-summary** 选项来完成的。

例 8-18 用来产生缺省汇总路由进入 NSSA 区域的配置

```
RouterA#  
router ospf 1  
area 1 nssa no-summary
```

(2) 缺省类型 7

例 8-19 显示了产生一个类型 7 缺省路由的配置。你可以在任何 NSSA ASBR 或者 NSSA ABR 上配置这个命令, 遵循以下规则:

- NSSA ASBR 只有当它在路由表中有一条缺省路由时才能产生一条缺省路由。
- NSSA ASB 在有或者没有一条缺省路由在它的路由表中时都能产生一条缺省路由。

例 8-19 产生类型 7 缺省路由到一个 NSSA 区域的配置

```
RouterA#  
router ospf 1  
area 1 nssa default-information-originate
```

8.4 OSPF 介质类型

OSPF 在多种介质类型上运行。在某些介质上, 例如多点接入介质或者点到点介质, 使用了 OSPF 缺省网络类型。因此, 没有必要在这些介质上配置任何网络类型。

本节深入 OSPF 中的每种介质类型细节并且介绍对每种介质使用何种网络类型。对 OSPF 来说, 介质可分成四类:

- 多点接入介质。
- 点到点介质。
- 非广播多点接入介质。
- 请求线路。

8.4.1 多点接入介质

多点接入介质包括以太网、快速以太网、吉比特以太网、FDDI、令牌环和类似的多点接入介质。OSPF 作为一个广播网络类型运行在这些介质之上。缺省时，OSPF 在这些介质上的广播网络类型就是配置好的。

在这种网络类型中，DR 和 BDR 被选举用于在网段中减少泛洪。OSPF 的组播能力用于形成邻接体关系和在网段上有效地分发信息到其它路由器上。在广播网络类型中，接口子网掩码在 Hello 分组中检查。如果两个路由器的掩码不同，则邻接体关系将不能形成。

因为这个网络类型是缺省决定的，所以对这种介质不需要特殊配置。图 8-25 显示了 OSPF 在多点接入介质上运行的例子。路由器 A 被选举为 DR 因为它有着高的优先级。路由器 B 被选举作 BDR。路由器 B 和 C 的优先级是相同的；因此，BDR 的选举是基于最大的路由器 ID。所有路由器都将与 DR 和 BDR 形成一个邻接体关系。DR 和 BDR 将特别监听组播地址 224.0.0.6（所有 DR 路由器），而所有的路由器将监听组播地址 224.0.0.5（所有的 SPF 路由器）。

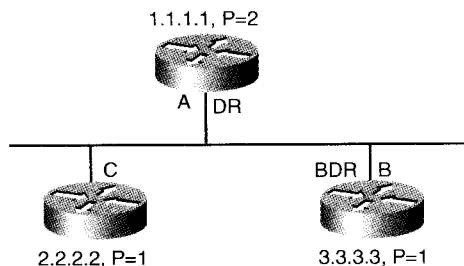


图 8-25 多点接入介质的例子

8.4.2 点到点介质

点到点介质包括 HDLC 和 PPP 封装链路、帧中继/ATM 点到点接口和类似的点到点接口。

在这些介质上 OSPF 网络类型缺省时是配置为点到点类型的。在这种介质类型上没有 DR 或者 BDR 选举发生。所有的 OSPF 分组都是基于组播的。将 OSPF 分组以组播发送的理由是，在某些没有编号的链路上，目的地址是不知道的。图 8-26 显示了一个点到点介质的例子。路由器 A 通过点到点链路与其它 3 个路由器相连。路由器 A 将与路由器 B、C 和 D 形成一个全邻接体关系。

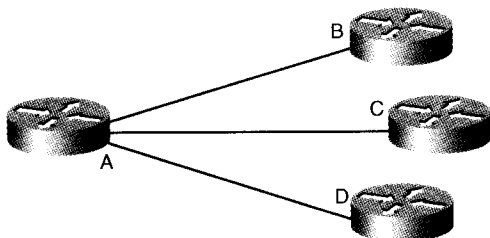


图 8-26 点到点介质的例子

8.4.3 非广播多点接入介质

许多介质是属于非广播多点接入类型 (NBMA) 的, 包括帧中继、X.25、SMDS 和 ATM。这种类型介质需要附加的配置。在这些介质上的 OSPF 网络缺省是非广播的。在这种情况下有多个网络类型选项可以被定义。这种介质可以在 OSPF 下以多种模式运行:

- 广播模式。
- 点到点模式。
- 点到多点模式。

1. 广播模式

在广播模式中, 广播网络是模拟的。DR 和 BDR 被选举出来。广播模式可以用两种方式模拟:

- 配置 **network-type** 广播。
- 配置 **neighbor** 语句。

图 8-27 显示了 NBMA 网络模式。在路由器之间的介质是帧中继。路由器 A 与所有其它路由器相连; 而路由器 B、C、D 和 E 却只与路由器 A 相连, 而不是互相连接, 这种连接是通过帧中继 PVC 来实现的。在这广播模式工作的唯一机会是当所有路由器形成完全网状互连 (fully meshed) 的时候。在一个部分网状互连的情况下, 如图 8-27 所示, 不推荐运行广播模式。

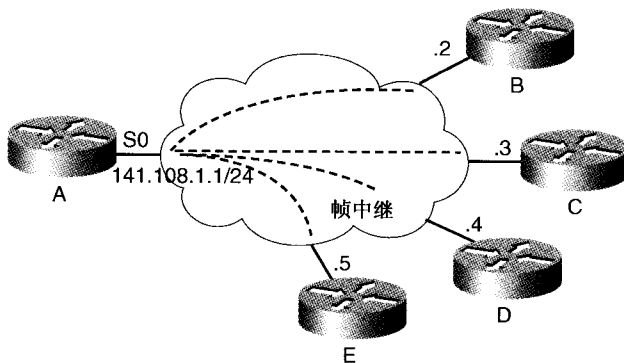


图 8-27 NBMA 介质例子

假设有形成完全网状互连的帧中继连接, 例 8-20 显示了在所有路由器上所需的配置。这个配置将帧中继接口的网络类型改成广播。这里有一个要注意的重要问题是, 在一个静态帧中继映射的情况下, 必须在最后用上关键词 **broadcast**; 否则, OSPF 组播 Hello 分组不会通过帧中继连接传送。

例 8-20 配置网络类型为广播类型

```
RouterA#  
interface serial 0  
  encapsulation frame-relay  
  ip ospf network-type broadcast
```

命令 **ip ospf network-type broadcast** 必须在所有路由器的帧中继接口上配置。

另一个模拟广播模式的办法是通过 **neighbor** 语句, 如例 8-21 所示。**neighbor** 语句必须在 hub 路由器上手动配置, 在这个例子中这个 hub 路由器是路由器 A。路由器 A 应该配置有一个高的优先级使它总是 DR 路由器。

例 8-21 在 hub 路由器上配置 neighbor 语句

```
RouterA#
interface serial 0
 encapsulation frame-relay
 ip address 141.108.1.1 255.255.255.0
 ip ospf priority 10
!
router ospf 1
 neighbor 141.108.1.2
 neighbor 141.108.1.3
 neighbor 141.108.1.4
 neighbor 141.108.1.5
```

2. 点到点模式

如果使用点到点模式, 每个 PVC 必须被定义为一个分离的点到点接口。因此, 对每个点到点接口需要一个分离的子网。例 8-22 显示了在 hub 路由器 A 上创建点到点接口所需的配置。在子接口中不需要任何网络类型, 因为缺省情况下所有子接口都有一个点到点网络类型。物理拓扑仍和图 8-27 中的一样。

这种模式的优点是虚电路 (VC) 代价可以在一个每接口的基础上配置。缺点是在每个点到点接口有许多地址空间被浪费了。另外, 由路由器 A 创建的路由器 LSA 的大小将会相当大, 因为它必须为每条点到点链路包括一个类型 3 端链路。

例 8-22 配置点到点接口

```
RouterA#
Interface Serial 0.1 point-to-point
 ip address 141.108.1.1 255.255.255.252
!
Interface Serial 0.2 point-to-point
 ip address 141.108.1.5 255.255.255.252
```

类似地, 还可创建其它子接口。

3. 点到多点模式

在点到多点模式中, 每一个与另一个路由器连通的路由器与那个路由器形成一个邻接体关系。在这种网络类型中没有选举 DR 或 BDR。发送组播 Hello 分组以发现邻居, 所以关键是第 2 层必须具有支持组播/广播的能力。同时, 对于整个 NBMA 云状网络, 只需一个子网, 如图 8-27 所示。

例 8-23 显示了点到多点网络类型所需的配置。为点到多点网络类型设置了路由器 A 的串行接口 0。这个网络类型必须在所有远端路由器 B、C、D 和 E 上配置。

例 8-23 用于点到多点网络类型的配置

```
RouterA#  
interface serial 0  
encapsulation frame-relay  
ip address 141.108.1.1 255.255.255.0  
ip ospf network-type point-to-multipoint
```

这是在非完全网状互连 NBMA 网络中推荐使用的网络类型。有时，一种介质是 NBMA 但不能支持组播，因此点到多点的网络类型不能使用。对这种类型的介质，引入了另一种网络类型，叫做点到多点非广播类型。假设在图 8-27 中 NBMA 网络不支持组播功能，这种新类型就可以使用。例 8-24 显示了这种网络类型所需的配置。这个网络类型仅配置在路由器 A 串行接口 0 上，但它必须在所有远端路由器串行接口上配置。这种网络类型需要 **neighbor** 语句，但是 DR 和 BDR 不会被选举。

例 8-24 配置点到多点非广播网络类型

```
RouterA#  
interface serial 0  
encapsulation frame-relay  
ip address 141.108.1.1 255.255.255.0  
ip ospf network-type point-to-multipoint non-broadcast  
!  
router ospf 1  
neighbor 141.108.1.2  
neighbor 141.108.1.3  
neighbor 141.108.1.4  
neighbor 141.108.1.5
```

8.4.4 请求线路

请求线路选项由 Cisco IOS 软件版本 11.2 引入。使用请求线路，形成了一个邻接体关系并且所有数据库在最初被交换。这样在死亡计时器到期之后，邻接体关系甚至在第 2 层连接断开的情况下仍然保持。当你不得不为保持使用这条链路而付出不必要的开销的时候这会很有用。ISDN 就是一个可以使用请求线路的情况的最好例子。只要 ISDN 链路是连通的，你就必须为这条链路付费。

使请求线路不同于普通链路的主要特性如下：

- 周期性的 Hello 分组被禁止。
- 周期性的 LSA 刷新被禁止。

周期性的 Hello 分组仅在点到点和点到多点网络类型上被禁止。所有其它网络类型的 Hello 分组仍在接口上发送。

每 30 分钟发生一次的周期性的 LSA 刷新不会在请求线路上发生，因为当请求线路连接建立起来后，一个独特的选项比特——DC 比特或者 Do Not Age(DNA)比特（早先在“Hello 分组”一节中讨论过了），被发送穿越线路。如果两个路由器协商 DC 比特成功了，它们对它作记录同时在 LSA 年龄字段中设置一个特别的比特，这是最重要的比特。通过设置这个比特，

LSA 在请求线路上停止计算年龄，因此没有周期性的更新被发送。

在两个普通情况中 LSA 的周期性的刷新被发送：

- 如果网络拓扑有变化。
- 如果在 OSPF 域中有一个路由器不理解请求线路。

在第一种情形中，不需要做更多事情因为路由器必须发送新的 LSA 信息去更新邻居关于拓扑的改变。可是在第二种情况下，有一种特别的方式来处理这种情形。

ABR 路由器，在图 8-28 中处于 A 和 C 之间的路由器，知道路由器 C 不能理解 DNA LSA 因为它看到了由路由器 C 产生的 LSA 中的选项同时因为在那个选项字段中 DC 比特被清除了。当这种情况发生时，ABR 指示有请求线路能力的路由器不要产生设置了 DNA 比特的 LSA，因为有一个不理解 DNA 比特的路由器。ABR 在骨干上产生一个指示 LSA (indication LSA)。告诉在骨干上的每个路由器不要产生任何 DNA LSA。这个指示 LSA 显示在例 8-25 中。这是一个类型 4 汇总 LSA，其中链路状态 ID 是 ABR 自己而不是 ASBR。

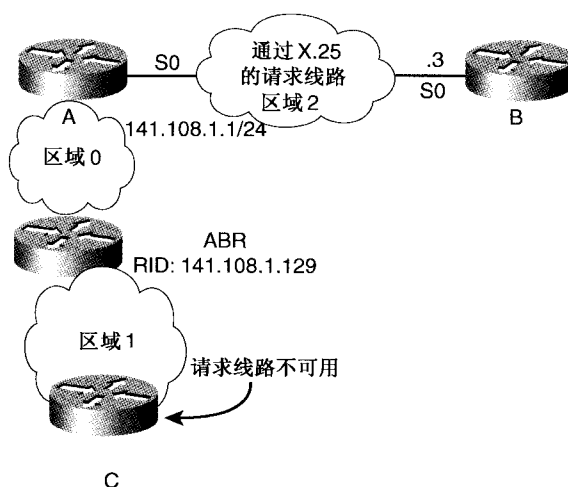


图 8-28 周期性 LSA 刷新将被通过一个请求线路发送的情形

例 8-25 指示 LSA 的例子

```
RouterA#show ip ospf database asbr-summary
Adv Router is not-reachable
LS age: 971
Options: (No TOS-capability, No DC)
LS Type: Summary Links(AS Boundary Router)
Link State ID: 141.108.1.129 (AS Boundary Router address)
Advertising Router: 141.108.1.129
LS Seq Number: 80000004
Checksum: 0xA287
Length: 28
Network Mask: /0
TOS: 0 Metric: 16777215
```

指示 LSA 的度量值被设为无穷，而且链路状态 ID 总是产生指示 LSA 的 ABR 的路由器 ID。在图 8-28 中，在路由器 A 和 B 之间的链路被配置为一条请求线路，但是因为在区域 1

每个邻居都能变成邻接体关系。Hello 协议负责来建立和维持邻接体关系。

Hello 分组被周期性地从所有路由器的功能接口发出。当路由器被列入邻居的 Hello 分组中时 two-way 通信被建立起来了。在广播和 NBMA 网络中, Hello 分组用于选举 DR/BDR。

在 two-way 通信建立之后, 就做出决定是否和邻居形成一个邻接体关系。这个决定基于邻居状态和网络类型。如果网络类型是广播或非广播的, 则仅仅与 DR 和 BDR 路由器建立邻接体关系。在所有其它网络类型中, 在两个邻居路由器之间建立邻接体关系。

建立邻接体关系的第一步是数据库的同步。每个路由器在 DBD 分组中描述它的链路状态数据库。在邻居间仅是交换 LSA 头。主和从设备的选举在数据库交换过程中发生。每个路由器对它在 DBD 交换过程中收到的 LSA 头进行记录。在 DBD 交换的最后, 它发送 LS 请求分组来请求那些在 DBD 交换期间看见了 LSA 头的 LSA。邻居路由器接着以 LS 更新分组回复, 其中列出了那些 LSA 整个的内容。这个 LS 更新分组通过发送链路状态确认分组而被确认。此时, 所有的数据库都交换了, 邻居进入 Full 状态阶段。

一个路由器可以有多个邻居状态:

- 关闭 (down)。
- 尝试 (attempt)。
- 初始 (init)。
- 2-way 状态。
- Exstart 状态。
- 交换 (exchange)。
- 调入 (loading)。
- Full 状态

下面几小节描述不同的 OSPF 状态细节。

8.5.1 OSPF 关闭状态

在图 8-29 中, R1 和 R2 运行 OSPF。邻居状态显示为关闭状态。这个状态表示还没有从邻居收到任何信息。



图 8-29 OSPF 关闭状态

8.5.2 OSPF 尝试状态

尝试状态对 NBMA 网络上的邻居是有效的。如果一个邻居处于这种状态, 则表示从它的邻居没有收到任何信息, 但是它做出了认真的努力来与邻居联系。认真的努力 (serious effort) 意思是这个路由器将不断地每隔一个 Hello 间隙就发送一个 Hello 分组来与邻居联系。

中有一个路由器不能理解 DNA LSA，所以不允许由区域 2 产生 DNA LSA。作为结果，LSA 的周期性刷新将被通过请求线路传送。

配置区域 1 为端区或者 NSSA 区域来作为一个解决方案。这是因为 Demand Circuit RFC 1793 的 2.5.1 节这么规定：“在常规 OSPF 区域中的 LSA 被允许有 DoNotAge 设置当且仅当在 OSPF 域中的每个路由器（除了那些在端区和 NSSA 中的）都能进行 DoNotAge 处理的情况下”。因此，如果区域 1 被定义为端区或者 NSSA，在区域 2 中的 LSA 将被允许让 LSA 设置 DNA 比特。

另一个建议是在一个非骨干区域中总包含一条请求线路。如果如图 8-28 中所示的相似的情形发生并且请求线路也是骨干区域的一部分时，就没办法来解决这个问题，因为骨干区域不能配置为端区或者 NSSA 区域。在一个端区配置请求线路的好处是如果在一个远端 OSPF 区域中的一些路由器不理解 DNA LSA，则由 ABR 产生的指示 LSA 将永远不能进入端区。因此，有请求线路能力的路由器仍可以在区域中产生设置了 DNA 比特的 LSA。例 8-26 显示了使一条链路成为一条请求线路所需要的配置。仅有一边需要在接口上运行 **demand circuit** 命令，因为，如果另一边有能力理解请求线路，它就会自动在 Hello 分组中协商这种功能，否则，它简单地忽略这个选项。

例 8-26 配置一条请求线路

```
RouterA#
interface serial 0
 encapsulation frame-relay
 ip address 141.108.1.1 255.255.255.0
 ip ospf network-type point-to-multipoint
 ip ospf demand-circuit
```

请求线路可以在任何网络类型上运行。区别在于，如果不是点到点或点到多点网络类型，Hello 分组将被禁止；可是，这些介质类型仍然能使用请求线路泛洪的健壮性（robustness），并且 LSA 不会在这个请求线路连接上超过年龄限制。

8.4.5 OSPF 介质类型汇总

表 8-4 说明了每种介质类型的缺省值，同时给出了推荐的网络类型。

表 8-4 介质类型和可能的 OSPF 网络类型

介质类型	缺省 OSPF 网络类型	推荐的 OSPF 网络类型
多点接入	广播	广播
点到点	点到点	点到点
非广播多点接入	非广播	点到多点
请求线路	—	点到多点

8.5 OSPF 邻接体

OSPF 在邻居路由器之间创建邻接体关系是为了交换路由信息。在一个广播环境中并非

在图 8-30 中, R1 正在发送一个 Hello 分组表明它还没有看到其它路由器并且不知道 DR 路由器。

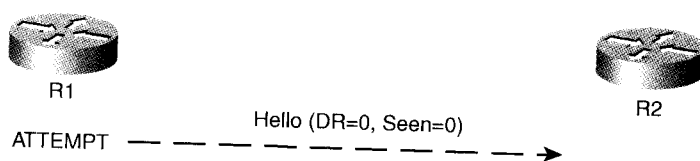


图 8-30 OSPF 尝试状态

8.5.3 OSPF 初始状态

初始状态是一个 one-way Hello。在图 8-31 中, R1 发送一个 Hello 分组。一旦收到这个 Hello 分组, R2 就宣布进入 one-way 状态, 因为 R2 在那个 Hello 分组中没有看到它自己 (它的路由器 ID)。

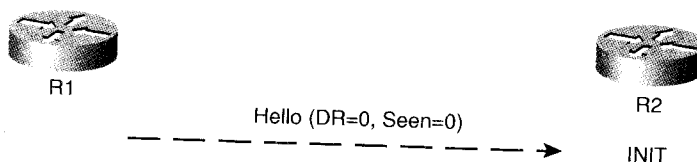


图 8-31 OSPF 初始状态

8.5.4 OSPF 2-way 状态

一个 OSPF 邻居当双向通信建立后就到达了 2-way 状态。这是一个 OSPF 邻接体关系形成的开始。在这个状态中选举出了 DR 和 BDR。在图 8-32, R2 发送一个 Hello 分组说 R2 已经看见了 R1 的 Hello 分组; R2 的路由器 ID 要大一些, 所以它也就选自己作为一个 DR。

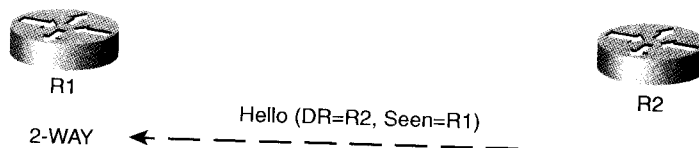


图 8-32 OSPF 2-way 状态

8.5.5 OSPF Exstart 状态

这个状态被用于数据库同步处理的初始化过程。在这个状态中选举出了主从设备。用于 DBD 交换的第一个序列号也在这个状态中决定。

在图 8-33 中, R1 发送第一个 DBD 分组。R2 也发送它的第一个 DBD 分组。有着最大路由器 ID 的路由器成为主设备。在这个例子中, R2 有一个最大的路由器 ID, 所以 R2 是

主设备。

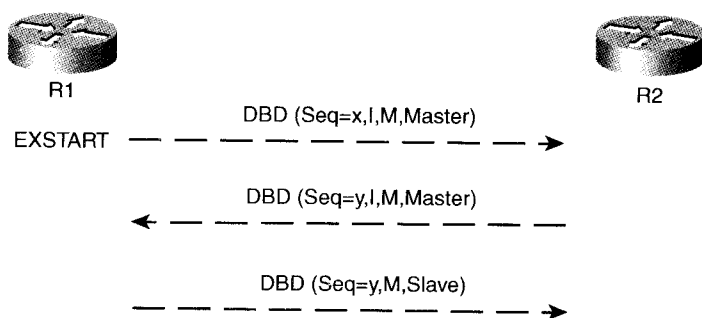


图 8-33 OSPF Exstart 状态

8.5.6 OSPF 交换状态

在交换状态中，路由器通过 DBD 分组描述它的整个链路状态数据库。每个 DBD 序列号被明白地确认。每次只有一个突出的 DBD 分组被允许。链路状态请求分组也在这种状态中被发送来请求一个新 LSA 的实例。图 8-34 显示了在活动中的交换过程。R1 和 R2 正交换它们的数据库信息。最后的箭头说明 M 比特被设置为 0。这表示主设备没有更多的数据要传送了。在这个 R1 阶段，从设备将发送数据库中所有剩下的数据，而且 R1 也将把 M 比特设为 0。这是所有路由器都已经交换完数据库的标志。

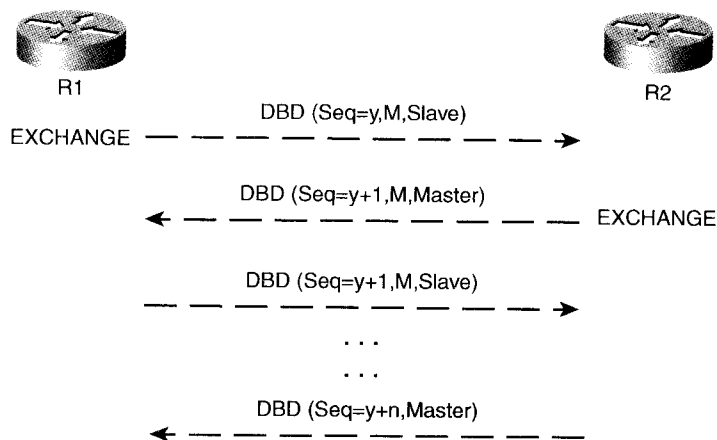


图 8-34 OSPF 交换状态

8.5.7 OSPF 装入状态

在装入状态中，LS 请求分组被发送去请求一个在交换过程中没有收到的 LSA 的最近的实例。在图 8-35 中，R1 处于调入状态并且正发送 LS 请求分组去接收 LSA 的一个最近的实例。