

目 录

第 1 章 理解 IP 路由	3
1.1 IP 编址的概念	4
1.1.1 IPv4 地址分类	5
1.1.2 IPv4 私有地址空间	6
1.1.3 子网化和可变长子网掩码	6
1.1.4 无类域间路由	8
1.2 静态和动态路由	9
1.3 动态路由	9
1.3.1 单播 IP 路由与组播 IP 路由	10
1.3.2 无类 IP 路由协议与有类 IP 路由协议	11
1.3.3 内部网关协议和外部网关协议	12
1.3.4 距离向量协议和链路状态协议	14
1.4 路由协议的管理距离	18
1.5 路由器的快速转发	19
1.6 小结	19
1.7 复习题	19
1.8 参考文献	20
第 2 章 理解 RIP 协议	23
2.1 度量	24
2.2 计时器	24
2.3 水平分割	24
2.4 带毒性逆转的水平分割	25
2.5 RIP-1 分组格式	25
2.6 RIP 行为	25
2.6.1 发送更新的 RIP 规则	25
2.6.2 接收更新的 RIP 规则	26
2.6.3 发送更新举例	27
2.6.4 接收更新举例	28
2.7 为什么 RIP 不支持非连续网络	29

考给

是一

于这

2.8 为什么 RIP 不支持可变长子网掩码.....	30
2.9 缺省路由和 RIP	31
2.10 RIP 协议扩展	32
2.10.1 路由标记	32
2.10.2 子网掩码	33
2.10.3 下一跳	33
2.10.4 组播能力	34
2.10.5 认证	34
2.11 兼容问题	34
2.12 小结	35
2.13 复习题	35
2.14 深入阅读	36
第 3 章 RIP 协议疑难解析	39
3.1 解决 RIP 常见问题的流程图.....	40
3.2 RIP 路由安装疑难解析	43
3.2.1 问题: RIP 路由不在路由表中	43
3.2.2 问题: RIP 没有安装所有可能的等价路径 ——原因: 命令 maximum-path 限制 RIP 安装多条路径	71
3.3 RIP 路由公布疑难解析	73
3.3.1 问题: 发送方没有公布 RIP 路由	73
3.3.2 问题: R2 的路由表中缺少子网路由——原因: 使用了自动汇总功能	89
3.4 RIP 路由汇总疑难解析	92
3.4.1 问题: RIP-2 路由表过大——原因: 关闭了自动汇总	92
3.4.2 问题: RIP-2 路由表过大——原因: 没有使用 ip 汇总地址	94
3.5 RIP 重发布问题疑难解析	95
3.6 按需拨号路由问题疑难解析	98
3.6.1 问题: RIP 广播占用 ISDN 链接 ——原因: 在关注流量定义中没有拒绝 RIP 广播	99
3.6.2 问题: RIP 更新没有通过拨号接口 ——原因: 在 dialer map 语句中缺少关键字 broadcast	102
3.7 RIP 中路由抖动问题疑难解析	103
第 4 章 理解 IGRP 协议	107
4.1 度量	107
4.2 计时器	109
4.3 水平分割	110
4.4 使用带有抑制逆转的水平分割	110
4.5 IGRP 分组格式	110
4.6 IGRP 行为	111

4.7
4.8
4.9
4.10
第 5 章
5.1
5.2
5.2
5.2
5.3
5.3
5.3
5.4
5.5
5.5
5.5
5.6
5.7
第 6 章
6.1
6.2
6.3
6.4
6.5
6.6
6.7
6.8
6.9
6.10
6.11
6.12
6.13
第 7 章

4.7	缺省路由与 IGRP	111
4.8	IGRP 中的非等价负载平衡	112
4.9	小结	114
4.10	复习题	114
第 5 章	IGRP 协议疑难解析	117
5.1	解决 IGRP 常见问题的流程图	118
5.2	IGRP 路由安装疑难解析	121
5.2.1	问题: IGRP 路由不在路由表中	121
5.2.2	问题: IGRP 没有安装所有可能的等价路径 ——原因: 缺省的最大路径把 IGRP 限制在四条路径内	143
5.3	IGRP 路由公布疑难解析	145
5.3.1	问题: 发送方没有公布 IGRP 路由	145
5.3.2	问题: 候选的缺省路由未被公布 ——原因: ip default-network 命令丢失	162
5.4	IGRP 路由重发布问题疑难解析	164
5.5	IGRP 中按需拨号路由问题疑难解析	166
5.5.1	问题: IGRP 广播使 ISDN 链路保持不断 ——原因: 关注流量的定义中没有拒绝 IGRP 广播	167
5.5.2	问题: IGRP 更新不通过拨号接口 ——原因: 在 dialer map 语句中丢失了 broadcast 关键字	168
5.6	IGRP 路由抖动问题疑难解析	170
5.7	变化量问题疑难解析	172
第 6 章	理解 EIGRP 协议	177
6.1	度量	178
6.2	EIGRP 的邻居关系	179
6.3	扩散更新算法	180
6.4	DUAL 有限状态机	182
6.5	EIGRP 的可靠传输协议	182
6.6	EIGRP 分组格式	183
6.7	EIGRP 行为	185
6.8	EIGRP 汇总	186
6.9	EIGRP 查询过程	187
6.10	缺省路由和 EIGRP	187
6.11	EIGRP 中的非等价负载平衡	188
6.12	小结	190
6.13	复习题	190
第 7 章	EIGRP 协议疑难解析	193

4.7	缺省路由与 IGRP	111
4.8	IGRP 中的非等价负载平衡	112
4.9	小结	114
4.10	复习题	114
第 5 章	IGRP 协议疑难解析	117
5.1	解决 IGRP 常见问题的流程图	118
5.2	IGRP 路由安装疑难解析	121
5.2.1	问题: IGRP 路由不在路由表中	121
5.2.2	问题: IGRP 没有安装所有可能的等价路径 ——原因: 缺省的最大路径把 IGRP 限制在四条路径内	143
5.3	IGRP 路由公布疑难解析	145
5.3.1	问题: 发送方没有公布 IGRP 路由	145
5.3.2	问题: 候选的缺省路由未被公布 ——原因: ip default-network 命令丢失	162
5.4	IGRP 路由重发布问题疑难解析	164
5.5	IGRP 中按需拨号路由问题疑难解析	166
5.5.1	问题: IGRP 广播使 ISDN 链路保持不断 ——原因: 关注流量的定义中没有拒绝 IGRP 广播	167
5.5.2	问题: IGRP 更新不通过拨号接口 ——原因: 在 dialer map 语句中丢失了 broadcast 关键字	168
5.6	IGRP 路由抖动问题疑难解析	170
5.7	变化量问题疑难解析	172
第 6 章	理解 EIGRP 协议	177
6.1	度量	178
6.2	EIGRP 的邻居关系	179
6.3	扩散更新算法	180
6.4	DUAL 有限状态机	182
6.5	EIGRP 的可靠传输协议	182
6.6	EIGRP 分组格式	183
6.7	EIGRP 行为	185
6.8	EIGRP 汇总	186
6.9	EIGRP 查询过程	187
6.10	缺省路由和 EIGRP	187
6.11	EIGRP 中的非等价负载平衡	188
6.12	小结	190
6.13	复习题	190
第 7 章	EIGRP 协议疑难解析	193

7.1 EIGRP 邻居关系疑难解析	193
7.1.1 查阅 EIGRP 日志获悉邻居改变	194
7.1.2 EIGRP 邻居问题——原因：单向链路	196
7.1.3 EIGRP 邻居问题——原因：非公共子网	197
7.1.4 EIGRP 邻居问题——原因：不匹配的掩码	199
7.1.5 EIGRP 邻居问题——原因：不匹配的 K 值	201
7.1.6 EIGRP 邻居问题——原因：不匹配的 AS 号	202
7.1.7 EIGRP 邻居问题——原因：被粘滞于活动状态	203
7.2 EIGRP 路由公布疑难解析	211
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	212
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	217
7.2.3 EIGRP 使用了不期望的度量公布路由	220
7.3 EIGRP 路由安装疑难解析	224
7.3.1 EIGRP 没有安装路由——原因：自动汇总或者手动汇总	224
7.3.2 EIGRP 没有安装路由——原因：更高的管理距离	227
7.3.3 EIGRP 没有安装路由——原因：重复的路由器 ID	229
7.4 EIGRP 路由抖动疑难解析	232
7.5 EIGRP 路由汇总疑难解析	235
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	236
7.5.2 EIGRP 汇总路由问题——原因：太多的汇总	238
7.6 EIGRP 重发布问题疑难解析	239
7.7 EIGRP 拨号备份问题疑难解析	245
7.8 EIGRP 错误消息	248
7.9 小结	249
第 8 章 理解 OSPF 协议	251
8.1 OSPF 分组细节	252
8.1.1 Hello 分组	253
8.1.2 数据库描述分组	254
8.1.3 链路状态请求分组	255
8.1.4 链路状态更新分组	255
8.1.5 链路状态确认分组	256
8.2 OSPF LSA 细节	256
8.2.1 路由器 LSA	257
8.2.2 网络 LSA	260
8.2.3 汇总 LSA	261
8.2.4 外部 LSA	264
8.3 OSPF 区域	267
8.3.1 普通区域	269
8.3.2 端区域	270

8.3.3 完全区域	270
8.3.4 非末梢区域	270
8.4 OSPF 邻居关系	270
8.4.1 多邻接	270
8.4.2 点-to-point	270
8.4.3 非广播多路访问	270
8.4.4 请	270
8.4.5 OS	270
8.5 OSPF 路由	270
8.5.1 OS	270
8.5.2 OS	270
8.5.3 OS	270
8.5.4 OS	270
8.5.5 OS	270
8.5.6 OS	270
8.5.7 OS	270
8.5.8 OS	270
8.6 小结	270
8.7 复习题	270
第 9 章 OSPF 配置	270
9.1 解决 O	270
9.1.1 OS	270
9.1.2 OS	270
9.1.3 OS	270
9.1.4 OS	270
9.1.5 OS	270
9.1.6 “C	270
9.1.7 OS	270
9.1.8 SP	270
9.2 OSPF 邻	270
9.2.1 问	270
9.2.2 问	270
9.2.3 问	270
9.2.4 问	270
9.2.5 问	270
9.2.6 问	270
9.3 OSPF 路	270
9.3.1 问	270

..... 193	8.3.3 完全端区域	271
..... 194	8.3.4 非完全端区域	271
..... 196	8.4 OSPF 介质类型	276
..... 197	8.4.1 多点接入介质	277
..... 199	8.4.2 点到点介质	277
..... 201	8.4.3 非广播多点接入介质	278
..... 202	8.4.4 请求线路	280
..... 203	8.4.5 OSPF 介质类型汇总	282
..... 211	8.5 OSPF 邻接体	282
样做 .. 212	8.5.1 OSPF 关闭状态	283
了 .. 217	8.5.2 OSPF 尝试状态	283
..... 220	8.5.3 OSPF 初始状态	284
..... 224	8.5.4 OSPF 2-way 状态	284
..... 224	8.5.5 OSPF Exstart 状态	284
..... 227	8.5.6 OSPF 交换状态	285
..... 229	8.5.7 OSPF 装入状态	285
..... 232	8.5.8 OSPF Full 状态	286
..... 235	8.6 小结	286
生 .. 236	8.7 复习题	287
..... 238	第 9 章 OSPF 协议疑难解析	289
..... 239	9.1 解决 OSPF 常见问题的流程图	290
..... 245	9.1.1 OSPF 邻居关系疑难解析	290
..... 248	9.1.2 OSPF 路由通告疑难解析	293
..... 249	9.1.3 OSPF 路由安装疑难解析	294
..... 251	9.1.4 OSPF 中路由重发布疑难解析	295
..... 252	9.1.5 OSPF 中路由汇总疑难解析	295
..... 253	9.1.6 “CPUHOG” 问题疑难解析	296
..... 254	9.1.7 OSPF 中按需拨号路由 (DDR) 问题疑难解析	296
..... 255	9.1.8 SPF 计算问题和路由抖动问题疑难解析	297
..... 255	9.2 OSPF 邻居关系疑难解析	297
..... 256	9.2.1 问题: OSPF 邻居表为空	298
..... 256	9.2.2 问题: OSPF 邻居停滞于 ATTEMPT 状态	323
..... 257	9.2.3 问题: OSPF 邻居停滞于 INIT 状态	326
..... 260	9.2.4 问题: OSPF 邻居停滞于 2-way 状态	
..... 261	——原因: 在所有路由器上都配置了优先级 0	336
..... 264	9.2.5 问题: OSPF 邻居停滞于 EXSTART / EXCHANGE 状态	337
..... 267	9.2.6 问题: OSPF 邻居停滞于 LOADING 状态	351
..... 269	9.3 OSPF 路由通告疑难解析	355
..... 270	9.3.1 问题: OSPF 邻居不通告路由	355

9.3.2 问题: OSPF 邻居 (ABR) 不通告汇总路由	363
9.3.3 问题: OSPF 邻居不通告外部路由	371
9.3.4 问题: OSPF 邻居不通告缺省路由	380
9.4 OSPF 路由安装疑难解析	391
9.4.1 问题: OSPF 不在路由表中安装任何路由	391
9.4.2 问题: OSPF 不在路由表中安装外部路由	406
9.5 OSPF 重发布问题疑难解析	413
9.6 OSPF 中路由汇总疑难解析	418
9.6.1 问题: 路由器不汇总区域间路由	
——原因: 没有在 ABR 上配置 area range 命令	419
9.6.2 问题: 路由器不汇总外部路由	
——原因: 没有在 ASBR 上配置 summary-address 命令	421
9.7 CPUHOG 问题疑难解析	423
9.7.1 问题: CPUHOG 消息出现在邻接体关系形成过程中	
——原因: 路由器没有运行分组调速代码	423
9.7.2 问题: CPUHOG 消息出现在 LSA 刷新阶段	
——原因: 路由器没有运行 LSA 组调速代码	424
9.8 OSPF 中按需拨号路由 (DDR) 问题疑难解析	426
9.8.1 问题: OSPF Hello 分组启动链路	
——原因: OSPF Hello 分组被允许作为关注流量	426
9.8.2 问题: 请求线路持续启动链路	427
9.9 SPF 计算和路由抖动疑难解析	437
9.9.1 SPF 经常运行——原因: 在网络中存在接口抖动	438
9.9.2 SPF 经常运行——原因: 在网络中存在邻居抖动	440
9.9.3 SPF 经常运行——原因: 路由器 ID 重复了	443
9.9.4 常见 OSPF 出错消息	447
9.9.5 “Unknown routing Protocol” 出错消息	447
9.9.6 “OSPF: Could not allocate routerid” 出错消息	447
9.9.7 “% OSPF-4-BADLSATYPE: Invalid Isa: Bad LSA type”	
类型 6 出错消息	448
9.9.8 “OSPF-4-ERRRCV” 出错消息	448
第 10 章 理解 IS-IS 协议	451
10.1 IS-IS 协议概述	452
10.2 IS-IS 协议的概念	453
10.2.1 IS-IS 节点、链路和区域	453
10.2.2 邻接体	455
10.2.3 分层路由	457
10.2.4 IS-IS 分组	458
10.2.5 IS-IS 度量	460

10.2.6	
10.2.7	
10.3 IS-IS	
10.3.1	
10.3.2	
10.3.3	
10.4 配置	
10.4.1	
10.4.2	
10.4.3	
10.4.4	
10.4.5	
10.5 小结	
10.6 IS-IS	
10.6.1	
10.6.2	
10.6.3	
10.6.4	
10.7 复习	
第 11 章 IS-IS	
11.1 IS-IS	
11.1.1	
11.1.2	
11.1.3	
11.2 IS-IS	
11.2.1	
11.2.2	
11.2.3	
11.3 IS-IS	
11.4 CLNS	
11.5 案例	
11.6 IS-IS	
11.7 小结	
第 12 章 理解 P	
12.1 IGMP	
12.1.1 IC	
12.1.2 IC	
12.1.3 组	

10.2.6 IS-IS 认证	462
10.2.7 ISO CLNP 编址	463
10.3 IS-IS 链路状态数据库	465
10.3.1 IS-IS 链路状态数据库概述	465
10.3.2 泛洪和数据库同步	468
10.3.3 最短路径优先 (SPF) 算法和 IS-IS 路由计算	469
10.4 配置 IS-IS 实现 IP 路由	470
10.4.1 点到点串行链路上的 IS-IS 配置	470
10.4.2 ATM 配置举例	477
10.4.3 IP 缺省路由公布	480
10.4.4 路由重发布	481
10.4.5 IP 路由汇总	483
10.5 小结	484
10.6 IS-IS 分组的附加信息	485
10.6.1 IS-IS 分组字段	485
10.6.2 Hello 分组	486
10.6.3 链路状态分组	488
10.6.4 序列号分组	489
10.7 复习题	490
第 11 章 IS-IS 协议疑难解析	493
11.1 IS-IS 邻接体问题疑难解析	495
11.1.1 问题 1: 部分或者所有邻接体没有启动	497
11.1.2 问题 2: INIT 状态的邻接体	502
11.1.3 问题 3: 没有形成 IS-IS 邻接体, 只形成了 ES-IS 邻接体	510
11.2 IS-IS 路由更新问题疑难解析	510
11.2.1 路由公布问题	512
11.2.2 路由重发布和二级到一级的路由泄漏问题	515
11.2.3 路由抖动问题	516
11.3 IS-IS 错误	519
11.4 CLNS ping 和 traceroute	520
11.5 案例研究: ISDN 配置问题	522
11.6 IS-IS 疑难解析命令总结	525
11.7 小结	525
第 12 章 理解 PIM 协议	529
12.1 IGMP 版本 1、IGMP 版本 2 和逆向路径转发的基本原理	530
12.1.1 IGMP 版本 1	530
12.1.2 IGMP 版本 2	531
12.1.3 组播转发 (RPF)	532

12.2 PIM 密集模式	533	15.3.3
12.3 PIM 稀疏模式	535	15.3.4
12.4 IGMP 分组格式和 PIM 分组格式	537	
12.4.1 IGMP 分组格式	537	15.4 BGP
12.4.2 PIM 分组/消息格式	538	15.4.1
12.5 小结	541	15.4.2
12.6 复习题	541	
第 13 章 PIM 协议疑难解析	543	15.4.3
13.1 IGMP 加入问题疑难解析	543	15.4.4
13.2 PIM 密集模式疑难解析	546	
13.3 PIM 稀疏模式疑难解析	551	15.5 BGP
13.4 小结	555	15.5.1
第 14 章 理解 BGP-4 协议	559	15.5.2
14.1 BGP-4 协议规范和功能	562	15.6 BGP
14.2 邻居关系	563	15.6.1
14.2.1 EBGP 邻居关系	564	15.6.2
14.2.2 IBGP 邻居关系	566	15.6.3
14.3 公布路由	566	15.6.4
14.4 接收路由	570	
14.5 策略控制	570	15.7 BGP
14.5.1 使用 BGP 属性进行策略控制	572	15.7.1
14.5.2 路由映射图在策略控制中的使用	585	
14.5.3 使用过滤表、分发列表、前缀列表、团体 以及出站路由过滤 (ORF) 进行策略控制	588	15.7.2
14.5.4 路由衰减	595	15.7.3
14.6 缩放大规模网络中的 IBGP —— 路由反射器和联盟	599	15.7.4
14.6.1 路由反射	600	
14.6.2 AS 联盟	603	15.8 小型
14.7 最优路径计算	605	15.8.1
14.8 小结	607	
14.9 复习题	607	15.8.2
第 15 章 BGP 协议疑难解析	611	
15.1 解决 BGP 常见问题的流程图	612	
15.2 用于解决 BGP 相关问题的 show 命令和 debug 命令	616	
15.3 BGP 邻居关系疑难解析	618	15.9 BGP
15.3.1 问题: 没有初始化的直接连接的 EBGP 邻居	618	15.9.1
15.3.2 问题: 非直接连接的 EBGP 邻居关系不能建立	622	

15.3.3 问题: IBGP 邻居关系不能建立	629
15.3.4 问题: BGP 邻居 (外部的和内部的) 不能建立	
——原因: 接口访问列表阻塞了 BGP 分组	629
15.4 BGP 路由公布/产生以及接收疑难解析	631
15.4.1 问题: BGP 路由没有被产生	631
15.4.2 向 IBGP/EBGP 邻居传播/产生 BGP 路由时的问题	
——原因: 错误配置了过滤器	637
15.4.3 只向 IBGP 邻居而不向 EBGP 邻居传播 BGP 路由时的问题	
——原因: BGP 路由来自另一个 IBGP 发言人	639
15.4.4 向 IBGP/EBGP 邻居传播 IBGP 路由时的问题	
——原因: IBGP 路由没有同步	645
15.5 BGP 路由未被装入路由表疑难解析	646
15.5.1 问题: IBGP 学习的路由没有被装入 IP 路由表	646
15.5.2 问题: EBGP 学习的路由未被装入 IP 路由表	653
15.6 BGP 路由反射问题疑难解析	660
15.6.1 问题: 配置错误——原因: 把 IBGP 邻居配置为路由反射器客户失败 ..	660
15.6.2 问题: 路由反射器客户存储了一个多余的 BGP 更新	
——原因: 客户到客户的反射	662
15.6.3 问题: 路由反射器和客户的收敛时间变长——原因: 使用了对等体组 ..	664
15.6.4 问题: 路由反射器和路由反射器客户之间的冗余丢失	
——原因: RR 中的簇列表检查丢掉了来自其它 RR 的冗余路由	665
15.7 BGP 策略引起的出口 IP 流量问题疑难解析	670
15.7.1 问题: 存在多个出口点, 但是流量只使用一个或者几个出口路由器	
——原因: BGP 策略定义导致流量只从一个地方出去	670
15.7.2 问题: 流量通过的接口与路由表显示的不同	
——原因: 路由的下一跳可以通过另一个路径到达	674
15.7.3 问题: 存在通往同一 BGP 邻居 AS 的多重 BGP 连接, 而流量却只使用	
一个连接流出——原因: BGP 邻居通过发送 MED 或者预先附加	
AS_PATH 来影响出口流量	677
15.7.4 问题: 出现不对称路由并导致一些问题, 尤其在使用 NAT 和对时间	
敏感的应用时——原因: 发出公布并接收公布	680
15.8 小型 BGP 网络中负载均衡问题疑难解析	683
15.8.1 问题: 当单个路由器对于同一 ISP 是双宿主时的负载均衡和出口流量管理	
——原因: BGP 在路由表中只安装了一个最优路径	683
15.8.2 问题: IBGP 网络中的负载均衡和出口流量管理——原因: 缺省情况下,	
即使存在多条等价的 BGP 路径, Cisco IOS 的 IBGP 也只允许安装	
一条路径到路由表中	685
15.9 BGP 策略引起的入口 IP 流量问题疑难解析	688
15.9.1 问题: 虽然存在通往某个 AS 的多个连接, 但是, 同一个 AS 中的所有流量	
都通过一个 BGP 邻居 X 进入——原因: BGP 邻居 X 有一个 BGP 策略配置,	

使它自己优先于其它的对等体而被选择，或者网络被公布为只吸引 来自 X 的流量	689
15.9.2 问题：AS 110 中的 BGP 邻居本来有多个连接，但是，一些来自 Internet 的流 量总是通过 AS 110 进入 100.100.100.0/24——原因：AS 109 中 100.100.100.0/24 的路由公布吸引了通过 AS 110 的 BGP 邻居的 Internet 流量	694
15.10 BGP 最优路径计算问题疑难解析	695
15.10.1 问题：具有最低 RID 的路径没有被选为最佳路径	695
15.10.2 问题：具有最低 MED 的路径没有被选择为最优路径	698
15.11 BGP 过滤疑难解析	701
15.11.1 问题：标准访问列表捕获子网失败	701
15.11.2 问题：扩展访问列表捕获正确的掩码路由失败	703
15.11.3 问题：用正规表达式进行 AS_PATH 过滤	707
15.12 总结	707
附录 复习题答案	711

第 1 章

理解 IP 路由

本章介绍 IP 路由，并对 IP 编址和 IP 路由协议的各种分类方法等相关概念进行深入剖析。此外，还简单介绍了与路由过滤和重发布等操作的实现和配置相关的技术。

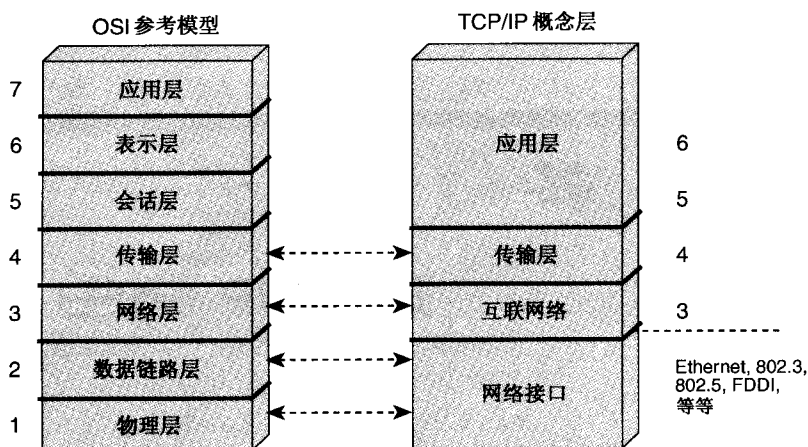
TCP/IP 协议栈是在 Internet 上交换信息的基本技术。TCP/IP 使用类似于开放系统互联（Open System Intenconnection, OSI）参考模型的分层方法来实现计算机通信，但层数小于 7。图 1-1 同时显示了 OSI 参考模型和 TCP/IP 协议栈。图中标出了两个协议栈的相关层。

IP 工作在 TCP/IP 协议栈的互联网络层。该层与 OSI 参考模型的网络层相对应。IP 层提供了无连接数据传输服务，即将信息分割成数据单元以分组（packet）或数据报（datagram）的形式从网络的一个地方传送到另一个地方。数据报传送服务模型的实质是在网络中的两个点之间不需要为传送数据预先建立永久的端到端路径。在基于分组的网络中，位于传送路径上的每一个路由器根据到达被发送分组的目的地址的最佳转发路径独立地进行本地决策。进行决策所需要的转发能力既可以通过路由协议动态获得，也可以通过手工配置的静态路由实现。

编址是进行数据转发处理的重要方面。任何定向的通信都具有源地址和目的地址。编址既可以使信源指定目的地址，也可以使目的节点识别信源。编址在数据报传送模式的操作中非常重要，原因是在 IP 转发过程中，从源结点到目的结点之间的任何数据传输路径所经过的中间结点并不固定。

如前所述，在 IP 数据报服务的基本设施中，从一个设备传送到另一个设备的信息首先被分割成分组。每个分组包括 IP 头、传输层（TCP 或 UDP）报文头和负载。负载是原始信息的一部分。每个 IP 分组是自包含的，并且通过一系列处于

传送路径上的中间设备链独立地传送到目的地。



网络中的路由器依靠路由协议或静态配置以流的形式将数据报转发至目的地。对目的地来说，数据路径上的每个结点只关心流出接口或者本地决定的（或者特定转发策略规定的）通往目的地址的最优路径链路。IP 转发过程经常被描述为一种基于目的地址的逐跳（hop-by-hop）转发机制。这意味着数据链路上每一跳的路由器通常基于目的地址转发分组。但是，现代路由器还可以使用基于策略的标准，例如利用分组的源地址控制转发。

在目的结点，属于同一个流的分组被合成为原来的信息。IP 编址将在下面“IP 编址的概念”一节中进行讨论。

基于第三层地址（即 IP 地址）从一个结点向无连接网络的另一个结点转发分组的过程也称为路由。路由器就是具有路由智能的网络设备。

那么路由器如何真正决定 Internet 上向何处以及怎样转发分组呢？这可以通过两种方式实现。前面已经提到路由器可以手工配置事先定好的路径信息，称为静态路由。也可以运行具有学习和共享路由信息能力的应用程序，通过这种形式获得和传播路由信息的方法称为动态路由。

1.1 IP 编址的概念

IP 编址是 IP 协议操作的核心。图 1-1 所示的 TCP/IP 协议栈表示了一个通向下面的物理层和数据链路层的网络接口。这样，就可以实现 IP 协议的介质无关性。介质无关性是 IP 协议能够被广泛接受和普遍存在的重要原因之一。IP 使用自己的编址模式，不依赖于下层的局域网（LAN）或广域网（WAN）中介质互联的设备，从而与它的介质无关体系结构相一致。因此，IP 能够成功地在由几种不同介质技术组成的网络架构上进行操作。这种适应能力与简单的协议栈结构共同促成 IP 协议的广泛使用。

IP 编址将地址赋给设备的每个独立的网络接口（使用基于链路的方法），而不是为整个设备赋一个地址（基于主机的方法）。设备的不同接口与被称为子网络（或子网）的网络链路相连，并且被赋予子网地址。接口的 IP 地址从相连链路的子网地址空间中分配。这种基于链

路的编址方式，其中的 IP 子网汇总路 IP 网络中地址解析解析为相应的数据

目前，有两类表示 IP 地址。准确地址。随着全球 Internet 不足，于是 128bit 协议出现的问题的疑 IPv6 中去。下列 IP

- IPv4 地址
- 私有 IPv4
- IPv4 子网
- 无类域间

1.1.1 IP

如前所述，32 地址模式需要网络连地址组也可称为地

最初，IP 网络的思想，是希望通分配。连接主机少分类的另一个好处

A、B、C、D 类包括一组 IPv4 地

表 1-1

地址类	首子
A	0xxxx
B	10xxx
C	110xx
D	1110x
E	11110

如表 1-1 所示定的地址类。

在这 5 种地址 D 类地址是为 IP 组

为了使每个单号（网络 ID）和主

路的编址方式,其优点是不必知道网络中所有主机的情况,路由器就可以通过跟踪 IP 路由表中的 IP 子网汇总路由信息。这对于同时将许多设备互联的以太网广播链路是非常有用的。在 IP 网络中地址解析协议(Address Resolution Protocol,ARP)被用来把直接相连主机的 IP 地址解析为相应的数据链路层地址。

目前,有两类 IP 地址:IPv4 地址和 IPv6 地址。在使用 IPv6 之前,IPv4 编址使用 32bit 表示 IP 地址。准确地说,这种 32bit 的编址模式可以提供 2^{32} (4294967295) 个不同的主机地址。随着全球 Internet 规模的扩大,这种 32bit 的 IPv4 编址模式已经在可预见的将来表现出不足,于是 128bit 的 IPv6 编址模式便应运而生。本书内容涵盖了 IPv4 环境下使用 IP 路由协议出现的问题的疑难解析。接下来讨论 IPv4 编址结构和相关概念,大多数内容仍可应用到 IPv6 中去。下列 IPv4 编址的标题将在后面的小节中讨论:

- IPv4 地址分类。
- 私有 IPv4 地址空间。
- IPv4 子网化和可变长子网掩码。
- 无类域间路由。

1.1.1 IPv4 地址分类

如前所述,32bit IPv4 编址模式可以定义大量的主机地址。但是,IP 采用的基于链路的编址模式需要网络连接与成组的地址相关联,从而使相连的主机可以被赋予特定的地址。这些地址组也可称为地址前缀,经典 IP 术语称之为 IP 网络号(IP network number)。

最初,IP 网络号定义具有严格的界限,并被分组成不同的地址类。对 IP 地址进行分类的思想,是希望通过创建支持不同数目主机的 IP 地址类,以便能更高效的分配 IP 地址空间分配。连接主机少的网络链路所分配的地址类别,应该恰好能够支持相应数目的主机。地址分类的另一个好处是可以使地址分配过程更加流畅,从而增强其可管理性。

A、B、C、D 和 E 5 类地址由 IP 地址首字节的最高几位设置进行定义和区分。每个地址类包括一组 IPv4 地址子网,每个子网支持一定数目的主机。表 1-1 列出了 5 个 IPv4 地址类。

表 1-1 IP 地址分类和表示

地址类	首字节的位模式	十进制首字节表示范围	点分十进制表示主机赋值范围
A	0xxxxxxx	1 到 127	1.0.0.1 到 126.255.255.254
B	10xxxxxx	128 到 191	128.0.0.1 到 191.255.255.254
C	110xxxxx	192 到 223	192.0.0.1 到 223.255.255.254
D	1110xxxx	224 到 239	224.0.0.1 到 239.255.255.254
E	11110xxx	240 到 255	240.0.0.1 到 255.255.255.255

如表 1-1 所示,IP 地址的首字节的特定位模式与一系列地址相对应,并被映射为一个特定的地址类。

在这 5 种地址类中,A、B 和 C 这 3 类是单播地址,用于单个信源到单个目的的通信。D 类地址是为 IP 组播应用保留的,即一对多的通信。E 类地址是为实验目的而保留的。

为了使每个单播地址类(A、B 和 C)支持最大数目的主机,32bit 地址字段被分成网络号(网络 ID)和主机号(主机 ID),如下所示:

- A 类：8bit 网络 ID，24bit 主机 ID。
- B 类：16bit 网络 ID，16bit 主机 ID。
- A 类：24bit 网络 ID，8bit 主机 ID。

图 1-2 显示了 A 类地址中 32bit 的分配情况。最高位固定为 0，首字节表示网络 ID，后边的 3 个字节为主机 ID。

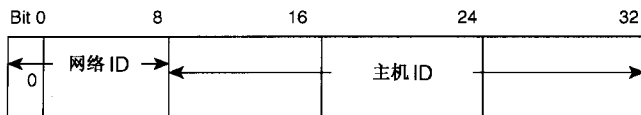


图 1-2 A 类地址 bit 的分配

以严格的界限为 IP 地址分类的思想也称为有类编址。IP 地址使用掩码把主机号和网络号分开。经过很多次改进，IP 地址才演化成现在的结构，地址编码和实际网络中的地址分配也比过去更加高效。在“子网化和可变长子网掩码”小节将提供更多这方面的知识。

使用一种称为点分十进制的形式表示 IP 地址，以使其更易于操作。点分十进制表示法将地址位分组为八位组并用点分开，然后每个二进制的八位组转换成等值的十进制数。表 1-1 的最后一列显示了每个地址类中的一系列地址的点分十进制表示法。

虽然引入有类地址可以使 IPv4 地址空间被有效地使用，但是严格的分类界限却限制了使用的高效性。由于有类编址的严格性和低效性，它逐渐被更加高效和灵活的无类编址所取代。

在无类编址中，任何 IP 网络号都被解释为某一长度的前缀。这种解释方法更加具有灵活性并且可以高效地利用 IPv4 地址空间。像 A 类地址这样的有类别一大块地址的集合为了能够分配给多个组织而可以分成很多的小块，而不必像以前规定的那样固定分配给一个组织。反之，无类编址也可以把多个 C 类地址合并成一个大块地址，而不必分割开独立处理。以这种方式聚合地址是为了节省与 Internet 相连的路由器资源，这种方式称为无类域间路由 (CIDR)。“无类域间路由”小节将进行深入讨论。

1.1.2 IPv4 私有地址空间

单播空间的有些地址块被单独保留下来，并指定为私有地址。私有地址空间是为没有与公众 Internet 相连的网络准备的。在 RFC1918 中，规定下列地址为 IPv4 私有地址空间部分：

- 10.0.0.0 到 10.255.255.255。
- 172.16.0.0 到 172.31.255.255。
- 192.168.0.0 到 192.168.255.255。

RFC 1700 对已保留和已分配参数提供了通用信息，包括已保留的地址等内容。部署 IPv4 私有地址空间的私有网络仍然可以使用网络地址转换 (Network Address Translation, NAT) 与公众 Internet 保持互连。

1.1.3 子网化和可变长子网掩码

在采用 CIDR 之前，每个有类网络号只可以分配给一个组织使用。但是，在这个组织内

可以使用子网化技术

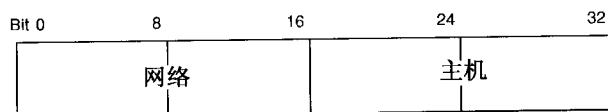
IP 子网化技术把入新一级层次。这种两个八位组中的一个 1-3 所示。

如果 IP 地址已 1-4 显示了新的掩码和机位。通常，通过说如，172.16.1.0 255.25

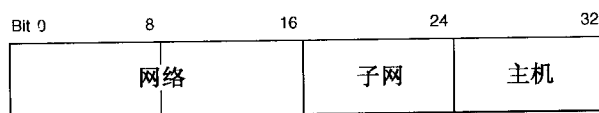
虽然有类编址为码才能够用在有类网

可以使用子网化技术将该地址分成多个小的地址组以供同一网络域的不同部门使用。

IP 子网化技术把一些主机位移动到网络 ID 字段作为网络号，从而 IP 地址分类结构中引入新一级层次。这种扩展的网络 ID 称为子网络号或 IP 子网。例如，B 类地址表示主机号的两个八位组中的一个可以用来创建 255 个子网，每个子网只有一个八位组标识主机位，如图 1-3 所示。



B 类地址格式

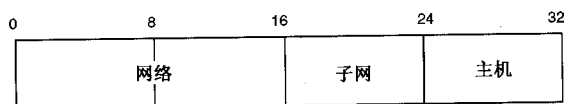


已子网化的 B 类地址格式

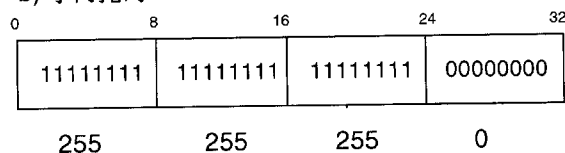
图 1-3 B 类子网举例

如果 IP 地址已经子网化，那么地址掩码已经调整为能够反映新的网络号和主机号。图 1-4 显示了新的掩码和由 B 类地址创建的相应子网。掩码中一系列的 1 表示网络位，0 表示主机位。通常，通过说明前缀长度表示 IP 地址，也就是说用掩码中 1 的个数表示 IP 地址。例如，172.16.1.0 255.255.255.0 表示 172.16.1.0/24。

a) 子网化 B 类地址



b) 子网掩码



c) B 类子网

B 类地址	8 位子网地址
172.16.0.0/16	172.16.0.0/24 172.16.1.0/24 172.16.2.0/24 ⋮ 172.16.255.0/24

图 1-4 子网掩码举例

虽然有类编址为了将一块地址进行更有效的分配而允许进行子网化，但是只有一致的掩码才能够用在有类网络环境中。VLSM 扩展了子网化的思想，它允许一个网络号使用不同的

掩码,这样就可以灵活地根据网络域不同网段的应用将地址分成不同大小的块,从而更有效地使用分配的地址块。例如,通过使用 VLSMB 类地址 172.16.0.0/16,将主机位的 8bit 作为子网位,可以分成带有 24bit 子网掩码的更小的子网。然后通过剩余主机位的 4 位将新生成的子网,如 172.16.1.0/24,进一步子网化。这将导致生成更小的块,如 172.16.1.0/28, 172.16.1.16/28 和 172.16.1.32/28 等等。VLSM 只能用在无类网络环境中,即该环境中的路由协议和相关路由软件支持无类编址。图 1-5 说明了如何使用 VLSM 进行子网化。

B 类地址	8 位子网地址	4 位 VLSM
172.16.0.0/16	172.16.0.0/24 172.16.1.0/24 172.16.2.0/24 ⋮ 172.16.255.0/24	172.16.2.0/28 172.16.2.16/28 172.16.2.32/28 ⋮ 172.16.2.240/28

图 1-5 VLSM 示例

1.1.4 无类域间路由

VLSM 可以提高已分配地址块的 IP 地址使用效率,但它却不能解决为各个组织分配地址的低效问题。有类地址块的低效使用将导致 IP 地址耗尽,全球 Internet 路由表中有类地址数目不断增长,很多组织被分配多个 C 类地址而不是一个单独的 B 类地址,所有这些促成了无类域间路由(classless interdomain routing, CIDR)的产生。

CIDR 允许任何长度的 IP 网络号,完全放弃了与有类概念相关的固定界限。图 1-6 举例说明了 CIDR 的两个优点。通过排除地址分类的思想,由单独的 C 类地址组成的诸如 192.168.0.0 到 192.168.255.0 的地址块可以方便地用 192.168.0.0/16 表示成一个块。这本质上意味着 256 个过去的 C 类地址可聚合成一个地址块,这称为 CIDR 块或超网。

有类地址	CIDR 超网
192.168.0.0/24 192.168.1.0/24 192.168.2.0/24 192.168.3.0/24 ⋮ 192.168.255.0/24	192.168.0.0/16

有类地址	CIDR 子网
131.108.0.0/16	131.108.0.0/18 131.108.64.0/18 131.108.128.0/18 131.108.192.0/18

图 1-6 CIDR 聚合和子网化

CIDR 使得网络地址分配更灵活。例如, 131.108.0.0/16 和 131.108.192.0/16 可以聚合为 131.108.0.0/14。

路由器可以手动配置,也可以自动配置。路由器使用某个专用协议来自动地收集路由信息,所以这些路由程序称为动态路由协议。

由于静态路由配置简单,与动态路由相比,动态路由的配置和管理的代价就是配置和管理的内存和处理器资源,以及解决复杂问题所需的时间。

虽然静态路由配置简单,但在规模可观的网络中,动态路由是现代的大型企业网络的首选。动态路由的另一个缺点是 OSPF 和 IS-IS 等动态路由协议。

本小节讨论 IP 路由协议。讨论各种 IP 路由协议时,将讨论各种 IP 路由协议决策的共同目标,以及广播协议和组播协议。

前面简要地讨论了 IP 地址就是分组的地址,这是 IP 的一个重要概念。从原理上说,IP 子网是 IP 地址空间的一个子集。

前面的“IPv4 地址空间”一节讨论了组播地址空间的地。所有 IP 单播子网同时寻址到子网上。单播路由协议

CIDR 使得网络号同样可以灵活地被子网化,从而分配给不同的组织进行域间路由交换。例如, 131.108.0.0/16 可以分成 4 个子块 (131.108.0.0/18, 131.108.64.0/18, 131.108.128.0/18 和 131.108.192.0/18), 并分配给 4 个不同的组织。

1.2 静态和动态路由

路由器可以手工配置静态路径信息,在转发具有匹配目的地址的分组时能够简单地强制路由器使用某个专用接口或下一跳 IP 地址进行转发。静态路由能够匹配各种网络地址。然而,利用路由器上自动收集和共享路由信息的分布式程序也可以获取路由信息。这些程序不仅能自动地收集路由信息,还可以实时跟踪网络连接状态从而尽可能及时、有效地提供路由信息,所以这些路由程序通常被称为动态路由协议。

由于静态路由是手工配置的,当路径改变时就需要手工干预重新更改路由器的配置,与之相比,动态路由协议在处理路由信息时的操作显然更加简便。但是,作为获得这种简便性的代价就是配置和故障诊断的复杂性。动态路由协议的操作同样是资源密集型的,需要大量的内存和处理器资源。因此,使用动态路由协议经常需要具备网络设计、路由器配置、调整 and 解决复杂问题所必需的先进知识和尖端的专业知识。

虽然静态路由不需太多系统资源,对配置和解决问题所需的技术技巧的要求也不高,但是在规模可观的网络中手工加入路由所需的大量精力使之不太可行。显然,对于基于 IP 网络的现代的大型企业和因特网服务提供商 (ISP) 来说,静态路由不是一个很好的选择。静态路由的另一个缺点是,在实现复杂的路由策略时,缺乏灵活性。就实现路由策略而言, BGP、OSPF 和 IS-IS 等动态路由协议所提供的智能和灵活性是其他方法无法比拟的。下一节将讨论动态路由协议。

1.3 动态路由

本小节讨论 IP 路由的本质,阐明大型网络用户使用动态自动路由的必要性。本小节还将讨论各种 IP 路由协议的特征和分类。虽然所有的路由协议都具有收集路由信息支持分组转发决策的共同目标,但是可以基于它们提供转发信息的数据流类型将路由协议分成两大类:单播协议和组播协议。

前面简要地说明了 IP 为标识网络的不同位置或子网而使用的编址模式。IP 分组中的目的地址就是分组的目标地址。发送者的地址存储在源地址字段中。IP 子网是了解 IP 编址的一个重要概念。IP 子网,简称为子网,曾在当初“IP 编址的概念”小节中提到过。从物理上说,IP 子网是接口 IP 地址共享同一网络号并使用相同子网掩码的一组互联的网络设备。

前面的“IPv4 编址分类”小节讨论了单播和组播地址。单播地址空间用于编址网络设备,组播地址空间的地址则是用来定义从同一个组播程序接收信息的一个组和一组用户。

所有 IP 单播子网的最后一个地址被称为广播地址,如 192.168.1.255/24。这个地址可以同时寻址到子网上的所有结点,也可以称之为定向广播。

单播路由协议可以用来处理单播网络信息,并为 IP 分组转发到某单播目的地址提供路由

智能。组播转发在概念上不同于单播转发，它需要特殊的路由程序支持组播分组的转发。

1.3.1 单播 IP 路由与组播 IP 路由

IP 网络中的两个设备通常向各自的 IP 地址发送单播消息进行通信。IP 结点可以有多个活跃接口，每个接口都必须配置一个单播空间的 IP 地址。接口地址在该接口直接相连的子网上唯一地定义了该设备。

Cisco 路由器也支持辅助逻辑子网的概念。路由器的接口不仅配置一个主地址，还可以配置多个辅助逻辑子网。另外，可以使用 Cisco 路由器的隧道和回环接口实现单播 IP 的可达性。目的地址字段中单播地址的分组基于 IP 路由表中的信息进行转发。可以用 show ip route 命令查看 Cisco 路由器的 IP 路由表。

如果分组目的字段中地址是一个组播地址空间（D 类），那么分组将被直接送到可能有多接收者的组播组。组播转发使用专门的机制以提高网络资源的使用效率。如果为向多目标传送分组设计了一个程序，它使用单播路由转发程序数据流的分组，那么这将需要进行不必要的重复，导致网络资源的浪费。可以使用组播传播（multicast propagation）避免这种情况，组播传播就是只在前往接收者的必要网络分支上复制组播分组。

图 1-7 解释了利用单播将分组从 SRC1 转发到两个分离的目的地 RCV1 和 RCV2 的情况。

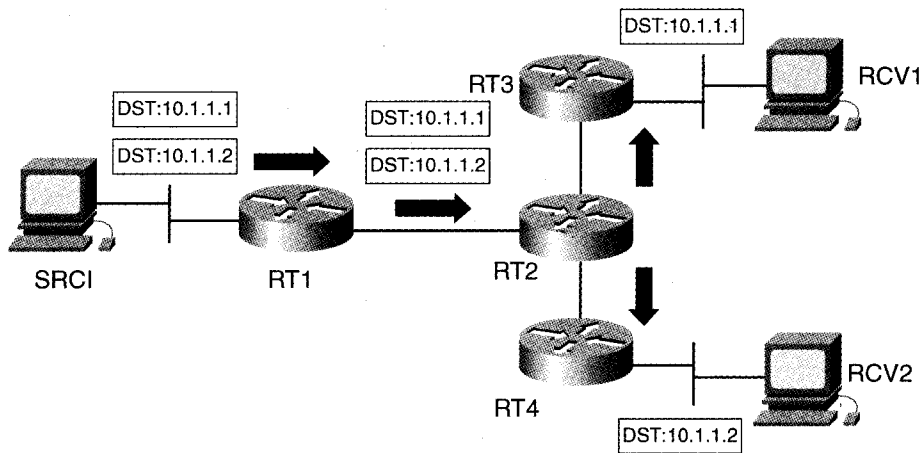


图 1-7 具有多个目的地址的单播转发

这种情况下，SRC1 产生两个相同的分组流，其目的地址分别是 10.1.1.1 和 10.1.1.2。属于每个流的分组分别进行处理，并通过 RT1 和 RT2 到达各自的目的，同时，在它们经过的路径上耗费网络资源（带宽和处理时间）。图 1-8 中采用了组播转发机制。

组播转发采用了只在通向不同接收者的网络路径分支处复制分组这种更有效的方法来传输信息。如图 1-8 所示，SRC1 只生成一个流，该流中的分组通过 RT1 和 RT2 转发。它们在 RT2 上被复制，然后被分别送到 RCV1 和 RCV2。

组播路由协议与单播路由协议的功能不同，它们在能够支持组播的路由器上需要使用反向路径转发（Reverse Path Forwarding, RPF）概念建立组播转发状态。RPF 可以用来确保从指向组播源的已知位置的接口收到组播分组，该接口通过适当的路由表规定。

第 12 章 “
表 1-2 列

表 1-2

单播
RIP(V1/V2)
IGRP
EIGRP
OSPF
IS-IS
BGP

Cisco IOS
Cisco IOS 软件
密集模式 (Spa
Protocol, MSDP
组播路由
Protocol, IGMP
离向量组播路由

本书出版
频流、远程培训
上将会变得非常

1.3.2

无类与有类
根据有类
块以外。但是，

分组的转发。

结点可以有很多
直接相连的子网

主地址，还可以
现单播 IP 的可达
以用 show ip route

接送到可能有多
如果为向多目标
将需要进行不必
避免这种情况，

和 RCV2 的情况。

RCV1

RCV2

和 10.1.1.2。属
在它们经过的路

更有效的方法来
RT2 转发。它们

器上需要使用反
以用来确保从指

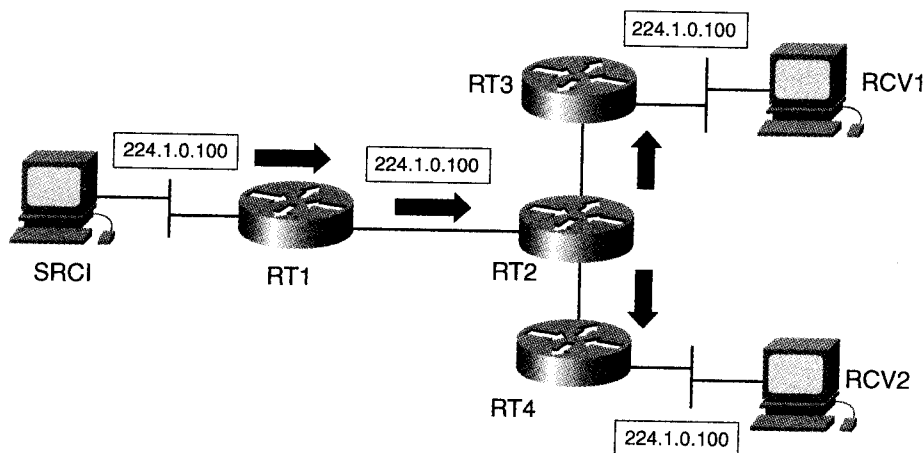


图 1-8 组播转发

第 12 章“理解 PIM 协议”将深入讨论 RPF，包括 IP 组播路由。
表 1-2 列出了常用的组播和单播路由协议。

表 1-2

单播路由协议和组播路由协议

单播	组播
RIP(V1/V2)	DVMRP
IGRP	PIM
EIGRP	MOSPF
OSPF	MOSPF
IS-IS	MSDP
BGP	

Cisco IOS 软件支持所有已列出的单播路由协议。但是，在所有已列出的组播路由协议中，Cisco IOS 软件只支持与协议无关的组播（Protocol Independent Multicast,PIM）的稀疏模式和密集模式（Sparse Mode/Dense Mode,SM/DM）、组播源发现协议（Multicast Source Discovery Protocol,MSDP）以及多协议 BGP。

组播路由环境还需要一个附加协议：互联网网关组播协议（Internet Gateway Multicast Protocol,IGMP）。组播 OSPF（Multicast OSPF,MOSPF）并没有获得支持，但是 IOS 具有与距离向量组播路由协议（Distance Vector Multicast Routing Protocol,DVMRP）互操作的能力。

本书出版时，Internet 还没有广泛地使用组播路由协议。但是，随着诸如无线电广播、视频流、远程培训、视频会议和游戏等大量面向组播的应用的出现，不久的将来，组播在 Internet 上将会变得非常普遍。

1.3.2 无类 IP 路由协议与有类 IP 路由协议

无类与有类 IP 路由协议的概念来自于 IP 地址最初的定义方式。

根据有类编址规则，网络号需要保持它自身的掩码，除了已经说明将其子网化成更小的块以外。但是，早期的诸如 RIP(Routing Information Protocol)这样的路由协议只能处理同一网

络域内地址只有一个掩码的情况,这个掩码可以是该网络自身的掩码,也可以是某个一致的子网掩码。诸如 RIP 的路由协议不能处理多于一种掩码的情况,例如 VLSM。这类协议称为有类协议(classful Protocol)(见表 1-3)。有类协议不能支持 VLSM 的原因是它们被设计成不把与路由相关的子网掩码与路由信息一起发布或传送,而是依靠简单直观的机制确定所学路由相关联的掩码。

全球范围内 Internet 的大规模增长,导致需要更有效地使用很有限的 IPv4 地址空间。而 IP 地址空间的可用地址越来越少。前面讲到的 VLSM 和 CIDR 这些无类概念的提出就是为了更有效地分配和使用地址。路由协议的功能也不断增强,以支持无类编址环境。为适应无类环境操作、处理 VLSM 地址以及 CIDR 而设计的路由协议称为无类路由协议(Classless Routing Protocol)。

表 1-3 根据有类和无类进行分类的路由协议给出一个列表。RIP-1 和 IGRP 属于有类协议,较新的 RIP-2、EIGRP、OSPF、IS-IS 和 BGP 属于无类协议。外部网关协议(Exterior Gateway Protocol,EGP)——边界网关协议(Border Gateway Protocol,BGP)的前身——是过时的协议,也是有类协议。

表 1-3

有类和无类 IP 路由协议

有类	无类
RIP-1	RIP-2
IGRP	EIGRP
EGP	OSPF
	Integrated IS-IS
	BGP

1.3.3 内部网关协议和外部网关协议

虽然很多单播路由协议是在 ARPANET (Internet 的前身) 早期开发的,但是路由信息协议(Routing Information Protocol,RIP)仍在广泛使用。很多政府研究机构和大学创建的独立网络都是 ARPANET 成功的标志,它们使用 RIP 进行动态路由操作。从 ARPANET 到 Internet 的演变需要使用某种更健壮的路由协议将大量孤立的网络互联起来。基于这种目的就选择了外部网关协议(EGP)。在不同 RIP 域之间 EGP 提供了有效的路由机制。因此, RIP 和 EGP 根据自己的能力在网络中担当的不同角色而被优化。RIP 用于域内路由,EGP 用于域间路由。后来,EGP 演变成为边界网关协议(BGP),RIP 也逐渐被其他一些更健壮的域内路由协议所代替。尤其是由 IETF 开发的开放式最短路径优先(Open Shortest Path First,OSPF)协议,它具有 RIP 所缺乏的路由代价智能化、更快收敛、在无类环境下进行操作等能力。现在可以对路由协议进行另一种分类:内部网关路由协议(在域内路由)和外部网关路由协议(在域间路由)。

图 1-9 给出了两个路由域: AS 65001 和 AS 65002,重叠区域(阴影部分)描述了每个域的边界路由器之间的互联。路由域用更规范的术语可以称为自治系统(autonomous system,As)。自治系统是处于统一管理下的一个独立的路由域。

如前所述,外部网关协议具有在两个路由域之间共享路由信息的能力。BGP 是在全球

Internet 上使用的。内部网关协议在自
合的 IGP。除 EGP、RIP、OSPF 和 IS-

表 1-4

距离向量
RIP-1
RIP-2
IGRP

为了提供比 IGP 更好的路由信息(IGRP)。IGRP

- 带宽。
- 延迟。
- 可靠性。
- 负载。
- 最大传输

Cisco 还将 IGRP 改名为 EIGRP,得到了比 IGRP 更好的路由信息。

OSPF 和 IS-IS 是链路状态路由协议,它们使用 Dijkstra 的命名)。

OSPF 和 IS-IS 是链路状态路由协议,它们使用 Dijkstra 的命名)。
两种协议都支持 VLSM 和 CIDR。
它们只是在结构上有所不同。
有趣的是:OSPF 和 IS-IS 都是无类路由协议。

Internet 上使用的对大量自治系统进行互联的唯一的 IP 域间路由协议，现在使用的是第 4 版。内部网关协议在自治系统内部提供了路由智能。Internet 的每个自治系统可以允许采用任何适合的 IGP。除 EGP（过时的路由协议）和 BGP 之外，所有其他单播协议，如 IGRP、EIGRP、RIP、OSPF 和 IS-IS 都是 IGP（见表 1-4）。

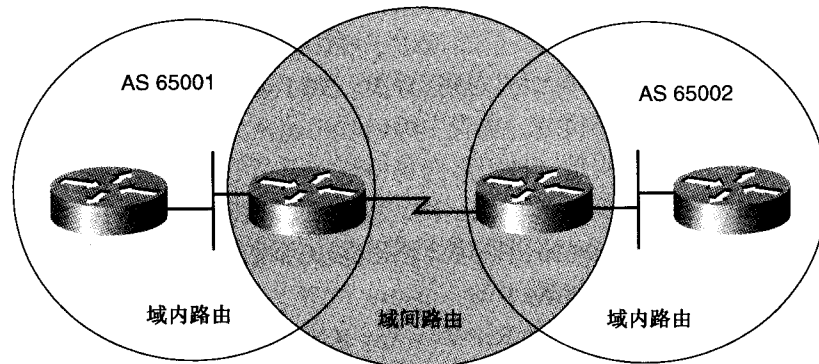


图 1-9 域内和域间路由

表 1-4

IGP 和 EGP 分类

	内部网络协议		外部网关协议
距离向量	高级距离向量	链路状态	路径向量
RIP-1	EIGRP	OSPF	BGP
RIP-2		集成的 IS-IS	
IGRP			

为了提供比 RIP 所支持的简单跳数更好的度量标准，Cisco 公司开发了内部网关路由协议（IGRP）。IGRP 引入了由几种参数组合而成的复杂度量标准：

- 带宽。
- 延迟。
- 可靠性。
- 负载。
- 最大传输单元（Maximum transmission unit, MTU）。

Cisco 还将 IGRP 发展成增强内部网关路由协议（EIGRP）。EIGRP 通过使用备份路由获得了比 IGRP 更好的收敛性。备份路由即可行后继（feasible successor）路由，它们在优先路由丢失后被装入路由表。与 IGRP 不同，EIGRP 支持 VLSM。

OSPF 和 IS-IS 都是在大规模 IP 网络经常使用的 IGP 协议。在开始时，IS-IS 是当作无连接网络协议（Connectionless Network Protocol, CLNP）设计的路由协议，但是后来被用于路由 IP，与此同时，IETF 将 OSPF 协议规范化。OSPF 和 IS-IS 均为链路状态协议，而 RIP、IGRP 和 EIGRP 均为距离向量协议。

OSPF 和 IS-IS 是在路由计算时使用最短路径优先（Shortest Path First, SPF）算法（根据 Dijkstra 的命名）的链路状态协议，当网络变化时使其收敛较快。

两种协议都支持两层分级路由结构。OSPF 和 IS-IS 是非常相似的协议，具有相同的能力。它们只是在结构上有所不同，这些内容已超过本书范围。

有趣的是：OSPF 完全只是为 IP 设计的，OSPF 分组封装在 IP 分组中。而 IS-IS 是为 CLNP

设计的,同时也支持 IP。但 IS-IS 分组没有封装在 IP 分组中,而是直接传给数据链路层协议。

本章的下一小节将对路由协议进行另一种分类:距离向量协议和链路状态协议。

1.3.4 距离向量协议和链路状态协议

本节从另一个角度讨论路由协议。前面论述了协议的有类与无类、IGP 与 EGP 这种普通分类。本节将基于设计和操作对路由协议进行分类。表 1-4 的第二行将协议分成 4 类,有两类比较特殊——距离向量和链路状态。从表中可以看出这两大类多用在 IGP 中。

EIGRP 与 IGRP 相同,其实是一种距离向量协议,只是它在自己的那一类中被当作先进的距离向量协议,因为它有很多优点,如:支持无类路由和快收敛性。BGP 在其自身类别中属于路径向量协议,因为它是域间路由协议,使用路由经过的一组自治系统组成的 AS 路径属性作为路由比较和选择的关键因素。

RIP 的版本一和版本二(RIP-1 和 RIP-2)以及 IGRP 属于距离向量协议,因为它们使用基于 Bellman-Ford 算法的路由计算方法。Bellman-Ford 算法在图论中用于计算有向图中两个节点之间的最短距离。有向图是通过有向链接互联的一组点的集合,这很像网络中的结点和链接。运行距离向量路由协议的路由器使用 Bellman-Ford 算法确定到网络中所有已知结点的最短路径。

OSPF 和集成的 IS-IS 都是链路状态协议,使用最短路径优先算法(Dijkstra)来进行路由计算。与 Bellman-Ford 算法相似,Dijkstra 算法用另一种方法来计算有向图中两点之间的最短距离。

EIGRP 没有继承它的前身——IGRP 所使用的 Bellman-Ford 算法,而是使用 Cisco 系统专用的扩散更新算法(Diffusing Update Algorithm/DUAL)优化路由计算。

路由计算协议所使用的算法类型经过了很长的时间才影响到协议的效率和收敛速度。后面将继续研究距离向量协议和链路状态协议的概念和操作规则。

1. 距离向量路由概念

本小节回顾了距离向量路由协议操作的重要概念,如度量、计数到无穷大、水平分割、保持计时器和触发更新。这些概念是对一般路由功能评价的标准,如:稳定性、收敛速度和循环避免。

距离向量度量

在 Bellman-Ford 算法中,每个路由器向所有已知的目的地址,从路由器自己的角度来说,向所有邻居广播最优路径。为路由器之间的链接赋予一个称为开销或度量的值。度量值由链接的特征,如跳数、带宽、延迟、可靠性、代价等等来决定。两个直接相连的结点之间的链接的跳数可以是任何值,但通常是 1。从任何路由器到已知目的地址的特定路径相关联的度量值是这条路径上所有链接的度量值之和。一般情况下,具有最小度量值的路径是最优路径。路由器可能有很多邻居,从而到达同一目的地址可能接收到多条路径。然后,路由器会计算每条路径的度量并根据最低度量值之类的标准选出最优路径。

RIP 把跳数作为度量,到任何可达目的地址的最大跳数是 15。度量值若为 16 或更大则代表无穷大。因此,跳数 15 定义了 RIP 网络可到达的最大宽度。这对基于 RIP 的网络规模加以限制,也就意味着 RIP 只适合于小规模的网络。跳数是特指从某个源到目的的结点

数，而没有考虑实际网络的特征，如：带宽、延迟和代价等。

另一种距离向量协议 IGRP 使用了只考虑诸如带宽、最大传输单元、链路可靠性和路径延迟等网络相关特征的度量系统。出口方向的每个链接的度量是通过考虑所有特征的公式计算得到的。这种多路径度量也称为复合度量。

不论从任何角度说，Bellman-Ford 算法都是使用由代价（度量）和每个路由器的下一跳信息组成的向量（距离向量）决定网络中的最优路径。再重复这个步骤计算接收到的路由中所有路径的代价，选出每个路由的最小代价的向量。因此，基于 Bellman-Ford 算法的路由协议称为距离向量协议（distance vector protocol）（见表 1-4）。

路由收敛

在拓扑变化时，路由器以前的某些最优路径会失效。然后，路由器使用新的和已存在的的信息确定所有受影响站点目的地址的替代最优路径。网络拓扑变化引起的通过重新计算路由而发现替代路由的行为称为路由收敛（routing convergence）。路由收敛可以由路由器失效、链接失效或者管理度量调整等事件触发。

RIP 和 IGRP 等距离向量协议与链路状态协议相比相对简单。但是，简单是要付出代价的。这种协议容易形成路由环，因为每个路由器根据邻居广播的最优路径确定自己的最优路径。路由环就是两个结点在指向某个相同目的的路径中互相将对方当作下一跳。路由环最主要的影响是延长了路由器确定一条路由失效并选择替代路径的时间。路由环还影响收敛时间。因此，应该尽可能快地将无用路由移出网络。下面讨论距离向量协议采用的防止或限制路由环影响并加快收敛的几种方法。标题如下：

- 计数到无穷大。
- 使用保持计时器。
- 使用水平分割和毒性逆转。
- 使用触发更新。

避免循环

运行距离向量协议的路由器决定向它广播路由的邻居相关的最优路径。距离向量协议的操作机制，尤其是距离向量协议广播路由的方式，使路由环境非常容易受到路由环的影响，例如一个运行距离向量协议的路由器向所有活跃接口广播路由更新。当一个路由器以这种方式广播所有已知路由时，被广播路由可能被发送回该路由器学习这条路由的源。这样，两个邻居结点就有可能将对方当成到某个目的地址最优路径上的下一跳。这种导致路由环的情况可以从图 1-10 中看出。

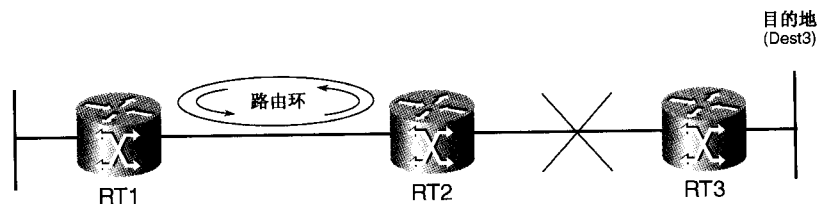


图 1-10 距离向量环境中的路由环

在图 1-10 中，RT1、RT2 和 RT3 串行地相互连接，将跳数作为度量值。RT3 将与目的链路相关的路由（Dest3）广播给 RT2，跳数是 1。RT2 认为到 Dest3 的距离是 2，并将它广播

给 RT1。RT1 将 Dest3 的跳数计为 3，并将 RT2 作为下一跳。然后，RT1 会将 Dest3 广播回给 RT2。因为这条路由的度量值（4 跳）大于从 RT3 收到的路由度量（两跳），RT2 没有使用这条路由。但是，如果 RT2 和 RT3 之间的连接断开，RT2 会移出原来的路由并装入到 Dest3 的度量为 4 的另一条路由，RT1 是下一跳。同时，RT1 也有将 RT2 作为下一跳的路由。这样，路由循环产生了，从 RT1 或 RT2 到 Dest3 的所有分组像乒乓球一样在两个路由器之间来回传送，直到分组中的生存时间（Time To Live TTL）过期。路由循环破坏了路由，最好在路由循环刚出现时尽快地限制它们。距离向量协议使用计数到无穷大的方法减少路由循环的影响。下一部分将讨论这种方法。

计数到无穷大

为了防止存在时间不定的路由环，距离向量协议对路由度量加强了限制：允许路由器在路由的度量值达到某一值时将该路由宣布为无效。如图 1-10 所示，RT1 和 RT4 向对方广播到达 Dest3 的路由，每次将从对方得到的路由的跳数在重新广播之前增加 1。这样，与 Dest3 相关的度量会持续增长。计数时给度量值设定了上限，超出这个限制则被认为无穷大，相应的路由项失效。RIP 的上限是 15。

保持计时器

当主路由失效时，保持计时器延迟寻找替代路由的路由响应行为。当路由器认为一条路由失效时，它将该路由在一段时间内置为保持（holddown）状态，这段时间就是保持时间。在保持时间内，即使替代路由可以使用，路由器也不会选择替代路由。处于保持状态的路由以无穷大的度量值被广播以利于更快地从网络中清除。清除无用路由有利于减少路由环的影响。

可以用图 1-10 来解释这种机制。当 RT2 从 RT3 收到的路由因为它们之间的连接断开而配置为无效时，RT2 将 Dest3 设为保持状态。因为 Dest3 处于保持状态，RT2 不会使用从 RT1 来的替代路由，而是将带有一个度量值的 Dest3 广播给 RT1。这促使 RT1 将 Dest3 从路由表中去掉。当保持计时器超时，RT1 和 RT2 已经将 Dest3 从路由表中移出，从而避免了潜在的路由循环。

使用保持机制还可以避免对导致连接抖动的环境相关故障作不必要的处理，并且可以明显地减少距离向量协议的收敛时间。

水平分割和毒性逆转

路由环主要是由重新发送回源地地址的路由项引起的。例如，在图 1-10 中，RT1 将路由 Dest3 送回 RT2 导致 RT1 和 RT2 之间的环，这使得 RT2 误认为 RT1 是到达 Dest3 的替代路径上的下一跳。

水平分割防止路由器将路由项从接收它的接口广播出去。如果使用了水平分割，那么 RT1 就不会将 Dest3 通过 RT1 和 RT3 之间的连接广播到 RT3（见图 1-10）。

毒性逆转与水平分割的原理基本相似，只是它允许路由从接收它的接口作为不可到达的路由项（度量值为无穷大）广播出去。这样，路由在相反方向上具有抑制。在图 1-10 中，如果使用了毒性逆转，RT1 将 Dest3 广播给 RT2，但是度量值是无穷大（对于 RIP 来说是 16 跳）。

使用毒性逆转会在很多路由器广播具有毒性的消息时过多地浪费带宽。但是，这种方法可以通过取代保持机制来加快路由收敛。这种情况下，替代路由传回源端时需要具有无穷大的度量值，这在原路由丢失时简化了寻找替代路径的操作。

定时更新和触发更新

运行如 RIP 和 IGRP 等距离向量协议的路由器以一定时间间隔广播路由表的全部内容。定时广播很大的路由表对大型网络是有着重要影响的。例如,即使没有任何路由变化出现,RIP 缺省情况下每隔 30s 向每个活跃接口广播所有路由。IGRP 的缺省更新时间间隔是 90s。

如果只是广播定时更新,路由器间不能很快地相互通知网络变化,从而影响收敛时间。另外,保持时间与更新间隔相关。因此,间隔越大,路由更新所消耗的带宽就越少,但是收敛时间就会越长。

触发(或者快速)更新通过网络发生变化时未等待到定时更新的时间点就立即发送更新,从而消除了定时更新带来的收敛延迟。快速更新发生在网络的结点和结点之间,减少了整个网络范围的收敛时间,即使有时不是很明显。

定时更新和触发更新的相交互可能导致某些不可预见的行为。

2. 链路状态协议

链路状态协议相对来说更为现代,它们在设计时融入一些功能克服了上面讨论的距离向量协议的许多缺点。因此,链路状态协议更先进,同时为有效地完成操作也需要更多的内存和处理器资源。因为具有更快的收敛、递增更新和分级的体系结构等特性,链路状态协议更适用于大型网络。IP 网络经常使用 OSPF 和 IS-IS 两种链路状态协议。

与共享最优路由信息的距离向量协议不同,链路状态协议为了使路由器得到网络间的拓扑结构布局在路由器之间交换拓扑(链路状态)信息。链路状态网络中的路由器通过迅速响应网络拓扑变化,不用避免循环或受保持机制和计数到无穷大机制的限制,因而比运行距离向量协议的路由器收敛更快。例如,RIP 和 IGRP 的收敛时间以分钟计,而在相同情况下 OSPF 和 IS-IS 的收敛时间则以秒来计。

链路状态协议为提高缩放性,支持将网络划分成区域(area)的分级体系(见图 1-11)。区域内的路由处于路由分级体系的第一级。各区通过骨干区互联,骨干上的路由组成了分级结构的第二级。

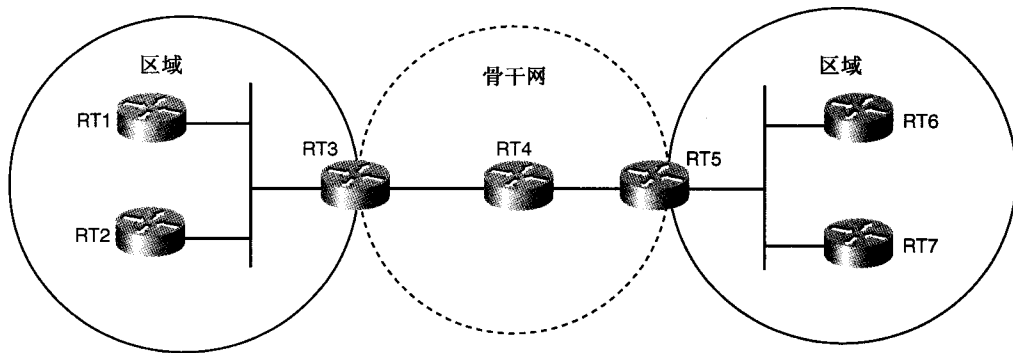


图 1-11 链路状态协议的区域和分级

处于同一区域的路由器或骨干路由器共享集中在链路状态数据库中的链路状态信息。在各自的数据库上运行最短路径优先算法可以看清区域和骨干的拓扑结构。这个程序还可以产生 IP 路由和转发表用到的最优路由。第 8 章“理解 OSPF 协议”和第 10 章“理解 IS-IS 协议”描述了链路状态协议的操作,并详细地介绍了相关协议。

链路状态协议的度量

OSPF 和 IS-IS 都使用链路带宽的度量值来进行度量。OSPF 更胜一筹，它能实现接口带宽到链路代价的自动转换。IS-IS 所有接口的缺省度量是 10。链路相关的度量或代价还可以手工配置。路由的度量是到目的地址的所有输出链路的度量之和。

第 8 章和第 10 章分别详细讲解了 OSPF 和 IS-IS 的度量。

1.4 路由协议的管理距离

本章前面各节在较高层次上从设计、体系结构和操作等角度概述了 IP 路由协议。本节简单介绍一些与实现相关、影响 Cisco 路由器协议操作的一些问题。每个协议的操作和配置细节在专门讲述协议的章节中可以看到。

Cisco 的 IOS 软件提供普通的命令资源配置和启用 IP 路由协议的功能。诸如 distance、distribute-list、redistribute、route-map、policy-map、access-list、prefix-list、offset-list 等命令通常称为协议无关命令（Protocol-independent command），因为它们可以在 Cisco 的 IOS 软件中以不同方式使路由协议功能等特征生效。在对路由协议的应用中，协议无关命令可以过滤路由、启用重发布、配置缺省路由以及实现不同的路由策略。在 www.cisco.com 上可以更详细地查到这些命令。本节讨论 distance 命令以及它所支持的特征——管理距离（administrative distance）。

前述的所有 IP 路由协议都可以在 Cisco 路由器上同时独立操作。通常 IP 网络只需要一个 IGP（OSPF 或 IS-IS）与 BGP 同时运行。当然，也可以根据网络的现状和历史情况使用多个 IGP 协议满足路由需要。

管理距离是 Cisco 专有的，区分同一网络中不同路由源得到的路由的方法。它提供了一种对路由信息源的可信度进行区分的简单机制。Cisco IOS 为了使从某个路由源得到的路由优于从另一个路由源得到的相似路由而赋给路由源一个数字值。管理距离值小的源优先使用。如果多个协议提供了同一条路由，那么从管理距离最低的源发出的那条路由将被放入路由表。表 1-5 列出了 IP 路由源的缺省管理距离。可以用 distance 命令修改缺省值。

表 1-5

IP 路由协议的管理距离

路由源	管理距离
连接接口	0
从接口流出的静态路由	1
到下一跳的……	1
EIGRP 汇总路由	5
外部 BGP	20
内部 EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP-1/RIP-2	120
EGP	140

续表

路由源	管理距离
外部 EIGRP	170
内部 BGP	200
未知	255

1.5 路由器的快速转发

虽然本书是介绍路由协议以及与路由相关问题的解决方法的,但是在本章也会简要阐述当今网络对于快速转发的需要,以及为此而研制出的路由器处理分组的多种绝妙方法。这些方法都对基本的基于 IP 路由表的决策过程进行了扩展。路由表对路由过程起着非常关键的作用,路由器不是直接使用路由表的内容,而是把路由表中的路由信息转换成数据结构存储起来,从而优化了高速分组转发。Cisco 提供了多种高速转发机制,如快速交换、最佳交换和 Cisco 特快转发 (Cisco Express Forwarding,CEF)。

通常需要通过检查快速转发表,如 CEF 转发信息库 (Forwarding Information Base,FIB) 和邻居表数据库处理路由问题。本书不对快速转发机制进行详细介绍。在 Cisco 的网站 www.cisco.com 上可以看到关于此类问题的更多信息。

1.6 小 结

本章是引言部分,主要回顾了有关 IP 路由的概念并解释路由对于无连接网络中信息传送的重要性。提供无连接信息传送的 IP 等协议以信息块(即数据报)的形式传送数据。IP 数据报也称为分组。分组由负载和头组成。IP 分组头包括允许分组在网络上通往目的地的最优路径上进行独立路由的目标地址。IP 是网络层协议,处理和转发分组的路由器运行自动收集 Internet 网络信息的路由协议。

IP 有类编址和无类编址的思想引出了关于 VLSM 和 CIDR 的讨论。VLSM 和 CIDR 适用于有效地址分配,本书中也讲到了它们的作用。

本章的后一部分讨论了动态路由协议的各种分类法:单播和组播、无类和有序、IGP 和 EGP 以及距离向量和链路状态等。对距离向量协议和链路状态协议的重要特征进行了讨论和比较。

对 Cisco IOS 协议无关命令的简单介绍引出了与路由协议相关的管理距离。管理距离是区分路由协议源的机制,它为不同的路由协议提供了 IOS 缺省信任度。

最后一节简单介绍了路由协议如何收集转发时用到的路由信息,并指出 Cisco 路由器将路由表的信息转化成快速分组转发使用的数据结构。

1.7 复 习 题

1. 什么是无连接数据网络?

第 2 章

理解 RIP 协议

本章涵盖下列有关 RIP 的关键主题：

- 度量，
- 计时器。
- 水平分割。
- 带毒性逆转的水平分割。
- RIP-1 分组格式。
- RIP 行为。
- 为什么 RIP 不支持非连续网络。
- 为什么 RIP 不支持变长子网掩码。
- 缺省路由和 RIP。
- RIP 协议扩展。
- 兼容性问题。

RIP(Routing Information Protocol, 路由信息协议)是使用跳数作为度量的距离向量协议。该协议非常简单，适用于小型网络。RIP 类似于由 UNIX 的 FreeBSD 版本发布的 gated。在写 RIP 版本 1 (RIP-1) 的 RFC 之前，存在多种 RIP 版本。

注意：

跳数即经过的路由器个数。例如，跳数为 2 就是目的地在两个路由器之后。

RIP 是有类协议，也就是说，在路由更新时不携带子网掩码信息。因为它不带有任何子网掩码信息，所以不能支持变长子网掩码和非连续网络。RIP 使得设备之间能够交换与其直接相连的网络信息，以及从其他 RIP 设备获得的其他网络信息。

RIP 每隔 30 秒发送一次路由信息，它的缺省更新计时器是 30 秒。这个计时器可以配置。保持计时器决定路由器在刷

2. 为什么无连接网络环境需要路由？列出路由器获得分组路由到目的地的信息的两种方法。
3. 内部网关协议（IGP）和外部网关协议（EGP）的功能之间有什么区别？
4. 列出基于操作方法和路由算法主要的两组 IP 路由协议。并且为每一类举两个例子。
5. 简要介绍链路状态路由协议的操作。
6. 无类和有类路由协议的主要区别是什么？为每种路由协议举一个例子。
7. Cisco 路由器的路由协议管理距离有何作用？
8. IS-IS 和 OSPF 的管理距离值分别是多少？
9. 如果路由器运行 OSPF 和 IS-IS 协议，并且这两个协议包含有相同的路由，那么 IP 路由表中将会使用哪个协议提供的信息？

1.8 参考文献

- Bates, T., R. Chandra, Y. Rekhter, and D. Katz. "Multi-Protocol Extensions for BGP4." RFC 2858, 2000.
- Bennett, Geoff. *Designing TCP/IP Internetworks*. New York, NY: John Wiley & Sons; 1997.
- Callon, R. "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments." RFC 1195. IETF 1990.
- Fuller, V., T. Li, J. Yu, and K. Varadhan. "Classless Interdomain Routing (CIDR): An Address Assignment and Aggregation Strategy." RFC 1519. IETF 1992.
- Hall, Eric A. *Internet Core Protocols: The Definitive Guide*. Sebastopol, CA: O'Reilly and Associates, 2000.
- Hedrick, C. "Routing Information Protocol." STD 34, RFC 1058, 1988.
<http://www.6bone.net/>
<http://www.cisco.com/warp/customer/701/3.html>. "Understanding IP Addresses."
<http://www.cisco.com/warp/public/103/index.shtml>
- Huitema, Christian. *Routing in the Internet, 2nd Edition*. Upper Saddle River, NJ: Prentice Hall, 2000.
- ISO 10589. "Intermediate System-to-Intermediate System Intradomain Routing Information Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service." (ISO 8473.)
- Li, Rekhter. "Border Gateway Protocol Version 4 (BGP 4)." RFC 1771, 1995.
- Maufer, Thomas. *Deploying IP Multicast in the Internet*. Upper Saddle River, NJ: Prentice Hall, 1997.
- Miller, Philip. *TCP/IP Explained*. Woburn, MA: Digital Press, 1997.
- Naugle, Mathew. *Network Protocol Handbook*. New York, NY: McGraw Hill, 1994.
- Perlman, Radia. *Interconnections 2nd Edition*. Reading, MA: Addison Wesley, 1999.

Reynolds, J. and Postel, J. "Assigned Numbers." RFC 1700. IETF 1994.

Rekhter, Y., B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. "Address Allocation for Private Internets." RFC 1918. IETF 1996.

新路由表的信息之前需要等待的时间。

RFC 1058 为 RIP 提供了一个使用 Bellman-Ford 算法来计算度量的规范。

2.1 度 量

RIP 的度量基于跳数，可以在 1 和 15 之间。度量为 16 表示无穷大，即该路由不可到达。问题是为什么要将这个度量值选为 16 呢？为什么不是 17 或 18 呢？RIP-1 的度量字段中分组格式只有 32 位。这意味着 RIP 在理论上可以支持 2^{32} 跳。虽然这个数字很大，但是，将度量值选择为 15 是为了避免计数到无穷大（也称为路由环）的问题。在拥有几百个路由器的大型网络中，如果将度量的无穷大值配置过高，路由环就会导致很长收敛时间。选择数字 16 就是为了减小收敛时间。将跳数限制为 15 的另一个原因是 RIP 是专门为小型网络设计的。它并不适合跳数会超过 15 的大型网络。

2.2 计 时 器

像所有的距离向量协议一样，RIP 每隔 30 秒发送一次定时更新。定时更新包括对整个路由表的广播。更新计时器控制着 30 秒的时间间隔。RIP 使用下列计时器：

- 更新（Update）计时器——每个更新之间的间隔时间。该值可配置，缺省为 30 秒。
- 失效（Invalid）计时器——经过该时间之后可疑路由变为失效。缺省为 180 秒。
- 保持（Hold-down）计时器——用来减小路由表中安装不正确路由的可能性的时间。缺省为 180 秒。
- 清空（Flush）计时器——在经过这段时间之后路由被移出路由表。缺省为 240 秒。

2.3 水平分割

水平分割是一种避免产生路由环的技术。如果使用了水平分割，从某个接口学到的路由就不会广播回该接口。例如，图 2-1 所示的路由器 1 以度量为 1 从相邻路由器 2 接收网络 X 上的更新，在使用水平分割时就不会把网络 X 的信息向路由器 2 广播。而如果没有使用水平分割，路由器 1 将会把网络 X 以度量值 2 向路由器 2 广播。如果网络 X 断开，路由器 2 则会认为路由器 1 仍然能够到达网络 X，所以将发往网络 X 的分组向路由器 1 发送，导致错误。

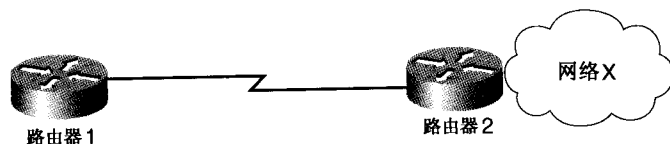


图 2-1 使用水平分割的例子

2.4 带毒性逆转的水平分割

另一种避免产生路由环的技术称为带毒性逆转的水平分割。如果使用这种技术，从某个接口学到的路由会发送回该接口，只是这些路由已具有毒性，也就是说，它们的度量值是 16（不可达）。在图 2-1 中，路由器 1 从路由器 2 接收到度量为 1 的网络 X 的更新信息。这时，如果使用带毒性逆转的水平分割，路由器 1 就会把网络 X 的信息送回路由器 2，但度量值是 16，即无穷大。

带毒性逆转的水平分割只会在出现链接失败时使用。它也可用在通常情况下，但不提倡这种做法，因为它会增大路由表。

2.5 RIP-1 分组格式

RIP 数据报最大为 512 个八位组。第一个字节是命令字段，可以为 rip update request 和 rip update response。下一字节是版本字段，值为 1 表示 RIP-1。下两个字节必须为 0。之后的两个字节用于地址簇标识符，再然后的 14 个字节用作 IP 地址如图 2-2 所示，IP 情况下，14 个字节中只有 4 个字节用作 IP 地址，剩余的 10 个字节在 RIP-1 中不使用，但是在 RIP-2 分组中使用。下面的 4 个字节表示 RIP 的度量，最大为 16。从地址簇标识符到度量字段可以重复 25 次，产生大小为 512 字节的最大 RIP 分组。

0	8	16	31
命令	版本号	必须为零	
地址族标识		必须为零	
IP 地址			
必须为零			
必须为零			
代价			

图 2-2 RIP-1 分组格式

2.6 RIP 行为

当 RIP 发送或接收更新时需要遵守一些规则。这一节讲述发送和接收更新的规则。

2.6.1 发送更新的 RIP 规则

RIP 发送更新时需要执行多次检查。在图 2-3 中，两个路由器都运行 RIP 协议。路由器 1

与两个主网络 131.108.0.0/16 和 137.99.0.0/16 相连。主网络 131.108.0.0 被分成两个子网：131.108.5.0/24 和 131.108.2.0/24，而 131.108.2.0/24 实际上和路由器 2 相连。

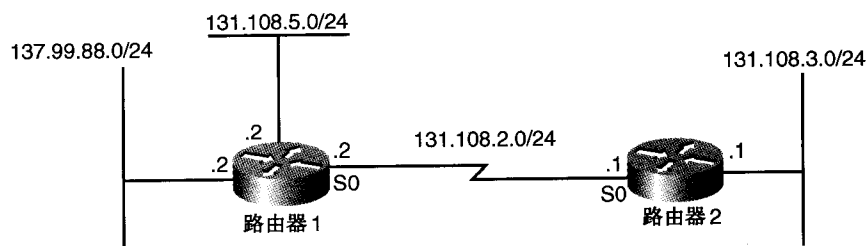


图 2-3 RIP 行为的例子

路由器 1 在向路由器 2 发送 RIP 更新之前执行图 2-4 所示的检查操作。

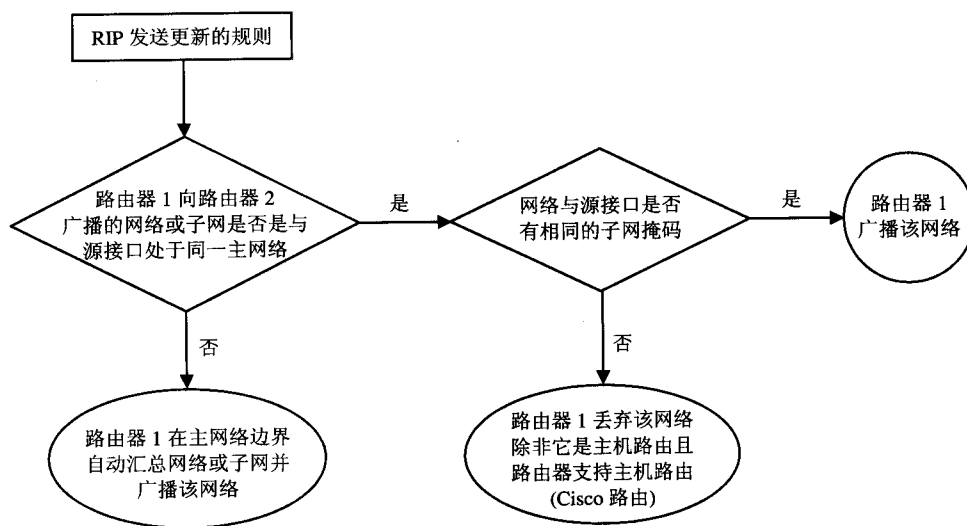


图 2-4 解释 RIP 发送更新规则的流程图

RIP 发送更新时需检查广播网络或子网是否与源 RIP 分组的接口处于同一主网络。如果没有处于同一主网络，网络就会自动聚合。也就是说，RIP 在路由更新时只发送主网络信息。例如，图 2-3 中路由器 1 向路由器 2 发送 RIP 更新时，它将自动把子网 137.99.88.0 合入 137.99.0.0。如果广播网络或子网与源 RIP 分组的接口处于同一主网络，RIP 就需要确定广播子网与源 RIP 更新的接口掩码是否相同。如果掩码相同，那么 RIP 广播该网络，否则，把该网络信息丢弃。

2.6.2 接收更新的 RIP 规则

当接收方收到 RIP 更新时，更新包括子网号、主机地址、网络号或者全 0 信息（表示缺省路由）：

- 子网号（如 131.108.1.0）。
- 主机地址（如 131.108.1.1）。

子网:

- 网络号 (如 131.108.0.0)。
- 缺省路由 (如 0.0.0.0)。

图 2-5 给出了 RIP 接收方的检查操作。

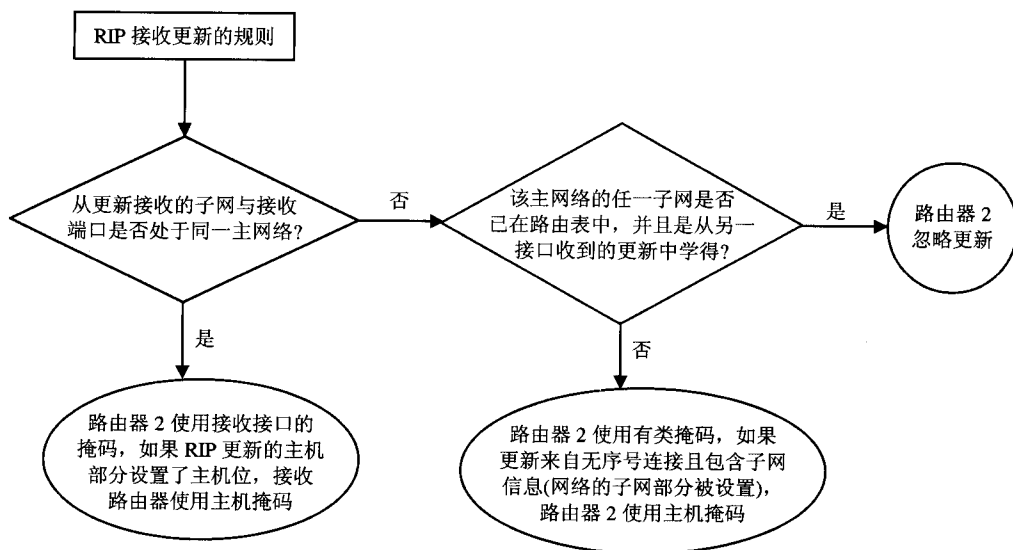


图 2-5 解释接收更新的 RIP 规则的流程图

RIP 接收更新时需要确定更新中的子网与接收接口是否处于同一主网络。如果是的话, 路由器 2 将使用接收接口的掩码。如果 RIP 更新的主机部分被赋值为主机位, 那么接收路由器使用主机掩码。

如果子网属于不同的主网络, RIP 检查该主网络的子网是否已在路由表中, 并确定能否从接收这些更新的接口之外的接口得到它们。注意, 这个更新中的网络必须是主网络。如果答案是“是”的话, 路由器 2 忽略该更新。如果答案是“否”, 路由器 2 使用有类掩码。

如果更新经过没有编号的链接, 它必须包括子网信息 (网络地址的子网部分的位必须被置位)。路由器 2 使用主机掩码。如果更新带有子网广播, 例如 131.108.5.127/25 或 D 类或 E 类地址, 必须忽略该 RIP 更新。

2.6.3 发送更新举例

这一节举例说明 RIP 发送更新的操作。图 2-6 中的两个路由器均运行 RIP 协议。路由器 1 和路由器 2 之间的链接是 131.108.0.0。路由器 1 的以太网接口也是 131.108.0.0。路由器 1 还和另一个主网络 137.99.0.0 相连。

在图 2-6 中, 路由器 1 向路由器 2 发送更新时执行以下检查:

1. 131.108.5.0/24 是否与更新分组的源网络 131.108.2.0/24 处于同一主网络?
2. 答案为是。131.108.5.0/24 与更新分组的源网络 131.108.2.0/24 具有相同的子网掩码?
3. 答案为是。路由器 1 广播网络信息。
4. 137.99.88.0/24 是否与更新分组的源网络 131.108.2.0/24 处于同一主网络?
5. 答案为否。路由器 1 在同一主网络边界汇总 137.99.88.0/24 并广播路由 137.99.0.0。

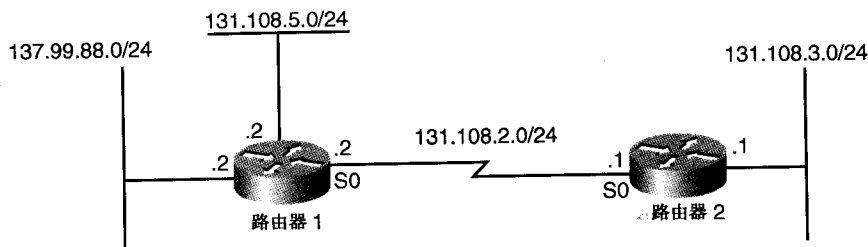


图 2-6 发送和接收更新的 RIP 行为举例

该过程使得路由器 1 其发往路由器 2 的更新中包括 131.108.5.0 和 137.99.0.0。可以在路由器 1 上用 `debug ip rip` 命令查看输出结果，如例 2-1 所示。

例 2-1 debug ip rip 命令输出显示了发送的 RIP 更新信息

```
Router1#debug ip rip
RIP: sending v1 update to 255.255.255.255 via Serial0 (131.108.2.2)
      subnet 131.108.5.0, metric 1
      network 137.99.0.0, metric 1
```

2.6.4 接收更新举例

例 2-2 利用 `debug ip rip` 命令的输出显示了路由器 2 从路由器 1 接收的路由更新。

例 2-2 debug ip rip 命令输出显示了接收的 RIP 更新信息

```
Router2#debug ip rip
RIP: received v1 update from 131.108.2.2 on Serial0
      131.108.5.0 in 1 hops
      137.99.0.0 in 1 hops
```

图 2-6 中，路由器 2 执行下列检查操作以确定接收的网络上使用什么掩码：

1. 已接收的主网络 137.99.0.0 与 131.108.2.0 是否相同，后者是接收更新的接口地址？
2. 答案为否。从其他接口学到的路由表中是否已经有主网络的子网？
3. 答案为否。因为 137.99.0.0 是 B 类地址，路由器 2 使用自然掩码 (/16)。
4. 子网 131.108.5.0 与接收更新的接口 131.108.2.0 子网是否属于同一主网络？
5. 答案为是。路由器 2 使用接收更新的接口的掩码/24。

该过程使网络和掩码进入路由器 2 的路由表，可以用 `show ip route` 命令显示（见例 2-3）。

例 2-3 show ip route 命令输出显示路由器 2 的路由表的网络和掩码

```
Router2#show ip route
R    137.99.0.0/16 [120/1] via 131.108.2.2, 00:00:07, Serial0
     131.108.0.0/24 is subnetted, 3 subnets
R    131.108.5.0 [120/1] via 131.108.2.2, 00:00:08, Serial0
C    131.108.2.0 is directly connected, Serial0
C    131.108.3.0 is directly connected, Ethernet0
```

2.7 为什么 RIP 不支持非连续网络

非连续网络是由被其他主网络分割的同一个主网络组成的。在图 2-7 中, 网络 131.108.0.0 被 137.99.0.0 的一个子网分开, 此处 131.108.0.0 便是一个不连续的网络。

RIP 是有类协议。当 RIP 在不同的网络边界上广播某个网络信息时, RIP 在主网络边界上汇总被广播的网络信息。在图 2-7 中, 当路由器 1 通过 137.99.88.0 向路由器 2 发送包含 131.108.5.0 的更新时, 它把 131.108.5.0/24 转换成 131.108.0.0/16。该过程称为自动汇总。

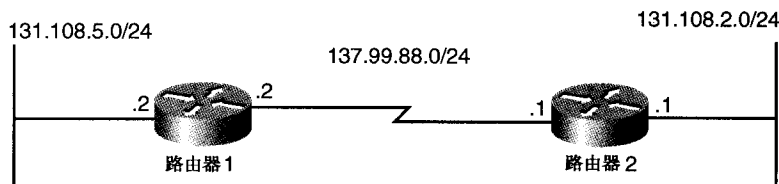


图 2-7 非连续网络举例

路由器 1 在向路由器 2 发送更新前先进行以下几步:

1. 131.108.5.0/24 是否与源更新的接口所在的子网 137.99.88.0/24 处于同一主网络?
2. 答案为否。路由器 1 汇总 131.108.5.0/24 并广播路由 131.108.0.0/16。

路由器 1 上的 **debug ip rip** 命令的输出显示了路由器 1 发送的更新, 如例 2-4 所示。

例 2-4 debug ip rip 命令输出显示了图 2-7 中路由器 1 发送的 RIP 更新信息

```
Router1#debug ip rip
RIP: sending v1 update to 255.255.255.255 via Serial0 (137.99.88.2)
      network 131.108.0.0, metric 1
```

路由器 2 从路由器 1 接收更新前执行以下几步:

1. 接收到的主网络 (131.108.0.0) 与接收更新的接口的子网 137.99.88.0/24 所在的主网络是否相同?

2. 答案为否。在从接收更新的其他接口得到的路由表中是否存在该主网络的任一子网?
3. 答案为是。路由器 2 忽略该更新。

路由器 2 的 **debug ip rip** 命令输出再一次显示了路由器 2 接收的 RIP 更新, 如例 2-5 所示。

例 2-5 debug ip rip 命令输出显示了图 2-7 中路由器 2 接收的 RIP 更新信息

```
Router2#debug ip rip
RIP: received v1 update from 137.99.88.1 on Serial0
      131.108.0.0 in 1 hops
```

路由器 2 的路由表如例 2-6 中 **show ip route** 命令的输出所示, 它表示 RIP 更新已被忽略。131.108.0.0 上的任何子网和网络的唯一入口是与 Ethernet0 直接相连的接口。

例 2-6 debug ip rip 命令输出显示了图 2-7 中路由器 2 的路由表没有反映路由器 1 发送的广播路由

```
137.99.0.0/24 is subnetted, 1 subnets
C      137.99.88.0 is directly connected, Serial0
      131.108.0.0/24 is subnetted, 3 subnets
C      131.108.2.0 is directly connected, Ethernet0
```

在指向指定子网的两个路由器上配置静态路由可以避免更新被忽略。例如对路由器 1 可作如下配置:

```
Router1(config)#ip route 131.108.2.0 255.255.255.0 137.99.88.1
```

在路由器 2 上进行如下配置:

```
Router2(config)#ip route 131.108.5.0 255.255.255.0 137.99.88.2
```

2.8 为什么 RIP 不支持可变长子网掩码

可变长子网掩码 (Variable-length Subnet masking, VLSM) 是指同一网络号可以使用不同子网掩码。RIP 和 IGRP 是有类协议, 它不能在更新中携带子网掩码信息。在 RIP 和 IGRP 发送更新之前, 它用源更新接口的子网掩码检查将被广播的网络的子网掩码。如果两者不匹配就丢弃更新。

下面的例子阐明了这一概念。图 2-8 中, 路由器 1 有三个子网, 两个子网掩码 (/24 和 /30)。

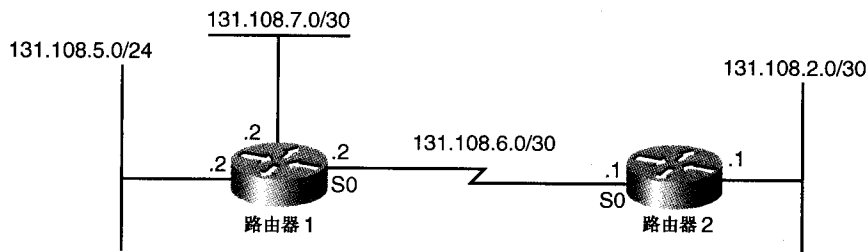


图 2-8 VLSM 网络举例

路由器 1 在向路由器 2 发送更新前执行下列操作:

1. 路由器 1 检查 131.108.5.0/24 是否与引起更新的接口网络 131.108.6.0/30 处于同一主网络。
2. 是属于同一主网络, 所以路由器 1 确定 131.108.5.0/24 与 131.108.6.0/30 的子网掩码是否相同。
3. 因为子网掩码不同, 路由器 1 丢弃该网络, 不广播该路由。
4. 路由器 1 确定 131.108.7.0/30 是否与引起更新的接口网络 131.108.6.0/30 属于同一主网络。
5. 属于同一主网络, 所以路由器 1 进一步确定 131.108.7.0/30 与 131.108.6.0/30 的子网掩码是否相同。
6. 因为两个子网掩码相同, 所以路由器 1 广播该网络信息。

前面几步决定了路由器 1 在发往路由器 2 的更新中只包括 131.108.7.0。例 2-7 中的 **debug ip rip** 命令可以显示出路由器 1 发送的更新。

例 2-7 **debug ip rip** 命令输出显示了路由器 1 发送给路由器 2 的 RIP 更新信息，如图 2-8 所示

```
RIP: sending v1 update to 255.255.255.255 via Serial0 (131.108.6.2)
      subnet 131.108.7.0, metric 1
```

注意，在例 2-7 的输出中，更新包括的子网只有 131.108.7.0。子网 131.108.5.0 具有不同的子网掩码，所以没有将其包括在内。这导致路由器 2 的路由表中包含用 **show ip route** 命令显示的下列入口（见例 2-8）。

例 2-8 从 **show ip route** 命令输出可以看出路由器 2 中的路由表不包括子网 131.108.5.0/25

```
Router2#show ip route
      131.108.0.0/30 is subnetted, 3 subnets
R       131.108.7.0 [120/1] via 131.108.2.2, 00:00:08, Serial0
C       131.108.6.0 is directly connected, Serial0
C       131.108.2.0 is directly connected, Ethernet0
```

为了避免路由更新时清除子网，可以在整个 RIP 网络上使用同一子网或在具有不同子网掩码的网络中使用静态路由。

2.9 缺省路由和 RIP

Cisco 的 RIP 实现支持传播缺省路由，即 0.0.0.0/0。当 RIP 在路由表中发现缺省路由时，自动将其在更新中进行广播。

一定要记住缺省路由必须具有有效的度量值。例如，如果从 OSPF 学到的缺省路由的度量是 20，RIP 将以无穷大（16）的度量值将其广播。所以，在这种情况下，**default-metric** 命令必须用在 **router rip** 命令中，以确保更新中的度量值有效。

无类和有序 IP 路由的概念非常重要，带有缺省路由的尤为重要。使用有序路由时，如果路由器收到一个无法识别的子网的分组，路由表中也没有缺省路由，那么路由器就会丢弃该分组。图 2-9 示意了这种行为。

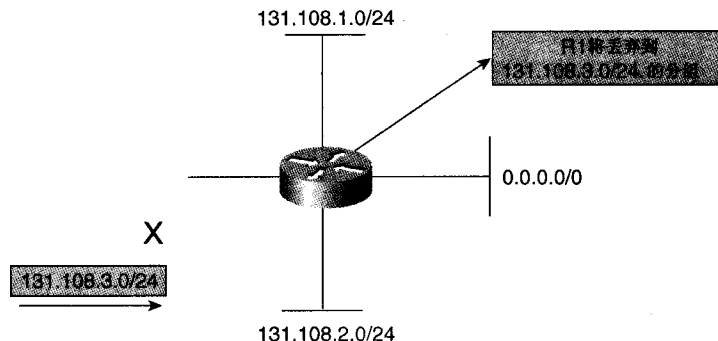


图 2-9 有序 IP 路由

前面几步决定了路由器 1 在发往路由器 2 的更新中只包括 131.108.7.0。例 2-7 中的 **debug ip rip** 命令可以显示出路由器 1 发送的更新。

例 2-7 **debug ip rip** 命令输出显示了路由器 1 发送给路由器 2 的 RIP 更新信息，如图 2-8 所示

```
RIP: sending v1 update to 255.255.255.255 via Serial0 (131.108.6.2)
      subnet 131.108.7.0, metric 1
```

注意，在例 2-7 的输出中，更新包括的子网只有 131.108.7.0。子网 131.108.5.0 具有不同的子网掩码，所以没有将其包括在内。这导致路由器 2 的路由表中包含用 **show ip route** 命令显示的下列入口（见例 2-8）。

例 2-8 从 **show ip route** 命令输出可以看出路由器 2 中的路由表不包括子网 131.108.5.0/25

```
Router2#show ip route
      131.108.0.0/30 is subnetted, 3 subnets
R       131.108.7.0 [120/1] via 131.108.2.2, 00:00:08, Serial0
C       131.108.6.0 is directly connected, Serial0
C       131.108.2.0 is directly connected, Ethernet0
```

为了避免路由更新时清除子网，可以在整个 RIP 网络上使用同一子网或在具有不同子网掩码的网络中使用静态路由。

2.9 缺省路由和 RIP

Cisco 的 RIP 实现支持传播缺省路由，即 0.0.0.0/0。当 RIP 在路由表中发现缺省路由时，自动将其在更新中进行广播。

一定要记住缺省路由必须具有有效的度量值。例如，如果从 OSPF 学到的缺省路由的度量是 20，RIP 将以无穷大（16）的度量值将其广播。所以，在这种情况下，**default-metric** 命令必须用在 **router rip** 命令中，以确保更新中的度量值有效。

无类和有类 IP 路由的概念非常重要，带有缺省路由的尤为重要。使用有类路由时，如果路由器收到一个无法识别的子网的分组，路由表中也没有缺省路由，那么路由器就会丢弃该分组。图 2-9 示意了这种行为。

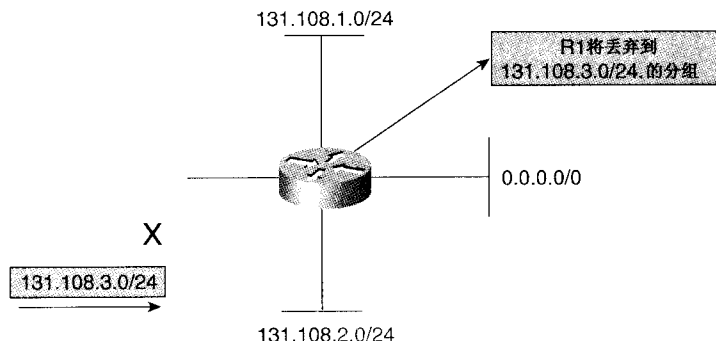


图 2-9 有类 IP 路由

图 2-9 中, 主机 X 向子网 131.108.3.0/24 发送信息。路由器 R1 将丢弃这些分组, 因为它不包含到 131.108.3.0/24 的路由。有类路由的特征也决定了不能将信息送到缺省路由。

如果 R1 启用 IP 无类路由, R1 将会把信息转发至缺省路由。

使用缺省网络或缺省路由时最好使用 IP 无类路由。

2.10 RIP 协议扩展

RIP 版本 2 (RIP-2) 对 RIP-1 进行了一些改进和增强。RIP-2 支持 VLSM 和非连续网络, 并增加了下列功能:

- 路由标记。
- 子网掩码。
- 下一跳度量。
- 组播能力。
- 认证。

图 2-10 给出 RIP-2 的分组格式。这一节会详细讨论每一项新增加的功能和新的分组字段。

0	8	16	31
命令	版本号	路由标记	
地址簇标识符		必须为零	
IP 地址			
子网掩码			
下一跳			
度量			

图 2-10 RIP-2 分组格式

2.10.1 路由标记

路由标记字段有两个字节, 这样 RIP 路由就可以赋予一个唯一的整数值。从路由表中可以看到赋予每个 RIP 路由的路由标记。路由标记在 RIP 重发布时起着非常重要的作用。任何重发布到 RIP 的路由都将通过标记来区分内部 RIP 信息和外部 RIP 信息。

在 RIP 中, 利用路由标记重发布路由时, 在其他协议中, 控制标准路由的重发布变得很容易。当重发布到其他协议时, RIP 路由只须比较赋予它们的标记而不用比较整个路由。

例如, 某路由器的 10 条静态路由在 RIP 中重发布, 标记为 20。这些静态路由将以标记 20 作为外部路由被广播到 RIP。如果另一些 RIP 路由也被重发布到 OSPF, 而 OSPF 只需要 10 条静态路由来重发布, 这时, OSPF 只须简单地匹配标记信息而不用再使用重发布命令列出所有的静态路由。另外, 如果 OSPF 在另一些路由器上把路由信息重发布回 RIP, RIP 就应该拒绝标记为 20 的路由。因此, 匹配标记也可以避免路由环。

因为它

...网络,

分组字

表中可用。任何

变得很自由。

只需要
命令列
RIP 就应

图 2-11 RIP-2 分组格式

2.10.4 组播能力

RIP-2 使用组播把更新发送给所有的邻居。这种做法减少了不必要的广播流量。RIP-2 使用的组播地址是 224.0.0.9。

网络上所有运行 RIP-2 的设备在 IP 地址为 224.0.0.9 的组播 MAC 地址 (01-00-5E-00-00-09) 上监听 RIP-2 组播分组。没有运行 RIP 的设备丢弃 RIP-2 消息以减少不必要的负载。

2.10.5 认证

RIP-2 支持简单的口令认证，以确认 RIP-2 邻居是否可信任。RIP-2 程序根据 RIP-2 分组中的地址簇标识符 (AFI) 确定是否使用认证。RIP-2 头中的 AFI 决定了分组中的其他部分使用哪一种地址。

如果 AFI 值为 0xFFFF，那么剩下的整个 RIP 分组包含认证信息。

图 2-12 给出了使用认证时的分组格式。

0	8	16	31
命令	版本号	未用	
0xFFFF		认证类型	
认证			
认证			
认证			
认证			

图 2-12 RIP-2 分组的认证信息

2.11 兼容问题

RIP-1 和 RIP-2 可以同时在网上运行。在网上同时运行两个协议时需注意以下几点：

- **自动汇总**——RIP-1 和 RIP-2 可以在网上同时运行。介绍 RIP-2 的文档 RFC 1723 建议使用两个协议时关闭自动汇总功能。
- **子网广播**——如果将一个更特定的子网广播到 RIP-1 路由器，该路由器会错误地将其误认为是主机路由更新。
- **查询**——当 RIP-2 路由器收到来自 RIP-1 路由器的查询请求时，它会回复 RIP-1 消息。但是如果该路由器配置为只发送 RIP-2 消息，请求将被忽略。
- **版本字段**——RIP 分组的版本字段决定如何处理 RIP-1 和 RIP-2 分组。

——如果 RIP 分组中版本号为 0，该分组被丢弃，不管接收路由器运行的是哪个版本。

——如果 RIP 分组中版本号为 1，所有的“必须为 0”字段将被检查（参考图 2-9）。如果版本号非 0，分组将被丢弃，不管接收路由器运行的是哪个版本。

——如果 RIP 分组中版本号为 2 且接收路由器运行 RIP-1，接收路由器只查看分组中的相关信息。所有的“必须为 0”字段被忽略。

2.12 小 结

RIP 是一种距离向量协议，它使用 Bellman-Ford 算法动态地计算 IP 路由。RIP 由于跳数 15 的限制适合运行在小型网络中。RIP 是一种在固定时间间隔（30 秒）内与运行 RIP 的其他路由器交换全部路由表的简单 IP 路由协议。在具有更多 IP 路由的更大的网络上，每隔 30 秒发送全部路由表是不现实的。这会为发送者和接收者带来更多的负担，并浪费不必要的带宽和处理时间。因此，RIP 用在跳数小于 15 并且路由也相对较少的小型网络中。

RIP 使用水平分割和毒性逆转的算法来避免环。水平分割通过不把路由广播回送到该路由的接口避免产生路由环。毒性逆转使路由以无穷大的 RIP 度量（16）广播，从而除去可以导致循环或损坏的 RIP 路由。

因为网络的任何变化都需要至少 30 秒才能被广播出去，保持的概念使 RIP 路由表可以等待三倍于广播间隔的时间。这种实现适用于 RIP 路由还没有被广播的时候，因为该路由有可能刚刚超过 30 秒就不能用了。接收路由器需要等待 90 秒才能将路由从路由表中消除。如果一条路由在 90 秒之前到来，它会重新进入路由表并向整个网络广播。

在早期的 IP 网络中，RIP 是小型 IP 网络使用的协议。从那以后，很多比 RIP 更健壮、更实时的新的 IP 协议被开发出来。它们可以适用于路由器超过 15 的网络。诸如 OSPF、IS-IS 和 EIGRP 等新协议的出现使 RIP 几乎在大些的网络中被淘汰。这些新协议在 RIP 受限的收敛性和可扩展性等方面有很多改进，并且支持 RIP-1 所不支持的 VLSM 和非连续网络。

虽然 RIP-2 在很多方面相对于 RIP 有所改进，如路由标记、查询、子网掩码、下一跳、组播和认证等，大型网络还是倾向于使用 OSPF、IS-IS 和 EIGRP 作为 IP 路由协议。

2.13 复 习 题

1. RIP 的最大度量是多少？
2. 为什么 RIP 不支持非连续网络？
3. 为什么 RIP 不支持 VLSM？
4. RIP 的缺省更新间隔是多少？
5. RIP 发送更新时使用什么传输协议和端口号？
6. 水平分割技术的目的是什么？
7. RIP 版本 2 缺省状态下能解决非连续网络的问题吗？
8. RIP 版本 2 使用广播发送更新吗？
9. RIP 是否支持认证？

2.14 深入阅读

可以参考下列 RFC 学到更多关于 RIP 的知识。在 www.isi.edu/in-notes/rfcxxxx.txt 可以在线阅读所有的 RFC，这里 xxxx 是你要阅读的 RFC 的号码。

RFC 1058, "Routing Information Protocol"

RFC 1723, "RIP Version 2"

RFC 2453, "RIP Version 2"

RFC 1582, "Extensions to RIP to Support Demand Circuits"

RFC 2091, "Triggered Extensions to RIP to Support Demand Circuits"

RFC 2082, "RIP-2 MD5 Authentication"

第 3 章

RIP 协议疑难解析

本章涵盖下列关键主题：

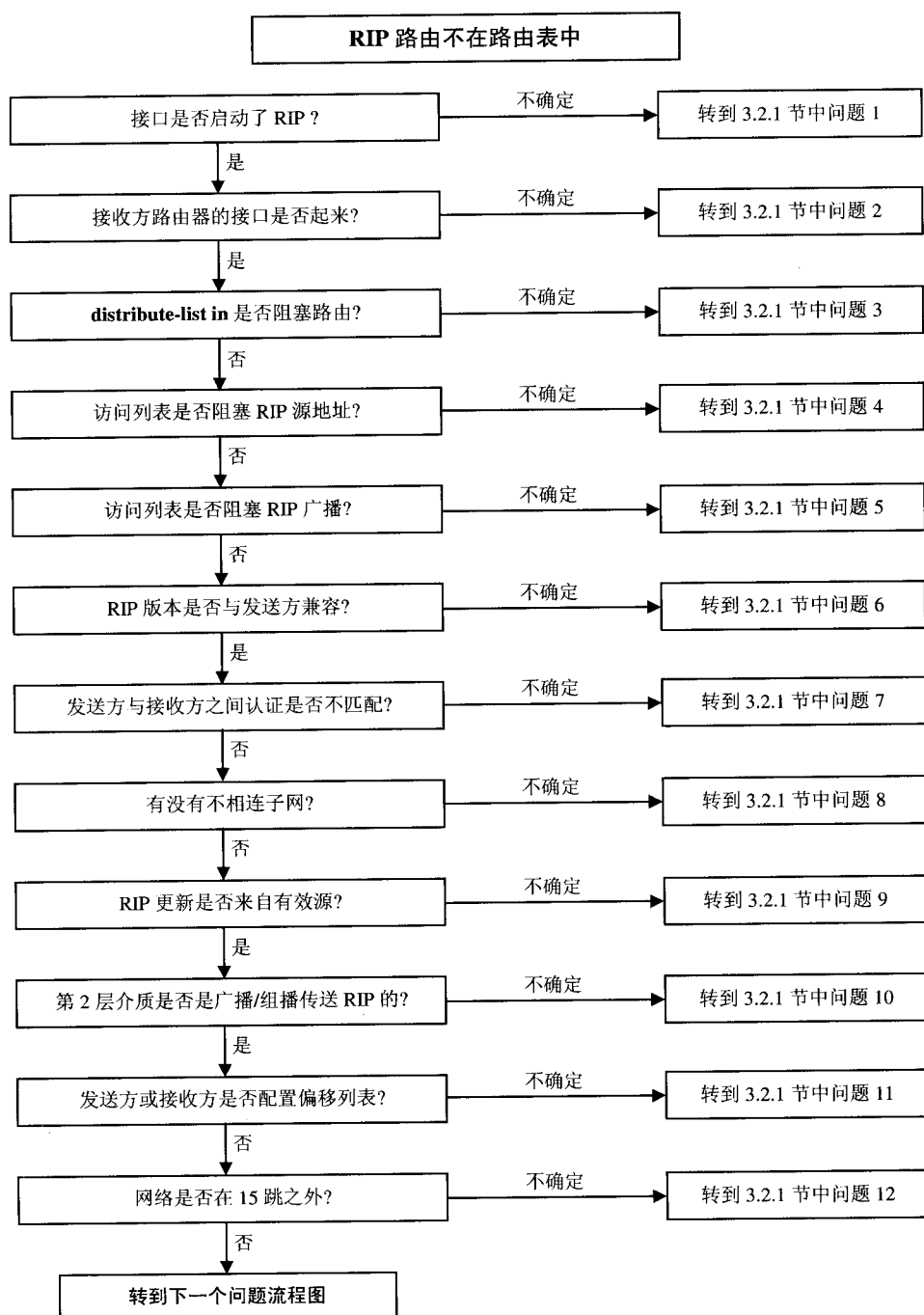
- RIP 路由安装疑难解析。
- RIP 路由公布疑难解析。
- RIP 中路由汇总疑难解析。
- RIP 重发布问题疑难解析。
- RIP 中按需拨号问题疑难解析。
- RIP 路由抖动问题疑难解析。

本章讨论 RIP 中经常出现的问题及解决方法。此时，没有 RIP 错误消息帮助解决 RIP 问题。因此，必须依靠调试、配置和 show 命令。这些方法将在本章必须使用它们的地方予以介绍。后面的流程图说明了如何利用本章使用的方法定位 RIP 的常见问题。

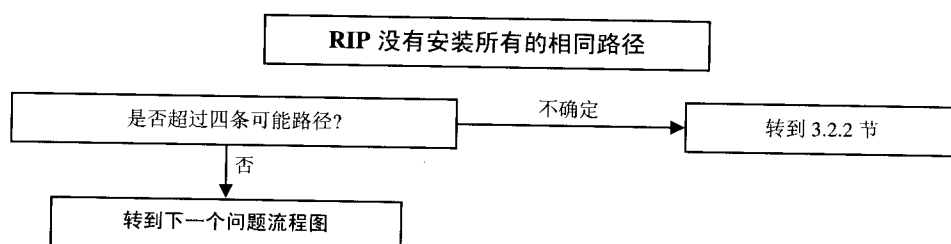
调试工具有时会占用大量的 CPU 时间，并引起网络拥塞。因此，在大型的网络（即运行 RIP 的网络或子网超过 100 个）上最好不要打开调试工具。有时同一问题可能由多个可能的原因引起——例如，第 2 层故障、network 语句错误和发送方漏掉 network 语句等。修复第 2 层和更正 network 语句仍然不能解决网络问题，因为发送方还是缺少 network 语句。因此，如果某种方法不能解决网络问题，就试一试其他方法。RIP 既指 RIP-1，也指 RIP-2。除非明确说明是 RIP-2，本章讨论的问题大多与 RIP-1 相关。

3.1 解决 RIP 常见问题的流程图

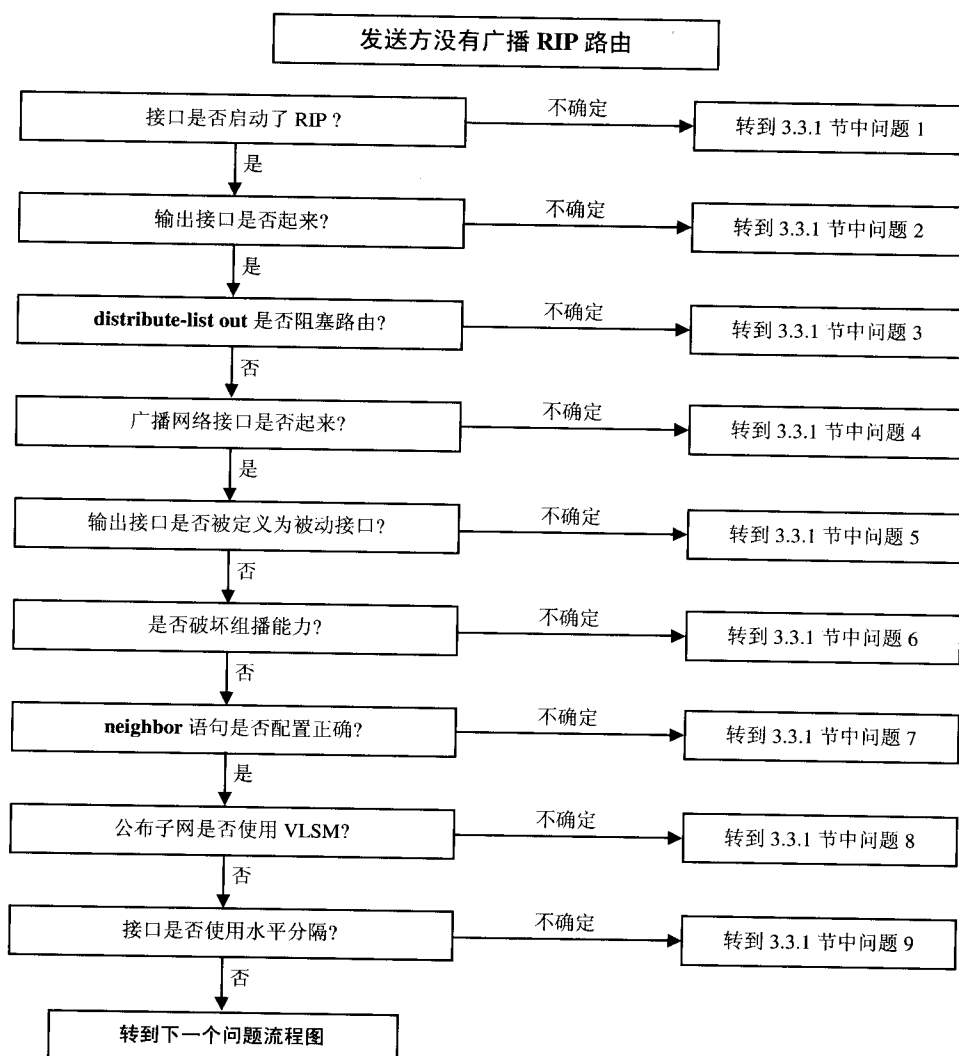
RIP 路由安装疑难解析



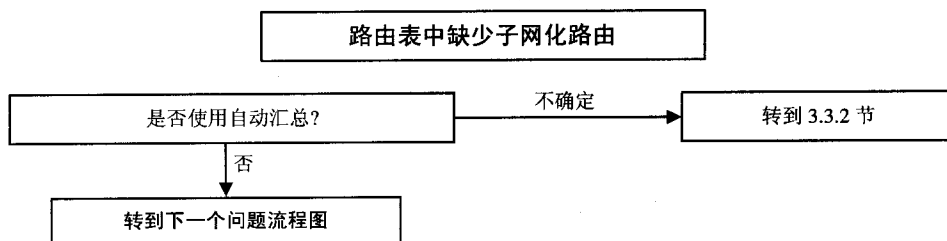
RIP 路由安装疑难解析



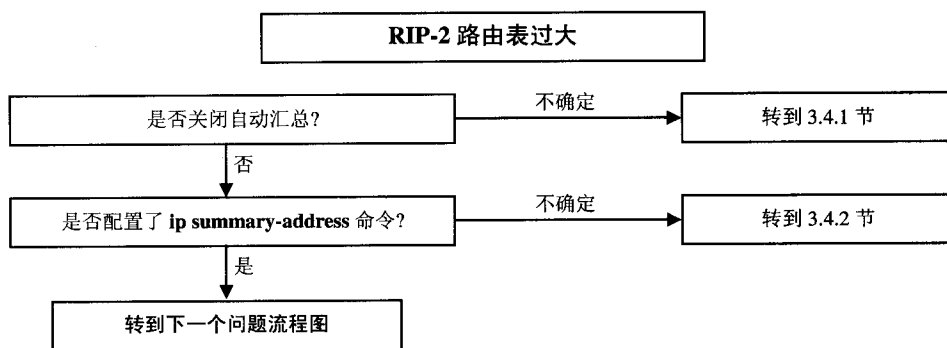
RIP 路由公布疑难解析



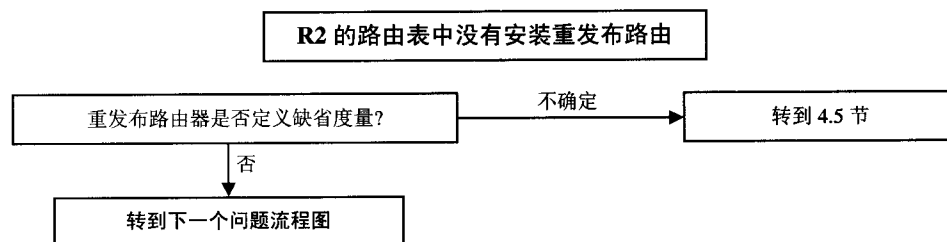
RIP 路由公布疑难解析



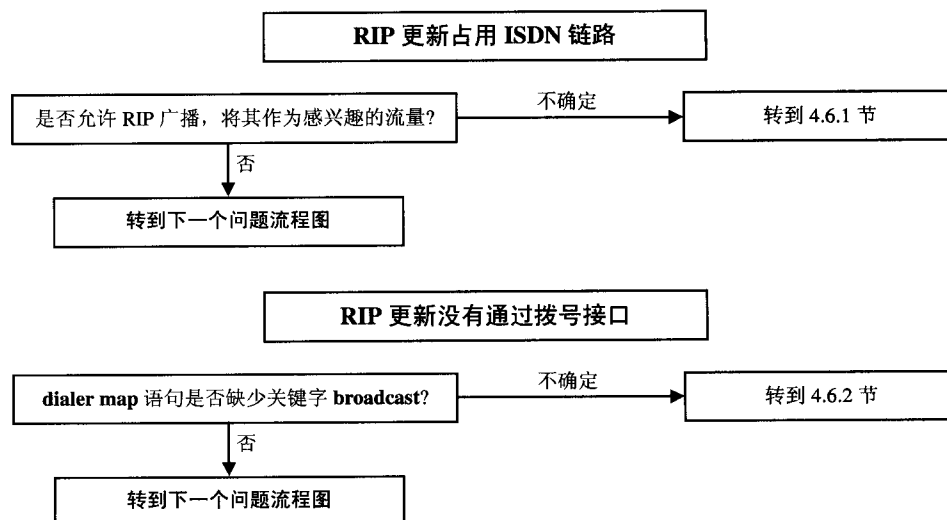
RIP 路由汇总疑难解析



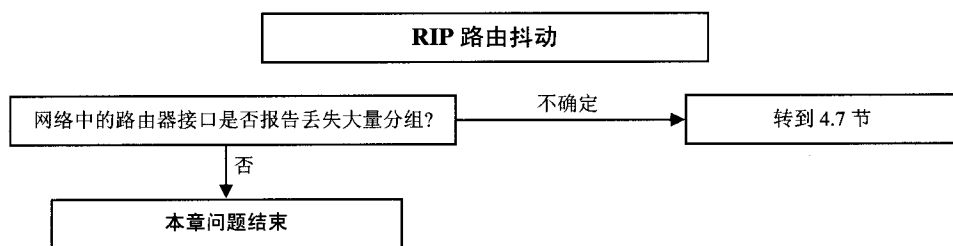
RIP 路由重发布疑难解析



RIP 中按需拨号路由疑难解析



RIP 中路由抖动疑难解析



3.2 RIP 路由安装疑难解析

本节讨论导致 RIP 路由无法安装到路由表中的几种可能情况。在问题解决列表中把本节放在最前面是因为 RIP 最常见的问题就是路由不能安装到路由表中。

如果路由没有安装到路由表中，路由器就不能转发发往路由表中不包含的目的地址的分组。这时就会产生可达性问题。用户开始抱怨不能连接到服务器或打印机。检查这种现象时，首先需要注意的就是“是否已经具有到达用户所抱怨的目的地址的路由？”

路由不能安装到路由表中有三种可能的情况：

- **接收方的问题**——路由器接收到 RIP 更新但没有安装 RIP 路由。
- **中间介质（第 2 层）的问题**——多与第 2 层相关，发送方已经发送了 RIP 更新，但是在传输中间丢失了，所以接收方没有收到。
- **发送方的问题**——发送方甚至没有公布 RIP 路由，所以接收方在路由表中看不到路由。

发送方的问题将在“RIP 路由公布疑难解析”一节中讨论。有两个问题与 RIP 安装相关：

- RIP 路由没有安装到路由表中。
- RIP 没有安装所有的等价路径路由。

第一种情况下，RIP 没有安装任何到某一网络的路径。第二种情况是 RIP 没有安装到某一网络的所有路径。注意，在第二种情况下，目的设备仍然可达，但是它不会列出所有的可能路径。

3.2.1 问题：RIP 路由不在路由表中

路由表必须有将分组发向目的网络的入口。如果没有发向特定目的的入口，路由器会丢弃发向该目的的所有分组。

例 3-1 给出的 R2 的路由表不包含到达网络 131.108.2.0 的入口。

例 3-1 R2 的路由表不包含到子网 131.108.2.0 的 RIP 路由

```

R2#show ip route 131.108.2.0
% Subnet not in table
R2#
  
```

产生该问题的可能原因如下：

- 缺少或使用了错误的 **network** 语句。
- 第 2 层不能工作。
- 分发列表阻塞了路由。
- 访问列表阻塞 RIP 源地址。
- 访问列表阻塞 RIP 广播或组播。
- 版本不兼容。
- 认证字段不匹配 (RIP-2)。
- 非连续网络。
- 无效源地址。
- 第 2 层的问题 (交换机、帧中继和其他第 2 层介质)。
- 定义了很大度量值的偏移量列表。
- 达到 RIP 跳数限制的路由。
- 发送方问题 (下一章讨论)。

图 3-1 提供一个网络场景将用作疑难解析 RIP 路由不在路由表中的问题是前述主要原因的基础。下面几小节将仔细分析如何疑难解析基于特定原因的这个问题。

图 3-1 给出的设置中路由器 1 和路由器 2 之间运行 RIP。

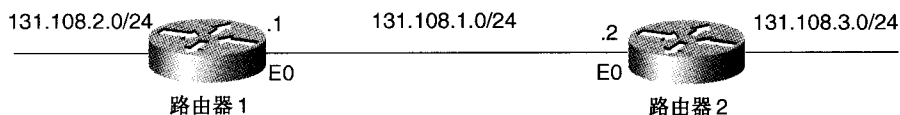


图 3-1 RIP 路由不在路由表中问题的拓扑结构示例

1. RIP 路由不在路由表中——原因：缺少或错误的 **network** 语句

当确定路由表中缺少 RIP 路由时，下一步就是去寻找原因。路由表中缺少路由可能有很多原因。本章开始处的流程图可以帮助查找在当时的情况下最有可能的原因。

发现路由表中缺少路由之后首先应该检查路由器的配置。还需要检查 **router rip** 之后是否正确配置了 **network** 语句。

当 **network** 语句配置好后，它会做两件事：

- 在接口上启用 RIP，激活其发送和接收 RIP 更新的能力。
- 在 RIP 更新分组中公布网络。

如果没有配置或错误地配置了 **router rip** 命令之后的 **network** 语句，这种问题就会产生。

图 3-2 给出了由于这个原因出现的问题的解决方法流程图。

(1) 调试和验证

例 3-2 给出了路由器 R2 的配置 (如图 3-1 所示)。在本章中，本例和许多其他例子都使用了回环接口。如果回环接口被其他接口代替，含意不会改变。建议将回环接口看作任何一个已经激活并正在工作而且具有有效 IP 地址的接口。

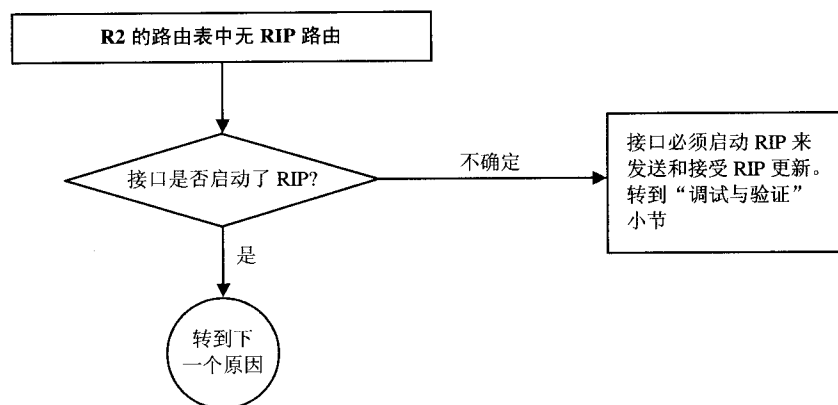


图 3-2 解决问题的流程图

例 3-2 图 3-1 中路由器 R2 的配置

```
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router rip
network 131.107.0.0
!
```

重新参考图 3-1 并与例 3-2 中 R2 的配置比较, 注意 R2 的配置中缺少网络 131.108.0.0。例 3-3 给出了 R2 上的 **show ip protocols** 命令的输出。该输出说明路由信息源同样没有将 131.108.1.1 显示为网关。

例 3-3 show ip protocols 缺少路由信息源的网关信息

```
R2#show ip protocols

Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 11 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
  Automatic network summarization is in effect
  Routing for Networks:
    131.107.0.0
  Routing Information Sources:
    Gateway      Distance      Last Update
  Distance: (default is 120)
```

调试命令

例 3-4 给出了 **debug ip rip** 的输出。在调试中 R2 忽略从 R1 来的 RIP 更新, 因为 Ethernet 0 上没有启动 RIP。原因是在路由器配置模式下 **router rip** 中缺少 131.108.0.0 的 **network** 命令。

例 3-4 debug ip rip 命令的输出表明从路由器 R1 发出的 RIP 更新被忽略

```
R2#debug ip rip
RIP protocol debugging is on
R2#
RIP: ignored v1 packet from 131.108.1.1 (not enabled on Ethernet0)
R2#
```

(2) 解决方法

如例 3-2 所示, 因为路由器 2 缺少 **network** 语句, 它会像例 3-4 中的调试输出那样忽略从 Ethernet 0 接口接收的 RIP 更新。如果配置的 **network** 语句不正确也会出现这种问题。以 C 类地址为例, 如果不配置 209.1.1.0, 而是配置 209.1.0.0, 因为 0 可以覆盖第三个八位位组的任何数字。RIP-1 是有类协议并使用有类 **network** 语句。如果配置了 **cidr** 语句, 那么 RIP 就不能正确工作。必须在配置中增加 **network** 语句来解决这个问题。

例 3-5 给出了路由器 2 的新配置以解决这个问题。

例 3-5 解决该问题的 R2 的新配置

```
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router rip
network 131.107.0.0
network 131.108.0.0
```

例 3-6 给出 R2 上 **show ip protocols** 的输出。该输出现在显示了网关信息。

例 3-6 show ip protocols 命令给出了 R1 的接口 IP 地址的网关设置

```
R2#show ip protocols

Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 12 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface        Send    Recv    Triggered    RIP    Key-chain
    Ethernet0        1      1      2
    Loopback0        1      1      2
```

(待续)

```

Automatic network summarization is in effect
Routing for Networks:
  131.108.0.0
Routing Information Sources:
  Gateway         Distance   Last Update
  131.108.1.1     120       00:00:09
Distance: (default is 120)

```

例 3-7 给出了 **show ip route** 的输出，显示出路由器 R2 在配置改变后学习 RIP 路由了。

例 3-7 **show ip route** 命令显示出解决问题后学习的路由

```

R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:11 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:11 ago, via Ethernet0
      Route metric is 1, traffic share count is 1

```

2. RIP 路由不在路由表中——原因：第 1/2 层故障

路由不在路由表中的另一个原因是第 1 层或第 2 层故障。如果第 1 层或第 2 层出现问题，并不是 RIP 的问题。下面是接口或线路协议出现问题时通常需要检查的项目列表：

- 没有插好线缆。
- 线缆松动。
- 线缆坏了。
- 收发器坏了。
- 端口坏了。
- 接口卡坏了。
- 在 WAN 链路情形中，电信运营商的第 2 层问题。
- 在背对背串行连接情形中缺少 **clock** 语句。

图 3-3 给出由这些原因引起的问题的解决方法流程图。

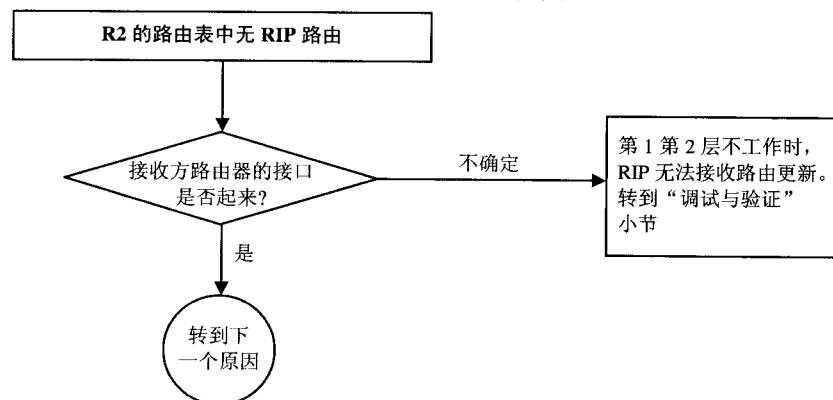


图 3-3 解决 RIP 路由不出现在路由表中问题的流程图

(1) 调试和验证

从例 3-8 可以看出以太网接口的线路协议出现问题, 也就是第 1 层或第 2 层的某些地方出现了问题。

例 3-8 show interface output 显示线路协议出现问题

```
R2#show interface ethernet 0
Ethernet0 is up, line protocol is down
Hardware is Lance, address is 0000.0c70.d41e (bia 0000.0c70.d41e)
Internet address is 131.108.1.2/24
```

调试

例 3-9 给出了 **debug ip rip** 的输出。在调试中, R2 没有发送或接收任何 RIP 更新, 因为第 2 层已不能正常工作。

例 3-9 debug ip rip 命令的输出显示没有发送任何内容

```
R2#debug ip rip
RIP protocol debugging is on
R2#
```

(2) 解决方法

RIP 在第 2 层之上运行。如果第 2 层不能工作, RIP 就不能发送或接收任何路由。

必须更正第 2 层的错误。有时, 问题可能很简单, 比如线缆松了; 问题也可能很复杂, 如硬件坏了, 此时必须更换硬件。

例 3-10 给出第 2 层更正之后 R2 上 **show interface Ethernet 0** 命令的输出。

例 3-10 更正第 1/2 层问题之后 Ethernet 0 接口开始工作

```
R2#show interface Ethernet0
Ethernet0 is up, line protocol is up
Hardware is Lance, address is 0000.0c70.d41e (bia 0000.0c70.d41e)
Internet address is 131.108.1.2/24
```

例 3-11 给出 **show ip route** 命令的输出, 表明解决第 1 或第 2 层问题后开始学到 RIP 路由。

例 3-11 更正第 1/2 层问题之后的路由表入口

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

3. RIP 路由不在路由表中——原因：分发列表阻塞路由

分发列表是路由更新的过滤机制。分发列表调用访问列表并检查是否允许网络被发布。如果访问列表不包含任何网络，路由更新会自动被拒绝。分发列表可以应用在进来的路由更新或发出的路由更新上。

本例中配置了 `distribute-list in`。因为访问列表不包含 131.108.0.0 的 `permit` 语句，所以 R2 不在路由表中安装这些路由。

图 3-4 给出了由于这个原因出现的问题的解决方法流程图。

(1) 调试和验证

例 3-12 给出了路由器 R2 的当前配置。在该配置中，`access-list 1` 允许网络 131.107.0.0 通过。但是每个访问列表的结尾有一个隐藏的 `deny` 命令，所以 131.108.0.0 同样将被拒绝，在该访问列表配置中，网络 131.108.0.0 不被允许，所以路由器没有安装 131.108.0.0 的任何子网。

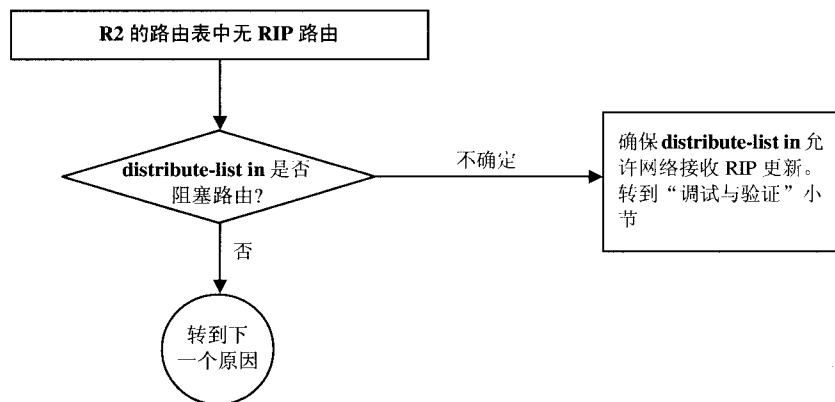


图 3-4 解决 RIP 路由不在路由表中的问题的流程图

例 3-12 R2 的配置显示网络 131.108.0.0 被 `access-list 1` 中隐藏的 `deny` 命令阻塞

```

interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router rip
network 131.108.0.0
distribute-list 1 in
!
access-list 1 permit 131.107.0.0 0.0.255.255
  
```

(2) 解决方法

使用分发列表时要检查两次访问列表以确定需要允许的网络确实在访问列表中被允许。例 3-12 的访问列表由于在每一个访问列表结尾隐藏 `deny` 命令，所以只允许 131.107.0.0 而拒绝所有其他的网络。在 `access-list 1` 中加入 `permit 131.108.0.0` 就可解决该问题。

例 3-13 给出路由器 R2 的新配置：访问列表中加入允许 131.108.0.0 通过的语句。

例 3-13 更正 R2 上的配置以解决该问题

```

interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!
router rip
network 131.108.0.0
distribute-list 1 in
!
access-list 1 permit 131.107.0.0 0.0.255.255
access-list 1 permit 131.108.0.0 0.0.255.255

```

例 3-14 显示出路由器 R2 在配置改变后学习 RIP 路由了。

例 3-14 R2 的路由表在更正后学习 RIP 路由了。

```

R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1

```

4. RIP 路由不在路由表中——原因：访问列表阻塞 RIP 源地址

访问列表可以基于源地址过滤信息流。扩展访问列表可以基于源和目的地址过滤信息流。为了过滤流入或流出的信息流，这些访问列表可以把接口级的命令应用在接口上：

```
ip access-group access-list number {in|out}
```

在 RIP 环境中使用访问列表时，必须肯定该访问列表不会阻塞 RIP 更新的源地址。本例中，R2 没有在路由表中安装 RIP 路由，因为 **access-list 1** 不允许源地址为 R1 的 RIP 更新通过。

图 3-5 给出了由这一原因引起的问题的解决方法流程图。

(1) 调试和验证

例 3-15 给出了路由表 R2 的当前配置。R2 中的访问列表不允许 RIP 更新的源地址 131.108.1.1 通过。图 3-1 中，131.108.1.1 是 R1 的 RIP 更新的源地址。因为每一个访问列表的最后有一个隐藏的 **deny**，所以 131.108.1.1 自动被拒绝了。

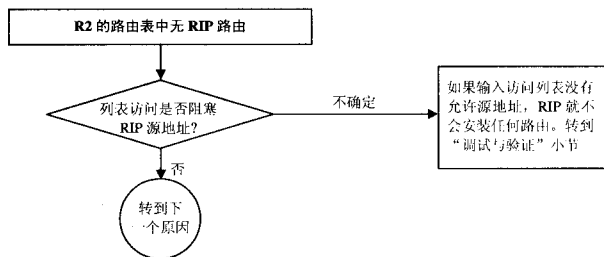


图 3-5 解决 RIP 路由不在路由表中的问题的流程图

例 3-15 access-list 1 不允许该源地址通过

```
R2#  
interface Loopback0  
ip address 131.108.3.2 255.255.255.0  
!  
interface Ethernet0  
ip address 131.108.1.2 255.255.255.0  
ip access-group 1 in  
!  
router rip  
network 131.108.0.0  
!  
access-list 1 permit 131.107.0.0 0.0.255.255
```

调试

例 3-16 中的 **debug ip rip** 的输出表明 RIP 只发送更新，不接收任何信息，原因是 R2 的输入访问列表不允许源地址 131.108.1.1 通过。

例 3-16 debug ip rip 输出表明 R2 不接收任何 RIP 更新

```
R2#debug ip rip  
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (131.108.1.2)  
RIP: build update entries  
      subnet 131.108.3.0 metric 1  
RIP: sending v1 update to 255.255.255.255 via Loopback0 (131.108.3.1)  
RIP: build update entries  
      subnet 131.108.1.0 metric 1  
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (131.108.1.2)  
RIP: build update entries  
      subnet 131.108.3.0 metric 1  
RIP: sending v1 update to 255.255.255.255 via Loopback0 (131.108.3.1)  
RIP: build update entries  
      subnet 131.108.1.0 metric 1  
R2#
```

(2) 解决方法

标准访问列表规定源地址。本例中，源地址是 131.108.1.1，即 R1 的发送方接口地址。R2 的标准访问列表不允许该源地址，所以 RIP 路由不会安装到 R2 的路由表中。为了解决该问题，应该在 access-list 1 中允许该源地址。

例 3-17 给出了解决该问题的新配置。

例 3-17 修改过的访问列表允许源地址通过

```
R2#  
interface Loopback0  
ip address 131.108.3.2 255.255.255.0  
!
```

(待续)

```

interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip access-group 1 in
!
router rip
network 131.108.0.0
!
access-list 1 permit 131.107.0.0 0.0.255.255
access-list 1 permit 131.108.1.1 0.0.0.0

```

如果 RIP 源地址在访问列表中禁止通过, 使用扩展访问列表时也会出现同样问题。上述解决方法仍然可以用于扩展访问列表。思想就是允许 RIP 更新的源地址通过。

例 3-18 给出了使用扩展访问列表时的配置。

例 3-18 使用扩展访问列表时的正确配置

```

R2#
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip access-group 100 in
!
router rip
network 131.108.0.0
!
access-list 100 permit ip 131.107.0.0 0.0.255.255 any
access-list 100 permit ip host 131.108.1.1 any

```

例 3-19 给出了路由器 R2 的路由表, 表明该路由器在配置变化后学到了 RIP 路由。

例 3-19 R2 在更正访问列表配置后接收到 RIP 路由

```

R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1

```

5. RIP 路由不在路由表中——原因: 访问列表阻塞 RIP 广播或组播 (在使用 RIP-2 时)

访问列表可以根据类型过滤分组。在输入接口上使用访问列表时, 必须确定它们不会阻塞 RIP 广播或 RIP-1 和 RIP-2 使用的 UDP 的 520 端口 (或使用 RIP-2 时的 RIP 组播地址)。

如果输入接口应用的访问列表中阻塞了这些地址, RIP 就不会在该接口学习到的路由表中安装任何路由。

图 3-6 给出了由于这个原因出现的问题的解决方法流程图。

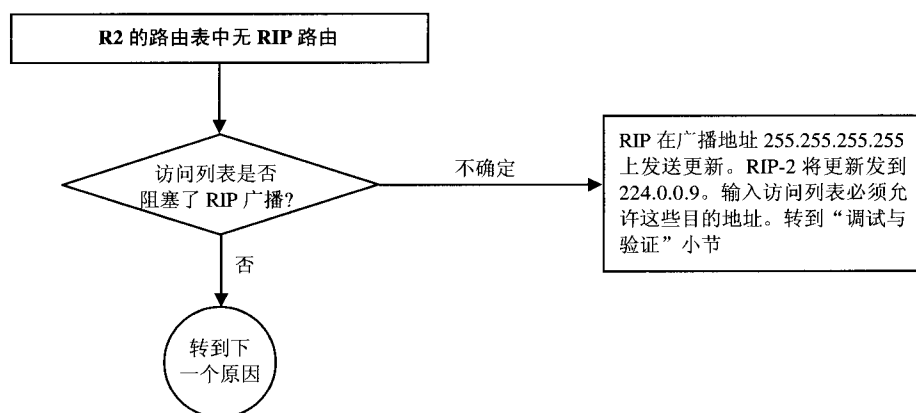


图 3-6 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

例 3-20 给出了路由表 R2 的当前配置。在该配置中，不允许 RIP 的目的地址为 255.255.255.255。这会导致 R2 的路由表中不能安装 RIP 路由。R1 发送的目的地址为 255.255.255.255 的 RIP 更新将被 R2 阻塞。

例 3-20 R2 的配置不允许 RIP-1 广播地址通过

```

R2#
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip access-group 100 in
!
router rip
network 131.108.0.0
!
access-list 100 permit ip 131.107.0.0 0.0.255.255 any
access-list 100 permit ip host 131.108.1.1 host 131.108.1.2
  
```

(2) 解决方法

RIP-1 向 255.255.255.255 广播路由更新。接收方路由器的输入访问列表中必须允许该地址通过，这样才能接收 RIP 更新。

例 3-21 给出了路由器 R2 的新配置。为了允许曾经被阻塞的 RIP 广播地址通过，必须修改 access-list 100。

例 3-21 配置路由器 R2 的输入访问列表以接收 RIP-1 广播通过

```

interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
  
```

(待续)

```
ip access-group 100 in
!
router rip
network 131.108.0.0
!
access-list 100 permit ip 131.107.0.0 0.0.255.255 any
access-list 100 permit ip host 131.108.1.1 host 131.108.1.2
access-list 100 permit ip host 131.108.1.1 host 255.255.255.255
```

在使用 RIP-2 时要有少许改动。如例 3-22 所示, 必须用组播地址代替广播地址。

例 3-22 配置路由器 R2 的输入访问列表以接收 RIP-2 组播

```
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip access-group 100 in
!
router rip
version 2
network 131.108.0.0
!
access-list 100 permit ip 131.107.0.0 0.0.255.255 any
access-list 100 permit ip host 131.108.1.1 host 131.108.1.2
access-list 100 permit ip host 131.108.1.1 host 224.0.0.9
```

例 3-23 给出了更正问题后路由器 R2 的路由表。

例 3-23 在更正访问列表配置后 R2 的路由表, 表明可以学习 RIP 路由了

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

6. RIP 路由不在路由表中——原因: RIP 版本类型不兼容

路由器上配置的 RIP 缺省版本为版本 1, 也就是说, 所有的接口只发送和接收 RIP-1 的分组。要允许 RIP-2, 必须在 router rip 之后加入 version 2 项。如果运行 RIP-1 的路由器接收到运行 RIP-2 的路由器发送的更新, 它就会忽略这些更新, 不会将新路由安装到路由表中。接收 RIP-2 分组的路由器的接口配置为允许接收 RIP-2 更新。

图 3-7 给出了由这一原因引起的问题的解决方法流程图。

```

Sending updates every 30 seconds, next due in 9 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: rip
Default version control: send version 1, receive version 1
Interface    Send  Recv Key-chain
Ethernet0    1     1
Loopback0    1     1

Routing for Networks:
  131.108.0.0

Routing Information Sources:
  Gateway      Distance    Last Update
  131.108.1.1      120      00:01:34
Distance: (default is 120)
R2#

```

例 3-27 给出了 R1 的配置。发送方 R1 被配置为发送版本 2 的分组。命令 `version 2` 使路由器只发送和接收 RIP-2 分组。

例 3-27 R1 的配置表明它可以允许 RIP-2 分组

```

R1#
router rip
  version 2
  network 131.108.0.0

```

例 3-28 显示了 `show ip protocols` 命令的输出，表明发送方 R1 只发送和接收版本 2 的分组。因为 `router rip` 之后加入了 `version 2` 命令。

例 3-28 `show ip protocols` 命令输出表示 R1 只发送和接收 RIP-2 分组。

```

R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 13 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
  Interface    Send  Recv  Key-chain
  Ethernet0/1  2     2
  Loopback1    2     2

Routing for Networks:
  131.108.0.0

Routing Information Sources:
  Gateway      Distance    Last Update
  131.108.1.2      120      00:04:09
Distance: (default is 120)

```

(2) 解决方法

如果将接收方 R2 配置为只接收 RIP-1 分组, R2 会忽略 RIP-2 的更新。必须在发送方将路由器 R1 配置为发送版本 1 和版本 2 的分组。当 R2 接收到版本 1 的分组时,它会在路由表中安装路由。R2 忽略 RIP-2 分组,因为它配置为允许 RIP-1。

例 3-29 给出 R1 的新配置。在该配置中,发送方(R1 的以太网接口)被配置为发送和接收 RIP-1 和 RIP-2 的分组。

例 3-29 R1 的新配置可发送和接收版本 1 和版本 2 的分组

```
R1#
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
ip rip send version 1 2
ip rip receive version 1 2
!
router rip
version 2
network 131.108.0.0
```

例 3-30 给出了 show ip protocols 的输出,表明接口 Ethernet 0 发送和接收版本 1 和版本 2 的分组。发送版本 1 和版本 2 更新的好处是在以太网网段的任何一台只运行 RIP-1 或只运行 RIP-2 的设备都可以在以太网上与 R1 通信。

例 3-30 show ip protocols 命令输出表明 R1 的接口 Ethernet 0 发送和接收 RIP-1 和 RIP-2 分组

```
R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 4 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface  Send  Recv  Key-chain
    Ethernet0   1 2   1 2
    Loopback0   2    2
  Routing for Networks:
    131.108.0.0
  Routing Information Sources:
    Gateway     Distance    Last Update
    131.108.1.2    120        00:00:07
  Distance: (default is 120)
R1#
```

例 3-31 给出了配置改变后路由器 R2 的路由表。

例 3-31 R1 配置为发送和接收 RIP-1 和 RIP-2 后的 R2 的路由表

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

7. RIP 路由不在路由表中——原因：认证字段不匹配 (RIP-2)

RIP-2 的选项之一是可以为提高安全性而为 RIP-2 更新增加认证。使用认证时，双方都要配置口令。口令也称为认证密钥。如果该密钥与另一端的口令密钥不匹配，双方都会忽略 RIP-2 更新。

图 3-8 给出了由这一原因引起的问题的解决方法流程图。

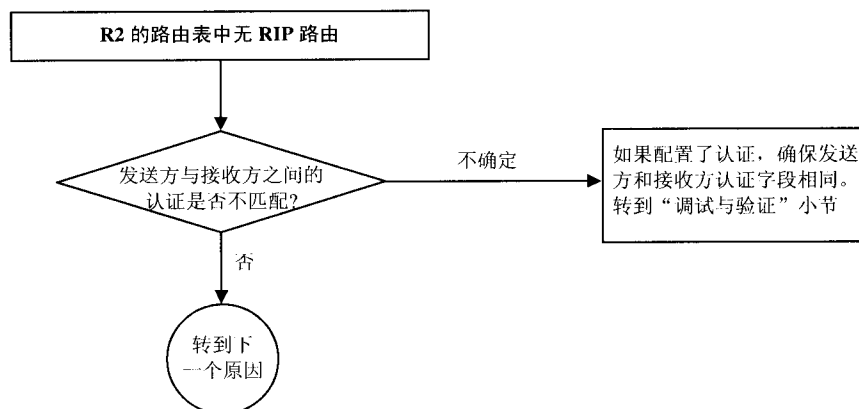


图 3-8 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

例 3-32 给出出现问题时路由器 R1 和 R2 的配置。在该配置中，R1 和 R2 的 RIP 认证密钥不同。R2 的以太网接口将该密钥配置为 cisco1，而 R1 将其配置为 Cisco。这两个密钥不匹配，所以它们忽略对方的更新，路由也不能安装到路由表中。

例 3-32 R1 和 R2 的配置显示双方配置的认证密钥不同

```
R2#
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip rip authentication key-chain cisco1
!
```

(待续)

```
router rip
version 2
network 131.108.0.0
!
R1#
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
ip rip authentication key-chain cisco
!
router rip
version 2
network 131.108.0.0
!
```

例 3-33 显示了 R2 上 **debug ip rip** 命令的输出, 表明 R2 从 R1 收到一个具有错误认证密钥的 RIP 分组。这意味着发送方和接收方的认证密钥不匹配。

例 3-33 debug ip rip 命令的输出显示 R2 收到的 RIP-2 分组未通过认证

```
R2#debug ip rip
RIP protocol debugging is on
RIP: ignored v2 packet from 131.108.1.1 (invalid authentication)
```

(2) 解决方法

RIP 使用认证时, 必须确定发送方和接收方配置了相同的认证密钥。有时, 在密钥的最后加入一个空格也会导致无法通过认证的问题, 因为空格也会被看作一个字符。最后, 这会导致查看配置也无法解决的问题。

从调试可以看出认证密钥存在的问题。要解决该问题, 需在发送方和接收方配置相同的密钥, 或是重新输入认证密钥, 确保结尾处没有空格。

例 3-34 给出了解决问题的新配置。在路由器 R2 上重新配置与路由器 R1 相匹配的认证密钥。

例 3-34 用正确的认证密钥配置 R2

```
R2#
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
ip rip authentication key-chain cisco
!
router rip
version 2
network 131.108.0.0
!
```

例 3-35 给出了配置改变后路由器 R2 的路由表。

例 3-35 在 R2 上重新配置认证密钥后 R2 的路由表

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

8. RIP 路由不在路由表中——原因：非连续网络

一个主网络被另一个主网络从中间分开的现象叫做非连续网络。第 2 章“理解 RIP 协议”为 RIP 不支持非连续网络提供了详细的解释。

图 3-9 给出了一个主网络被另一个主网络分开时存在的非连续网络。

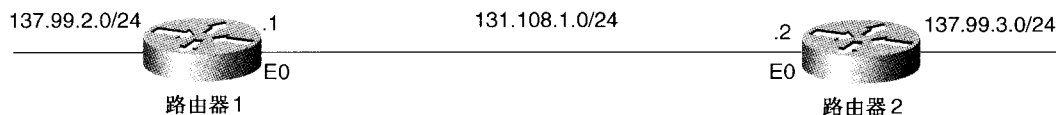


图 3-9 非连续网络举例

图 3-10 给出了由这一原因引起的问题的解决方法流程图。

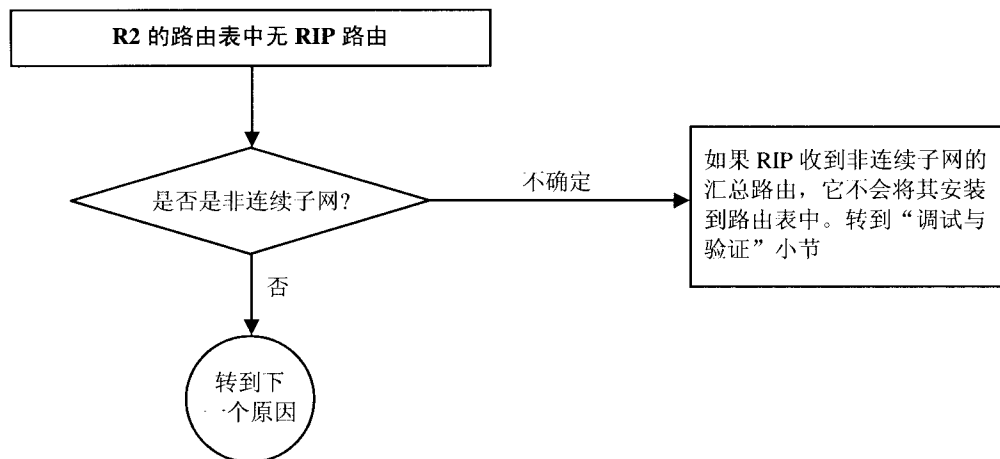


图 3-10 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

例 3-36 给出了路由器 R1 和 R2 的配置。路由器 R1 和 R2 的以太网接口上使用 **network** 命令启用了 RIP。

例 3-36 非连续网络环境中 R1 和 R2 的配置

```
R2#
interface Loopback0
```

(待续)

```
ip address 137.99.3.2 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.2 255.255.255.0
!

router rip
network 131.108.0.0
network 137.99.0.0
!
```

```
R1#
interface Loopback0
ip address 137.99.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router rip
network 131.108.0.0
network 137.99.0.0
!
```

例 3-37 显示了路由器 R1 和 R2 上 **debug ip rip** 命令的输出。两者输出都表明发送了网络 137.99.0.0 的信息。

例 3-37 debug ip rip 命令的输出表明两个路由器都发送了汇总的主网络地址

```
R2#debug ip rip
RIP protocol debugging is on
RIP: received v1 update from 131.108.1.1 on Ethernet0
    137.99.0.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (131.108.1.2)
RIP: build update entries
    network 137.99.0.0 metric 1
R2#

R1#debug ip rip
RIP protocol debugging is on
R1#
RIP: received v1 update from 131.108.1.2 on Ethernet0
    137.99.0.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (131.108.1.1)
RIP: build update entries
    network 137.99.0.0 metric 1
```

因此，两个路由器都忽略从对方发来的 137.99.0.0 更新。因为 R1 和 R2 已与主网络相连，它们会忽略更新。

(2) 解决方法

RIP 没有在路由表中安装路由 137.99.0.0，因为如前所述，RIP 不支持非连续网络。该问

题存在几种解决方法。最快的方法是在每个路由器上为子网 137.99.0.0 配置静态路由。第二种方法是启用 RIP-2。还有一种方法是用另一种支持非连续网络的 IP 路由协议取代 RIP，如 OSPF、IS-IS、EIGRP 等等。

例 3-38 给出了在路由器 R1 和 R2 上解决问题的配置变化。该配置将静态路由加入到非连续子网中。因为非连续网络中不能传递 RIP-1 的子网信息，唯一的解决方法是用静态路由进行修正。

例 3-38 配置静态路由可以解决该问题

```
R1#  
interface Loopback0  
ip address 137.99.2.1 255.255.255.0  
!  
interface Ethernet0  
ip address 131.108.1.1 255.255.255.0  
!  
router rip  
network 131.108.0.0  
network 137.99.0.0  
!  
ip route 137.99.3.0 255.255.255.0 131.108.1.2
```

```
R2#  
interface Loopback0  
ip address 137.99.3.2 255.255.255.0  
!  
interface Ethernet0  
ip address 131.108.1.2 255.255.255.0  
!  
router rip  
network 131.108.0.0  
network 137.99.0.0  
!  
ip route 137.99.2.0 255.255.255.0 131.108.1.1
```

例 3-39 给出了使用 RIP-2 解决该问题的替代方法。该解决方法是运行配置为 no auto-summary 的 RIP-2。增加了 no auto-summary 命令后，RIP-2 穿过主网络边界时不会进行汇总。特定的子网信息仍然被传送。

例 3-39 在非连续网络环境中 RIP-2 的配置

```
router rip  
version 2  
network 131.108.0.0  
network 137.99.0.0  
no auto-summary
```

例 3-40 给出了解决该问题后 R2 的路由表。

例 3-40 R2 的路由表表明在配置 no auto-summary 之后通过 RIP-2 学习到了 137.99.2.0/24

```
R2#show ip route 137.99.2.0
Routing entry for 137.99.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

9. RIP 路由不在路由表中——原因：无效源

RIP 在路由表中安装路由时会进行源地址的有效性检查。如果源地址与本地接口不在同一子网中，RIP 将忽略该更新，并且不把从这个源地址传送来的路由安装到路由表中。

图 3-11 给出了产生无效源问题的网络图。

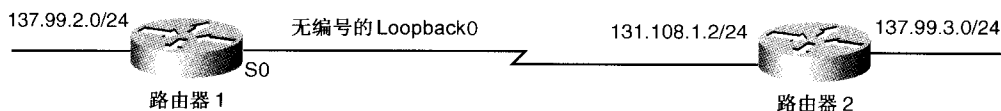


图 3-11 无效路由源的网络图

在图 3-11 中，路由器 1 的 Serial 0 接口是无编号的 Loopback 0。路由器 2 的串口是有编号的。当路由器 2 从路由器 1 收到 RIP 更新时，它会认为源无效，因为源地址与路由器 2 的 Serial 0 接口不在同一子网。

图 3-12 给出了由这一原因引起的问题的解决方法流程图。

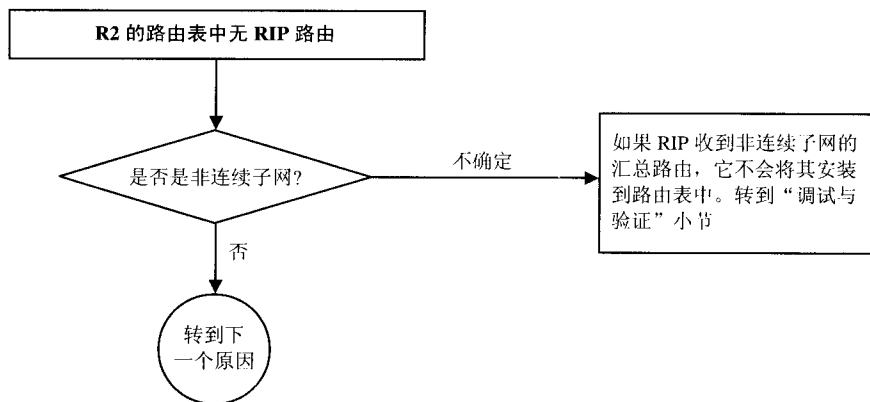


图 3-12 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

例 3-41 给出了路由器 R1 和 R2 的配置。这样配置时，路由器 R1 的 Serial 0 接口是无编号的 Loopback 0。而 R2 的 Serial 0 接口是有编号的。

例 3-41 R1 和 R2 的配置表明 R1 的 Serial 0 接口是无编号的，而 R2 接口是有编号的

```
R2#
interface Loopback0
```

(待续)

```
ip address 131.108.3.2 255.255.255.0
!
interface Serial0
ip address 131.108.1.2 255.255.255.0
!
router rip
network 131.108.0.0
!
```

```
R1#
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Serial0
ip unnumbered Loopback0
!
router rip
network 131.108.0.0
!
```

例 3-42 中 **debug ip rip** 命令的输出表明, R2 因为源的有效性检查忽略了从 R1 发送来的 RIP 更新。因为从 R1 发来的 RIP 更新不在同一子网, 所以 R2 不会将该路由安装到路由表中。

例 3-42 debug ip rip 消息表明 R2 正在接收不同于自己的端口的源地址发送来的更新

```
R2#debug ip rip
RIP protocol debugging is on
RIP: ignored v1 update from bad source 131.108.2.1 on Serial0
R2#
```

(2) 解决方法

如果一边是有编号的, 另一边是无编号的, 那么应该关闭该检查。在远程拨号访问一个路由器时经常出现这种情况。访问路由器的拨号接口是无编号的, 而且所有的远程路由器都是把 IP 地址分配给它们的拨号接口。

例 3-43 给出了路由器 R2 的新配置以解决该问题。

例 3-43 R2 的配置以关闭源有效性检查

```
R2#
interface Loopback0
ip address 131.108.3.2 255.255.255.0
!
interface Serial0
ip address 131.108.1.2 255.255.255.0
!
router rip
no validate-update-source
network 131.108.0.0
!
```

例 3-44 说明了在改变 R2 的配置后，路由已安装到路由表中。

例 3-44 关闭源有效性检查后 R2 的路由表

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 00:00:01 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago
      Route metric is 1, traffic share count is 1
```

10. RIP 路由不在路由表中——原因：第 2 层问题（交换机、帧中继、其他第 2 层介质）

第 2 层问题有时候会破坏组播或广播能力，然后进一步影响第 3 层的组播，这样 RIP 就不能正常工作了。第 3 层组播或广播会转换成第 2 层的广播或组播。如果第 2 层处理该层组播或广播时存在问题，RIP 更新就不能传送出去。调试命令显示从某个源产生的广播或组播没有发送出去。

图 3-13 给出了运行 RIP 时出现帧中继问题的网络图。在图 3-13 中，路由器 1 和路由器 2 通过第 2 层介质（例如，帧中继、X.25、以太网、FDDI 等）相连。

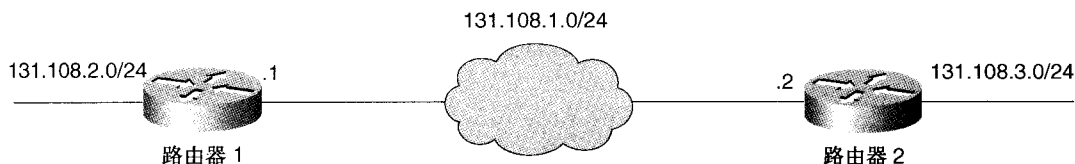


图 3-13 帧中继环境中运行 RIP 的两个路由器

图 3-14 给出了由这一原因引起的问题的解决方法流程图。

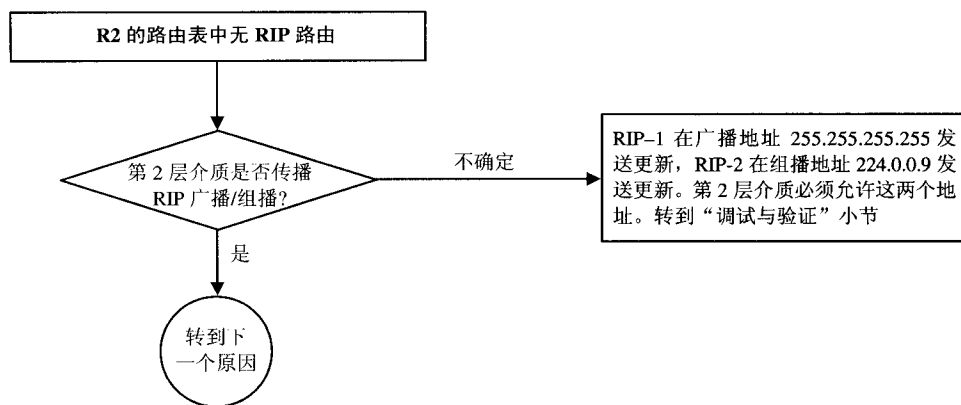


图 3-14 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

例 3-45 给出了 debug ip rip 命令的输出，表明 R1 发送和接收 RIP 更新都没有问题。R2 可以发送 RIP 更新但没有被收到。这表示 RIP 更新在第 2 层丢失。

例 3-45 access-list 100 的 debug ip packet 命令表明 R1 在网络上发送了 RIP 更新，但是 R2 没有收到

```
R1#debug ip packet 100 detail
IP packet debugging is on (detailed) for access list 100
R1#
IP: s=131.108.1.1 (Ethernet0), d=255.255.255.255, len 132, sending broadcast/
    multicast
    UDP src=520, dst=520
IP: s=131.108.1.1 (Ethernet0), d=255.255.255.255, len 132, rcvd 2
    UDP src=520, dst=520
```

```
R2#debug ip packet 100 detail
IP packet debugging is on (detailed) for access list 100
R2#
IP: s=131.108.1.2 (Ethernet0), d=255.255.255.255, len 132, sending broadcast/
    multicast
    UDP src=520, dst=520
IP: s=131.108.1.2 (Ethernet0), d=255.255.255.255, len 132, sending broadcast/
    multicast
    UDP src=520, dst=520
```

例 3-46 中，在 debug 中使用 access-list 100 来查看 RIP 的广播或组播。

例 3-46 在 debug 中使用 access-list 100 最小化信息流

```
access-list 100 permit ip any host 255.255.255.255
access-list 100 permit ip any host 224.0.0.9
```

例 3-47 给出了查找运行 RIP-2 时出现的问题的方法。Ping 组播地址 224.0.0.9——如果邻居没有回答，那么第 2 层的组播地址肯定出现了问题。

例 3-47 组播 ping 失败意味着在第 2 层丢失 R2 的组播

```
R2#ping 224.0.0.9
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.0.0.9, timeout is 2 seconds:
.....
R2#
```

(2) 解决方法

RIP-1 在广播地址 255.255.255.255 发送更新。使用 RIP-2 时，在组播地址 224.0.0.9 发送更新。如果这两个地址在第 2 层被阻塞或不能够在第 2 层传送，RIP 就不能正确工作。第 2 层可以是一个简单的以太网交换设备、帧中继网络、桥网络等。解决第 2 层问题的方法超出了本书范围。

例 3-48 给出了解决第 2 层的问题后 RIP 路由安装到了路由表中。

例 3-48 解决第 2 层问题后 R2 将路由安装到路由表中

```
R2#show ip route 131.108.2.0
```

(待续)

```

Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 00:00:01 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago
      Route metric is 1, traffic share count is 1
  
```

11. RIP 路由不在路由表中——原因：偏移量列表定义的度量值过大

偏移量列表可以增加流入或流出的 RIP 更新的度量值。使用偏移量列表可以直接影响路由表。该表可以应用在访问列表定义的网络中。如果偏移量值很大，比如 14 或 15，那么 RIP 度量值就会在通过两个路由器之后达到无穷大。这就是偏移量列表值必须保持最小的原因。

图 3-15 给出了由于错误配置偏移量列表而出现问题的网络设置。

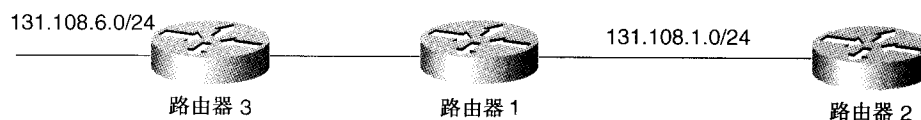


图 3-15 错误配置偏移量列表而出现问题的网络拓扑

例 3-49 说明路由器 131.108.6.0 不在 R2 的路由表中。

例 3-49 R2 的路由表缺少 R3 的子网

```

R2#show ip route 131.108.6.0
% Subnet not in table
  
```

图 3-16 给出了由这一原因引起的问题的解决方法流程图。

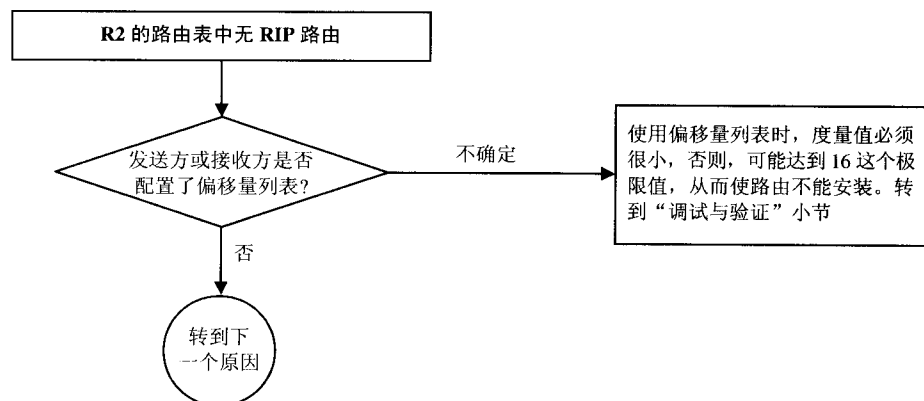


图 3-16 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

只有通过检查 RIP 的一般行为才能解决该问题。

例 3-50 表明 R2 接收到除 131.108.6.0/24 之外的其他 RIP 路由。

例 3-50 R2 的路由表中没有 131.108.6.0/24

```
R2#show ip route RIP

      131.108.0.0/24 is subnetted, 4 subnets
R       131.108.5.0 [120/1] via 131.108.1.1, 00:00:06, Ethernet1
R       131.108.3.0 [120/1] via 131.108.1.1, 00:00:06, Ethernet1
```

这表明问题出在 131.108.6.0/24，而通常不是 RIP。原因是 R3 接收 R1 发送来的其他 RIP 路由，所以从 R1 发来的 RIP 更新很正常。

例 3-51 给出 R1 的路由表，131.108.6.0/24 在路由表中。

例 3-51 R1 在路由表中看到 131.108.6.0/24

```
R1#show ip route 131.108.6.0
Routing entry for 131.108.6.0/24
  Known via "rip", distance 120, metric 1
```

那么为什么 R2 不能安装 131.108.6.0/24 呢？可能有以下几种原因：

- R1 没有向 R2 公布。
- R1 公布了，但 R2 没有收到。
- R2 收到了，但因为度量太大而将其丢弃。

最简单的解决方法是快速配置检查。

例 3-52 给出了路由器 R1 的配置

例 3-52 R1 为 131.108.6.0/24 配置偏移量列表的值过大

```
R1#
router rip
version 2
offset-list 1 out 15 Ethernet0/1
network 131.108.0.0
!
access-list 1 permit 131.108.6.0
```

管理员用很大的度量值配置了偏移量列表。该偏移量列表可以改变 RIP 更新的度量值。

从配置可以猜出通过 access-list 1 的所有更新的度量值会增加 15。例 3-52 中，access-list 1 允许 131.108.6.0。这意味着 131.108.6.0 的度量值是 16，这对 RIP 来说意味着不可到达。接收到更新以后，R2 会拒绝它。

如例 3-53 那样运行 debug ip rip 命令证实这一点。

例 3-53 R2 上 debug ip rip 命令表明接收到的 131.108.6.0 度量值无穷大

```
R2#debug ip RIP

RIP: received v2 update from 131.108.1.1 on Ethernet1
      131.108.6.0/24 -> 0.0.0.0 in 16 hops (inaccessible)
```

因为 16 在 RIP 中代表无穷大，所以 R2 拒绝 131.108.6.0/24 进入路由表。

(2) 解决方法

偏移量列表一般不用于 RIP 网络中。偏移量列表用于网络中存在多条等跳数冗余路径且管理员希望某一条路径优越于其他路径的情况。

例如, 假设 R1 和 R2 之间存在两条链路。一条链路可能会被拥塞和延迟。

管理员可能需要在短时间内将到达某个目的子网的 IP 信息流切换到不拥塞的链路, 以得到更好的吞吐量和减少拥塞。偏移量列表可以通过增大通往拥塞子网的接口上的 RIP 度量值很容易地完成这个任务。

例 3-54 给出了路由器 R1 的新配置。

要解决这个问题, 可配置偏移量值使跳数不会达到最大值。

例 3-54 使用正确的偏移量列表值配置 R1

```
R1#  
router rip  
version 2  
offset-list 1 out 1 Ethernet0/1  
network 131.108.0.0  
!  
access-list 1 permit 131.108.6.0
```

例 3-55 给出了解决问题后路由器 R2 的路由表

例 3-55 更正偏移量列表配置后 R2 的路由表包含了 131.108.6.0/24 的入口

```
R2#show ip route 131.108.6.0  
Routing entry for 131.108.6.0/24  
Known via "rip", distance 120, metric 1
```

12. RIP 路由不在路由表中——原因: 路由达到 RIP 跳数限制

RIP 度量值最大为 15 跳。如果网络大于 15 跳, 那么 RIP 就不适用。

图 3-17 给出了产生 RIP 跳数限制问题的网络设置。

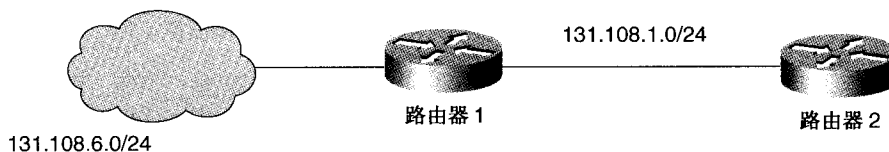


图 3-17 产生 RIP 跳数限制问题的网络设置

R2 接收 RIP 路由更新, 但由于路由的跳数很大 (超过 15), R2 没有将路由安装到路由表中, 如例 3-56 所示。

例 3-56 R2 的路由表不含路由 131.108.6.0

```
R2#show ip route 131.108.6.0  
% Subnet not in table
```

图 3-18 给出了由这一原因引起的问题的解决方法流程图。

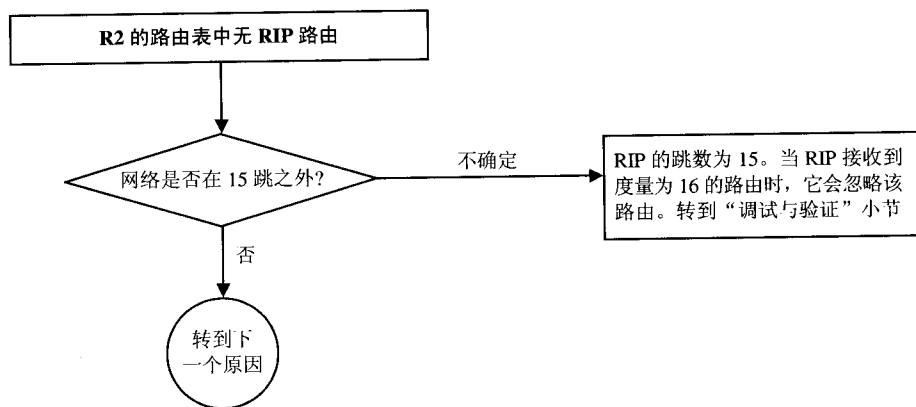


图 3-18 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

诊断该问题最有逻辑性的方法是检查 R1，看它是否收到 131.108.6.0/24。

例 3-57 表明路由器 R1 接收到 131.108.6.0/24 的 RIP 路由。

例 3-57 R1 的路由表中有 131.108.6.0/24，度量值为 15（最大的 RIP 度量值）

```

R1#show ip route 131.108.6.0
Routing entry for 131.108.6.0/24
Known via "rip", distance 120, metric 15
  
```

R1 此时收到路由，但是度量值是 15。R1 在 15 的基础上增加 1，然后公布到 R2，产生了无穷大度量值，要在 R1 中证明这一点，你可以运行 `debug ip rip` 命令实时查看该过程。从而阻止路由安装到路由表中。

例 3-58 给出了路由器 R1 上 `debug ip rip` 命令的输出。

例 3-58 `debug ip rip` 命令的输出表明 R1 将 131.108.6.0/24 以度量值 16 进行公布

```

R1#debug ip rip
RIP protocol debugging is on
RIP: sending v2 update to 224.0.0.9 via Ethernet1 (131.108.1.1)
131.108.6.0/24 -> 0.0.0.0, metric 16, tag 0
  
```

例 3-59 给出了在路由器 R2 上运行 `debug ip rip` 的输出结果。路由器 R2 接收该更新但是因为度量值表明网络非常远、不可达而丢弃该更新。

例 3-59 R2 上的 `debug ip rip` 命令输出表明 R2 接收到度量值为无穷大的路由

```

R2#debug ip rip
RIP protocol debugging is on
RIP: received v2 update from 131.108.1.1 on Ethernet1
131.108.6.0/24 -> 0.0.0.0 in 16 hops (inaccessible)
  
```

(2) 解决方法

这是路由经过超过 15 个设备之后经常出现的问题。现在的 IP 网络一般都超过 15 个路由

器。除了另选一种没有 15 跳限制的协议之外没有别的方法能够解决这个问题，可以使用 OSPF、EIGRP 或 IS-IS 等协议。

3.2.2 问题：RIP 没有安装所有可能的等价路径——原因：命令 `maximum-path` 限制 RIP 安装多条路径

缺省情况下，Cisco 路由器为了负载均衡只支持四条等价路径。命令 `maximum-path` 可以将等价路径数增大到六条。如果没有正确地配置命令，即使已经存在多条路径，命令 `maximum-path` 也将只允许一条路径到达目的地。不仅仅需要负载均衡时应当配置命令 `maximum-path 1`。

图 3-19 和例 3-60 提供了由于 `maximum-path` 命令限制 RIP 安装多条路径造成的缺少所有可能的等价路径的网络场景及其解决方法。本节会详细介绍如何解决这一问题。

图 3-19 给出了产生 RIP 不能安装所有等价路径问题的网络设置。

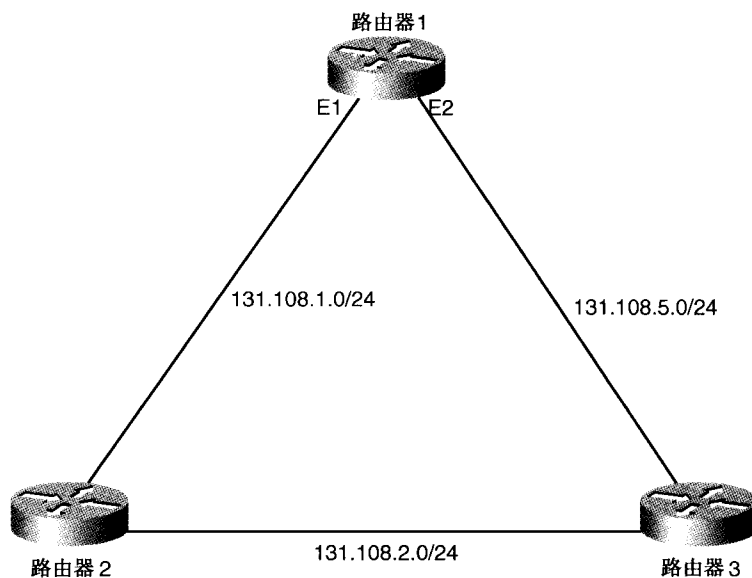


图 3-19 易产生等价路径问题的 RIP 网络

例 3-60 给出了路由器 R1 的路由表，路由表中只安装了一条路由。缺省时，所有的路由协议都支持多条等价路径（负载均衡）。如果存在多条等价路径，它们都需要安装到路由表中。

例 3-60 R1 只为 131.108.2.0/24 安装了一条路径

```
R1#show ip route rip
131.108.0.0/24 is subnetted, 1 subnets
R 131.108.2.0 [120/1] via 131.108.5.3, 00:00:09, Ethernet2
```

图 3-20 给出了解决由于该原因产生的问题的解决方法流程图。

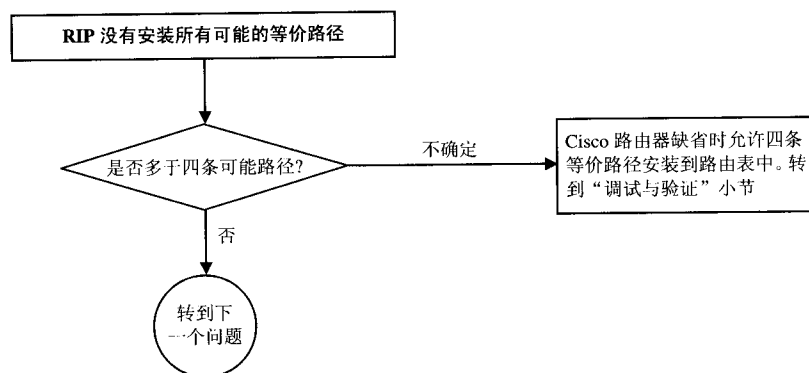


图 3-20 解决 RIP 路由不在路由表中的问题的流程图

(1) 调试和验证

例 3-61 给出了路由表 R1 上 **debug ip rip** 命令的输出。该输出表明路由器 R1 接收到两条等价路由。

例 3-61 R1 上 **debug ip rip** 的输出表明 R1 接收到两条关于网络 131.108.2.0 的更新

```

R1#debug ip rip
RIP protocol debugging is on
R1#
RIP: received v2 update from 131.108.5.3 on Ethernet2
131.108.2.0/24 -> 0.0.0.0 in 1 hops
RIP: received v2 update from 131.108.1.2 on Ethernet1
131.108.2.0/24 -> 0.0.0.0 in 1 hops
  
```

路由表中只安装了一条路由。你会看到由于配置了 **maximum-paths 1**，路由表中只有一条路由，而不是两条。

例 3-62 给出了路由器 R1 的当前配置。

例 3-62 用 **maximum-paths 1** 配置 R1

```

R1#
router rip
version 2
network 131.108.0.0
maximum-paths 1
  
```

(2) 解决方法

缺省情况下，Cisco IOS 允许在路由表中最多安装四条等价路由。如果按照例 3-63 的配置，可以增加到六条。

例 3-63 给出了在路由表中安装六条等价路由的配置。

例 3-63 在路由表中最多允许六条路径

```

R1#
router rip
maximum-paths 6
  
```

本例适用于多于四条路径，但只有四条路径安装到路由表中的情况。因为缺省时可以有四条等价路径，所以需要增加 **maximum-paths** 到允许五条甚至六条路由。

3.3 RIP 路由公布疑难解析

到目前为止讨论的所有问题都仅限于接收端或中间（第2层）的故障。

路由表不能安装路由还存在第三种可能。发送方发送 RIP 更新可能因为某些原因而存在故障。因此，接收方不能将 RIP 路由安装到路由表中。本节讨论发送方可能存在的问题。

本节讨论阻止 RIP 路由公布的几种网络情况。有些原因与路由器安装问题——如，缺少 **network** 语句或接口停止工作相重叠。本节的前提是已经排除了“疑难解析 RIP 路由安装问题”一节中讲到的问题之后依然存在的问题。下面介绍如何继续解决问题。

发送方最容易出现的与 RIP 路由公布相关的两个问题是：

- 发送方没有公布 RIP 路由。
- 丢失子网路由。

3.3.1 问题：发送方没有公布 RIP 路由

一般运行 RIP 的 IP 网络内部的路由器之间的路由表视图是一致的。也就是说，所有的路由器的路由表都包含到达网络的所有 IP 子网的全部可达性信息。这与在某个路由器上过滤某个子网不同。理想状态下，所有的 RIP 路由器都拥有整个网络的路由。

如果某个路由器的路由信息与另一个路由器的不同，有两种可能的原因：

- 有些路由器没有公布 RIP 路由。
 - 有些路由器没有接收 RIP 路由。
- 本节处理发送 RIP 更新时出现的问题。
- 图 3-21 给出了解决由于下列原因而引起发送方没有公布 RIP 路由的一种网络情况：
- 缺少或不正确的 **network** 语句。
 - 输出接口不工作。
 - **distribute-list out** 阻塞路由。
 - 公布的网络接口不能工作。
 - 定义的输出接口是被动接口。
 - 组播能力出现问题（帧中继封装失败）。
 - 错误配置 **neighbor** 语句。
 - 公布的子网是 VLSM。
 - 使用水平分割。

图 3-21 给出了路由器 R1 没有向 R2 发送 RIP 路由的网络设置。

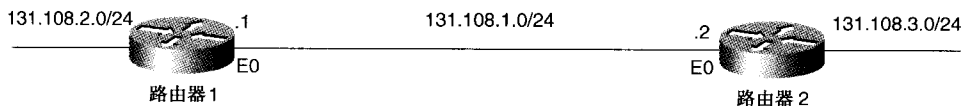


图 3-21 路由器 R1 没有向 R2 发送 RIP 路由的网络设置

1. 发送方没有公布 RIP 路由——原因: network 语句丢失或错误

在路由器接口启动 RIP 必须在命令 **router rip** 之后加入 **network** 语句。**network** 语句决定哪个接口应该启动 RIP。如果错误地配置或没有配置 **network** 语句, RIP 就不会运行在那个接口上, 从而 RIP 路由也不能从那个接口公布出去。

图 3-22 给出了问题的解决方法流程图。

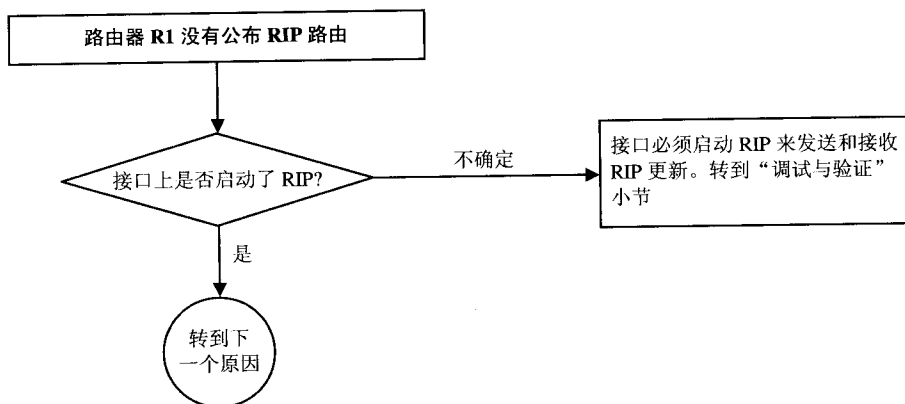


图 3-22 解决发送方不公布 RIP 路由的问题的流程图

(1) 调试和验证

例 3-64 给出了路由器 R1 的当前配置。

例 3-64 R1 的配置表明错误地配置了 network 语句

```

R1#
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router rip
network 131.107.0.0
  
```

例 3-64 中, **router rip** 命令之后错误地配置了 **network** 语句。应该配置 131.108.0.0, 可是实际配置的是 131.107.0.0。这样不能在接口上启用 RIP, 也就不可能发送任何更新。

(2) 解决方法

router rip 命令之后有时也会配置一些无类语句, 假设它会包括整个网络——例如:

```

router rip
network 131.0.0.0
  
```

network 语句不会通过 131.255.255.255 而包括 131.0.0.0, 因为 131.0.0.0 是无类网络而 RIP 是有类协议。同理, 如果有多个有类 C 类地址, 不可能使用一个 **network** 语句就可以包含所有的地址。例如, 假设通过 200.1.4.0 得到了 200.1.1.0。这并不说明可以使用下列命令语法:

```

router rip
  
```

```
network 200.1.0.0
```

这里的 **network** 语句对 RIP-1 来说是没有任何意义的, 因为 RIP-1 是有类协议。在 RIP 中公布所有 4 个网络的正确方法是:

```
router rip
network 200.1.1.0
network 200.1.2.0
network 200.1.3.0
network 200.1.4.0
```

例 3-65 给出了 R1 的正确配置。

例 3-65 纠正 R1 配置中的 network 语句

```
R1#
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router rip
network 131.108.0.0
```

例 3-66 给出了路由器 R2 的路由表, 表明已学习到 RIP 路由。

例 3-66 R2 的路由表表明纠正 network 语句之后学到了 RIP 路由

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:11 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:11 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

2. 发送方没有公布 RIP 路由——原因: 输出接口不工作

RIP 协议在第 3 层工作。RIP 不能通过不工作的接口发送更新。接口不工作有以下几种可能:

- 接口本身正常, 线路协议不能工作。
- 接口不能工作, 线路协议也不能工作。
- 接口被配置为不工作, 线路协议不能工作。

如果输出接口表现出以上任何症状, RIP 都不能通过网络发送更新。在这里需要着重说明, 在所有这些可能的原因中, 线路协议都是不能工作的。这是检查第 2 层连接状态的重要信息。

图 3-23 给出了发送方不能公布 RIP 路由的解决方法流程图。

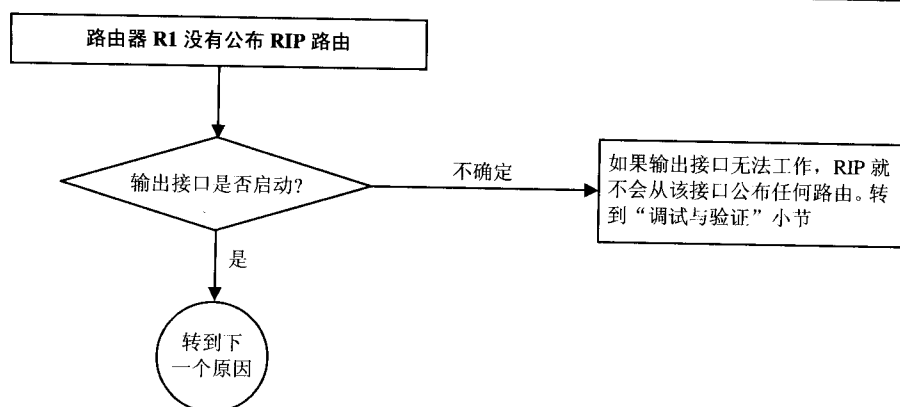


图 3-23 解决发送方不能公布 RIP 路由的问题的流程图

(1) 调试和验证

例 3-67 表明接口 Ethernet 0 不能工作。

例 3-67 R1 的输出接口 Ethernet 0 表明线路协议不能工作

```

R1#show interface ethernet 0
Ethernet0 is up, line protocol is down
Hardware is Lance, address is 0000.0c70.d31e (bia 0000.0c70.d31e)
Internet address is 131.108.1.1/24
  
```

例 3-68 给出了 `debug ip rip` 命令的输出。在调试中, R1 不能发送或接收 RIP 更新, 因为第 2 层不能正常工作。

例 3-68 `debug ip rip` 命令的输出表明 R1 的 Ethernet 0 接口没有发送或接收任何信息

```

R1#debug ip rip
RIP protocol debugging is on
R1#
  
```

在调试中, 因为故障, 该命令没有任何输出。

(2) 解决方法

RIP 在第 2 层之上运行。如果第 2 层不能工作, RIP 就无法发送或接收任何路由。要解决该问题, 必须先解决第 2 层或第 1 层的问题。有时, 问题可能很简单, 比如线缆松动或必须替换坏线缆, 也可能很复杂, 比如硬件故障, 必须替换硬件。

例 3-69 给出了改正第 2 层问题之后的接口 Ethernet 0

例 3-69 解决第 2 层问题后 R1 的输出接口 Ethernet 0 正常工作

```

R1#
Ethernet0 is up, line protocol is up
Hardware is Lance, address is 0000.0c70.d31e (bia 0000.0c70.d31e)
Internet address is 131.108.1.1/24
  
```

例 3-70 给出了 R2 的路由表。

例 3-70 R1 的 Ethernet 0 接口正常工作，所以 RIP 发送更新，并且 R2 路由表中包含 RIP 路由

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

3. 发送方没有公布 RIP 路由——原因: distribute-list out 阻塞路由

distribute-list out 可以过滤从接口发送出去的路由。如果接收者没有收到某条应该收到的路由，那么检查一下 distribute-list out 是否将其过滤掉。如果是这样，需要修改访问列表。

图 3-24 给出了该问题的解决方法流程图。

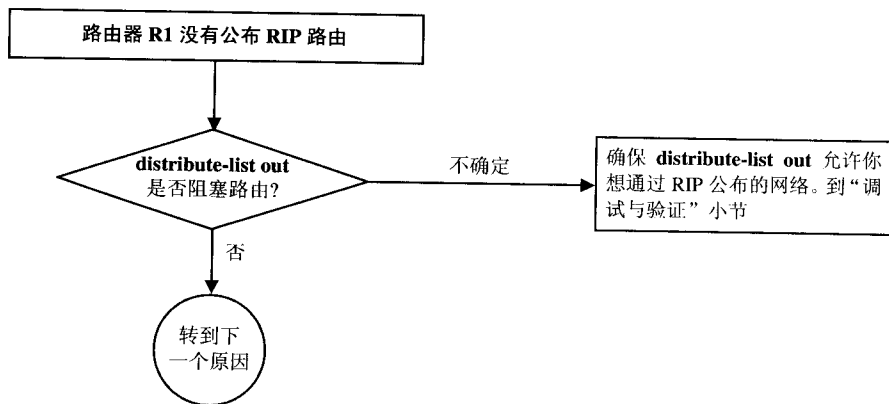


图 3-24 解决发送方不公布 RIP 路由问题的流程图

(1) 调试和验证

例 3-71 给出了路由器 R1 的配置情况。在该配置中，**access-list 1** 不允许网络 131.108.0.0 通过，所以 R1 不公布 131.108.X.X 网络，包括 131.108.2.0/24。

例 3-71 access-list 1 不允许网络 131.108.0.0 通过

```
R1#
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router rip
network 131.108.0.0
distribute-list 1 out
!
access-list 1 permit 131.107.0.0 0.0.255.255
```

(2) 解决方法

使用分发列表时，两次检查访问列表确保需要允许的网络确实显式地在访问列表中被允许。如果没有，它们就会被拒绝。例 3-72 中配置访问列表只允许 131.107.0.0。每一个访问列表末尾隐含的 **deny any** 命令使 131.108.0.0 被拒绝。为解决该问题，在 **access-list 1** 中允许 131.108.0.0，如例 3-72 所示。

例 3-72 重新配置 access-list 1 以允许 131.108.0.0 网络

```
interface Loopback0
ip address 131.108.2.1 255.255.255.0
!
interface Ethernet0
ip address 131.108.1.1 255.255.255.0
!
router rip
network 131.108.0.0
distribute-list 1 out
!
access-list 1 permit 131.108.0.0 0.0.255.255
```

例 3-73 给出了路由器 R2 的路由表。

例 3-73 在 access-list 1 中允许 131.108.0.0 网络后 R2 的路由表包含该网络的入口

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1
```

4. 发送方没有公布 RIP 路由——原因：公布的网络接口不能工作

公布的网络可能不能工作，所以连接的路由就从路由表中被删除。这时，RIP 开始以度量 16 公布该网络。当超过保持时间时，它不再公布该网络。而当公布网络起来工作时，RIP 又会在更新中重新进行公布。

图 3-25 给出了解决该问题的流程图。

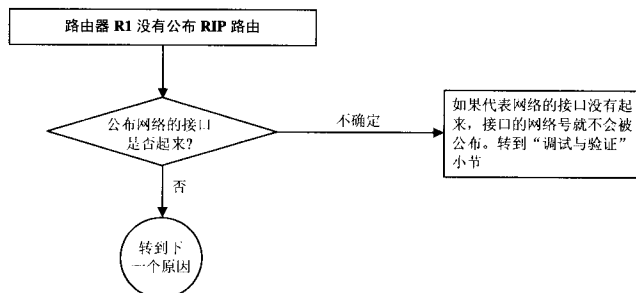


图 3-25 解决发送方不公布路由的问题的流程图

(1) 调试和验证

例 3-74 表明 R1 的 Ethernet 1 接口的线路协议不能工作，也就是在第 2 层存在问题。该接口与需要公布的网络直接相连。因此，网络不能被公布到邻居路由器。

例 3-74 show interface 输出显示公布网络的线路协议不能工作

```
R1#show interface Ethernet 1
Ethernet1 is up, line protocol is down
Hardware is Lance, address is 0000.0c70.d51e (bia 0000.0c70.d51e)
Internet address is 131.108.2.1/24
```

当公布的网络接口不能工作时，RIP 检测不能工作的原因。RIP 将不再 RIP 公布更新中的网络。在例 3-74 中，接口 Ethernet 1 不能工作，所以 RIP 不再在更新中公布 131.108.2.0/24。

(2) 解决方法

必须在第 2 层或第 1 层解决问题。有时问题可能很简单，如线缆松动，问题也可能很复杂，如硬件损坏，必须更换硬件。在解决第 2 层问题之后，重新使用 **show interface** 命令查看当前的状态，确保改变状态的网络已经能够工作。

例 3-75 表明公布网络接口的线路协议已工作。

例 3-75 show interface 输出表明解决第 2 层问题后 Ethernet 1 接口的线路协议正常工作

```
R1#show interface Ethernet 1
Ethernet1 is up, line protocol is up
Hardware is Lance, address is 0000.0c70.d51e (bia 0000.0c70.d51e)
Internet address is 131.108.2.1/24
```

当接口重新成为活动接口，RIP 就开始在定期更新中公布网络。例 3-76 表明不能工作的路由重新回到了 R2 的路由表中。

例 3-76 show ip route 命令的输出可以看出解决第二层问题后网络出现在 R2 的路由表中

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
Known via "rip", distance 120, metric 1
Redistributing via rip
Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
Routing Descriptor Blocks:
* 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
Route metric is 1, traffic share count is 1
```

5. 发送方没有公布 RIP 路由——原因：输出接口定义为被动接口

也可能因为别的原因导致路由器虽然拥有完整路由表但是并不向其他运行 RIP 的路由器公布路由。当 RIP 网络的所有路由器不都有完整的路由表时，可能会使网络的某一部分与另一部分失去联系。如果输出接口定义为被动接口，它就不会公布任何接口上的 RIP 更新。

图 3-26 给出了问题的解决方法流程图。

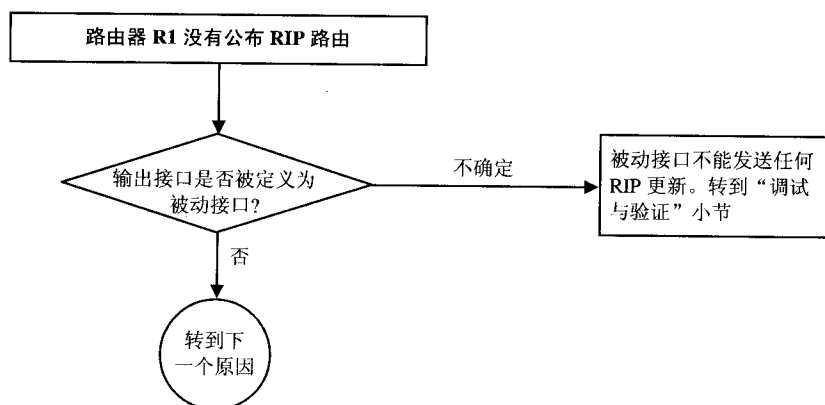


图 3-26 解决发送方不公布 RIP 路由的问题的流程图

(1) 调试和验证

例 3-77 给出了 **show ip protocols** 的输出，表明输出接口被定义为被动接口。

例 3-77 **show ip protocols** 的输出表明 R1 的输出接口是被动接口

```

R1#show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 26 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface Send Recv Key-chain
    Loopback0 1 1 2
  Routing for Networks:
    131.108.0.0
  Passive Interface(s) Ethernet0
  Routing Information Sources:
    Gateway Distance Last Update
    131.108.1.2 120 00:00:26
  Distance: (default is 120)
  
```

例 3-78 给出了路由器 R1 的配置，输出接口被定义为被动接口。

例 3-78 在 RIP 中配置 **passive interface** 命令

```

router rip
  passive-interface Ethernet0
  network 131.108.0.0
  
```

(2) 解决方法

当 RIP 中的接口定义为被动接口时，RIP 从该接口接收更新，但不会发送任何更新。

在例 3-78 中，接口 Ethernet 0 是被动接口，所以 R1 不能在 Ethernet 0 发送任何更新。有时，某些网络被公布，而另一些被过滤。在这种情况下，不能使用被动接口。分发列表有选

择地过滤更新比较适用于这种情况。

如果错误地配置了 **passive-interface**, 那么可以使用该命令的 **no** 形式来去掉配置中的该命令以解决问题。

例 3-79 显示了解决该问题的新配置。

例 3-79 解决 passive-interface 问题

```
router rip
network 131.108.0.0
```

例 3-80 给出解决问题后 R2 的路由表。

例 3-80 去掉 passive-interface 命令之后 R2 的路由表

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Serial0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Serial0
      Route metric is 1, traffic share count is 1
```

6. 发送方没有公布 RIP 路由——原因：组播能力出现问题（帧中继封装失败）

在有些网络中, 路由器接口不会自动地传播组播或公布流量, 除非被配置为这样做。这可能是个很主要的问题, 因为 **RIP-1** 更新在广播地址发送而 **RIP-2** 使用组播地址交换路由。除非在每个接口启用广播或组播之后才能在网络之间传播路由信息。无广播的组播接入 (NBMA) 帧中继是这种接口行为的典型例子。

图 3-27 给出故意将网络配置为受损的组播形式的网络情况, 以说明帧中继 **RIP** 更新不能穿过 R1。

图 3-27 中路由器 1 和路由器 2 通过帧中继相连。路由器 1 没有向路由器 2 公布 **RIP** 路由。



图 3-27 NBMA 帧中继网络容易遭受损坏组播能力的问题

图 3-28 给出了由这一原因引起的问题的解决方法流程图。

(1) 调试和验证

例 3-81 给出了路由器 R1 的配置。本例中, 帧中继提供第 2 层封装。在该配置中, **frame-relay map** 语句最后没有关键字 **broadcast**, 因此, 所有的广播流量和组播流量都不能通过 NBMA 网络。关键字 **broadcast** 使路由器复制必要的广播信息并通过专门的线路传送出去。

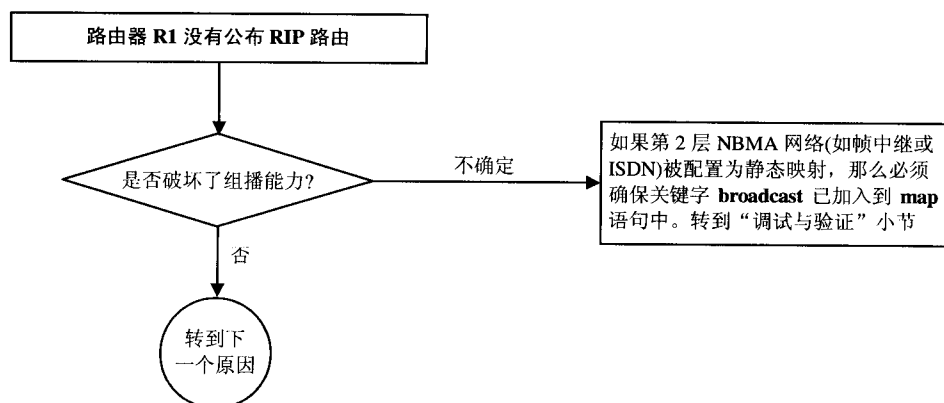


图 3-28 解决发送方不公布 RIP 路由的问题的流程图

例 3-81 R1 的 frame-relay map 语句缺少关键字 broadcast

```

R1#
interface Serial3
 ip address 131.108.1.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 131.108.1.2 16
 !
  
```

例 3-82 给出了 **debug ip packet** 命令的输出。该调试只包括从 R1 广播出去的源流量。如例 3-82 所示, R1 配置了 **access-list 100**。

例 3-82 用 access-list 100 配置 R1 以限制 debug 输出

```

R1#:
access-list 100 permit ip host 131.108.1.1 host 255.255.255.255
  
```

R1 配置了 **access-list 100**, 允许所有来自 131.108.1.1 发往 255.255.255.255 的分组通过。在例 3-83 中, R1 运行 **debug ip packet detail** 命令, 用 **access-list 100** 限制 R1 作为源发出的到 255.255.255.255 的流量。例 3-83 中调试输出表明存在封装错误, 即它们没有放在正确的第 2 层帧中。

例 3-83 R1 的 debug ip packet 输出表明 RIP 更新的封装错误

```

R1#debug ip packet 100 detail
IP packet debugging is on (detailed) for access list 100
R1#
IP: s=131.108.1.1 (local), d=255.255.255.255 (Serial3), len 112, sending broad/
    multicast
UDP src=520, dst=520
IP: s=131.108.1.1 (local), d=255.255.255.255 (Serial3), len 112, encapsulation
    failed
UDP src=520, dst=520
  
```

(2) 解决方法

在帧中继 (NBMA) 环境中运行的 RIP 必须将第 2 层配置为支持广播流量, 否则 RIP 更新就不能传送。当使用静态 map-ping 时, 确保在 **frame-relay map** 语句末尾加入了关键字 **broadcast**。

例 3-84 给出了路由器 R1 带有正确的 **frame-relay map** 语句的新配置。

例 3-84 允许广播流量通过 NBMA 环境的正确配置

```
R1#:  
interface Serial3  
  ip address 131.108.1.1 255.255.255.0  
  encapsulation frame-relay  
  frame-relay map ip 131.108.1.2 16 broadcast  
!
```

例 3-85 给出了 R2 带有 RIP 路由的路由表。

例 3-85 路由器 R1 用 frame-relay map 语句正确配置后 R2 的带有 RIP 路由的路由表

```
R2#show ip route 131.108.2.0  
Routing entry for 131.108.2.0/24  
  Known via "rip", distance 120, metric 1  
  Redistributing via rip  
  Last update from 131.108.1.1 on Serial0, 00:00:07 ago  
  Routing Descriptor Blocks:  
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Serial0  
      Route metric is 1, traffic share count is 1
```

7. 发送方没有公布 RIP 路由——原因: 错误配置 neighbor 语句

RIP 在非广播环境中利用单播方法发送 RIP 更新。发送单播 RIP 更新必须认真配置 **neighbor** 语句。如果 **neighbor** 语句中邻居地址配置错误, 那么 RIP 就不能向邻居发送单播更新。

图 3-29 给出了由这一原因引起的问题的解决方法流程图。

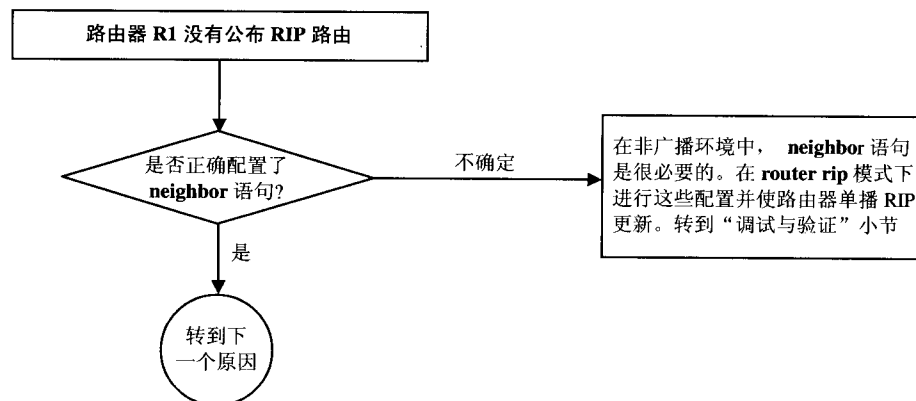


图 3-29 解决不能发送 RIP 路由的问题的流程图

(1) 调试和验证

例 3-86 给出了路由器 R1 的 RIP 配置。配置表明 **neighbor** 语句没有正确配置。不是指向 131.108.1.2，而是指向实际上不存在的 131.108.1.3。

例 3-86 使用了错误的 neighbor 语句的路由器 R1 的配置

```
router rip
network 131.108.0.0
neighbor 131.108.1.3
```

(2) 解决方法

例 3-86 中 RIP 向实际上不存在的邻居地址 131.108.1.3 发送单播更新。

要解决该问题，必须重新正确配置 **neighbor** 语句。

例 3-87 给出了路由器 R1 的正确配置。

例 3-87 使用正确的 neighbor 语句配置路由器 R1

```
R1# router rip
network 131.108.0.0
neighbor 131.108.1.2
```

例 3-88 表明 R2 的路由表中安装了 RIP 路由。

例 3-88 正确配置了 RIPneighbor 语句后 R2 路由表包含 RIP 路由入口。

```
R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Serial0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Serial0
      Route metric is 1, traffic share count is 1
```

8. 发送方没有公布 RIP 路由——原因：公布的子网是 VLSM

几乎所有的 IP 网络都通过初始 IP 块的变长子网掩码（VLSM）有效地使用 IP 地址。因为 RIP-1 不支持 VLSM 路由，路由 VLSM 的路由信息成为运行 RIP 的网络中常见的问题。

图 3-30 给出了因为使用 VLSM 而产生问题的网络设置。该图显示路由器 1 有一个掩码为/25 的接口。注意 131.108.0.0 是到两个不同掩码 131.108.1.0/24 和 131.108.2.0/25 的可变子网。

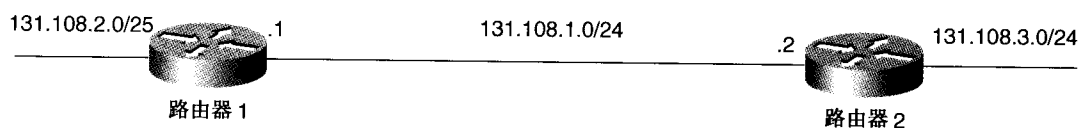


图 3-30 产生 RIP 问题的 VLSM 网络举例

RIP-1 不能公布子网掩码，所以它不支持 VLSM，不能向掩码为/24 的 RIP 接口公布/25 的路由。

图 3-31 给出了由这一原因引起的问题的解决方法流程图。

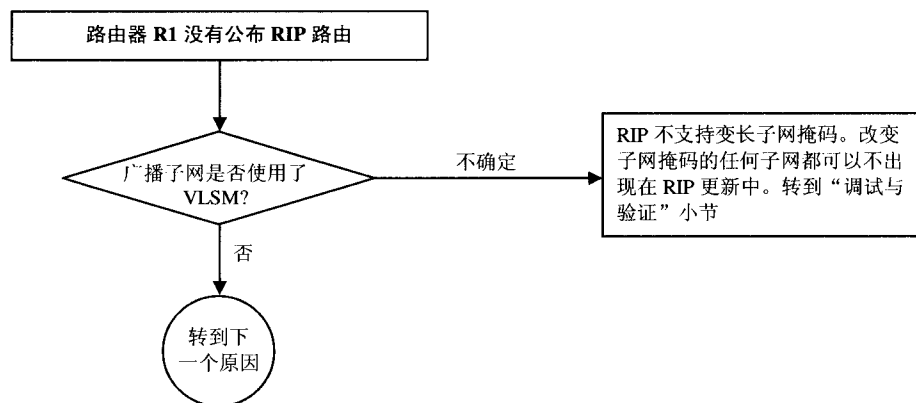


图 3-31 解决发送方不公布 RIP 路由的问题的流程图

(1) 调试和验证

例 3-89 表明 R1 的回环接口的子网掩码配置为/25 (255.255.255.128)。源发 RIP 更新的接口的子网掩码是/24 (255.255.255.0)。

例 3-89 显示 VLSM 子网的配置

```
R1#:  
interface Loopback0  
ip address 131.108.2.1 255.255.255.128  
!  
interface Ethernet0  
ip address 131.108.1.1 255.255.255.0  
!  
router rip  
network 131.108.0.0
```

(2) 解决方法

RIP-1 不能携带子网掩码信息。因此,任何与源发 RIP 更新的接口不同掩码的子网都不能公布 RIP 信息。RIP 在发送更新之前都会进行检查,以确保 RIP 公布的子网与发送 RIP 更新的接口具有相同的掩码。如果掩码不同,RIP 将丢弃更新,不会进行公布。

可以通过更改子网掩码使其与源发 RIP 更新的接口掩码相匹配,也可以通过使用支持 VLSM 的 RIP-2 协议解决该问题。

例 3-90 给出了解决问题的配置变更。

例 3-90 将 RIP 配置为公布 VLSM 路由

```
R1#:  
interface Loopback0  
ip address 131.108.2.1 255.255.255.0  
!  
interface Ethernet0
```

(待续)

```

ip address 131.108.1.1 255.255.255.0
!

router rip
version 2
network 131.108.0.0

```

例 3-91 给出解决问题后路由器 R2 的路由表。

例 3-91 解决对 VLSM 支持问题之后路由器 R2 的路由表

```

R2#show ip route 131.108.2.0
Routing entry for 131.108.2.0/25
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 131.108.1.1 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
    * 131.108.1.1, from 131.108.1.1, 00:00:07 ago, via Ethernet0
      Route metric is 1, traffic share count is 1

```

9. 发送方没有公布 RIP 路由——原因：使用水平分割

水平分割是 RIP 控制避免产生路由环的一种方法。有时使用水平分割控制路由环是非常必要的。例如，当从一个接口接收更新但不能从相同的接口再发送更新时，使用水平分割就非常必要。在其他的环境中必须关掉水平分割，例如在一个 hub-and-poke 的帧中继环境中，spoke 之间没有线路，它们通过 hub 路由器，如图 3-32 所示。

另一个值得一提的特殊情况是路由器有一个带下一跳地址的外部路由，该地址通过其他 RIP 路由器的一些接口获得。当这些外部路由重发布至 RIP 时，路由器因为使用水平分割而不会在相同的接口公布该路由。另外，如果为接口配置了第 2 地址，也应该关闭水平分割，否则，第 2 地址不会从该接口公布到其他路由器。

图 3-33 给出了在打开水平分割时产生 RIP 问题的网络设置。路由器 1 没有向路由器 3 公布所有的 RIP 路由。

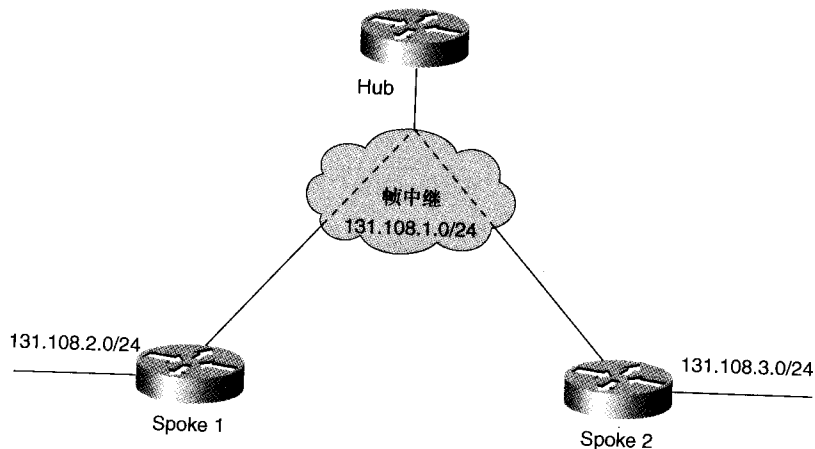


图 3-32 hub-and-spoke 帧中继网络需要关闭水平分割

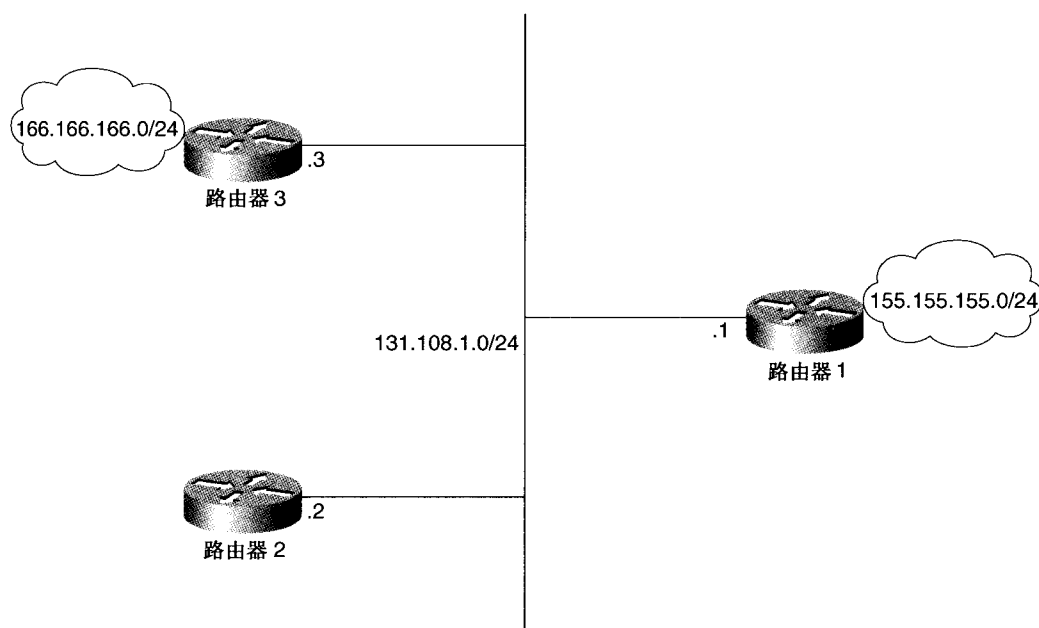


图 3-33 使用水平分割的网络易于产生 RIP 问题

图 3-34 给出了该问题的解决方法流程图。

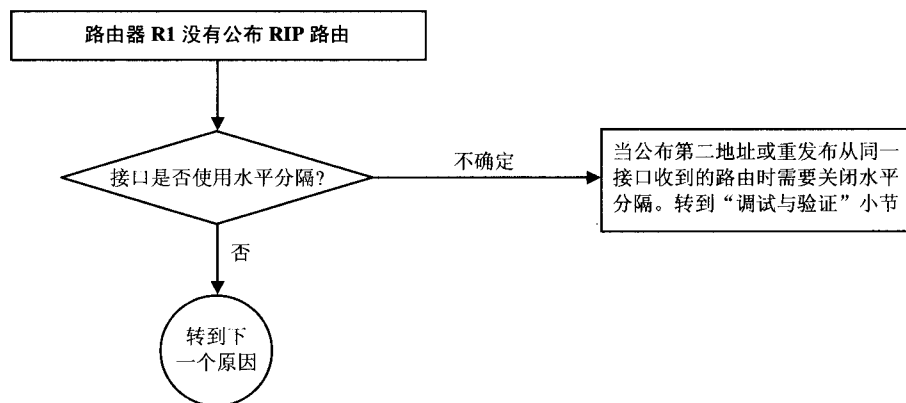


图 3-34 解决发送方不公布 RIP 路由问题的流程图

(1) 调试和验证

例 3-92 给出 R1 的当前配置。

例 3-92 将 166.166.166.0/24 重发布到 R1 的 RIP 中

```

R1#
router rip
 redistribute static
 network 131.108.0.0
!
ip route 155.155.0.0 255.255.0.0 10.10.10.4
ip route 166.166.166.0 255.255.255.0 131.108.1.3
  
```

例 3-93 表明路由器 R2 的路由表中没有路由 166.166.166.0/24，但是有 155.155.155.0/24。

例 3-93 R2 的路由表中没有路由 166.166.166.0/24

```
R2#show ip route rip
R    155.155.0.0/16 [120/1] via 131.108.1.1, 00:00:07, Ethernet0
```

例 3-94 给出了路由器 R1 上 debug ip rip 的输出。R1 只公布 155.155.0.0/16，没有公布 166.166.166.0/24。因此 R2 的路由表中没有路由 166.166.166.0/24。

例 3-94 debug ip rip 命令的输出表明 R1 没有公布 166.166.166.0

```
R1#debug ip rip
RIP protocol debugging is on

RIP: sending v1 update to 255.255.255.255 via Ethernet0 (131.108.1.1)
RIP: build update entries
     network 155.155.0.0 metric 1
```

(2) 解决方法

问题产生的原因是 166.166.166.0/24 的下一跳是 131.108.1.2。使用水平分割时，RIP 抑制了从与学习 166.166.166.0/24 的接口相同的接口学到的更新。注意，路由 155.155.155.0/24 是 R1 公布的，因为该路由的下一跳地址是 10.10.10.4，与 R1 的接口不同。

解决方法就是在 R1 的 Ethernet 0 接口上关闭水平分割。

相似的情况是，如果将 R1 的二级地址定义为 166.166.166.0/24，就不会在关闭水平分割后在 RIP 更新中公布二级接口地址。

例 3-95 给出了解决问题后路由器 R1 的新配置。

例 3-95 在 R1 的 Ethernet 0 接口上关闭水平分割

```
R1#
interface Ethernet0
 ip address 131.108.1.1 255.255.255.0
 no ip split-horizon
```

例 3-96 说明在配置改变后，R2 在 RIP 更新中收到 166.166.166.0/24。

例 3-96 关闭水平分割后 R2 的路由表可以证明 RIP 更新影响路由 166.166.166.0/24

```
R2#show ip route rip
R    155.155.0.0/16 [120/1] via 131.108.1.1, 00:00:08, Ethernet0
R    166.166.0.0/16 [120/1] via 131.108.1.1, 00:00:08, Ethernet0
```

当用第二 IP 地址配置接口时也会出现问题。例 3-97 给出了使用第二 IP 地址的接口配置。

例 3-97 用第二地址配置接口

```
R1#
interface Ethernet0
 ip address 131.108.2.1 255.255.255.0 secondary
 ip address 131.108.1.1 255.255.255.0
```

如果打开水平分割，Ethernet 0 接口上不会公布第二地址。

同理，如图 3-35 所示，假设同一个以太网上有三台路由器——R1、R2 和 R3。

同前边的例子一样，R1 和 R3 运行 OSPF，R1 和 R2 运行 RIP。现在 R3 通用 OSPF 向 R1 公布某些路由，R1 必须重发布至 RIP。R1 因为使用了水平分割，所以不会将 OSPF 路由公布给 RIP。解决方法就是关闭水平分割。

当用第二 IP 地址配置接口时也会出现问题。例 3-97 给出了使用第二 IP 地址的接口配置。

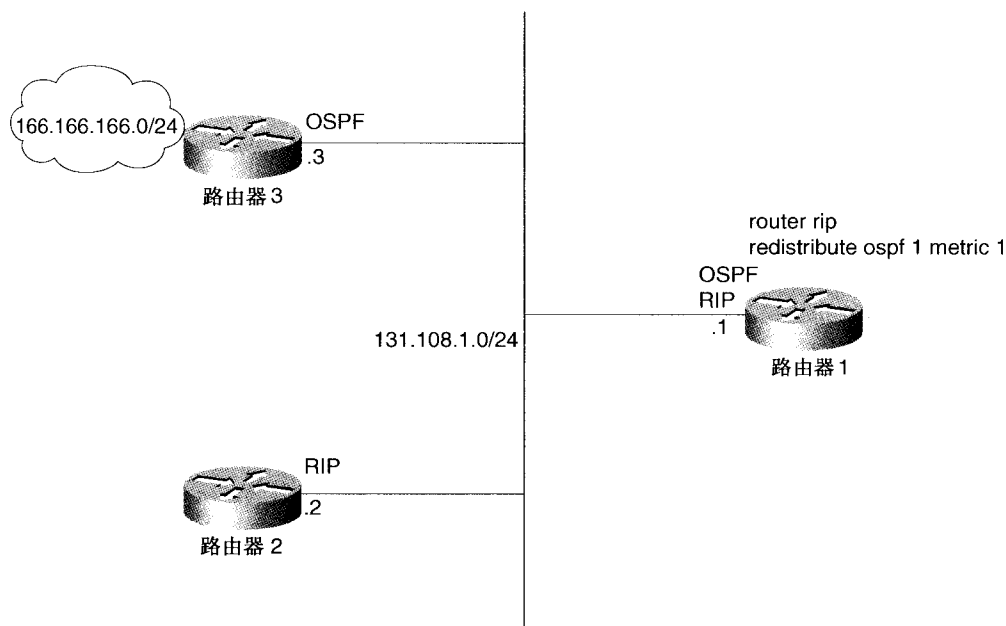


图 3-35 另一个容易产生 RIP 问题的使用水平分割的网络

关闭水平分割的原因基本上有三种。而如果关闭水平分割又可能产生路由环问题。

3.3.2 问题：R2 的路由表中缺少子网路由——原因：使用了自动汇总功能

有时 RIP 不公布子网路由。当 RIP 向主网络边界发送更新时，更新将自动汇总。这其实并不是错误，而这样做又可以减小路由表的大小。

图 3-36 给出的 R1 的子网是 155.155.0.0 的网络设置，R2 的路由表中却没有这些路由的子网设置。可能是 R1 没有向 R2 公布这些路由，也可能是 R2 没有收到这些路由。更大的可能是 R1 没有公布关于子网 155.155.0.0/16 的详细信息。

例 3-98 给出了缺少子网路由 155.155.0.0/16 但具有主网络路由的 R2 的路由表。这说明 R1 公布了路由，但是将子网汇总为 15.155.0.0/16。

例 3-99 用 RIP-1 配置 R1

```
R1#
interface Loopback1
 ip address 131.108.2.1 255.255.255.0
!
interface Loopback3
 ip address 155.155.155.1 255.255.255.0
!
interface Ethernet0
 ip address 131.108.1.1 255.255.255.0
!
router rip
 network 131.108.0.0
 network 155.155.0.0
```

例 3-100 给出了路由器 R2 的路由表。注意 R2 接收的是 155.155.0.0/16，而不是 R1 上的配置 155.155.155.0/24。R2 接收的是路由 131.108.2.0/24，它与接口 Ethernet 0 处于同一主网络，该主网络将 R1 与 R2 相连。

例 3-100 R2 的路由显示了子网路由如何在有类边界进行汇总

```
R2#show ip route RIP
R    155.155.0.0/16 [120/1] via 131.108.1.1, 00:00:22, Ethernet0
     131.108.0.0/24 is subnetted, 3 subnets
R    131.108.2.0 [120/1] via 131.108.1.1, 00:00:22, Ethernet0
```

(2) 解决方法

RIP-1 不可能产生这类问题，因为 RIP-1 是有类路由协议。RIP-1 自动把经过配置不同主网络接口的任何更新汇总到一个自然类边界。

从例 3-100 中 R2 的路由表可以看出 155.155.155.0/24 从配置了 131.108.0.0 的接口被公布出去。这把 155.155.155.0/24 汇总为 B 类边界 155.155.0.0/16。

RIP-1 中不会出现这类问题，因为 RIP-1 是有类协议，网络也应该基于这个基础设计。但是使用 RIP-2 时，Cisco 路由器可以配置为关闭自动汇总过程。

例如，R1 的配置可以改为运行 RIP-2 过程，而不是 RIP-1 过程。

例 3-101 给出了解决使用 RIP-2 时出现问题的配置。

例 3-101 关闭 RIP-2 的自动汇总

```
router rip
 version 2
 network 131.108.0.0
 network 155.155.0.0
 no auto-summary
```

例 3-102 给出了路由器 R2 的路由表，它说明路由器 2 接收到了正确的子网路由 155.155.155.0/24。

例 3-102 路由器 R2 的路由表显示路由器 2 正在接收子网路由 155.155.155.0/24

```
R2#show ip route 155.155.0.0
155.155.0.0/24 is subnetted, 1 subnets
R    155.155.155.0 [120/1] via 131.108.1.1, 00:00:21, Ethernet0
131.108.0.0/24 is subnetted, 3 subnets
R    131.108.2.0 [120/1] via 131.108.1.1, 00:00:21, Ethernet0
```

3.4 RIP 路由汇总疑难解析

路由汇总是指汇总或减少路由表中路由的数量。例如，131.108.1.0/24、131.108.2.0/24 和 131.108.3.0/24 可以变成一个路由项（即 131.108.0.0/16 或 131.108.0.0/22），后者只包括这三个子网。路由汇总（本节内容既包括自动汇总也包括手工汇总）可以减少路由表的大小。本节讨论与路由汇总最相关的问题——RIP-2 路由表太大。最可能的两个原因如下：

- 关闭了自动汇总。
- 没有使用 ip summary-address。

图 3-38 给出了可能产生很大的路由表的网络设置情况。

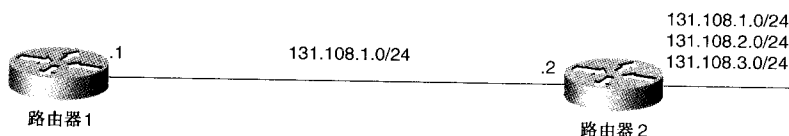


图 3-38 可能产生大路由表的网络设置

3.4.1 问题：RIP-2 路由表过大——原因：关闭了自动汇总

当 RIP 更新通过主网络时，它汇总到有类边界。例如，当 131.108.1.0/24、131.108.2.0/24 和 131.108.3.0/24 公布到没有 131.108.X.X 地址的网络接口时，它们就会自动汇总为 131.108.0.0/16。关闭自动汇总会增大路由表的大小，有时必须关闭该功能（例如，如果存在前面所讨论的非连续网络）。

图 3-39 给出了该原因引起的问题的解决方法流程图。

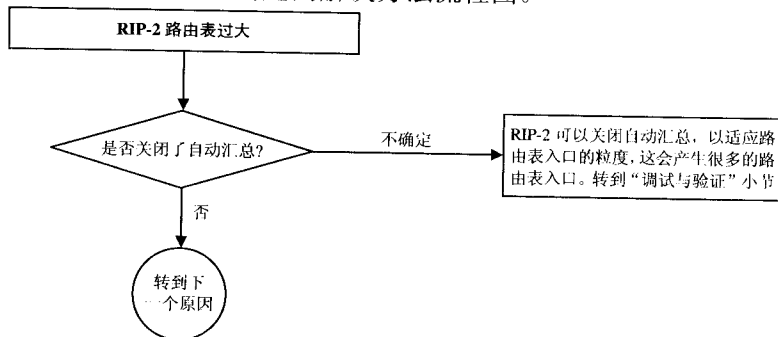


图 3-39 解决 RIP-2 路由表过大问题的流程图

(1) 调试和验证

例 3-103 给出了产生这类问题时 R2 的配置。在该配置中, R2 关闭了自动汇总。

例 3-103 关闭 R2 上 RIP 的自动汇总

```
R2#  
router rip  
version 2  
network 132.108.0.0  
network 131.108.0.0  
no auto-summary
```

例 3-104 给出 R1 的路由表。该路由表只有四个路由项,但是在例 3-103 配置的实际网络中,应该有几百条路由。R1 接收到 131.108.0.0/16 的每个子网。在本例中,只有三条,还可能更少。

例 3-104 R1 的路由表中有子网路由

```
R1#show ip route rip  
131.108.0.0/24 is subnetted, 3 subnets  
R    131.108.3.0 [120/1] via 132.108.1.2, 00:00:24, Serial3  
R    131.108.2.0 [120/1] via 132.108.1.2, 00:00:24, Serial3  
R    131.108.1.0 [120/1] via 132.108.1.2, 00:00:24, Serial3  
R1#
```

(2) 解决方法

因为在 R2 上 RIP 的配置中关闭了自动汇总功能,所以 R1 在路由表中就能够看到子网路由。而使用自动汇总功能后,所有的子网路由都消失了。

例 3-105 给出了 R2 的另一种配置。该配置使用自动汇总功能以减少路由表的大小。因为这是缺省配置,所以你可能在它的配置中看到。启用自动汇总的命令是在 router rip 之后使用 auto-summary。

例 3-105 R2 使用自动汇总减少路由表的大小

```
R2#  
router rip  
version 2  
network 132.108.0.0  
network 131.108.0.0
```

例 3-106 给出了减小后的路由表。

例 3-106 自动汇总减少了路由器 R1 的路由表的大小

```
R1#show ip route rip  
R 131.108.0.0/16 [120/1] via 132.108.1.2, 00:00:01, Serial3  
R1#
```

3.4.2 问题：RIP-2 路由表过大——原因：没有使用 ip 汇总地址

图 3-40 给出了另一种产生大路由表的网络设置。

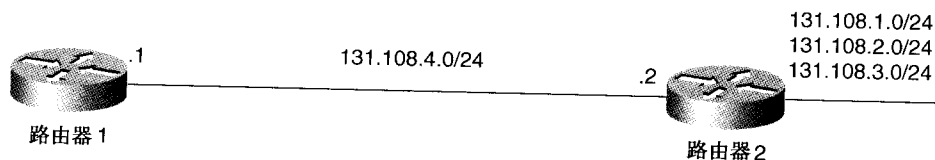


图 3-40 可能产生大路由表的网络设置

从图 3-40 可以看出，R2 有网络 131.108.0.0 的几条子网路由。注意，R1 和 R2 之间的链接也是网络 131.108.0.0 的一部分，所以不能使用自动汇总解决接收可以汇总的子网路由的问题。只有当 R1 和 R2 之间的链接在不同主网络时才能使用自动汇总功能。

图 3-41 给出了该原因引起的问题的解决方法流程图。

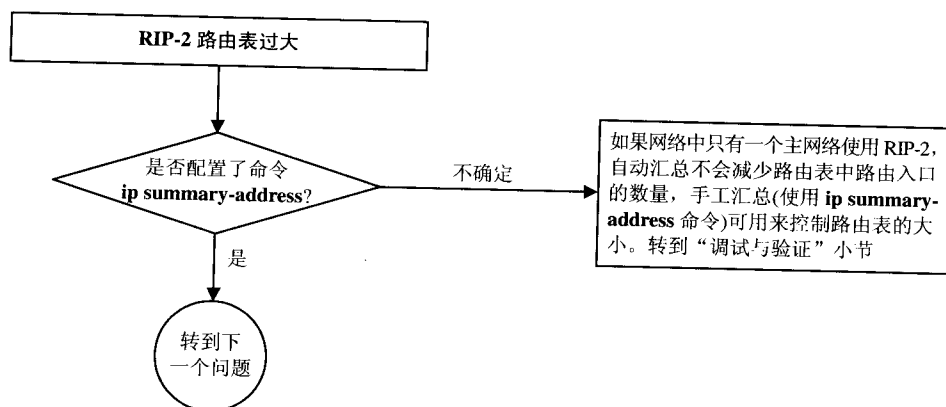


图 3-41 解决过大的 RIP-2 路由表问题的流程图

(1) 调试和验证

从例 3-107 可以看出 R2 的配置中没有在串口 1 使用 ip summary-address 命令来汇总路由。

例 3-107 R2 的串口没有配置为汇总路由

```

R2#
interface Serial1
 ip address 131.108.4.2 255.255.255.0
!
router rip
 version 2
 network 131.108.0.0
  
```

例 3-108 给出了 R1 的路由表。在本例中，只有三条路由。而在一个真实的网络中，如果像例 3-107 那样配置，路由的数量就可能多得非常可怕。

例 3-108 R1 的路由表中包含子网路由

```
R1#show ip route rip
      131.108.0.0/24 is subnetted, 3 subnets
R       131.108.3.0 [120/1] via 131.108.4.2, 00:00:04, Serial3
R       131.108.2.0 [120/1] via 131.108.4.2, 00:00:04, Serial3
R       131.108.1.0 [120/1] via 131.108.4.2, 00:00:04, Serial3
R1#
```

(2) 解决方法

在上一节所述的情况下，打开自动汇总对处于同一主网络的整个网络没有帮助。设想一个有成千个子网的使用 B 类地址的网络。这时，自动汇总不会起到任何作用，因为没有穿过任何主网络边界。Cisco IOS 12.0.7 T 中引入了一项汇总的新功能。该功能与 EIGRP 的手工汇总非常相似。

例 3-109 给出了解决该问题的新配置。该配置能够减少路由表的大小。可以根据不同的掩码使用该命令，以使在很多连续子网块的网络中，路由器能够被配置为把网络的子网块汇总为更小的块，从而减少公布到 RIP 网络的路由。

例 3-109 对 RIP 进行手工汇总

```
R2#:  
interface Serial1  
 ip address 131.108.4.2 255.255.255.0  
 ip summary-address rip 131.108.0.0 255.255.252.0  
!  
router rip  
 version 2  
 network 131.108.0.0
```

在前面的配置中，R2 在串口上对 RIP 路由进行汇总。网络 131.108.0.0 中的任何网络子网都会被汇总到一个 131.108.0.0 主网络，掩码是 255.255.252.0。这样 R2 只须发布一条汇总路由 131.108.0.0/22，把子网压缩为 131.108.0.0。

例 3-110 给出了进行汇总使路由项减少的路由器 R1 的路由表。

例 3-110 汇总后 R1 的路由表减小

```
R1#show ip route rip  
R       131.108.0.0/22 [120/1] via 131.108.4.2, 00:00:01, Serial3  
R1#
```

3.5 RIP 重发布问题疑难解析

本节讨论 RIP 重发布时可能遇到的问题。重发布是指将另一个路由协议、静态路由或直连路由注入到 RIP 中的过程。使用路由重发布时一定要避免路由环问题。另外，还要在处理过程中定义好度量（跳数），以免产生问题。

RIP 重发布最常见的问题是重发布路由没有进入接收这些路由的 RIP 路由器的路由表中。当路由表中没有发往目的地址的路由时，任何数据都无法到达目的地。产生这种现象最常见的原因是 RIP 重发布中没有定义度量。

在 RIP 中，路由的度量可以看作跳数，它给出了路径中存在的路由器的数量。如第 2 章所述，15 是 RIP 支持的最大跳数。超过 15 的任何数都是无穷大度量，因此也都会在接收后被丢弃。

图 3-42 给出了重发布路由不能安装到接收路由器的路由表中的网络设置。

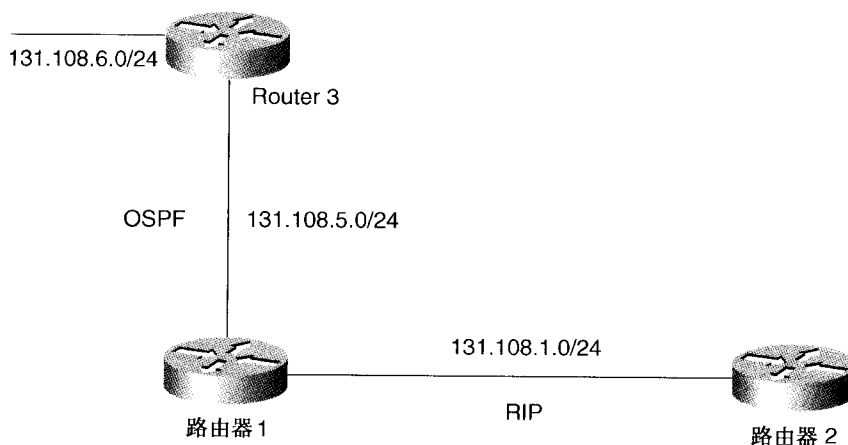


图 3-42 易产生重发布问题的网络

R1 和 R3 在区域 0 运行 OSPF，R1 和 R2 运行 RIP。R3 将 131.108.6.0/24 通过 OSPF 发布出去。在 R1 中，OSPF 路由正重发布到 RIP，但 R2 不能通过 RIP 接收 131.108.6.0/24。

图 3-43 给出了解决由于该原因产生的问题的方法流程图。

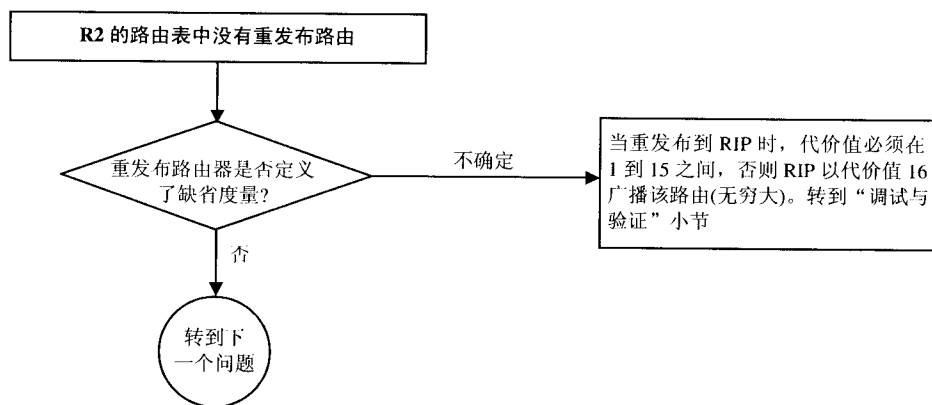


图 3-43 解决路由重发布问题的流程图

(1) 调试和验证

为解决该问题，需要先检查 R1 是否接收到了路由 131.108.6.0/24。

例 3-111 表明 R3 正通过 OSPF 将 131.108.6.0/24 发布到 R1。