

实验 8.4.1：研究路由表查找过程

学习目标

完成本实验后，您将能够：

- 根据拓扑图进行网络布线。
- 清除启动配置并将路由器重置为默认状态。
- 在路由器上执行基本的配置任务。
- 确定 1 级路由和 2 级路由。
- 修改配置以实现静态路由和默认路由。
- 启用有类路由并研究有类路由行为。
- 启用无类路由并研究无类路由行为。

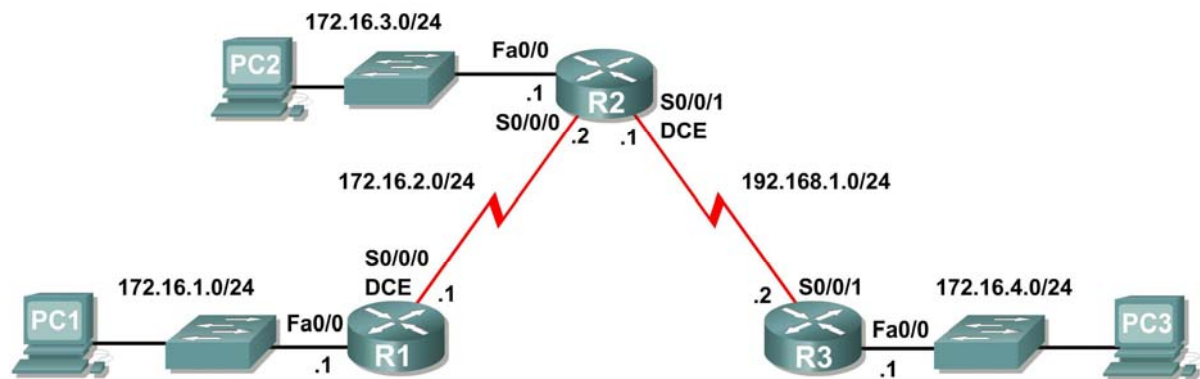
场景

在本实验练习中，有两个独立的场景。在第一个场景中，您将查看路由表中的 1 级路由和 2 级路由。在第二个场景中，您将查看有类路由行为和无类路由行为。

- 场景 A：1 级路由和 2 级路由
- 场景 B：有类路由行为和无类路由行为

场景 A：1 级路由和 2 级路由

拓扑图



地址表

设备	接口	IP 地址	子网掩码	默认网关
R1	Fa0/0	172.16.1.1	255.255.255.0	不适用
	S0/0/0	172.16.2.1	255.255.255.0	不适用
R2	Fa0/0	172.16.3.1	255.255.255.0	不适用
	S0/0/0	172.16.2.2	255.255.255.0	不适用
	S0/0/1	192.168.1.1	255.255.255.0	不适用
R3	Fa0/0	172.16.4.1	255.255.255.0	不适用
	S0/0/1	192.168.1.2	255.255.255.0	不适用
PC1	网卡	172.16.1.10	255.255.255.0	172.16.1.1
PC2	网卡	172.16.3.10	255.255.255.0	172.16.3.1
PC3	网卡	172.16.4.10	255.255.255.0	172.16.4.1

任务 1：准备网络。

步骤 1：布置一个与拓扑图中类似的网络。

您可以在实验中使用任何路由器，只要它具备拓扑图中所要求的接口即可。

注意：如果您选用 1700 路由器、2500 路由器或 2600 路由器，则所显示的路由器输出和接口描述会有所不同。

步骤 2：清除路由器中现有的所有配置。

任务 2：执行基本的路由器配置。

按照以下指导说明对 R1 路由器、R2 路由器和 R3 路由器执行基本配置：

1. 配置路由器主机名。
2. 禁用 DNS 查找。
3. 配置执行模式口令。
4. 配置当日消息标语。
5. 配置控制台连接口令。
6. 配置 VTY 连接口令。

任务 3：配置并激活串行地址和以太网地址。

步骤 1：配置 R1、R2 和 R3 的接口。

使用拓扑图下面表格中的 IP 地址配置 R1、R2 和 R3 路由器的接口。

步骤 2：检验 IP 地址和接口。

使用 **show ip interface brief** 命令检验 IP 地址配置正确并且接口处于活动状态。

完成后，应确保将运行配置保存到路由器的 NVRAM 中。

步骤 3：配置 PC1、PC2 和 PC3 的以太网接口。

使用拓扑图下面表格中的 IP 地址和默认网关配置 PC1、PC2 和 PC3 的以太网接口。

步骤 4：从 PC ping 默认网关, 测试 PC 配置。

任务 4：配置 RIP。

在每台路由器上配置 RIPv1 路由。包括每个直连网络的 **network** 语句。

任务 5：观察当前删除的路由和添加到路由表中的路由。

步骤 1：查看 R1 路由器上的路由表。

路由表中显示了哪些网络？

步骤 2：使用 `debug ip routing` 命令观察 R1 路由器上出现的路由表变化情况。

```
R1#debug ip routing
IP routing debugging is on
```

步骤 3：关闭 `Serial0/0/0` 接口并观察调试输出。

```
R1(config-if)#shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to administratively
down
RT: interface Serial0/0/0 removed from routing table
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state
to down
RT: del 172.16.2.0 via 0.0.0.0, connected metric [0/0]
RT: delete network route to 172.16.2.0
RT: NET-RED 172.16.2.0/24
RT: del 172.16.3.0 via 172.16.2.2, rip metric [120/1]
RT: delete network route to 172.16.3.0
RT: NET-RED 172.16.3.0/24
RT: del 192.168.1.0 via 172.16.2.2, rip metric [120/1]
RT: delete network route to 192.168.1.0
RT: NET-RED 192.168.1.0/24
```

步骤 4：查看 R1 路由器上的路由表，观察禁用 `Serial0/0/0` 接口后，路由表的变化情况。

```
R1# show ip route
```

<省略了部分输出内容>

<Output omitted>

Gateway of last resort is not set

```
      172.16.0.0/24 is subnetted, 1 subnets
C      172.16.1.0 is directly connected, FastEthernet0/0
R1#
```

步骤 5：启用 Serial0/0/0 接口并观察调试输出。

```
R1(config-if)#no shutdown
```

```
RT: SET_LAST_RDB for 172.16.2.0/24  
NEW rdb: is directly connected
```

```
RT: add 172.16.2.0/24 via 0.0.0.0, connected metric [0/0]
```

```
RT: NET-RED 172.16.2.0/24RT: SET_LAST_RDB for 172.16.0.0/16  
NEW rdb: via 172.16.2.2
```

```
RT: add 172.16.3.0/24 via 172.16.2.2, rip metric [120/1]
```

```
RT: NET-RED 172.16.3.0/24RT: SET_LAST_RDB for 192.168.1.0/24  
NEW rdb: via 172.16.2.2
```

```
RT: add 192.168.1.0/24 via 172.16.2.2, rip metric [120/1]
```

```
RT: NET-RED 192.168.1.0/24
```

为什么首先添加的是到 172.16.2.0/24 的路由？

为什么添加其它路由前会有延迟？

步骤 6：使用 no debug ip routing 命令或 undebug all 命令禁用调试输出。

任务 6：确定 1 级路由和 2 级路由。

步骤 1：查看 R1 路由表。

```
R1#show ip route
```

<省略了部分输出内容>

```
Gateway of last resort is not set
```

```
172.16.0.0/24 is subnetted, 3 subnets
```

```
C    172.16.1.0/24 is directly connected, FastEthernet0/0
```

```
C    172.16.2.0/24 is directly connected, Serial0/0/0
```

```
R    172.16.3.0/24 [120/1] via 172.16.2.2, 00:00:14, Serial0/0/0
```

```
R    192.168.1.0/24 [120/1] via 172.16.2.2, 00:00:14, Serial0/0/0
```

```
R1#
```

其中哪些是 1 级路由？

为什么这些路由是 1 级路由？

这些 1 级路由中是否有最终路由？

为什么该路由是最终路由？

这些 1 级路由中是否有父路由？

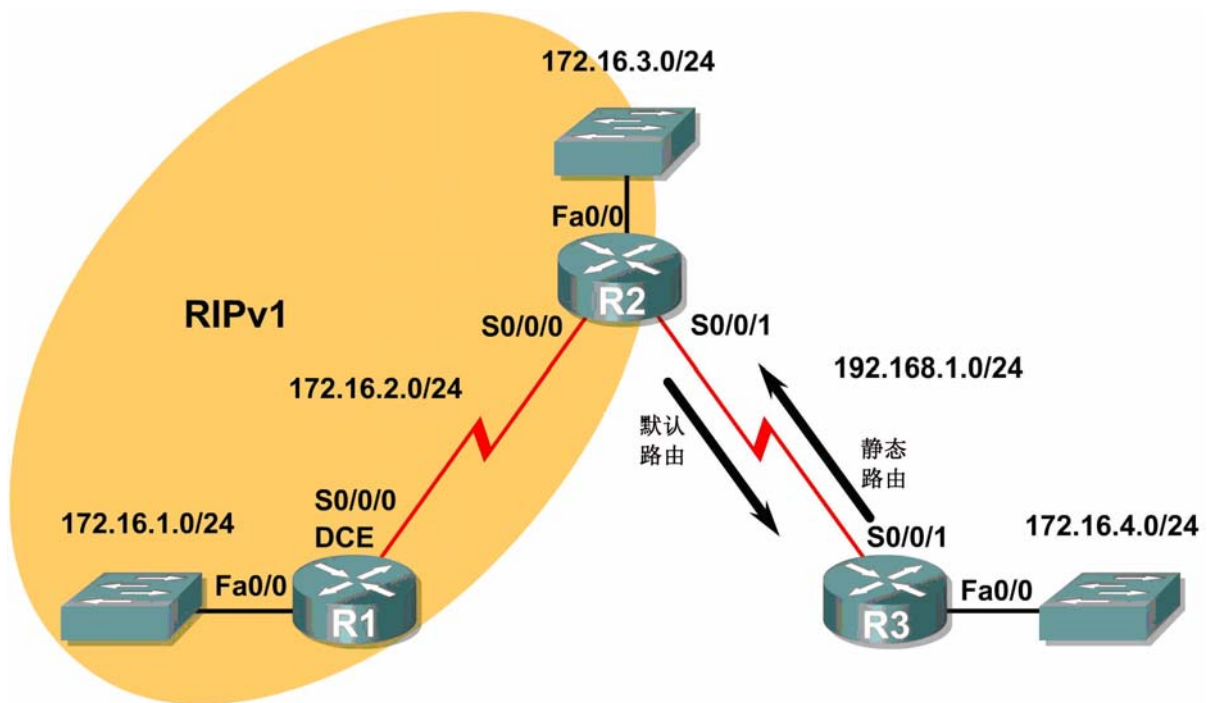
为什么该路由是一级父路由？

这些路由中哪些是 2 级路由？

为什么这些路由是 2 级路由？

场景 B：有类路由行为和无类路由行为

拓扑图



任务 1：将场景 A 改为场景 B

步骤 1：删除 R3 的 RIP 配置，并配置一条通往 172.16.0.0/16 的静态路由。

```
R3(config)#no router rip
R3(config)#ip route 172.16.0.0 255.255.0.0 Serial0/0/1
```

步骤 2：从 R2 的 RIP 配置中删除 192.168.1.0 网络。

```
R2(config)#router rip
R2(config-router)#no network 192.168.1.0
```

步骤 3：在 R2 路由器上添加一条通往 R3 的静态默认路由。

在配置中包括 **default-information originate** 命令，以便在 RIP 更新中包含这条默认静态路由。

```
R2(config)#ip route 0.0.0.0 0.0.0.0 Serial0/0/1
R2(config)#router rip
R2(config-router)#default-information originate
```


任务 2：在路由器上启用有类路由行为

步骤 1：使用 `no ip classless` 命令配置路由查找过程，以使用有类路由查找。

```
R1
R1(config)#no ip classless
```

```
R2
R2(config)#no ip classless
```

```
R3
R3(config)#no ip classless
```

步骤 2：查看 R2 路由器上的路由表。

```
R2#show ip route
```

〈省略了部分输出内容〉

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
      172.16.0.0/24 is subnetted, 4 subnets
R       172.16.1.0 [120/1] via 172.16.2.1, 00:00:00, Serial0/0/0
C       172.16.2.0 is directly connected, Serial0/0/0
C       172.16.3.0 is directly connected, FastEthernet0/0
C      192.168.1.0/24 is directly connected, Serial0/0/1
S*     0.0.0.0/0 is directly connected, Serial0/0/1
```

```
R2#
```

步骤 3：从 R2 ping PC3 并观察结果。

```
R2#ping 172.16.4.10
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5 )
```

Ping 操作失败，这是因为路由器正在使用有类路由行为。

R2 路由器上的路由查找过程搜索路由表，发现目的地址的前 16 位与父路由 172.16.0.0/16 匹配。因为目的地址与父路由匹配，所以接下来会检查子路由。

172.16.0.0/16 父网络的子路由有哪些？

子路由与数据包的目的地址必须满足多少位的匹配条件才能用该路由来转发数据包？ _____

Ping 数据包的目的地址是否与 172.16.0.0/16 的子路由匹配？ _____

由于已使用 **no ip classless** 命令对 R2 路由器进行了配置，R2 使用的是有类路由行为，因此，如果某条 1 级父路由与目的地址匹配，则路由器不会在该父路由的子路由的范围之外搜索匹配位数较少的路由。即使配置了默认静态路由，路由器也不会使用该路由，而会丢弃数据包。

任务 3：在路由器上启用无类路由行为

步骤 1：使用 **ip classless** 命令重新启用无类路由。

```
R1
R1(config)#ip classless
```

```
R2
R2(config)#ip classless
```

```
R3
R3(config)#ip classless
```

步骤 2：查看 R2 上的路由表。

请注意，即使更改了路由器配置，使用无类路由行为，路由表也不会有任何变化。

```
R2#show ip route
```

<省略了部分输出内容>

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0

      172.16.0.0/24 is subnetted, 4 subnets
R       172.16.1.0 [120/1] via 172.16.2.1, 00:00:00, Serial0/0/0
C       172.16.2.0 is directly connected, Serial0/0/0
C       172.16.3.0 is directly connected, FastEthernet0/0
C      192.168.1.0/24 is directly connected, Serial0/0/1
S*     0.0.0.0/0 is directly connected, Serial0/0/1
R2#
```

步骤 3：再次从 R2 向 PC3 发送 Ping 命令，观察输出结果。

```
R2#ping 172.16.4.10
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent, round-trip min/avg/max = 28/28/28 ms
```

Ping 操作成功，这是由于当前路由器使用的是无类路由行为。

数据包的目的地址与 1 级父路由 172.16.0.0/16 匹配，但与该父路由的所有子路由都不匹配。

由于配置了无类路由行为，路由器会继续搜索路由表，查找匹配位数较少的路由，这样的路由仍旧算是匹配路由。默认路由的掩码是 /0，也就是说不需要匹配。在无类路由行为中，即使数据包的目的 IP 地址与其它路由都不匹配，但总会与默认路由匹配。

```
S* 0.0.0.0/0 is directly connected, Serial0/0/1
```

由于在 R2 路由器上配置了默认路由，因此会使用该路由将数据包转发给 PC3。

步骤 4：查看 R3 上的路由表，确定 Ping 命令生成的流量如何返回到 R2。

```
R3#show ip route
```

```
<output omitted>
```

```
Gateway of last resort is not set
```

```
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
S      172.16.0.0/16 is directly connected, Serial0/0/1
C      172.16.4.0/24 is directly connected, FastEthernet0/0
C      192.168.1.0/24 is directly connected, Serial0/0/1
R3#
```

请注意，在 R3 的路由表中，172.16.4.0/24 子网路由和 172.16.0.0/16 有类网络路由都是 172.16.0.0/16 父路由的 2 级子路由。在本例中，R3 使用 172.16.0.0/16 子路由，通过 Serial 0/0/1 接口将返回流量转发给 R2。