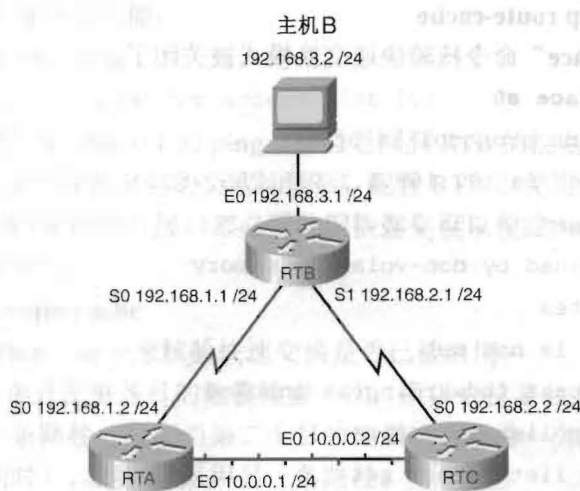


### 1.6.1 实验 1-1: 采用 RIP 的等开销负载均衡



#### 1. 目标

在这个实验中, 我们通过使用高级的调试命令来观察基于各数据包和各目的地的等可开销负载均衡。

#### 2. 背景

路由器 RTB 有两条到网络 10.0.0.0 的路径。我们打算用高级的调试特性来核验两条路径都被用于为到 10.0.0.0 的负载分担数据流, 并且同时测试基于各数据包和各目的地的负载均衡。

#### 3. 步骤

在开始这个实验前, 建议先删除各路由器的初始配置后再重新启动路由器。这样可以防止由残留配置所造成的问题。在我们准备好设备后, 进行步骤 1。

步骤 1: 根据实验 1-1 图示组建和配置网络。别忘了设置主机 B 的 IP 地址和缺省网关。

该配置要求使用子网 0, 所以根据我们所使用的 IOS 版本, 可能需要输入 “ip subnet-zero” 命令。使用 RIPv1 路由协议, 并且用适当的 “network” 命令在所有活跃接口上启用路由更新功能。

```
RTA(config)#router rip
```

```
RTA(config-router)#network 192.168.1.0
```

```
RTA(config-router)#network 10.0.0.0
```

确保在其他两台路由器上也用适当的 “network” 命令配置 RIP。

用 “ping” 命令测试所有接口之间的连通性。

步骤 2: 用 “show ip route” 命令检查路由器 RTB 上的路由表。在它的路由表中应该有两条到网络 10.0.0.0 的路由。如果不是这样的话, 就需要进行排错。

RIP 将用等开销路由自动进行负载均衡。注意, 两条路由的度量值 (在这种情况下, 是跳数) 都为 1。RIP 不能进行非等开销负载均衡, 但 IGRP 可以。

步骤 3: 要配置基于各数据包的负载均衡, S0 和 S1 都必须使用进程交换。进程交换强迫路由器为每个被路由数据包的目的地网络而检索路由表。与此相反, 快速交

将初始的路由表检索储存在高速缓存中,并用缓存起来的信息为到相同目的地的数据包选路转发。

在路由器 RTB 的两个串行接口上用下面的接口配置命令,启用进程交换:

RTB(config-if)#no ip route-cache

用“show ip interface”命令核验快速交换模式被关闭了:

RTB#sh ip interface s0

```
Serial0 is up,line protocol is up
Internet address is 192.168.1.1 255.255.255.0
Broadcast address is 255.255.255.255
Address determined by non-volatile memory
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is enabled
Outgoing access list is not set
Inbound access list is not set
Proxy ARP is enabled
Security level is default
Split horizon is enabled
ICMP redirects are always sent
ICMP unreachable are always sent
ICMP mask replies are never sent
IP fast switching is disabled
<output omitted>
```

步骤 4: 因为路由表中有两条到同一个目的地网络的路由,所以一半数据包通过一条路径进行发送,另一半通过另一条路径进行发送。路径选择随所收到的每个数据包而交替变换。我们可以用“debug ip packet”命令观察这个过程,该命令将输出与路由器所发送和接收的 IP 数据包有关的信息:

RTB#debug ip packet

打开调试功能后,从位于 192.168.3.2 的主机 B 向 10.0.0.1 发送几个 ping 数据包,然后返回到路由器 RTB 的控制台。当 ping 被发送时,路由器将输出 IP 数据包信息。用下面的命令在几个 ping 成功之后停止调试:

RTB#undebug all

让我们来分析一下“debug”的输出。它可能有点儿令人迷惑,因为 ping 请求和 ping 答复是混在一起的。看一下包括 d=10.0.0.1 (目的地址)的输出行。在这些输出行上,看一下数据包被发送出去的接口。输出接口应该在串口 0 和串口 1 之间交替变换:

```
IP:s=192.168.3.2 (Ethernet0),d=10.0.0.1 (Serial1),g=192.168.2.1,len 84,forward
IP:s=192.168.3.2 (Ethernet0),d=10.0.0.1 (Serial0),g=192.168.1.1,len 84,forward
IP:s=192.168.3.2 (Ethernet0),d=10.0.0.1 (Serial1),g=192.168.2.1,len 84,forward
IP:s=192.168.3.2 (Ethernet0),d=10.0.0.1 (Serial0),g=192.168.1.1,len 84,forward
```

我们也可以让“debug”命令只输出我们感兴趣的信息。为达该目的,我们需要配置一

```
RTB(config)#access-list 101 permit icmp any 10.0.0.0 0.255.255.255
```

RTB#debug ip packet 101

```
IP packet debugging is on for access list 101
```

从我们的工作站重复向 10.0.0.1 的 ping，并且返回到 RTB 的控制台来观看输出。

```
RTB(config-if)#ip route-cache
```

用“**show ip interface**”命令来核验快速交换是否已被启用。

步骤 6: 因为路由表只根据各目的地被检索一次, 所以去往某台主机的所有数据包都将遵循同一条路径。只有当第二个目的地强迫进行另一次路由表检索 (或当缓存条目到期时), 才可能采用另一条路径。

使用“**debug ip packet 101**”命令，并且从我们的工作站 **ping** 10.0.0.1。注意数据包被发送出去时所用的串行接口。

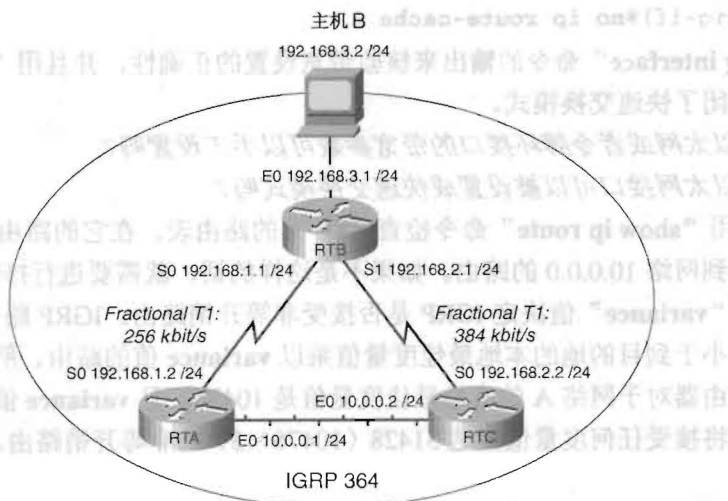
现在 ping 10.0.0.2。这次，路由器应该已经选择了一个不同的外出接口。

我们只看到一个 **ping** 的 **debug** 输出信息，因为在采用快速交换模式的情况下，处理器只被用于初始的路由表检索。当路由器存在对某个目的地址的路由缓存条目时，“**debug ip packet**”将不再显示与去往该目的地数据包有关的信息，因为处理器不参与每次的转发决定。

尽管我们不会得到答复，但我们可以通过向没有被用到的地址 10.0.0.3 和 10.0.0.4 发送 ping 来观察路由器选择什么路径。

最后，我们可以用“**show ip cache**”命令来查看路由缓存的内容。注意对于 10.0.0.1 和 10.0.0.2，还有我们最近在网络 10.0.0.0 上 ping 的任何其他地址，都存在映射关系。

主机 B  
192.168.3.2/24



### 1. 目标

在这个实验中,我们将通过使用高级的调试命令来观察在 IGRP 网络上的非等可开销负载均衡。

### 2. 背景

RTB 有两条到网络 10.0.0.0 的路径,但它们有不同的 IGRP 度量值。我们打算配置非等开销负载均衡,并且用高级调试特性来验证之。

### 3. 步骤

在开始这个实验前,建议先删除各路由器的初始配置后再重新启动路由器。这样可以防止由残留配置所造成的问题。在我们准备好设备后,进行步骤 1。

步骤 1: 根据实验 1-2 的图示组建和配置网络。该配置要求使用子网 0,所以根据我们使用的 IOS 版本,可能需要输入“**ip subnet-zero**”命令。使用 IGRP 路由协议,假定 AS 号为 364,并且用适当的“**network**”命令在所有的活跃接口上启用路由更新功能。

```
RTA(config)#router igrp364
```

```
RTA(config-router)#network 192.168.1.0
```

```
RTA(config-router)#network 10.0.0.0
```

确保在其他两台路由器上也用适当的“**network**”命令配置 IGRP。

用“**ping**”命令核验所有接口之间的连通性。

因为 IGRP 的度量值在它的计算中包括带宽,所以,我们必须手工配置串行接口的带宽以确保准确性。(在设置适当的带宽之前,路由器 RTB 到网络 10.0.0.0 的另一条路径不是非等开销的。)用下面的命令进一步为 RTB 配置正确的带宽和进程交换:

```
RTB(config)#in s0
```

```
RTB(config-if)#bandwidth 256
```

```
RTB(config-if)#no ip route-cache
```

```
RTB(config-if)#in s1
```

```
RTB(config-if)#bandwidth 384
```

```
RTB(config-if)#no ip route-cache
```

用“**show interface**”命令的输出来核验带宽设置的正确性,并且用“**show ip interface**”命令来确保关闭了快速交换模式。

问题 1. 以太网或者令牌环接口的带宽参数可以手工设置吗?

问题 2. 以太网接口可以被设置成快速交换模式吗?

步骤 2: 用“**show ip route**”命令检查 RTB 上的路由表。在它的路由表中应该只有一条到网络 10.0.0.0 的路由。如果不是这样的话,就需要进行排错。

步骤 3: “**variance**”值决定 IGRP 是否接受非等开销路由。IGRP 路由器只接受等于或小于到目的地的本地最佳度量值乘以 **variance** 值的路由。所以,如果 IGRP 路由器对于网络 A 的本地最佳度量值是 10476,且 **variance** 值是 3,那么路由器将接受任何度量值高达 31428 ( $10476 \times 3$ ) 的非等开销路由。IGRP 路由器将只



接受到同一网络的最多4条路径。注意，只有在路径中的下一台路由器比当前路由器更靠近目的地（也就是说，有更低的度量值）时，才安装另一条路径。

缺省地，IGRP 的 **variance** 值被设置为 1，也就意味着只有等于本地最佳度量值的路由才被安装。因此，将 **variance** 值设为 1 将关闭非等开销负载均衡。

用下面的命令配置 RTB 以启用非等开销负载均衡：

```
RTB(config)#router igrp 364
```

```
RTB(config-router)#variance 10
```

问题3. 根据命令帮助特性，**variance** 的最大值是多少？

步骤4: 再次查看 RTB 的路由表。RTB 应该有两条度量值不等的到网络 10.0.0.0 的路径。

问题4. 通过 RTA 到 10.0.0.0 的路由的 IGRP 度量值是什么？

问题5. 通过 RTC 到 10.0.0.0 的路由的 IGRP 度量值是什么？

步骤5: 现在，我们可以在调试的同时通过从 RTB 向网络 10.0.0.0 发送一个 **ping** 数据包来测试非等开销负载均衡。首先，配置一个访问控制列表来限制从 RTB 到网络 10.0.0.0 的 **ping** 请求的调试输出：

```
RTB(config)#access-list 101 permit icmp any 10.0.0.0 0.255.255.255
```

启用调试功能，并用访问控制列表来过滤输出：

```
RTB(config)#debug ip packet 101
```

最后，从主机 B **ping** 10.0.0.1 和 10.0.0.2。

问题6. 数据包是按目的地还是按各数据包被负载均衡？

问题7. 非等开销负载均衡和等开销负载均衡有什么不同？