



Laboratório de Pesquisa em Redes e Multimídia

O Protocolo RIP

Prof. José Gonçalves

Departamento de Informática - UFES

zgonc@inf.ufes.br



Universidade Federal do Espírito Santo
Departamento de Informática

RIP – Routing Information Protocol

- É o protocolo *distance vector* mais antigo e ainda em grande uso.
- Possui duas versões:
 - RIPv1 – classful
 - RIPv2 – classless
- É baseado nos algoritmos desenvolvidos por Bellman, Ford e Fulkerson.

Operação do RIP

- O RIP usa a porta 520 do UDP. Todas as mensagens RIP são encapsuladas em segmentos UDP.
- O RIP define dois tipos de mensagens:
 - *Request*: são usadas para requerer dos roteadores vizinhos informações de roteamento (que eles enviem um *update*).
 - *Response*: carrega o update.
- A métrica usada é o *hop count*
 - Hop count = 1: rede diretamente conectada
 - Hop count = 16: rede inatingível

Operação do RIP (cont.)

- No *startup*, o RIP faz um broadcast da mensagem de *Request* em toda interface em que o RIP está habilitado.
- O RIP entra então em um *loop*, esperando por mensagens RIP *Request* ou RIP *Response* de outros roteadores.
- Os vizinhos que recebem o *Request* enviam o *Response* contendo a sua tabela de rotas.
- Se a rota incluída no *update* é nova ela é registrada na tabela de roteamento, junto com o endereço do endereço que a anunciou.

Operação do RIP (cont.)

- Se a rota é para uma rede que já está na tabela, a entrada existente será substituída apenas se a nova rota apresentar um *hop count* menor.
- Se o *hop count* anunciado for maior do que o existente **e** se ele foi originado por um roteador vizinho já gravado na tabela, a rota será marcada como *unreachable* por um período de tempo especificado no *holddown timer* (180 s).
 - Não aceita imediatamente, espera 180s (“ceticismo”).
- Se ao final do período o vizinho ainda está anunciando o novo *hop count*, então aceita a rota.

RIP Timers: *Update Timer*

- Após o *startup*, o roteador envia, espontaneamente, a cada 30s, em cada interface em que o RIP está ativado, mensagens de *Response* (ou seja, os *Updates*).
- A mensagem de *Response* contém a tabela de rotas do roteador, com exceção das entradas suprimidas pela regra de *split horizon*.
- O *update timer* que inicia esse *update* periódico inclui uma variável randômica para prevenir o problema da sincronização de tabelas de roteamento (o que contribuiria para o aumento de colisões na rede).
- Como resultado, o tempo entre *updates* de um processo RIP típico pode variar de 25 a 35s (30s na média).
 - A variável usada pelo CISCO IOS, RIP_JITTER, subtrai até 15% (4.5s) do tempo de *update*. Logo, para roteadores CISCO, os *updates* variam de 25.5 a 30s.

RIP Timers: *ExpirationTimer*

- O RIP também emprega um *timer* para limitar o tempo que uma entrada pode permanecer na tabela de rotas
 - *Expiration timer* ou *timeout*
 - No CISCO IOS é chamado de *invalid timer*
- Sempre que uma nova rota é registrada na tabela o *invalid timer* correspondente é iniciado com valor de 180s (ou seja, 6 períodos de *update*).
- O *invalid timer* é resetado sempre que um *update* é ouvido para aquela rota.
- Se o *update* não for ouvido em 180s a rota é marcada como inatingível (faz *hop count* = 16).

RIP Timers: *Garbage Collection/Flush Timer*

- Um outro timer é o *garbage collection* ou *flush timer*. Este timer é setado em 240s, 60s a mais que o *invalid timer*.
- A rota é anunciada com a métrica *unreachable* até que o *garbage collection timer* expira, instante em que a rota é removida da tabela.
- A RFC 1058 prescreve um tempo de 120s a mais que o *invalid timer* mas o CISCO IOS implementa 60s.

RIP Timers: *Holddown Timer*

- Um *update* com um *hop count* maior do que a métrica registrada na tabela de rotas faz a rota entrar em *holddown*.
- Isto significa que o roteador espera por confirmação deste novo *hop count* durante 180s, isto é, três períodos de *update*.
- Embora a RFC não se refira a *holddown timers*, ele existe na implementação do RIP da CISCO.

Manipulando os Timers

- Os quatro *timers* do RIP podem ser manipulados pelo comando:
 - *timers basic update invalid holddown flush*
- Comando a ser manipulado com muito cuidado.

Triggered Updates (Updates não Periódicos)

- *Triggered updates* foram projetados para reduzir o tempo de convergência da rede.
 - Permite, portanto, reduzir o período no qual *loops* entre roteadores existem na rede.
- Um *triggered update* ocorre sempre que a métrica para uma rota é alterada.
- Diferentemente dos *updates* regulares, que enviam toda a tabela, pode incluir apenas a(s) entrada(s) alterada(s).

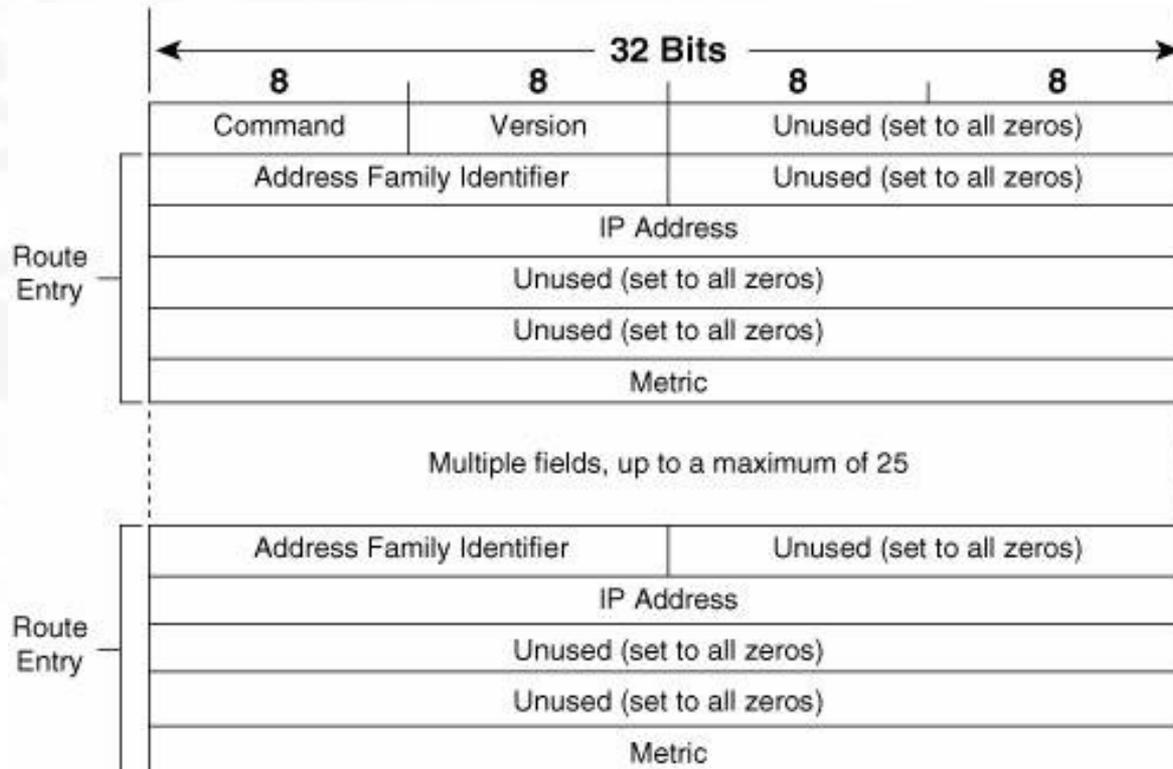
Split Horizon with Poison Reverse

- O RIP emprega o *Split Horizon with Poison Reverse*, uma evolução do *Split Horizon*. Nele, as rotas aprendidas de um roteador vizinho não são suprimidas no anúncio da tabela (como faz o *split horizon*) mas, sim, são reportadas com métrica *infinity* (mando a rota mas a marco como inatingível).
- Quando um roteador recebe do roteador vizinho rotas anunciadas como *infinity*, essas rotas são eliminadas imediatamente da tabela, sem esperar pelo *timeout* do *expiration timer*.
 - Com isso, um loop é eliminado muito mais rapidamente da rede.
- *Poisson reverse* aumenta o tamanho da informação sendo trocada já que agora é enviada toda a tabela. Isso não constitui um problema nas LANs mas pode ser problema em conexões ponto-a-ponto.

Hosts Operando em *Silent Mode*

- Alguns hosts podem usar o RIP em *silent mode*.
- Nesse modo, os hosts não geram RIP *updates* mas ouvem e atualizam as suas tabelas de rotas caso haja mudanças.
 - `% routed -q` habilita RIP em *silent mode* em hosts Unix.

Formato da Mensagem RIPv1



- Uma rota = 20 bytes
- Espaço para até 25 rotas
- Mensagem RIP = $4 + (20 \times 25) = 504$ bytes
- Header UDP = 20 bytes
- Tamanho da datagrama RIP = 524 bytes, sem incluir o cabeçalho do IP (que seriam mais 20 bytes).

Formato da Mensagem RIPv1 (cont.)

- Command: 1-request 2-reponse
- Version: 1 (RIPv1)
- Address Family Identifier: 2 para o IP
 - Exceção: um *request* por *full table* coloca esse campo em zero.
- IP Address: endereço destino da rota
 - Rede, sub-rede ou host
- Metrics: hop count, valores entre 1 e 16
- Zeros: influência do protocolo XNS e intenção de que o RIP se adequasse a um amplo conjunto de famílias de endereços.

RIP Request Message

- Pode requisitar:
 - (1) toda a tabela de rotas; ou
 - (2) informação sobre rota específica
- No caso (1) a mensagem de *request* terá uma única entrada, onde o AFI é setado em zero, o IP Address é 0.0.0.0 e a métrica é 16.
- Um dispositivo recebendo tal *request* responde via *unicast*, anunciando toda a sua tabela de rotas o endereço requisitante, levando em consideração regras como *split horizon* e *boundary summarization*.
- Algumas ferramentas de diagnóstico podem necessitar de informações sobre rotas específicas. Neste caso (2), o *request* lista as entradas específicas (os endereços em questão).

Classful Routing

- Quando um pacote chega em um roteador RIPv1 ele examina a sua tabela de rotas, isto é, executa um (*classful*) *routing table lookup*.
- Inicialmente, a porção NetID do endereço destino é examinada e a tabela consultada para saber se ocorre um match.
- Se não houver um match no nível de endereço classful (classe A, B ou C - major network), o pacote é descartado e uma mensagem de ICMP Destination Unreachable é enviada para a fonte do pacote.
- Se existe um match para a porção de rede, as subredes listadas para aquela rede são examinadas.
- Se um match para alguma das subredes é encontrado, o pacote é roteado senão uma mensagem ICMP é gerada.

```
MtPilate#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
Gateway of last resort is not set
 10.0.0.0 255.255.0.0 is subnetted, 9 subnets
R    10.10.0.0 [120/3] via 10.5.5.1, 00:00:20, Serial1
      [120/3] via 10.1.1.1, 00:00:21, Ethernet0
R    10.11.0.0 [120/3] via 10.5.5.1, 00:00:21, Serial1
      [120/3] via 10.1.1.1, 00:00:21, Ethernet0
R    10.8.0.0 [120/2] via 10.1.1.1, 00:00:21, Ethernet0
      [120/2] via 10.5.5.1, 00:00:21, Serial1
R    10.9.0.0 [120/2] via 10.5.5.1, 00:00:21, Serial1
      [120/2] via 10.1.1.1, 00:00:21, Ethernet0
R    10.3.0.0 [120/1] via 10.1.1.1, 00:00:21, Ethernet0
      [120/1] via 10.5.5.1, 00:00:21, Serial1
C    10.1.0.0 is directly connected, Ethernet0
R    10.6.0.0 [120/1] via 10.1.1.1, 00:00:21, Ethernet0
      [120/1] via 10.5.5.1, 00:00:22, Serial1
R    10.7.0.0 [120/2] via 10.1.1.1, 00:00:22, Ethernet0
      [120/2] via 10.5.5.1, 00:00:22, Serial1
C    10.5.0.0 is directly connected, Serial1
 172.25.0.0 255.255.255.0 is subnetted, 3 subnets
R    172.25.153.0 [120/1] via 172.25.15.2, 00:00:03, Serial0
R    172.25.131.0 [120/1] via 172.25.15.2, 00:00:03, Serial0
C    172.25.15.0 is directly connected, Serial0
```

Exemplo

- O RIP faz balanceamento de carga *equal-cost* se mais de uma rota existe para um mesmo destino, com igual número de *hop count*.
- [distância administrativa / métrica]

Exemplo (cont.)

- Destino: 192.168.35.3
 - Não há match para 192.168.35.0, logo o pacote é descartado.
- Destino: 172.25.33.89
 - *Match* para a rede 172.25.0.0/24 (classe B). Subredes são então analisadas.
 - Não há *match* para a subrede 172.25.33.0, logo o pacote é descartado.
- Destino: 172.25.153.220
 - *Match* para a rede 172.25.0.0/24.
 - *Match* para a subrede 172.25.153.0
 - Pacote é roteado para o *next-hop router* 172.25.15.2

Distância Administrativa do RIP

Route Source	Administrative Distance
Connected interface ^[*]	0
Static route	1
EIGRP summary route	5
External BGP	20
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
External EIGRP	170
Internal BGP	200
Unknown	255

- Métrica: medida de preferência por uma rota.
- Distância administrativa: medida de preferência pela maneira pela qual a rota foi descoberta.

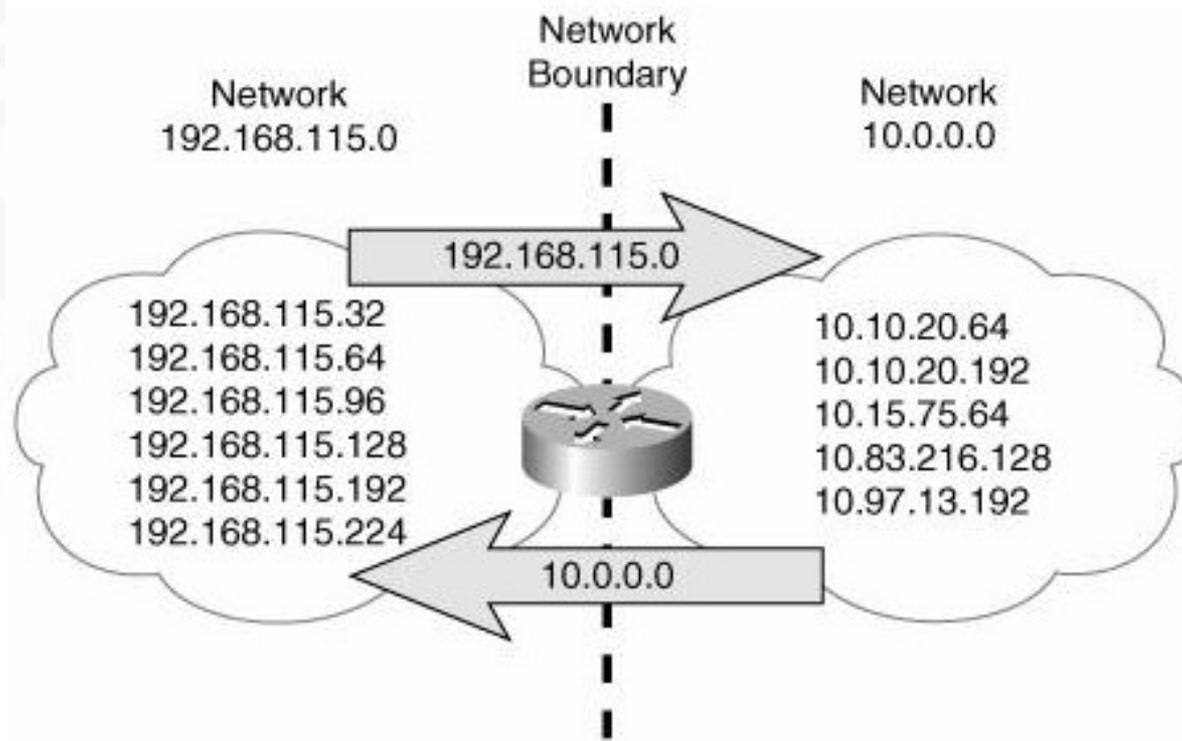
Subredes Diretamente Conectadas

- O RIPv1 não tem provisão para subrede (i.e., não tem campo de subrede na mensagem). Por conta disso, não há nenhuma máscara associada às subredes individuais na tabela de rotas.
- Se o roteador recebe um pacote com destino 172.25.131.23 não há como saber a fronteira entre NetID e HostID, ou mesmo se o endereço é subnetado.
- O único recurso do roteador é assumir que a máscara configurada em uma das suas interfaces ligadas à rede 172.25.0.0 é usada consistentemente na internet.
- Ele usará máscara localmente configurada para a rede 172.25.0.0 (definida no exemplo como 255.255.255.0) para derivar a subrede do endereço destino.

Sumarização no Roteador de Borda

- Como o RIP interpreta a subrede de uma *major network* se ele não tem nenhuma interface atachada àquela rede?
- Sem uma interface da classe A, B ou C do destino o roteador não tem como saber a máscara de subrede correta a ser usada e, portanto, não tem como identificar a subrede.
- Solução: ele precisa ter apenas uma entrada apontando para um roteador que está diretamente conectado.

Sumarização no Roteador de Borda (cont.)



Sumarização no Roteador de Borda (cont.)

- Desta forma, as tabelas de rotas de roteadores dentro da rede 192.168.115.0 tem uma entrada que direciona pacotes para 10.0.0.0 para o roteador de borda.
- O roteador de borda tem uma interface diretamente ligada à rede 10.0.0.0 e, portanto, possui uma máscara de subrede com a qual deriva a subrede para onde rotear o pacote dentro da nuvem.

Sumarização no Roteador de Borda (cont.)

```
Raleigh#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
```

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

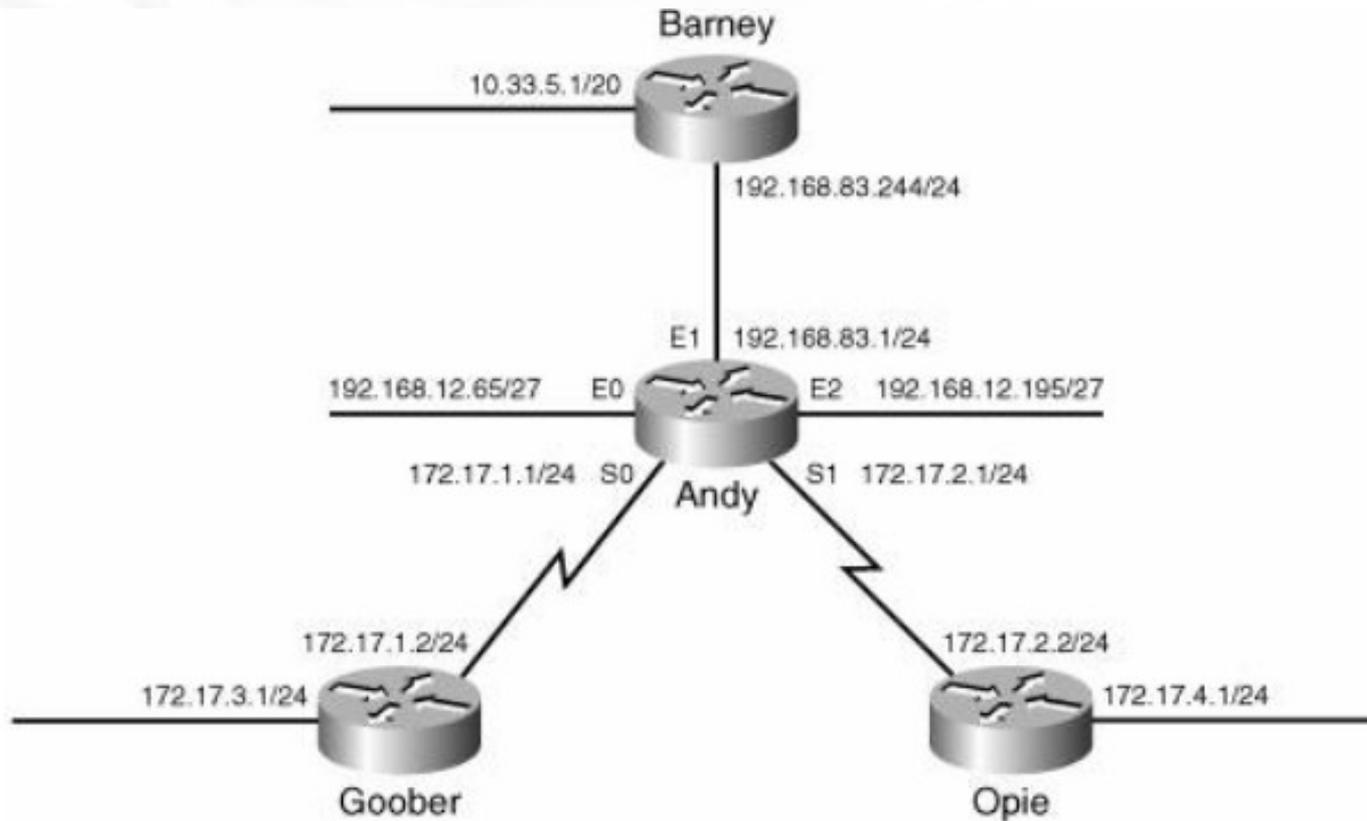
```
R    10.0.0.0 [120/1] via 192.168.115.40, 00:00:10, Ethernet1  
    192.168.115.0 255.255.255.240 is subnetted, 6 subnets  
C    192.168.115.32 is directly connected, Ethernet1  
R    192.168.115.64 [120/1] via 192.168.115.99, 00:00:13, Ethernet0  
C    192.168.115.96 is directly connected, Ethernet0  
C    192.168.115.128 is directly connected, Serial0  
R    192.168.115.192 [120/1] via 192.168.115.99, 00:00:13, Ethernet0  
R    192.168.115.224 [120/1] via 192.168.115.130, 00:00:25, Serial0
```

```
Raleigh#
```

Configurando o RIP

- Dois passos:
 - Habilitar o RIP com o comando **router rip**.
 - Especificar cada *major network* que pode rodar o RIP, usando com o comando **network**.
- A natureza classful do RIP e a sumarização na fronteira da rede (*subnet hiding*) impõem que nenhuma subrede pode ser especificada com o comando **network**.

Configurando o RIP (cont.)

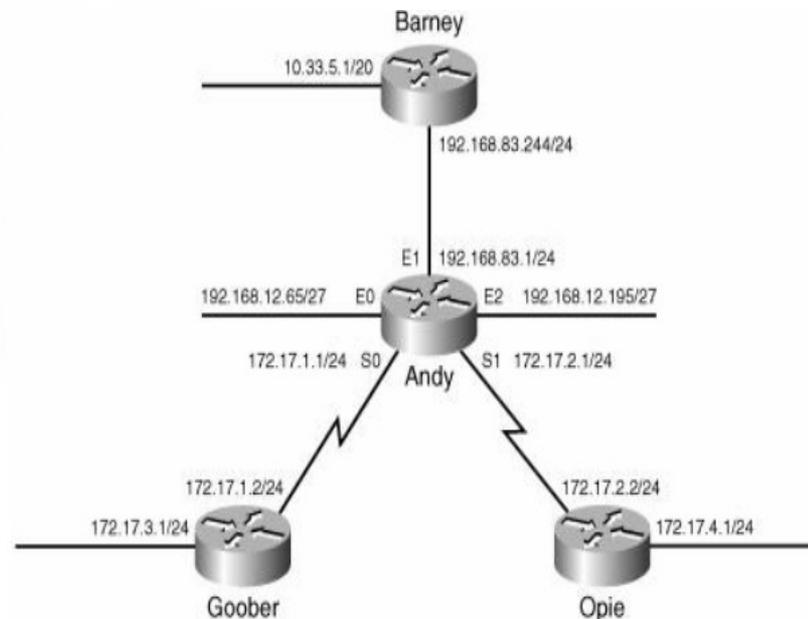


Configurando o RIP (cont.)

```
Goober(config)#router rip  
Goober(config-router)#network 172.17.0.0
```

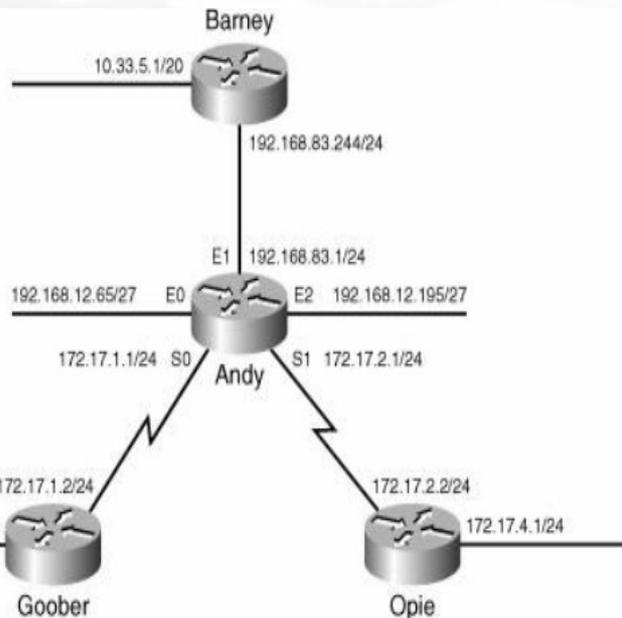
```
Opie(config)#router rip  
Opie(config-router)#network 172.17.0.0
```

```
Barney(config)#router rip  
Barney(config-router)#network 10.0.0.0  
Barney(config-router)#network 192.168.83.0
```



Debugging RIP

(Observar Split Horizon e Summarization)



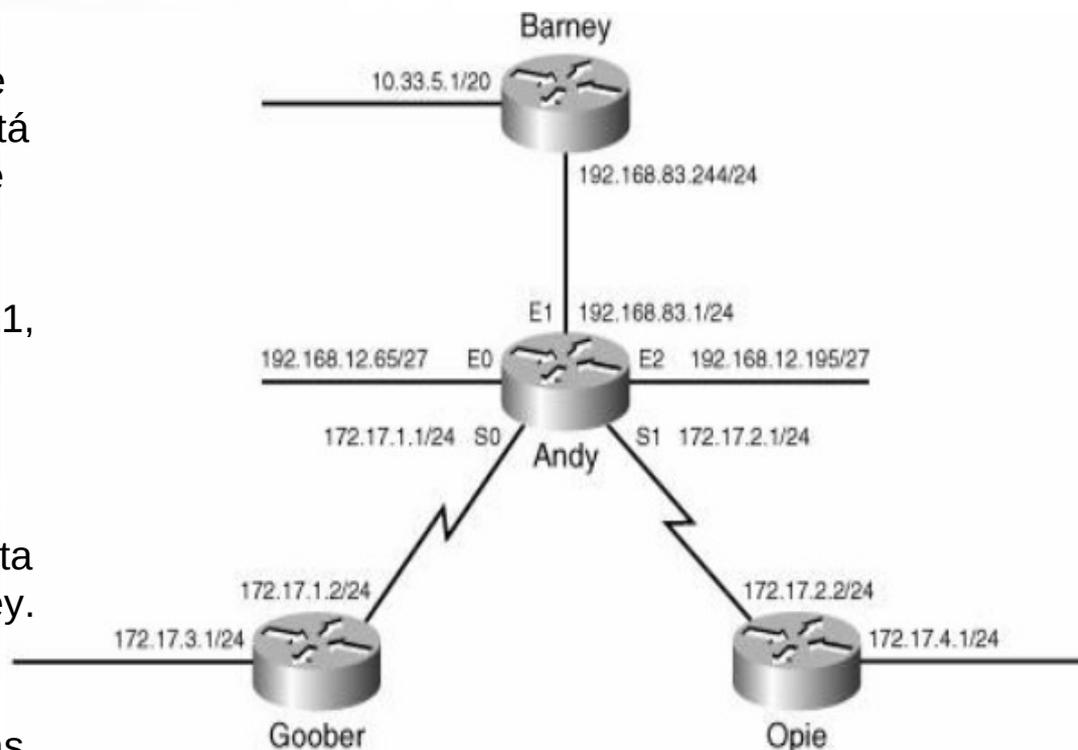
```

Andy#debug ip rip
RIP protocol debugging is on
Andy#
RIP: sending v1 update to 255.255.255.255 via Ethernet0 (192.168.12.65)
RIP: build update entries
  subnet 192.168.12.192, metric 1
  network 10.0.0.0, metric 2
  network 192.168.83.0, metric 1
  network 172.17.0.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Ethernet1 (192.168.83.1)
RIP: build update entries
  network 192.168.12.0, metric 1
  network 172.17.0.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Ethernet2 (192.168.12.195)
RIP: build update entries
  subnet 192.168.12.64, metric 1
  network 10.0.0.0, metric 2
  network 192.168.83.0, metric 1
  network 172.17.0.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.17.1.1)
RIP: build update entries
  subnet 172.17.4.0, metric 2
  subnet 172.17.2.0, metric 1
  network 10.0.0.0, metric 2
  network 192.168.83.0, metric 1
  network 192.168.12.0, metric 1
RIP: sending v1 update to 255.255.255.255 via Serial1 (172.17.2.1)
RIP: build update entries
  subnet 172.17.1.0, metric 1
  subnet 172.17.3.0, metric 2
  network 10.0.0.0, metric 2
  network 192.168.83.0, metric 1
  network 192.168.12.0, metric 1
RIP: received v1 update from 172.17.1.2 on Serial0
  172.17.3.0 in 1 hops
RIP: received v1 update from 192.168.83.244 on Ethernet1
  10.0.0.0 in 1 hops
RIP: received v1 update from 172.17.2.2 on Serial1
  172.17.4.0 in 1 hops
  
```

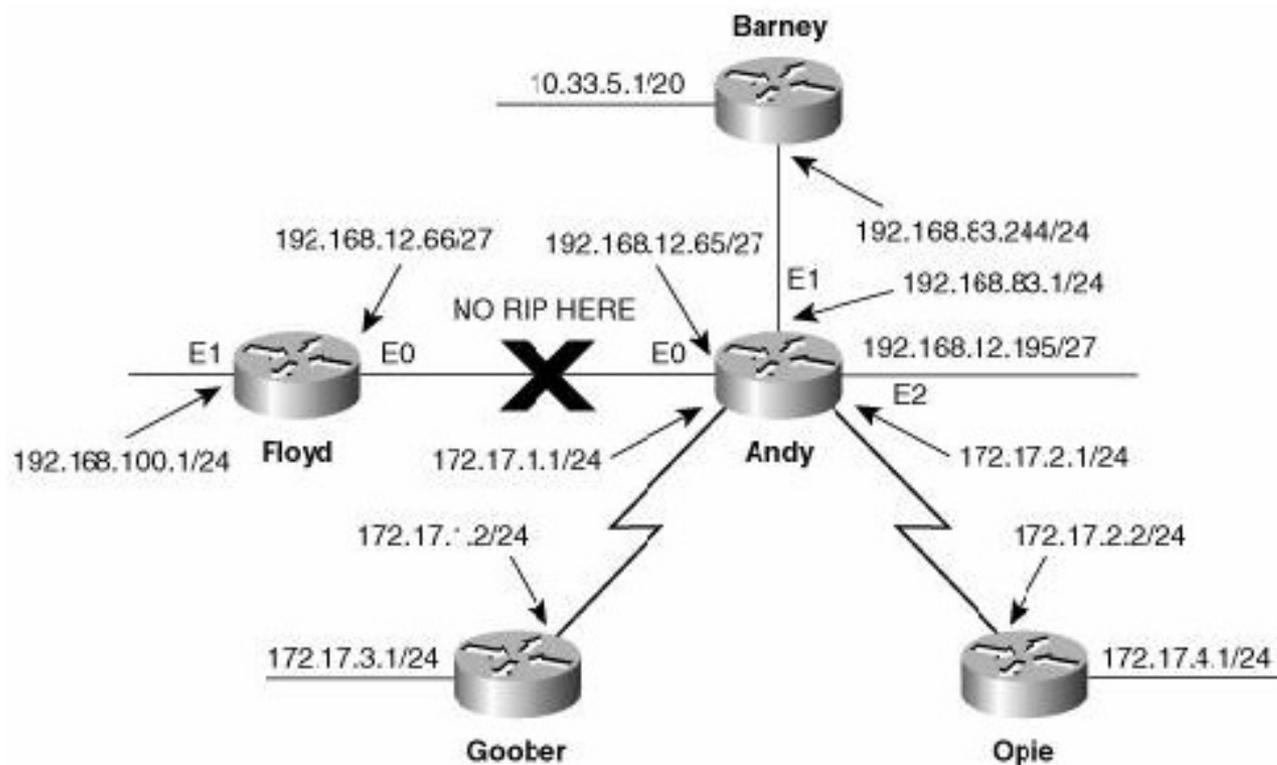
Debugging RIP (cont.)

(Observar Split Horizon e Summarization)

No exemplo, o comando *debug ip rip* foi acionado em Andy. De particular interesse aqui é o “subnet hiding” que o roteador está executando. As subredes 192.168.12.64 e 192.168.12.192 são anunciadas entre as interfaces E0 e E1, ambas ligadas à rede 192.168.12.0, mas são sumarizadas em E1, S0, e S1, todas essas ligadas a diferentes redes. Do mesmo modo, as redes 192.168.83.0 e 172.17.0.0 estão sendo sumarizadas na fronteira “classful”. Note também que Andy está recebendo uma rota sumarizada para a rede 10.0.0.0 de Barney. Finalmente, o *split horizon* pode ser aqui observado. Por exemplo, o anúncio para Barney através de E1 não contém entradas para 10.0.0.0 ou para 192.168.83.0.



Interfaces Passivas



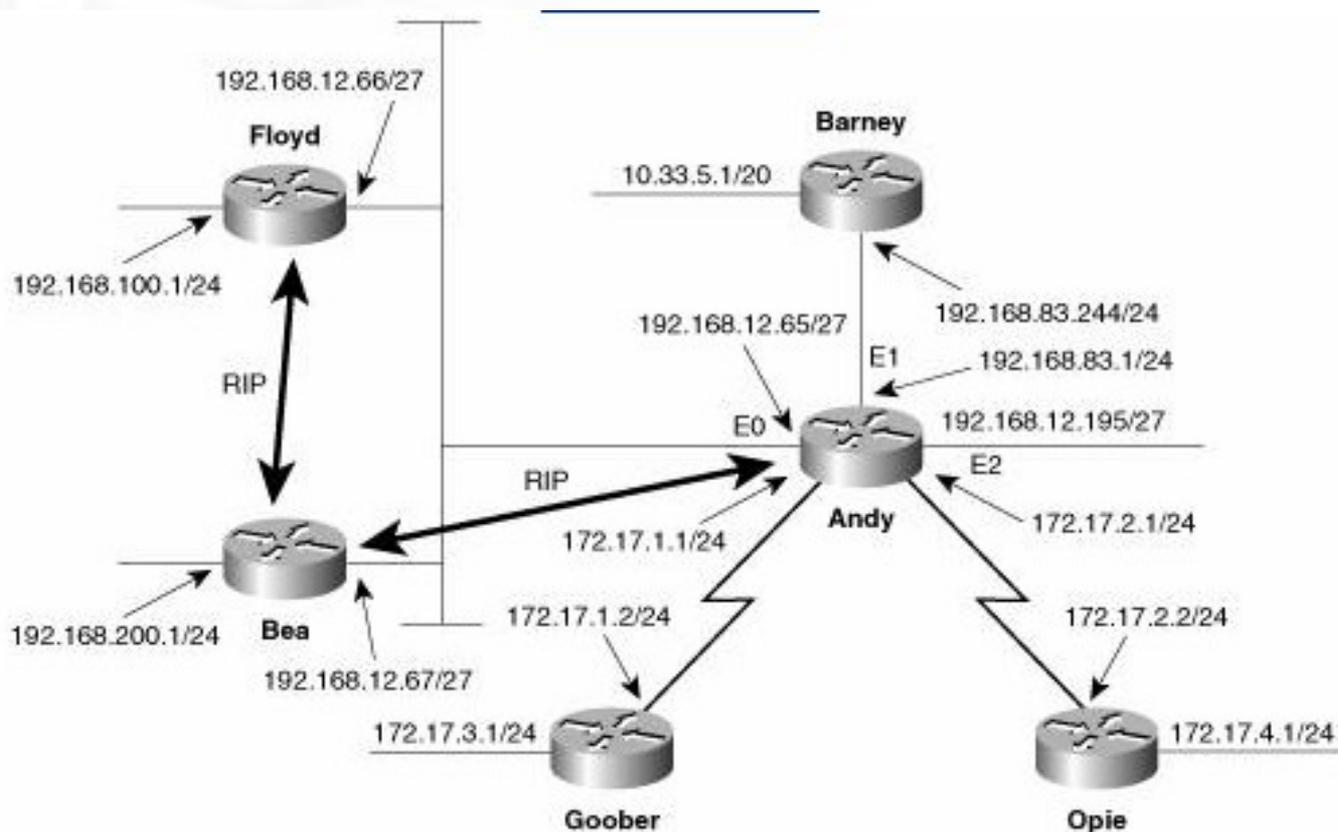
Interfaces Passivas (cont.)

- É desejado que não haja tráfego RIP entre Floyd e Andy.
- Em FLoyd:
 - Basta liberar o RIP apenas na rede 192.168.12.100, bloqueando o broadcast na rede 192.168.12.64
- Em Andy:
 - Como ele tem duas interfaces ligadas à rede 192.168.12.0, esta rede não pode ser bloqueada.
 - Para bloquear o broadcast em uma interface conectada a uma sub-rede de uma rede com o RIP ativado, usamos o comando `passive-interface`.
 - Com este comando, o roteador ainda ouve RIP updates, altera sua tabela de rotas de acordo, mas não envia mais o broadcast (age como um “silent host”).
 - Para prevenir um roteador de aprender rotas, deve-se fazer um procedimento mais complexo (“Route Filtering”).

Interfaces Passivas (cont.)

- Em Floyd:
 - `router rip network 192.168.100.0`
- Em Andy:
 - `router rip`
 - `passive-interface Ethernet0`
 - `network 172.17.0.0`
 - `network 192.168.12.0`
 - `network 192.168.83.0`

Configurando Unicast Updates



Configurando Unicast Updates (cont.)

- Agora, o roteador Bea é adicionado e as seguintes condições são estabelecidas:
 - Floyd e Andy continuam não podendo trocar informações de RIP.
 - Bea e Andy podem trocar informações de roteamento.
 - Bea e Floyd também trocam RIP *updates*.
- Para este cenário, é necessário adicionar o comando `neighbor`. O comando `neighbor` permite ao RIP enviar updates *unicast* de rotas para um dado roteador (i.e., uma interface dele) mesmo com o comando `passive-interface` ainda prevenindo o broadcast de updates no link.

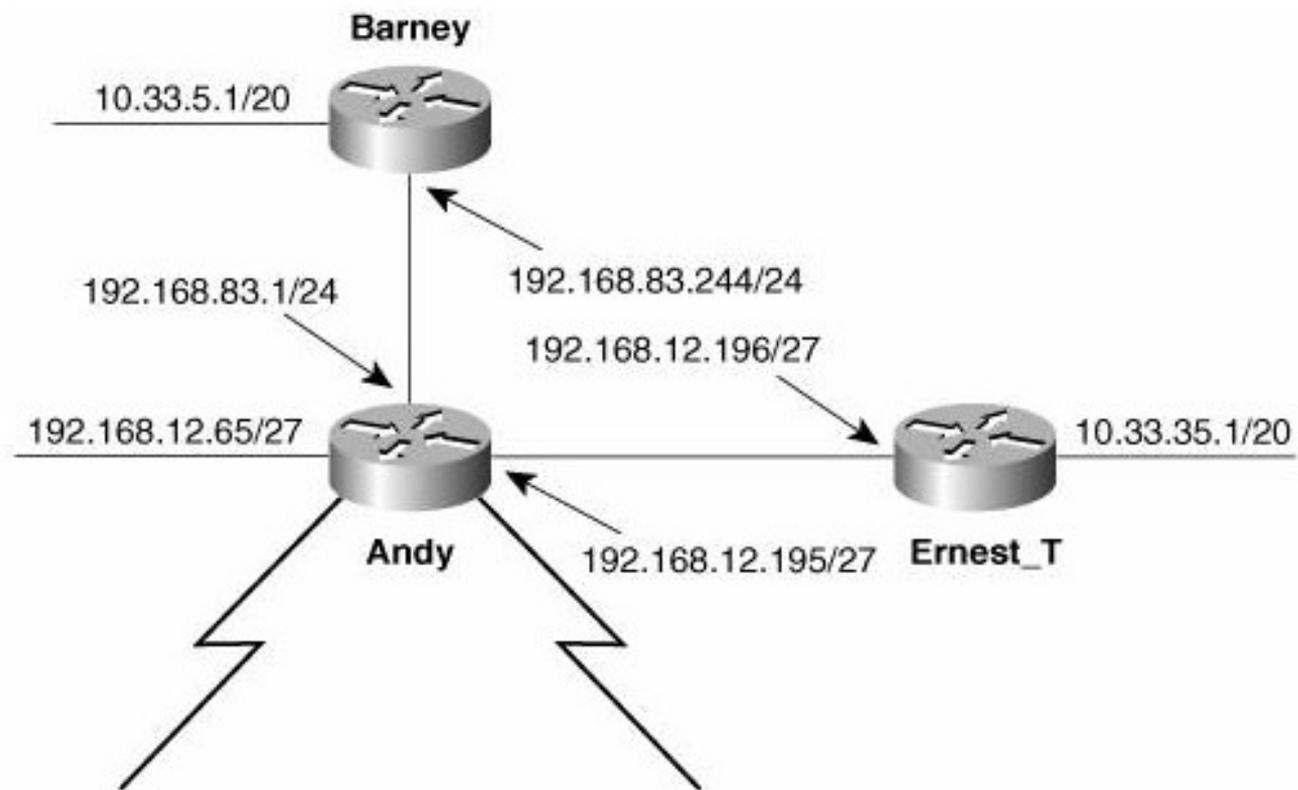
Interfaces Passivas (cont.)

- Em Bea:
 - router rip
 - network 192.168.12.0
 - network 192.168.200.0
- Em Andy:
 - router rip
 - passive-interface Ethernet0
 - network 172.17.0.0
 - network 192.168.12.0
 - network 192.168.83.0
 - neighbor 192.168.12.67
- Em Floyd:
 - router rip
 - passive-interface Ethernet0
 - network 192.168.12.0
 - network 192.168.100.0
 - neighbor 192.168.12.67

Configurando Unicast Updates (cont.)

```
Andy#debug ip rip events
RIP event debugging is on
Andy#
RIP: received v1 update from 192.168.12.67 on Ethernet0
RIP: Update contains 1 routes
RIP: sending v1 update to 255.255.255.255 via Ethernet1 (192.168.83.1)
RIP: Update contains 4 routes
RIP: Update queued
RIP: Update sent via Ethernet1
RIP: sending v1 update to 255.255.255.255 via Ethernet2 (192.168.12.195)
RIP: Update contains 6 routes
RIP: Update queued
RIP: Update sent via Ethernet2
RIP: sending v1 update to 255.255.255.255 via Serial0 (172.17.1.1)
RIP: Update contains 7 routes
RIP: Update queued
RIP: Update sent via Serial0
RIP: sending v1 update to 255.255.255.255 via Serial1 (172.17.2.1)
RIP: Update contains 7 routes
RIP: Update queued
RIP: Update sent via Serial1
RIP: sending v1 update to 192.168.12.67 via Ethernet0 (192.168.12.65)
RIP: Update contains 4 routes
RIP: Update queued
RIP: Update sent via Ethernet0
RIP: received v1 update from 172.17.1.2 on Serial0
RIP: Update contains 1 routes
RIP: received v1 update from 172.17.2.2 on Serial1
RIP: Update contains 1 routes
RIP: received v1 update from 192.168.12.67 on Ethernet0
RIP: Update contains 1 routes
```

Redes Não Contíguas



Redes Não Contíguas (cont.)

- O roteador Ernest é adicionado à internet. Uma das suas interfaces usa um endereço de sub-rede da rede 10.33.0.0.
- Problema:
 - como as sub-redes de 10.33.0.0 não são contíguas, Barney e Ernest vão ambos divulgar uma rota sumarizada para 10.0.0.0. Com isso, Andy pensará que tem dois caminhos de igual custo para a rede 10.0.0.0.
- Solução:
 - Configurar sub-redes da rede 10.0.0.0 nos links 192.168.83.0/24 e 192.168.12.192/27, usando o conceito de *endereço ip secundário*.

Redes Não Contíguas (cont.)

Example 5-15. Barney is configured with secondary IP addresses.

```
interface e0
 ip address 10.33.55.1 255.255.240.0 secondary
```

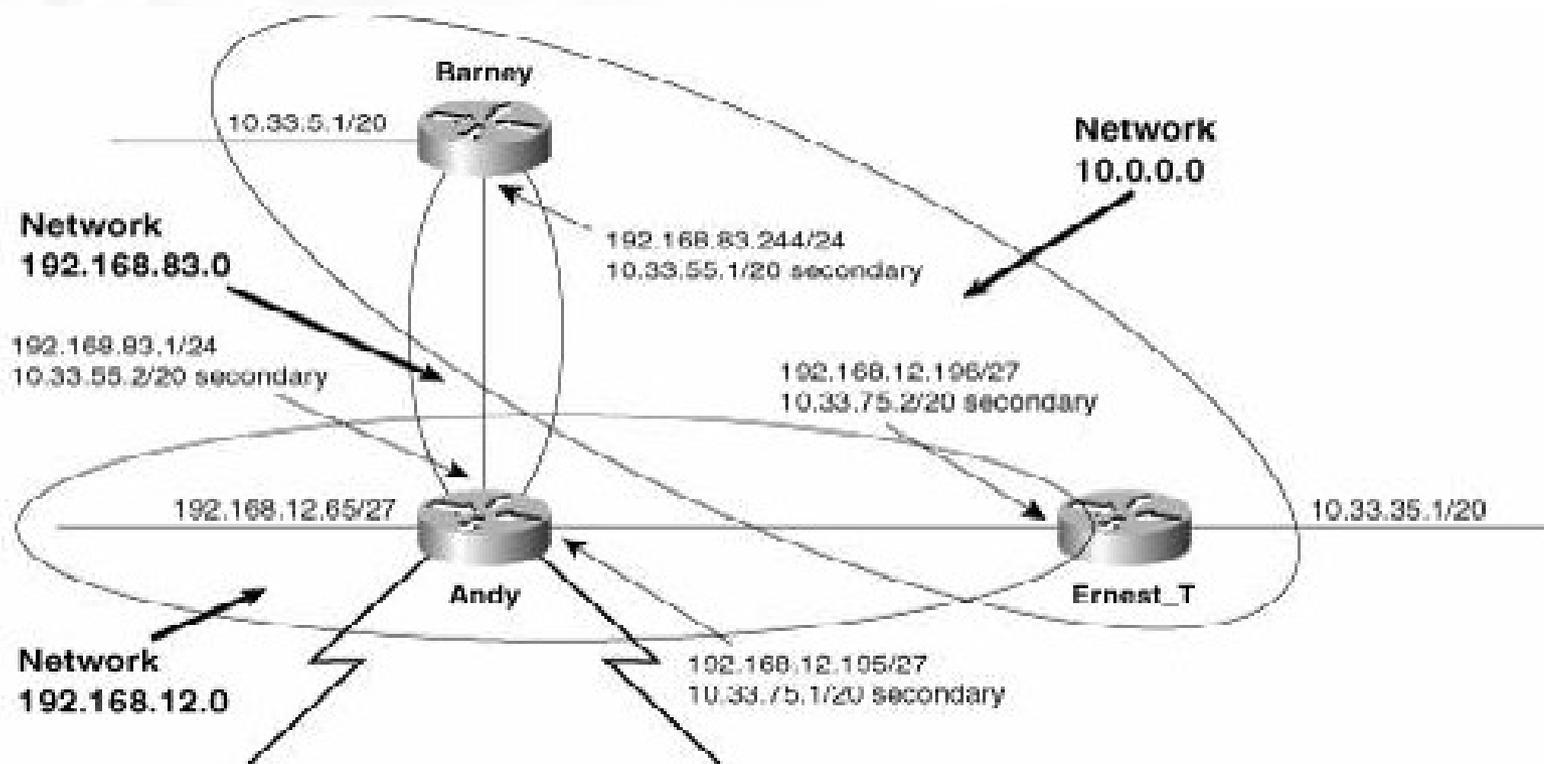
Example 5-16. Andy is configured with secondary IP addresses, and a new network is added to RIP.

```
interface e1
 ip address 10.33.55.2 255.255.240.0 secondary
interface e2
 ip address 10.33.75.1 255.255.240.0 secondary
router rip
 network 10.0.0.0
```

Example 5-17. Ernest_T is configured with secondary IP addresses.

```
interface e0
 ip address 10.33.75.2 255.255.240.0 secondary
```

Redes Não Contíguas (cont.)



Redes Não Contíguas (cont.)

```
Ernest_T#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

```
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
```

```
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
```

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

```
 10.0.0.0 255.255.240.0 is subnetted, 4 subnets
```

```
 C    10.33.32.0 is directly connected, Ethernet1
```

```
 R    10.33.48.0 [120/1] via 10.33.75.1, 00:00:05, Ethernet0
```

```
 R    10.33.0.0 [120/2] via 10.33.75.1, 00:00:05, Ethernet0
```

```
 C    10.33.64.0 is directly connected, Ethernet0
```

```
 R    192.168.83.0 [120/1] via 192.168.12.195, 00:00:05, Ethernet0
```

```
        [120/1] via 10.33.75.1, 00:00:05, Ethernet0
```

```
 192.168.12.0 255.255.255.224 is subnetted, 2 subnets
```

```
 R    192.168.12.64 [120/1] via 192.168.12.195, 00:00:05, Ethernet0
```

```
 C    192.168.12.192 is directly connected, Ethernet0
```

```
 R    192.168.200.0 [120/2] via 192.168.12.195, 00:00:05, Ethernet0
```

```
        [120/2] via 10.33.75.1, 00:00:05, Ethernet0
```

```
 R    172.17.0.0 [120/1] via 192.168.12.195, 00:00:05, Ethernet0
```

```
        [120/1] via 10.33.75.1, 00:00:05, Ethernet0
```

```
Ernest_T#
```