

Administração de Redes 2019/20

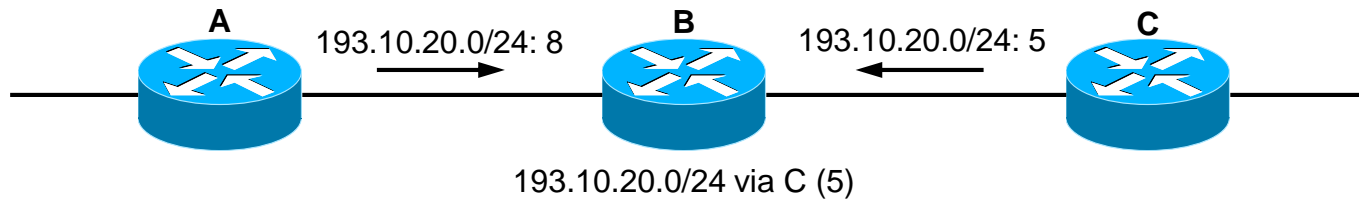
Route Information Protocol (RIP)

Route Information Protocol (RIP)

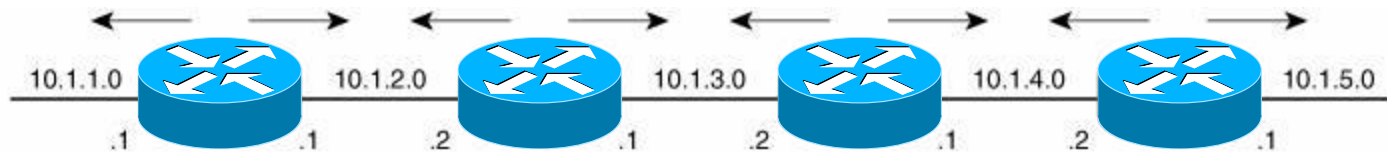
- Protocolo de encaminhamento de vetores de distâncias (DV)
 - Vetores enviados a cada 30 segundos (anúncios periódicos)
 - Encapsulados em segmentos UDP, porta 520
- Métrica = número de saltos
 - Valor máximo = 15
 - Métrica 16 considerada infinita (destino inatingível)
- Tempo de convergência longo
- Escalabilidade limitada
 - Apto apenas para redes de pequenas dimensões
- Versão original com classes, versão 2 sem classes

Protocolos de Vectores de Distâncias

- Inicialmente, cada *router* conhece apenas as sub-redes a que está directamente ligado
- Periodicamente, cada *router* anuncia aos seus vizinhos os destinos que conhece e o respectivo custo
 - Inicialmente apenas as suas sub-redes
 - Depois também os destinos que aprendeu através de anúncios de vizinhos
 - As mensagens são enviadas em *broadcast* ou *multicast*
 - Mais eficiente, e não exige que o *router* conheça os vizinhos
- Entre todos os anúncios recebidos para um dado destino, o *router* escolhe o de menor custo
 - Encaminhamento por rumor

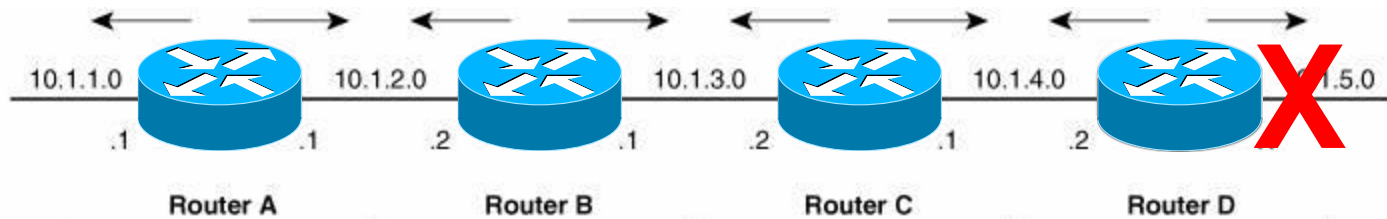


Encaminhamento por rumor



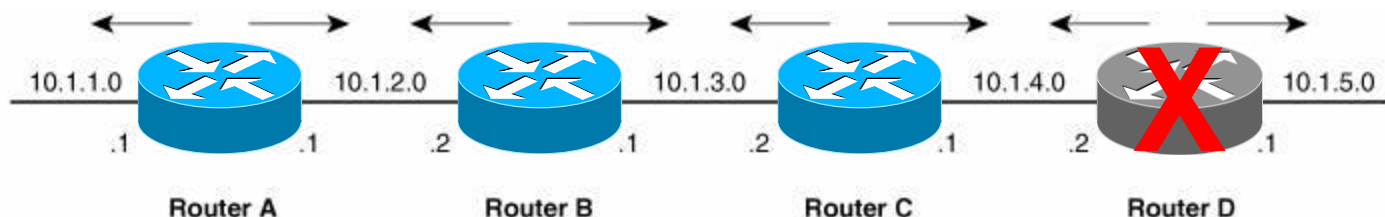
	Router A			Router B			Router C			Router D		
	NET	VIA	HOPS	NET	VIA	HOPS	NET	VIA	HOPS	NET	VIA	HOPS
t_0	10.1.1.0	--	0	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0
	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0	10.1.5.0	--	0
t_1	10.1.1.0	--	0	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0
	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0	10.1.5.0	--	0
	10.1.3.0	10.1.2.2	1	10.1.1.0	10.1.2.1	1	10.1.2.0	10.1.3.1	1	10.1.3.0	10.1.4.1	1
	10.1.4.0	10.1.3.2	1	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.4.2	1			
t_2	10.1.1.0	--	0	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0
	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0	10.1.5.0	--	0
	10.1.3.0	10.1.2.2	1	10.1.1.0	10.1.2.1	1	10.1.2.0	10.1.3.1	1	10.1.3.0	10.1.4.1	1
	10.1.4.0	10.1.2.2	2	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.4.2	1	10.1.2.0	10.1.4.1	2
	10.1.5.0	10.1.3.2	2	10.1.5.0	10.1.3.2	2	10.1.1.0	10.1.3.1	2			
t_3	10.1.1.0	--	0	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0
	10.1.2.0	--	0	10.1.3.0	--	0	10.1.4.0	--	0	10.1.5.0	--	0
	10.1.3.0	10.1.2.2	1	10.1.1.0	10.1.2.1	1	10.1.2.0	10.1.3.1	1	10.1.3.0	10.1.4.1	1
	10.1.4.0	10.1.2.2	2	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.4.2	1	10.1.2.0	10.1.4.1	2
	10.1.5.0	10.1.2.2	3	10.1.5.0	10.1.3.2	2	10.1.1.0	10.1.3.1	2	10.1.1.0	10.1.4.1	3

Envenenamento de rota (route poisoning)



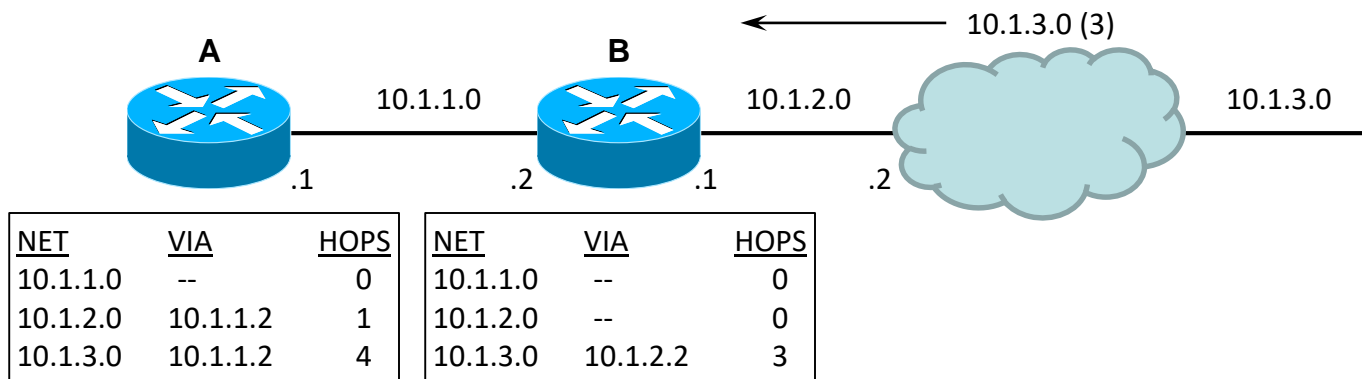
- Se a rede 10.1.5.0 ficar inacessível, D anuncia-a como inatingível
 - Envenenamento de rota (*route poisoning*)
 - Consiste em anunciar a rota com custo infinito (16)
- O envenenamento de rotas acelera o processo de convergência

Temporizador de invalidação de rotas



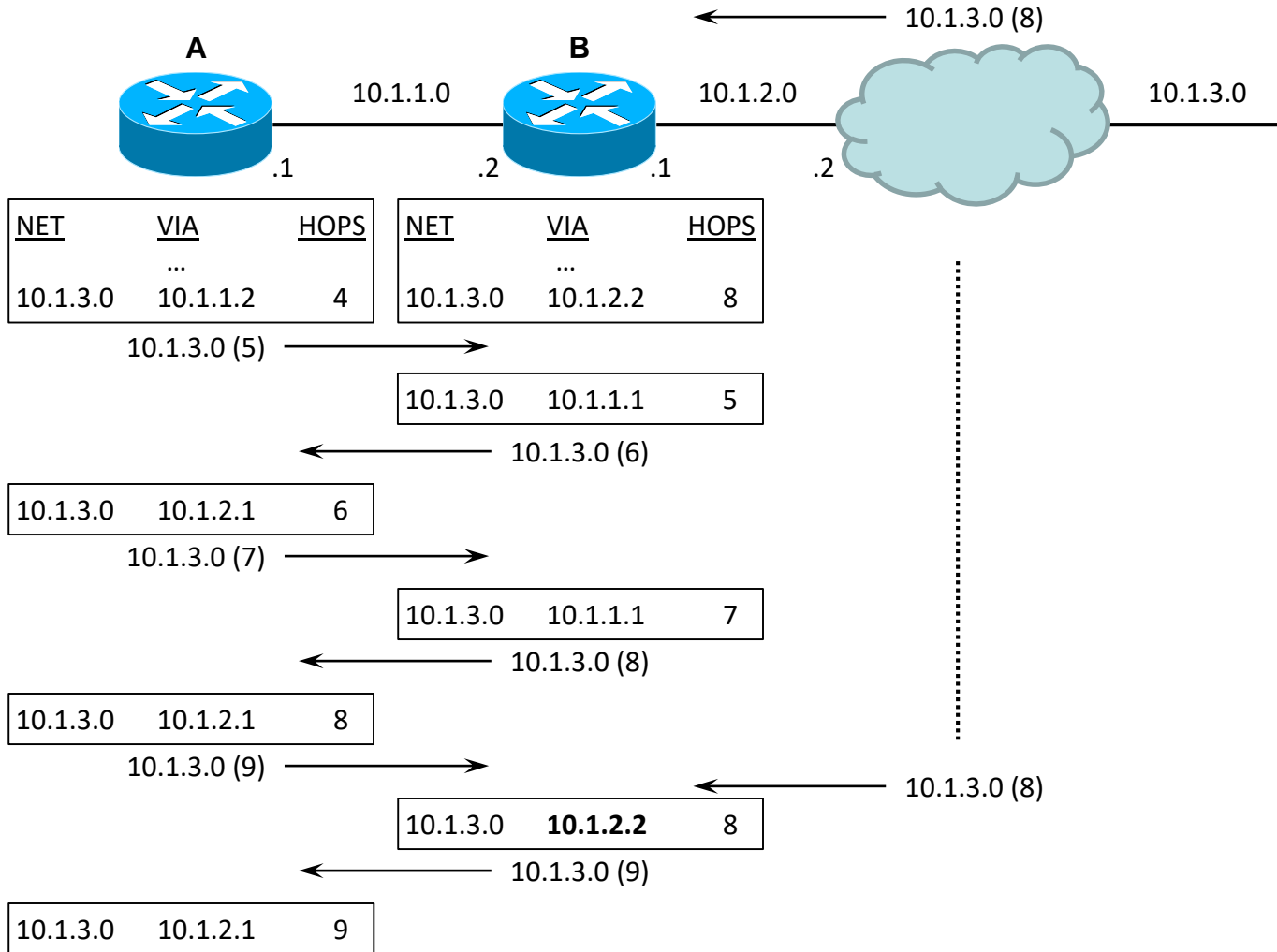
- Se o router D falhar, C não recebe notificação de destino inatingível
 - Routers "mortos" não enviam mensagens...
- Temporizador de invalidação associado a cada rota
 - Se passar esse tempo sem receber novo anúncio, o router marca o destino como inválido
 - Passa essa informação aos vizinhos (route poisoning)
 - Temporizador de invalidação tem que ser maior que o tempo entre anúncios
 - É possível que um anúncio se perca devido a erros e/ou congestionamento
 - Valor típico: seis vezes o tempo entre anúncios (180 seg. no RIP)

Efeito de ressalto

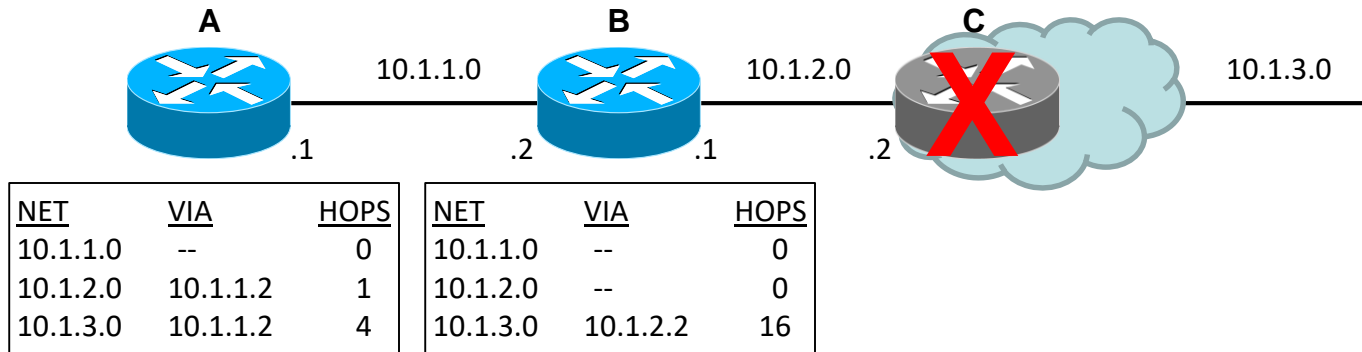


- Caminho de B até 10.1.3.0 aumenta de 3 para 8 saltos
- Antes de B conseguir enviar novo anúncio a A, recebe o anúncio deste com custo 5
 - O anúncio de B também pode perder-se
- B passa a usar A como próximo salto para 10.1.3.0
- B anuncia 10.1.3.0 com 6 saltos
- A anuncia 10.1.3.0 com 7 saltos ...
- Ciclo de encaminhamento até voltar a convergir!!!

Efeito de ressalto



Contagem até ao infinito

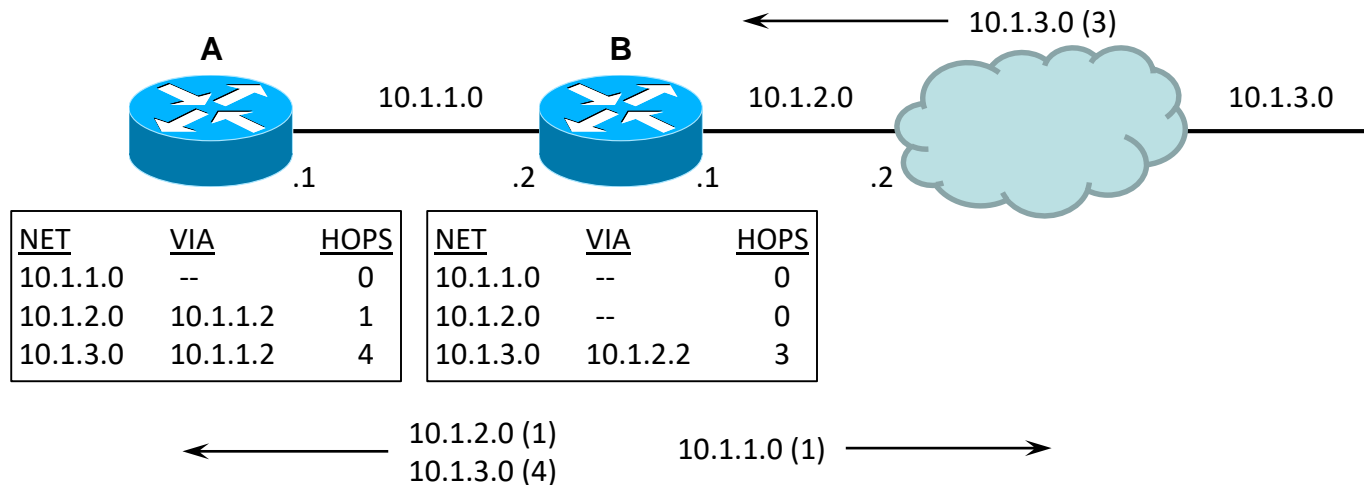


- Se em vez de um aumento de custo houver falha do *router C*
- B detecta falha na rota (temporizador de invalidação)
- B recebe de A anúncio da rota com custo 5
 - Passa a usar A como próximo salto
- Efeito de ressalto continua até se atingir métrica infinita
 - Razão para o RIP usar um valor de "infinito" tão baixo (16)
- Contudo, um valor baixo de infinito
 - Limita o diâmetro máximo da rede
 - Limita as possibilidades de manipulação dos custos

Horizonte dividido (*split horizon*)

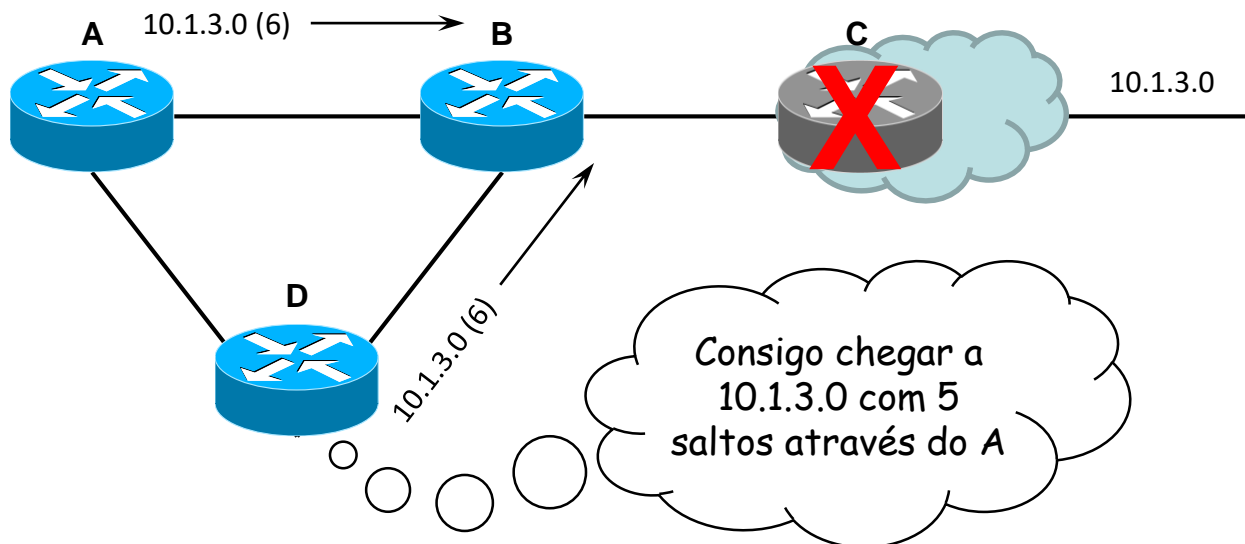
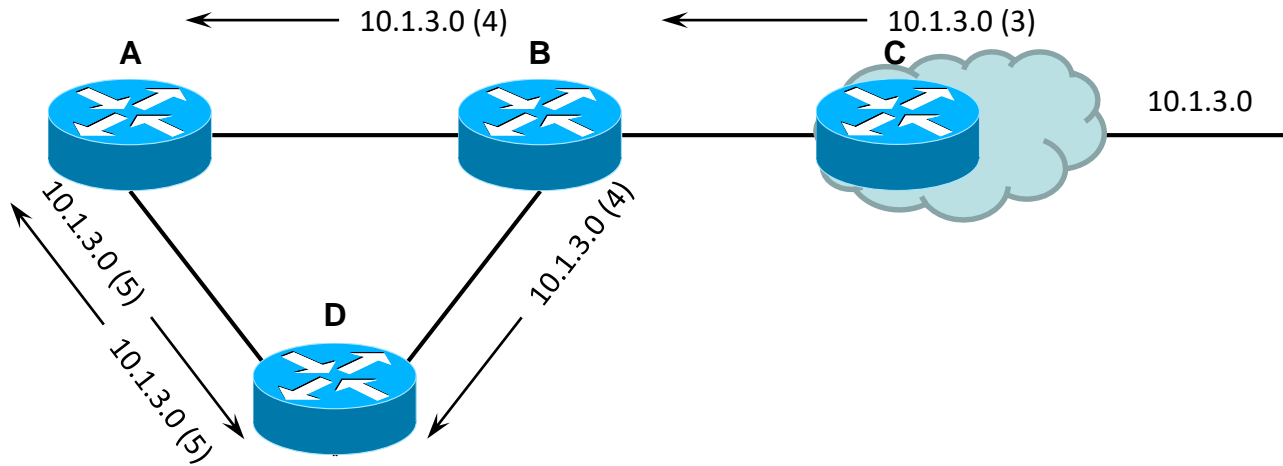
- Não anunciar numa interface as rotas que aprendeu por ela
ou
- Anunciar numa interface as rotas que se aprendeu por ela com custo infinito (16)
 - Horizonte dividido com inverso venenoso (*poisonous reverse*)
 - Mais rápido/eficaz: "é melhor ter más notícias que nenhuma"
- Evita efeito de ressalto / contagem até ao infinito em topologias não emalhadas 😊
- Em topologias emalhadas esses problemas ainda podem ocorrer ☹️

Horizonte dividido (split horizon)



- A não anuncia a B a rota 10.1.3.0
- O único caminho que B pode conhecer para esse destino é através de 10.1.2.2
- Quando B detecta a falha desse caminho, 10.1.3.0 fica inatingível
- A é informado desse facto
- O encaminhamento convergiu

Horizonte dividido (split horizon)



Actualizações instantâneas

- Em Inglês, "*triggered updates*" ou "*flash updates*"
- Anúncios enviados imediatamente após detectar alguma alteração
 - Sem esperar pelo próximo anúncio periódico
- Incluem apenas a(s) rota(s) afectada(s)
- Aceleram o processo de convergência

Temporizador de supressão

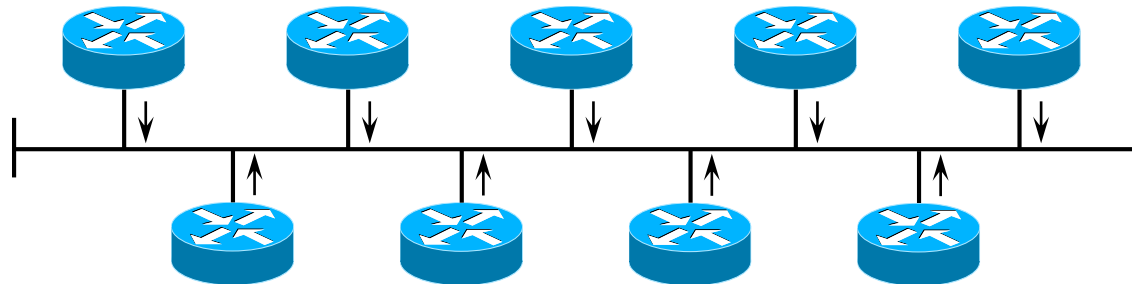
- Em Inglês, "*holddown timer*"
- Quando um destino é dado como inatingível, o *router* deixa de acreditar em anúncios desse destino durante um certo tempo
- Previne o uso de informação espúria durante o processo de reconvergência
- Duração do temporizador é um compromisso:
 - Demasiado curta torna-o ineficaz
 - Demasiado longa aumenta o tempo de reconvergência
 - Também causa problemas em redes com falhas intermitentes
- Valor típico: 180 s

Temporizador de descarga

- Em Inglês, "*flush timer*"
- Conta o tempo desde que foi recebido o último anúncio válido para uma rota até ela ser removida da tabela de encaminhamento
- Valor típico: 240 s
 - Superior ao temporizador de invalidação
 - Rota continua a ser usada para reenvio de pacotes mesmo depois de dada como inatingível para o protocolo de encaminhamento
 - Durante esse período é anunciada com métrica 16 — rota envenenada
 - Após expirar o temporizador de descarga, a rota deixa de ser anunciada

Assincronia nos anúncios

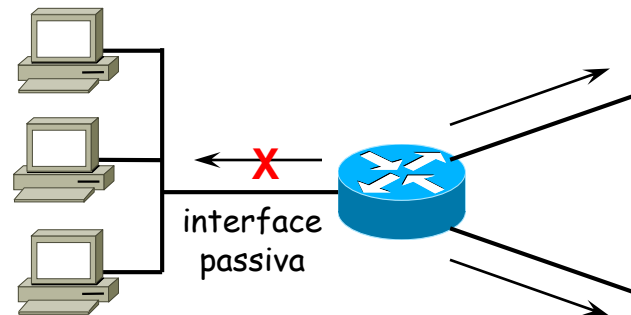
- Verificou-se que, mesmo começando em instantes diferentes, os *routers* tendiam a enviar anúncios ao mesmo tempo
 - Devido aos tempos de processamento
- Isto pode causar perda de anúncios devido a colisões:



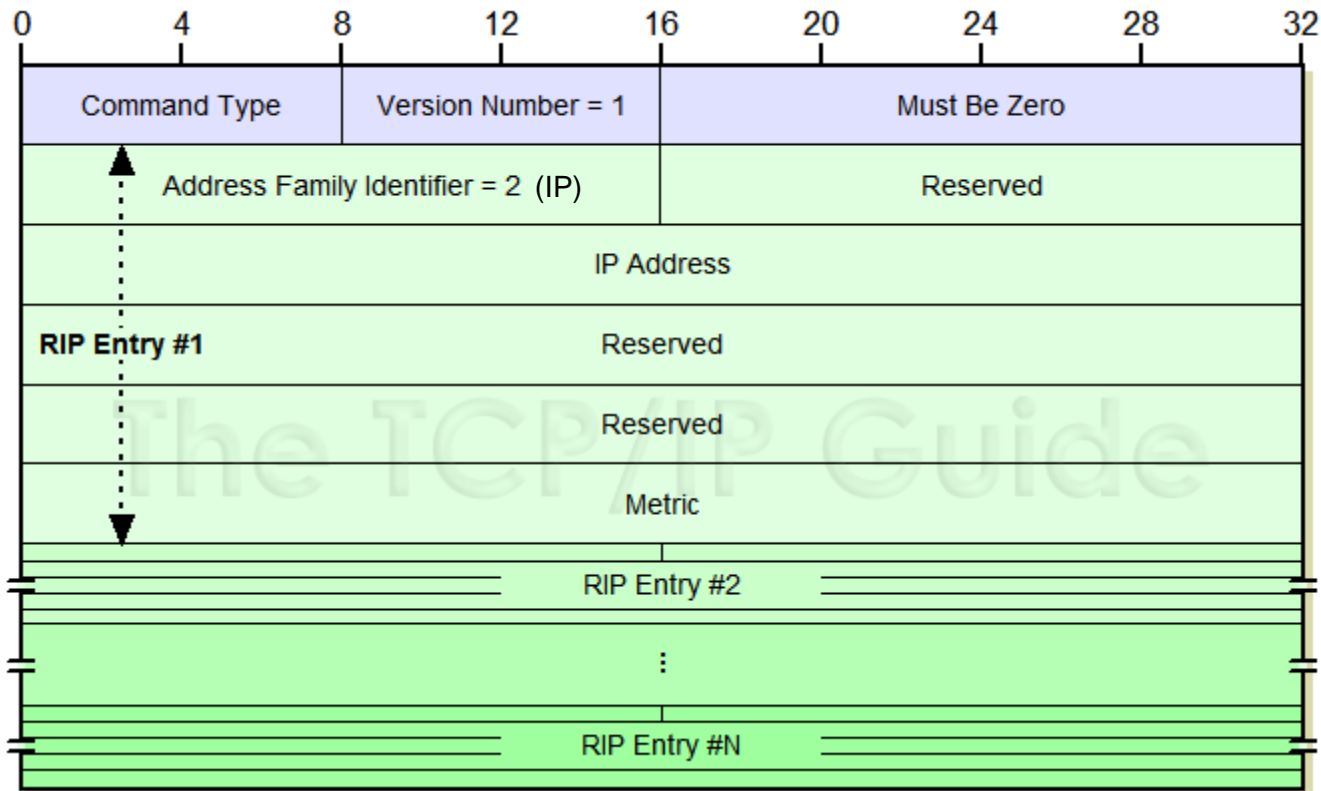
- Duas soluções:
 - Temporizadores independentes do processamento
 - Variação aleatória (*timing jitter*) no instante de envio de anúncio
 - Nos *routers Cisco*, o período varia entre 25.5 e 30 s

Interfaces passivas

- Em ligações só com terminais (pontas) não faz sentido enviar actualizações
 - Terminais não "falam" protocolos de encaminhamento
- Também pode não se querer enviar actualizações por questões de política de encaminhamento
- Configurando uma interface como passiva, deixam de ser enviados anúncios por ela
 - Mesmo que o seu prefixo pertença aos blocos configurados no protocolo de encaminhamento
- Prefixo da ligação continua a ser anunciado nas outras interfaces



RIP versão 1



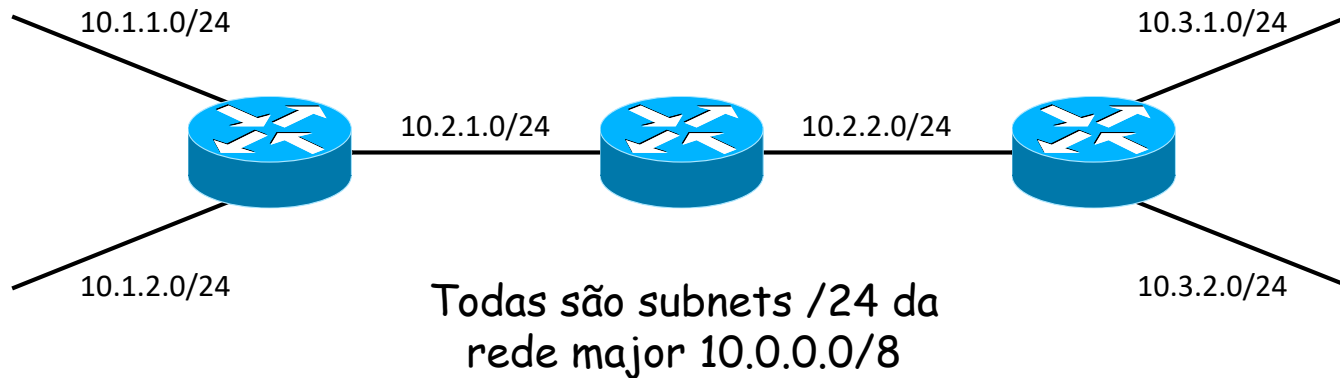
- Até 25 anúncios por mensagem
- Muitos campos vazios (reservados / zero)
 - Para usar em "versões futuras" do protocolo

RIP versão 1

- Mensagens encapsuladas em UDP (porta 520) e enviadas para o endereço de difusão local (255.255.255.255)
- Comandos:
 - Pedido
 - Pode ser para um prefixo específico ou para todos (0.0.0.0)
 - No arranque do protocolo é enviado um pedido para todos
 - Resposta (anúncio)
 - Enviada em resposta a um pedido ou de forma espontânea
- Sem campo para a máscara — protocolo com classes
 - Máscara inferida dos primeiros bits do prefixo
- Métrica (custo) entre 1 e 16
 - Bastavam 4 bits, mas são usados 32...

Subnetting com RIPv1

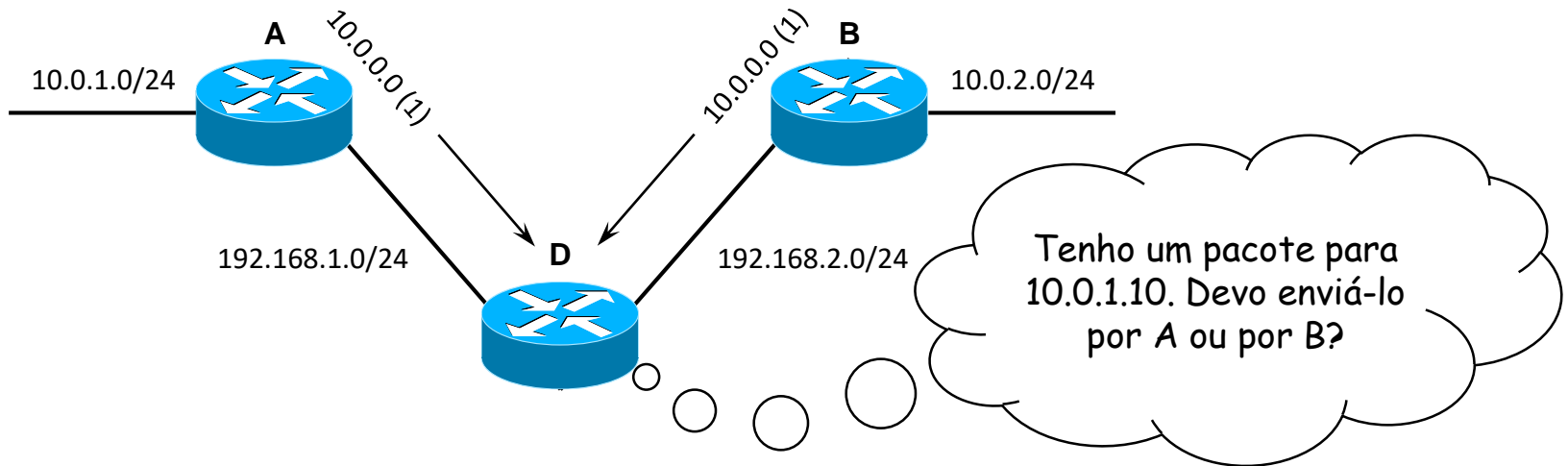
- Suporte para subnetting muito limitado — subnets têm que
 - Ser contíguas
 - Ter todas o mesmo comprimento de prefixo



- Máscara de rede para anúncios recebidos
 - A mesma da interface onde o anúncio foi recebido, se esta pertencer à mesma rede major
 - A da classe a que pertence o prefixo anunciado, caso contrário

Subnetting com RIPv1

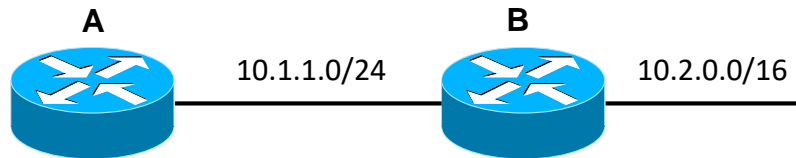
Problema com subnets não-contíguas:



- D recebe dois anúncios para 10.0.0.0 com o mesmo custo
 - Assume que são caminhos de igual custo para o mesmo destino
- Dependendo do tipo de ECMP e dos endereços, pacotes podem
 - Chegar todos
 - Não chegar nenhum
 - Chegar metade (pacote sim, pacote não)

Subnetting com RIPv1

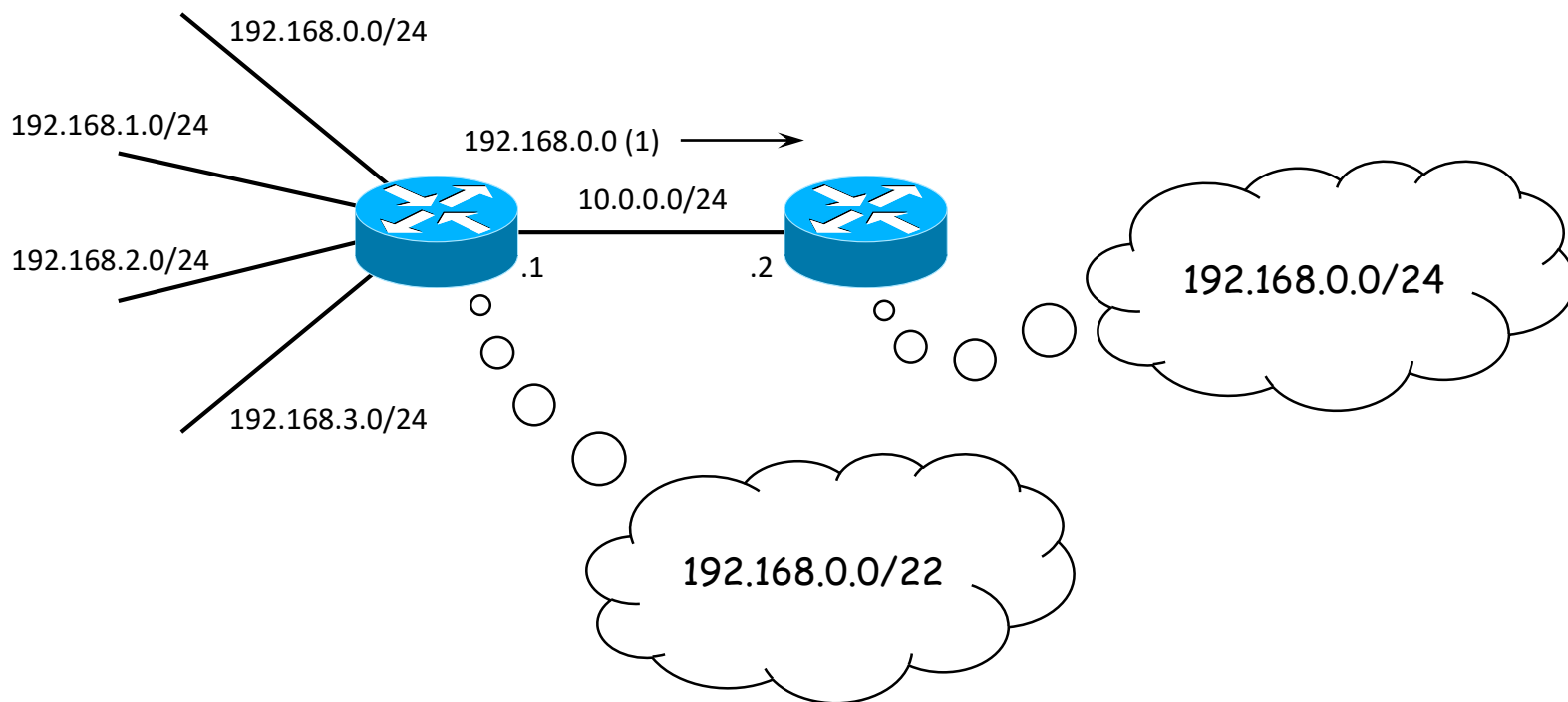
Problema com subnetting de comprimento variável:



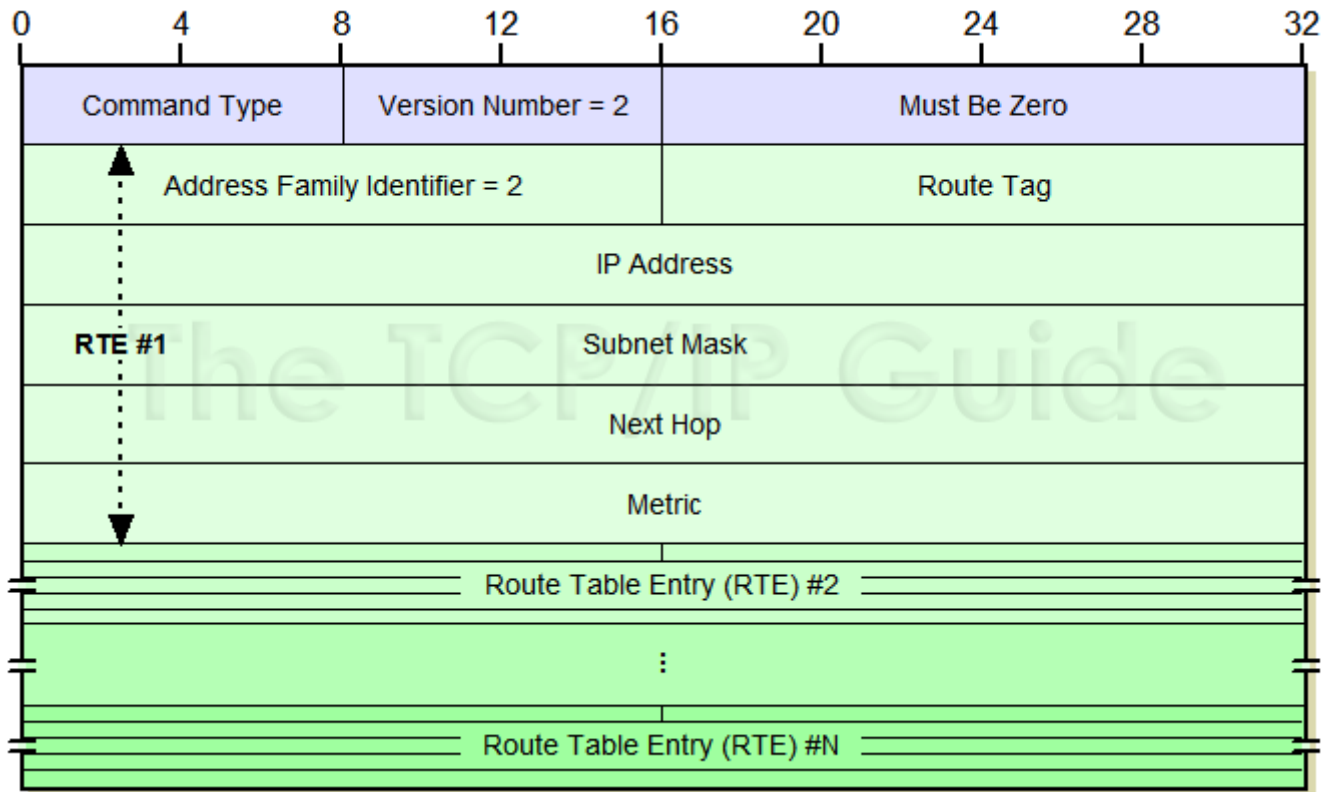
- B não anuncia 10.2.0.0 a A por ter máscara diferente
 - A não consegue enviar pacotes para 10.2.x.y
- Mesmo que B anunciasse 10.2.0.0, A interpretaria o anúncio como sendo para 10.2.0.0/24
 - A conseguiria enviar pacotes para 10.2.0.x
 - Mas não para 10.2.y.z (com $y \neq 0$)

Impossibilidade de supernetting com RIPv1

- Com *supernetting*, os anúncios e a interface onde são recebidos pertencem a redes major diferentes
 - Máscara assumida é a da rede major a que pertence o anúncio
 - Logo, é impossível agregar redes major numa *supernet*



RIP versão 2

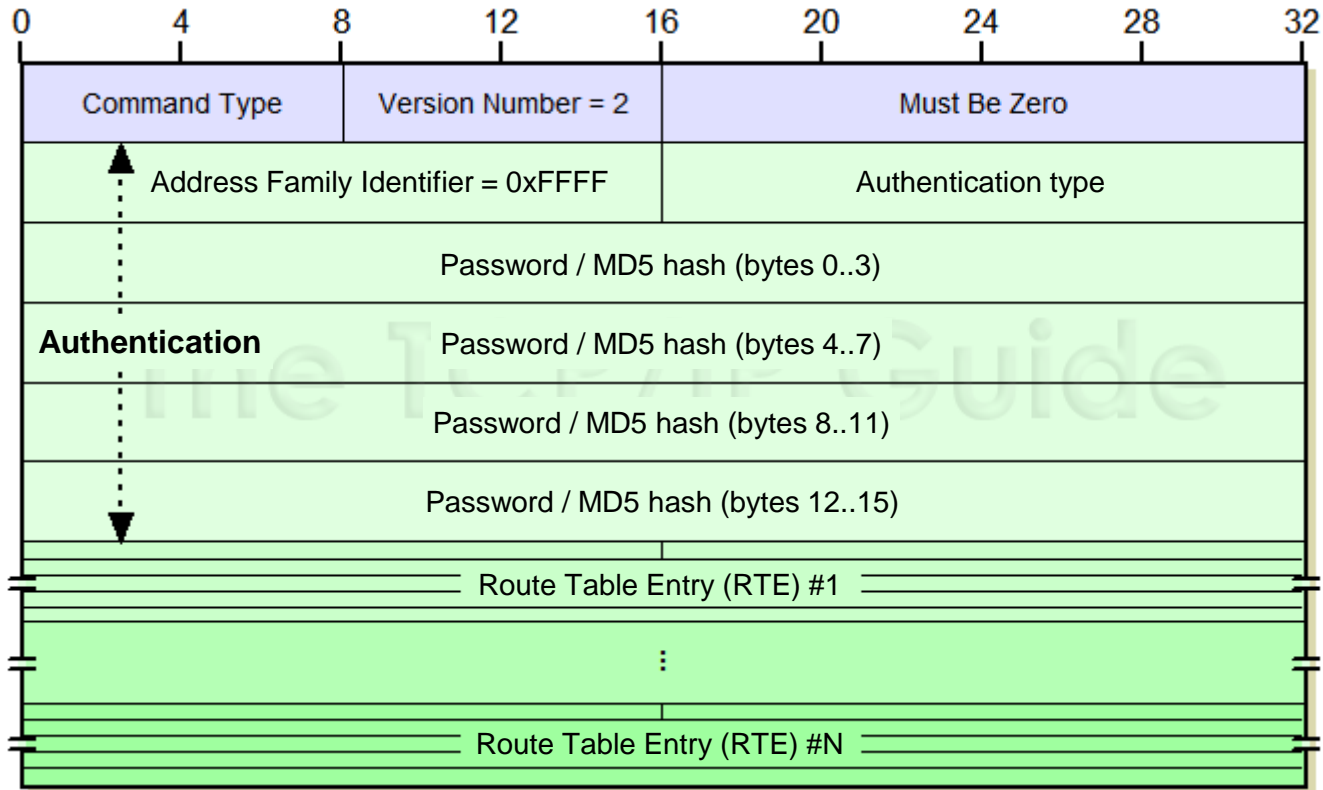


- Campos vazios no RIPv1 passam a ser usados no RIPv2
- Campo para a máscara → protocolo sem classes

RIP versão 2

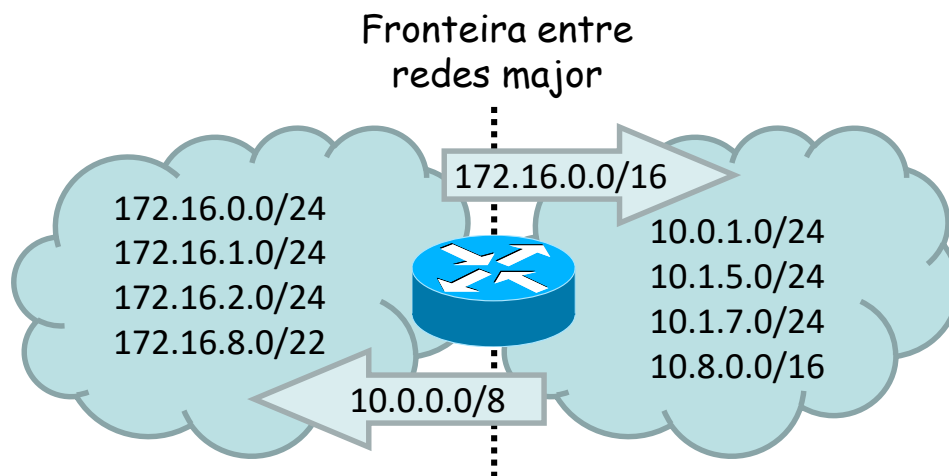
- Mensagens enviadas para 224.0.0.9
 - Endereço *multicast* "todos os routers RIPv2"
 - Evita que os terminais recebam os anúncios (que não lhes interessam)
- Possível especificar próximo salto diferente do anunciante
 - Útil se esse próximo salto não "falar" RIP
- Possível etiquetar rotas externas ou redistribuídas
 - E.g., com o número do sistema autónomo de onde foram recebidas
- Suporta autenticação das mensagens (opcional)
 - Consome uma das 25 entradas possíveis
 - Evita que mensagens forjadas afectem o encaminhamento
 - Por *password* em *clear text* ou *hash MD5* da mensagem com uma *password*

Autenticação no RIPv2



Sumarização automática

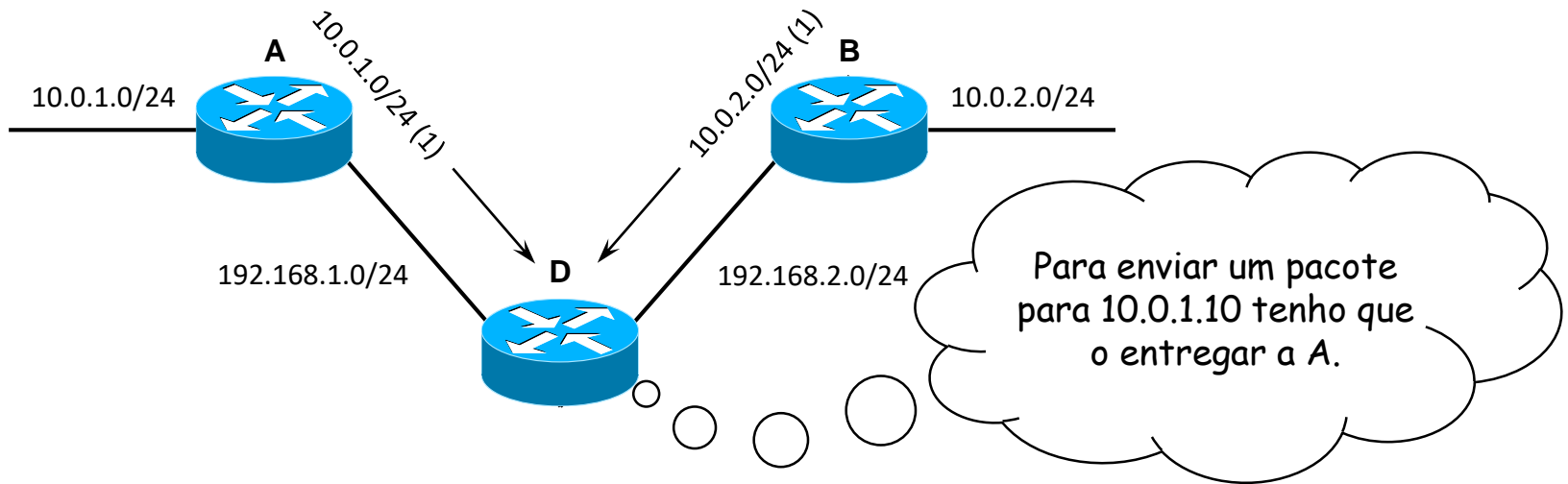
- No Cisco IOS, o RIPv2 vem pré-configurado para sumarizar subnets na fronteira entre redes major
- Pode ser útil para poupar tráfego de encaminhamento



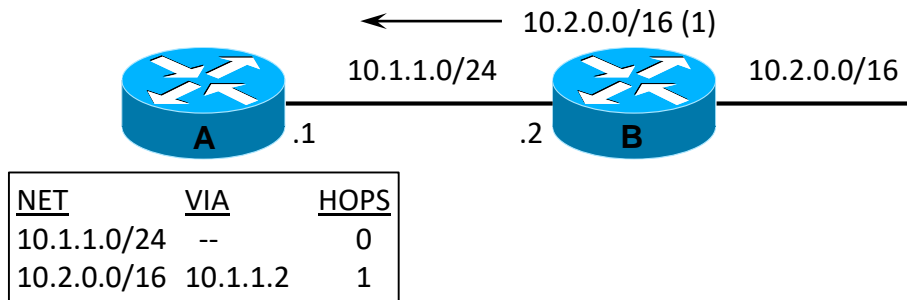
- Para o RIPv2 ser completamente sem classes é necessário desactivá-la
 - Comando no auto-summary

Subnetting com RIPv2

- Suporta subnets não-contíguas
 - Necessário desactivar sumarização automática

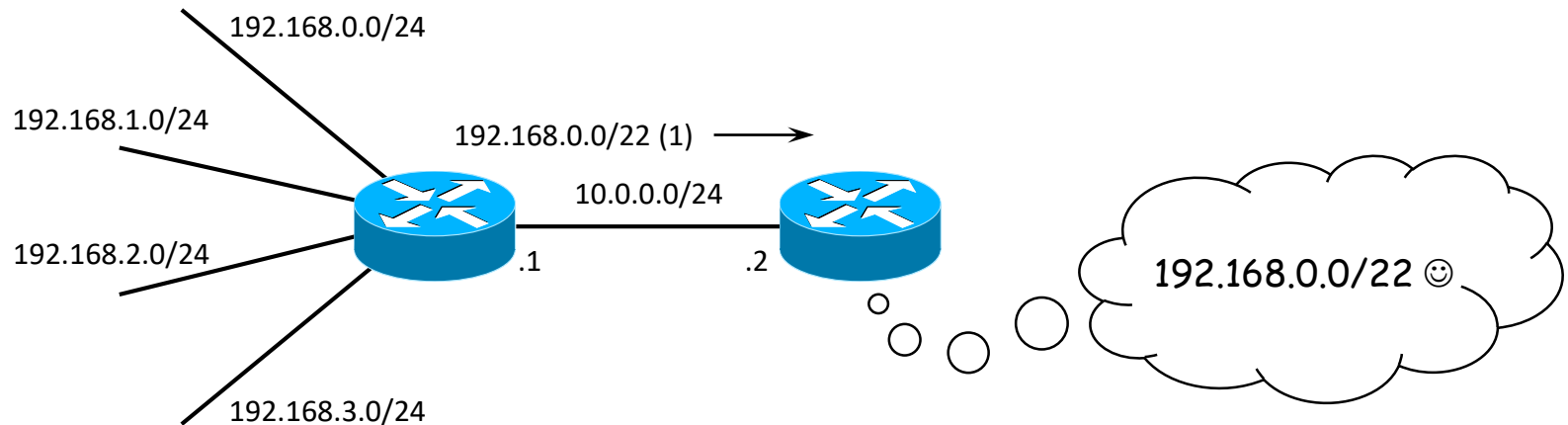


- Suporta subnetting de comprimento variável



Supernetting com RIPv2

- Suporta *supernetting*



NOTA: A agregação numa supernet, como neste exemplo, não é automática.