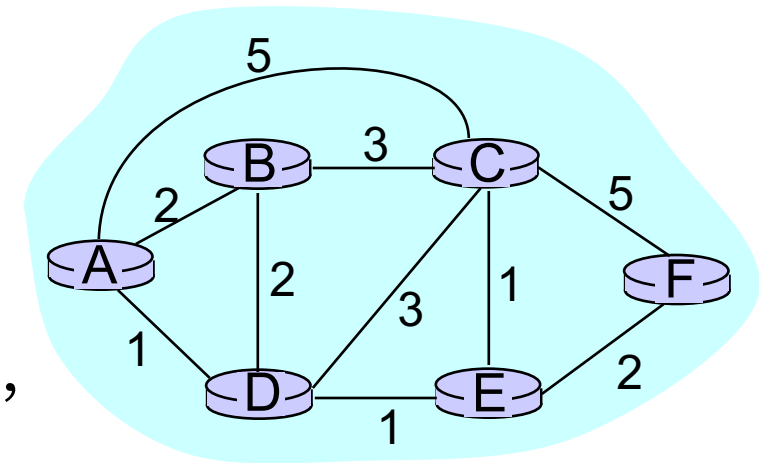


Camada de rede

- ❑ Protocolos de encaminhamento
 - Estado-da-ligação
 - Vetor-distância
- ❑ Engenharia de tráfego
- ❑ Endereçamento IP
- ❑ Encaminhamento na Internet
 - Intra-domínio
 - Inter-domínio
- ❑ Internet Protocol (IP), IPv6, DHCP e NAT

Encaminhamento por caminho mais curto

- ❑ Rede modelada por um grafo: nós e ligações
- ❑ Cada ligação e cada caminho têm um comprimento
- ❑ O comprimento de um caminho é a soma dos comprimentos das suas ligações
- ❑ O objetivo é determinar caminhos mais curtos, isto é, de menor comprimento entre origens e destinos



Protocolos de encaminhamento

❑ Estado-da-ligação

- Nós difundem o comprimento das suas ligações com os vizinhos por toda a rede
- Cada nó determina caminhos mais curtos autonomamente

❑ Vetor-distância

- Nós anunciam estimativas das distâncias aos seus vizinhos
- Cada nó calcula a sua distância a cada destino a partir das estimativas anunciadas pelos seus vizinhos

❑ Vetor-caminho

- Como em vetor-distância, mas as estimativas das distâncias são acompanhadas pelo caminho que elas percorreram desde o destino

Difusão de LSAs

Pseudo-código no nó u

inicialização

para todo v

$$\perp sa_{\perp}[v] \quad := \quad 0$$
$$\text{lsa}_u[u] := \text{lsa}_u[u] + 1$$

envia $\langle \text{LSA}, u, \text{lsa}_u[u] \rangle$ a todos os vizinhos

recebe $\langle LSA, w, n \rangle$ do vizinho v

```
se (n > lsa_u[w])
```

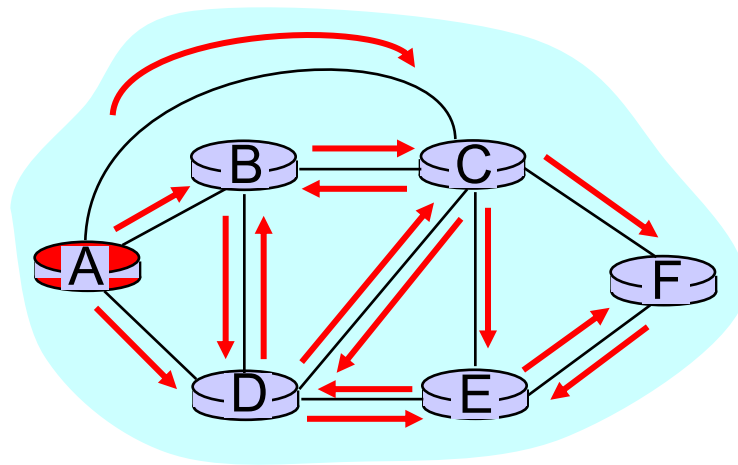
envia $\langle \text{LSA}, w, n \rangle$ a todos os vizinhos exceto v

$$\text{lsa}_1[w] := n$$

ligação de A a v é alterada

$$\text{lsa}_v[u] := \text{lsa}_v[u] + 1$$

envia $\langle \text{LSA}, u, \text{lsa}_u[u] \rangle$ a todos os vizinhos

 $\text{lsa}_u[w]$ – número do último LSA com origem no nó w conhecido em u 

Algoritmo de Dijkstra

Pseudo-código no nó u

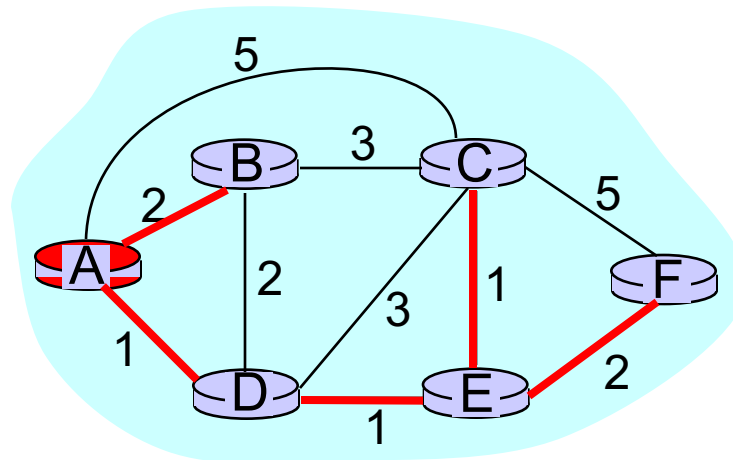
```
para todo  $v$ 
     $d_u[v] := \text{infinito}$ 
     $p_u[v] := \text{nulo}$ 
 $d_u[u] := 0$ 
 $N := \text{todos os nós}$ 

enquanto ( $N$  não é vazio)
    extrair de  $N$  o nó  $w$  tal que  $d_u[w]$  é mínimo
    para ( $v$  vizinho de  $w$ )
        se ( $d_u[v] > d_u[w] + c[w, v]$ )
             $d_u[v] := d_u[w] + c[w, v]$ 
             $p_u[v] := w$ 
```

$d_u[v]$ – distância do nó u ao nó v
 $p_u[v]$ – predecessor no caminho mais curto de u até v
 $c[w, v]$ – comprimento da ligação entre os nós w e v

Algoritmo de Dijkstra: exemplo

iteração	N	d[B],p[B]	d[C],p[C]	d[D],p[D]	d[E],p[E]	d[F],p[F]
→ 1	BCDEF	2,A	5,A	1,A	infinito	infinito
→ 2	BCEF	2,A	4,D		2,D	infinito
→ 3	BCF	2,A	3,E			4,E
→ 4	CF		3,E			4,E
→ 5	F					4,E
→ 6						



Protocolo estado-da-ligação: mecanismos

- ❑ Protocolo HELLO para descobrir vizinhos
- ❑ Difusão de LSAs (*Link-State Advertisements*)
 - Número de sequência: controlo da difusão; atualização da topologia
 - Idade: descarte de informação antiga; refrescamento de LSAs
- ❑ Sincronismo da base-de-dados de LSAs entre vizinhos quando uma nova ligação é adicionada
- ❑ Cálculo de caminhos mais curtos
- ❑ Preenchimento da tabela de expedição

Vetor-distância

Pseudo-código no nó u

inicialização

```
para todo vizinho  $v$  e destino  $w$   
     $d_{tab_u}[v, w] := \text{infinito}$   
 $d_u[u] := 0$   
envia  $\langle d_u[u], u \rangle$  a todos os vizinhos
```

recebe $\langle d, w \rangle$ do vizinho v

```
 $d_{tab_u}[v, w] := c[u, v] + d$   
 $d_u[w] := \min \{d_{tab_u}[y, w], y \text{ vizinho}\}$   
se  $w = u$   
     $d_u[w] := \min \{d_u[w], 0\}$   
se  $d_u[w]$  foi alterado  
    envia  $\langle d_u[w], w \rangle$  a todos os vizinhos
```

$c[u, v]$ – comprimento da ligação do nó u ao nó v
 $d_{tab_u}[v, w]$ – distância do nó u ao nó w via nó v
 $d_u[w]$ – distância do nó u ao nó w

Vetor-distância

Pseudo-código no nó u

ligação de t a u foi estabelecida ou o seu comprimento foi alterado
para todo w

envia $\langle d_u[w], w \rangle$ ao nó t

ligação de u a v foi desativada

para todo w

$d_u[w] := \min \{d_{tab_u}[u, v], v \text{ vizinho de } u\}$

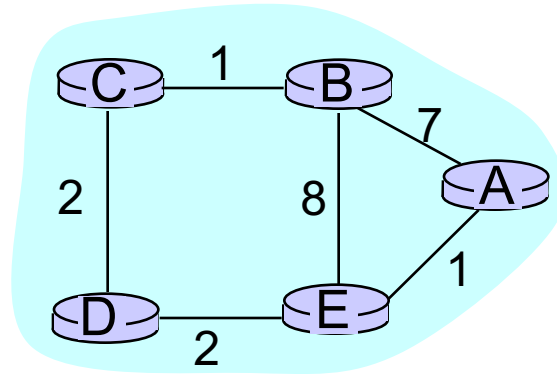
se $w = u$

$d_u[w] := \min \{d_u[w], 0\}$

se ($d_u[w]$ foi alterado)

envia $\langle d_u[w], w \rangle$ a todos os vizinhos

Exemplo de uma tabela de distâncias



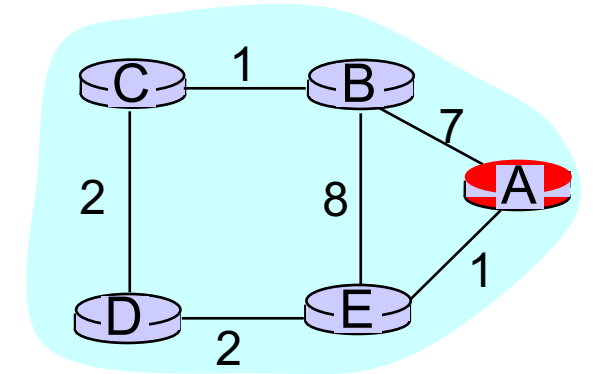
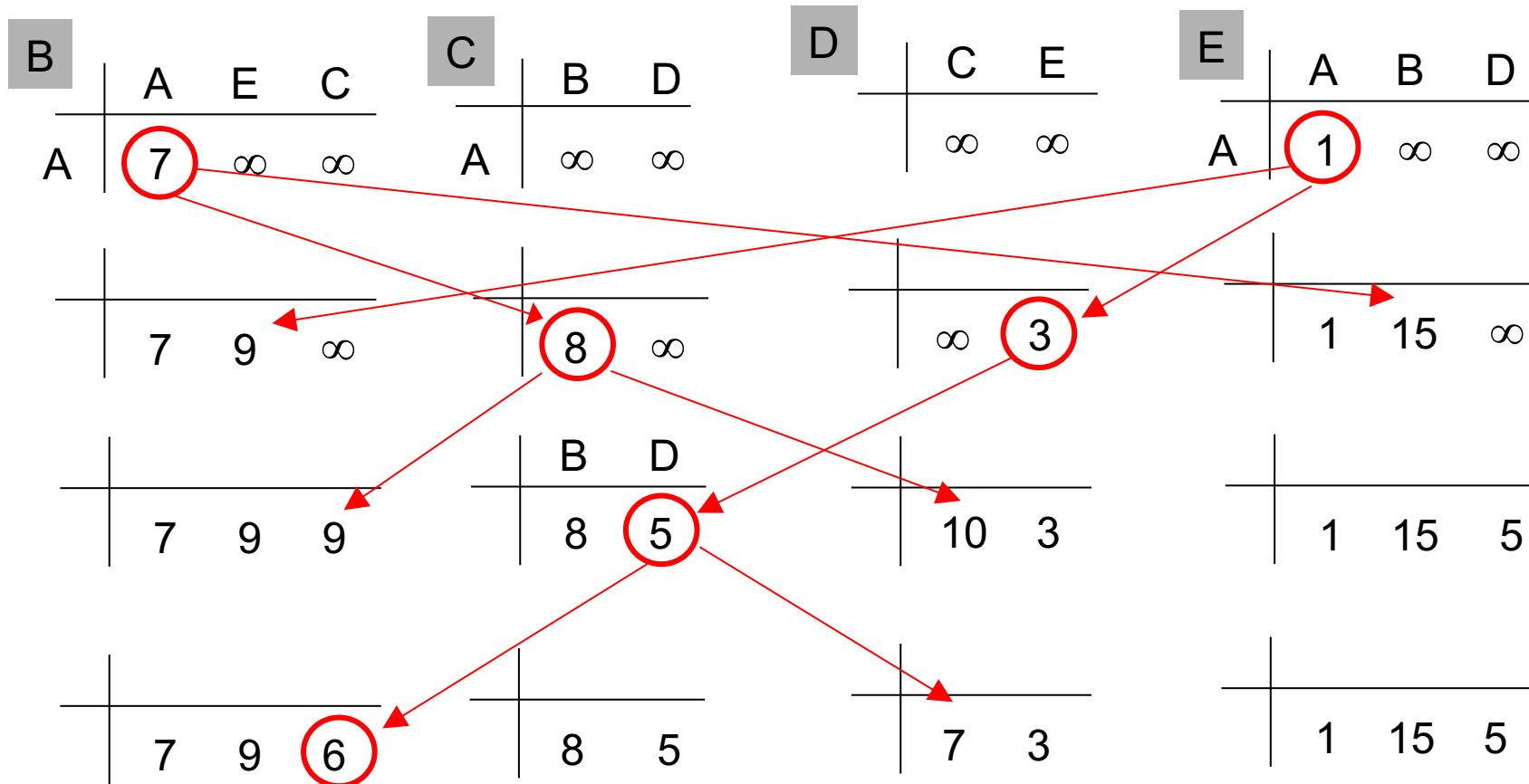
$$dtab_E[D,C] = c[E,D] + 2 = 4$$

$$dtab_E[D,A] = c[E,D] + 3 = 5 \text{ ciclo!}$$

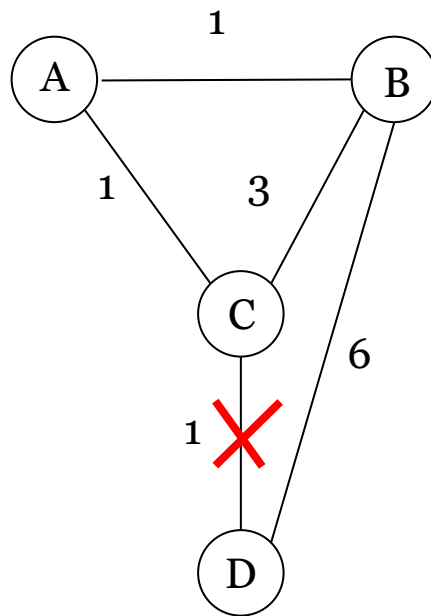
$$dtab_E[B,A] = c[E,B] + 6 = 14 \text{ ciclo!}$$

		vizinho		
dtab _E [,]		A	B	D
destino	A	1	14	5
	B	7	8	5
	C	6	9	4
	D	4	11	2

Vetor-distância: exemplo



Vetor distância: contagem para o infinito



Destino D

Anúncio de D

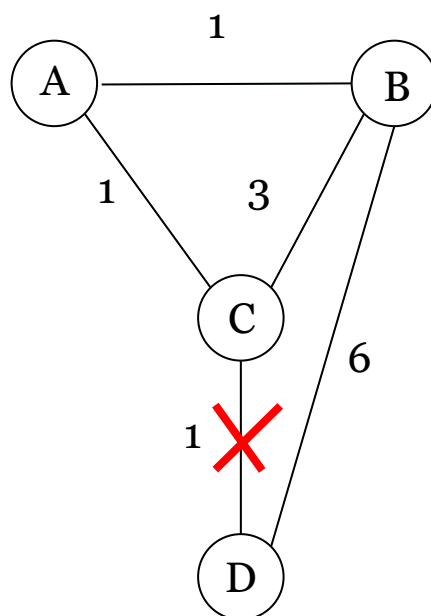
d_A	d_B	d_C
∞	∞	∞
∞	<u>6</u>	<u>1</u>
<u>2</u>	<u>4</u>	1
2	<u>3</u>	1
2	3	1

Falha de CD

d_A	d_B	d_C
2	3	<u>3</u>
<u>4</u>	3	3
4	<u>5</u>	<u>5</u>
<u>6</u>	5	5
6	<u>6</u>	<u>7</u>
<u>7</u>	6	7
7	6	<u>8</u>
7	6	8

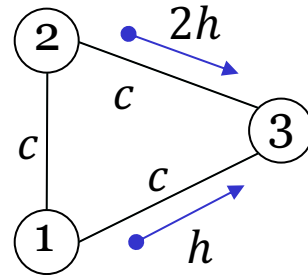
Vetor-caminho

Falha de CD

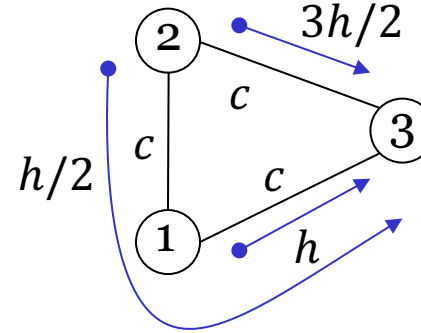


d_A, p_A	d_B, p_B	d_C, p_C
2, ACD	3, $BACD$	$=$
$=$	3, $BACD$	$-$
$-$	<u>6, BD</u>	$-$
<u>7, ABD</u>	6, BD	<u>9, CBD</u>
7, ABD	6, BD	<u>8, $CABD$</u>
7, ABD	6, BD	8, $CABD$

Engenharia de tráfego



$$h_{23} = 2h \quad h_{13} = h$$



	Carga total (plus)	Carga máxima (max)
Minimizar carga (min, plus)	$3h/c$	$2h/c$
Balancear carga (min, max)	$7h/2c$	$3h/2c$

Encaminhamento hierárquico

- ❑ Partição da rede em áreas
- ❑ Agregação de endereços de área
 - Redução do número de entradas nas tabelas de expedição
 - Caminhos extremo-a-extremo deixam de ser ótimos
- ❑ Protocolo de encaminhamento intra-área pode ser diferente do protocolo de encaminhamento inter-área

Encaminhamento hierárquico: exemplo

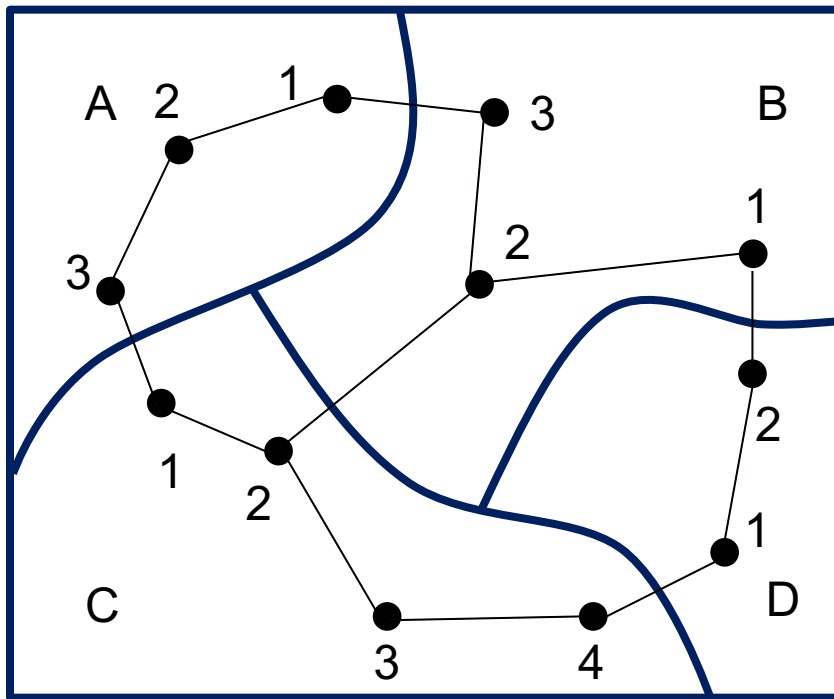


Tabela de expedição em A.3

Destino	Saída	Comprimentos
A.2	A.2	(0,1)
A.1	A.2	(0,2)
B	B.3	(1,3)
C	C.1	(1,1)
D	C.1	(2,1)

- Caminho dos pacotes: A.3-A.2-A.1-B.3-B.2
- Caminho mais curto: A.3-C.1-C.2-B.2

Endereços e prefixos IP (IPv4)

Endereço IP

223.1.1.147 → 11011111 00000001 00000001 10010011

Prefixo IP

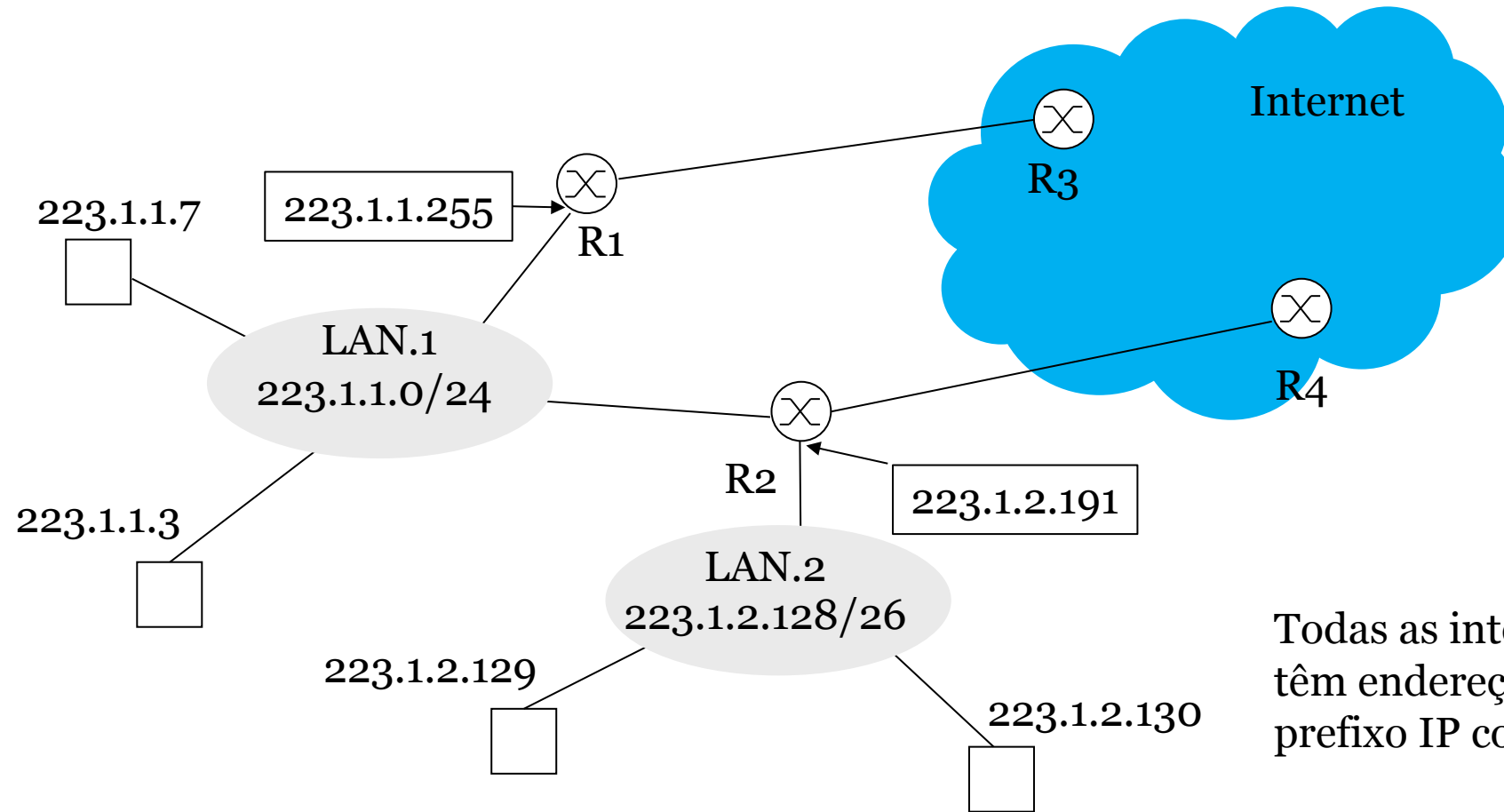
223.1.1.128/26
26 bits de máscara
64 endereços

223.1.1.128	→	11011111 00000001 00000001 10000000
•		•
•		•
•		•
223.1.1.191	→	11011111 00000001 00000001 10111111

O endereço 223.1.1.147 pertence ao prefixo 223.1.1.128/26

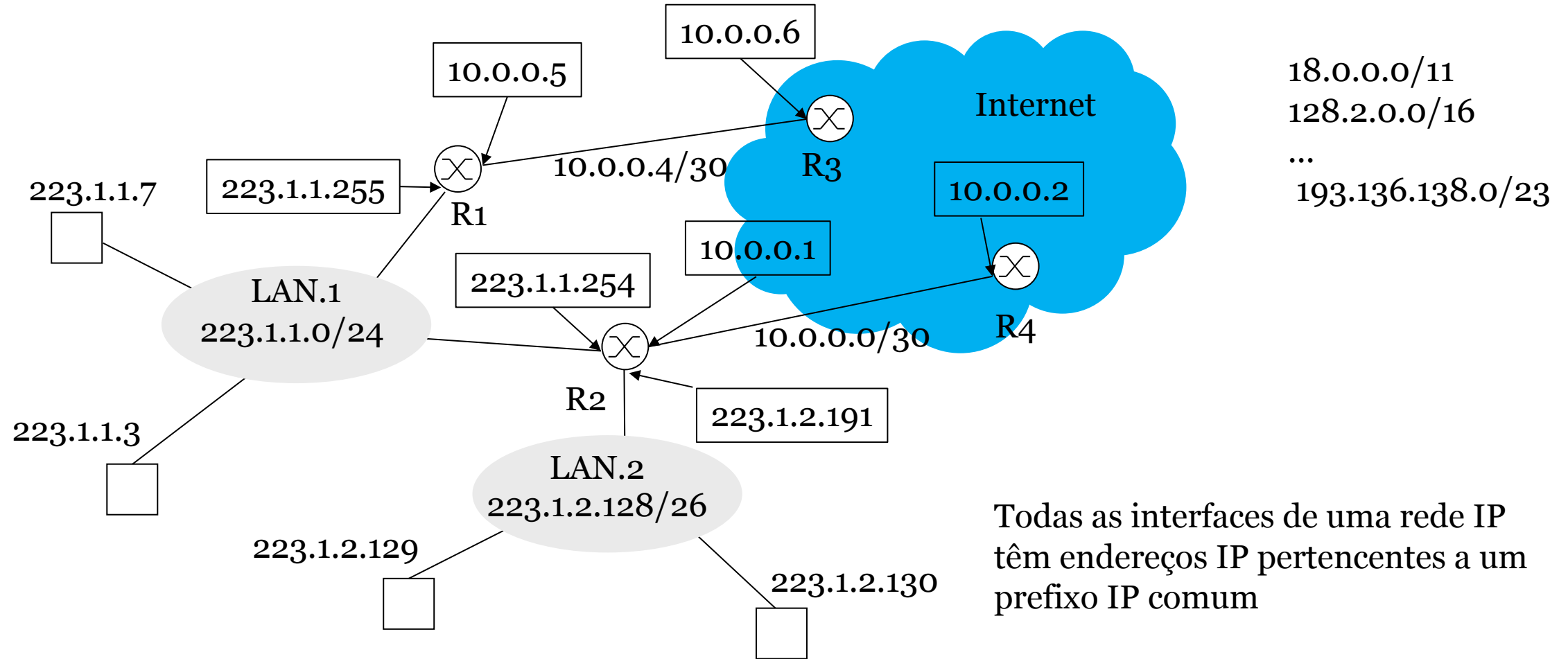
O prefixo 223.1.1.128/26 está contido no prefixo 223.1.1.0/24

Endereços e prefixos IP

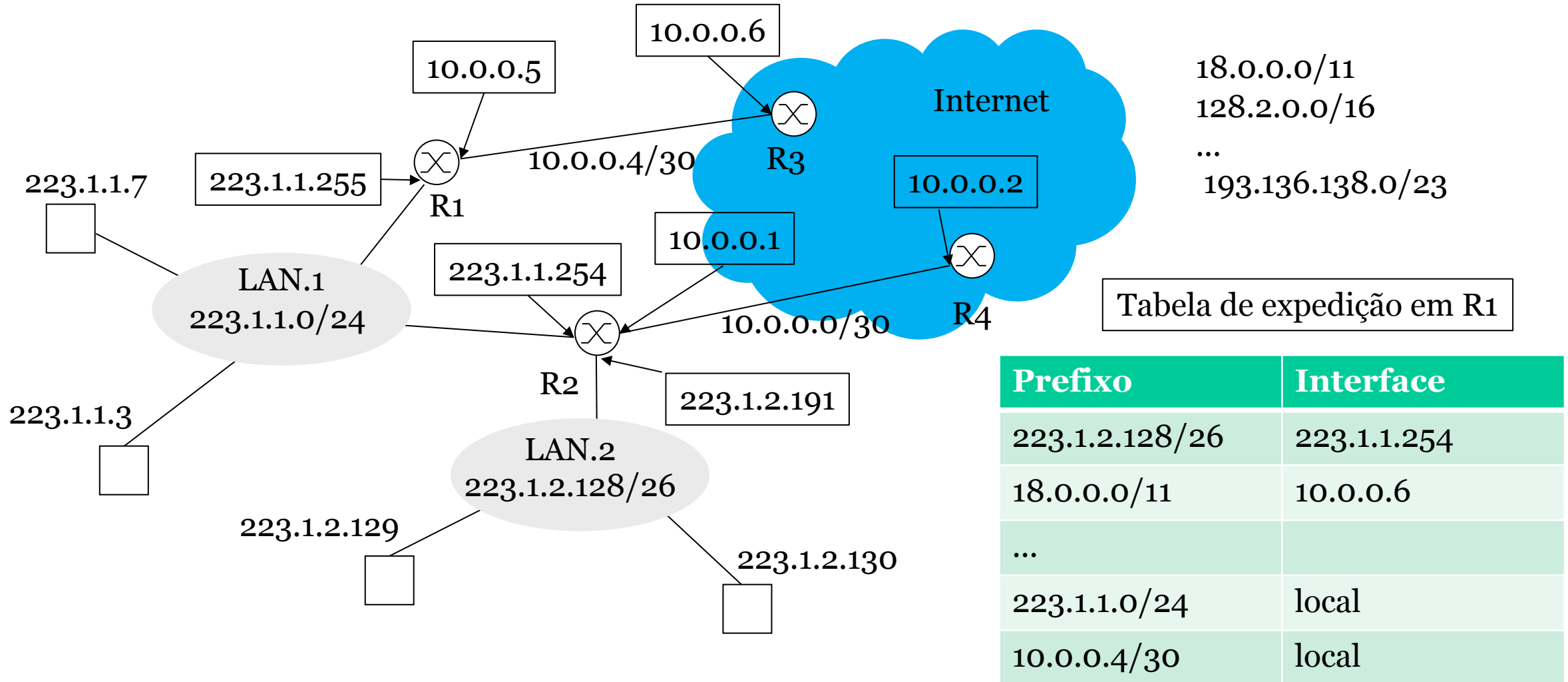


Todas as interfaces de uma rede IP têm endereços IP pertencentes a um prefixo IP comum

Endereços e prefixos IP



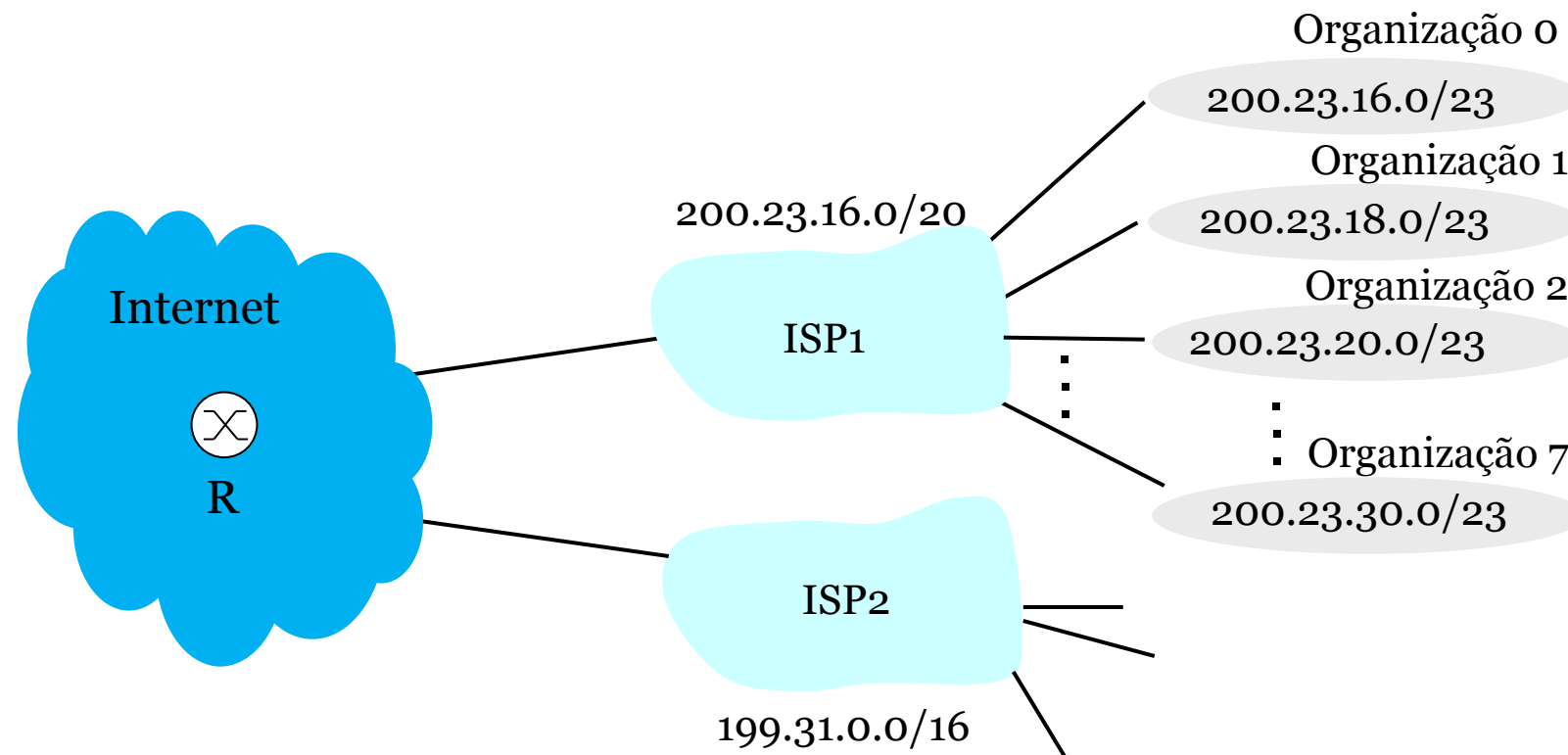
Tabelas de expedição



Agregação de endereços IP

Tabela de expedição em R

Prefixo	Saída
200.23.16.0/20	ISP1
199.31.0.0/16	ISP2

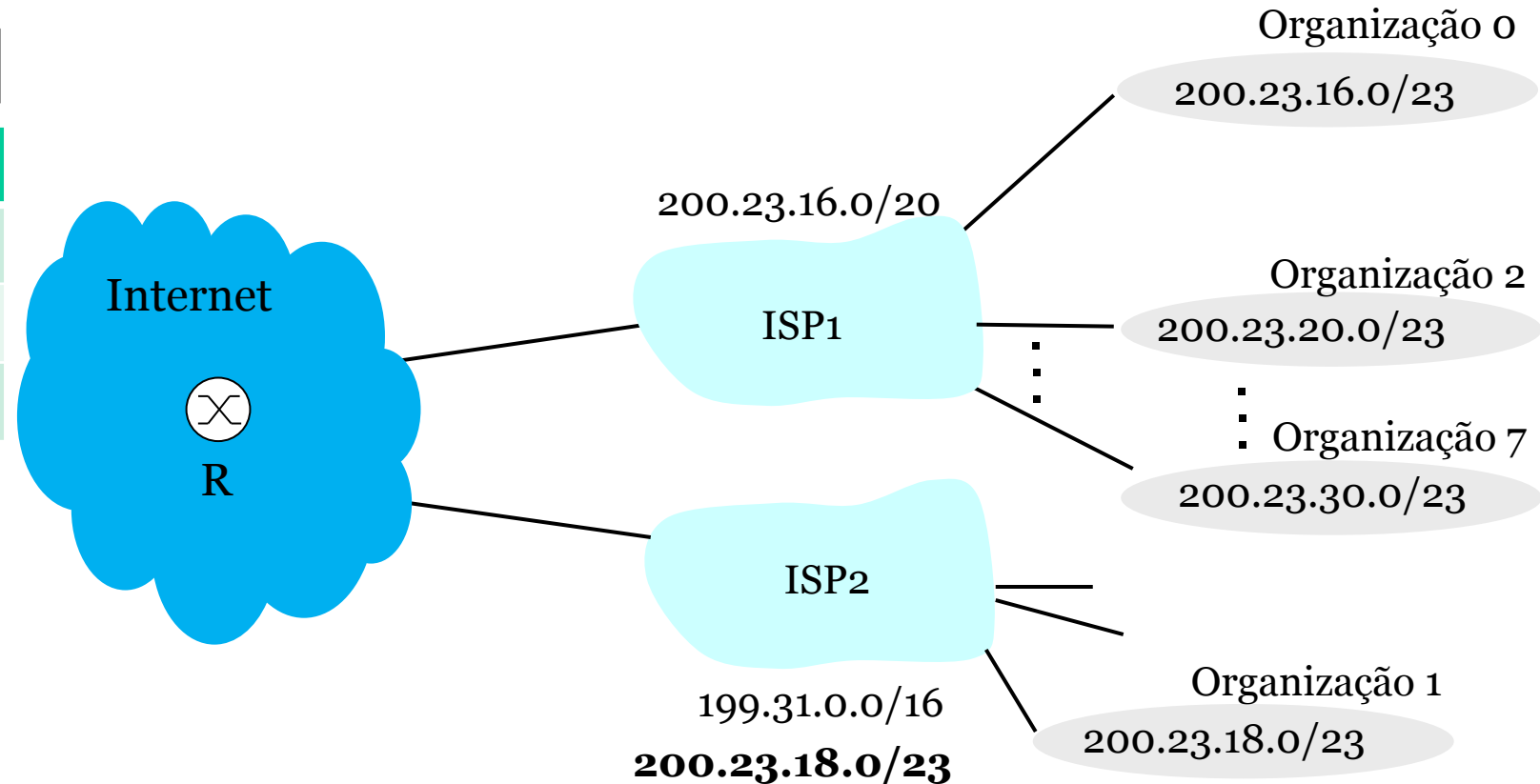


O encaminhador R só tem entradas na sua tabela de expedição para os prefixos agregados

Expedição por caminho mais longo

Tabela de expedição em R

Prefixo	Saída
200.23.16.0/20	ISP1
200.23.18.0/23	ISP2
199.31.0.0.0/16	ISP2



O prefixo 200.23.18.0/23 está contido no prefixo 200.23.16.0/20; expedição por prefixo mais longo

Encaminhamento na Internet

- ❑ A Internet consiste num número de sistemas autónomos interligados (*Autonomous System, AS*)
 - AS terminal
 - AS multi-terminal
 - AS de trânsito
- ❑ Encaminhamento intra-AS (*Interior Gateway Protocols, IGP*)
 - RIP: *Routing Information Protocol*
 - OSPF: *Open Shortest Path First*
- ❑ Encaminhamento inter-AS (*Exterior Gateway Protocol, EGP*)
 - BGP: *Border Gateway Protocol*

Routing Information Protocol (RIP)

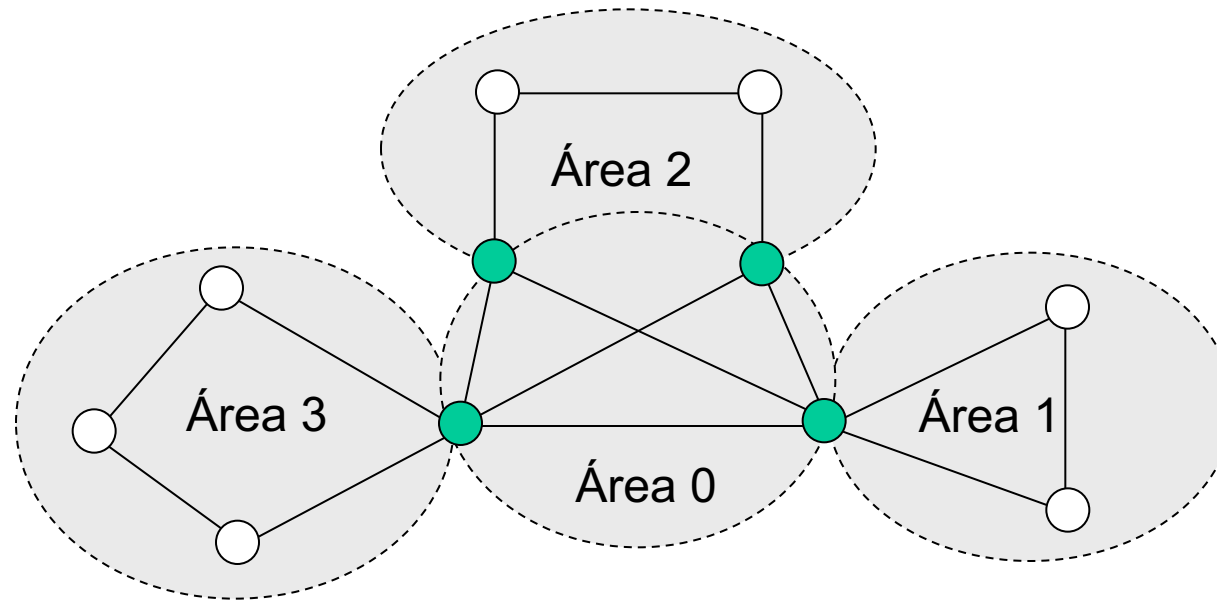
- ❑ Vetor distância
- ❑ Comprimento das ligações é unitário
- ❑ Diâmetro da rede inferior a 16
- ❑ Cada encaminhador envia periodicamente distâncias aos vizinhos
 - Encaminhadores só guardam distâncias aos destinos (e não via vizinhos)
- ❑ Ausência prolongada distâncias vindas de um vizinho indica falha na ligação
- ❑ Opera sobre UDP (!)
- ❑ Processo que executa RIP é o `routed`

Open Shortest Path First (OSPF)

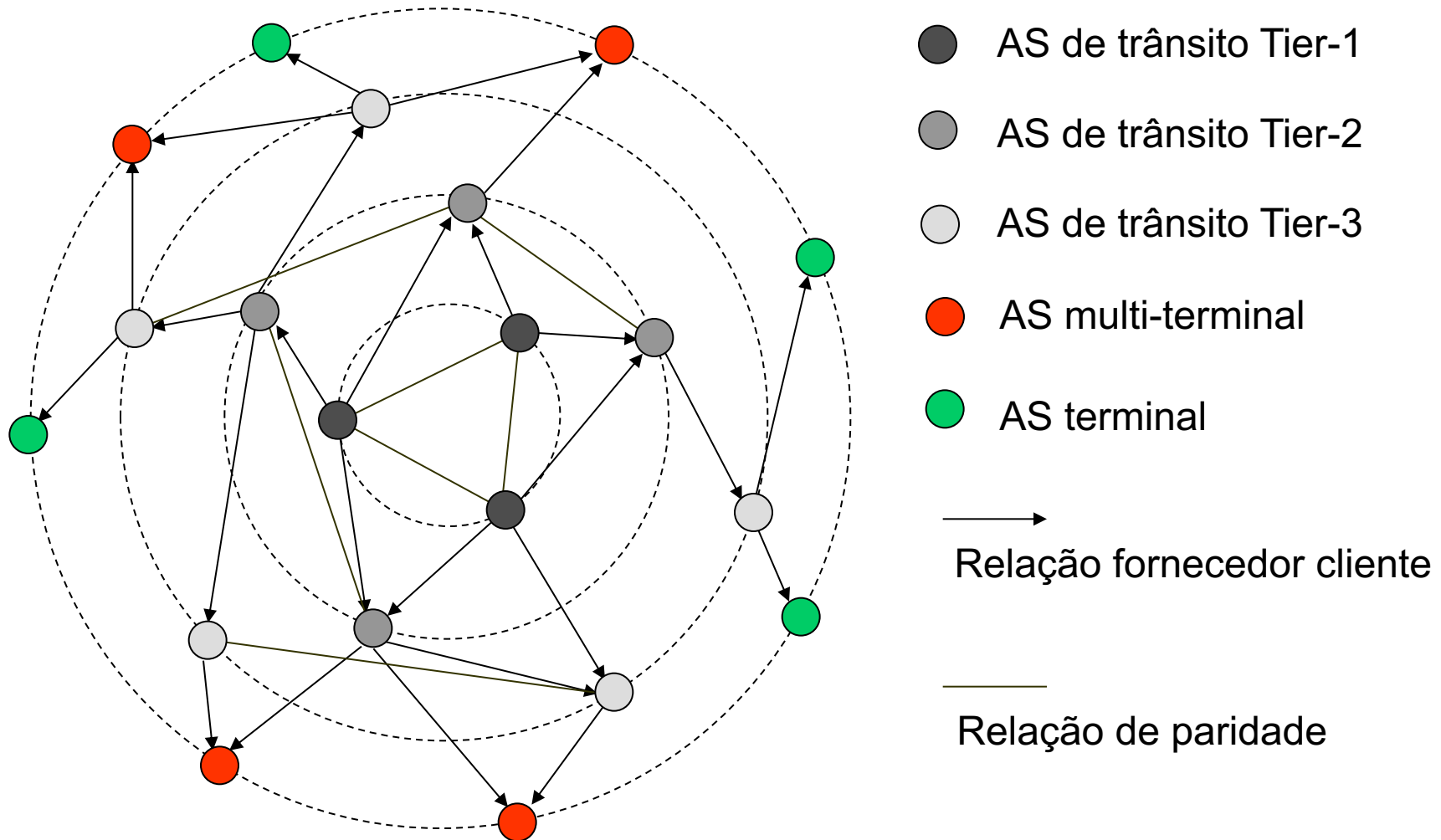
- ❑ Estado da ligação
- ❑ Múltiplos caminhos do mesmo comprimento entre pares origem destino (*Equal Cost Multipath*, ECMP)
- ❑ Segurança
- ❑ Suporte de hierarquias dentro de uma AS
- ❑ Suporte de *multicast*
- ❑ Opera sobre IP
- ❑ Processo que executa OSPF é o `gated`

OSPF hierárquico

- ❑ Encaminhamento intra-área: estado-da-ligação
- ❑ Encaminhamento inter-área: vetor-distância
- ❑ Toda a comunicação inter-área atravessa a área 0



A Internet



Border Gateway Protocol (BGP)

- ❑ Vetor caminho
 - Cada encaminhador fronteira anuncia todo o caminho de ASes até um prefixo IP destino
- ❑ Políticas na escolha e anúncio de caminhos
 - Cada AS pode decidir não anunciar os seus caminhos a algumas ASes vizinhas
 - Cada AS atribui um nível de preferência aos caminhos anunciados pelos vizinhos
- ❑ *External* BGP (eBGP) entre ASes
- ❑ *Internal* BGP (iBGP) entre encaminhadores de uma AS
- ❑ Opera sobre TCP

Parâmetros de rotas eBGP

❑ LOCAL-PREF

- Nível de preferência atribuído localmente a uma rota
- Não é passado para fora da AS

❑ AS-PATH

- Sequência de ASes atravessados pela rota
- Acrescentado pela AS que anuncia a rota

❑ NEXT-HOP

- Endereço IP do encaminhador fronteira da AS vizinha que anuncia a rota

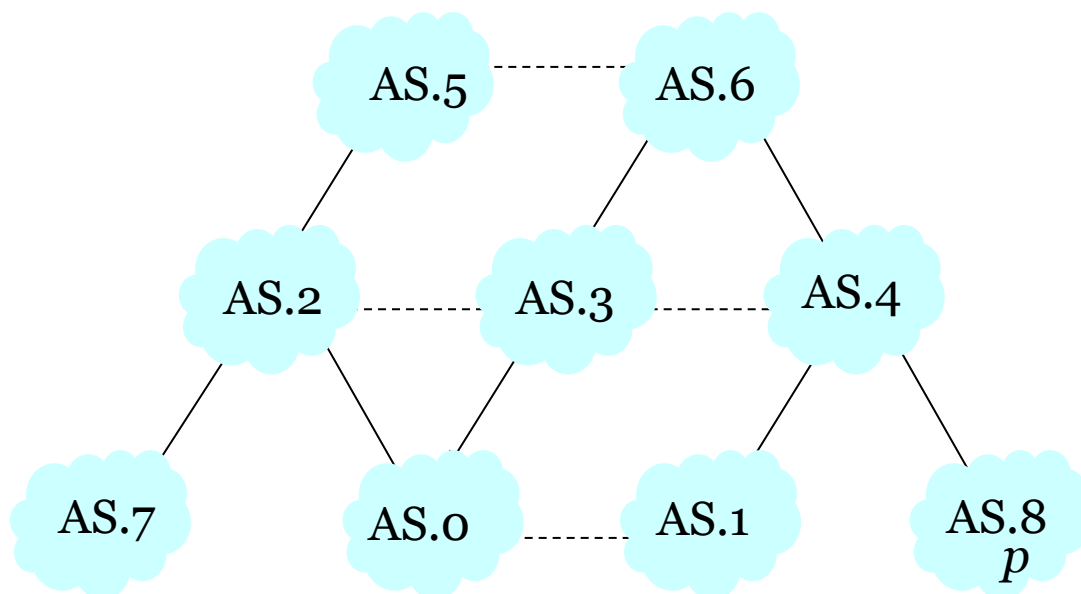
❑ MED

- Discrimina ligações diferentes a uma mesma AS vizinha
- Algum controlo sobre o tráfego que entra na AS

LOCAL-PREF, AS-PATH e filtragem

1. Cliente (LOCAL-PREF = 300)
2. Par (LOCAL-PREF = 200)
3. Fornecedor (LOCAL-PREF = 100)

De \ Para	Cli.	Par	For.
Cli	V	V	V
Par	V	X	X
For.	V	X	X



——— Fornecedor-Cliente (Fornecedor desenhado acima)
 Par-Par

Caminho AS.0 AS.1 AS.4 AS.8 é inválido

Rotas

LOCAL-PREF; AS AS-PATH; prefixo

100; AS.0 AS.3 AS.4 AS.8; *p*

200; AS.5 AS.6 AS.4 AS.8; *p*

NEXT-HOP: acoplamento inter-AS–intra-AS

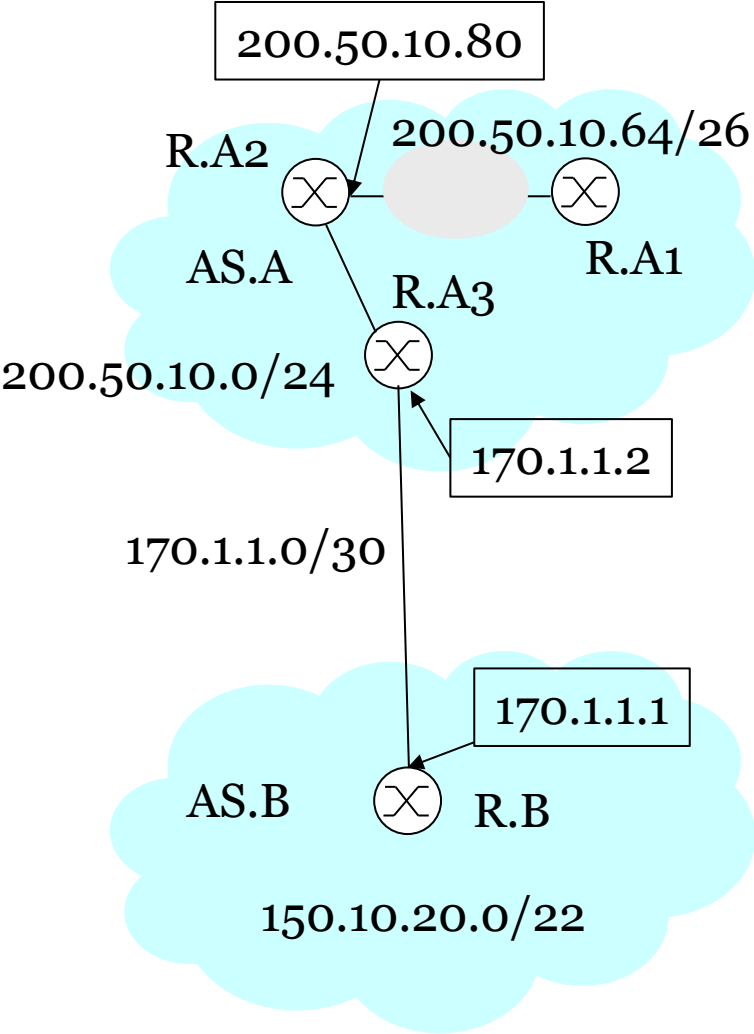


Tabela de expedição de R

Prefixo	Interface
150.10.20.0/22	170.1.1.1
170.1.1.0/30	200.50.10.80
200.50.10.64/26	local

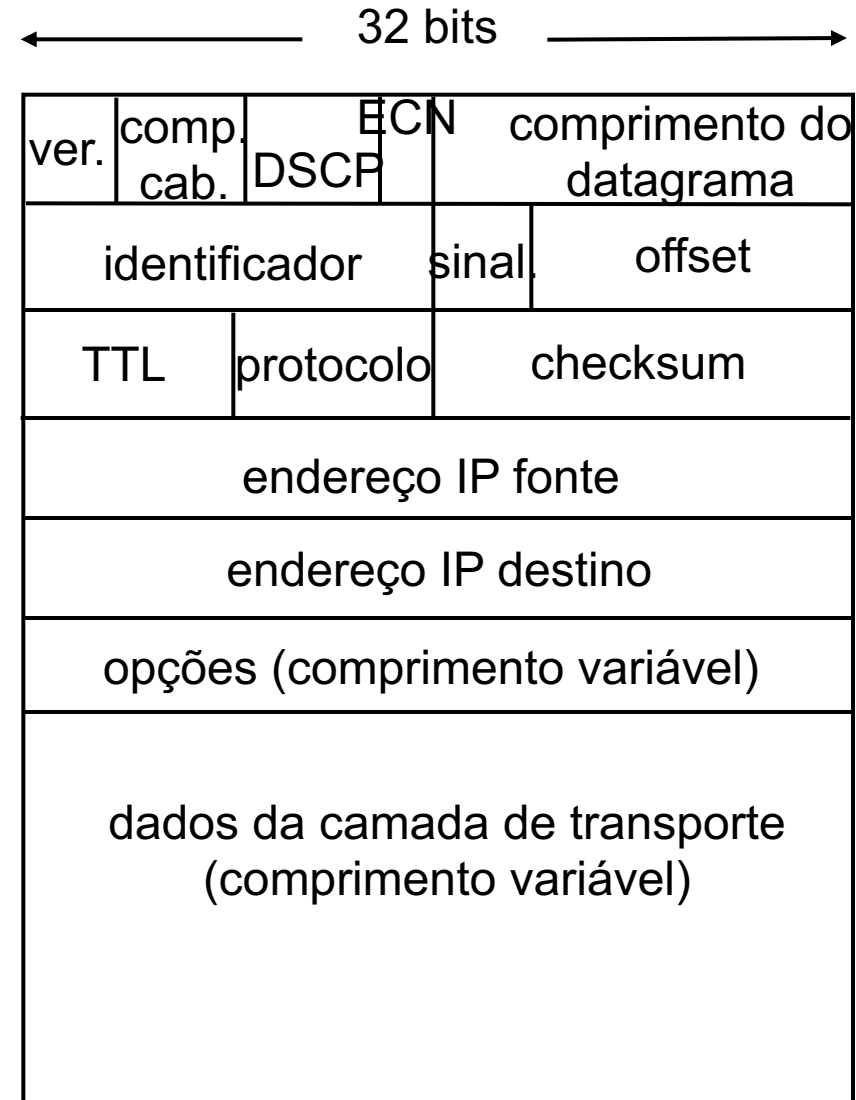
Anúncios de rotas

AS-PATH; prefixo; NEXT-HOP

R.B ->R.A3: AS.B; 150.10.20.0/22; 170.1.1.1 (eBGP)
R.A3 -> R.A1: AS.B; 150.10.20.0/22; 170.1.1.1 (iBGP)

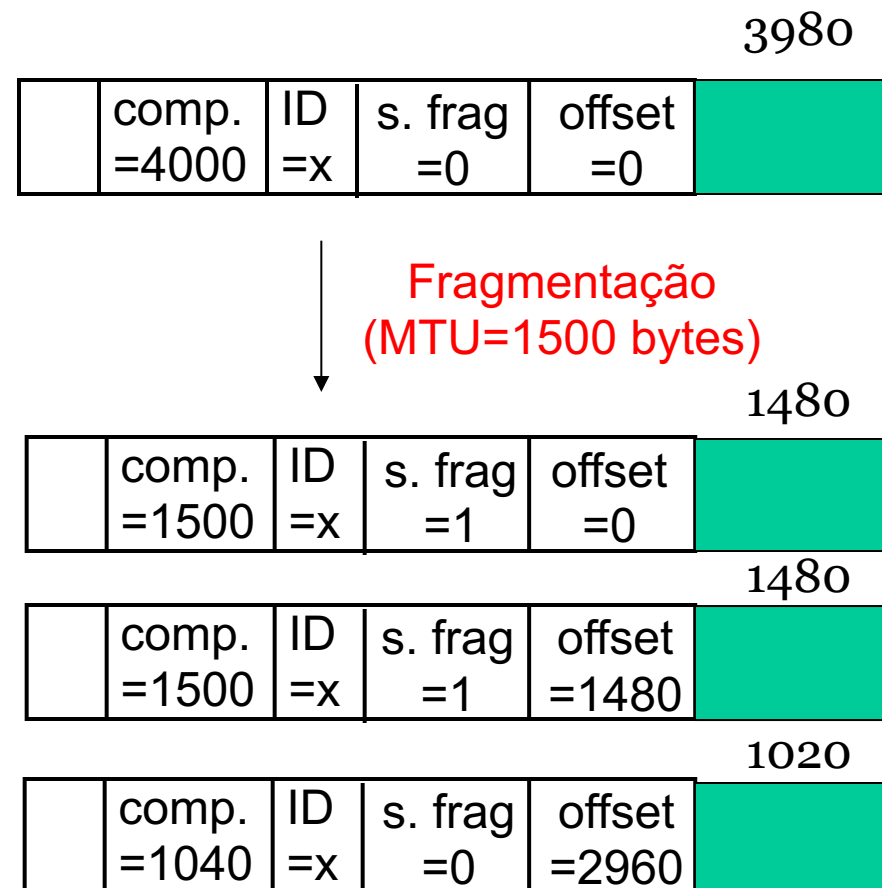
Anatomia de um datagrama IP

- ❑ DSCP
 - Tipo de serviço
- ❑ ECN
 - Notificação de congestão
- ❑ Protocolo
 - Chave de desmultiplexagem para a camada de transporte: TCP, UDP, ...
- ❑ Identificador, offset e sinalizadores
 - Fragmentação e reconstrução de datagramas
- ❑ TTL
 - Decrementado em cada encaminhador



Fragmentação e reconstrução

- ❑ Cada ligação tem um MTU
- ❑ Datagramas fragmentados na camada IP
- ❑ Reconstrução apenas no destino
- ❑ Campos no cabeçalho proporcionam reconstrução
- ❑ Offset indica número do primeiro byte do fragmento, em múltiplos de 8 bytes



Internet Control Message Protocol (ICMP)

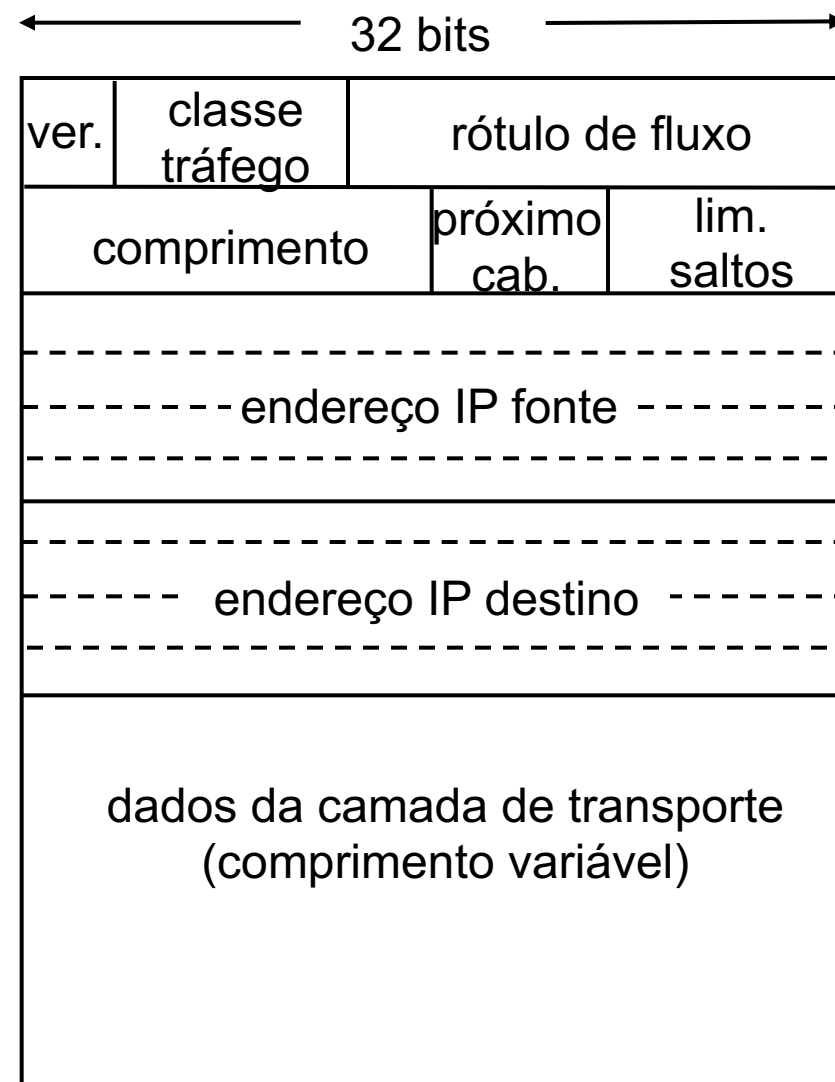
- ❑ Utilizado para propagação de informação de controlo
- ❑ Mensagens ICMP transportadas em datagramas IP
 - Tipo
 - Código
 - 8 primeiros bytes do datagrama que causou o erro
- ❑ Traceroute
 - ICMP time exceeded (TTL é zero)
- ❑ Ping
 - ICMP echo request
 - ICMP echo reply

IPv6

- ❑ Motivação principal
 - Escassez de endereços IP
- ❑ Razões adicionais
 - Expedição e comutação rápidas de datagramas
 - Qualidade de serviço
 - Autoconfiguração
 - Endereços *anycast*
- ❑ Formato dos datagramas
 - Endereços de 128 bits
 - Não permite fragmentação
 - Não tem checksum

Anatomia de um datagrama IPv6

- ❑ Classe de tráfego
 - Prioridades
- ❑ Rótulo de fluxo
 - QoS ?
- ❑ Comprimento
 - Dos dados apenas
- ❑ Próximo cabeçalho
 - Chave de desmultiplexagem
- ❑ Limite de saltos
 - Semelhante ao TTL



Representação de endereços IPv6

□ Endereço IPv6

- 128 bits separados por dois-pontos em 8 campos cada 16 bits
- cada campo de 16 bits representado por 4 caracteres hexadecimais

2001:0db8:130f:0000:0000:7000:0000:140b

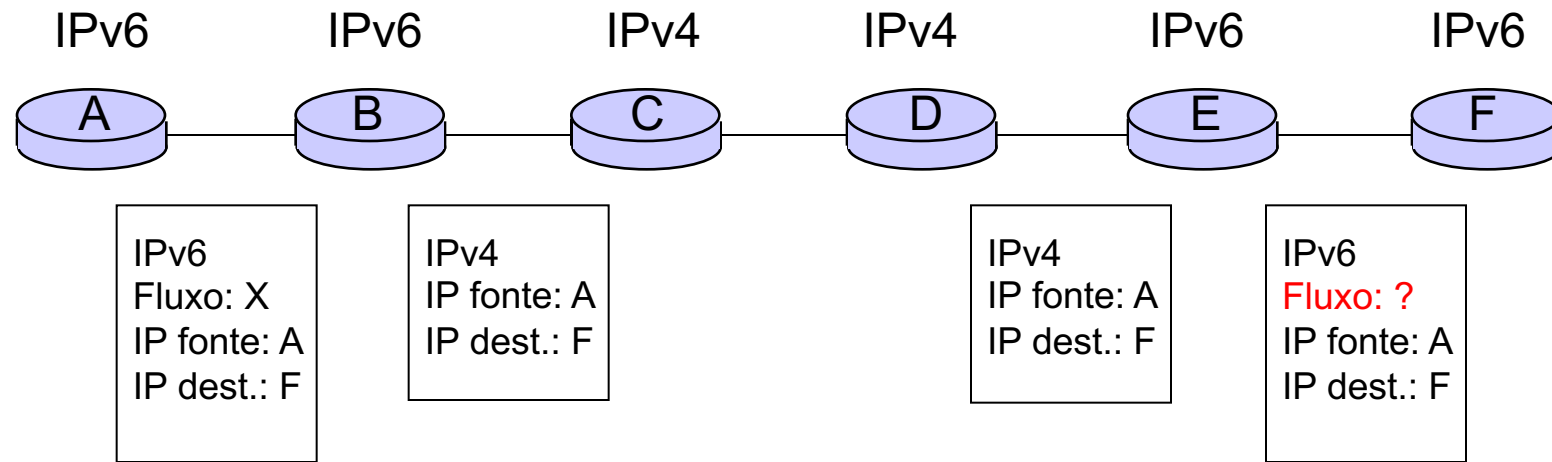
2001:db8:130f:0:0:7000:0:140b (zeros à esquerda são retirados)

2001:db8:130f::7000:0:140b (uma sucessão de zeros substituídos por ::)

Migração IPv4 - IPv6

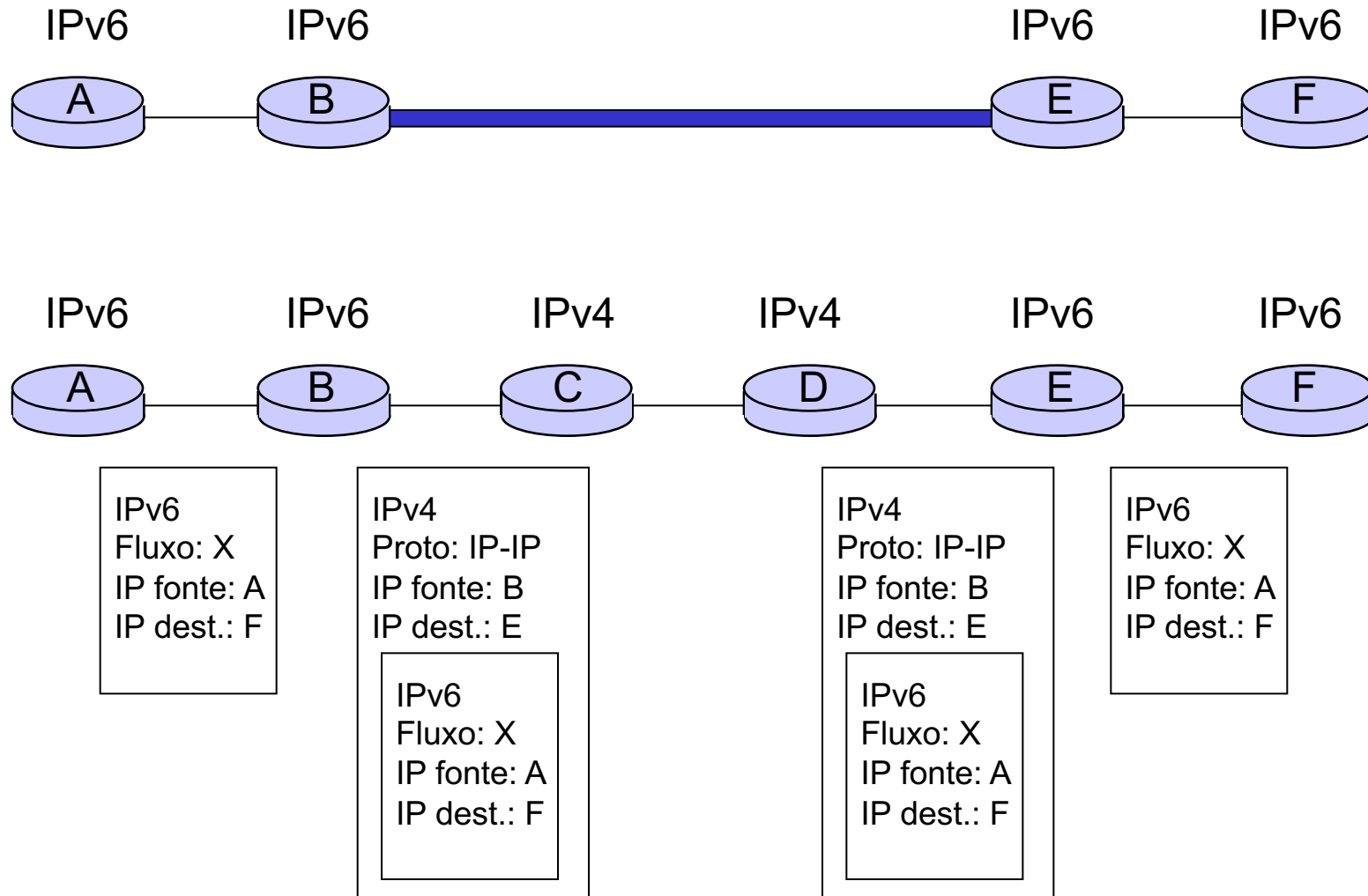
- ❑ Não vai haver um dia R em que todos os encaminhadores mudam de IPv4 para IPv6
- ❑ Migração de IPv4 para IPv6 terá que ser gradual
- ❑ Soluções de migração
 - Pilha dupla IPv4/IPv6: alguns encaminhadores conseguem traduzir datagramas entre os dois formatos
 - **Túneis: datagramas IPv6 transportados como dados de datagramas IPv4**

Pilha dupla



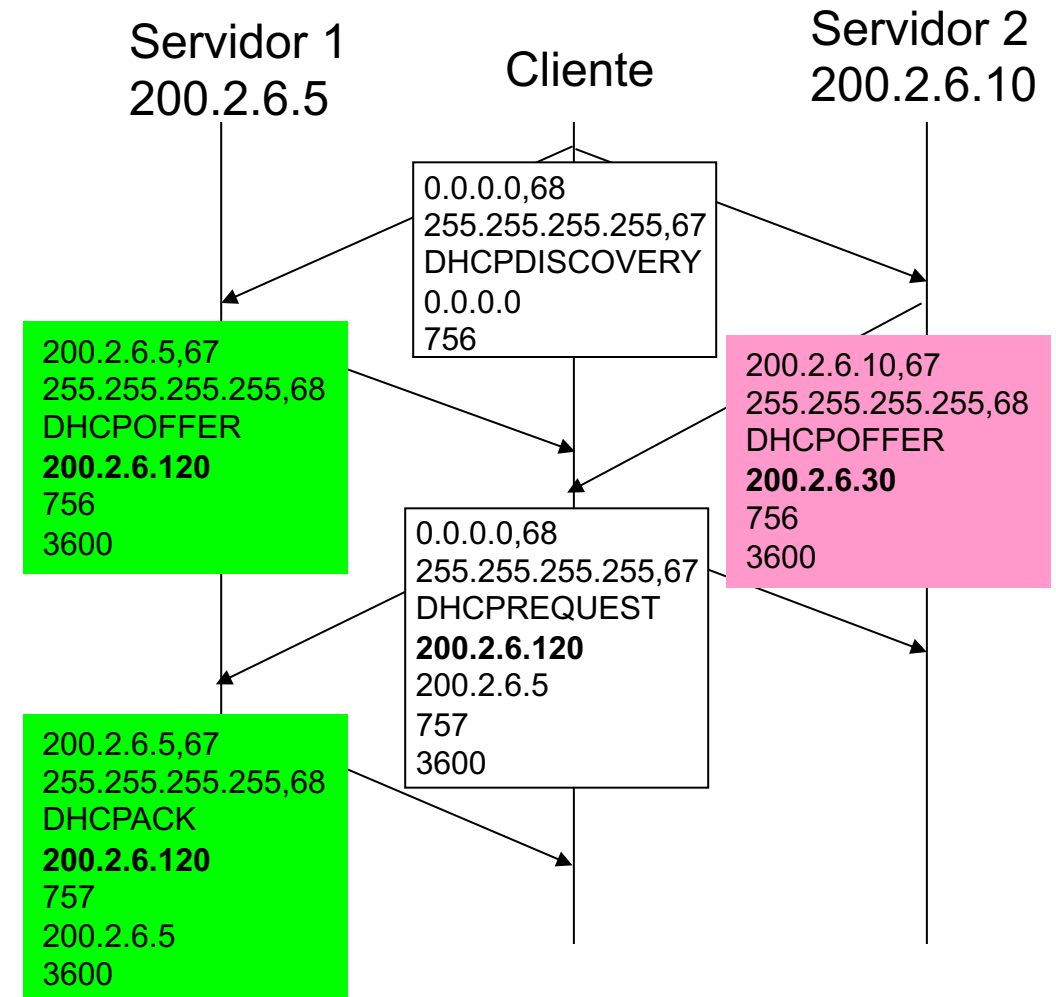
- ❑ Há informação no cabeçalho do datagrama IPv6 inicial que é perdida quando ele é traduzido num datagrama IPv4

Túnel

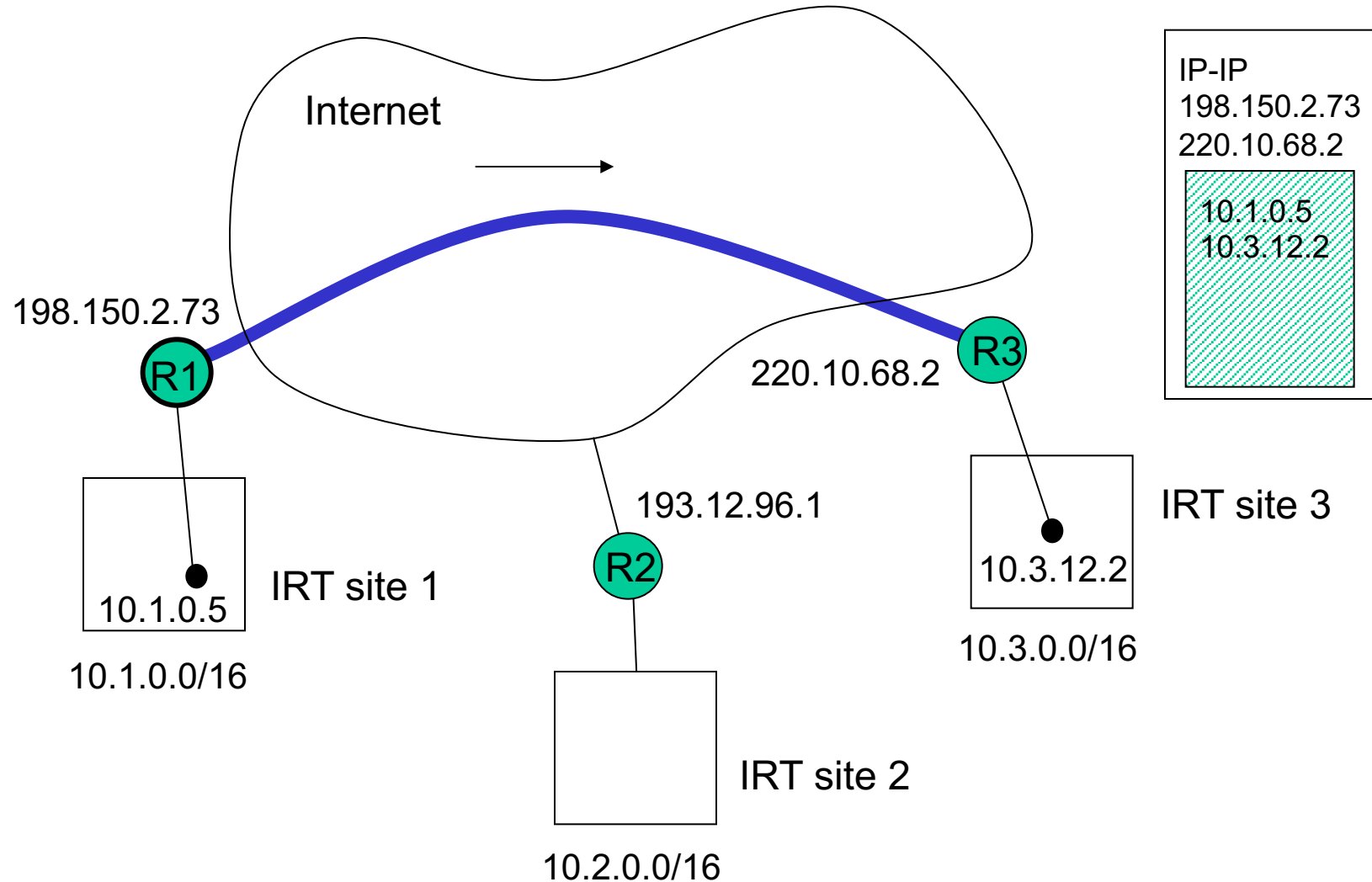


Dynamic Host Configuration Protocol (DHCP)

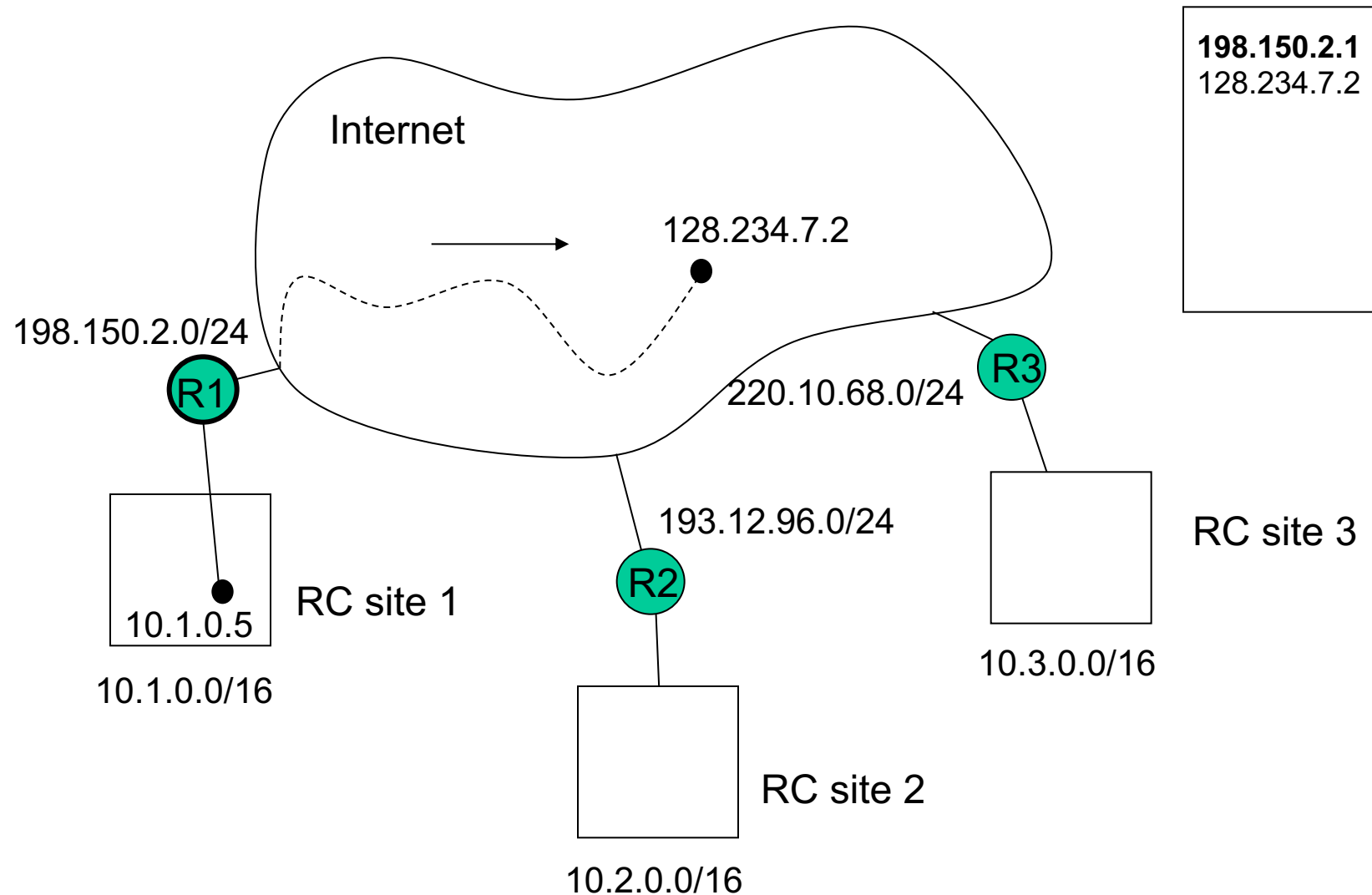
- ❑ Atribuição dinâmica de endereços IP
 - Partilha de endereços
 - Configuração automática
- ❑ Porto UDP bem-conhecido: 67 e 68
- ❑ Mensagens
 - DHCPDISCOVER
 - DHCPOFFER
 - DHCPREQUEST
 - DHCPACK



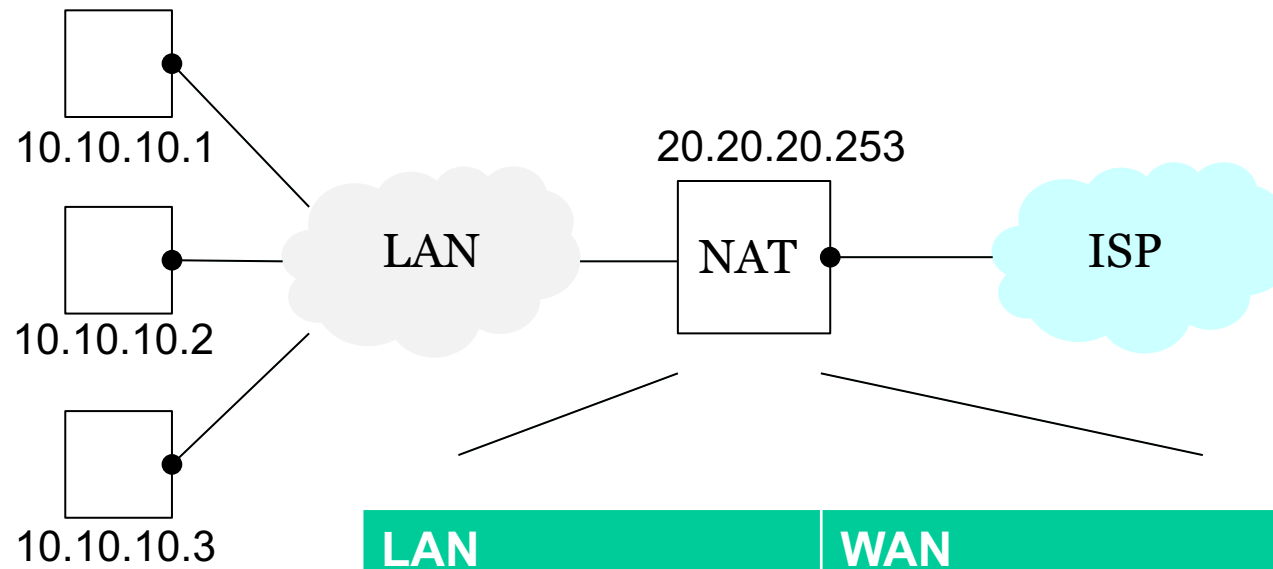
Virtual Private Networks (VPN)



Network Address Translation (NAT)



Network Address Translation (NAT)



LAN	WAN
10.10.10.1:3800	20.20.20.253:3800
10.10.10.1:3900	20.20.20.253:3900
10.10.10.2:3800	20.20.20.253:3801
10.10.10.3:3900	20.20.20.253:3901