

Administração de Redes 2023/24

Network Address Translation (NAT)

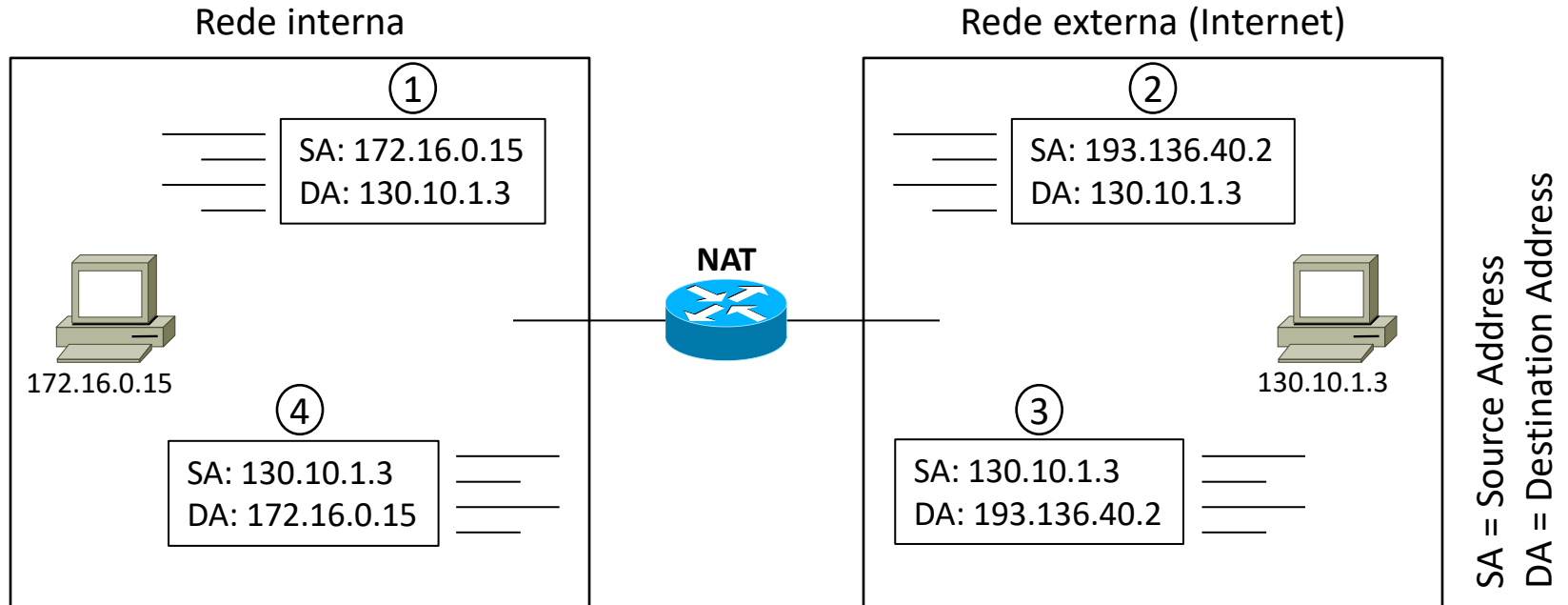
Motivação

- Escassez de endereços IP — motivação primordial
- Nem todas as máquinas de uma rede necessitam de acesso ao exterior (e.g., impressoras)
- Das que necessitam, nem todas acedem ao mesmo tempo
- É possível usar um número de endereços IP públicos (encaminháveis na Internet) inferior ao de máquinas na rede interna
 - Poupança de endereços
 - Internamente, as máquinas são numeradas com endereços IP de blocos privados (reutilizáveis livremente dado não serem encaminháveis na Internet)
- Traduzindo também as portas, pode usar-se um único endereço IP público para dar acesso ao exterior a múltiplas máquinas em simultâneo
 - Poupança de endereços muito maior
 - Frequentemente, os ISP fornecem apenas um endereço IP público (nos acessos domésticos / mais baratos)

Outras aplicações para o NAT

- Fusão de redes com endereços (privados) duplicados
- Tornar a numeração da rede independente do ISP
 - Facilidade de migração
 - *Dual homing* sem gerar rotas globais com prefixos demasiado longos
- Distribuição de carga
 - Múltiplas réplicas internas de um servidor vistas do exterior com um único endereço IP
 - Serviços implementados em máquinas diferentes vistos do exterior como estando no mesmo endereço IP
- Segurança (relativa)
 - Endereços reais das máquinas "escondidos" do exterior
 - Limitações no estabelecimento de conexões do exterior para o interior
 - É melhor ser uma *firewall* a fazê-lo...

NAT – Conceito



1. Primeiro pacote numa conexão para o exterior chega ao NAT
2. NAT escolhe um endereço público livre e traduz o endereço de origem
 - Tradução 172.16.0.15 → 193.136.40.2 é guardada numa tabela
3. NAT recebe do exterior pacote destinado a 193.136.40.2 (resposta)
4. Usando a tabela, faz a tradução inversa

Blocos de endereçamento privado

- Endereços não encaminháveis na Internet
 - Acesso à Internet só através de NAT
- Podem ser usados em qualquer rede
 - Não há conflito com endereços IP públicos na Internet
- 10.0.0.0/8 (de 10.0.0.0 a 10.255.255.255)
 - 1 sub-rede de classe A
- 172.16.0.0/12 (de 172.16.0.0 a 172.31.255.255)
 - 16 sub-redes de classe B
- 192.168.0.0/16 (de 192.168.0.0 a 192.168.255.255)
 - 256 sub-redes de classe C

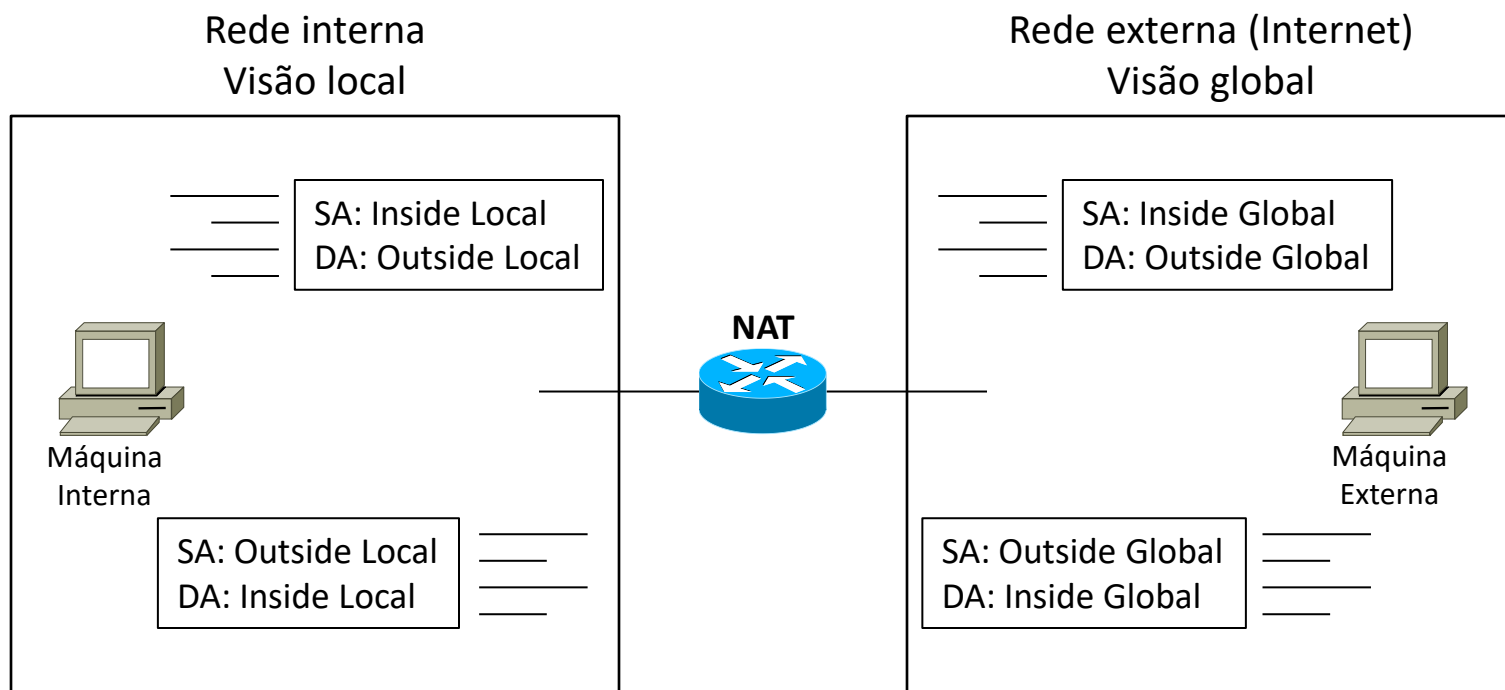
Tipos de NAT

- NAT Básico
 - Apenas os endereços IP são traduzidos
 - Outros campos também têm que ser ajustados, nomeadamente *checksums*
 - Também designado NAT Puro
- NAT com Tradução de Portas
 - Tradução de endereços IP e de portas (TCP, UDP)
 - O mesmo endereço IP público pode ser usado para várias máquinas internas
 - Maior poupança de endereços
 - Também designado como
 - Network Address and Port Translation (NAPT)
 - Port & Address Translation (PAT)
 - Address Overloading
- Masquerading
 - Nome dado em Linux a variante do NAPT em que é usado como endereço público o endereço da interface externa do NAT
 - Cisco IOS também suporta, mas não lhe dá um nome específico

Alguma terminologia usada no Cisco IOS

- Inside/Outside refere-se à localização do terminal (no interior ou exterior do NAT)
- Local/Global refere-se ao ponto de vista (do interior ou do exterior do NAT)
- Quatro tipos de endereços
 - **Inside Local (IL)** é o endereço da máquina interna conforme ele é visto na rede interna (i.e., o endereço que ela realmente usa)
 - **Inside Global (IG)** é o endereço da máquina interna conforme ele é visto no exterior (normalmente é um endereço traduzido)
 - **Outside Global (OG)** é o endereço da máquina externa conforme ele é visto no exterior
 - **Outside Local (OL)** é o endereço da máquina externa conforme ele é visto na rede interna (pode ser um endereço traduzido)

Inside/Outside, Local/Global



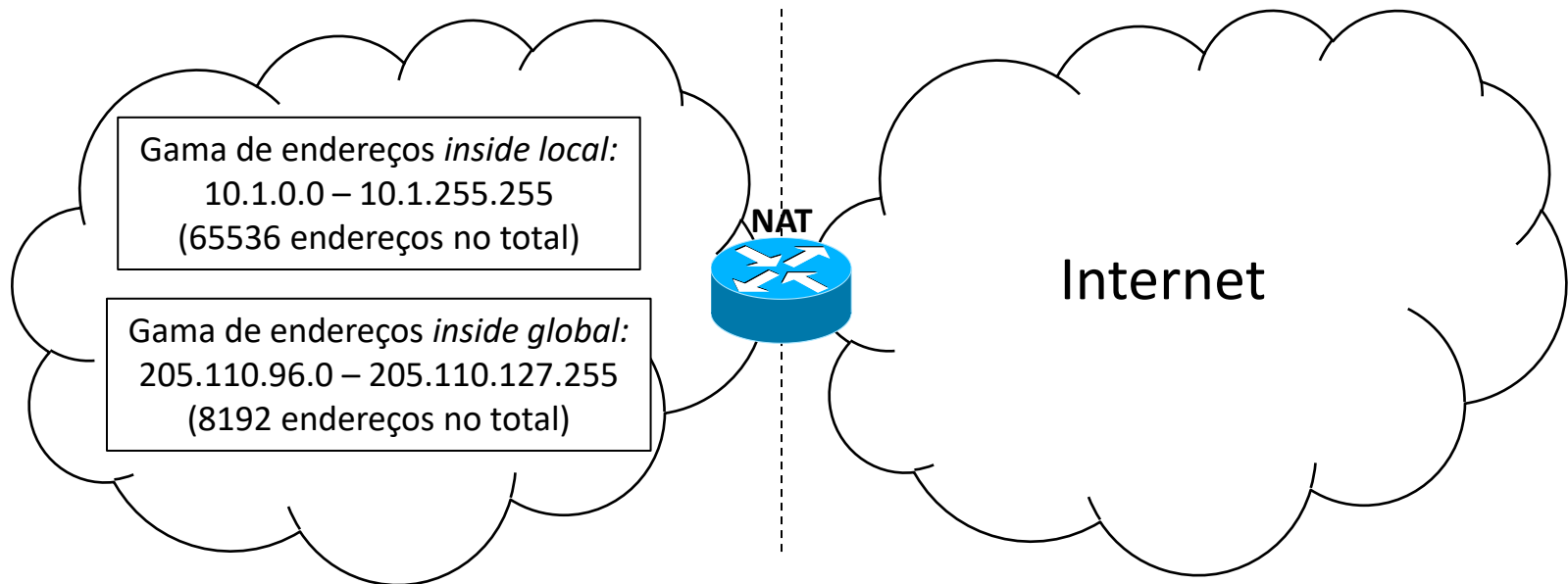
SA = Source Address

DA = Destination Address

NAT Básico (NAT Puro)

- Apenas são traduzidos endereços IP
- Endereços podem ser
 - Obtidos dinamicamente de uma *pool*
 - Normalmente menor que a gama de endereços privados usada
 - Configurados num mapeamento directo (1:1, estático)
 - Neste caso não há poupança de endereços
 - Tradução fixa é útil para ter servidores acessíveis do exterior
- Dependendo da aplicação, podem ser traduzidos
 - Apenas os endereços internos (mais comum)
 - Apenas os endereços externos
 - Ambos

Poupança de endereços IP



- Mapeamentos são criados quando uma máquina interna acede ao exterior (Internet)
- São apagados após um tempo de inatividade
 - Endereço público fica livre para ser usado por outra máquina interna

Exemplo de tabela NAT

- Após alguns acessos ao exterior, a tabela poderá conter

```
NATrouter#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
---	205.110.96.2	10.1.1.20	---	---
---	205.110.96.3	10.1.197.64	---	---
---	205.110.96.1	10.1.63.148	---	---

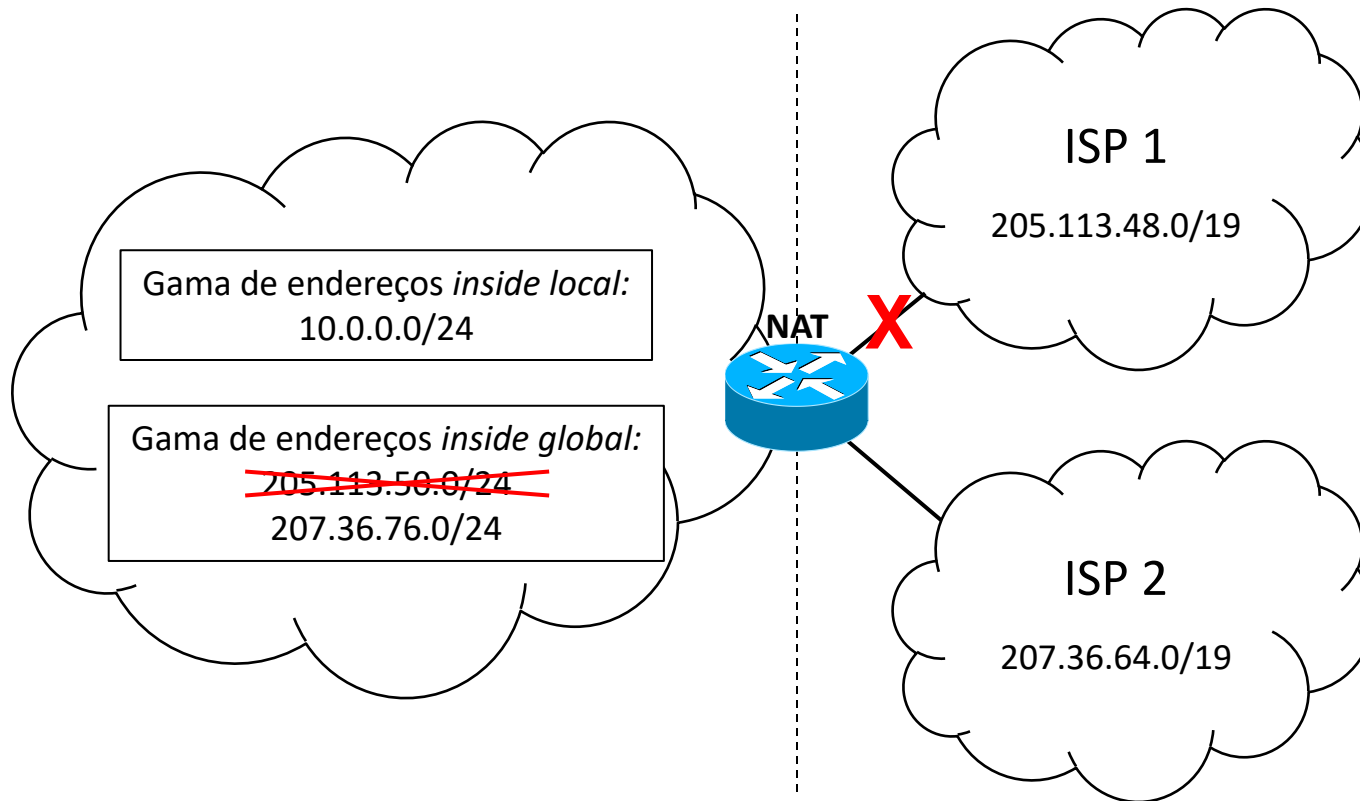
```
NATrouter#
```

- Note-se que, neste caso,
 - Apenas os endereços *inside* estão a ser traduzidos
 - A tradução é independente da máquina externa
 - Não há informação de portas na tabela → NAT Básico

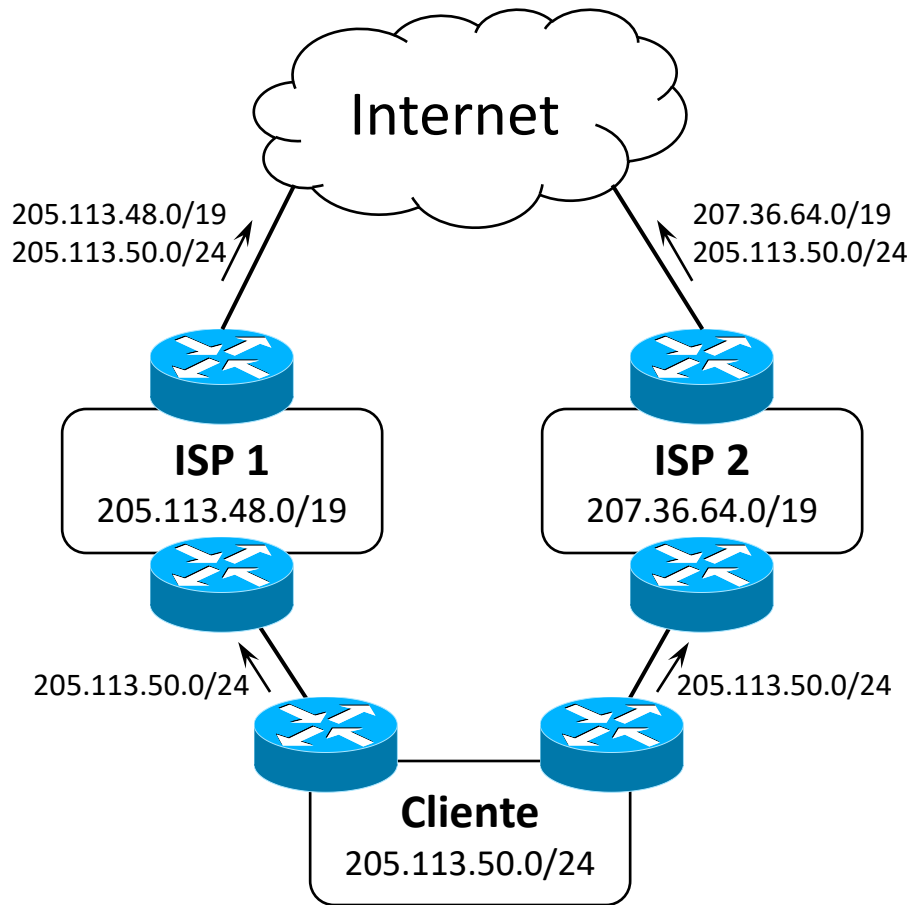
Migração de ISP

- Ao mudar de ISP
 - Novo ISP atribui um bloco diferente de endereços
 - ISP antigo atribui o bloco anteriormente usado a outro cliente
 - Normalmente obrigaria a alterar o endereço IP de cada máquina ☹
- Para manter independência do ISP, a rede pode ser numerada com endereços privados
 - NAT com mapeamento directo (estático) entre endereços privados e públicos
 - E.g., 10.0.0.x é sempre mapeado para 205.113.50.x
- Ao mudar de ISP basta mudar a gama de endereços *inside global*
 - E.g., 10.0.0.x passa a ser mapeado para 207.36.76.x
- Não é preciso alterar o endereço IP de cada máquina ☺

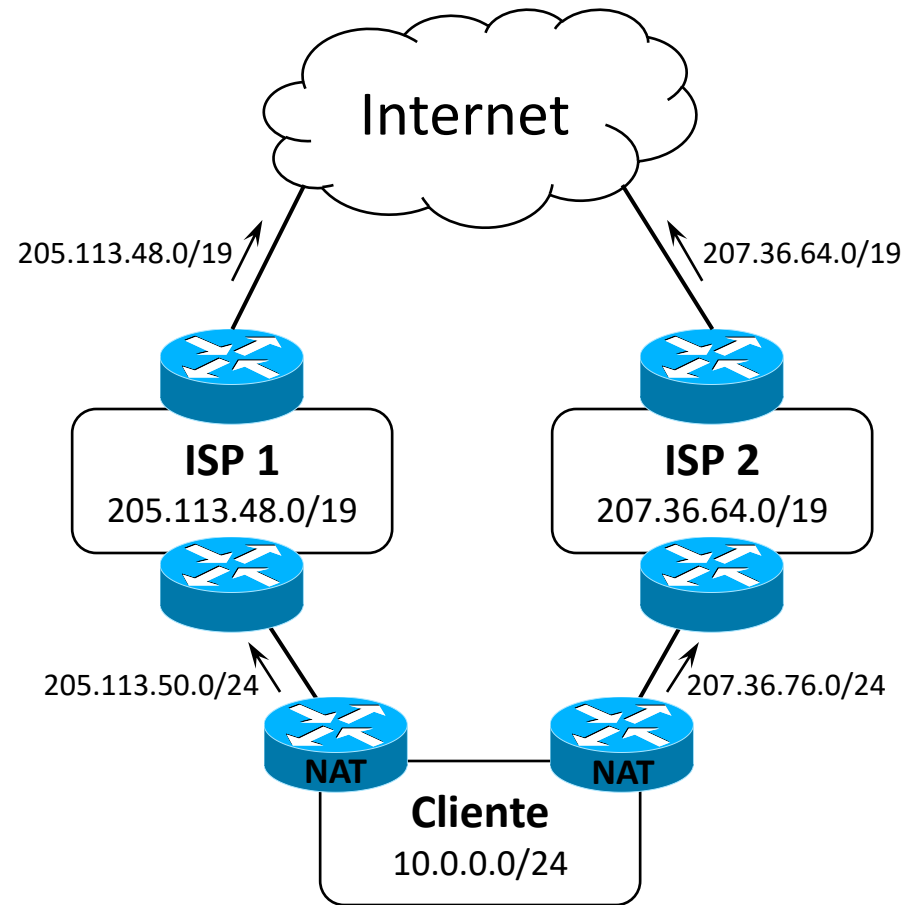
Migração de ISP



Ligação a múltiplos ISP

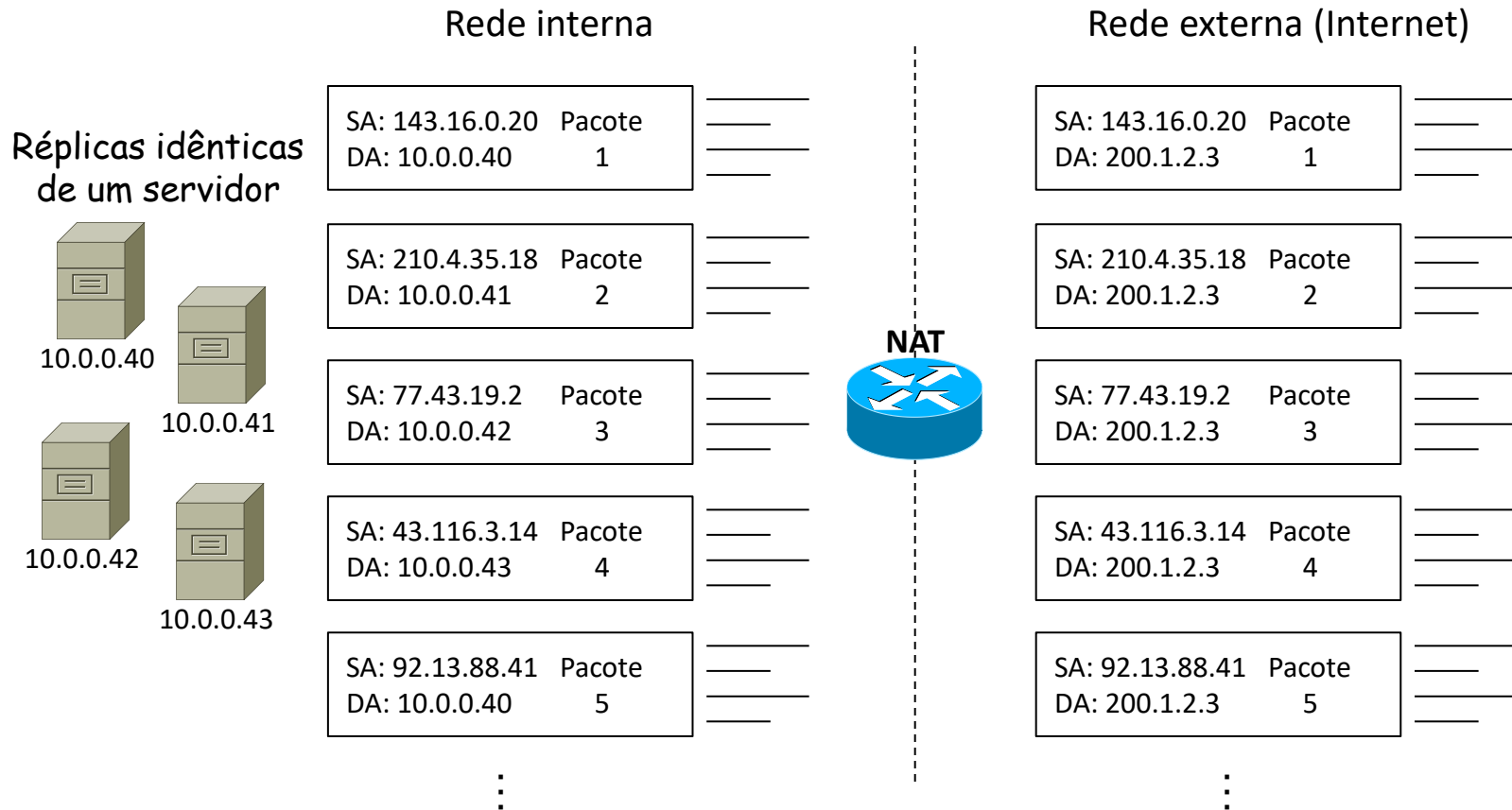


Sem NAT



Com NAT

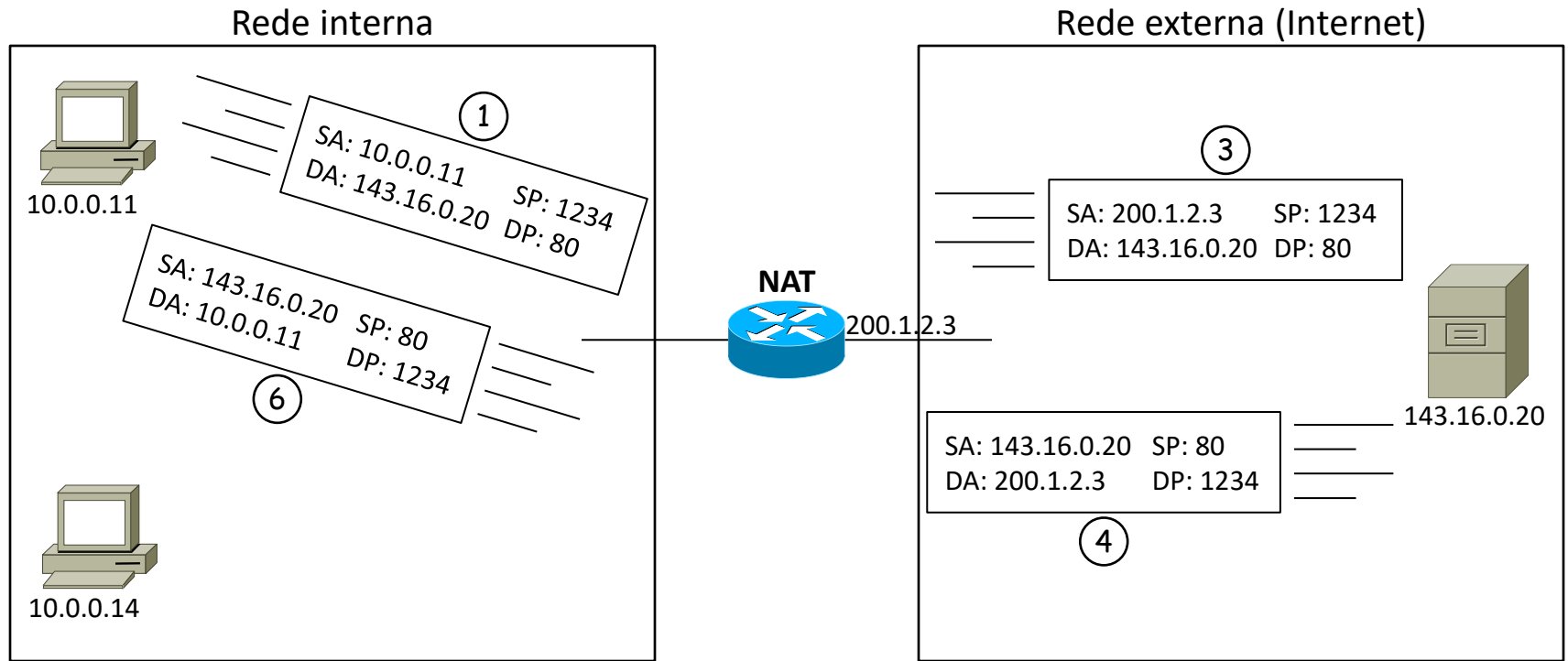
Distribuição de carga



NAT com Tradução de Portas

- Além do endereço IP, traduz-se também a porta (TCP, UDP)
 - Tabela de NAT guarda também essa informação
 - Um único endereço público pode ser usado por muitas máquinas internas em simultâneo
 - Porta permite desambiguar
- Poupança de endereços muito maior que NAT Básico
 - Portas têm 16 bits — um endereço público pode ser utilizado por até 65535 máquinas internas em simultâneo
 - Menos se se usarem apenas portas efémeras
 - Tipo de NAT mais usado em redes caseiras e de pequenas empresas
 - Acesso Internet de baixo custo com um único IP público atribuído

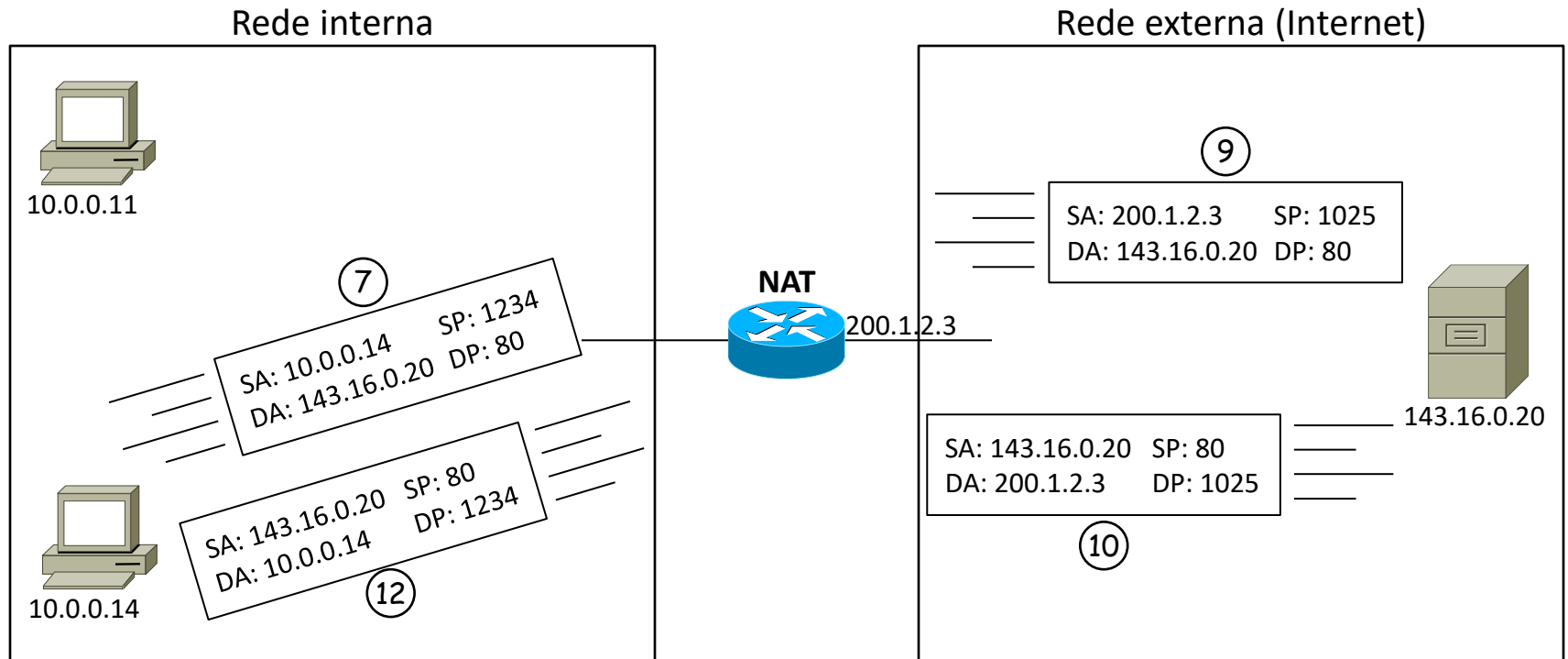
NAT com Tradução de Portas



	Pro	Inside global	Inside local	Outside local	Outside global
②	tcp	200.1.2.3:1234	10.0.0.11:1234	143.16.0.20:80	143.16.0.20:80



NAT com Tradução de Portas



	Pro	Inside global	Inside local	Outside local	Outside global
8	tcp	200.1.2.3:1234	10.0.0.11:1234	143.16.0.20:80	143.16.0.20:80
	tcp	200.1.2.3:1025	10.0.0.14:1234	143.16.0.20:80	143.16.0.20:80

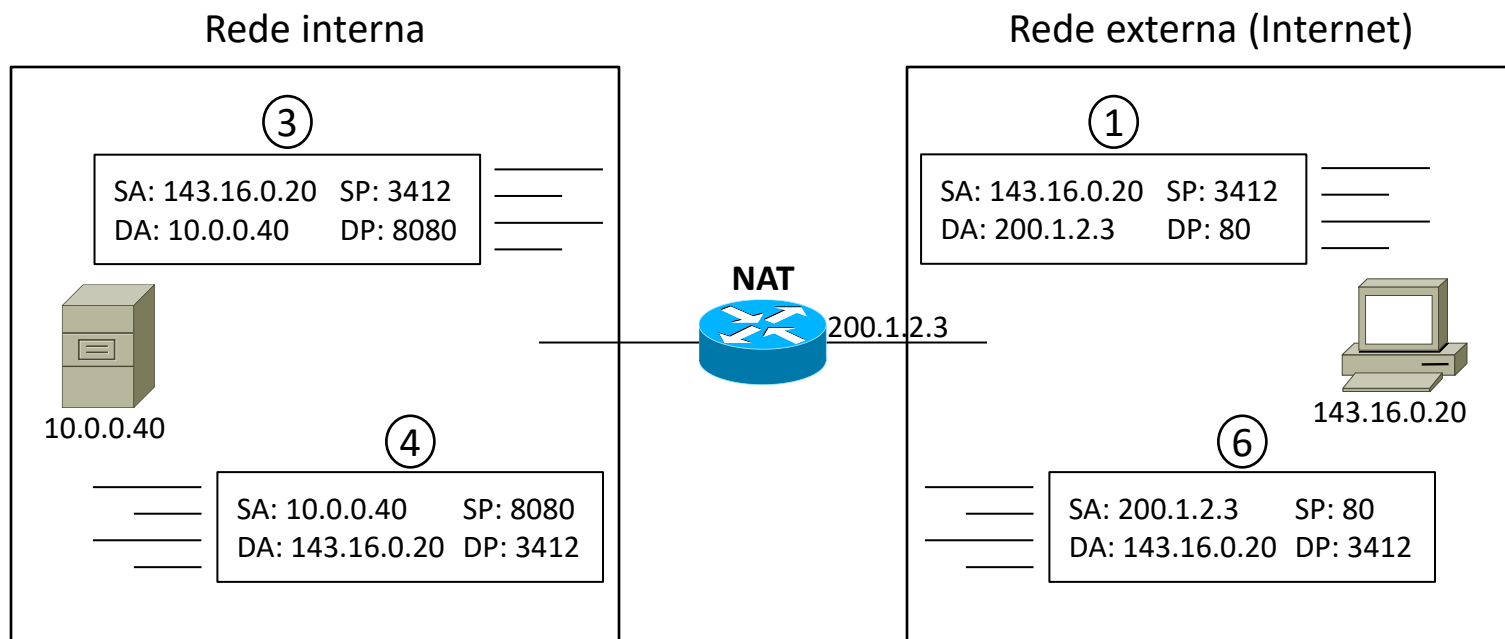
A porta só é traduzida se houver necessidade



Redirecionamento de portas

- No exemplo anterior, as conexões são estabelecidas do interior para o exterior
 - Entradas dinâmicas são criadas na tabela NAT quando sai o primeiro pacote
- Para ter servidores visíveis na Internet é preciso permitir conexões do exterior para o interior
 - Quando chega ao NAT um pacote para um endereço da *pool*, não sabe para que endereço interno o deve traduzir
 - Com *masquerading*, o NAT acha que o pacote é para si próprio
- Solução: entradas estáticas na tabela NAT
 - Pacotes para um IP+porta *inside global* especificado são traduzidos sempre para o mesmo IP+porta *inside local*
 - Normalmente designado *Port Forwarding*

Redirecccionamiento de portas



Pro **Inside global**

tcp 200.1.2.3:80

Inside local

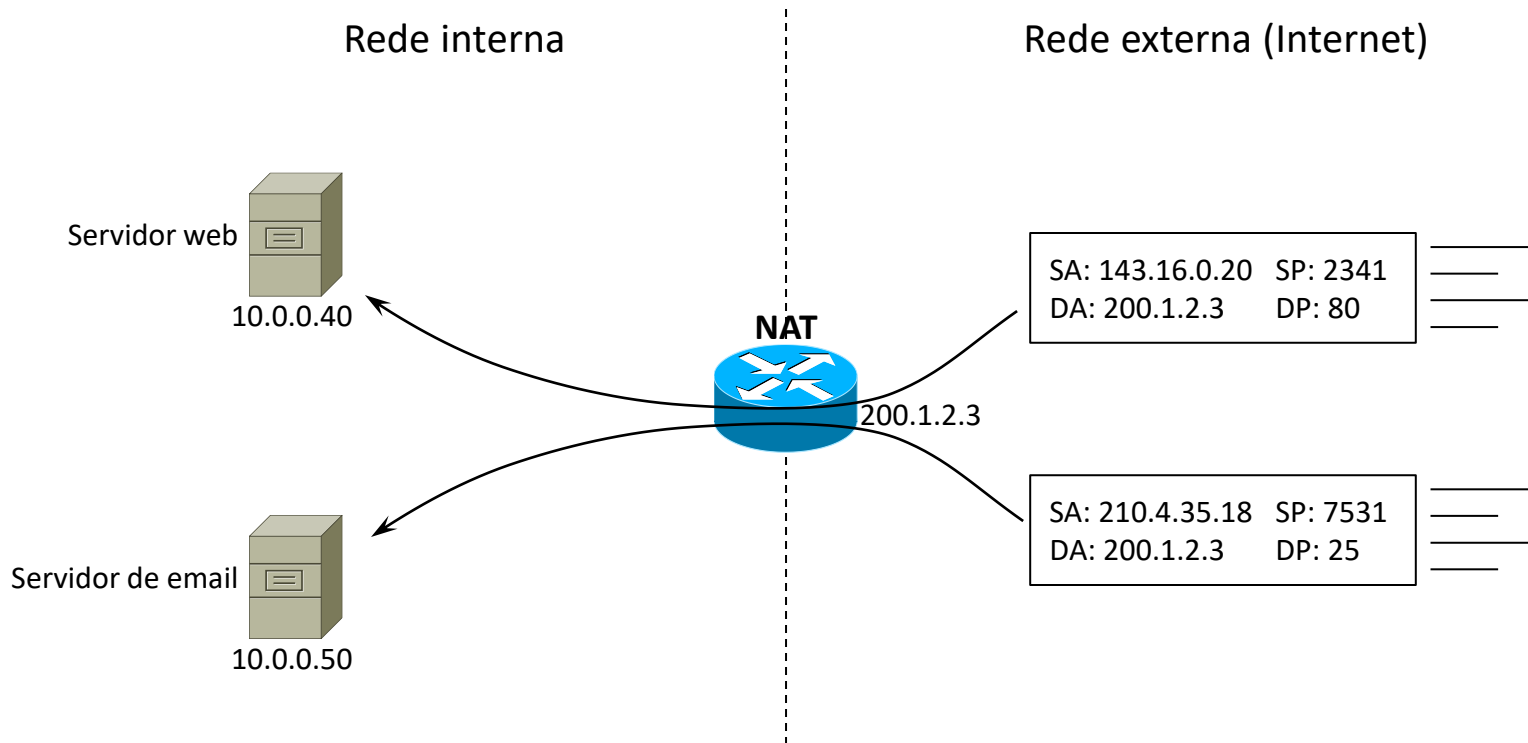
10.0.0.40:8080

Outside local

Outside global



Servidores diferentes com um único endereço IP público

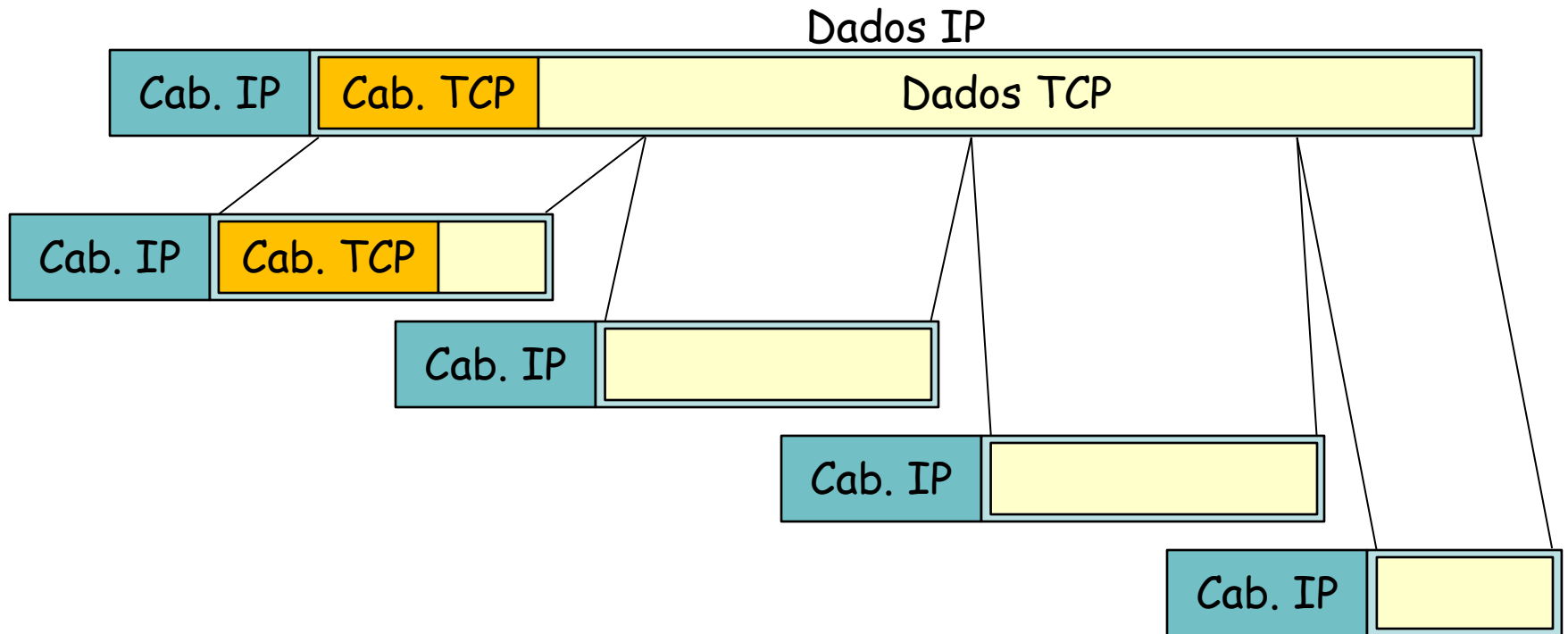


Pro	Inside global	Inside local	Outside local	Outside global
tcp	200.1.2.3:25	10.0.0.50:25	---	---
tcp	200.1.2.3:80	10.0.0.40:80	---	---

NAT e checksums

- *Checksum* do IP calculado sobre todo o cabeçalho, incluindo endereços
 - Se um endereço muda, o *checksum* tem que ser recalculado
 - Pode ser recalculado de forma incremental
- O cálculo do *checksum* do TCP e UDP inclui um pseudo-cabeçalho que inclui os endereços IP
 - Também é necessário recalcular este *checksum*, mesmo com NAT Básico

NAPT e fragmentação IP



- Cabeçalho TCP (ou UDP) só no primeiro fragmento
- Outros fragmentos não têm informação sobre portas
- Fragmentação IP incompatível com NAPT

NAT e cifragem

- Alguns protocolos incluem informação sobre endereços IP nas mensagens
- Para o NAT funcionar com esses protocolos, é necessário alterar o conteúdo das mensagens
- Se elas estiverem cifradas, não podem ser alteradas
 - Se estiverem protegidas por *hash* criptográfico também não
- Nalguns tipos de VPN (e.g., com alguns modos do IPSec), alterando um endereço IP o pacote fica corrompido
 - NAT tem que ser feito fora do túnel protegido

NAT e segurança

- Há quem acredite que o uso de NAT acrescenta alguma segurança às redes
 - Esconde endereçamento interno
 - Nalguns tipos de NAT, evita estabelecimento de conexões de fora para dentro
- Estes aspectos podem atrasar um atacante, mas não o impedem totalmente
- Evitar conexões de fora para dentro também se faz com *firewall* (dispositivo especificamente para segurança)
- Segurança é uma "má" razão para usar NAT...

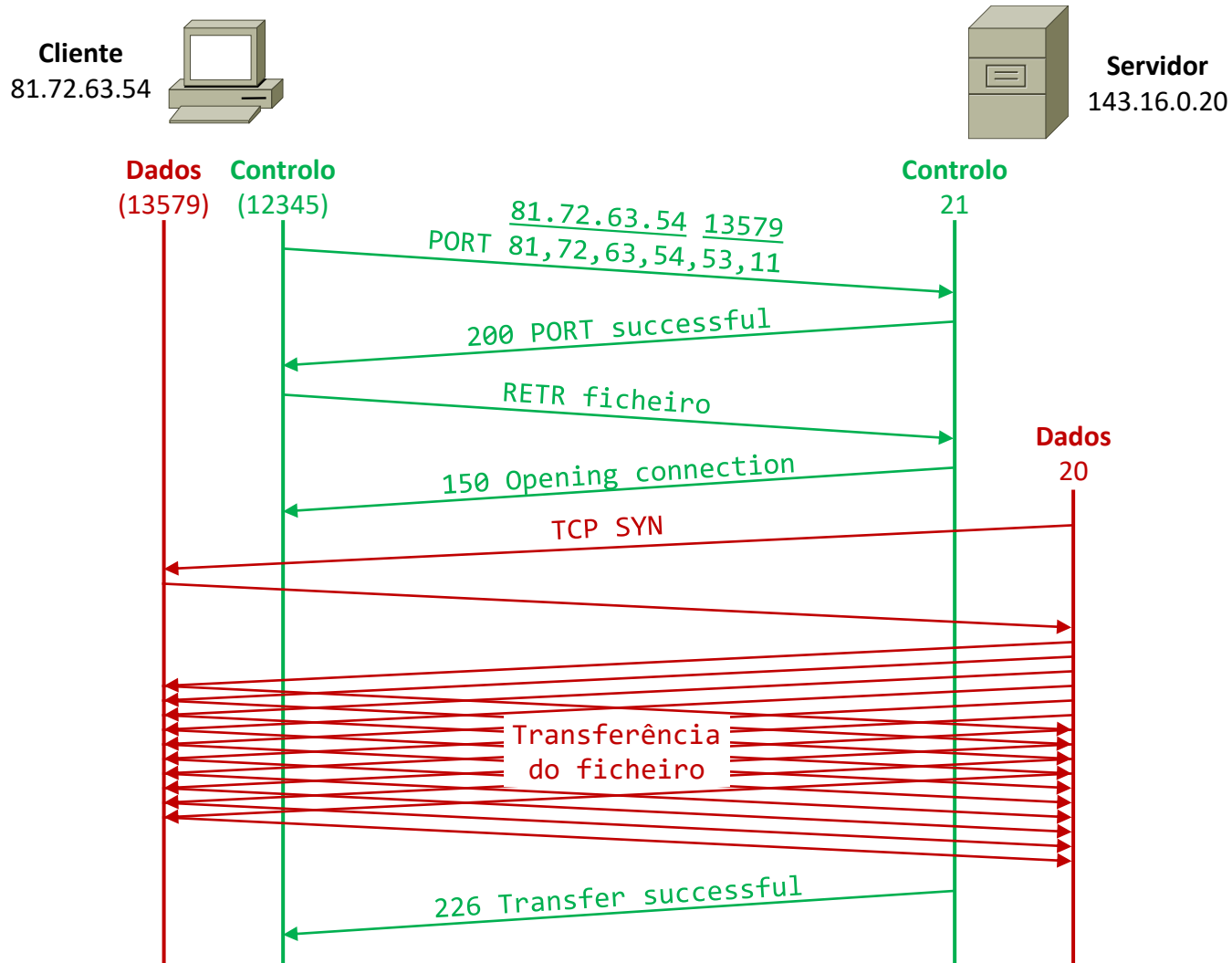
Compatibilidade com protocolos

- Há aspectos de alguns protocolos que trazem problemas de compatibilidade com o NAT
 - Inclusão de endereços IP nas mensagens
 - Obriga a interceptar a conexão e alterar as mensagens de acordo com o NAT
 - Uso de múltiplas conexões relacionadas
 - Obriga a interceptar a conexão e interpretar as mensagens para associar as diferentes conexões
 - Estabelecimento de conexões de fora para dentro do NAT
 - Obriga à criação prévia de entradas para as portas em questão (com base na sinalização)
- Cisco IOS inclui este tipo de suporte para uma série de protocolos
- No Linux existem módulos que implementam a funcionalidade necessária e o suporte para protocolos específicos

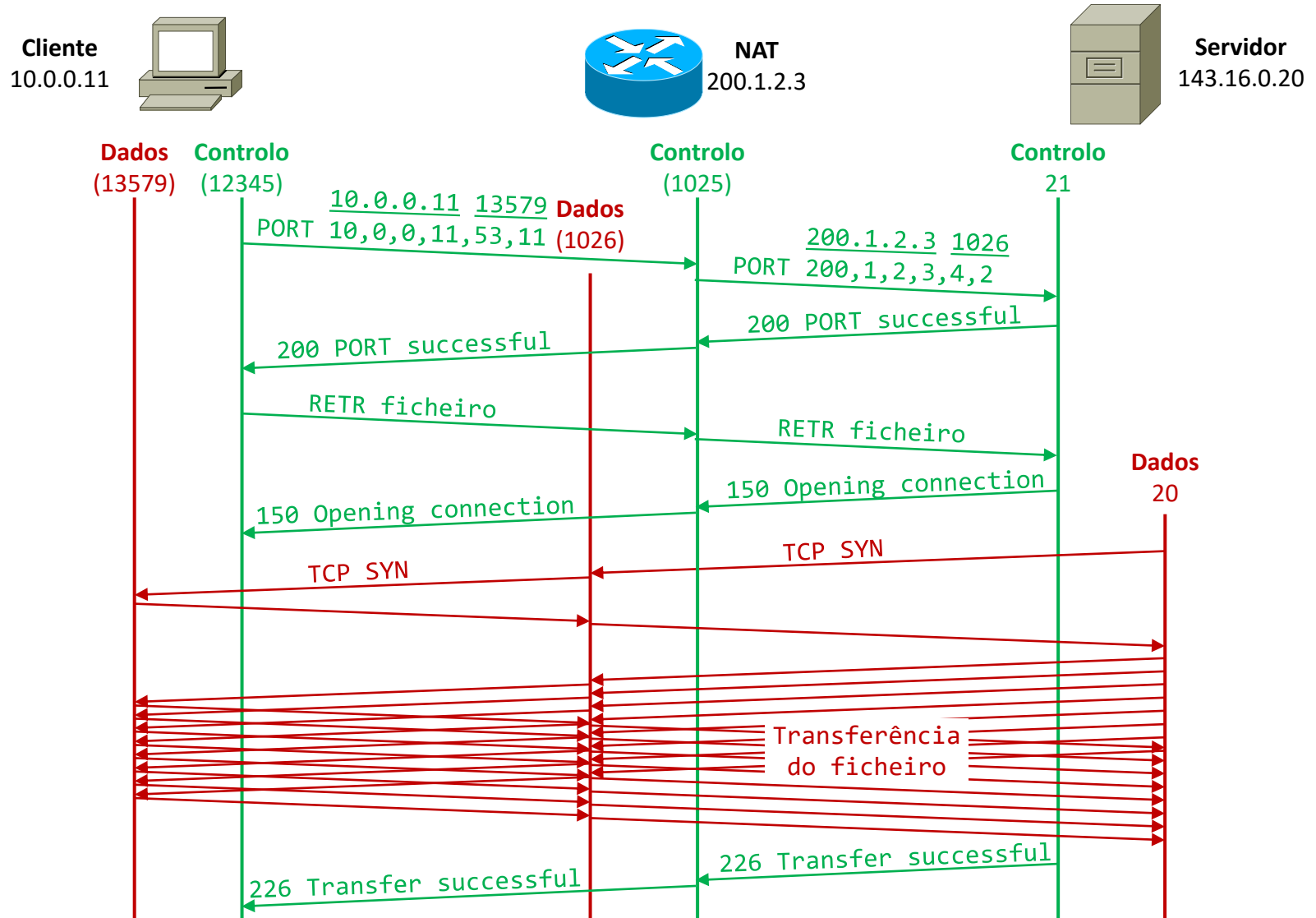
Alguns protocolos com questões de compatibilidade com o NAT

- ICMP
 - Alguns tipos de mensagem ICMP (e.g., *Unreachable*) incluem um excerto do pacote que lhes deu origem
 - Para o NAT ser transparente, é necessário traduzir os endereços IP desses excertos
- DNS
 - Um servidor de DNS na rede interna responde com endereços *inside local*
 - É preciso traduzi-los para *inside global* se o pedido DNS vier de fora
- FTP
 - Usa conexões separadas (mas relacionadas) para controlo e dados
 - Na conexão de controlo, indica endereços IP e portas para estabelecer conexões de dados
 - Em modo activo, as conexões de dados são estabelecidas do servidor para o cliente
- Protocolos de encaminhamento em geral
 - Não funcionam com NAT pelo meio — NAT só pode estar na fronteira

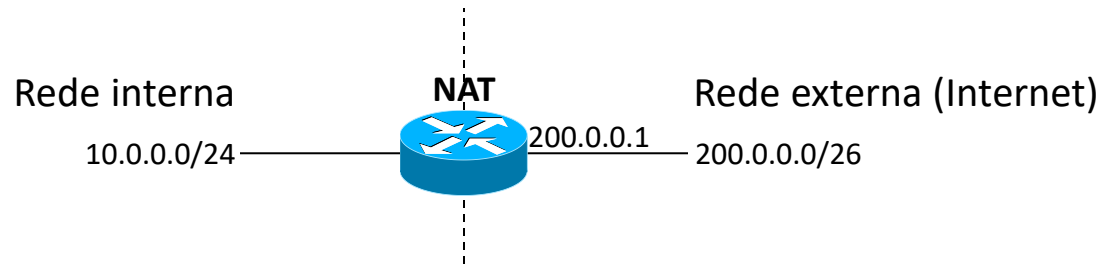
FTP (modo activo)



FTP (modo activo) com NAT

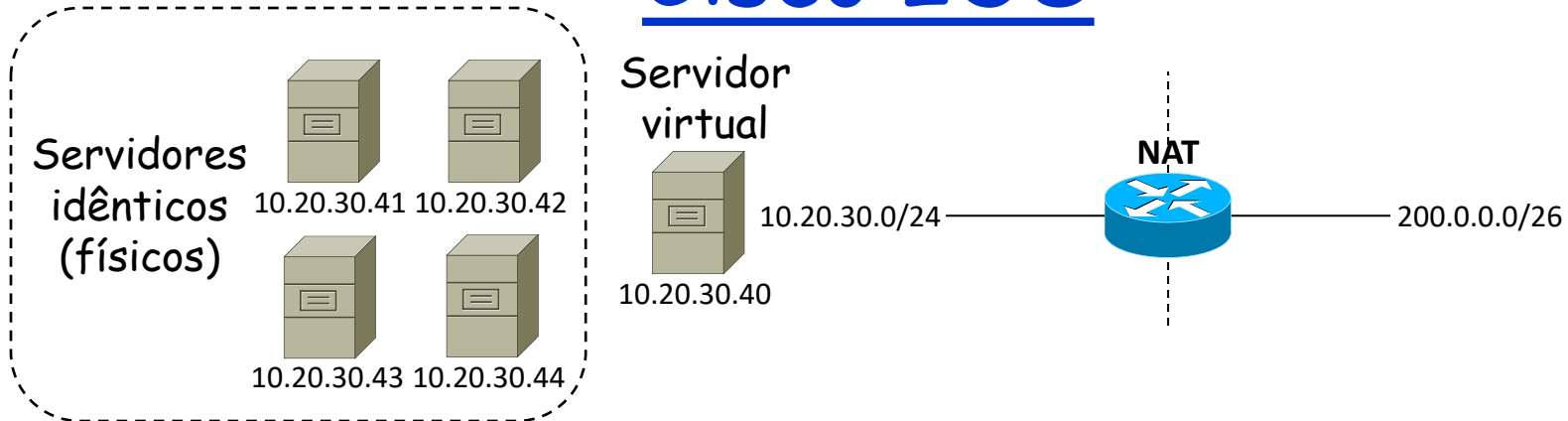


NAT Básico em Cisco IOS



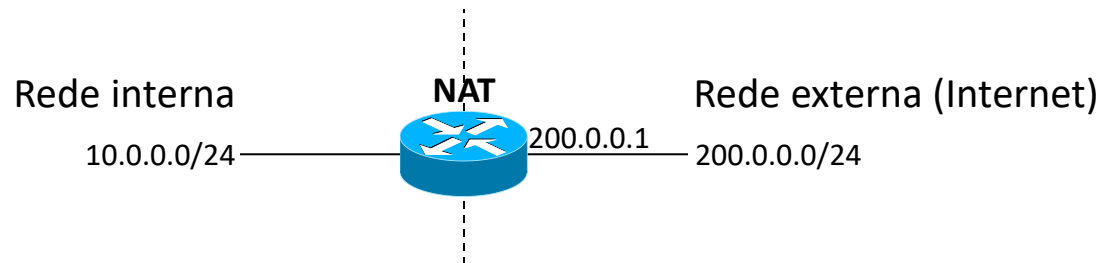
```
interface FastEthernet 1/0
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
!
interface FastEthernet 1/1
 ip address 200.0.0.1 255.255.255.192
 ip nat outside
!
ip nat pool endpub 200.0.0.2 200.0.0.40 prefix-length 26
ip nat inside source list 10 pool endpub
!
access-list 10 permit 10.0.0.0 0.0.0.255
!
```

Distribuição de carga com NAT em Cisco IOS



```
interface FastEthernet 1/0
ip address 10.20.30.1 255.255.255.0
ip nat inside
!
interface FastEthernet 1/1
ip address 200.0.0.1 255.255.255.192
ip nat outside
!
ip nat pool servers 10.20.30.41 10.20.30.44 prefix-length 29 type rotary
ip nat inside destination list 10 pool servers
!
access-list 10 permit host 10.20.30.40
!
```

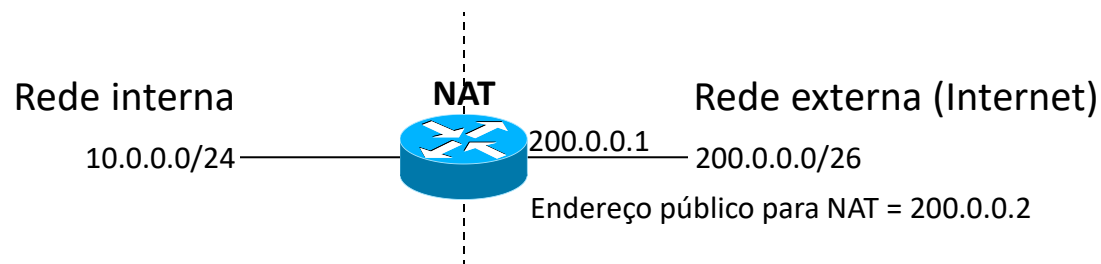
NAT Básico com mapeamento estático (netmap) em Cisco IOS



```
interface FastEthernet 1/0
  ip address 10.0.0.1 255.255.255.0
  ip nat inside
!
interface FastEthernet 1/1
  ip address 200.0.0.1 255.255.255.0
  ip nat outside
!
ip nat inside source static network 10.0.0.0 200.0.0.0 /24
!
```

É necessário que a gama de endereços públicos tenha o mesmo tamanho que a rede a ser traduzida

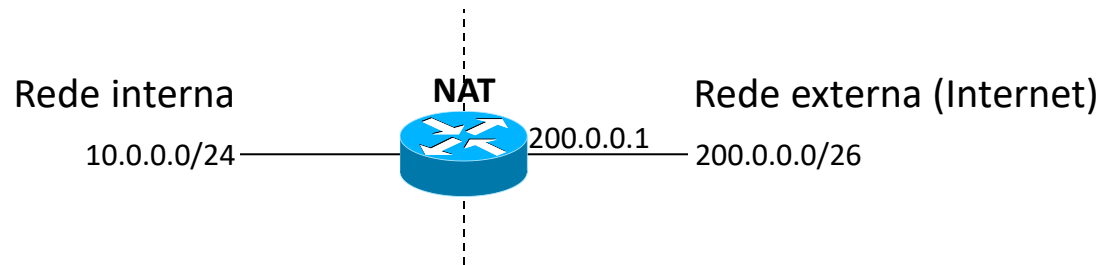
NAPT (PAT) em Cisco IOS



```
interface FastEthernet 1/0
  ip address 10.0.0.1 255.255.255.0
  ip nat inside
!
interface FastEthernet 1/1
  ip address 200.0.0.1 255.255.255.192
  ip nat outside
!
ip nat pool endpub 200.0.0.2 200.0.0.2 prefix-length 26
ip nat inside source list 10 pool endpub overload
!
access-list 10 permit 10.0.0.0 0.0.0.255
!
```

Só um endereço público para NAT, diferente do endereço da interface do *router*

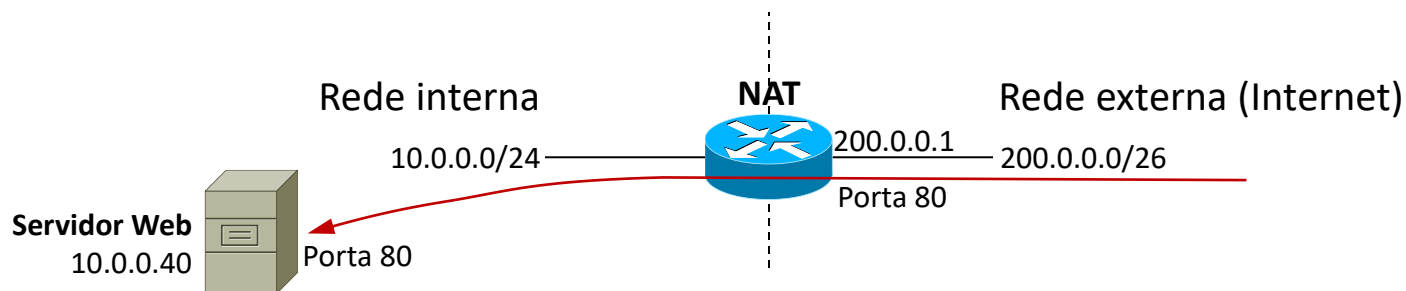
Masquerading* em Cisco IOS



```
interface FastEthernet 1/0
  ip address 10.0.0.1 255.255.255.0
  ip nat inside
!
interface FastEthernet 1/1
  ip address 200.0.0.1 255.255.255.192
  ip nat outside
!
ip nat inside source list 10 interface FastEthernet 1/1 overload
!
access-list 10 permit 10.0.0.0 0.0.0.255
!
```

*NAPT usando o endereço que, no momento, estiver atribuído à interface indicada (masquerading é um termo originário do Linux)

Port Forwarding em Cisco IOS



```
interface FastEthernet 1/0
ip address 10.0.0.1 255.255.255.0
ip nat inside
!
interface FastEthernet 1/1
ip address 200.0.0.1 255.255.255.192
ip nat outside
!
ip nat inside source list 10 interface FastEthernet 1/1 overload
ip nat inside source static tcp 10.0.0.40 80 interface FastEthernet 1/1 80
!
access-list 10 permit 10.0.0.0 0.0.0.255
!
```

Debugging do NA(P)T em Cisco IOS

```
Router#debug ip nat detailed
```

```
1: NAT: Allocated Port for 10.0.0.11 -> 200.0.0.1: wanted 34567 got 34567
2: NAT*: i: tcp (10.0.0.11, 34567) -> (193.136.39.12, 80) [30470]
3: NAT*: s=10.0.0.11->200.0.0.1, d=193.136.39.12 [30470]
4: NAT*: o: tcp (193.136.39.12, 80) -> (200.0.0.1, 34567) [0]
5: NAT*: s=193.136.39.12, d=200.0.0.1->10.0.0.11 [0]
6: NAT*: i: tcp (10.0.0.11, 34567) -> (193.136.39.12, 80) [30471]
7: NAT*: s=10.0.0.11->200.0.0.1, d=193.136.39.12 [30471]
8: NAT*: i: tcp (10.0.0.11, 34567) -> (193.136.39.12, 80) [30472]
9: NAT*: s=10.0.0.11->200.0.0.1, d=193.136.39.12 [30472]
10: NAT*: o: tcp (193.136.39.12, 80) -> (200.0.0.1, 34567) [36412]
11: NAT*: s=193.136.39.12, d=200.0.0.1->10.0.0.11 [36412]
```

NOTAS

Linha 1: Porta usada internamente estava livre no NAT, não é necessário traduzi-la; entrada criada na tabela NAT

Linha 2: Pacote recebido na interface interna (daí a linha anterior)

Linha 3: Tradução do endereço de origem nesse pacote (o de destino mantém-se)

Linha 4: Pacote recebido na interface externa

Linha 5: Tradução do endereço de destino nesse pacote (o de origem mantém-se)

Debugging do NA(P)T em Cisco IOS

```
Router#debug ip nat detailed
```

```
1: NAT: Allocated Port for 10.0.0.50 -> 200.0.0.1: wanted 34567 got 1026
2: NAT*: i: tcp (10.0.0.50, 34567) -> (193.136.39.12, 80) [62942]
3: NAT*: TCP s=34567->1026, d=80
4: NAT*: s=10.0.0.50->200.0.0.1, d=193.136.39.12 [62942]
5: NAT*: o: tcp (193.136.39.12, 80) -> (200.0.0.1, 1026) [0]
6: NAT*: TCP s=80, d=1026->34567
7: NAT*: s=193.136.39.12, d=200.0.0.1->10.0.0.50 [0]
8: NAT*: i: tcp (10.0.0.50, 34567) -> (193.136.39.12, 80) [62943]
9: NAT*: TCP s=34567->1026, d=80
10: NAT*: s=10.0.0.50->200.0.0.1, d=193.136.39.12 [62943]
```

NOTAS

Linha 1: Neste caso a porta usada internamente já estava ocupada no NAT, é preciso usar outra e fazer a tradução

Linha 3: Tradução da porta de origem (de 34567 para 1026) no pacote que sai

Linha 6: Tradução da porta de destino (de 1026 para 34567) no pacote que entra

Outros comandos úteis

```
Router#show ip nat translations
```

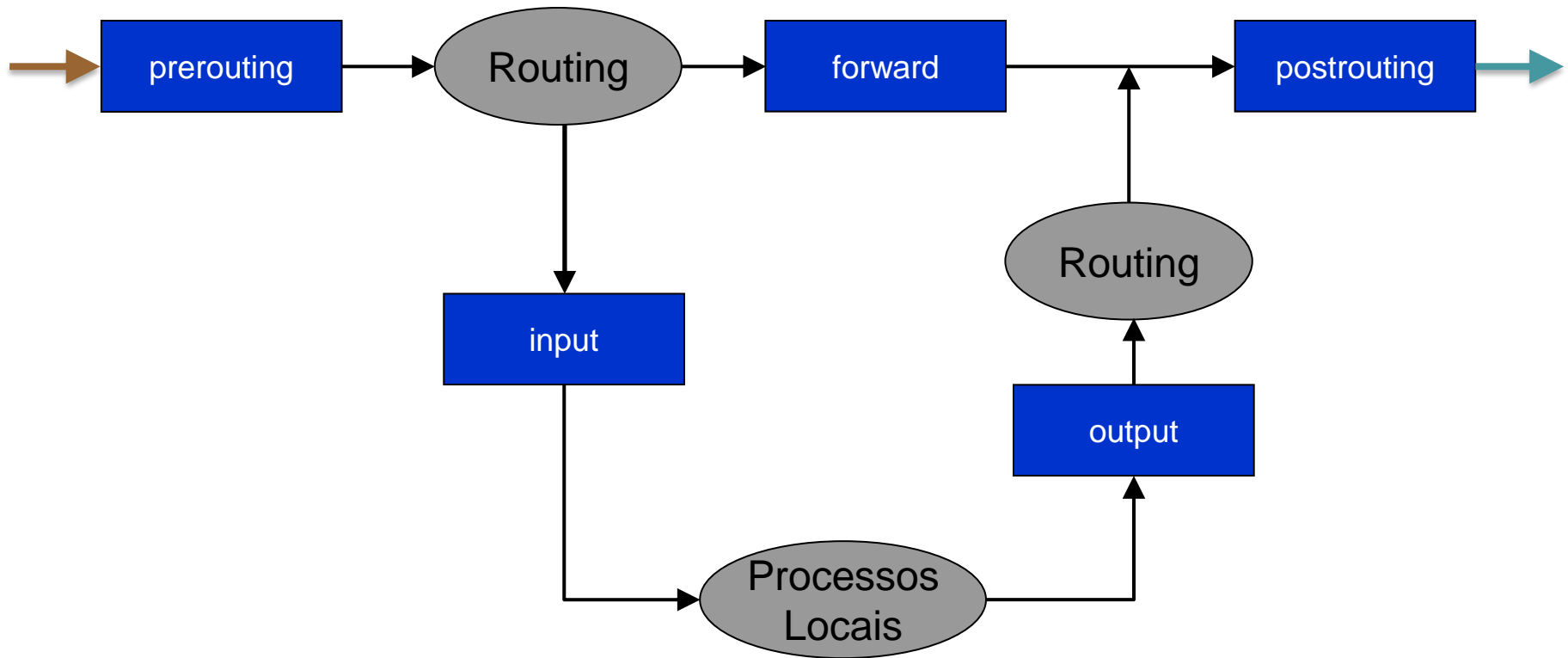
Pro	Inside global	Inside local	Outside local	Outside global
icmp	200.0.0.1:4358	10.0.0.11:4358	133.16.8.1:4358	133.16.8.1:4358
tcp	200.0.0.1:1026	10.0.0.50:34567	193.136.39.12:80	193.136.39.12:80
tcp	200.0.0.1:34567	10.0.0.11:34567	193.136.39.12:80	193.136.39.12:80
tcp	200.0.0.1:80	10.0.0.40:80	---	---
---	200.0.0.2	10.0.0.7	---	---




```
Router#show access-lists
```

```
Standard IP access list 10
```

```
10 permit 10.0.0.0, wildcard bits 0.0.0.255
```

Percursos dos pacotes no *kernel* Linux



-  Netfilter *hooks* (pontos onde se podem “apanhar” pacotes)
-  Pacotes recebidos da rede
-  Pacotes enviados para a rede

nftables

- *Framework* para filtragem de pacotes no Linux
- Usado para implementar *firewalls* e NAT
- Substitui o iptables e funcionalidades relacionadas
- <https://wiki.nftables.org/>

nftables

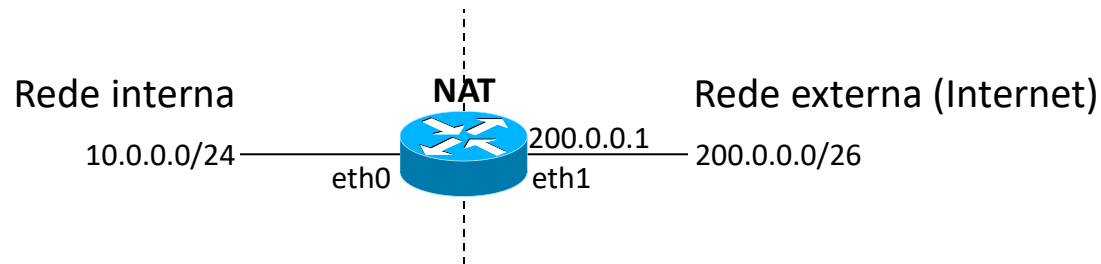
- Baseado em **tabelas** aplicáveis a **famílias** de pacotes (ip, ip6, inet, ...) e que contêm **cadeias (chains)** de **regras** a aplicar
- As **cadeias** podem estar associadas a Netfilter **hooks**
 - Neste caso chamam-se **base chains**
 - Um valor numérico de **prioridade** define a ordem de tratamento das **cadeias** num **hook** (valor numérico mais baixo → mais prioritária)
- **Base chains** podem ser de diferentes tipos (filter, route, nat) consoante a aplicação
- Uma **regra** apanha pacotes com determinadas características e define um ou mais **veredictos** a aplicar-lhes, que podem ser
 - Tratamentos predefinidos
 - E.g., snat (*source nat*), dnat (*destination nat*), masquerade, redirect
 - Passar o pacote a outra **chain** (jump ou goto)
- É possível criar mapas para valores (**map**) ou veredictos (**vmap**)

nftables

- O nftables não tem tabelas nem cadeias predefinidas
- As configurações nos *slides* seguintes assumem a configuração prévia de
 - Uma tabela nat
 - Cadeias prerouting e postrouting nessa tabela associadas aos *hooks* com o mesmo nome

```
table ip nat {  
    chain prerouting { type nat hook prerouting priority -100; }  
    chain postrouting { type nat hook postrouting priority 100; }  
}
```

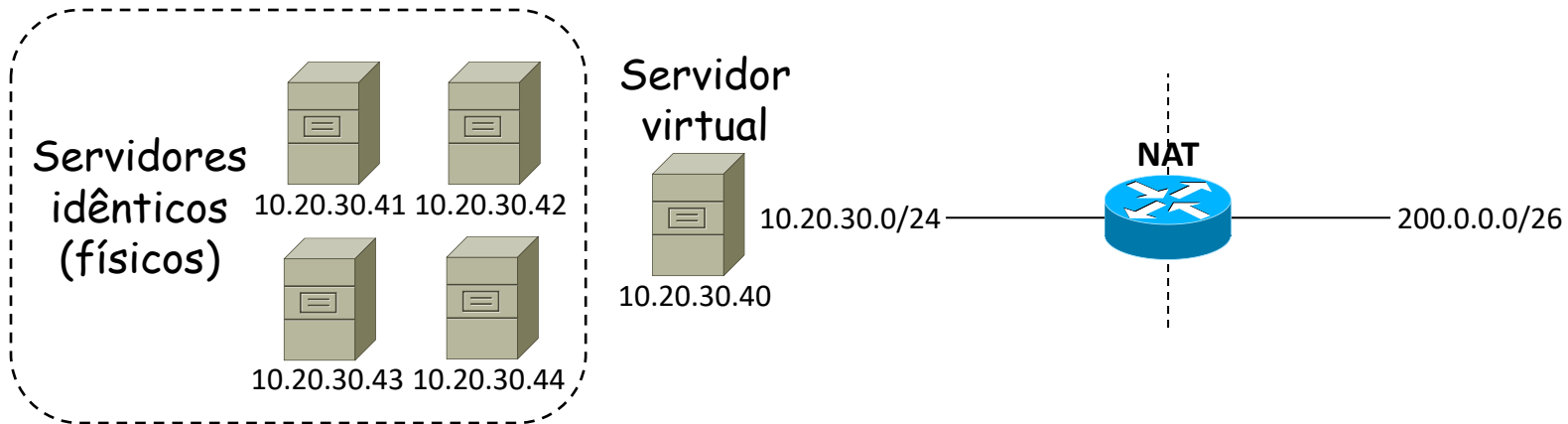
NAT Básico em Linux



```
nft add rule ip nat postrouting oif eth1 ip saddr 10.0.0.0/24 \
  snat to 200.0.0.2-200.0.0.40
```

- Se a dado momento a gama de endereços inside global não for suficientemente grande, começa a ser feita também tradução das portas (ou seja, NATPT)
- A regra é aplicada ao primeiro pacote de um fluxo; pacotes subsequentes (nos dois sentidos) são traduzidos de forma consistente com o primeiro (isto é válido para todas as variantes de NAT)
- Source NAT tem que ser feito na cadeia postrouting e Destination NAT tem que ser feito na cadeia prerouting

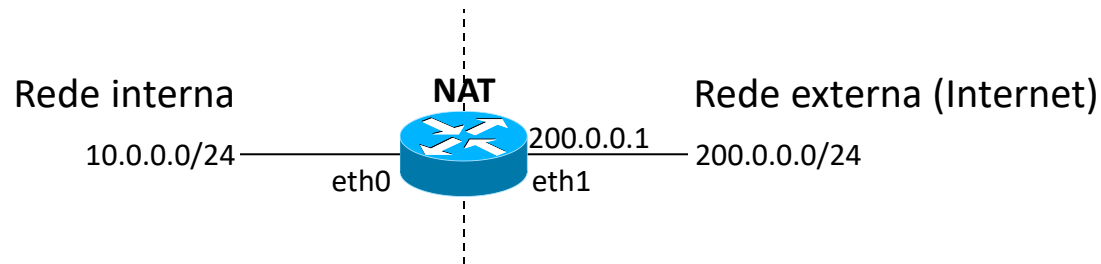
Distribuição de carga com NAT em Linux



```
nft add rule ip nat prerouting ip daddr 10.20.30.40 dnat to \  
  numgen inc mod 4 map { \  
    0: 10.20.30.41, \  
    1: 10.20.30.42, \  
    2: 10.20.30.43, \  
    3: 10.20.30.44  \  
  } \  
}
```

- numgen inc mod 4 cria números sequenciais mod 4: (0, 1, 2, 3, 0, 1, ...)
- Esses números são mapeados para os endereços dos servidores físicos
- O resultado é uma distribuição *round robin* de conexões pelos diferentes servidores

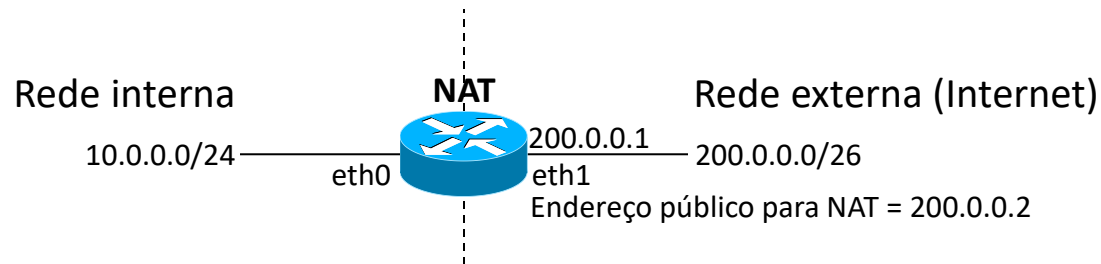
NAT Básico com mapeamento estático (netmap) em Linux



```
nft add rule ip nat prerouting 'iif eth1 ip daddr 200.0.0.0/24
  dnat prefix to 10.0.0.0/24'
nft add rule ip nat postrouting 'oif eth1 ip saddr 10.0.0.0/24 \
  snat prefix to 200.0.0.0/24'
```

- Aplica uma *wildcard mask* ao endereço para obter apenas a parte de *host* e faz *bitwise or* com o prefixo para o qual quer mapear
- Uma regra para conexões iniciadas no interior (clientes) e outra para conexões iniciadas no exterior (servidores)
- Se a gama de endereços inside global pertencer à sub-rede da interface externa é ainda necessário activar Proxy ARP
- Entre plicas para evitar que a *bash* interprete erradamente os caracteres especiais '&' e '|'

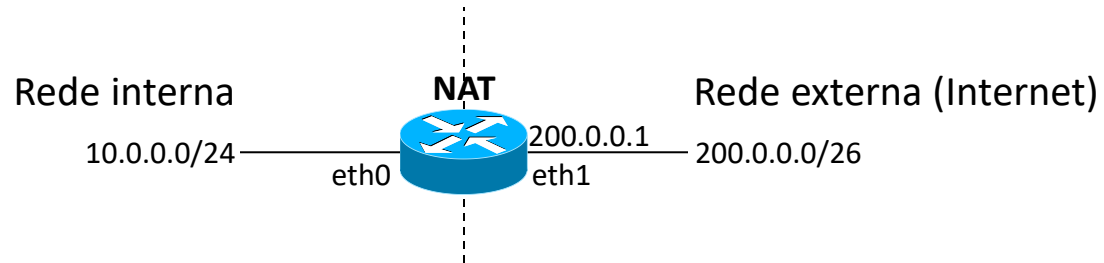
NAPT (PAT) em Linux



```
nft add rule ip nat postrouting oif eth1 ip saddr 10.0.0.0/24 snat to 200.0.0.2
```

- Só um endereço público para NAT, diferente do endereço da interface do *router*
- Se houver mais de uma máquina a tentar aceder ao exterior usando a mesma porta de origem, começa a ser feita também tradução das portas (ou seja, NAPT)

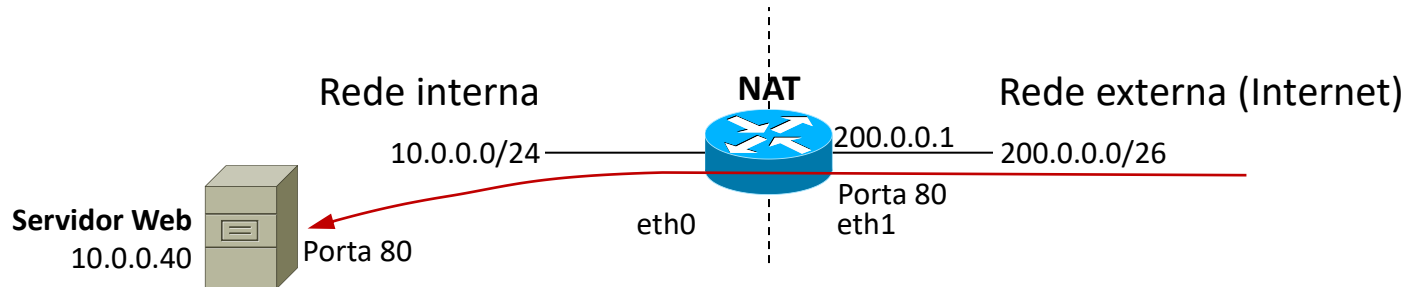
Masquerading em Linux



```
nft add rule ip nat postrouting oif eth1 ip saddr 10.0.0.0/24 masquerade
```

- É usado para NAT o endereço que, no momento, estiver atribuído à interface indicada

Port Forwarding em Linux



```
nft add rule ip nat postrouting oif eth1 ip saddr 10.0.0.0/24 masquerade  
nft add rule ip nat prerouting iif eth1 tcp dport 80 dnat to 10.0.0.30:80
```

- Se a porta *inside local* for a mesma que a *inside global*, pode omitir-se o :80
- A primeira regra é para *masquerading* e é desnecessária se não forem ser estabelecidas conexões de dentro para fora

Persistência das regras nftables

- As regras criadas nos slides anteriores perdem-se num *reboot*
- Para as tornar persistentes pode-se
 - Activar o serviço nftables: `systemctl enable nftables`
 - Guardar as regras num ficheiro e referenciá-lo na configuração do serviço:

```
nft list table nat > /etc/nftables/myNATtable.nft  
echo 'include "/etc/nftables/myNATtable.nft"' >> /etc/sysconfig/nftables.conf
```

Alguns comandos úteis

- Listar tabela nat
`nft list table nat`
- Idem, mostrando os *handles*
`nft -a list table nat`
- Apagar uma regra
`nft delete rule ip nat chain handle handle`
- Limpar todas as regras de NAT
`nft flush table nat`