

RCI 2024/2025

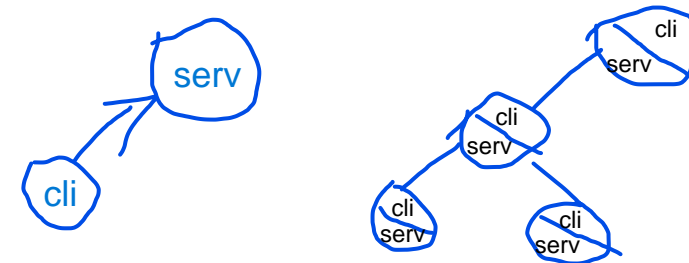
Redes de Dados Identificados por Nome

# Rede NDN - Rede de Dados Identificados por Nome

- Uma *rede de dados identificados por nome* (*named data network*, NDN) é um sistema distribuído que tem por objetivo disponibilizar objetos de dados globalmente a todos os seus membros.
- Cada objeto está associado a um nome globalmente único.
- Cópias de um mesmo objeto podem estar distribuídas por nós distintos da rede.
- Um utilizador que pretenda obter determinado objeto lança uma mensagem de interesse na rede com o nome do objeto.
- Um nó que tenha armazenada uma cópia do objeto responde à mensagem de interesse com uma mensagem de objeto contendo o objeto e o seu nome.
- A resposta é enviada sobre a mesma interface pela qual a mensagem de interesse foi recebida.
- Um nó que não tenha armazenada uma cópia do objeto reencaminha a mensagem de interesse para os seus vizinhos na rede.
- Um nó intermédio que receba uma mensagem de objeto reencaminha-a pela mesma interface pela qual recebeu anteriormente a correspondente mensagem de interesse e armazena o objeto em cache.

## Características de uma rede NDN:

- Os objetos têm nomes que são identificadores *anycast*.
- Os nós não necessitam de identificadores.
- Não há necessidade de um diretório que associe nomes a identificadores, como no caso da internet e do DNS.
- Os objetos viajam na rede em sentido inverso ao das mensagens de interesse e vão ficando disponíveis nos nós por onde passam.



# Rede de sobreposição

Para simplificar, restringe-se a topologia de adjacências a uma árvore

## Comandos da interface de utilizador da aplicação

Comando	Descrição
<b>join (j) <i>net</i></b>	Entrada do nó na rede <b><i>net</i></b> . Os valores de <b><i>net</i></b> são representados por três dígitos, podendo variar entre <b>000</b> e <b>999</b> .
<b>direct join (dj) <i>net conIP conTCP</i></b>	Entrada do nó na rede <b><i>net</i></b> , ligando-se ao nó com identificador <b><i>conIP</i></b> <b><i>conTCP</i></b> , sem comunicação com o servidor de nós. Se <b><i>conIP</i></b> for <b>0.0.0.0</b> , então a rede é criada com apenas o nó.
<b>create (c) <i>name</i></b>	Criação de um objeto com nome <b><i>name</i></b> . Os valores de <b><i>name</i></b> são representados por sequências alfanuméricas com um máximo de 100 caracteres. Para simplificar, cria-se apenas o nome do objeto.
<b>delete (dl) <i>name</i></b>	Remoção do objeto com nome <b><i>name</i></b> .
<b>retrieve (r) <i>name</i></b>	Pesquisa do objeto com nome <b><i>name</i></b> .
<b>show topology (st)</b>	Visualização dos identificadores dos vizinho externo, vizinho de salvaguarda e vizinhos internos.
<b>show names (sn)</b>	Visualização dos nomes dos objetos guardados no nó.
<b>show interest table (si)</b>	Visualização de todas as entradas da tabela de interesses pendentes.
<b>leave (l)</b>	Saída do nó da rede.
<b>exit (x)</b>	Fecho da aplicação.

cada nó corre a aplicação

cada nó tem sessões TCP estabelecidas com os seus vizinhos

cada nó é identificado pelo endereço IP da máquina que aloja a aplicação e pelo porto TCP de escuta

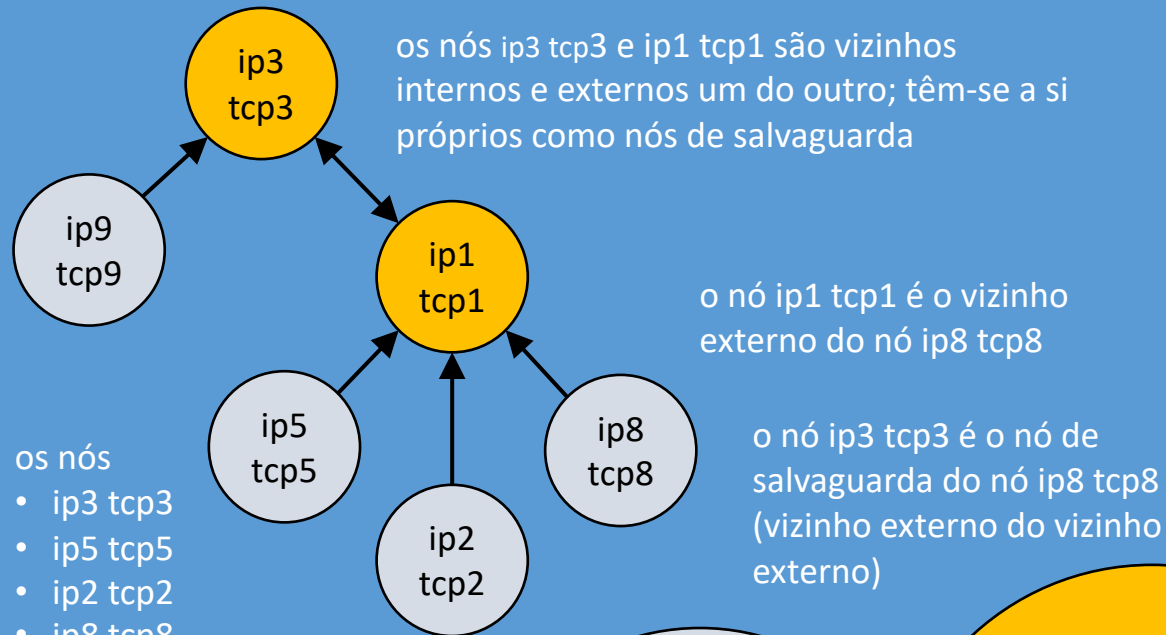
## Invocação da aplicação

**ndn** **cache** **IP** **TCP** **regIP** **regUDP**

nome da aplicação  
tamanho da cache  
endereço IP da máquina que aloja a aplicação  
porto TCP do servidor da aplicação  
endereço IP e porto UDP do servidor de nós

# Topologia em árvore – informação de estado

## Exemplo



os nós

- ip3 tcp3
- ip5 tcp5
- ip2 tcp2
- ip8 tcp8

são vizinhos internos do nó ip1 tcp1

cada nó é identificado pelo endereço IP da máquina que aloja a aplicação e pelo porto TCP de escuta

**ip8 tcp8**

$id = ip8\ tcp8$   
 $vzext = ip1\ tcp1$   
 $vzsalv = ip3\ tcp3$   
 $intr = \emptyset$

**ip1 tcp1**

$id = ip1\ tcp1$   
 $vzext = ip3\ tcp3$   
 $vzsalv = ip1\ tcp1$   
 $intr = \{ip3\ tcp3, ip5\ tcp5, ip2\ tcp2, ip8\ tcp8\}$

**ip3 tcp3**

$id = ip3\ tcp3$   
 $vzext = ip1\ tcp1$   
 $vzsalv = ip3\ tcp3$   
 $intr = \{ip1\ tcp1, ip9\ tcp9\}$

Variáveis:

- $id$  - identificador do nó
- $vzext$  -  $id$  do vizinho externo, inicialmente  $vzext := id$
- $vzsalv$  -  $id$  do vizinho de salvaguarda, inicialmente  $vzsalv := -$
- $intr$  - conjunto com os ids dos vizinhos não externos, inicialmente  $intr := \emptyset$

- Cada adjacência da rede NDN é concretizada numa única sessão TCP, sendo os dois nós que partilham a adjacência *vizinhos* um do outro.
- Cada nó tem apenas um vizinho externo, podendo ter múltiplos vizinhos internos, ou nenhum.
- Em redes com mais do que um nó, há exatamente dois nós que são vizinhos externos um do outro. Esses nós têm-se a si próprios como nós de salvaguarda.

# Topologia: entrada de um nó sem recurso ao servidor de nós

1/2

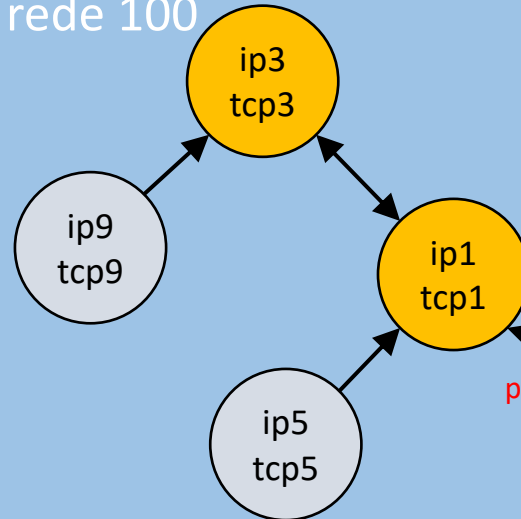
Variáveis:

- *id* - identificador do nó
- *vzext* - *id* do vizinho externo, inicialmente  $vzext := id$
- *vzsalv* - *id* do vizinho de salvaguarda, inicialmente  $vzsalv := -$
- *intr* - conjunto com os ids dos vizinhos não externos, inicialmente  $intr := \emptyset$



no nó entrante o utilizador invoca o comando  
`djoin` ~~ip1 tcp1~~

rede 100



Máquina de estados (nó ip8 tcp8)

- Comando `djoin` ~~ipY tcpY~~

Liga-se a ipY tcpY

$vzext := ipY tcpY$  (1)

Envia ENTRY  $id(ip) id(tcp)$  a  $vzext$  (2)

(2) ENTRY  $ip08 tcp08 <LF>$

por TCP

o nó entrante  
liga-se e envia  
mensagem de  
entrada

ip8:tcp8

$id = ip8 tcp8$

(1)  $vzext = ip1 tcp1$

$vzsalv = -$

$intr = \emptyset$

rede 100

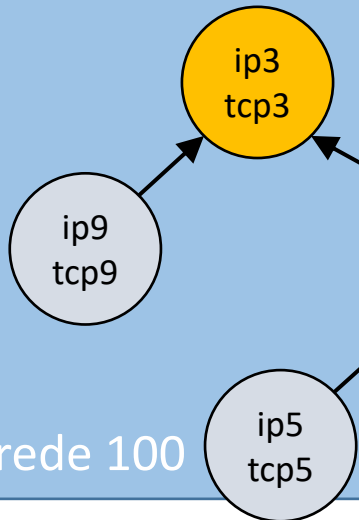
Máquina de estados (nó ip1 tcp1)

- Receção de ENTRY  $ipX tcpX$  :

$intr := intr \cup \{ipX tcpX\}$

Envia SAFE  $vzext(ip) vzext(tcp)$  a  $ipX tcpX$

o nó ip1 tcp1 atualiza a lista de vizinhos não  
externos e responde com o seu nó externo



ip1 tcp1

$id = ip1 tcp1$

$vzext = ip3 tcp3$

$vzsalv = ip1 tcp1$

$intr = [ip3 tcp3, ip5 tcp5, ip8 tcp8]$

SAFE  $ip3 tcp3 <LF>$

por TCP

ip8 tcp8

Máquina de estados (nó ip8 tcp8)

- Receção de SAFE  $ipX tcpX$  :

$vzsalv := ipX tcpX$

o nó entrante  
atualiza o nó  
de salvaguarda

ip8:tcp8

$id = ip8 tcp8$

$vzext = ip1 tcp1$

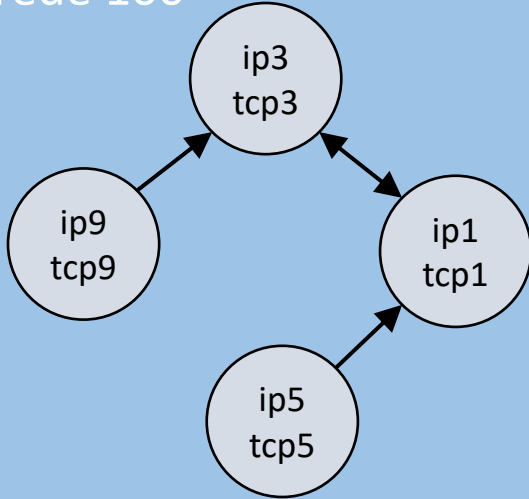
$vzsalv = ip3 tcp3$

$intr = \emptyset$

# Topologia: entrada de um nó

2/2

rede 100



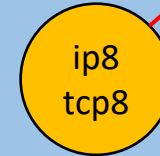
nó entrante

no nó entrante o utilizador  
invoca o comando  
`join 100`

Servidor de nós

por UDP

NODES 100

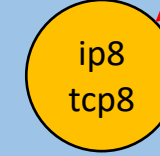


o nó entrante pede ao  
servidor de nós  
informação sobre os nós  
existentes na rede 100

Servidor de nós

por UDP

NODESLIST 100  
ip3 tcp3  
ip1 tcp1  
ip9 tcp9  
ip5 tcp5



o servidor de nós envia  
uma lista com uma linha  
por nó na rede 100



O nó entrante escolhe  
um desses nós

Nós da rede 100:  
ip3 tcp3  
**ip1 tcp1**  
ip9 tcp9  
ip5 tcp5



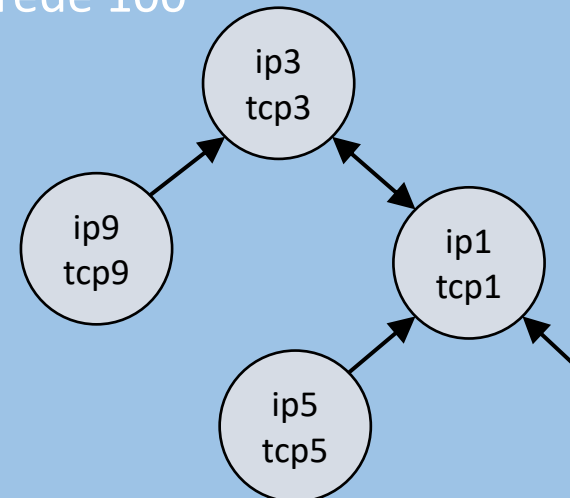
e procede como se o utilizador  
tivesse invocado o comando

`djoin 100 ip1 tcp1`

...

(ver slide anterior)

rede 100



Servidor de nós

OKREG

por UDP

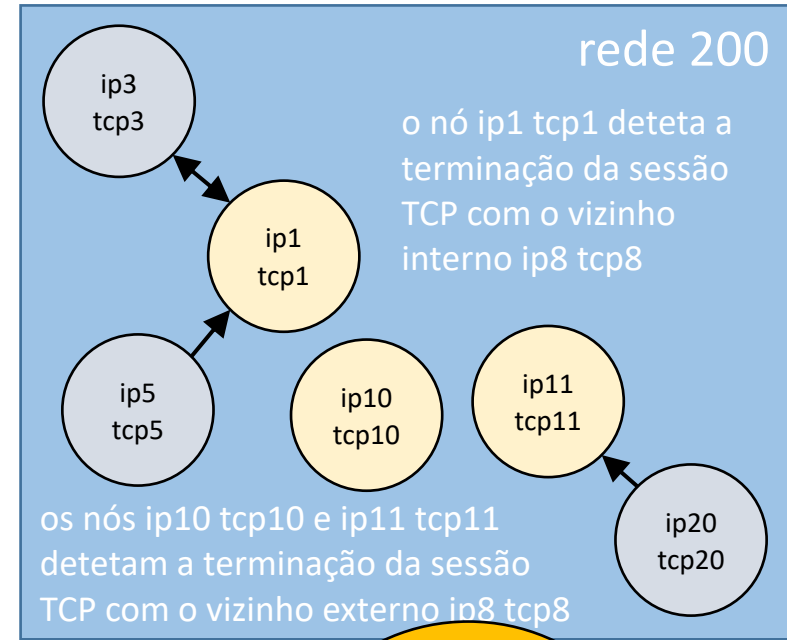
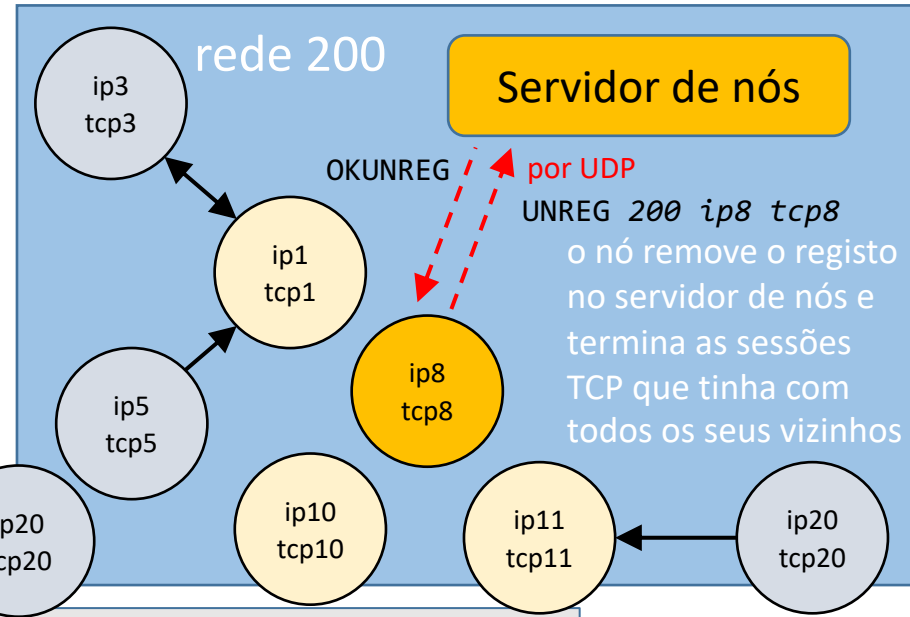
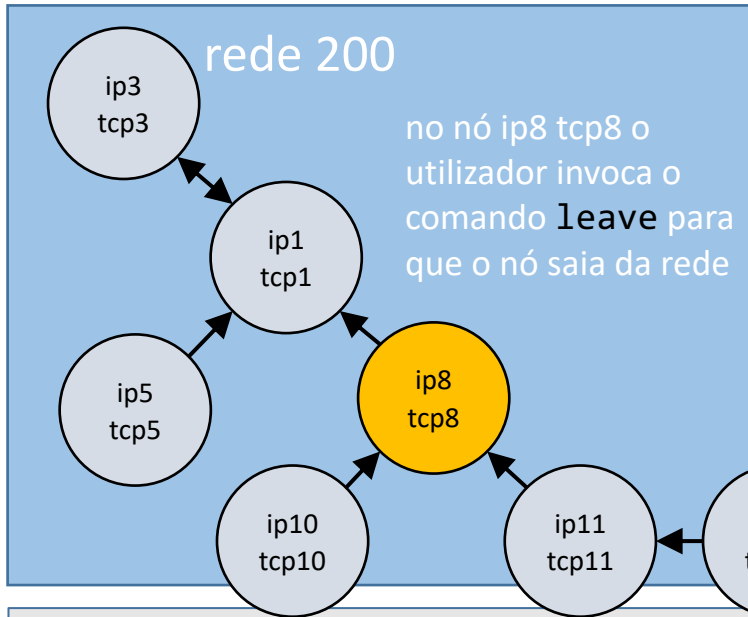
REG 100 ip8 tcp8

**Ip8 tcp8**

id = ip8 tcp8  
vzext = ip1 tcp1  
vzsalv = ip3 tcp3  
intr = ∅

o nó entrante  
registra-se no  
servidor de nós

# Topologia: saída de um nó que não é salvaguarda de si próprio 1/4



- Terminação da sessão com o vizinho x

**Se  $x \in \text{intr}$  então**

$\text{intr} := \text{intr} - \{x\}$

**Se  $\text{vzext} = x$  e  $\text{vzsalv} \neq id$  então**

Liga-se a  $\text{vzsalv}$

$\text{vzext} := \text{vzsalv}$

Envia ENTRY  $id(ip)$   $id(tcp)$  a  $\text{vzext}$

Para todo  $y \in \text{intr}$  envia SAFE  $\text{vzext}(ip)$   $\text{vzext}(tcp)$  a  $y$

**Se  $\text{vzext} = x$  e  $\text{vzsalv} = id$  e  $\text{intr} \neq \emptyset$  então**

Escolhe  $y \in \text{intr}$

$\text{vzext} := y$

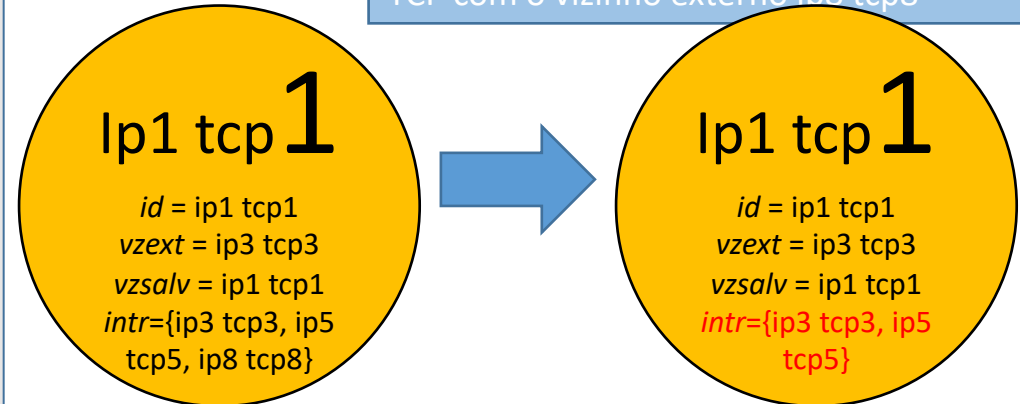
envia ENTRY  $id(ip)$   $id(tcp)$  a  $\text{vzext}$

Para todo  $z \in \text{intr}$  envia SAFE  $\text{vzext}(ip)$   $\text{vzext}(tcp)$  a  $z$

**Se  $\text{vzext} = x$  e  $\text{vzsalv} = id$  e  $\text{intr} = \emptyset$  então**

$\text{vzext} := id$

Máquina de estados  
(nó ip1 tcp1)

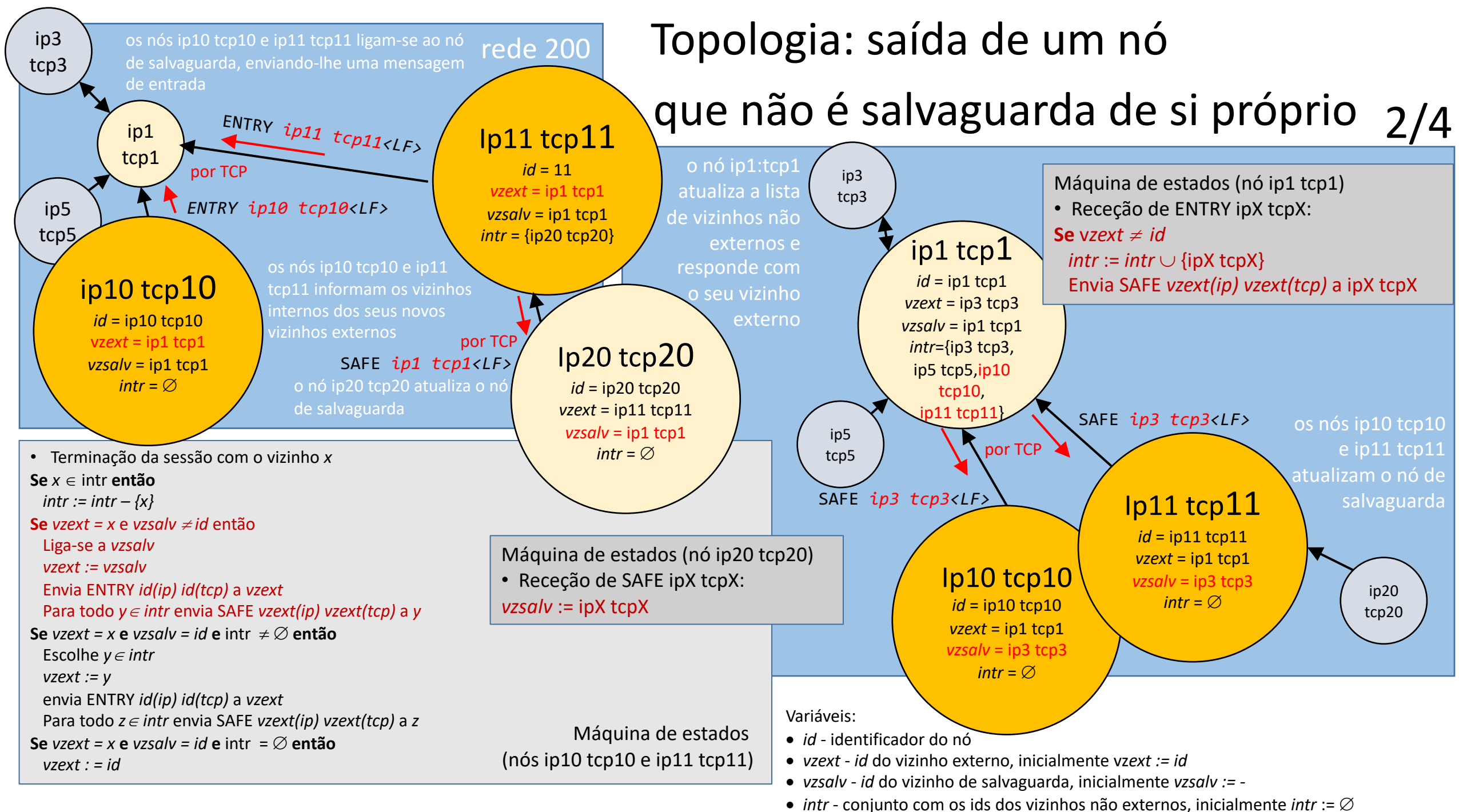


Variáveis:

- $id$  - identificador do nó
- $\text{vzext}$  -  $id$  do vizinho externo, inicialmente  $\text{vzext} := id$
- $\text{vzsalv}$  -  $id$  do vizinho de salvaguarda, inicialmente  $\text{vzsalv} := -$
- $\text{intr}$  - conjunto com os ids dos vizinhos não externos, inicialmente  $\text{intr} := \emptyset$



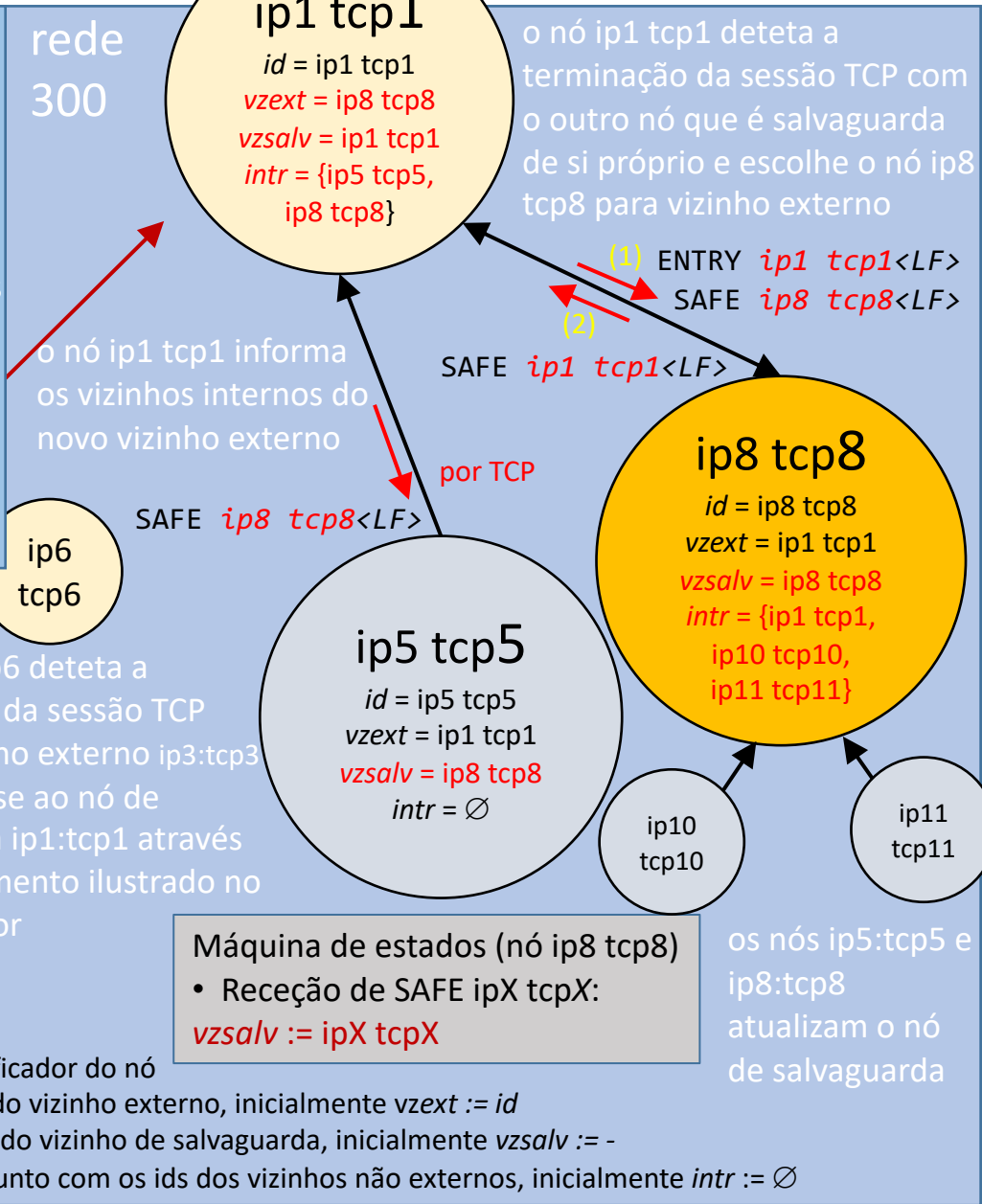
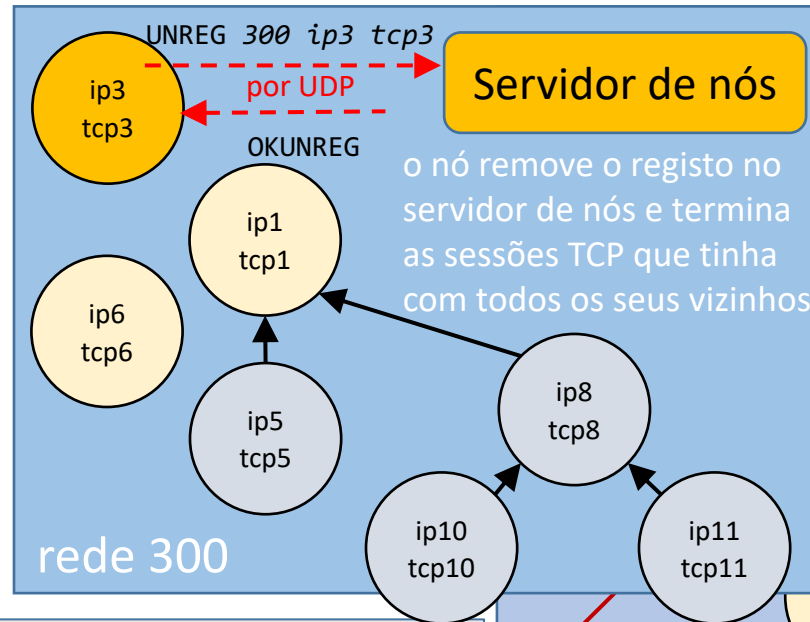
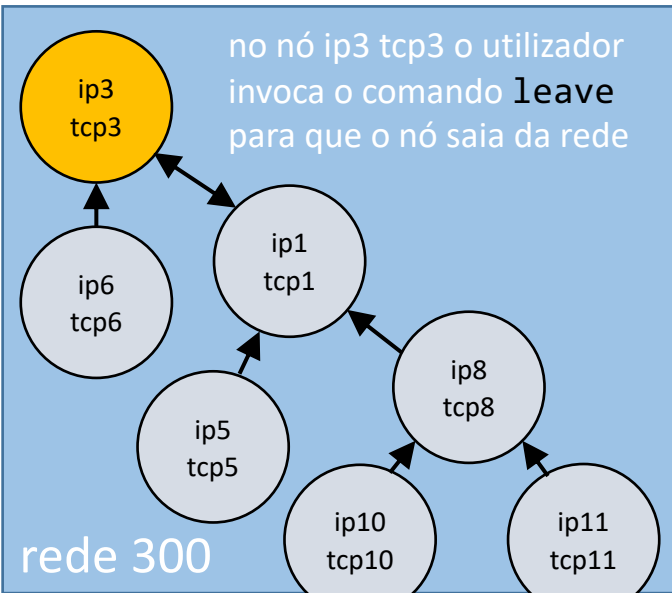
# Topologia: saída de um nó que não é salvaguarda de si próprio 2/4





# Topologia: saída de um nó que é salvaguarda de si próprio

3/4



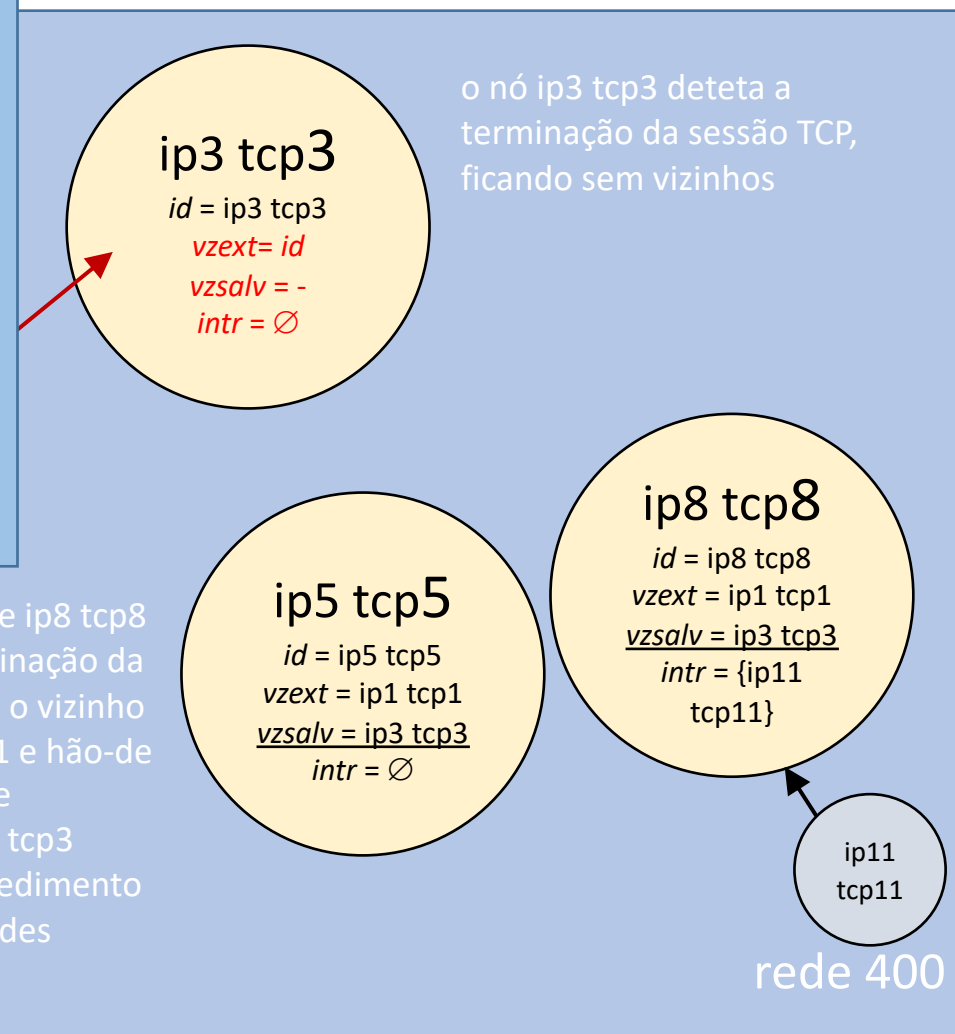
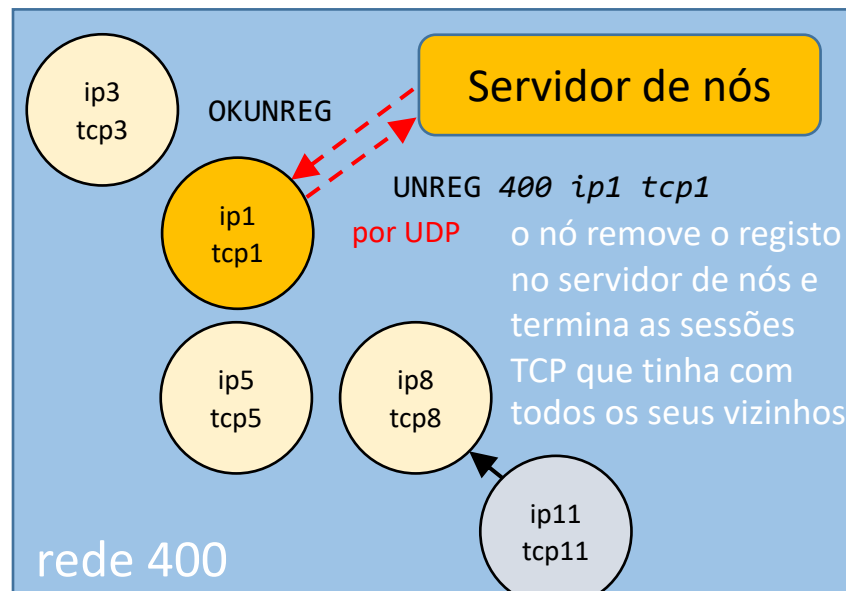
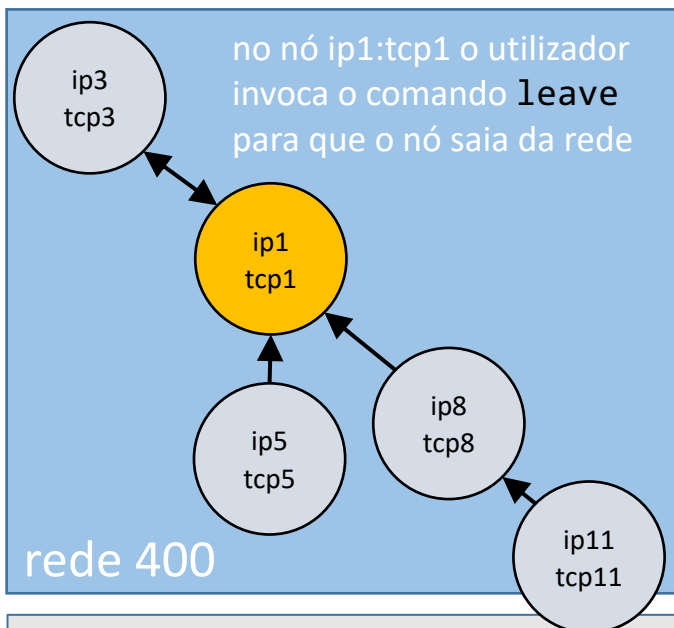
- Terminação da sessão com o vizinho x
- Se**  $x \in \text{intr}$  **então**  
 $\text{intr} := \text{intr} - \{x\}$
- Caso contrário se**  $\text{vzsalv} \neq \text{id}$  **então**  
 Liga-se a vzsalv  
 $\text{vzext} := \text{vzsalv}$   
 Envia ENTRY id(ip) id(tcp) a vzext  
 Para todo  $y \in \text{intr}$  envia SAFE vzext(ip) vzext(tcp) a y
- Caso contrário se**  $\text{intr} \neq \emptyset$  **então**  
 Escolhe  $y \in \text{intr}$   
 $\text{vzext} := y$   
 envia ENTRY id(ip) id(tcp) a vzext  
 Para todo  $z \in \text{intr}$  envia SAFE vzext(ip) vzext(tcp) a z
- Caso contrário**  
 $\text{vzext} := \text{id}$

Máquina de estados  
(nó ip1 tcp1)

# Topologia: saída de um nó que é salvaguarda de si próprio

4/4

novο cenário



- Terminação da sessão com o vizinho  $x$   
**Se**  $x \in intr$  **então**  
 $intr := intr - \{x\}$   
**Caso contrário se**  $vzsalv \neq id$  **então**  
Liga-se a  $vzsalv$   
 $vzext := vzsalv$   
Envia ENTRY  $id(ip)$   $id(tcp)$  a  $vzext$   
Para todo  $y \in intr$  envia SAFE  $vzext(ip)$   $vzext(tcp)$  a  $y$   
**Caso contrário se**  $intr \neq \emptyset$  **então**  
Escolhe  $y \in intr$   
 $vzext := y$   
envia ENTRY  $id(ip)$   $id(tcp)$  a  $vzext$   
Para todo  $z \in intr$  envia SAFE  $vzext(ip)$   $vzext(tcp)$  a  $z$   
**Caso contrário**  
 $vzext := id$

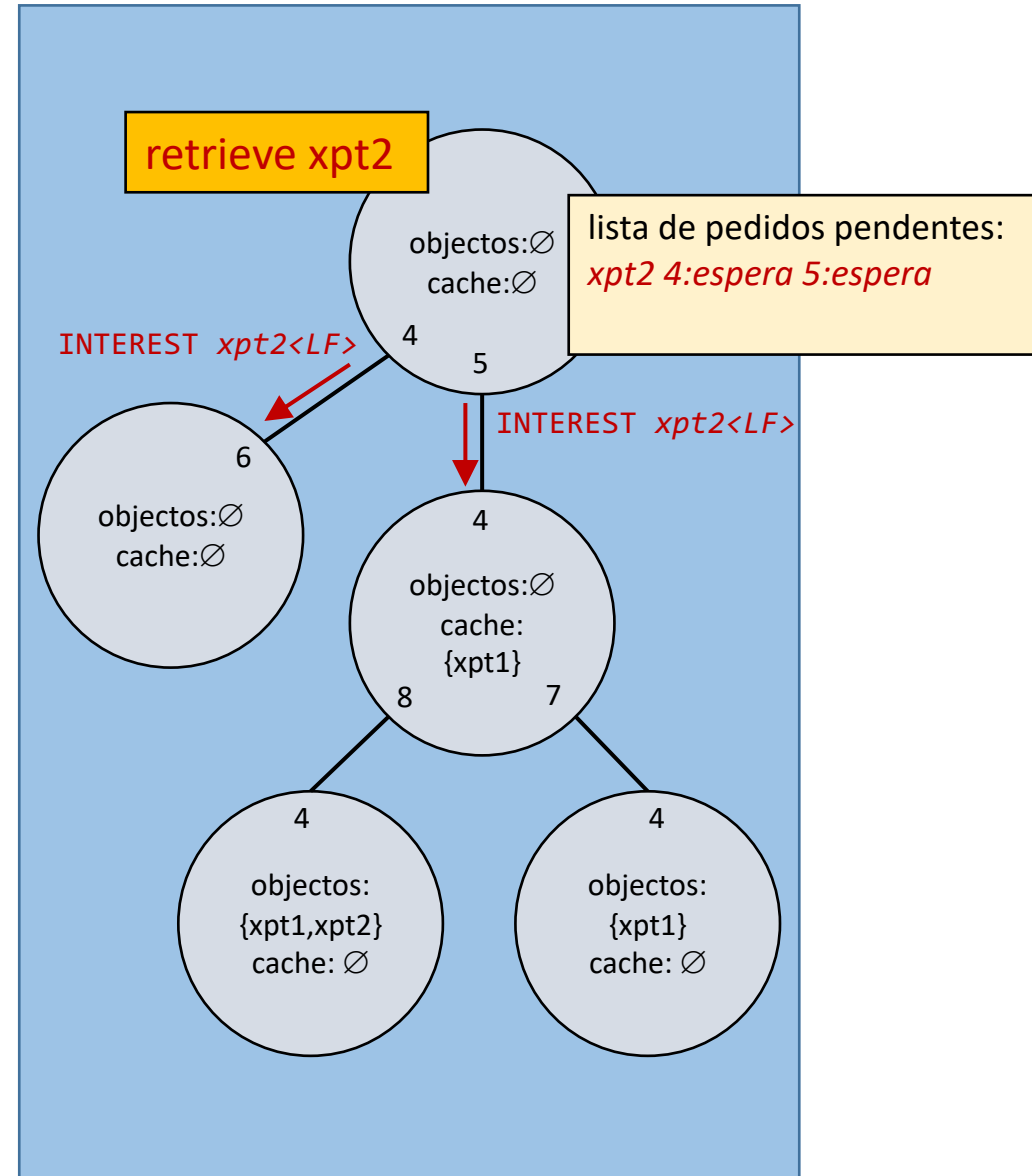
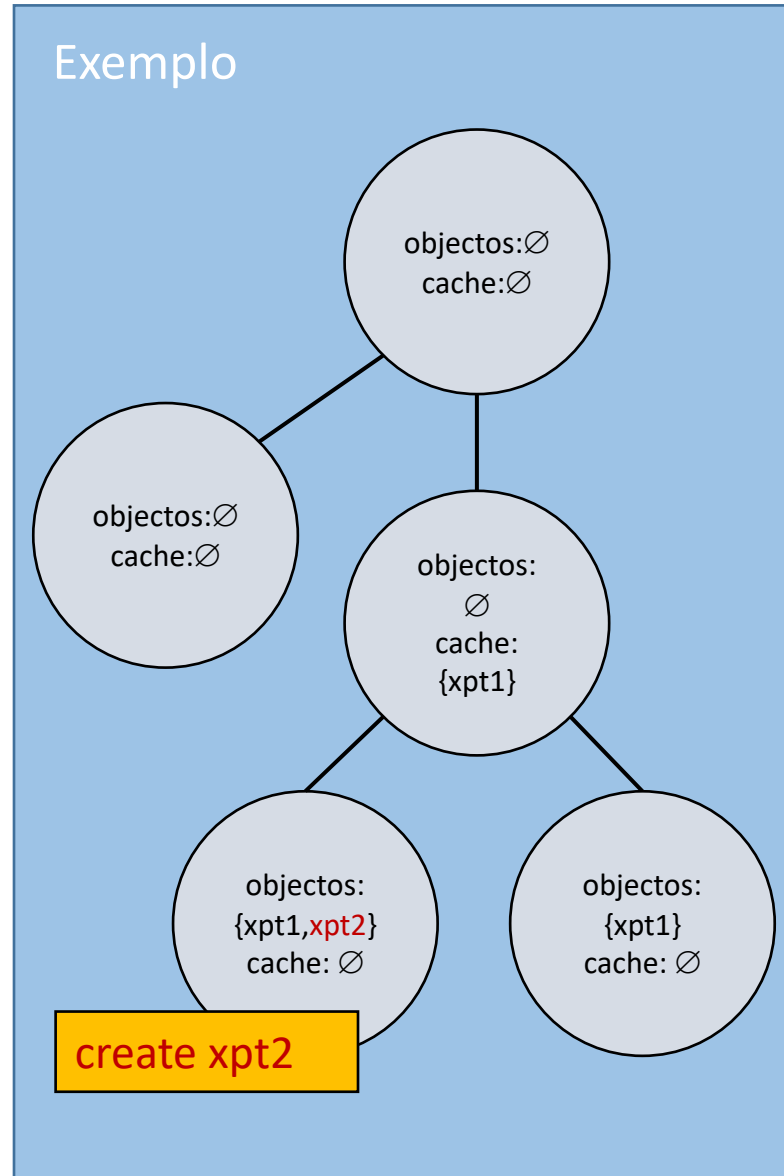
Máquina de estados  
(nó ip3:tcp3)

os nós ip5 tcp5 e ip8 tcp8 detetam a terminação da sessão TCP com o vizinho externo ip1 tcp1 e hão-de ligar-se ao nó de salvaguarda ip3 tcp3 através do procedimento ilustrado nos slides anteriores

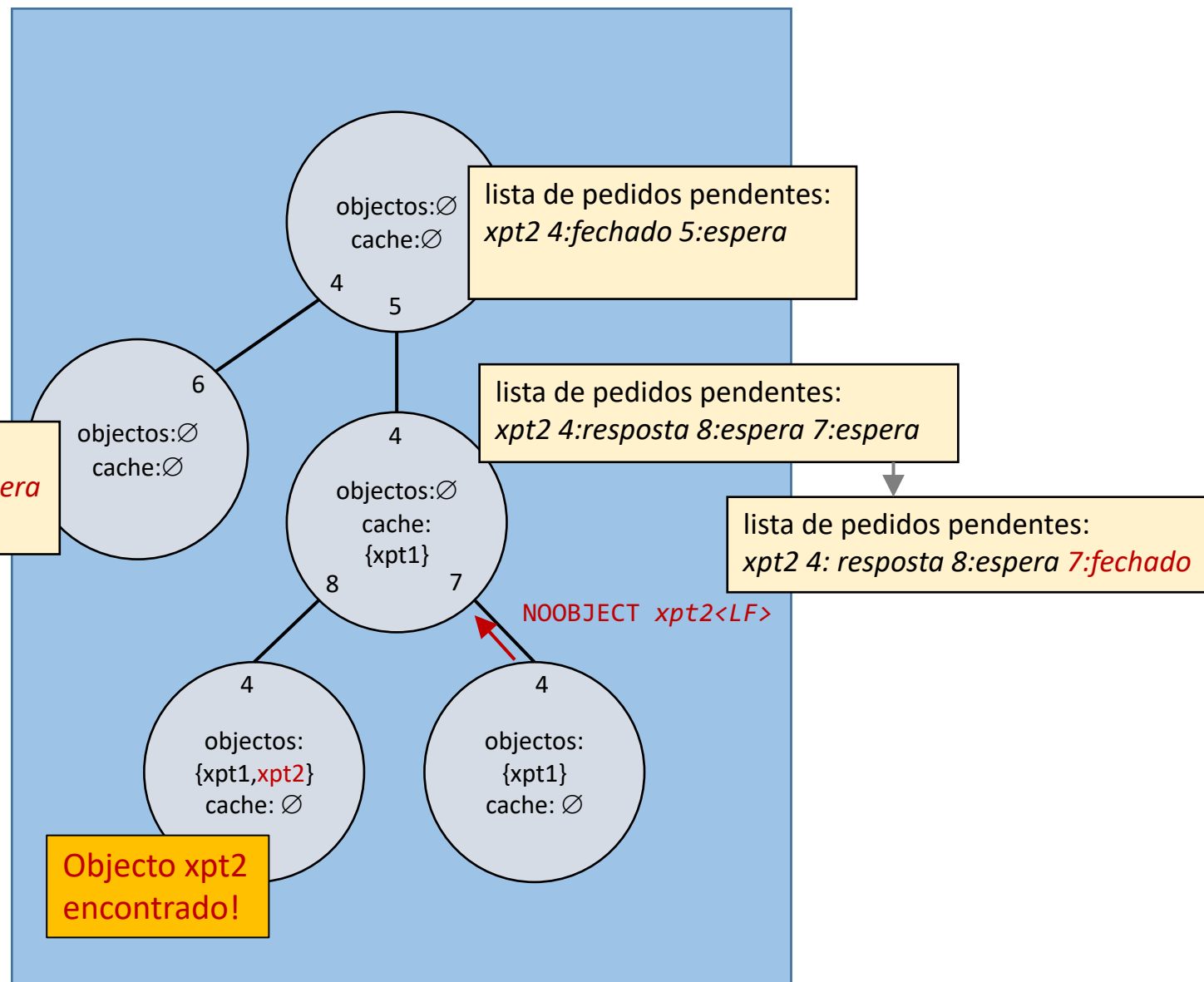
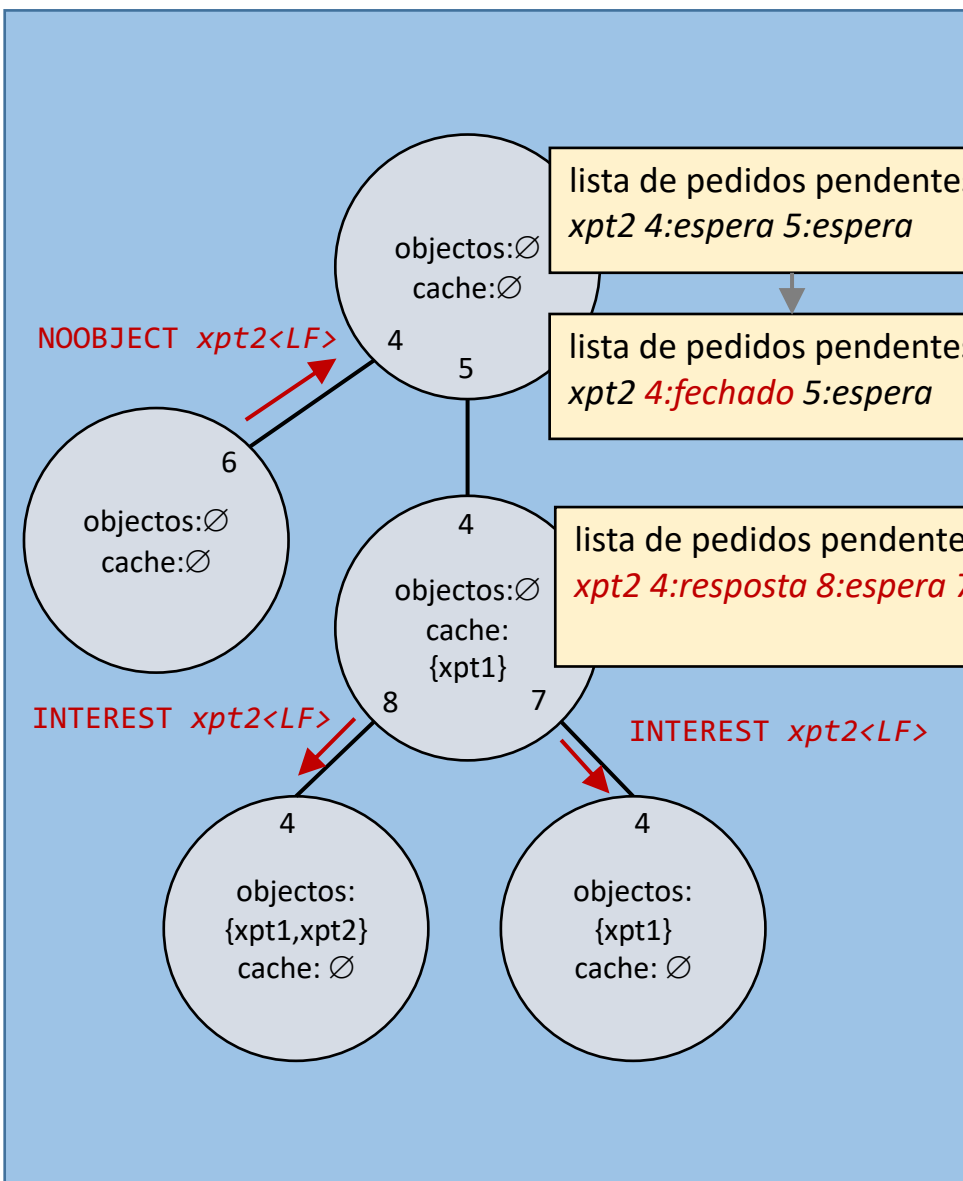
Variáveis:

- $id$  - identificador do nó
- $vzext$  -  $id$  do vizinho externo, inicialmente  $vzext := id$
- $vzsalv$  -  $id$  do vizinho de salvaguarda, inicialmente  $vzsalv := -$
- $intr$  - conjunto com os ids dos vizinhos não externos, inicialmente  $intr := \emptyset$

# Obtenção de objetos na rede NDN (1/3)



# Obtenção de objetos na rede NDN (2/3)



# Obtenção de objetos na rede NDN (3/3)

