

RCI 2022/2023

Transferência de Conteúdos em Redes Arbóreas

Rede Arbórea

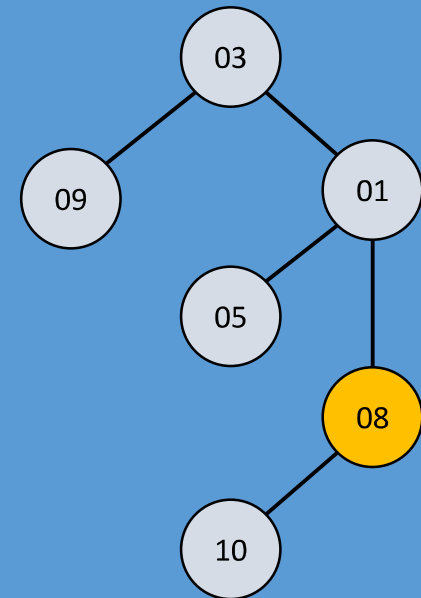
- Cada rede tem um identificador único (ex: 001, 002...).
- Cada rede é composta por nós, com identificadores únicos a cada rede (ex: 01, 02...), interligados em árvore.
- Cada nó contém um conjunto de conteúdos com nomes únicos a cada nó (ex: exame_RCI.txt, bibliografia_RCI.pdf...).
- Cada nó dispõe de uma interface de utilizador que permite a um utilizador descarregar de outro nó um conteúdo, o qual deverá ser encaminhado para o nó do utilizador.
- Cada nó mantém uma tabela de expedição com entradas, cada uma das quais associa um nó destino ao vizinho ao longo do único caminho na árvore até lá.
- As tabelas de expedição são povoadas com base no nó de origem das mensagens que atravessam a rede e podem estar incompletas a determinado instante.
- Uma mensagem com destino a um nó que não conste da tabela de expedição é expedida para todos os vizinhos excepto para o vizinho de onde foi recebida.

Exemplo

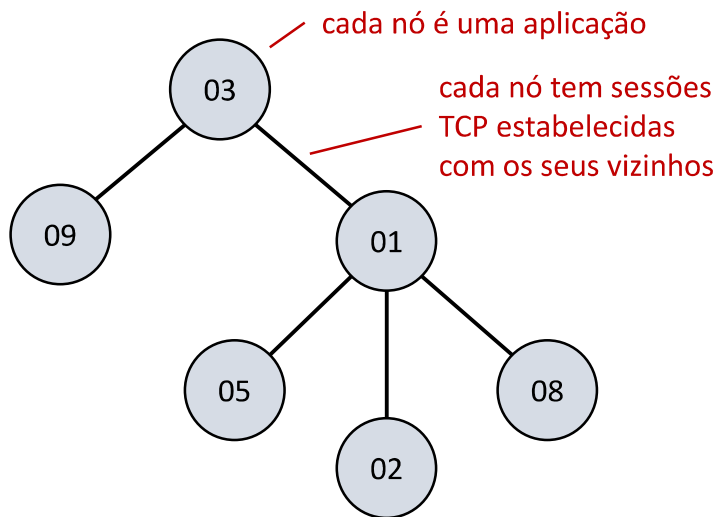
Lista de conteúdos do nó 08:

- exame_RCI.txt
- bibliografia_RCI.pdf

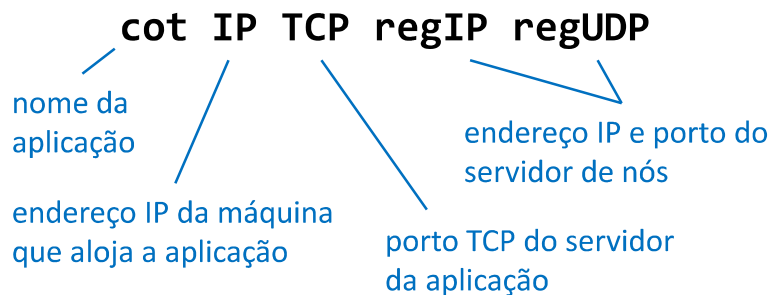
Tabela de expedição do nó 08	
Destino	Vizinho
01	01
03	01
10	10



Concretização da rede arbórea



Invocacãõ da aplicacãõ

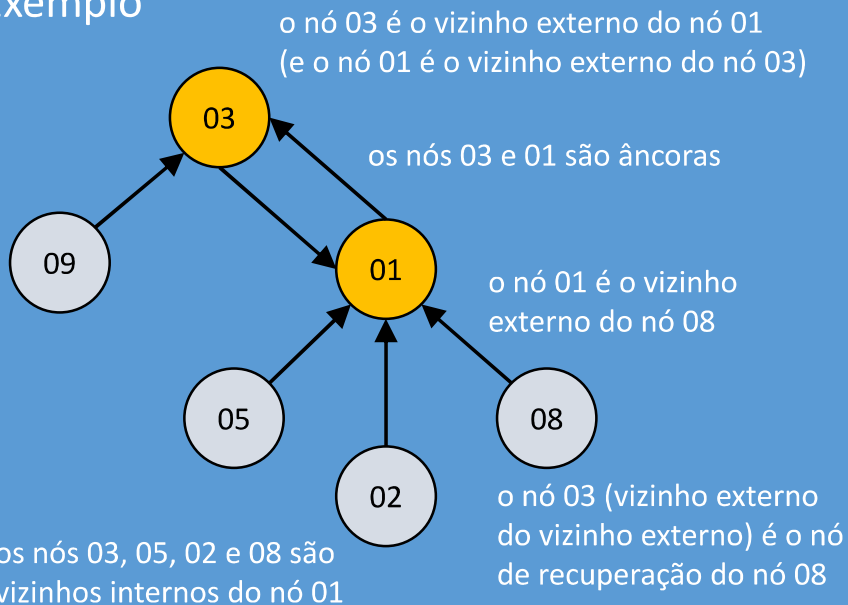


Comandos da interface de utilizador da aplicacãõ

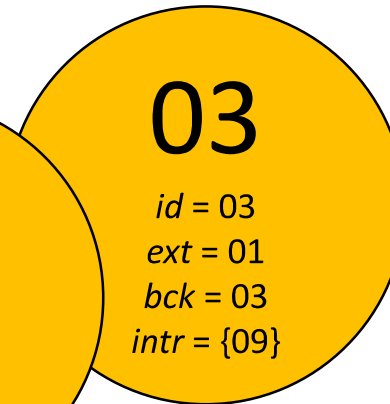
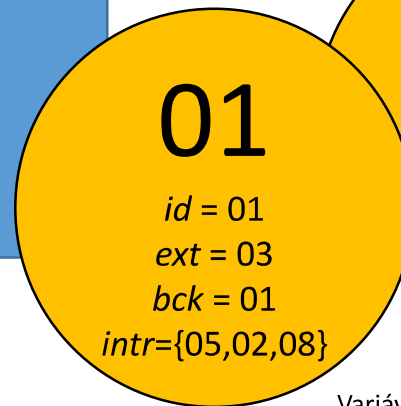
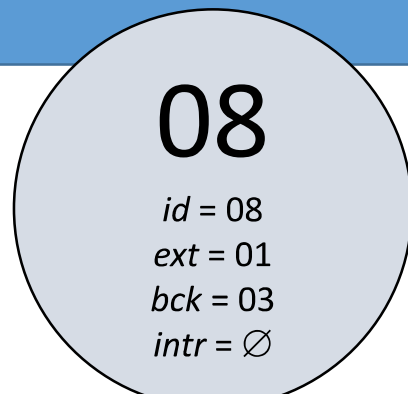
Comando	Descriçãõ
join net id	Entrada de um nó na rede net com identificador id .
djoin net id bootid bootIP bootTCP	Entrada de um nó na rede net com identificador id , que se sabe ser único na rede. É passado à aplicacãõ o identificador e o contacto de um nó da rede, através dos parâmetros bootid , bootIP e bootTCP , ao qual o nó se deverá ligar sem interrogar o servidor de nós.
create name	É criado um conteúdo de nome name .
delete name	É apagado o conteúdo de nome name .
get dest name	Pesquisa do conteúdo com o nome name localizado no nó dest .
show topology (st)	Mostra os identificadores e os contactos dos vizinhos internos, do vizinho externo e do vizinho de recuperaçãõ.
show names (sn)	Mostra os nomes dos conteúdos presentes no nó.
show routing (sr)	Mostra a tabela de expedicãõ do nó.
leave	Saída do nó da rede.
exit	Fecho da aplicacãõ.

Topologia da Rede Arbórea

Exemplo



- Cada aresta da árvore é substantiada numa sessão TCP, sendo os dois nós que partilham a aresta *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, chamados *âncora*, que são vizinhos externos um do outro.
- Os nós âncora têm-se a si próprios como nós de recuperação.



Variáveis:

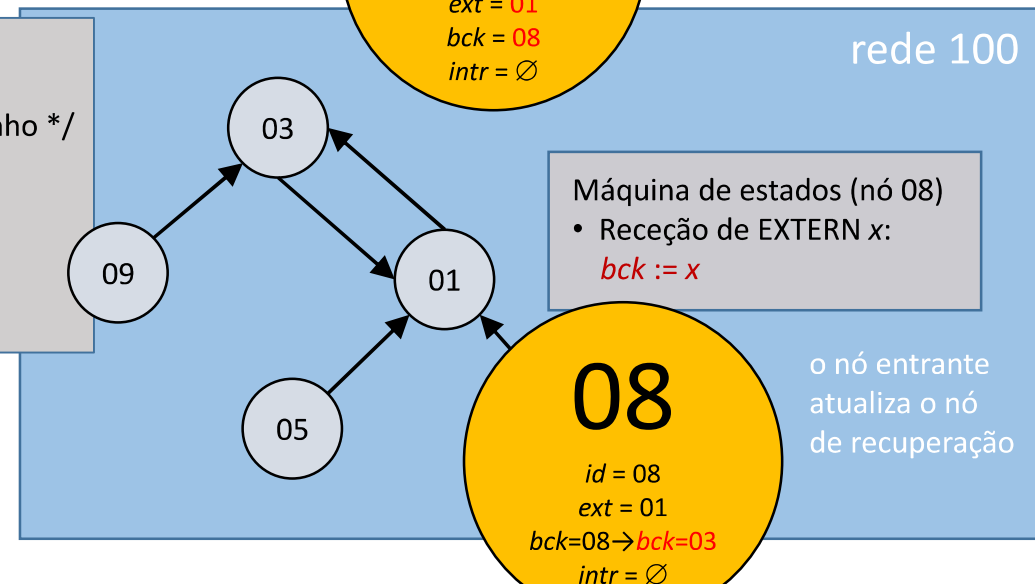
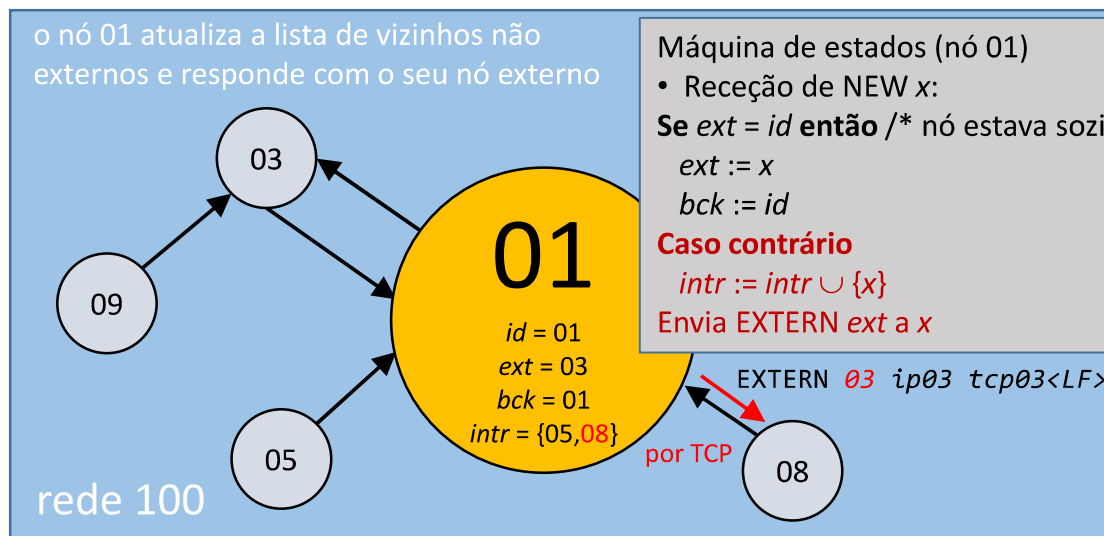
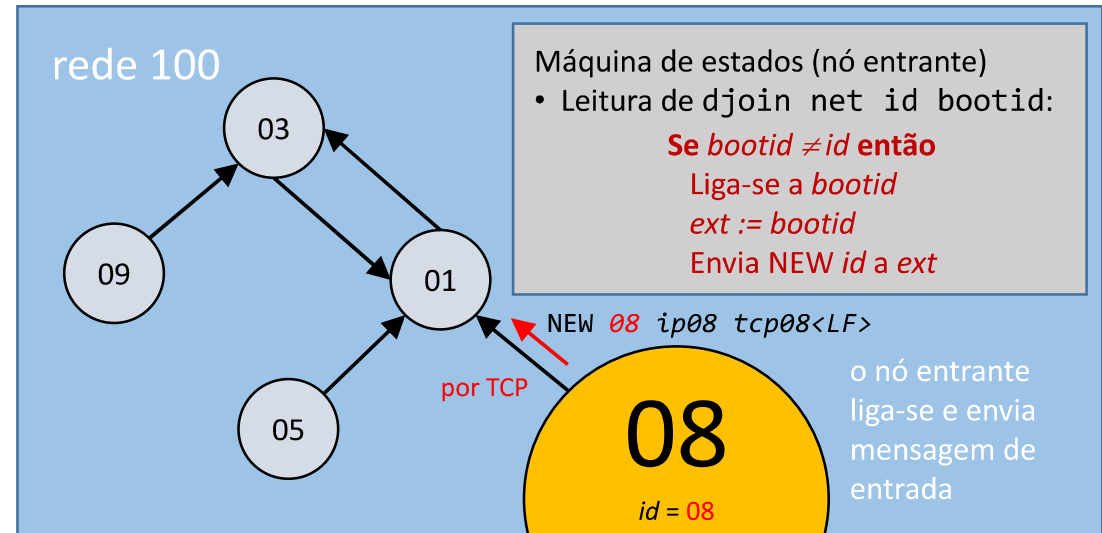
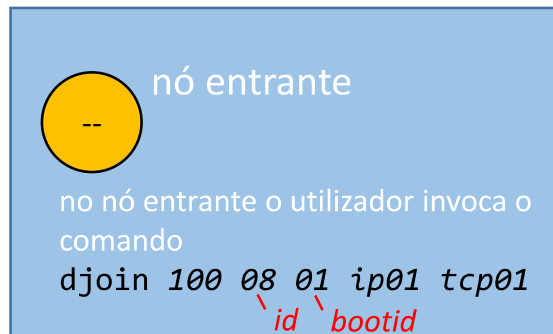
- *id* - identificador do nó
- *ext* - *id* do vizinho externo, inicialmente *ext* := *id*
- *bck* - *id* do vizinho de recuperação, inicialmente *bck* := *id*
- *intr* - conjunto com os ids dos vizinhos não externos, inicialmente *intr* := \emptyset

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

1/2

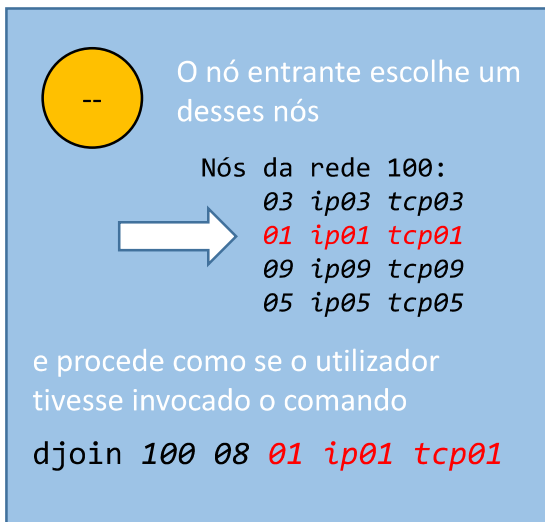
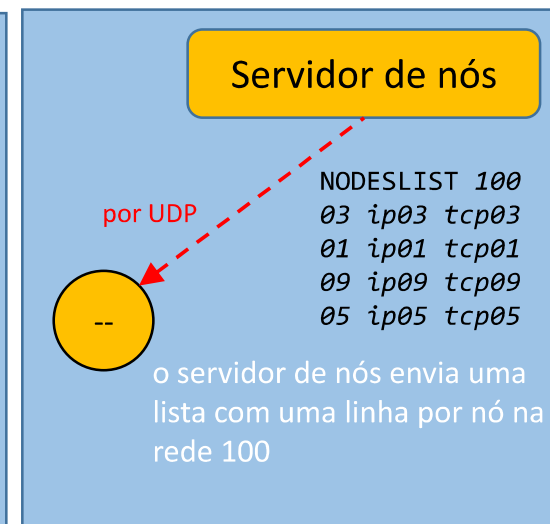
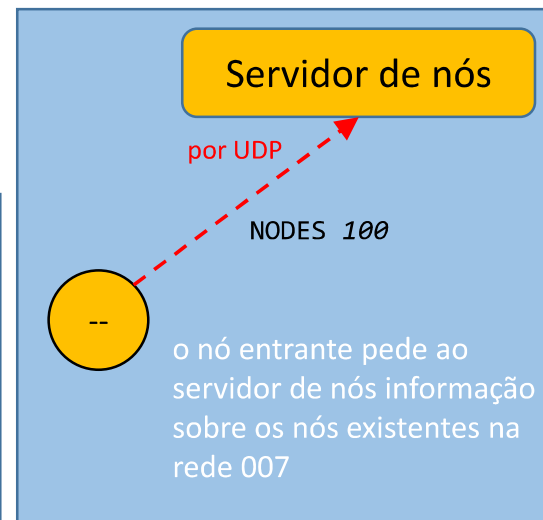
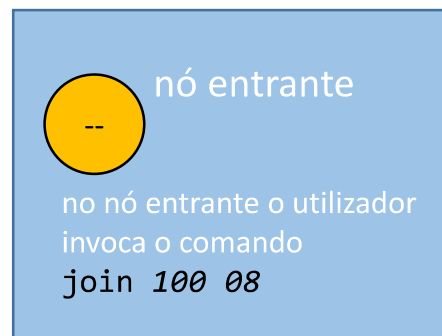
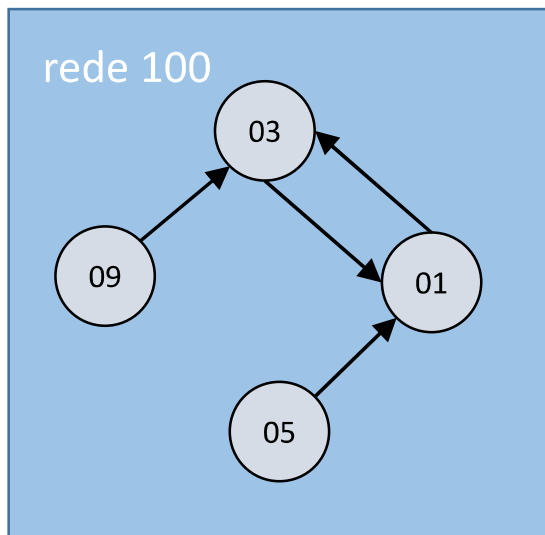
Variáveis:

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



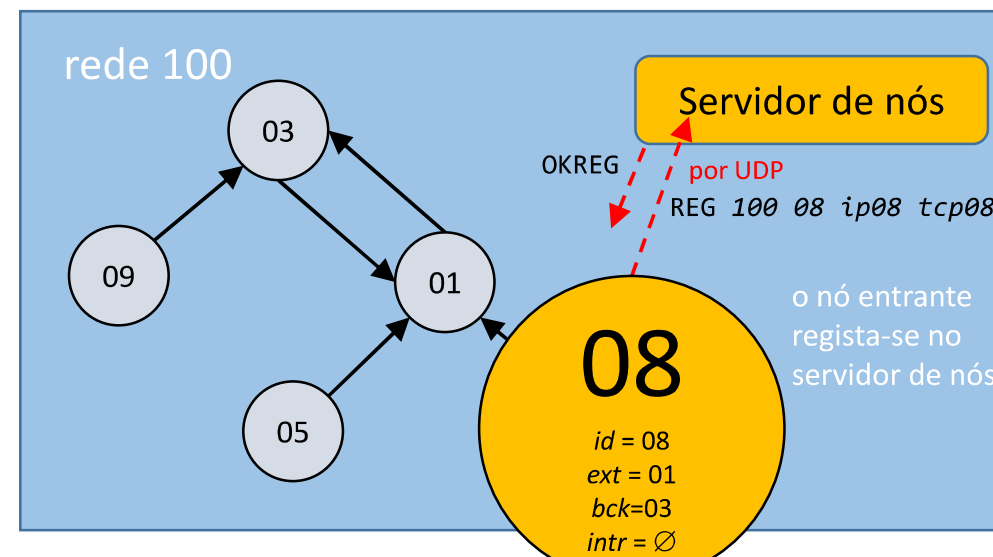
Topologia: entrada de um nó

2/2

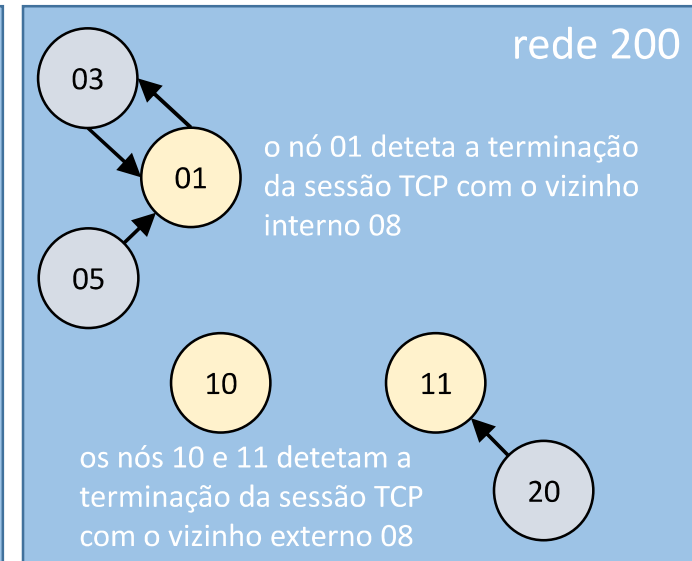
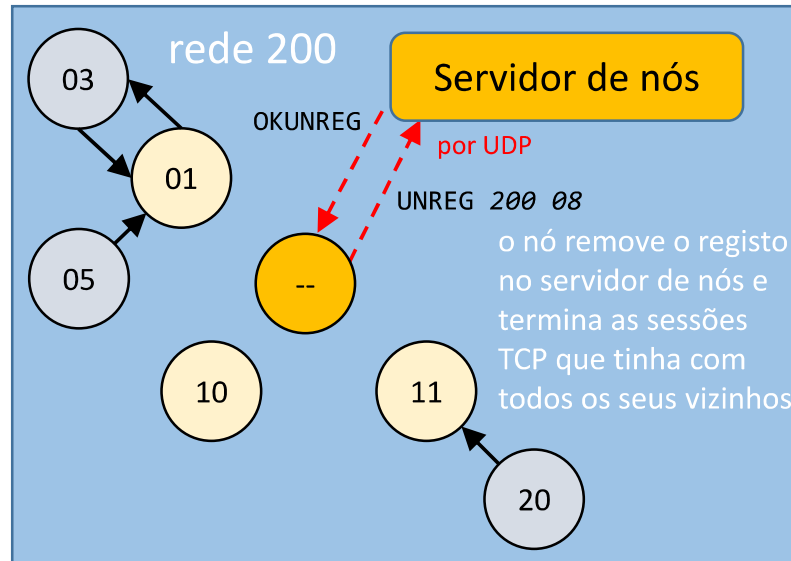
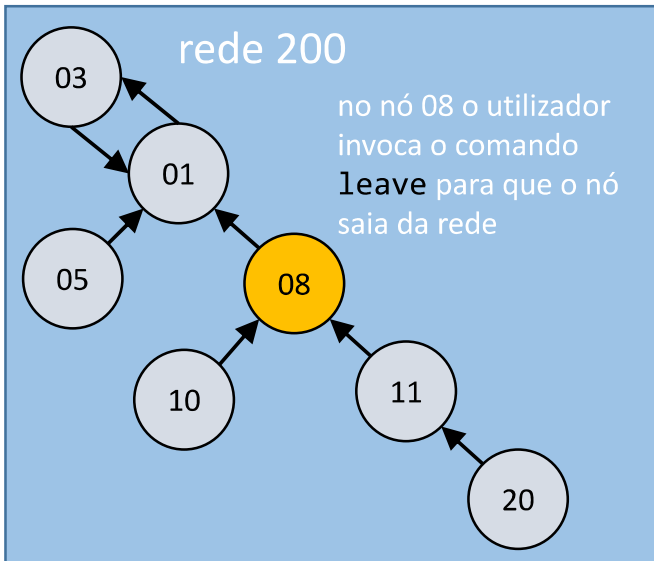


...

(ver slide anterior)



Topologia: saída de um nó que não é âncora 1/3



- Terminação da sessão com o vizinho x

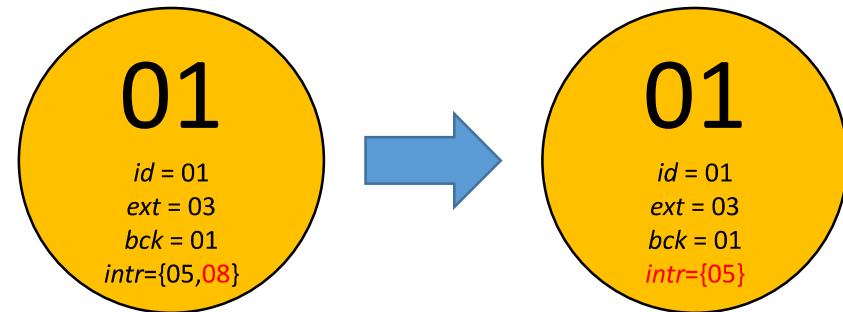
Se $x \in \text{intr}$ então /* term. com vizinho que não é externo */
 $\text{intr} := \text{intr} - \{x\}$

Caso contrário se $\text{bck} \neq \text{id}$ então /* term. com vizinho externo, não é âncora */
 Liga-se a bck
 $\text{ext} := \text{bck}$
 Envia NEW id a ext
 Para todo $x \in \text{intr}$ envia EXTERN ext a x

Caso contrário se $\text{intr} \neq \emptyset$ então /* term. com v. e., é âncora, tem vizinho interno */
 Escolhe $y \in \text{intr}$
 $\text{ext} := y$
 Para todo $x \in \text{intr}$ envia EXTERN ext a x
 $\text{intr} := \text{intr} - \{y\}$

Caso contrário /* nó ficou sozinho */
 $\text{ext} := \text{id}$

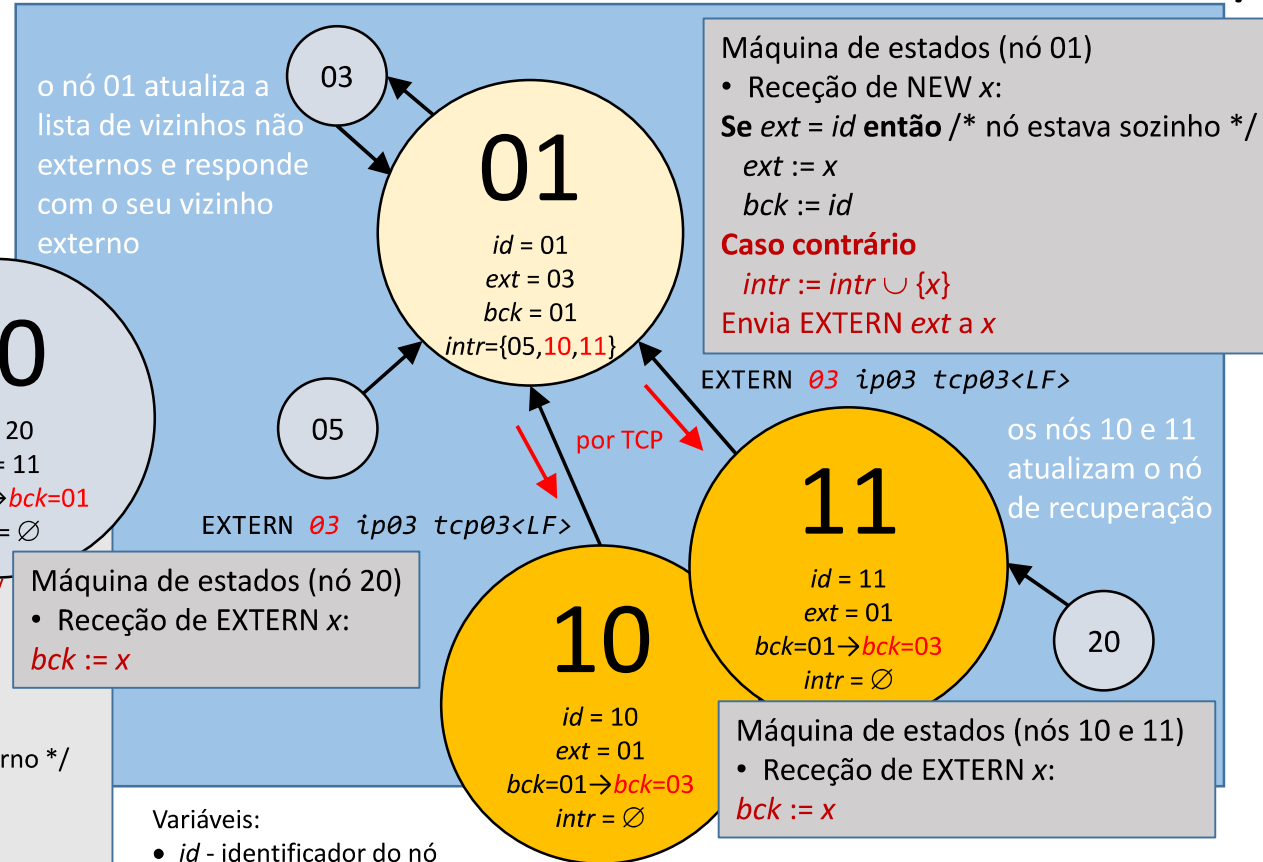
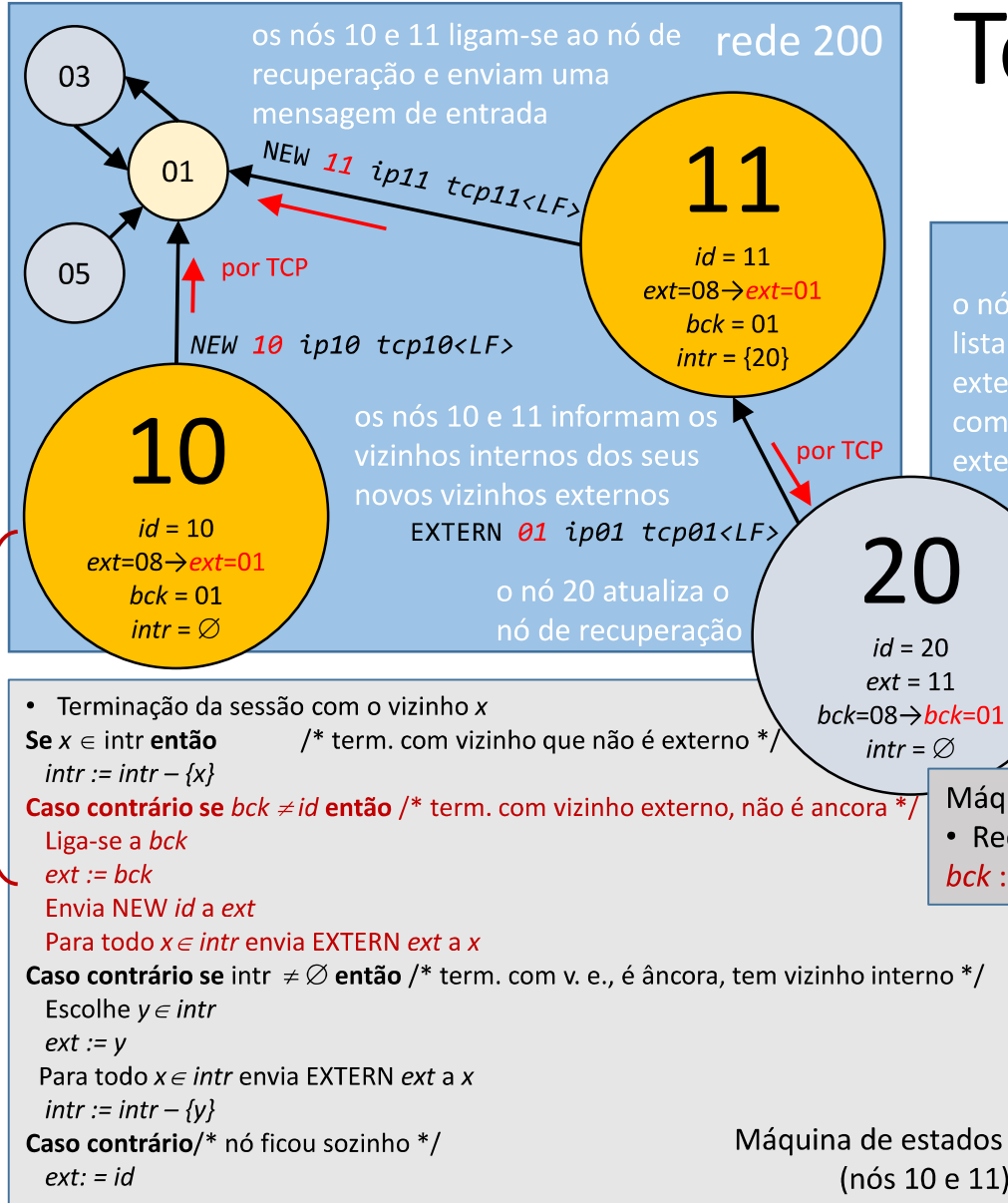
Máquina de estados (nó 01)



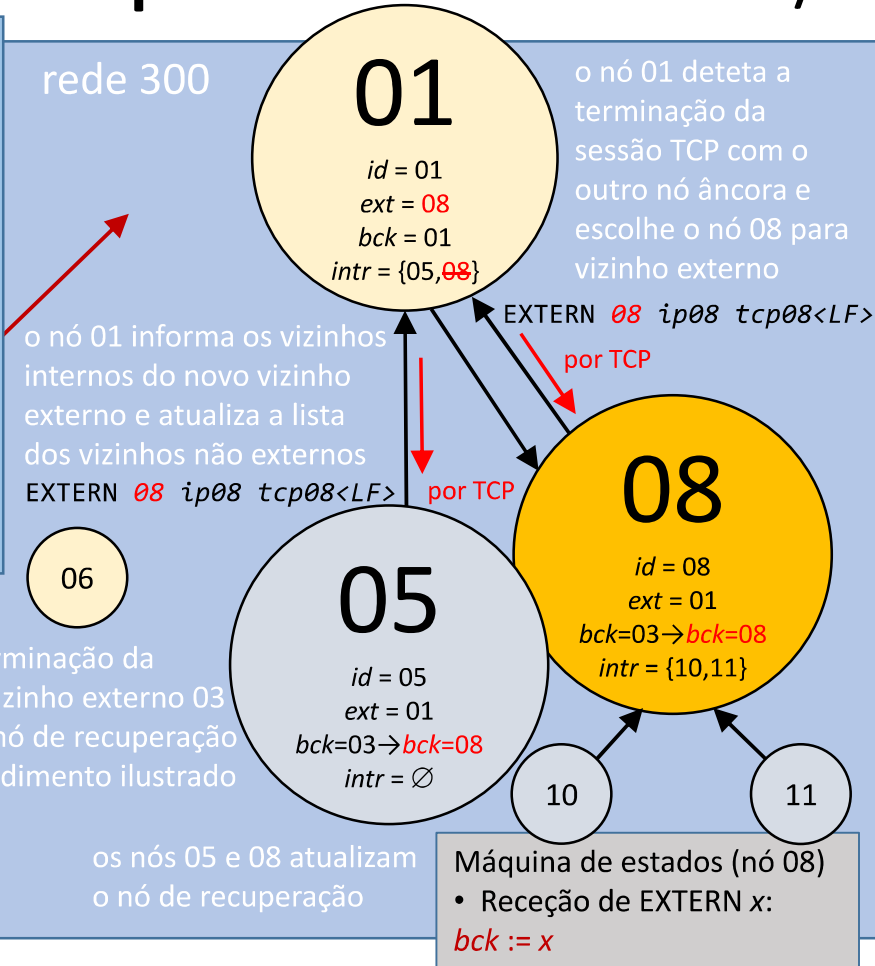
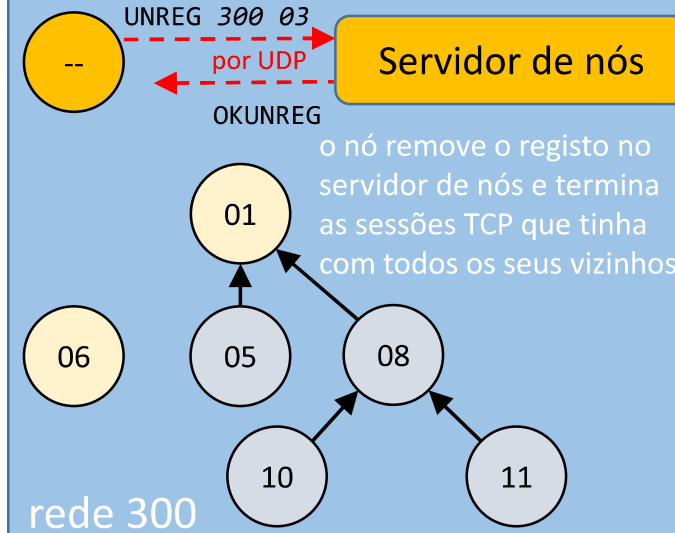
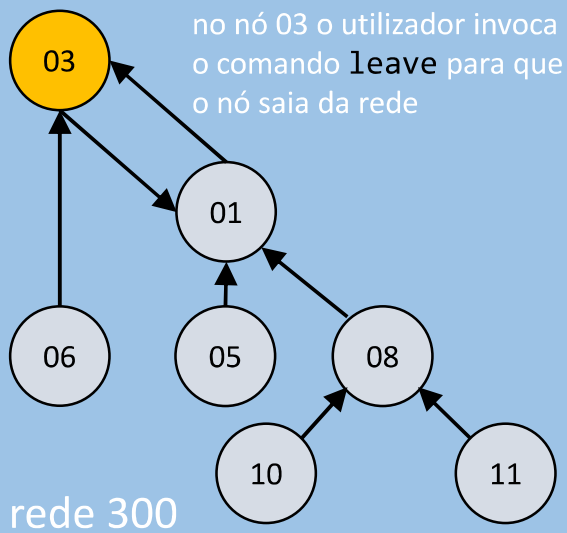
Variáveis:

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

Topologia: saída de um nó que não é âncora 2/3



Topologia: saída de um nó que é âncora 3/3



- Terminação da sessão com o vizinho x

Se $x \in \text{intr}$ **então** /* term. com vizinho que não é externo */
 $\text{intr} := \text{intr} - \{x\}$

Caso contrário se $\text{bck} \neq \text{id}$ **então** /* term. com vizinho externo, não é ancora */
 Liga-se a bck
 $\text{ext} := \text{bck}$
 Envia NEW id a ext
 Para todo $x \in \text{intr}$ envia EXTERN ext a x

Caso contrário se $\text{intr} \neq \emptyset$ **então** /* term. com v. e., é âncora, tem vizinho interno */
 Escolhe $y \in \text{intr}$
 $\text{ext} := y$
 Para todo $x \in \text{intr}$ envia EXTERN ext a x
 $\text{intr} := \text{intr} - \{y\}$

Caso contrário /* nó ficou sozinho */
 $\text{ext} := \text{id}$

Máquina de estado (nó 0)

Máquina de estados
(nó 01)

Variáveis:

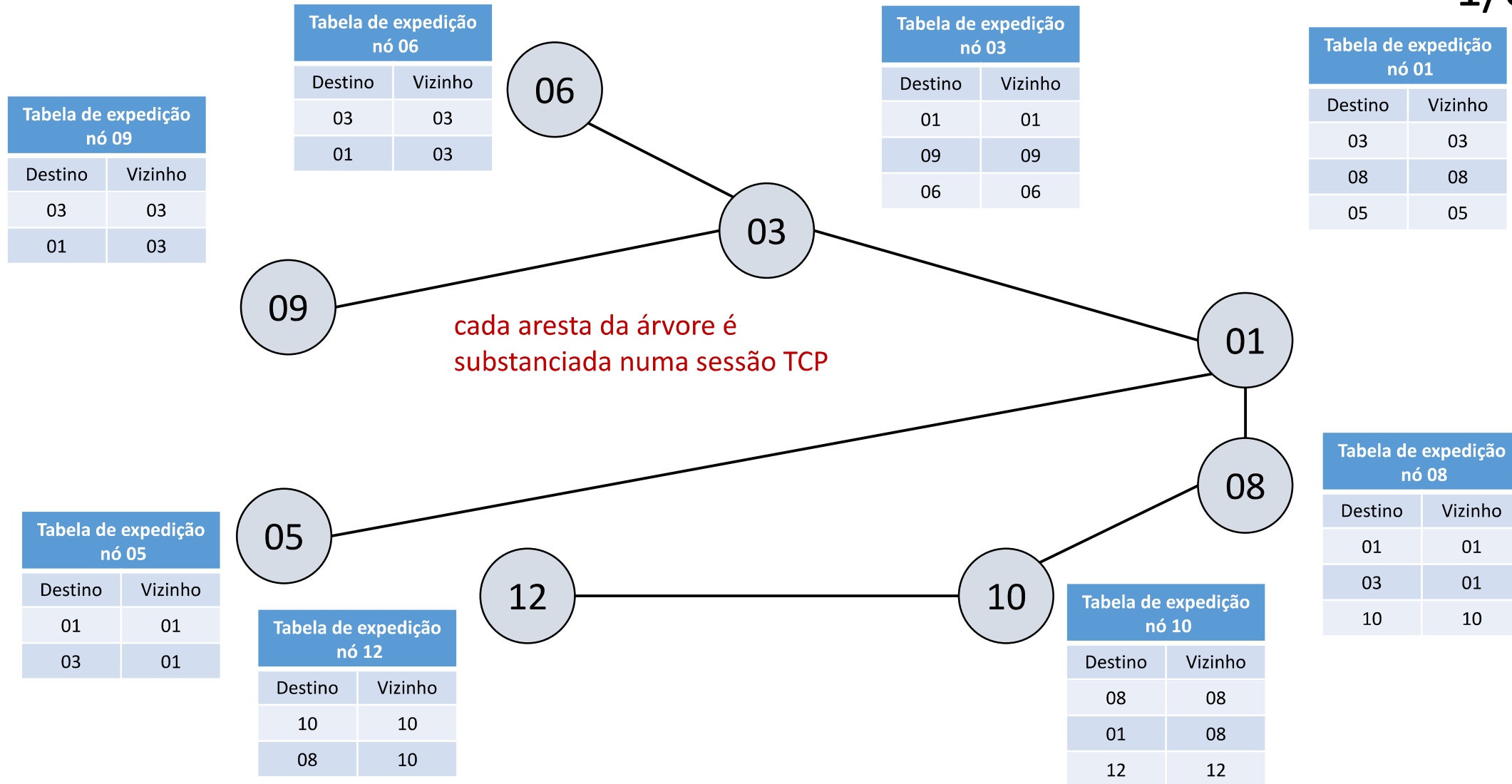
- *id* - identificador do nó
- *ext* - *id* do vizinho externo, inicialmente $ext := id$
- *bck* - *id* do vizinho de recuperação, inicialmente $bck := id$
- *intr* - conjunto com os ids dos vizinhos não externos, inicialmente $intr := \emptyset$

- $ext - id$ do vizinho externo, inicialmente $ext := id$

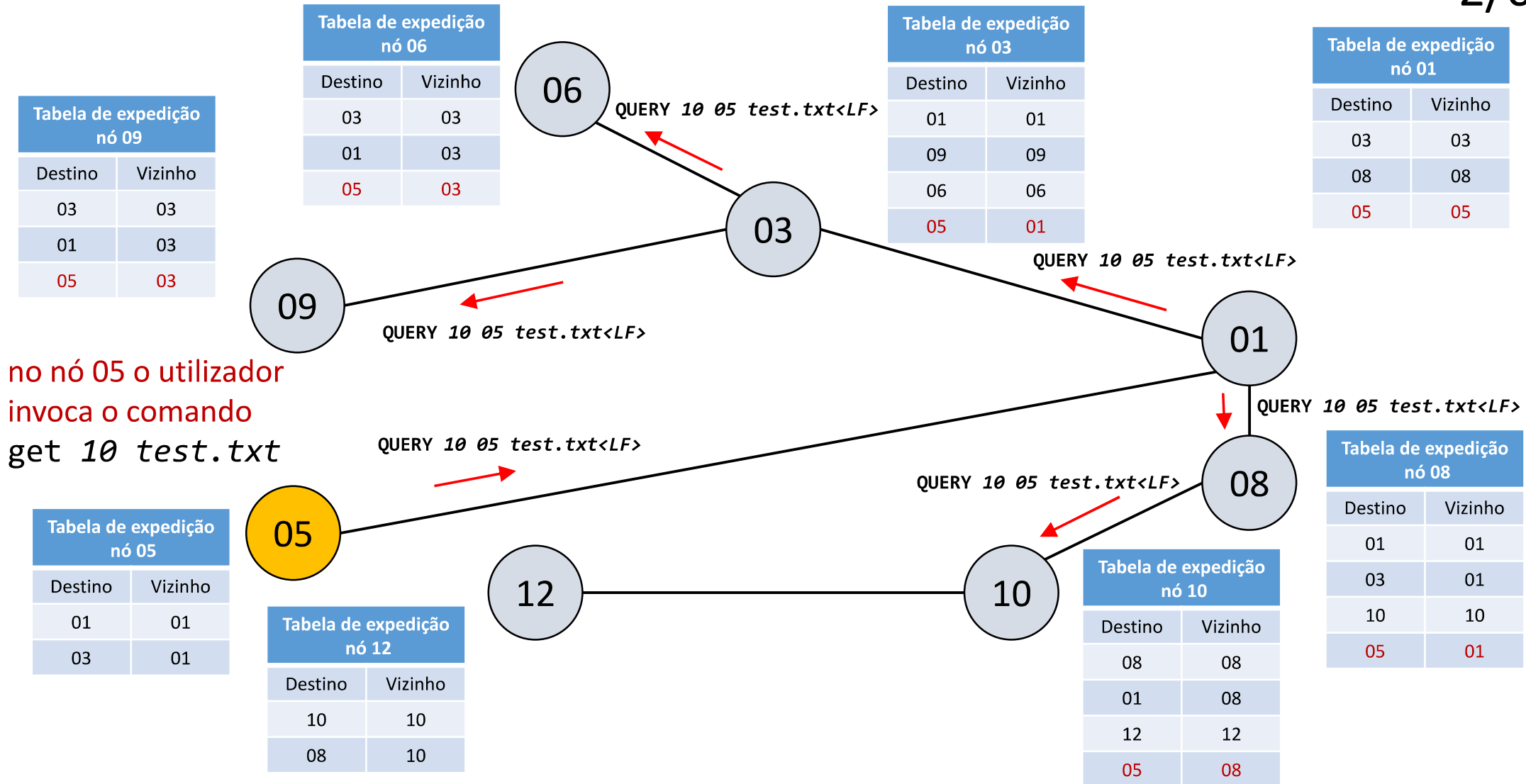
- *bck* - id do vizinho de recuperação, inicialmente $bck := id$

- *intr* - conjunto com os ids dos vizinhos não externos, inicialmente $intr := \emptyset$

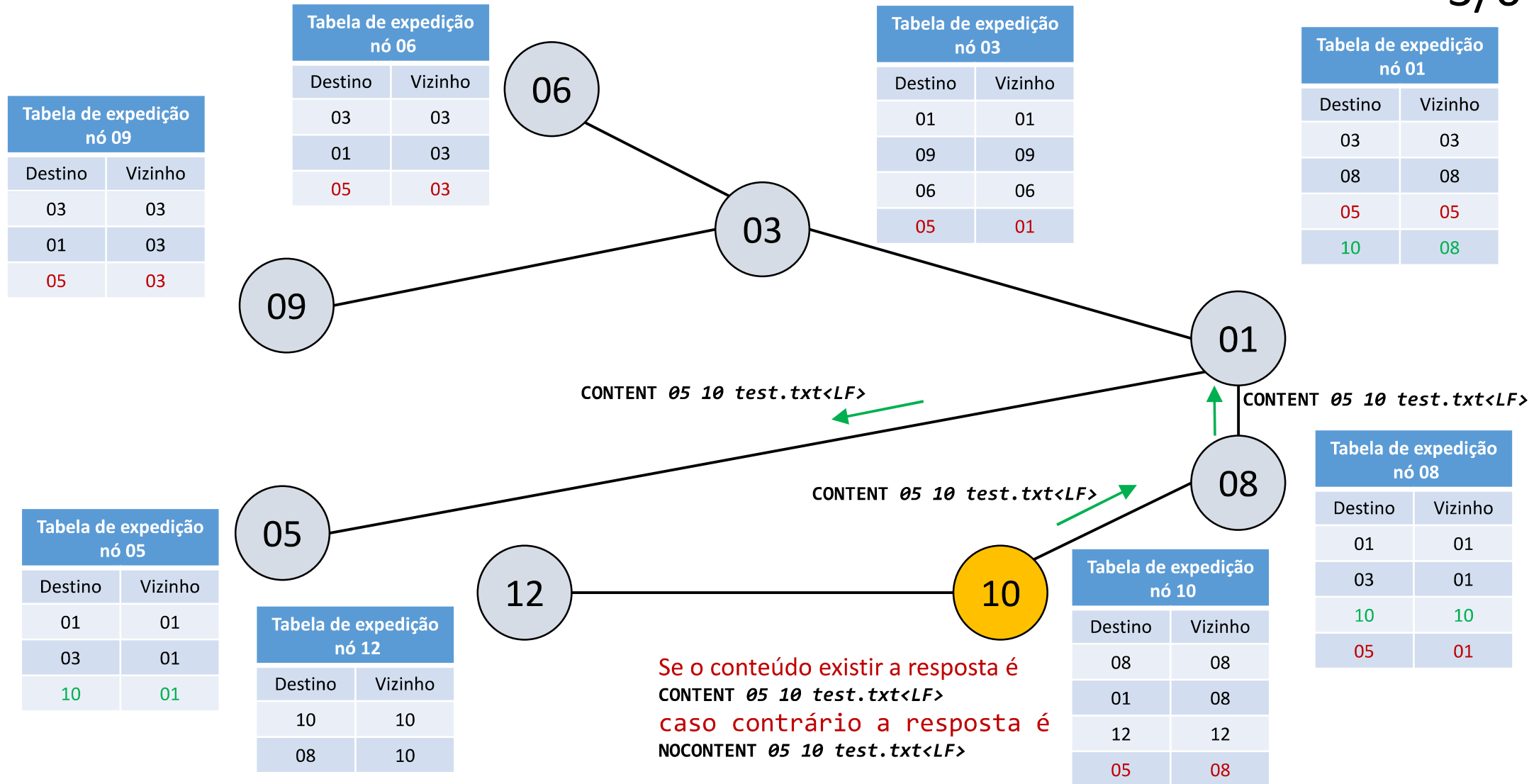
Pesquisa e encaminhamento de conteúdos ^{1/6}



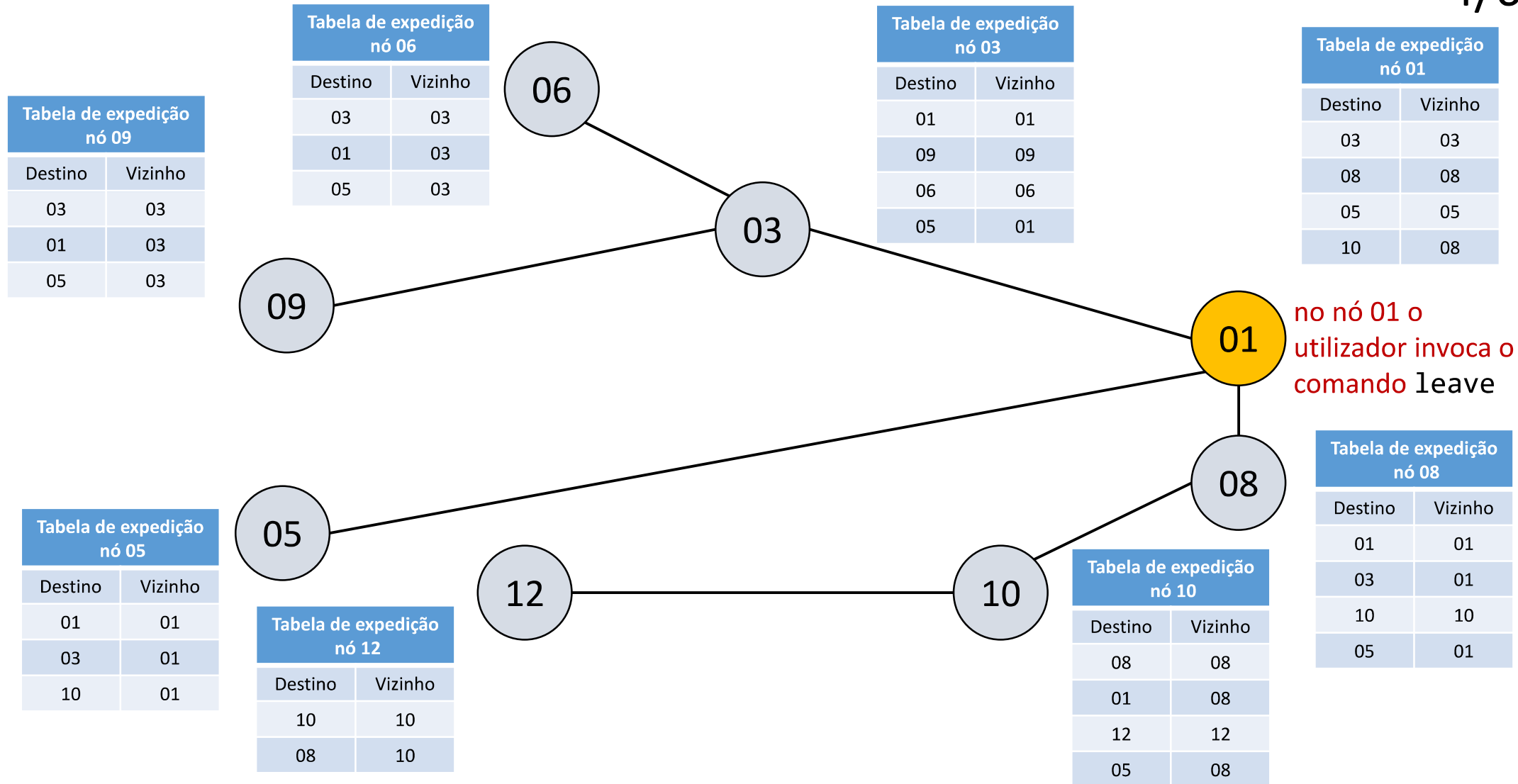
Pesquisa e encaminhamento de conteúdos ^{2/6}



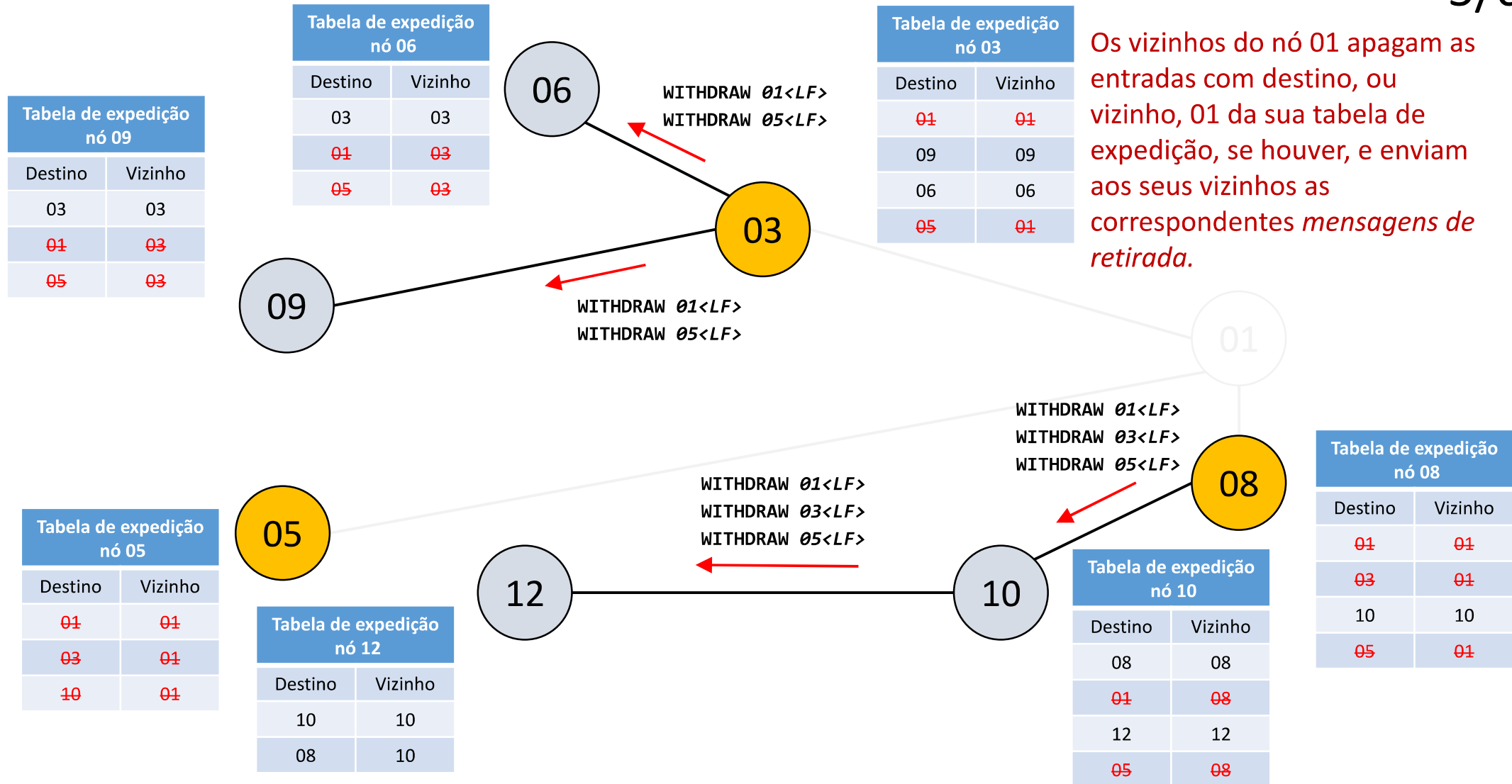
Pesquisa e encaminhamento de conteúdos 3/6



Pesquisa e encaminhamento de conteúdos 4/6



Pesquisa e encaminhamento de conteúdos ^{5/6}



Pesquisa e encaminhamento de conteúdos 6/6

