

RCI 2022/2023

Transferência de Conteúdos em Redes Arbóreas

Rede Arbórea

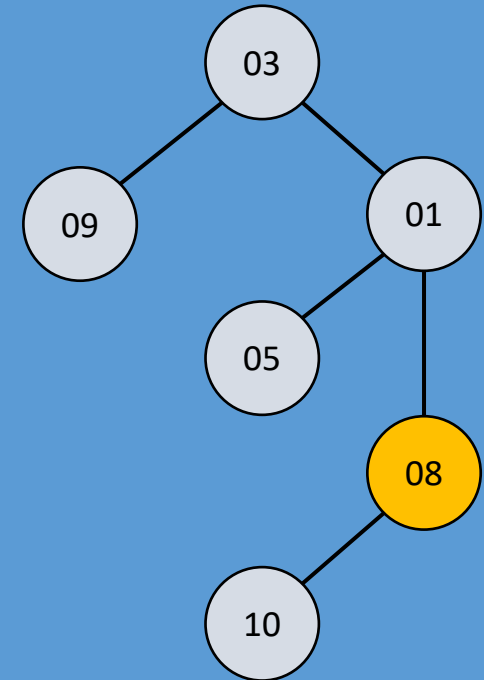
- Cada rede tem um identificador único (ex: 001, 002...).
- Cada rede é composta por nós, com identificadores únicos a cada rede (ex: 01, 02...), interligados em árvore.
- Cada nó contém um conjunto de conteúdos com nomes únicos a cada nó (ex: exame_RCI.txt, bibliografia_RCI.pdf...).
- Cada nó dispõe 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

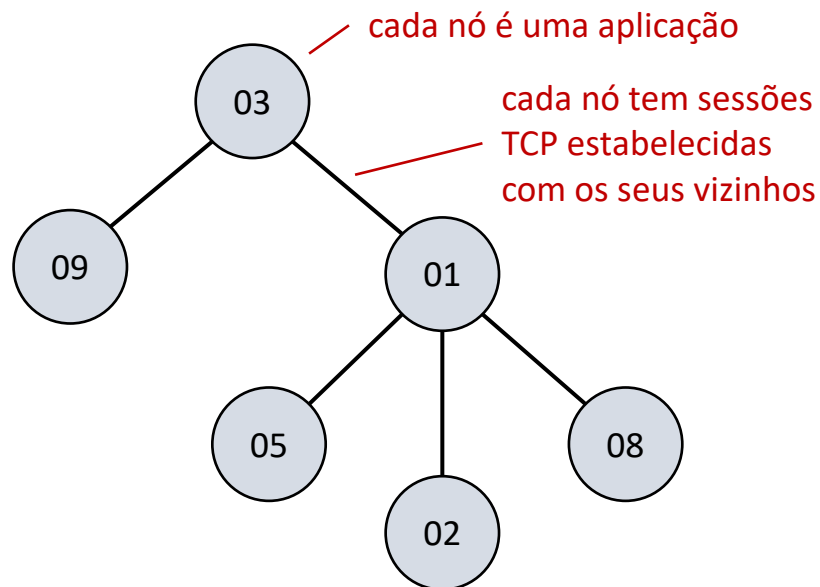
Lista de conteúdos do nó 08:

- exame_RCI.txt
- bibliografia_RCI.pdf

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



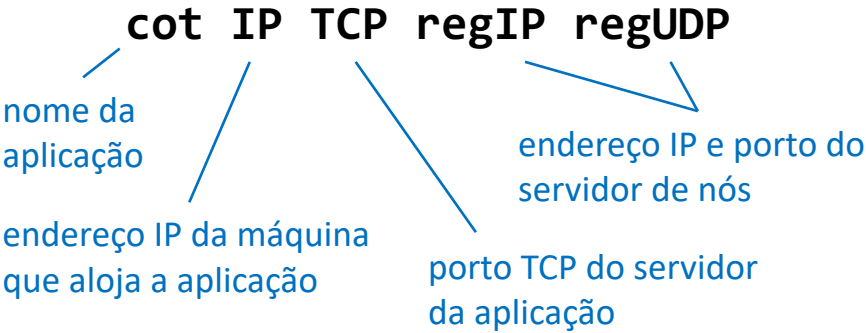
Concretização da rede arbórea



Comandos da interface de utilizador da aplicação

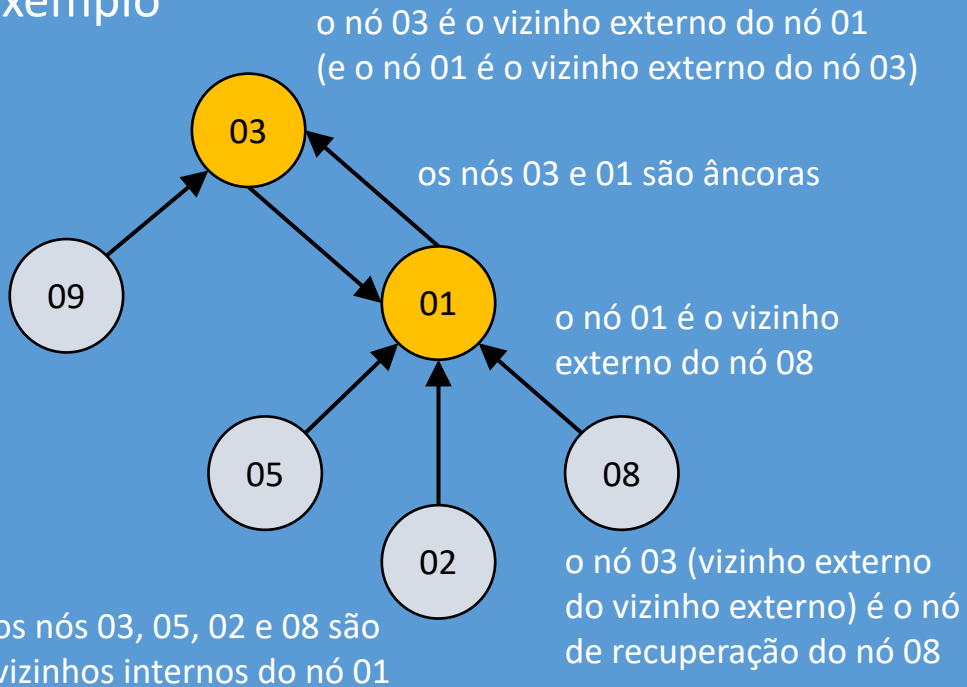
Comando	Descrição
<code>join net id</code>	Entrada de um nó na rede net com identificador id .
<code>djoin net id bootid bootIP bootTCP</code>	Entrada de um nó na rede net com identificador id , que se sabe ser único na rede. É passado à aplicação o identificador e o contacto de um nó da rede, através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

Invocação da aplicação

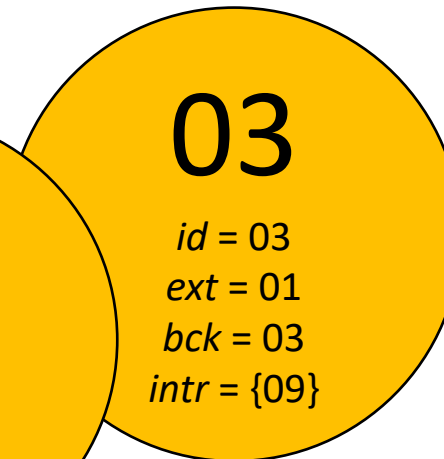
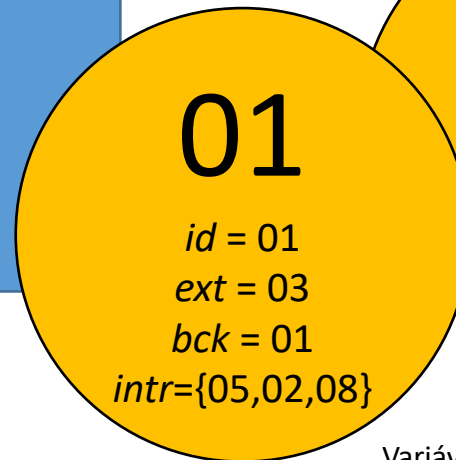
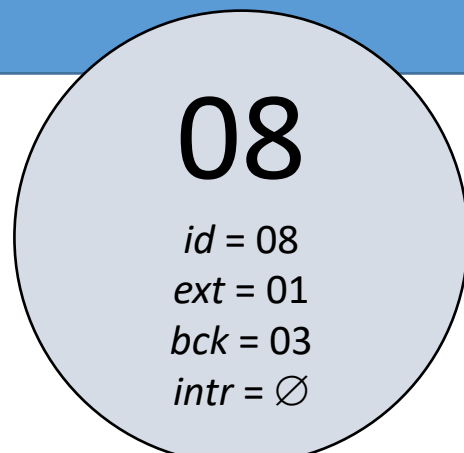


Topologia da Rede Arbórea

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:

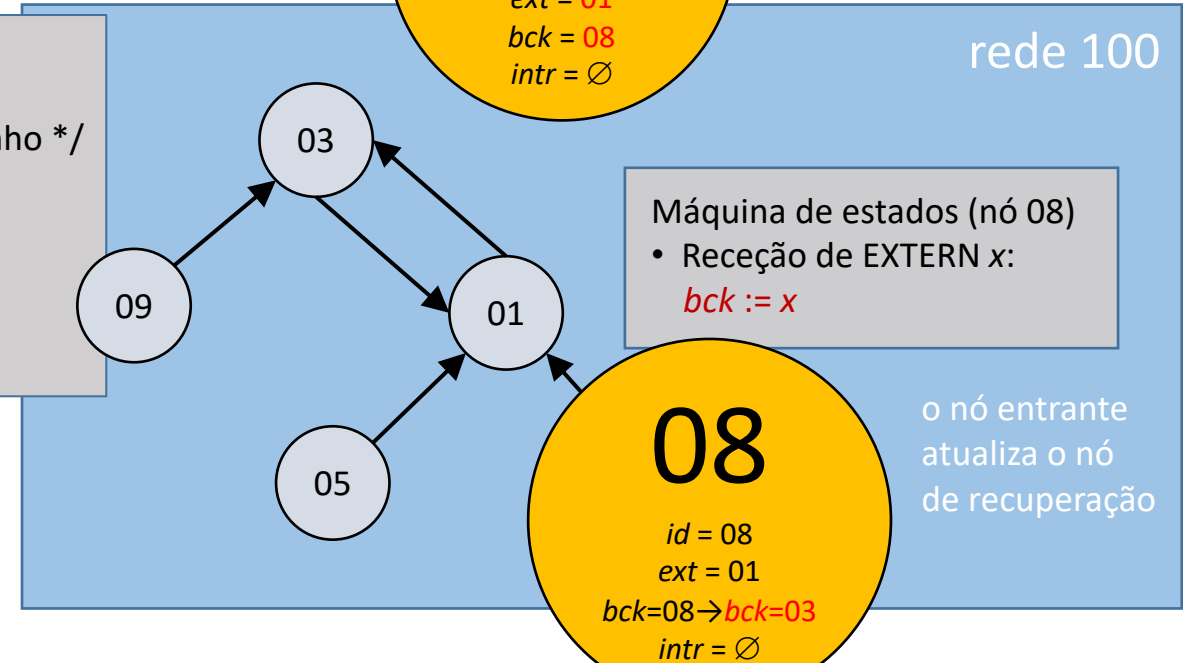
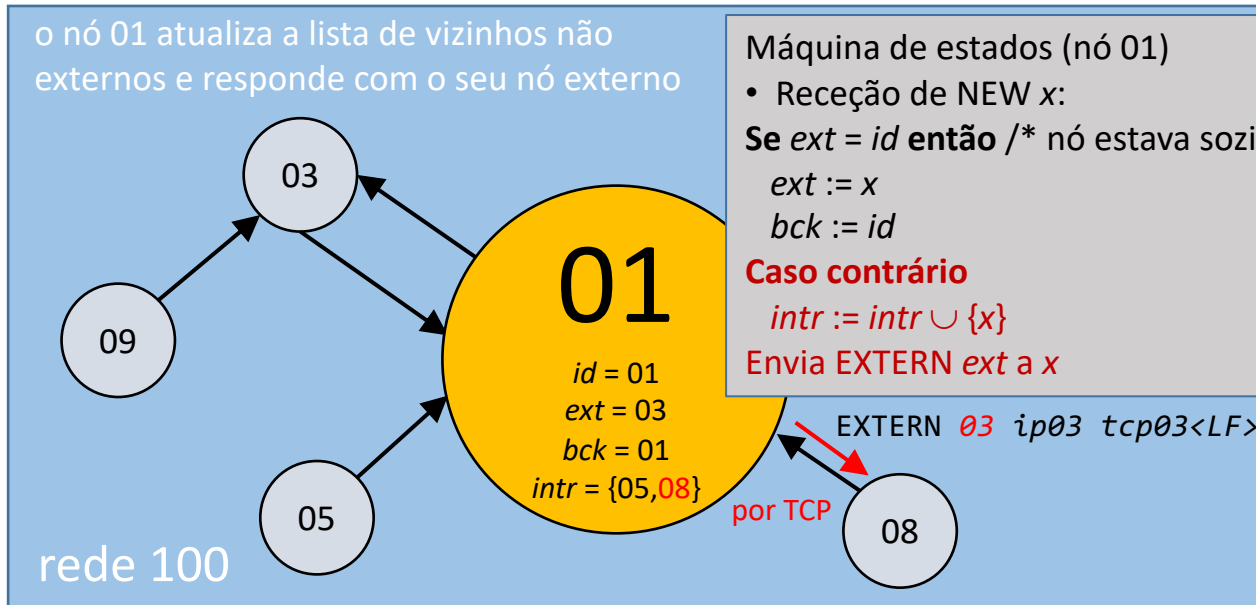
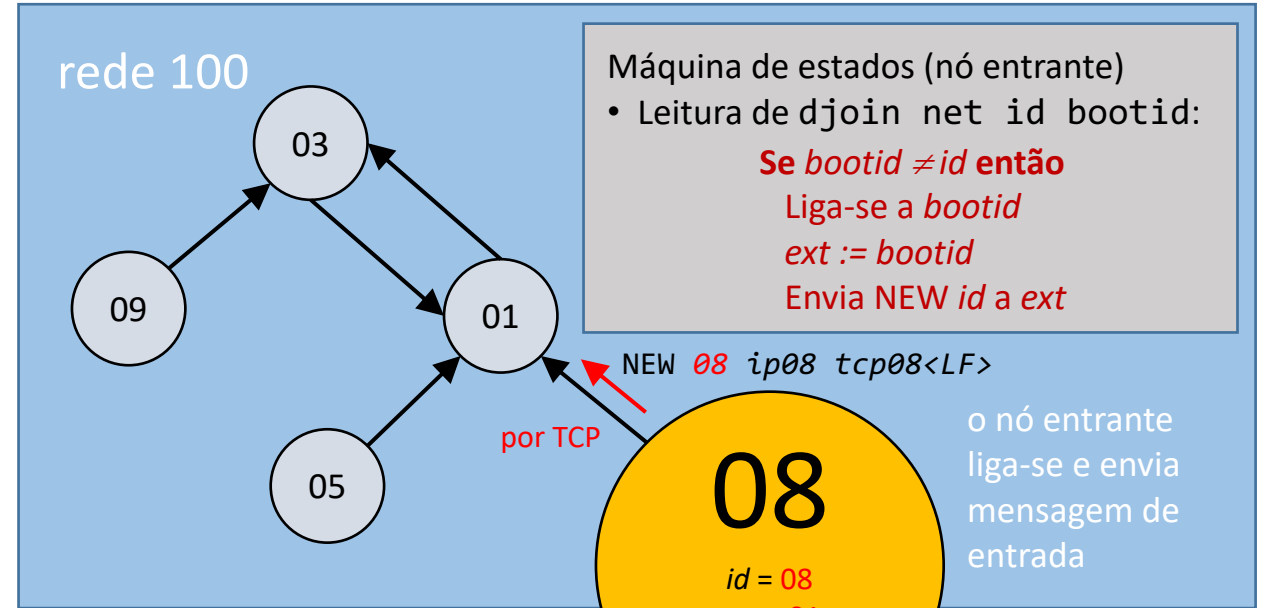
- *id* - identificador do nó
- *ext* - *id* do vizinho externo, inicialmente *ext* := *id*
- *bck* - *id* do vizinho de recuperação, inicialmente *bck* := *id*
- *intr* - conjunto com os ids dos vizinhos não externos, inicialmente *intr* := \emptyset

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

1/2

Variáveis:

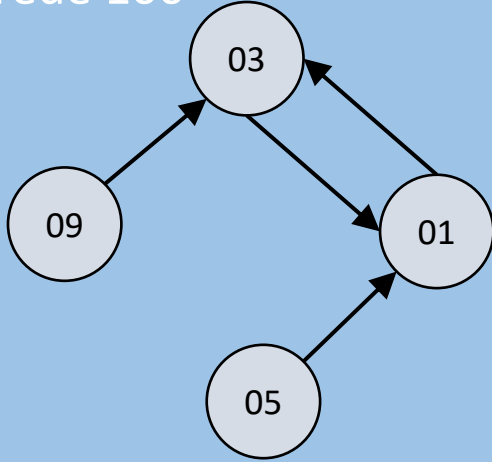
- *id* - identificador do nó
- *ext* - *id* do vizinho externo, inicialmente *ext* := *id*
- *bck* - *id* do vizinho de recuperação, inicialmente *bck* := *id*
- *intr* - conjunto com os ids dos vizinhos não externos, inicialmente *intr* := \emptyset



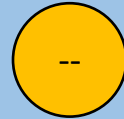
Topologia: entrada de um nó

2/2

rede 100



nó entrante

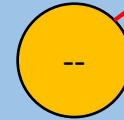


no nó entrante o utilizador
invoca o comando
`join 100 08`

Servidor de nós

por UDP

NODES 100



o nó entrante pede ao
servidor de nós informação
sobre os nós existentes na
rede 007

Servidor de nós

por UDP

NODESLIST 100
03 ip03 tcp03
01 ip01 tcp01
09 ip09 tcp09
05 ip05 tcp05



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:

03 ip03 tcp03
01 ip01 tcp01
09 ip09 tcp09
05 ip05 tcp05



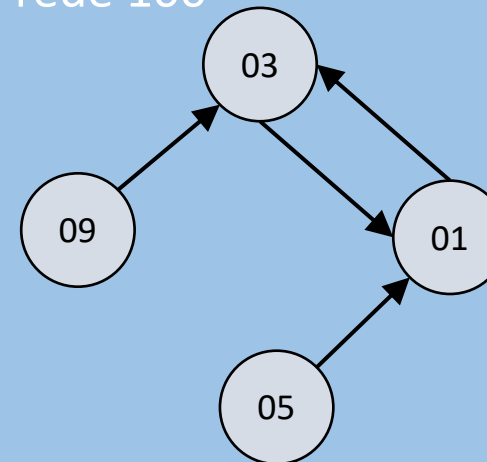
e procede como se o utilizador
tivesse invocado o comando

`djoin 100 08 01 ip01 tcp01`

...

(ver slide anterior)

rede 100



Servidor de nós

OKREG

por UDP

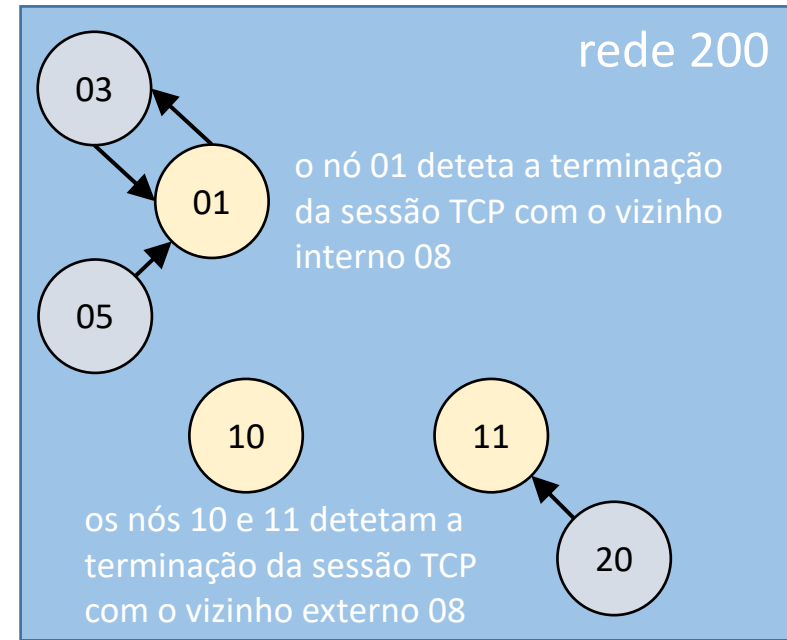
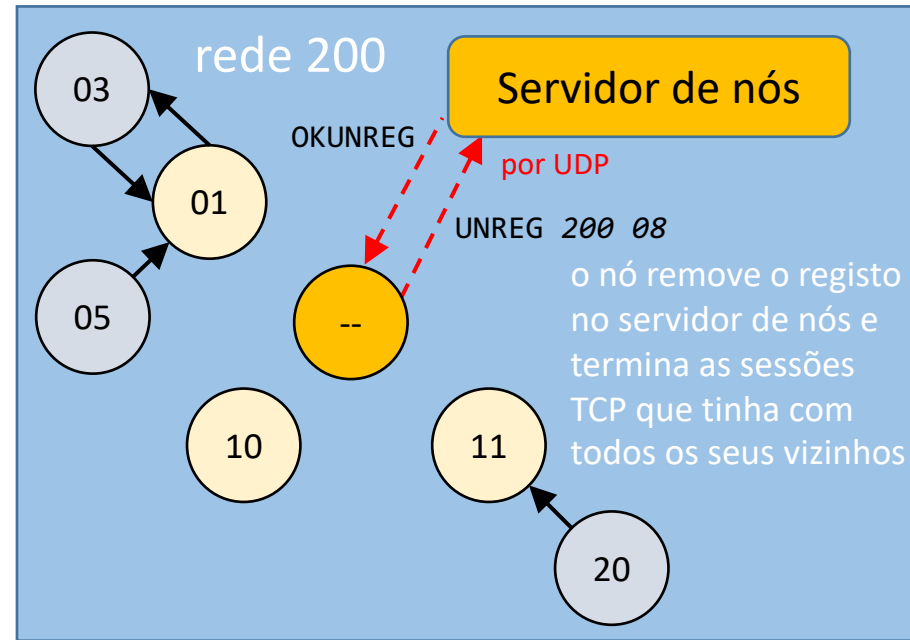
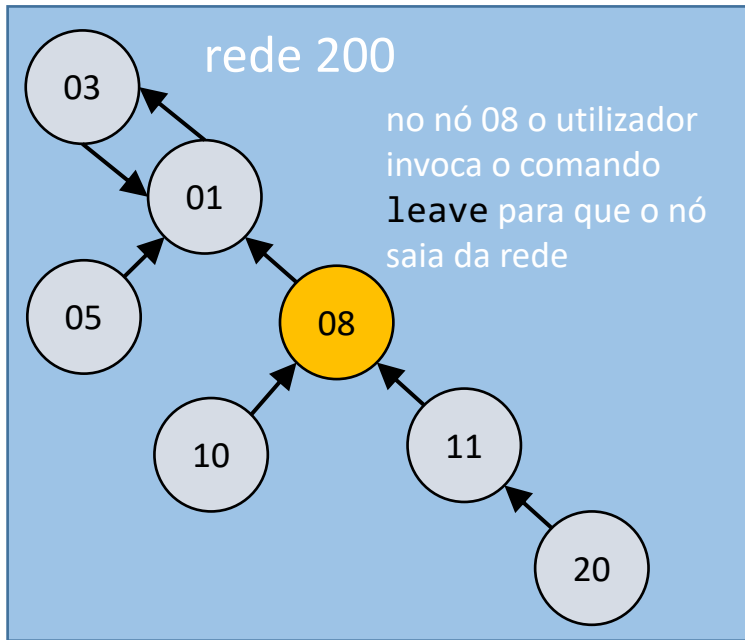
REG 100 08 ip08 tcp08

08

id = 08
ext = 01
bck=03
intr = ∅

o nó entrante
registra-se no
servidor de nós

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



- Terminação da sessão com o vizinho x

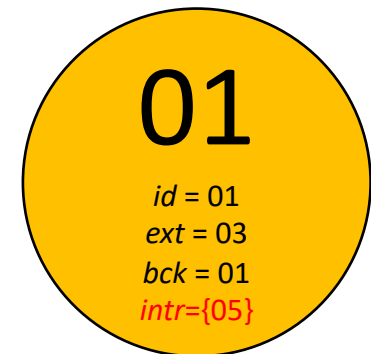
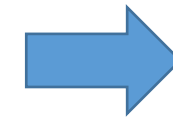
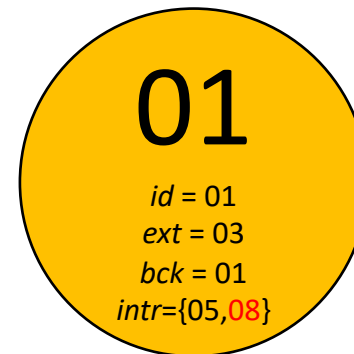
Se $x \in \text{intr}$ então /* term. com vizinho que não é externo */
 $\text{intr} := \text{intr} - \{x\}$

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 */
 $\text{ext} := \text{id}$

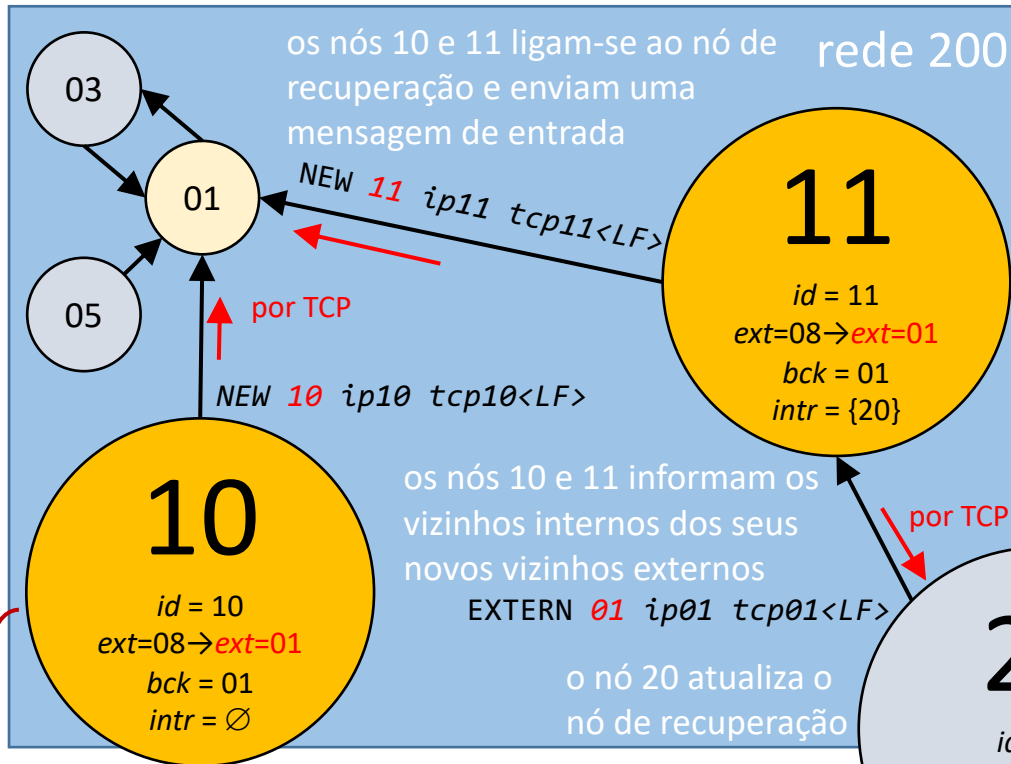
Máquina de estados (nó 01)



Variáveis:

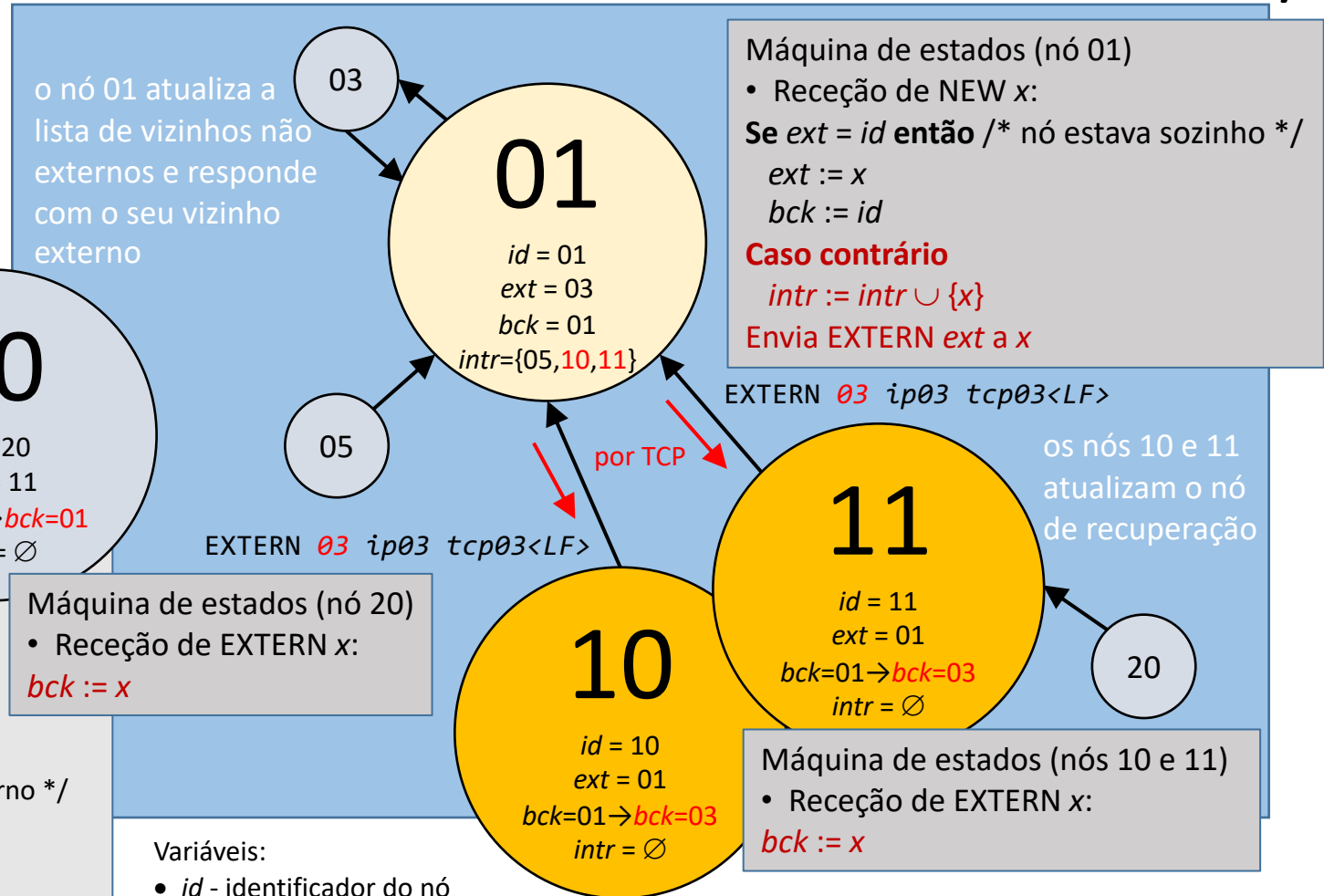
- id - identificador do nó
- ext - id do vizinho externo, inicialmente $\text{ext} := \text{id}$
- bck - id do vizinho de recuperação, inicialmente $\text{bck} := \text{id}$
- intr - conjunto com os ids dos vizinhos não externos, inicialmente $\text{intr} := \emptyset$

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



- 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
(nós 10 e 11)



Máquina de estados (nó 01)

- Receção de NEW x:
- Se $\text{ext} = \text{id}$ então /* nó estava sozinho */
 $\text{ext} := x$
 $\text{bck} := \text{id}$
- Caso contrário**
 $\text{intr} := \text{intr} \cup \{x\}$
 Envia EXTERN ext a x

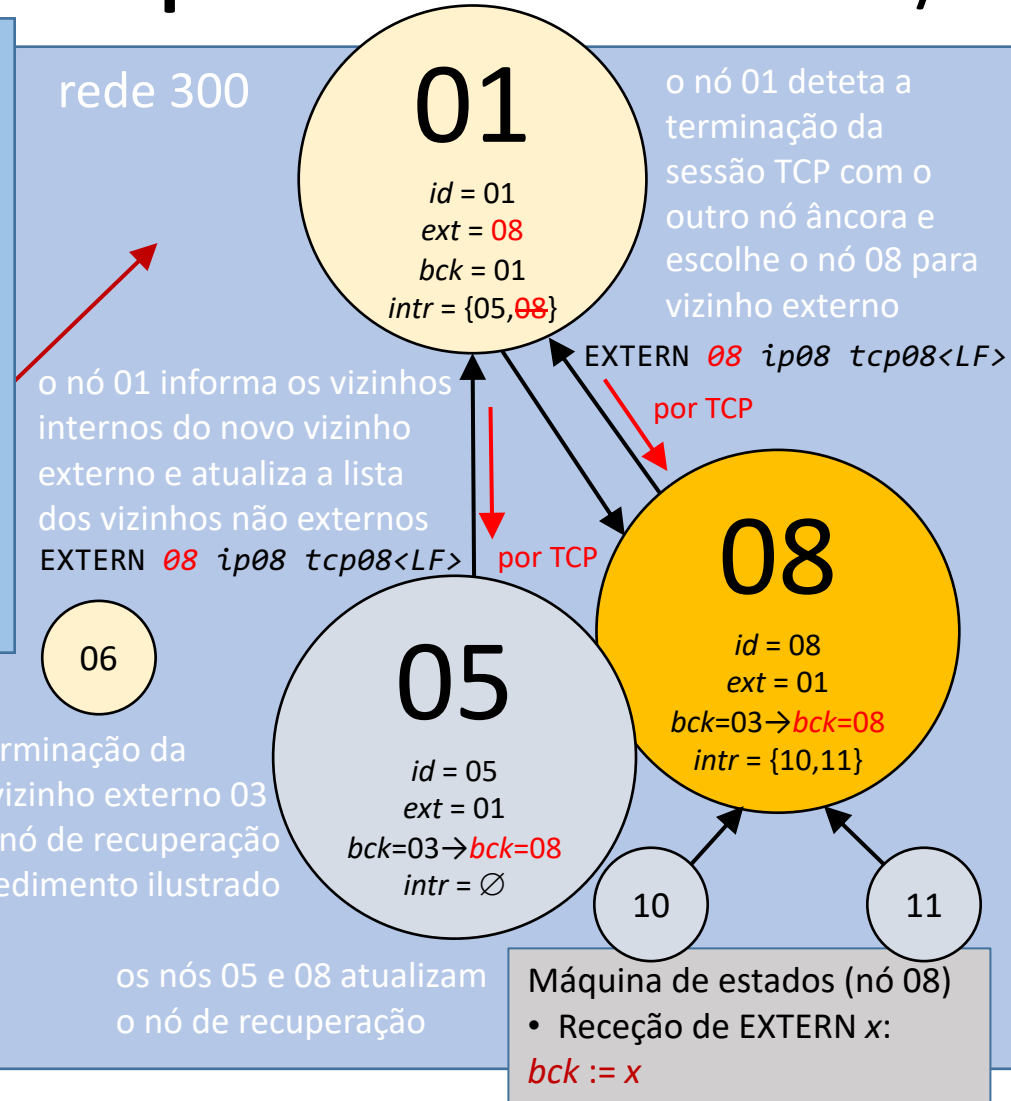
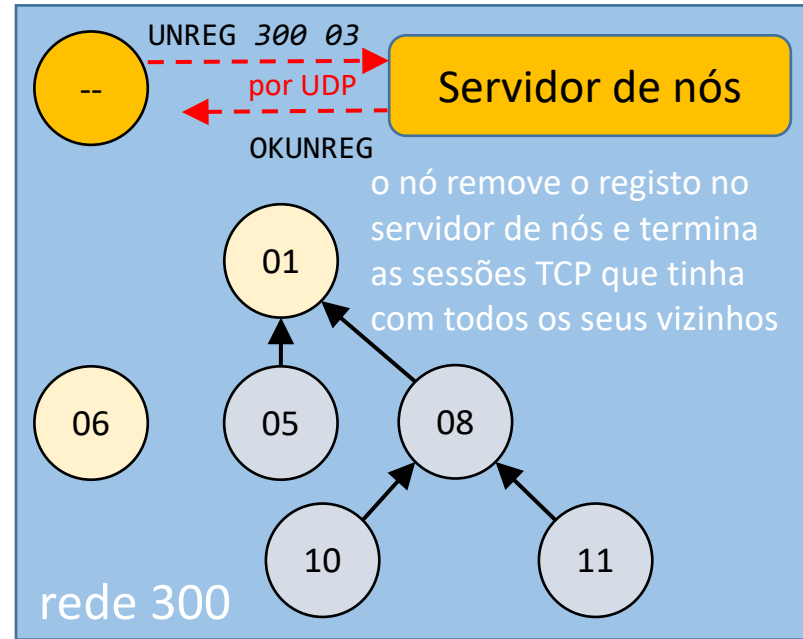
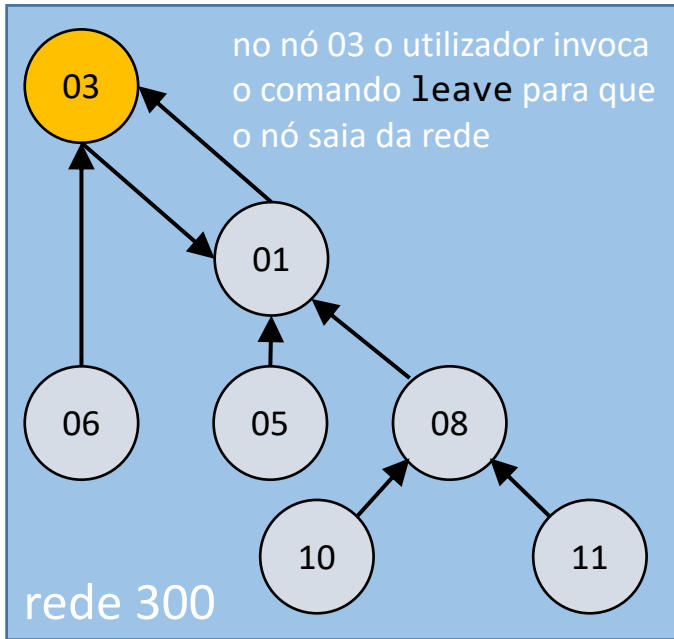
Máquina de estados (nó 20)

- Receção de EXTERN x:
- $\text{bck} := x$

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 é âncora 3/3



- Terminação da sessão com o vizinho x

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

Caso contrário se $\text{bck} \neq \text{id}$ **então** $\text{ /* term. com vizinho externo, não é âncora */ }$
 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** $\text{ /* term. com v. e., é âncora, tem vizinho interno */ }$
 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 $\text{ /* nó ficou sozinho */ }$
 $\text{ext} := \text{id}$

Máquina de estado
 (nó 0)

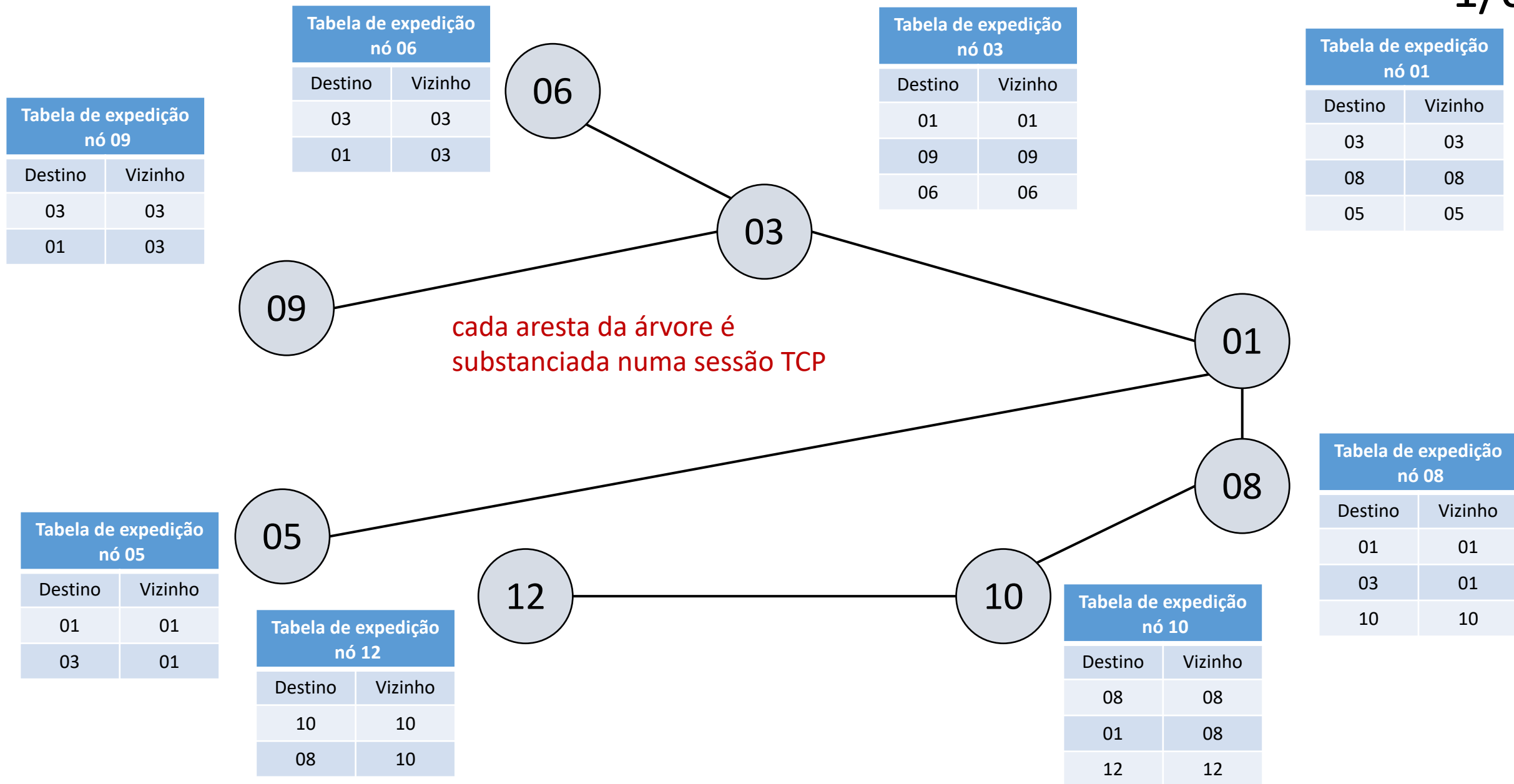
o nó 06 deteta a terminação da sessão TCP com o vizinho externo 03 e há-de ligar-se ao nó de recuperação 01 através do procedimento ilustrado no slide anterior

os nós 05 e 08 atualizam
o nó de recuperação

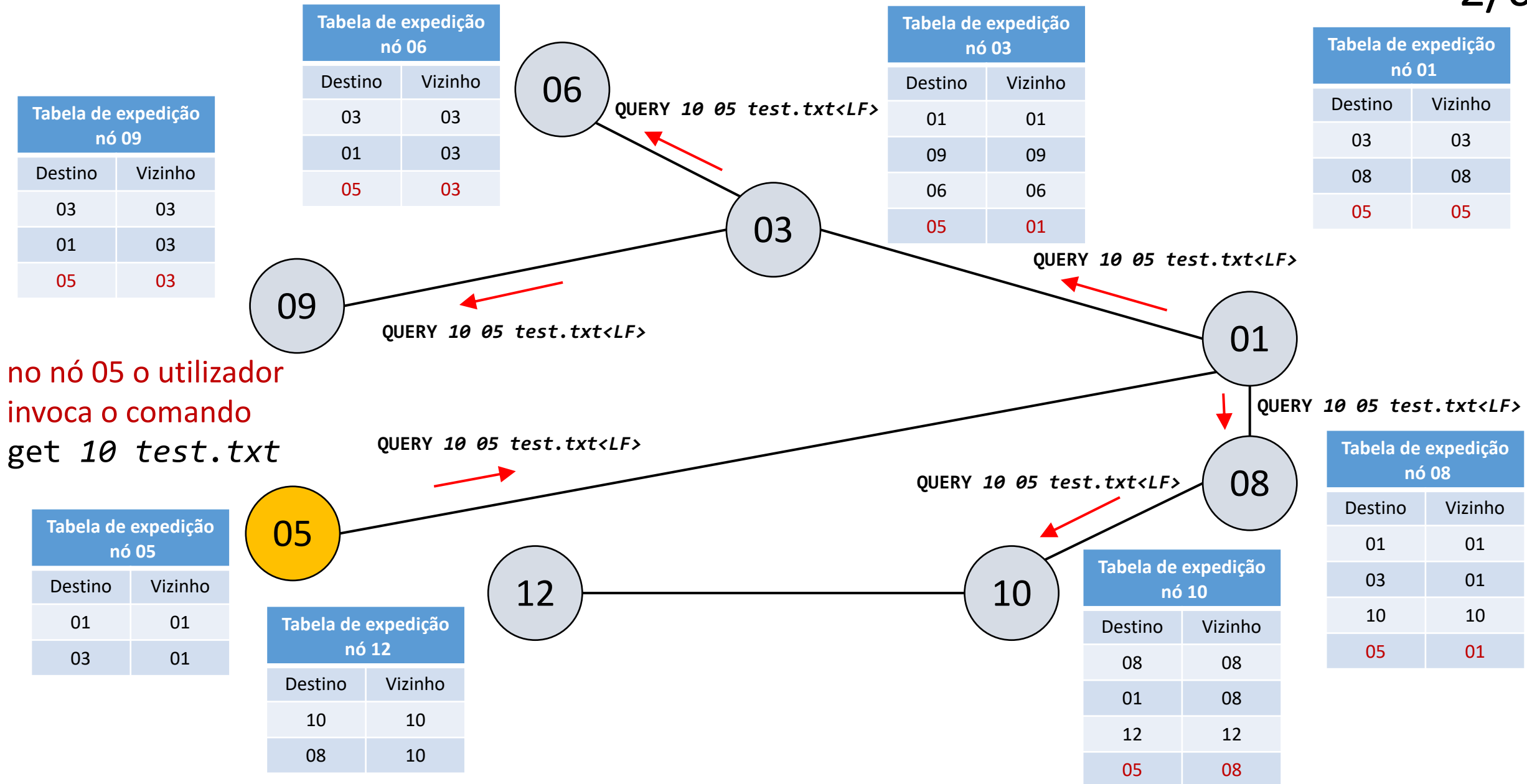
Máquina de estados (nó 08)

- Receção de EXTERN x :
 $bck := x$

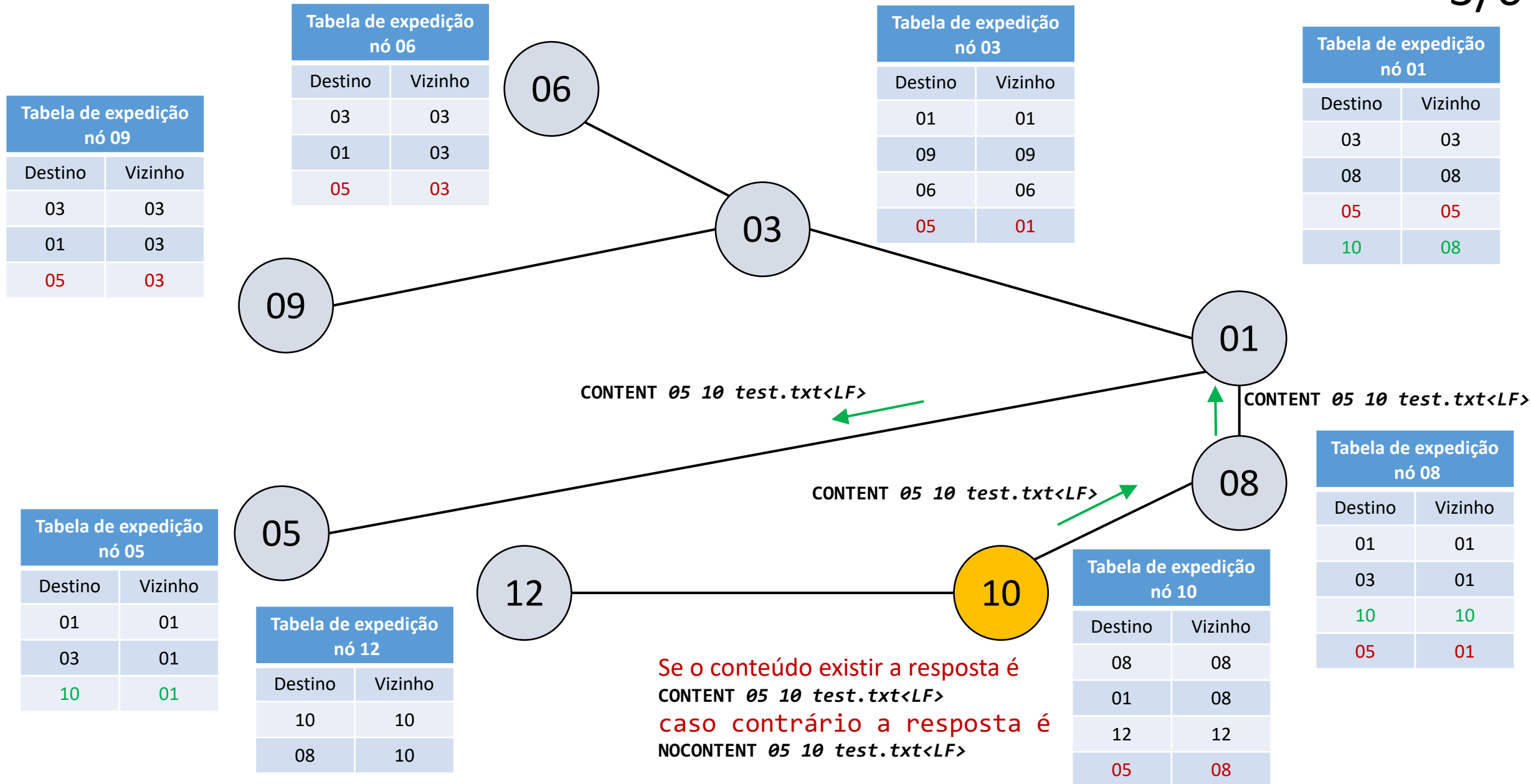
Pesquisa e encaminhamento de conteúdos ^{1/6}



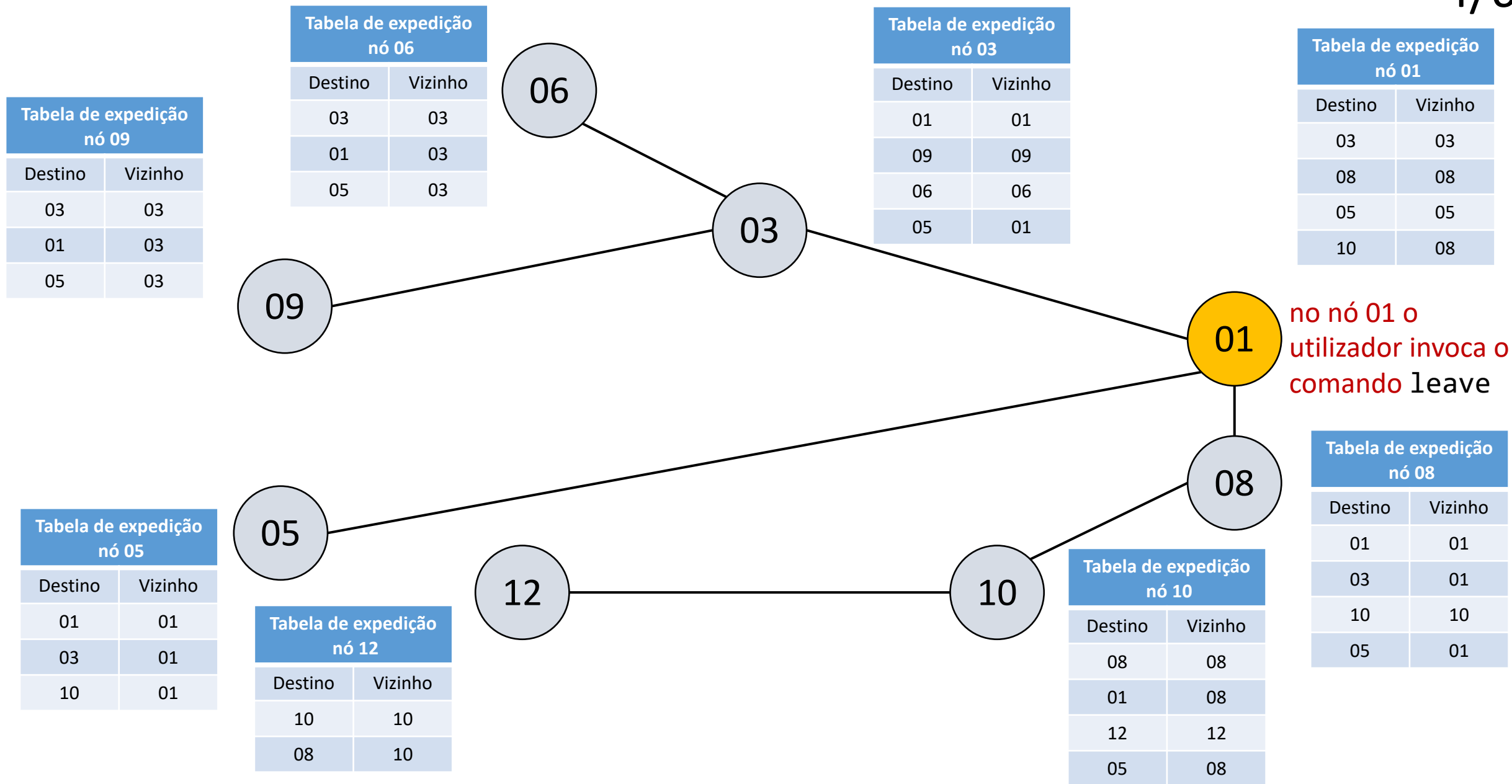
Pesquisa e encaminhamento de conteúdos ^{2/6}



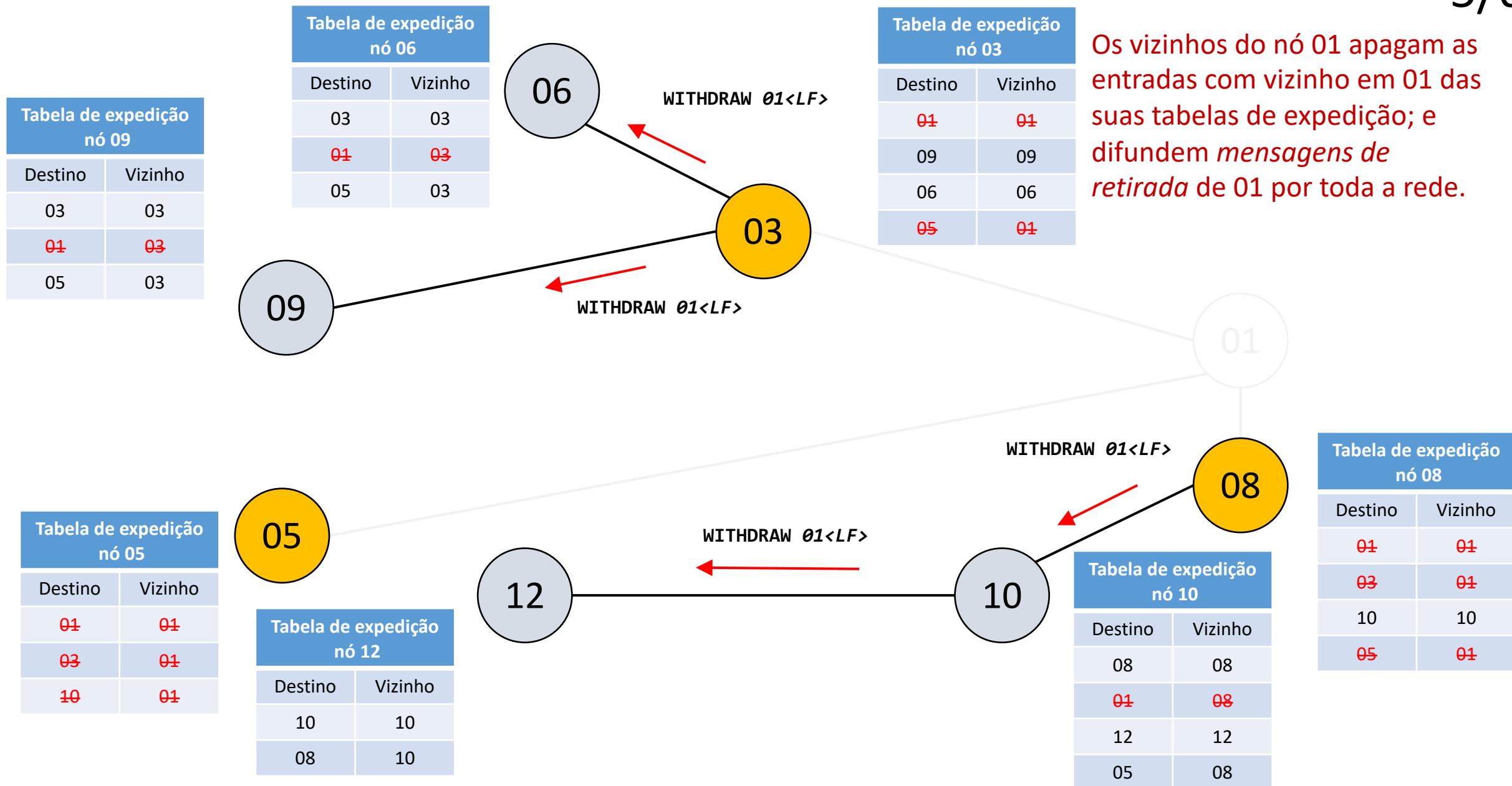
Pesquisa e encaminhamento de conteúdos ^{3/6}



Pesquisa e encaminhamento de conteúdos 4/6



Pesquisa e encaminhamento de conteúdos 5/6



Pesquisa e encaminhamento de conteúdos 6/6

