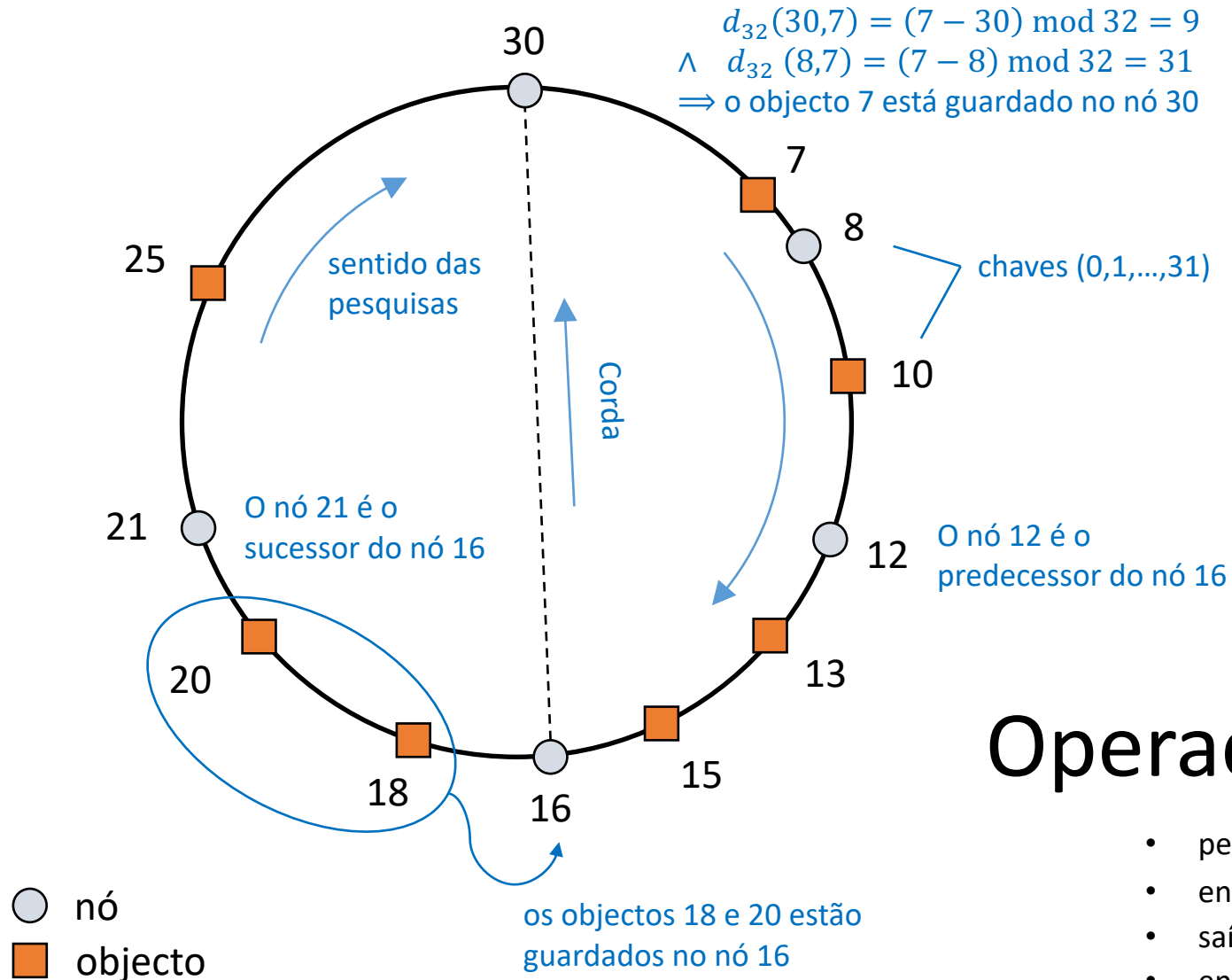


# RCI Base-de-Dados em Anel com Cordas

# Anel com cordas

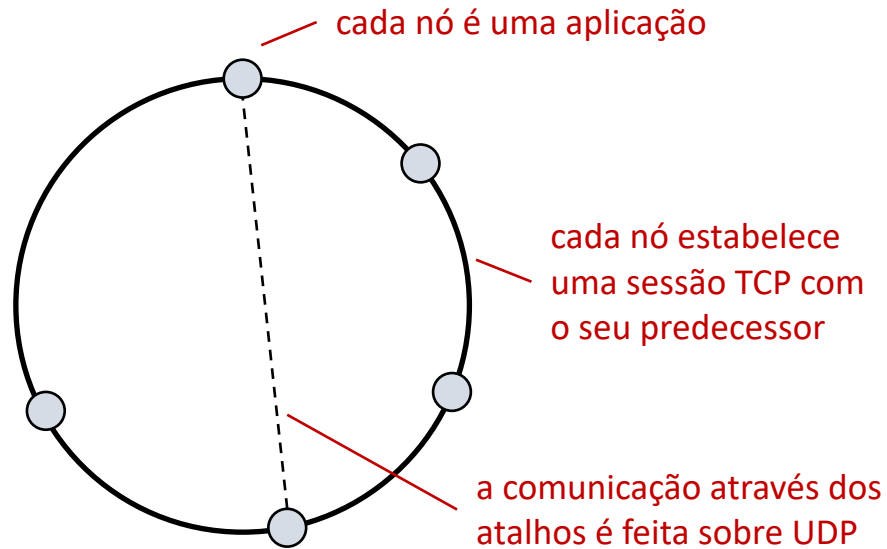


- Chaves entre 0 e 31 ordenadas circularmente
- Cada objeto e cada nó tem uma chave única entre 0 e 31.
- A distância da chave  $k$  à chave  $l$  é  $d_N(k, l) = (l - k) \bmod N$ .
- Cada objeto está albergado no nó para o qual a distância da chave do nó à chave do objeto é a menor.
- O sucessor do nó  $i$ ,  $s(i)$ , é o nó que está à menor distância de  $i$ .
- O predecessor do nó  $i$ ,  $p(i)$ , é o nó que tem  $i$  como sucessor,  $s(p(i)) = p(s(i)) = i$ .
- Um nó pode ter no máximo um atalho,  $a(i)$ .

## Operações no anel

- pesquisa de uma chave;
- entrada de um nó sabendo a sua posição no anel;
- saída de um nó;
- entrada de um nó não sabendo a sua posição no anel.

# Concretização do anel com cordas



## Invocação da aplicação

`ring i i.IP i.port`

nome da aplicação

chave do nó

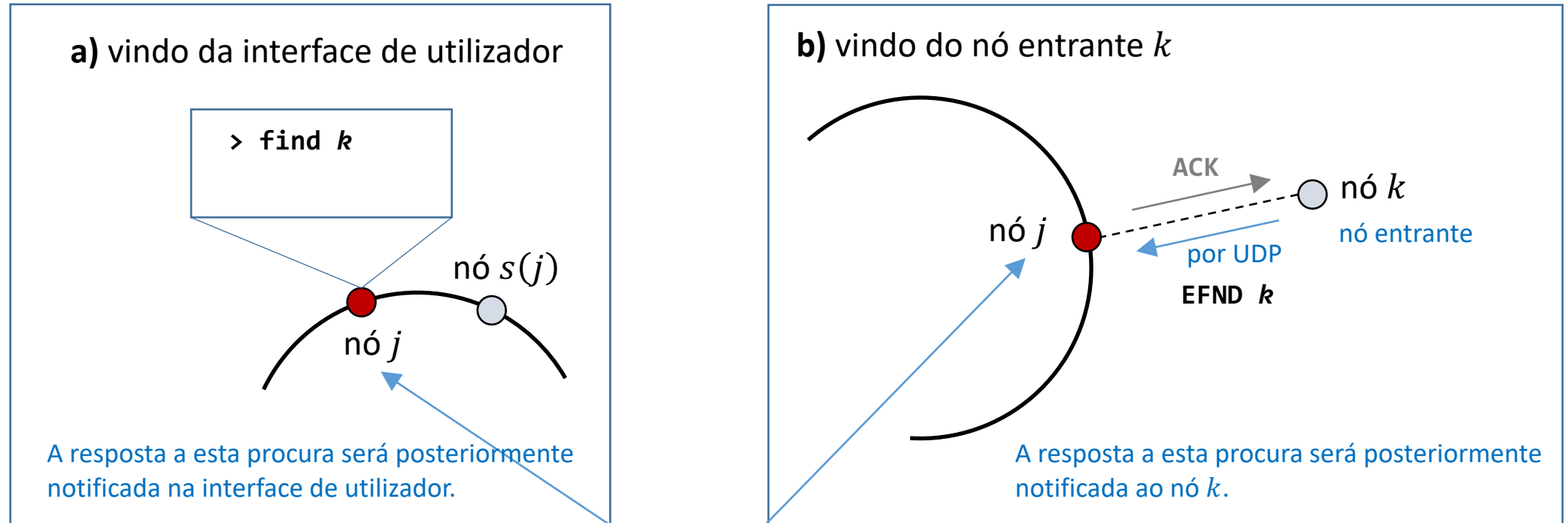
endereço IP e porto usados nos servidores TCP e UDP do nó

## Comandos da interface de utilizador da aplicação

Comando	Descrição
<code><u>n</u>ew</code>	Criação de um anel contendo apenas o nó.
<code><u>b</u>entry <i>boot boot.IP boot.port</i></code>	Entrada do nó no anel ao qual pertence o nó <i>boot</i> com endereço IP <i>boot.IP</i> e porto <i>boot.port</i> .
<code><u>p</u>entry <i>pred pred.IP pred.port</i></code>	Entrada do nó no anel sabendo que o seu predecessor será o nó <i>pred</i> com endereço IP <i>pred.IP</i> e porto <i>pred.port</i> .
<code><u>c</u>hord <i>i i.IP i.port</i></code>	Criação de um atalho para o nó <i>i</i> com endereço IP <i>i.IP</i> e porto <i>i.port</i> .
<code><u>s</u>how</code>	Mostra do estado do nó, consistindo em: (i) a sua chave, endereço IP e porto; (ii) a chave, endereço IP e porto do seu sucessor; (iii) a chave, endereço IP e porto do seu predecessor; e, por fim, (iv) a chave, endereço IP e porto do seu atalho, se houver.
<code><u>f</u>ind <i>k</i></code>	Procura da chave <i>k</i> , retornando a chave, o endereço IP e o porto do nó à qual a chave pertence.
<code><u>l</u>ease</code>	Saída do nó do anel.
<code><u>e</u>xit</code>	Fecho da aplicação.

# Procura da chave $k$ pelo nó $j$

acontecimentos de entrada com origem exterior ao anel

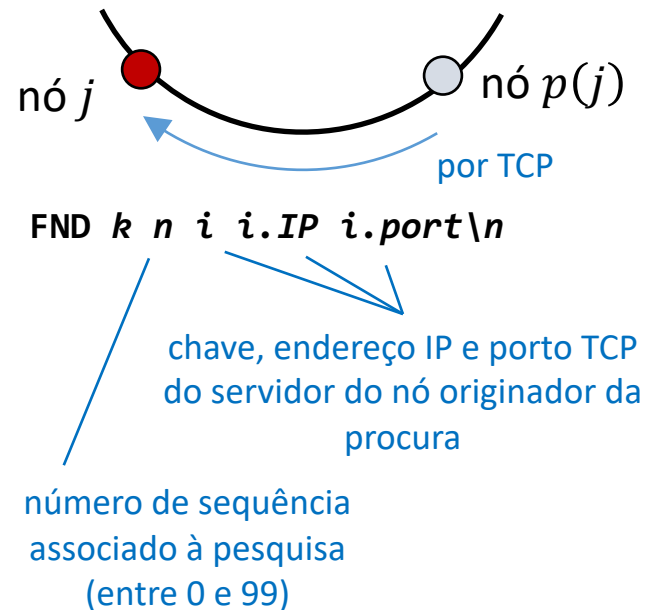


o nó  $j$  é o originador da procura

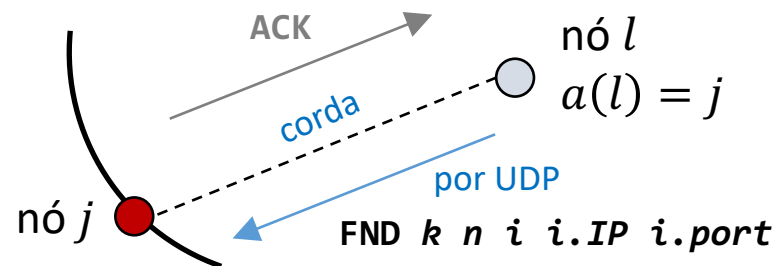
# Procura da chave $k$ pelo nó $j$

acontecimentos de entrada com origem noutros nós do anel

c) vindo do predecessor do nó  $j$



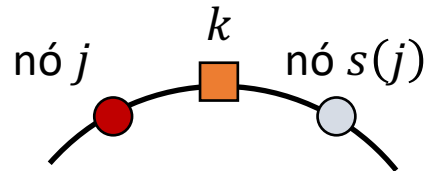
d) vindo do atalho de outro nó para o nó  $j$



Nota: cada nó tem no máximo um atalho, mas pode ser atalho de múltiplos outros nós.

# Procura da chave $k$ no nó $j$ - acções

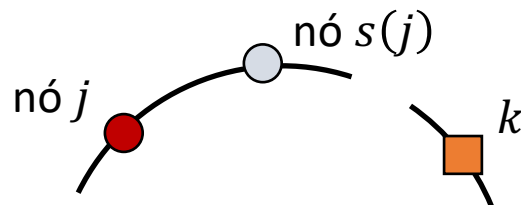
Se  $d_N(j, k) < d_N(s(j), k)$  a chave  $k$  está no nó  $j$ , então a procura por  $k$  termina.



Se  $j$  é o originador da procura, então ele notifica o utilizador ou o nó entrante, consoante o caso.

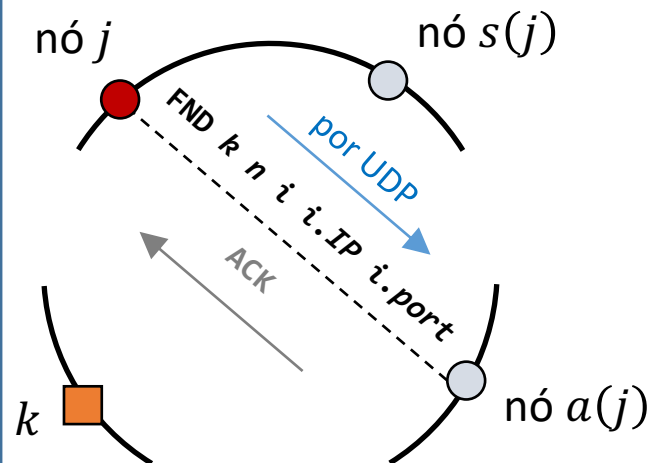
Se  $j$  não é o originador da procura, então ele origina uma resposta destinada ao originador da procura.

Se a chave  $k$  não está no nó  $j$ , então ele delega a procura da chave  $k$ .

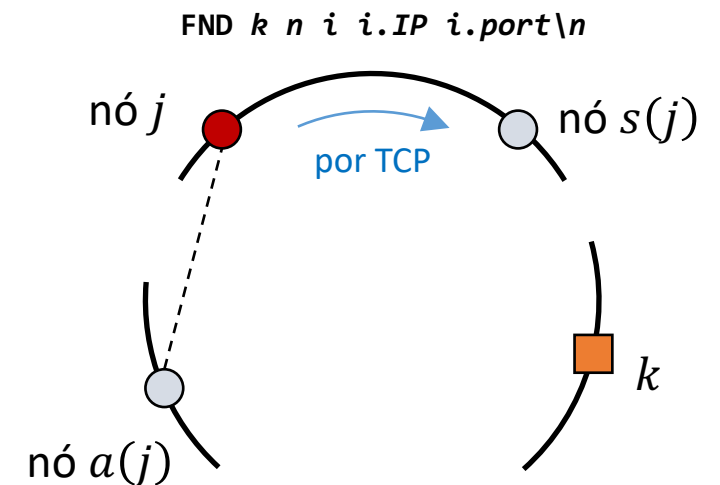


## Delegação da procura da chave $k$

Se  $a(j)$  existir e  $d_N(a(j), k) < d_N(s(j), k)$ , então a procura é delegada em  $a(j)$ .



Caso contrário, a procura é delegada em  $s(j)$ .

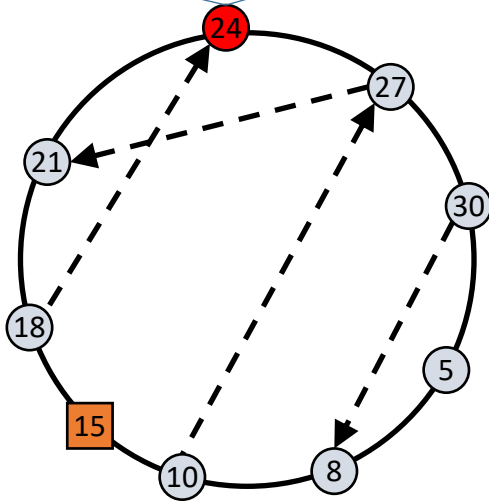


As respostas são originadas pelos nós que detêm as chaves que foram procuradas e têm um tratamento em tudo igual às procuras, diferenciando no comando que é **RSP** em vez de **FND**.

# Exemplo de procura - I

interface de utilizador

> find 15



O nó 24 é encarregado de procurar a chave 15.  
É criado um identificador, entre 0 e 99, por exemplo 88, que ficará associado a esta procura.  
O nó 24 será o originador da procura.

$$d_{32}(24,15) = 23$$

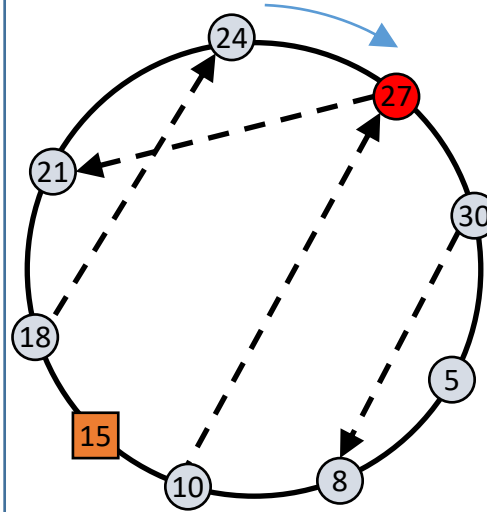
$$d_{32}(27,15) = 20$$

$$d_{32}(24,15) > d_{32}(27,15)$$

Conclusão: a chave 15 não está no nó 24.  
Assim, a procura será delegada.  
Não existe atalho no nó 24.  
A procura é delegada no nó 27.

(por TCP)

FND 15 88 24 24.IP 24.port\n



O nó 27 é encarregado de procurar a chave 15.  
O nó 24 é o originador da procura.

$$d_{32}(27,15) = 20$$

$$d_{32}(30,15) = 17$$

$$d_{32}(27,15) > d_{32}(30,15)$$

Conclusão: a chave 15 não está no nó 27.  
Assim, a procura será delegada.  
Existe atalho para o nó 21.

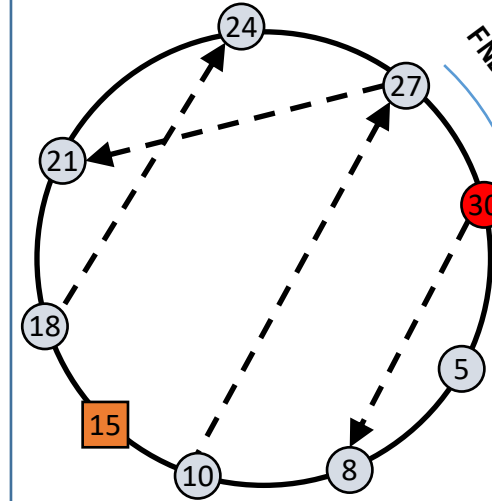
$$d_{32}(21,15) = 26$$

$$d_{32}(21,15) > d_{32}(30,15)$$

Conclusão: o nó 30 está mais perto da chave 15.  
A procura é delegada no nó 30.

(por TCP)

FND 15 88 24 24.IP 24.port\n



O nó 30 é encarregado de procurar a chave 15.  
O nó 24 é o originador da procura.

$$d_{32}(30,15) = 17$$

$$d_{32}(5,15) = 10$$

$$d_{32}(30,15) > d_{32}(5,15)$$

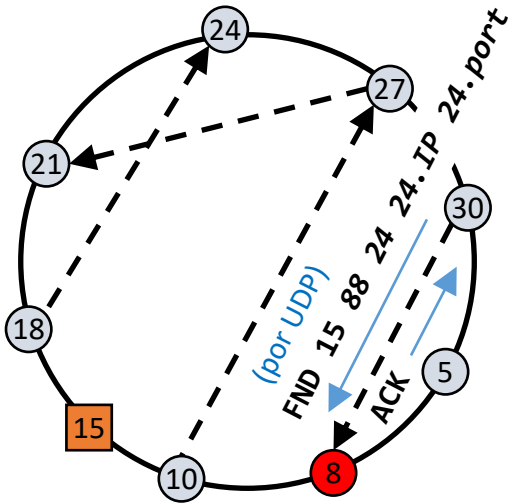
Conclusão: a chave 15 não está no nó 30.  
Assim, a procura será delegada.  
Existe atalho para o nó 8.

$$d_{32}(8,15) = 7$$

$$d_{32}(8,15) < d_{32}(5,15)$$

Conclusão: o nó 8 está mais perto da chave 15.  
A procura é delegada no nó 8 (através do atalho).

# Exemplo de procura - II



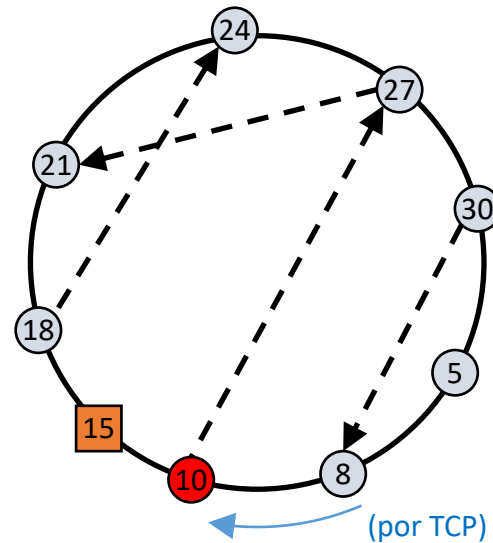
O nó 8 é encarregado de procurar a chave 15.  
O nó 24 é o originador da procura.

$$d_{32}(8,15) = 7$$

$$d_{32}(10,15) = 5$$

$$d_{32}(8,15) > d_{32}(10,15)$$

Conclusão: a chave 15 não está no nó 8.  
Assim, a procura será delegada.  
Não existe atalho no nó 8.  
A procura é delegada no nó 10.



FND 15 88 24 24.IP 24.port\n

O nó 10 é encarregado de procura a chave 15.  
O nó 24 é o originador da procura.

$$d_{32}(10,15) = 5$$

$$d_{32}(18,15) = 29$$

$$d_{32}(10,15) < d_{32}(18,15)$$

Conclusão: a chave 15 está no nó 10; a procura termina.  
A resposta será agora enviada ao nó 24 (originador da procura),  
com o mesmo identificador 88, usado na procura.  
O nó 10 é o originador desta resposta.  
Existe atalho para o nó 27.

$$d_{32}(18,24) = 6$$

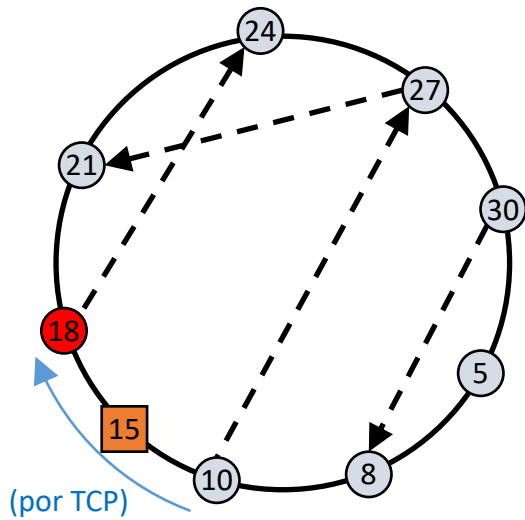
$$d_{32}(27,24) = 29$$

$$d_{32}(27,24) > d_{32}(18,24)$$

Conclusão: o nó 18 está mais perto da chave 24.  
A resposta é delegada no nó 18.



# Exemplo de resposta



RSP 24 88 10 10.IP 10.port\n

O nó 18 é encarregado de responder ao nó 24.  
O nó 10 é o originador da resposta.

$$d_{32}(18,24) = 6$$

$$d_{32}(21,24) = 3$$

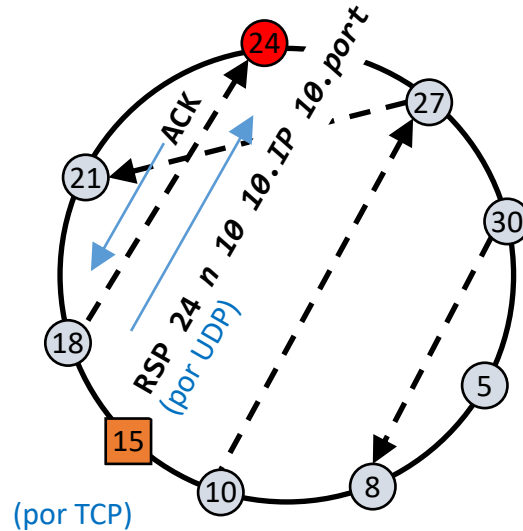
$$d_{32}(18,24) > d_{32}(21,24)$$

Conclusão: a chave 24 não está no nó 18.  
Assim, a resposta será delegada.  
Existe atalho para o nó 24.

$$d_{32}(24,24) = 0$$

$$d_{32}(24,24) < d_{32}(21,24)$$

Conclusão: o nó 24 está mais perto do nó 24.  
A resposta é delegada no nó 24 (através do atalho).



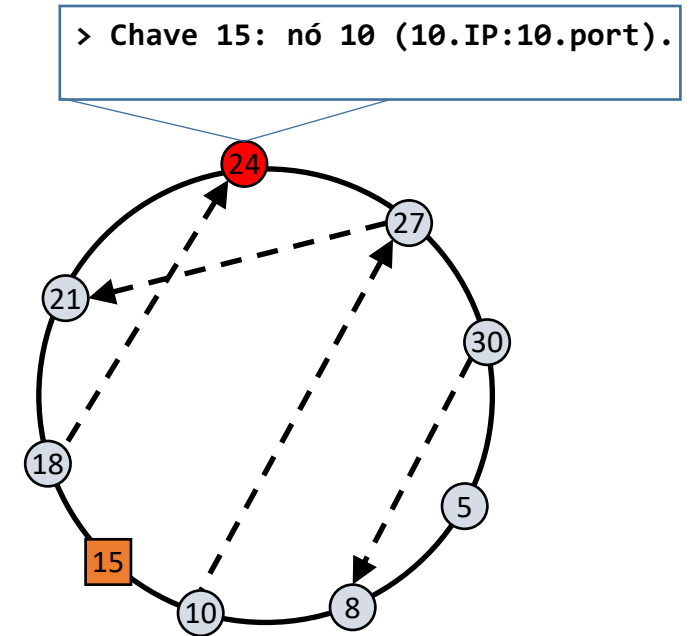
O nó 24 é encarregado de responder ao nó 24.  
O nó 10 é o originador desta resposta.

$$d_{32}(24,24) = 0$$

$$d_{32}(27,24) = 29$$

$$d_{32}(24,24) < d_{32}(27,24)$$

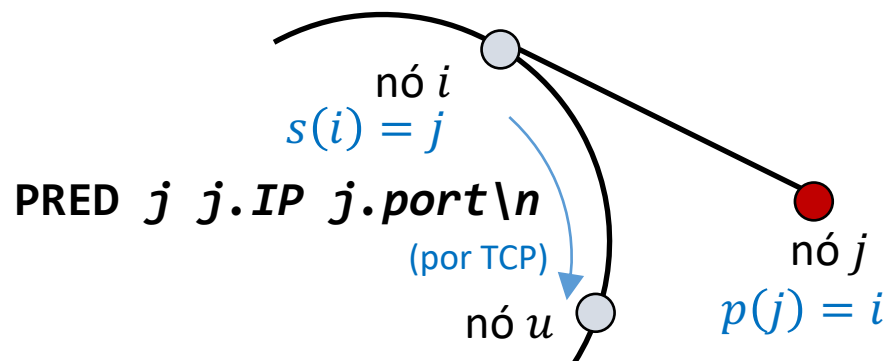
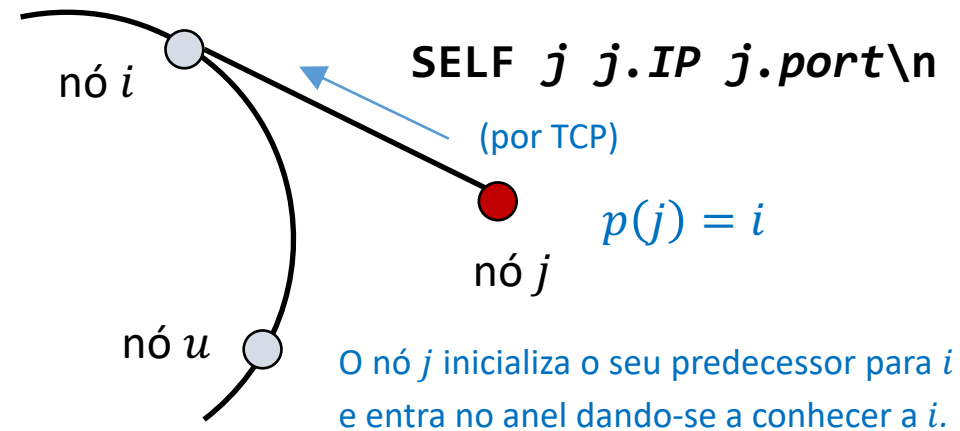
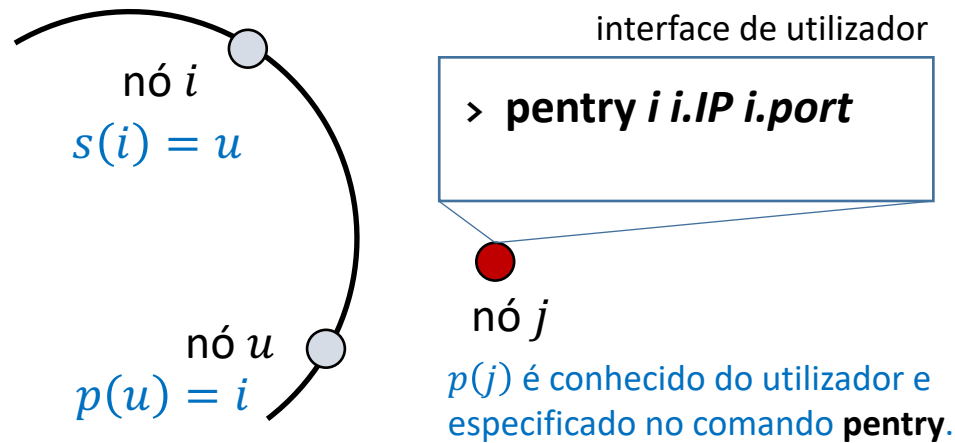
Conclusão: a resposta chegou ao seu destino.



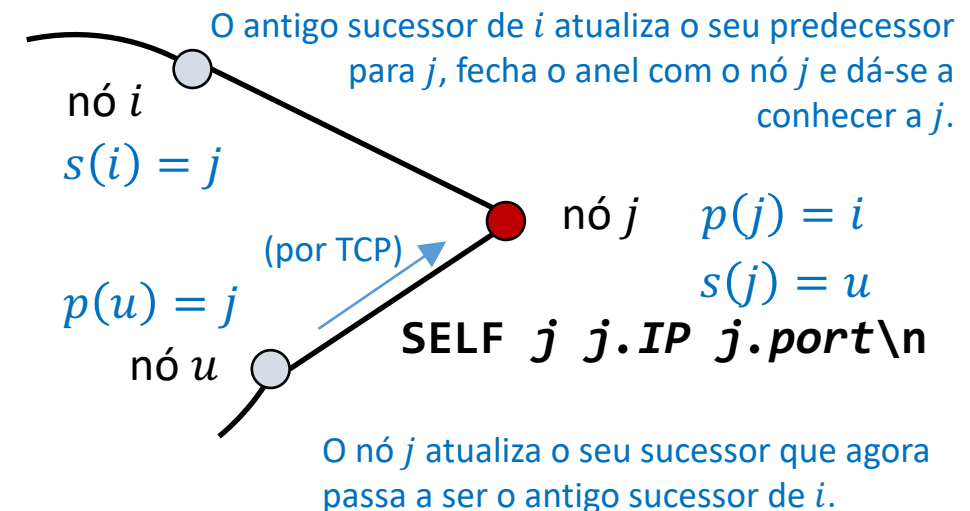
A resposta que terminou tem como identificador 88.  
A este identificador está associado o pedido de procura da chave 15, feito através da interface de utilizador.  
O resultado da procura é assim apresentado na interface de utilizador.

# Entrada do nó $j$ sabendo posição no anel

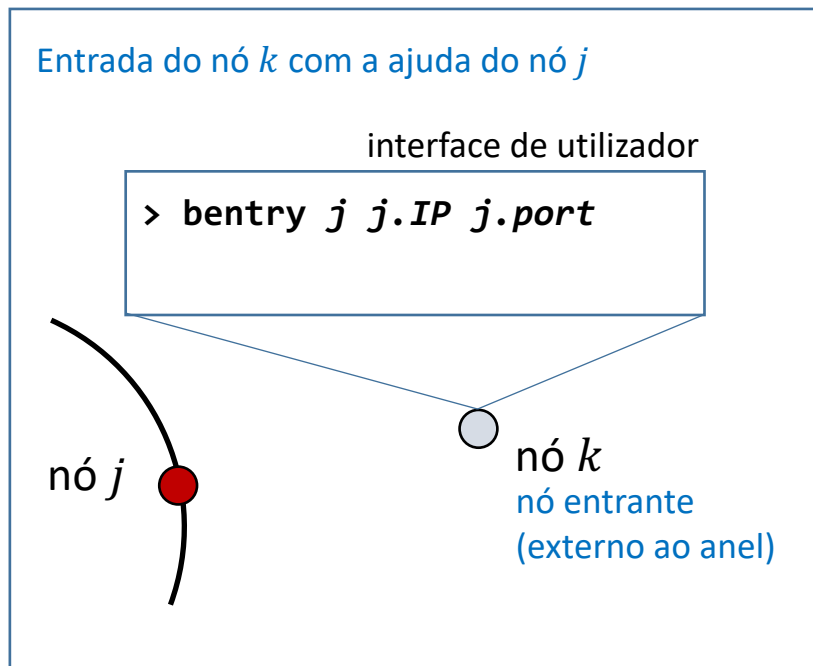
Entrada do nó  $j$  com o nó  $i$  como predecessor



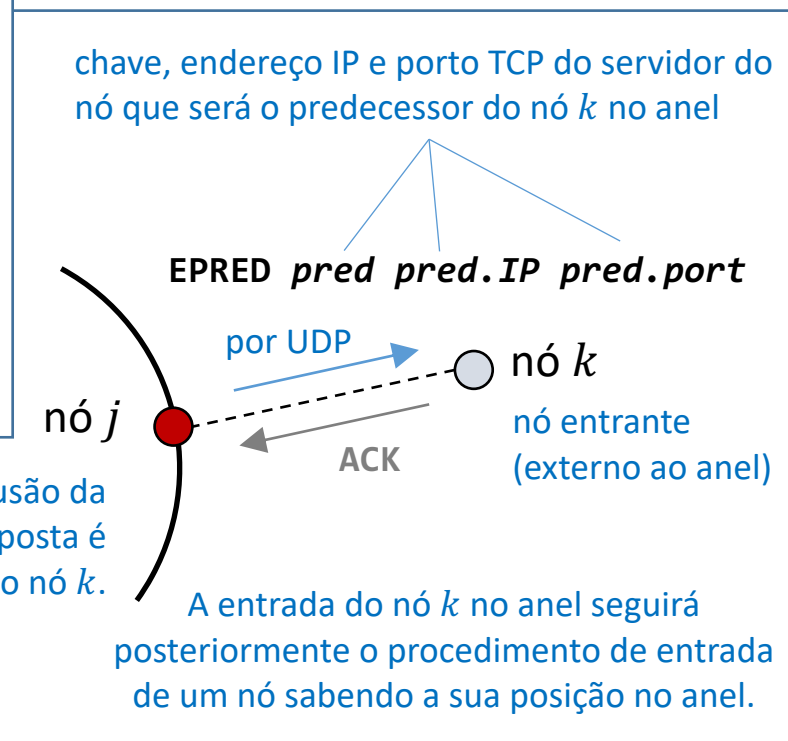
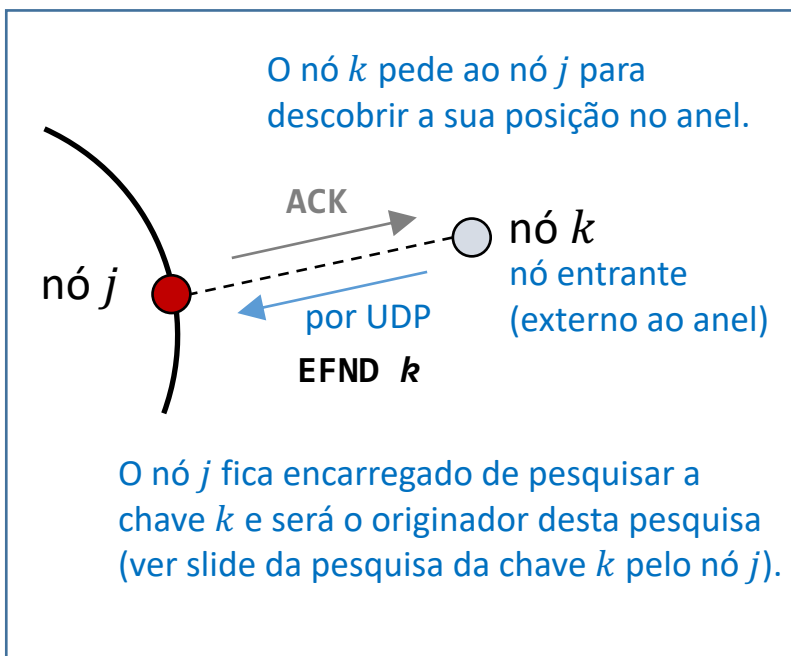
O nó  $i$  atualiza o seu sucessor para  $j$ , informa o seu antigo sucessor sobre o nó  $j$  e abre o anel.



# Entrada do nó $k$ não sabendo posição no anel



## Descoberta da posição do nó $k$ no anel



# Saída do nó $j$ do anel

