

Roteamento em DHT estática - INF1406

Instruções

Compilamos todos os programas utilizando os comandos:

```
>cargo build --bin client
>cargo build --bin server
```

Os testes são realizados com 2 terminais abertos. Primeiro disparamos os 16 nós do anel em background usando o comando:

```
>target/debug/server 0 16 & target/debug/server 1 16 & target/debug/server 2 16 & target/debug/server 3 16 & target/debug/server 4 16 &
target/debug/server 5 16 & target/debug/server 6 16 & target/debug/server 7 16 & target/debug/server 8 16 & target/debug/server 9 16 &
target/debug/server 10 16 & target/debug/server 11 16 &
target/debug/server 12 16 & target/debug/server 13 16 &
target/debug/server 14 16 & target/debug/server 15 16
```

(lembre-se de matar todos os processos em background gerados ao final dos testes usando `>killall -9 server``)

Depois, cada um dos terminais vai rodar uma instância do cliente:

```
>target/debug/client
```

Em cada um dos terminais, entre com a porta que aquele cliente vai usar para receber as respostas (nós usamos as portas 7877 e 7878) (os nós dos servidores ocupam as portas 7879..7894)

Feito isso, estamos prontos para realizar os testes. Eles consistem em digitar nos terminais dos clientes o número do nó para o qual a requisição vai ser feita, seguido da requisição em si, que vai estar em um dos formatos a seguir:

```
`C+<chave>+`          ----> Consulta da <chave>
`I+<chave>+<valor>+`----> Inserção do <valor> na <chave>
```

(Vale notar que nem a chave nem o valor pode utilizar o caracter '+')

Teste:

Passo 1

Consulta direta em chave não existente

Input no cliente 1:

```
>7
```

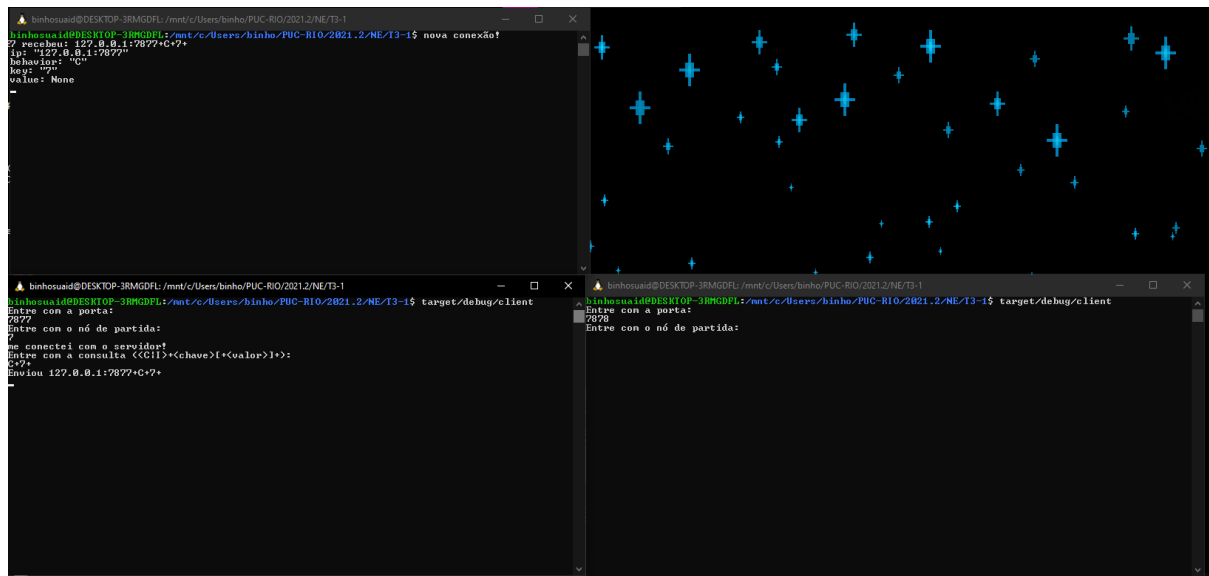
```
>C+7+
```

Resultado esperado:

O servidor recebe a conexão e a atende direto sem repassar para outro nó

O cliente 1 vai ficar bloqueado, até que a inserção seja feita pelo outro cliente.

Output:



```
binhosuaid@DESKTOP-3RMGDFL:/mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ ./nova_conexao.py
? recebeu: 127.0.0.1:7877+C+7+
ip: "127.0.0.1:7877"
hostname: "C"
key: "7"
value: None

binhosuaid@DESKTOP-3RMGDFL:/mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ ./target/debug/client
Entre com a porta:
7877
Entre com o nó de partida:
7
me conectei com o servidor!
Entre com a consulta <<CII>>+<chave>[+<valor>]+>:
C+7+
Enviou 127.0.0.1:7877+C+7+

binhosuaid@DESKTOP-3RMGDFL:/mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ ./target/debug/client
Entre com a porta:
7877
Entre com o nó de partida:
```

Passo 2

Inserção direta

Input no cliente 2:

>7

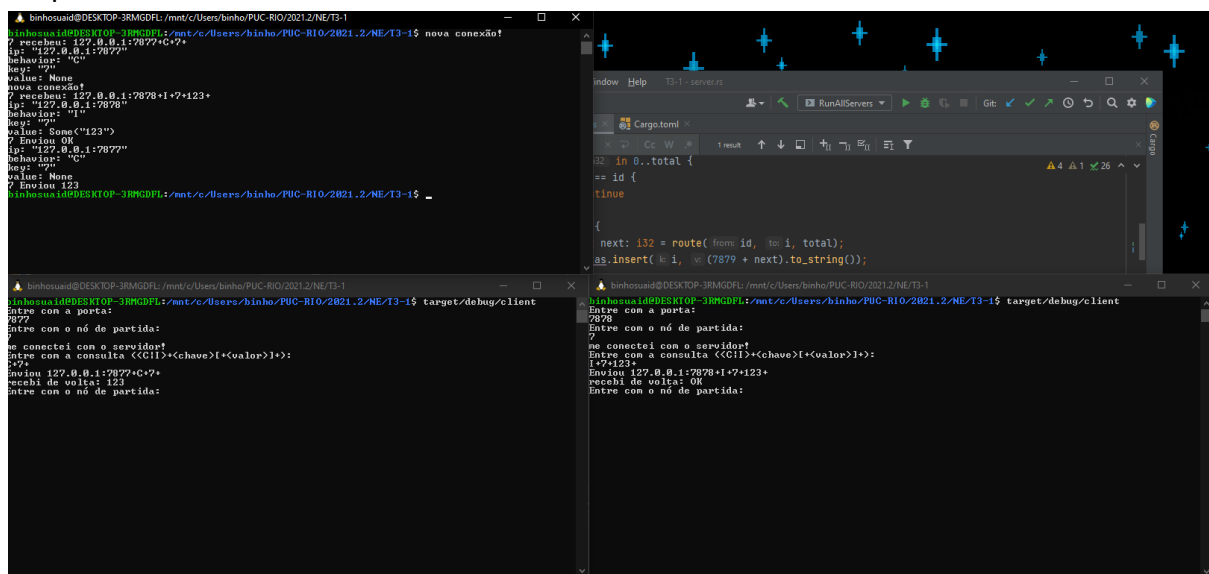
>I+7+123+

Resultado esperado:

O servidor recebe a conexão e a atende direto sem repassar para outro nó

O cliente 2 vai receber um OK, e o cliente 1 que estava esperando essa chave vai receber o '123'.

Output:



The screenshot shows a terminal window with network traffic and a code editor. The terminal output is as follows:

```
binhosuaid@DESKTOP-3RMGDPL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$  
7 recebeu 127.0.0.1:7877+C*?  
ip: "127.0.0.1:7877"  
behavior: "C"  
key: "7"  
value: None  
nova conexão!  
7 recebeu 127.0.0.1:7878+I+7+123+  
ip: "127.0.0.1:7878"  
behavior: "I"  
key: "7"  
value: Some("123")  
7 Envia OK  
ip: "127.0.0.1:7877"  
behavior: "C"  
key: "7"  
value: None  
7 Envia 123  
binhosuaid@DESKTOP-3RMGDPL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$
```

The code editor shows the following code:

```
in 0..total {  
  == id {  
    tinue  
  }  
  next: 132 = route( from id, to i, total);  
  as.insert( k: i, v: (7879 + next).to_string());  
}
```

Passo 3

Consulta direta em chave existente

Input no cliente 2:

>7

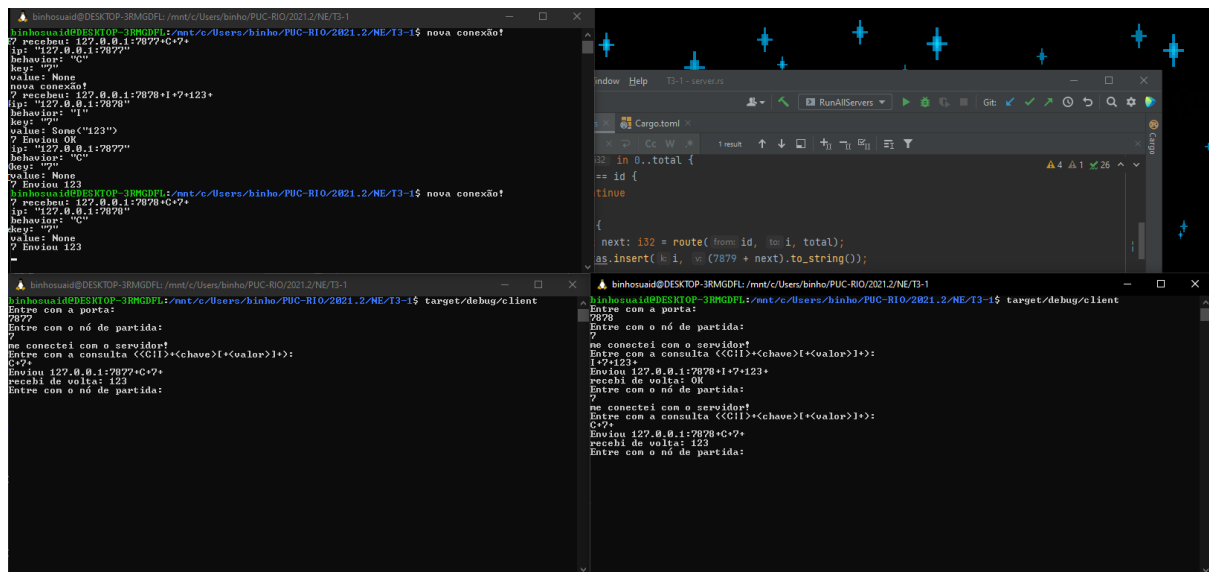
>C+7+

Resultado esperado:

0 servidor recebe a conexão e a atende direto sem repassar para outro nó

0 cliente 1 recebe '123' imediatamente

Output:



```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$  
7 recebeu: 127.0.0.1:7877+C+7+  
ip: "127.0.0.1:7877"  
behavior: "C"  
key: ""  
value: None  
nova conexão!  
7 recebeu: 127.0.0.1:7878+I+7+123+  
ip: "127.0.0.1:7878"  
behavior: "I"  
key: ""  
value: Some("123")  
7 Enviou OK  
ip: "127.0.0.1:7877"  
behavior: "C"  
key: ""  
value: None  
7 Enviou 123  
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$  
7 recebeu: 127.0.0.1:7878+C+7+  
ip: "127.0.0.1:7878"  
behavior: "C"  
key: ""  
value: None  
7 Enviou 123  
=
```

```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client  
Entre com a porta:  
7877  
Entre com o nó de partida:  
7  
Se conectei com o servidor!  
Entre com a consulta <<CII>>+<chave>[+<valor>]]+:  
C+7+  
Enviou 127.0.0.1:7877+C+7+  
recebi de volta: 123  
Entre com o nó de partida:
```

```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client  
Entre com a porta:  
7878  
Entre com o nó de partida:  
7  
Se conectei com o servidor!  
Entre com a consulta <<CII>>+<chave>[+<valor>]]+:  
I+7+123+  
Enviou 127.0.0.1:7878+I+7+123+  
recebi de volta: OK  
Entre com o nó de partida:  
7  
Se conectei com o servidor!  
Entre com a consulta <<CII>>+<chave>[+<valor>]]+:  
C+7+  
Enviou 127.0.0.1:7878+C+7+  
recebi de volta: 123  
Entre com o nó de partida:
```

```
index.html  
132 in 0..total {  
  == id {  
    tinue  
  }  
  next: 132 = route( from: id, to: 1, total);  
  as.insert( k: 1, v: (7879 + next).to_string());  
}
```

Passo 4

Consulta com 1 nó de distância em chave inexistente

Input no cliente 1:

>1

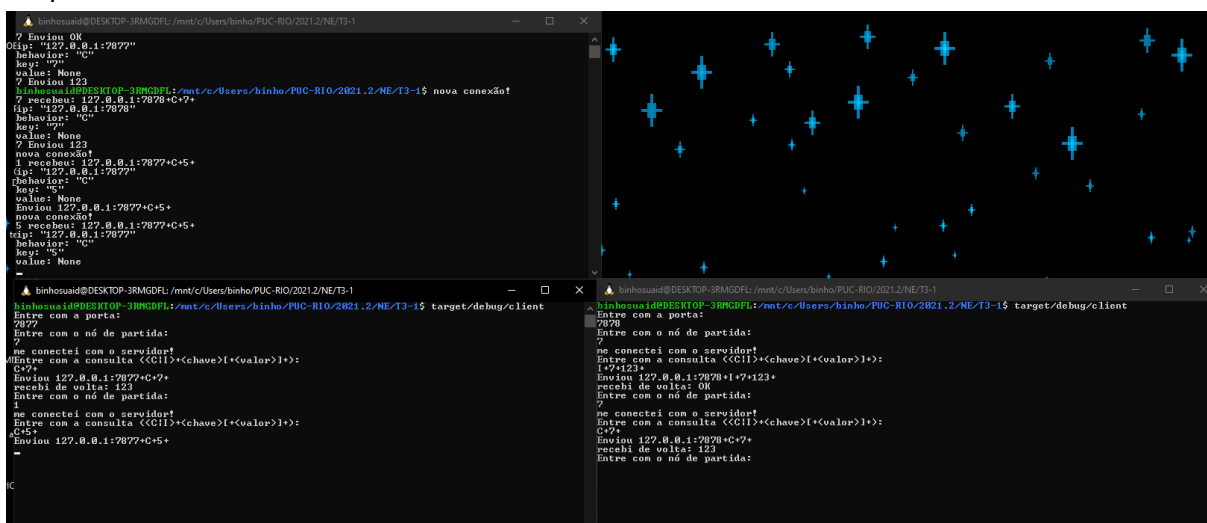
>C+5+

Resultado esperado:

O servidor recebe a conexão e repassa ela uma vez para o nó 5

O cliente 1 vai ficar bloqueado, até que a inserção seja feita pelo outro cliente.

Output:



```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1
7 Envia OK
8 ip: "127.0.0.1:7877"
behavior: "C"
key: "7"
value: None
7 Envia 123
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ nova conexão
7 recebe 127.0.0.1:7878+C+7+
ip: "127.0.0.1:7878"
behavior: "C"
key: "7"
value: None
7 Envia 123
nova conexão!
1 recebe 127.0.0.1:7877+C+5+
ip: "127.0.0.1:7877"
behavior: "C"
key: "5"
value: None
Envia 127.0.0.1:7877+C+5+
nova conexão!
5 recebe 127.0.0.1:7877+C+5+
ip: "127.0.0.1:7877"
behavior: "C"
key: "5"
value: None
=

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
Entre com a porta:
7877
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]*>:
C+7+
Envia 127.0.0.1:7877+C+7+
recebi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]*>:
C+5+
Envia 127.0.0.1:7877+C+5+
=

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
Entre com a porta:
7878
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]*>:
1+7+123+
Envia 127.0.0.1:7878+1+7+123+
recebi de volta: OK
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]*>:
C+7+
Envia 127.0.0.1:7878+C+7+
recebi de volta: 123
Entre com o nó de partida:
```

Passo 5

Inserção com 1 nó de distância

Input no cliente 2:

>1

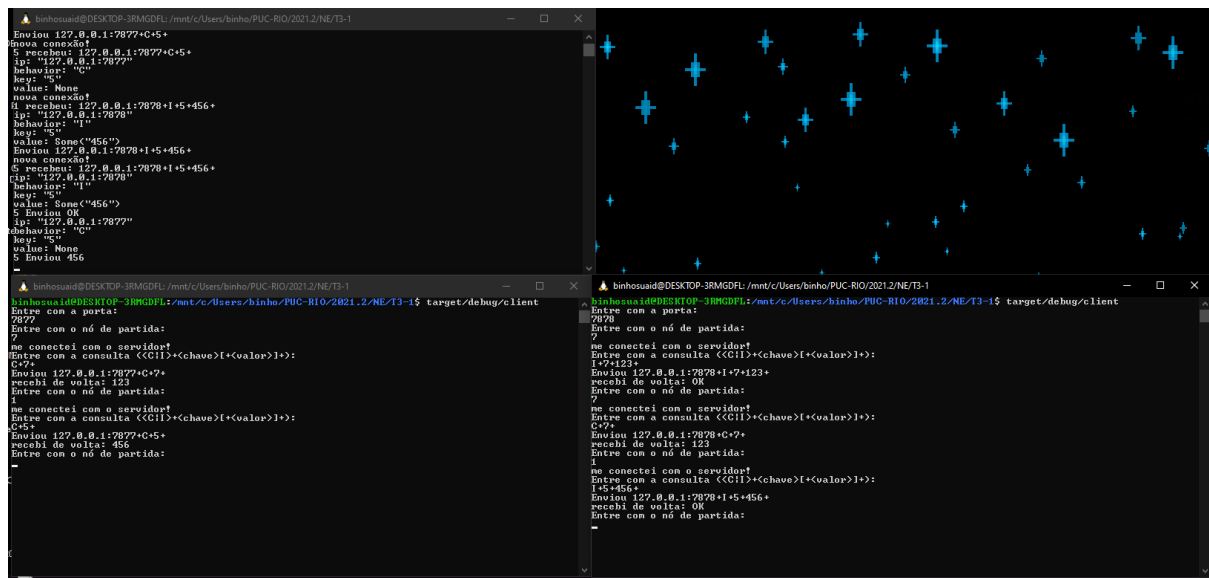
>I+5+456+

Resultado esperado:

O servidor recebe a conexão e repassa ela uma vez para o nó 5

O cliente 2 vai receber um OK, e o cliente 1 que estava esperando essa chave vai receber o '456'.

Output:



The image displays three terminal windows from a Kali Linux desktop environment. The top window shows a Wireshark capture of network traffic between 127.0.0.1 and 127.0.0.1. It details the transmission of a packet with key '5' and value 'None', followed by a packet with key '5' and value '456'. The bottom-left window shows the output of a 'target/debug/client' program, which logs the process of connecting to a server and receiving a response. The bottom-right window shows the output of a 'target/debug/client' program, which logs the process of connecting to a server and receiving a response.

```
binhousaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1
Envio 127.0.0.1:7877+C+5+
nova conexão!
5 recebeu: 127.0.0.1:7877+C+5+
ip: "127.0.0.1:7877"
behavior: "C"
key: "5"
value: None
nova conexão!
6 recebeu: 127.0.0.1:7878+I+5+456+
ip: "127.0.0.1:7878"
behavior: "I"
key: "5"
value: Some("456")
Envio 127.0.0.1:7878+I+5+456+
nova conexão!
6 recebeu: 127.0.0.1:7878+I+5+456+
ip: "127.0.0.1:7878"
behavior: "I"
key: "5"
value: Some("456")
5 Enviou OK
ip: "127.0.0.1:7877"
behavior: "C"
key: "5"
value: None
5 Enviou 456

binhousaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
Entre com a porta:
7877
Entre com o nó de partida:
7
me conectei com o servidor!
Entre com a consulta <<CII><<chave>I+<valor>I+>:
C+7+
Envio 127.0.0.1:7877+C+7+
recebi de volta: 123
Entre com o nó de partida:
1
me conectei com o servidor!
Entre com a consulta <<CII><<chave>I+<valor>I+>:
C+5+
Envio 127.0.0.1:7877+C+5+
recebi de volta: 456
Entre com o nó de partida:

binhousaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
Entre com a porta:
7878
Entre com o nó de partida:
7
me conectei com o servidor!
Entre com a consulta <<CII><<chave>I+<valor>I+>:
I+7+123+
Envio 127.0.0.1:7878+I+7+123+
recebi de volta: OK
Entre com o nó de partida:
1
me conectei com o servidor!
Entre com a consulta <<CII><<chave>I+<valor>I+>:
I+5+456+
Envio 127.0.0.1:7878+I+5+456+
recebi de volta: OK
Entre com o nó de partida:
```

Passo 6

Consulta com 1 nó de distância em chave existente

Input no cliente 2:

>1

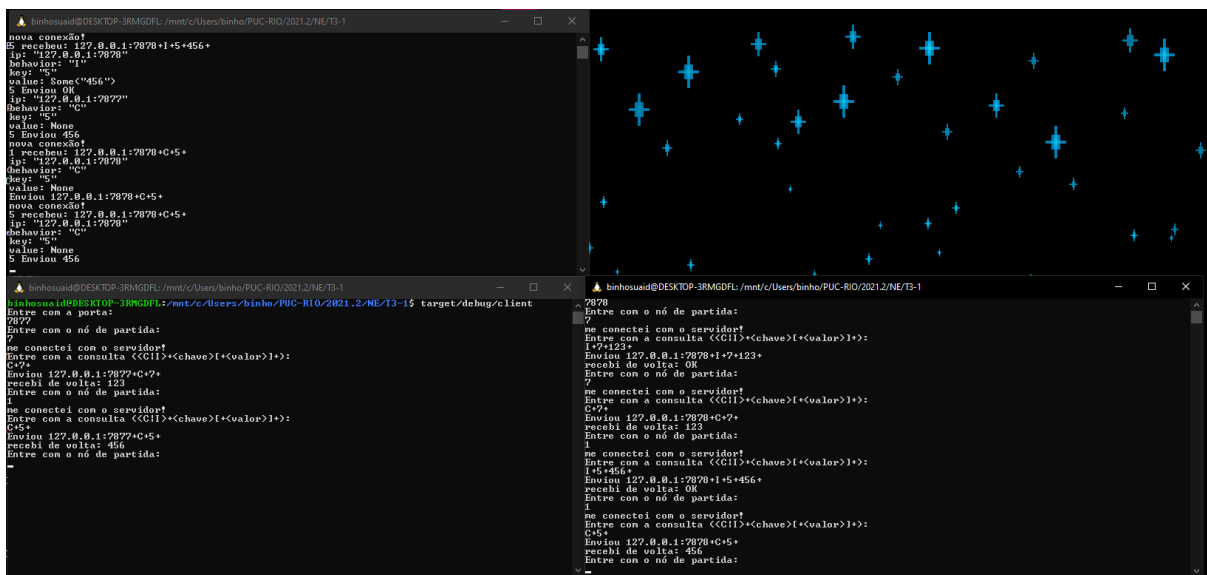
>C+5+

Resultado esperado:

O servidor recebe a conexão e repassa ela uma vez para o nó 5

O cliente 1 recebe '456' imediatamente

Output:



```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1
nova conexão!
S recebeu: 127.0.0.1:7878+1+5+456+
ip: "127.0.0.1:7878"
behavior: "1"
key: "5"
value: Some("456")
S Enviou OK
ip: "127.0.0.1:7877"
behavior: "C"
key: "C"
value: None
S Enviou 456
nova conexão!
I recebeu: 127.0.0.1:7878+C+5+
ip: "127.0.0.1:7878"
behavior: "C"
key: "C"
value: None
Enviou 127.0.0.1:7878+C+5+
nova conexão!
S recebeu: 127.0.0.1:7878+C+5+
ip: "127.0.0.1:7878"
behavior: "C"
key: "C"
value: None
S Enviou 456

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
7877
Entre com a porta:
7877
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<C1>>+<chave>[+<valor>]+:
C+7+
Enviou 127.0.0.1:7877+C+7+
recebi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<C1>>+<chave>[+<valor>]+:
C+5+
Enviou 127.0.0.1:7877+C+5+
recebi de volta: 456
Entre com o nó de partida:
7878
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<C1>>+<chave>[+<valor>]+:
1+7+123+
Enviou 127.0.0.1:7878+1+7+123+
recebi de volta: OK
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<C1>>+<chave>[+<valor>]+:
C+7+
Enviou 127.0.0.1:7878+C+7+
recebi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<C1>>+<chave>[+<valor>]+:
1+5+456+
Enviou 127.0.0.1:7878+1+5+456+
recebi de volta: OK
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<C1>>+<chave>[+<valor>]+:
C+5+
Enviou 127.0.0.1:7878+C+5+
recebi de volta: 456
Entre com o nó de partida:
```

Passo 7

Consulta com vários nós de distância em chave inexistente

Input cliente 1:

>2

>C+1+

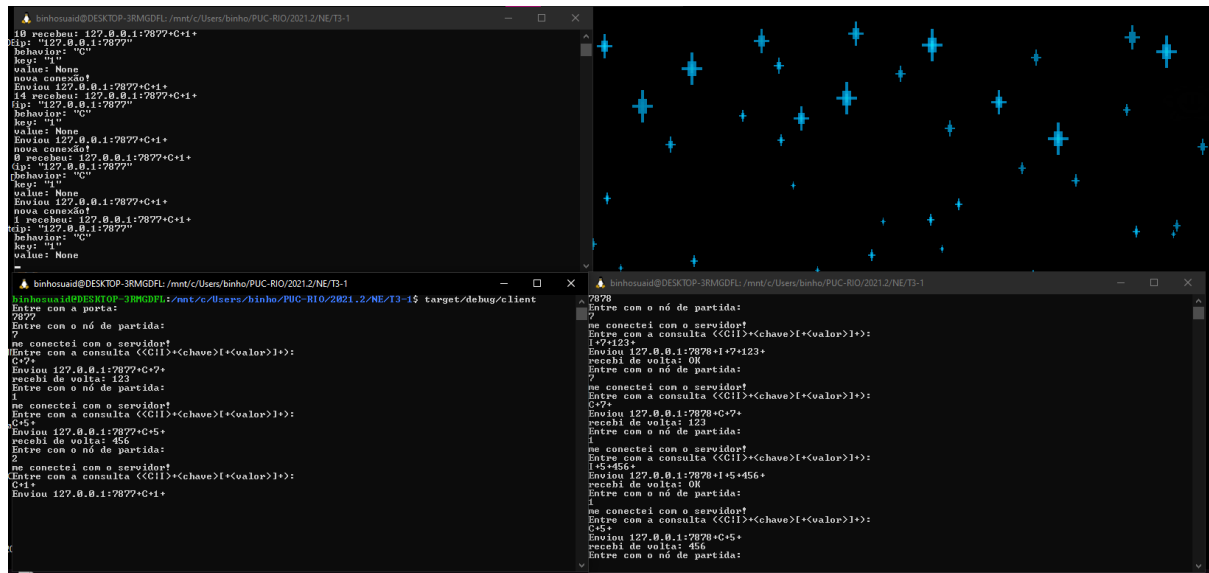
Resultado esperado:

O Servidor vai repassar a mensagem 4 vezes antes de atendê-la

(2->10->14->0->1)

O cliente 1 vai ficar bloqueado, até que a inserção seja feita pelo outro cliente.

Output:



```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1
10 recebeu: 127.0.0.1:7877+C+1+
key: "127.0.0.1:7877"
behavior: "C"
value: None
nova conexão!
Enviou 127.0.0.1:7877+C+1+
14 recebeu: 127.0.0.1:7877+C+1+
key: "127.0.0.1:7877"
behavior: "C"
value: None
Enviou 127.0.0.1:7877+C+1+
nova conexão!
0 recebeu: 127.0.0.1:7877+C+1+
key: "127.0.0.1:7877"
behavior: "C"
value: None
Enviou 127.0.0.1:7877+C+1+
nova conexão!
1 recebeu: 127.0.0.1:7877+C+1+
key: "127.0.0.1:7877"
value: None
=

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
7877
Entre com a porta:
7877
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
C+7+
Enviou 127.0.0.1:7877+C+7+
recebi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
C+5+
Enviou 127.0.0.1:7877+C+5+
recebi de volta: 456
Entre com o nó de partida:
2
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
C+1+
Enviou 127.0.0.1:7877+C+1+

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
7878
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
1+7+123+
Enviou 127.0.0.1:7878+1+7+123+
recebi de volta: OK
Entre com o nó de partida:
7
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
C+7+
Enviou 127.0.0.1:7878+C+7+
recebi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
1+5+456+
Enviou 127.0.0.1:7878+1+5+456+
recebi de volta: OK
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII><<chave>[+<valor>]]>:
C+5+
Enviou 127.0.0.1:7878+C+5+
recebi de volta: 456
Entre com o nó de partida:
```


Passo 8

Inserção com vários nós de distância

Input cliente 2:

>2

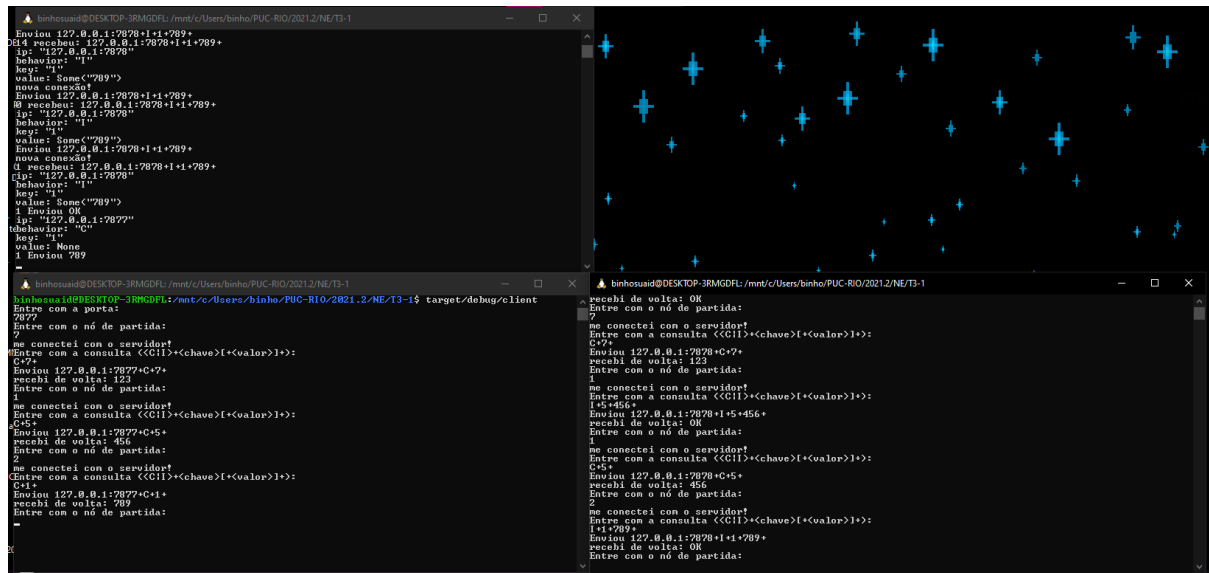
>I+1+789+

Resultado esperado:

O Servidor vai repassar a mensagem 4 vezes antes de atendê-la
(2->10->14->0->1)

O cliente 2 vai receber um OK, e o cliente 1 que estava esperando essa chave vai receber o '789'.

Output:



```
binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1
Enviou 127.0.0.1:7878+I+1+789+
DE14 recebeu: 127.0.0.1:7878+I+1+789+
ip: "127.0.0.1:7878"
behavior: "I"
key: "I"
value: Some("789")
nova conexão!
Enviou 127.0.0.1:7878+I+1+789+
DE14 recebeu: 127.0.0.1:7878+I+1+789+
ip: "127.0.0.1:7878"
behavior: "I"
key: "I"
value: Some("789")
Enviou 127.0.0.1:7878+I+1+789+
nova conexão!
DE14 recebeu: 127.0.0.1:7878+I+1+789+
ip: "127.0.0.1:7878"
behavior: "I"
key: "I"
value: Some("789")
1 Enviou OK
DE14 recebeu: "C"
key: "I"
value: None
1 Enviou 789
=

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
Entre com a porta:
7877
Entre com o nó de partida:
2
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+7+
Enviou 127.0.0.1:7877+C+7+
recbi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+5+
Enviou 127.0.0.1:7877+C+5+
recbi de volta: 456
Entre com o nó de partida:
2
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+1+
Enviou 127.0.0.1:7877+C+1+
recbi de volta: 789
Entre com o nó de partida:
=

binhosuaid@DESKTOP-3RMGDFL: /mnt/c/Users/binho/PUC-RIO/2021.2/NE/T3-1$ target/debug/client
recbi de volta: OK
Entre com o nó de partida:
2
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+7+
Enviou 127.0.0.1:7878+C+7+
recbi de volta: 123
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+5+456+
Enviou 127.0.0.1:7878+I+5+456+
recbi de volta: OK
Entre com o nó de partida:
1
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+5+
Enviou 127.0.0.1:7878+C+5+
recbi de volta: 456
Entre com o nó de partida:
2
ne conectei com o servidor!
Entre com a consulta <<CII>*<chave>[*<valor>]>:
C+1+789+
Enviou 127.0.0.1:7878+I+1+789+
recbi de volta: OK
Entre com o nó de partida:
```

Passo 9

Consulta com vários nós de distância em chave existente

Input cliente 2:

>2

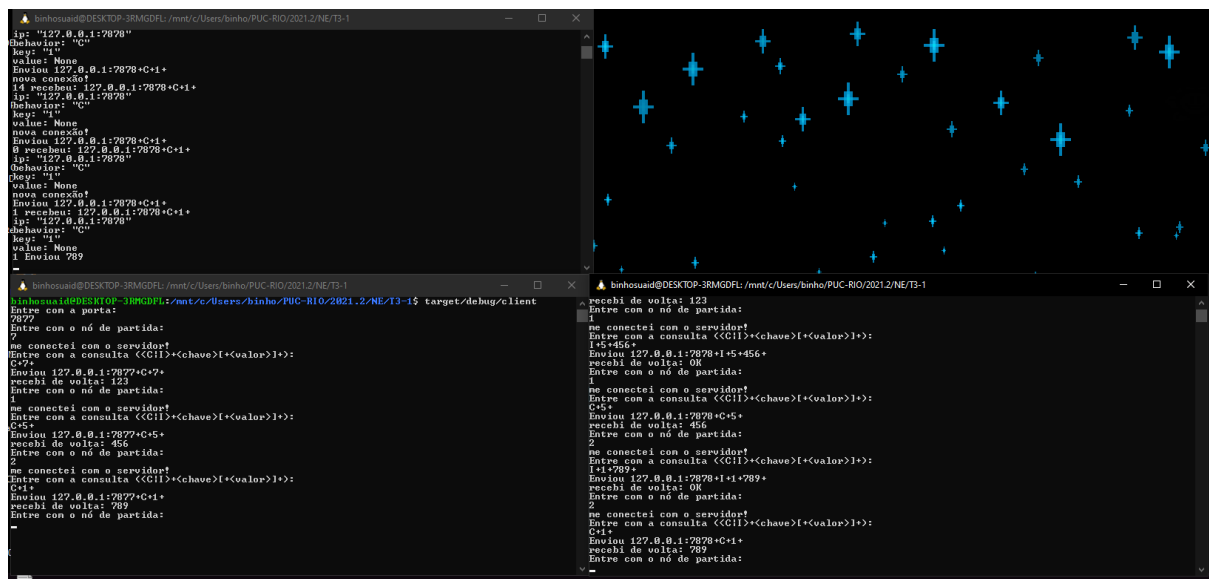
>C+1+

0 Servidor vai repassar a mensagem 4 vezes antes de atendê-la

(2->10->14->0->1)

0 cliente 1 recebe '789' imediatamente

Output:



The screenshot displays a network simulation environment. On the left, a terminal window shows the following log output:

```
ip: "127.0.0.1:7878"
Behavior: "C"
key: "i"
value: None
Enviou 127.0.0.1:7878+C+1+
nova conexão!
14 recebeu: 127.0.0.1:7878+C+1+
ip: "127.0.0.1:7878"
Behavior: "C"
key: "i"
value: None
nova conexão!
Enviou 127.0.0.1:7878+C+1+
0 recebeu: 127.0.0.1:7878+C+1+
ip: "127.0.0.1:7878"
Behavior: "C"
key: "i"
value: None
nova conexão!
Enviou 127.0.0.1:7878+C+1+
1 recebeu: 127.0.0.1:7878+C+1+
ip: "127.0.0.1:7878"
Behavior: "C"
key: "i"
value: None
1 Enviou 789
```

On the right, a network map shows several nodes (blue squares) connected by lines, representing the network topology. Below the terminal, a command prompt shows the execution of a target/debug/client command, which triggers the simulation.