



Escola de Artes, Ciências e Humanidades
da Universidade de São Paulo

ACH2147 - Desenvolvimento de Sistemas de Informação
Distribuídos

Luan Pereira Pinheiro - 13672471

Marcos Vilela Rezende Júnior - 13729806

São Paulo

2024

Parte 1

Qual o paradigma de programação escolhido?

Para a resolução desse Exercício-Programa, escolhemos utilizar a programação orientada a objetos (POO). A escolha por esse paradigma se dá pela facilidade de modulação apresentada, permitindo que cada componente do sistema (nós da rede, métodos de busca, mensagens, etc.) seja representado como uma classe com responsabilidades bem definidas, o que ajuda na organização e manutenção do código, permitindo que alterações em uma parte do sistema sejam feitas sem afetar outras partes.

Como foi feita a divisão do programa em threads?

A divisão em threads foi utilizada principalmente em “ConnectionManager” buscando permitir que múltiplas conexões de clientes sejam processadas simultaneamente, aumentando a eficiência e capacidade do servidor, além de permitir interação do usuário com o servidor via terminal ao mesmo tempo. Ademais, ao utilizar essa técnica, o servidor pode continuar aceitando novas conexões enquanto processa conexões existentes.

Foi escolhido utilizar operações bloqueantes ou não bloqueantes?

Escolhemos utilizar operações bloqueantes, devido a facilidade de serem implementadas e entendidas. Assim, não é necessário lidar com a complexidade adicional de verificar continuamente se há dados disponíveis ou se uma conexão foi estabelecida. Além disso, como estamos utilizando threads para lidar com cada conexão de cliente de forma independente, a natureza bloqueante das chamadas não impede o processamento paralelo. Cada thread pode bloquear esperando por dados sem afetar outras threads ou a capacidade do servidor de aceitar novas conexões. Adicionalmente, adicionamos bloqueio das estruturas de dados e variáveis que podem ser usadas por múltiplas threads, de modo que as threads não as possam acessar ao mesmo tempo, o que poderia gerar um comportamento indesejado e imprevisível ao programa, especialmente em casos de escrita simultânea e conflitos de leitura-escrita.

Parte 2

Demonstrar o funcionamento das operações HELLO e BYE:

```
Escolha o comando:
  [0] Listar vizinhos
  [1] HELLO
  [2] SEARCH (flooding)
  [3] SEARCH (random walk)
  [4] SEARCH (busca em profundidade)
  [5] Estatisticas
  [6] Alterar valor padrao de TTL
  [9] Sair
1
Escolha o vizinho:
Ha 1 vizinhos na tabela:
  [0] 127.0.0.1 5002
0
Encaminhando mensagem "127.0.0.1:5001 3 100 HELLO" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5001 3 100 HELLO"
Escolha o comando:
  [0] Listar vizinhos
  [1] HELLO
  [2] SEARCH (flooding)
  [3] SEARCH (random walk)
  [4] SEARCH (busca em profundidade)
  [5] Estatisticas
  [6] Alterar valor padrao de TTL
  [9] Sair
```

Comando HELLO do lado do remetente

```
Servidor criado: 127.0.0.1:5002
Escolha o comando:
  [0] Listar vizinhos
  [1] HELLO
  [2] SEARCH (flooding)
  [3] SEARCH (random walk)
  [4] SEARCH (busca em profundidade)
  [5] Estatisticas
  [6] Alterar valor padrao de TTL
  [9] Sair
Mensagem recebida: "127.0.0.1:5001 1 100 HELLO"
Adicionando vizinho na tabela: 127.0.0.1:5001
Mensagem recebida: "127.0.0.1:5001 3 100 HELLO"
Vizinho ja esta na tabela: 127.0.0.1:5001
```

Comando HELLO do lado do receptor (a primeira mensagem ocorreu no adiç o de vizinhos a partir de arquivo)

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
9
Voce escolheu 9
Saindo...
Encaminhando mensagem "127.0.0.1:5001 4 100 BYE" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5001 4 100 BYE"

```

Comando BYE do lado do remetente

```

Mensagem recebida: "127.0.0.1:5001 4 100 BYE"
Removendo vizinho da tabela 127.0.0.1:5001

```

Comando BYE do lado do receptor

Demonstrar que a lógica do TTL está sendo aplicada:

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5002 2 100 HELLO"
Adicionando vizinho na tabela: 127.0.0.1:5002
6
Digite novo valor de TTL
1
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
2
Digite a chave a ser buscada
oi
Encaminhando mensagem "127.0.0.1:5003 2 1 SEARCH FL 5003 oi 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 2 1 SEARCH FL 5003 oi 1"

```

Mensagem de busca sendo enviada com o valor de TTL 1

```
Mensagem recebida: "127.0.0.1:5003 3 1 SEARCH FL 5003 oi 1"
TTL igual a zero, descartando mensagem
```

Mensagem sendo descartada pelo nó que a recebeu, devido ao TTL ser zero

Demonstrar que a busca por flooding é capaz de encontrar uma chave que existe na rede:

```
2
Digite a chave a ser buscada
oi
Encaminhando mensagem "127.0.0.1:5003 2 100 SEARCH FL 5003 oi 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 2 100 SEARCH FL 5003 oi 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 2 100 VAL FL oi tchau 1"
Valor encontrado! Chave: oi valor: tchau
```

Nó que enviou a mensagem recebendo o valor encontrado

```
Mensagem recebida: "127.0.0.1:5003 2 100 SEARCH FL 5003 oi 1"
Encaminhando mensagem "127.0.0.1:5003 2 99 SEARCH FL 5002 oi 2" para 127.0.0.1:5001
Envio feito com sucesso: "127.0.0.1:5003 2 99 SEARCH FL 5002 oi 2"
```

Nó intermediário repassando a mensagem de busca

```
Mensagem recebida: "127.0.0.1:5003 2 99 SEARCH FL 5002 oi 2"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5001 2 100 VAL FL oi tchau 2" para 127.0.0.1:5003
Envio feito com sucesso: "127.0.0.1:5001 2 100 VAL FL oi tchau 2"
```

Nó que possui a chave retornando o valor para a origem

Demonstrar que as mensagens de busca por flooding não são re-enviadas para o nó remetente:

```
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 1 100 HELLO"
Adicionando vizinho na tabela: 127.0.0.1:5001
Mensagem recebida: "127.0.0.1:5003 2 100 SEARCH FL 5003 oi 1"
Encaminhando mensagem "127.0.0.1:5003 2 99 SEARCH FL 5002 oi 2" para 127.0.0.1:5001
Envio feito com sucesso: "127.0.0.1:5003 2 99 SEARCH FL 5002 oi 2"
0
Ha 2 vizinhos na tabela:
[0] 127.0.0.1 5003
[1] 127.0.0.1 5001
```

Embora a tabela possua dois vizinhos, a mensagem só foi reenviada para o vizinho 1, já que foi recebida pelo vizinho 0

Demonstrar que as mensagens de busca por flooding repetidas são descartadas:

```
Mensagem recebida: "127.0.0.1:5003 3 100 SEARCH FL 5003 oi 1"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5001 3 100 VAL FL oi tchau 1" para 127.0.0.1:5003
Envio feito com sucesso: "127.0.0.1:5001 3 100 VAL FL oi tchau 1"
Mensagem recebida: "127.0.0.1:5003 3 99 SEARCH FL 5002 oi 2"
Flooding: Mensagem repetida!
```

Nó recebeu a mesma mensagem de vizinhos diferentes e descartou a segunda mensagem

Demonstrar que a busca por random walk é capaz de encontrar uma chave existente tanto numa rede sem ciclos quanto numa rede com ciclos:

```

3
Digite a chave a ser buscada
oi
Encaminhando mensagem "127.0.0.1:5002 3 100 SEARCH RW 5002 oi 1" para 127.0.0.1:5003
Envio feito com sucesso: "127.0.0.1:5002 3 100 SEARCH RW 5002 oi 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5003 4 100 VAL RW oi tchau 1"
Valor encontrado! Chave: oi valor: tchau
0
Ha 2 vizinhos na tabela:
[0] 127.0.0.1 5001
[1] 127.0.0.1 5003

```

Valor sendo encontrado em uma rede com ciclo de 3 nós

```

Mensagem recebida: "127.0.0.1:5002 3 100 SEARCH RW 5002 oi 1"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5003 4 100 VAL RW oi tchau 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 4 100 VAL RW oi tchau 1"
0
Ha 2 vizinhos na tabela:
[0] 127.0.0.1 5001
[1] 127.0.0.1 5002

```

O vizinho que recebeu a mensagem e retornou o valor encontrado

```

0
Ha 2 vizinhos na tabela:
[0] 127.0.0.1 5002
[1] 127.0.0.1 5003

```

Os vizinhos do último dos três nós, provando que a rede tem ciclos

```

3
Digite a chave a ser buscada
oi
Encaminhando mensagem "127.0.0.1:5003 2 100 SEARCH RW 5003 oi 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 2 100 SEARCH RW 5003 oi 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 2 100 VAL RW oi tchau 2"
Valor encontrado! Chave: oi valor: tchau
0
Ha 1 vizinhos na tabela:
[0] 127.0.0.1 5002

```

Valor sendo encontrado em uma rede sem ciclos

```

Mensagem recebida: "127.0.0.1:5003 2 100 SEARCH RW 5003 oi 1"
Encaminhando mensagem "127.0.0.1:5003 2 99 SEARCH RW 5002 oi 2" para 127.0.0.1:5001
Envio feito com sucesso: "127.0.0.1:5003 2 99 SEARCH RW 5002 oi 2"
0
Ha 2 vizinhos na tabela:
[0] 127.0.0.1 5003
[1] 127.0.0.1 5001

```

O vizinho que realizou o intermédio entre o remetente e o detentor da chave

```

Mensagem recebida: "127.0.0.1:5003 2 99 SEARCH RW 5002 oi 2"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5001 2 100 VAL RW oi tchau 2" para 127.0.0.1:5003
Envio feito com sucesso: "127.0.0.1:5001 2 100 VAL RW oi tchau 2"
0
Ha 1 vizinhos na tabela:
[0] 127.0.0.1 5002

```

O vizinho que retornou o valor encontrado. Pelas tabelas de vizinhos, nota-se que a rede não tem ciclos

Demonstrar que a busca em profundidade é capaz de encontrar uma chave existente tanto numa rede sem ciclos quanto numa rede com ciclos:

```

0
Ha 2 vizinhos na tabela:
[0] 127.0.0.1 5003
[1] 127.0.0.1 5001
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
4
Digite a chave a ser buscada
oi
Encaminhando mensagem "127.0.0.1:5002 3 100 SEARCH BP 5002 oi 1" para 127.0.0.1:5003
Envio feito com sucesso: "127.0.0.1:5002 3 100 SEARCH BP 5002 oi 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5003 3 100 VAL BP oi tchau 1"
Valor encontrado! Chave: oi valor: tchau

```

Valor sendo encontrado em uma rede sem ciclos


```

0
Ha 1 vizinhos na tabela:
  [0] 127.0.0.1 5002
Escolha o comando:
  [0] Listar vizinhos
  [1] HELLO
  [2] SEARCH (flooding)
  [3] SEARCH (random walk)
  [4] SEARCH (busca em profundidade)
  [5] Estatisticas
  [6] Alterar valor padrao de TTL
  [9] Sair
Mensagem recebida: "127.0.0.1:5002 3 100 SEARCH BP 5002 oi 1"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5003 3 100 VAL BP oi tchau 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 3 100 VAL BP oi tchau 1"

```

O vizinho que retornou o valor encontrado

```

0
Ha 1 vizinhos na tabela:
  [0] 127.0.0.1 5002

```

A tabela de vizinho do último dos três nós, provando que a rede não tem ciclos

```

Escolha o comando:
  [0] Listar vizinhos
  [1] HELLO
  [2] SEARCH (flooding)
  [3] SEARCH (random walk)
  [4] SEARCH (busca em profundidade)
  [5] Estatisticas
  [6] Alterar valor padrao de TTL
  [9] Sair
4
Digite a chave a ser buscada
oi
Encaminhando mensagem "127.0.0.1:5003 3 100 SEARCH BP 5003 oi 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 3 100 SEARCH BP 5003 oi 1"
Escolha o comando:
  [0] Listar vizinhos
  [1] HELLO
  [2] SEARCH (flooding)
  [3] SEARCH (random walk)
  [4] SEARCH (busca em profundidade)
  [5] Estatisticas
  [6] Alterar valor padrao de TTL
  [9] Sair
Mensagem recebida: "127.0.0.1:5001 3 100 VAL BP oi tchau 2"
Valor encontrado! Chave: oi valor: tchau
0
Ha 2 vizinhos na tabela:
  [0] 127.0.0.1 5001
  [1] 127.0.0.1 5002

```

Valor sendo encontrado em uma rede com ciclo

```

Mensagem recebida: "127.0.0.1:5003 3 100 SEARCH BP 5003 oi 1"
Encaminhando mensagem "127.0.0.1:5003 3 99 SEARCH BP 5002 oi 2" para 127.0.0.1:5001
Envio feito com sucesso: "127.0.0.1:5003 3 99 SEARCH BP 5002 oi 2"
0
Ha 2 vizinhos na tabela:
  [0] 127.0.0.1 5001
  [1] 127.0.0.1 5003

```

Vizinho que realizou intermédio entre o remetente e o detentor da chave

```

Mensagem recebida: "127.0.0.1:5003 3 99 SEARCH BP 5002 oi 2"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5001 3 100 VAL BP oi tchau 2" para 127.0.0.1:5003
Envio feito com sucesso: "127.0.0.1:5001 3 100 VAL BP oi tchau 2"
0
Ha 2 vizinhos na tabela:
  [0] 127.0.0.1 5002
  [1] 127.0.0.1 5003

```

O vizinho que retornou o valor encontrado. Pelas tabelas de vizinhos, nota-se que a rede possui ciclo

Demonstrar a coleta correta de estatísticas:

```
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 2 100 HELLO"
    Adicionando vizinho na tabela: 127.0.0.1:5001
2
Digite a chave a ser buscada
a
Encaminhando mensagem "127.0.0.1:5003 3 100 SEARCH FL 5003 a 1" para 127.0.0.1:5002
    Envio feito com sucesso: "127.0.0.1:5003 3 100 SEARCH FL 5003 a 1"
Encaminhando mensagem "127.0.0.1:5003 3 100 SEARCH FL 5003 a 1" para 127.0.0.1:5001
    Envio feito com sucesso: "127.0.0.1:5003 3 100 SEARCH FL 5003 a 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 3 100 VAL FL a above 1"
    Valor encontrado! Chave: a valor: above
5
Estatísticas
    Total de mensagens de flooding vistas: 0
    Total de mensagens de random walk vistas: 0
    Total de mensagens de busca em profundidade vistas: 0
    Media de saltos ate encontrar destino por flooding: 1
    Media de saltos ate encontrar destino por random walk: -nan
    Media de saltos ate encontrar destino por busca em profundidade: -nan
```

Estatística correta após uma busca por flooding

```
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair

2
Digite a chave a ser buscada
b
Encaminhando mensagem "127.0.0.1:5003 4 100 SEARCH FL 5003 b 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 4 100 SEARCH FL 5003 b 1"
Encaminhando mensagem "127.0.0.1:5003 4 100 SEARCH FL 5003 b 1" para 127.0.0.1:5001
Envio feito com sucesso: "127.0.0.1:5003 4 100 SEARCH FL 5003 b 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair

Mensagem recebida: "127.0.0.1:5001 4 100 VAL FL b backticks 2"
Valor encontrado! Chave: b valor: backticks

5
Estatísticas
Total de mensagens de flooding vistas: 0
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: 1.5
Media de saltos ate encontrar destino por random walk: -nan
Media de saltos ate encontrar destino por busca em profundidade: -nan
```

Estatística correta após segunda busca por flooding

```
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
3
Digite a chave a ser buscada
c
Encaminhando mensagem "127.0.0.1:5003 5 100 SEARCH RW 5003 c 1" para 127.0.0.1:5001
Envio feito com sucesso: "127.0.0.1:5003 5 100 SEARCH RW 5003 c 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 5 100 VAL RW c C 1"
Valor encontrado! Chave: c valor: C
5
Estatísticas
Total de mensagens de flooding vistas: 0
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: 1.5
Media de saltos ate encontrar destino por random walk: 1
Media de saltos ate encontrar destino por busca em profundidade: -nan
```

Estatísticas corretas após busca por random walk

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
4
Digite a chave a ser buscada
d
Encaminhando mensagem "127.0.0.1:5003 6 100 SEARCH BP 5003 d 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 6 100 SEARCH BP 5003 d 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5001 6 100 VAL BP d da 2"
Valor encontrado! Chave: d valor: da
5
Estatisticas
Total de mensagens de flooding vistas: 0
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: 1.5
Media de saltos ate encontrar destino por random walk: 1
Media de saltos ate encontrar destino por busca em profundidade: 2

```

Estatísticas corretas após busca em profundidade

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5002 3 100 SEARCH FL 5002 a 1"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5003 7 100 VAL FL a above 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 7 100 VAL FL a above 1"
5
Estatisticas
Total de mensagens de flooding vistas: 1
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: 1.5
Media de saltos ate encontrar destino por random walk: 1
Media de saltos ate encontrar destino por busca em profundidade: 2

```

Estatísticas corretas após ver uma mensagem de flooding

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5002 5 100 SEARCH RW 5002 c 1"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5003 8 100 VAL RW c C 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 8 100 VAL RW c C 1"
5
Estatisticas
Total de mensagens de flooding vistas: 1
Total de mensagens de random walk vistas: 1
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: 1.5
Media de saltos ate encontrar destino por random walk: 1
Media de saltos ate encontrar destino por busca em profundidade: 2

```

Estatísticas corretas após ver uma mensagem de random walk

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "127.0.0.1:5002 6 100 SEARCH BP 5002 d 1"
Chave encontrada!
Encaminhando mensagem "127.0.0.1:5003 9 100 VAL BP d da 1" para 127.0.0.1:5002
Envio feito com sucesso: "127.0.0.1:5003 9 100 VAL BP d da 1"
5
Estatisticas
Total de mensagens de flooding vistas: 1
Total de mensagens de random walk vistas: 1
Total de mensagens de busca em profundidade vistas: 1
Media de saltos ate encontrar destino por flooding: 1.5
Media de saltos ate encontrar destino por random walk: 1
Media de saltos ate encontrar destino por busca em profundidade: 2

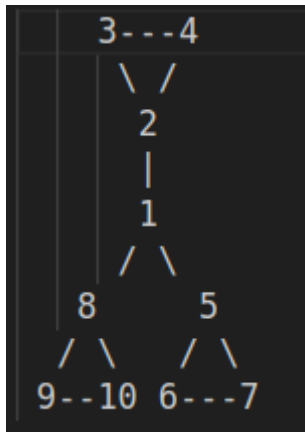
```

Estatísticas corretas após ver uma mensagem de busca em profundidade

Parte 3

Executar o programa em pelo menos dois computadores e em uma topologia um pouco mais complexa (à escolha do grupo, com pelo menos 10 nós). Em seguida, escolher dois nós da rede para fazer as operações de busca algumas vezes (cada nó deve buscar sempre a mesma chave). Analise as estatísticas obtidas.

Foi utilizada a topologia três triângulos:



Os nós de 1 a 5 estavam no primeiro computador e os outros 5 no outro computador.

```
0
Ha 3 vizinhos na tabela:
    [0] 192.168.31.109 5005
    [1] 192.168.31.106 5008
    [2] 192.168.31.109 5002
```

Lista de vizinhos do nó 1

```
0
Ha 3 vizinhos na tabela:
    [0] 192.168.31.109 5001
    [1] 192.168.31.109 5004
    [2] 192.168.31.109 5003
```

Lista de vizinhos do nó 2

```
0
Ha 2 vizinhos na tabela:
    [0] 192.168.31.109 5002
    [1] 192.168.31.109 5004
```

Lista de vizinhos do nó 3


```
0
Ha 2 vizinhos na tabela:
  [0] 192.168.31.109 5002
  [1] 192.168.31.109 5003
```

Lista de vizinhos do nó 4

```
0
Ha 3 vizinhos na tabela:
  [0] 192.168.31.106 5007
  [1] 192.168.31.106 5006
  [2] 192.168.31.109 5001
```

Lista de vizinhos do nó 5

```
0
Ha 2 vizinhos na tabela:
  [0] 192.168.31.109 5005
  [1] 192.168.31.106 5007
```

Lista de vizinhos do nó 6

```
0
Ha 2 vizinhos na tabela:
  [0] 192.168.31.109 5005
  [1] 192.168.31.106 5006
```

Lista de vizinhos do nó 7

```
0
Ha 3 vizinhos na tabela:
  [0] 192.168.31.106 5009
  [1] 192.168.31.106 5010
  [2] 192.168.31.109 5001
```

Lista de vizinhos do nó 8

```
0
Ha 2 vizinhos na tabela:
  [0] 192.168.31.106 5008
  [1] 192.168.31.106 5010
```

Lista de vizinhos do nó 9

```
0
Ha 2 vizinhos na tabela:
  [0] 192.168.31.106 5008
  [1] 192.168.31.106 5009
```

Lista de vizinhos do nó 10

Foram realizadas três operações de busca (uma de cada tipo) a partir dos nós 9 e 10. As buscas do nó 10 foram por uma chave que não há na rede, para que pudéssemos ver as mensagens “passeando” por toda a rede, e as buscas do nó 9 foram por uma chave que há na tabela do nó 3.

```
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "192.168.31.106:5009 2 100 HELLO"
Adicionando vizinho na tabela: 192.168.31.106:5009
Mensagem recebida: "192.168.31.106:5008 3 100 HELLO"
Adicionando vizinho na tabela: 192.168.31.106:5008
4
Digite a chave a ser buscada
chave
Encaminhando mensagem "192.168.31.106:5010 3 100 SEARCH BP 5010 chave 1" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5010 3 100 SEARCH BP 5010 chave 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "192.168.31.106:5010 3 80 SEARCH BP 5009 chave 21"
BP: Ciclo detectado, devolvendo a mensagem...
Encaminhando mensagem "192.168.31.106:5010 3 79 SEARCH BP 5010 chave 22" para 192.168.31.106:5009
Envio feito com sucesso: "192.168.31.106:5010 3 79 SEARCH BP 5010 chave 22"
Mensagem recebida: "192.168.31.106:5010 3 77 SEARCH BP 5008 chave 24"
BP: Nao foi possivel localizar a chave chave
```

Busca em profundidade do nó 10

```

Envio feito com sucesso: "192.168.31.106:5010 4 100 SEARCH RW 5010 chave 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatisticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "192.168.31.106:5010 4 91 SEARCH RW 5009 chave 10"
Encaminhando mensagem "192.168.31.106:5010 4 90 SEARCH RW 5010 chave 11" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5010 4 90 SEARCH RW 5010 chave 11"
Mensagem recebida: "192.168.31.106:5010 4 78 SEARCH RW 5009 chave 23"
Encaminhando mensagem "192.168.31.106:5010 4 77 SEARCH RW 5010 chave 24" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5010 4 77 SEARCH RW 5010 chave 24"
Mensagem recebida: "192.168.31.106:5010 4 69 SEARCH RW 5008 chave 32"
Encaminhando mensagem "192.168.31.106:5010 4 68 SEARCH RW 5010 chave 33" para 192.168.31.106:5009
Envio feito com sucesso: "192.168.31.106:5010 4 68 SEARCH RW 5010 chave 33"
Mensagem recebida: "192.168.31.106:5010 4 66 SEARCH RW 5008 chave 35"
Encaminhando mensagem "192.168.31.106:5010 4 65 SEARCH RW 5010 chave 36" para 192.168.31.106:5009
Envio feito com sucesso: "192.168.31.106:5010 4 65 SEARCH RW 5010 chave 36"
Mensagem recebida: "192.168.31.106:5010 4 63 SEARCH RW 5008 chave 38"
Encaminhando mensagem "192.168.31.106:5010 4 62 SEARCH RW 5010 chave 39" para 192.168.31.106:5009
Envio feito com sucesso: "192.168.31.106:5010 4 62 SEARCH RW 5010 chave 39"
Mensagem recebida: "192.168.31.106:5010 4 60 SEARCH RW 5008 chave 41"
Encaminhando mensagem "192.168.31.106:5010 4 59 SEARCH RW 5010 chave 42" para 192.168.31.106:5009
Envio feito com sucesso: "192.168.31.106:5010 4 59 SEARCH RW 5010 chave 42"
Mensagem recebida: "192.168.31.106:5010 4 32 SEARCH RW 5009 chave 69"
Encaminhando mensagem "192.168.31.106:5010 4 31 SEARCH RW 5010 chave 70" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5010 4 31 SEARCH RW 5010 chave 70"
Mensagem recebida: "192.168.31.106:5010 4 17 SEARCH RW 5009 chave 84"
Encaminhando mensagem "192.168.31.106:5010 4 16 SEARCH RW 5010 chave 85" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5010 4 16 SEARCH RW 5010 chave 85"
Mensagem recebida: "192.168.31.106:5010 4 14 SEARCH RW 5009 chave 87"
Encaminhando mensagem "192.168.31.106:5010 4 13 SEARCH RW 5010 chave 88" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5010 4 13 SEARCH RW 5010 chave 88"

```

Busca em random walking do nó 10

```

2
Digite a chave a ser buscada
chave
Encaminhando mensagem "192.168.31.106:5010 5 100 SEARCH FL 5010 chave 1" para 192.168.31.106:5009
Envio feito com sucesso: "192.168.31.106:5010 5 100 SEARCH FL 5010 chave 1"
Encaminhando mensagem "192.168.31.106:5010 5 100 SEARCH FL 5010 chave 1" para 192.168.31.106:5008
Mensagem recebida: "192.168.31.106:5010 5 98 SEARCH FL 5008 chave 3"
Flooding: Mensagem repetida!
Envio feito com sucesso: "192.168.31.106:5010 5 100 SEARCH FL 5010 chave 1"

```

Busca em flooding do nó 10

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
5
Estatísticas
Total de mensagens de flooding vistas: 1
Total de mensagens de random walk vistas: 9
Total de mensagens de busca em profundidade vistas: 2
Media de saltos ate encontrar destino por flooding: -nan
Media de saltos ate encontrar destino por random walk: -nan
Media de saltos ate encontrar destino por busca em profundidade: -nan

```

As estatísticas do nó 10. Condizem com o que vimos que ocorreu nas buscas (visto que a mensagem só é considerada como vista quando é recebida e não quando é enviada) e são números que fazem sentido. Na busca em flooding, como não há a chave na rede, é esperado que cada nó interaja uma vez com cada um dos seus vizinhos, que é o que ocorre. Na busca em random walking, é esperado que cada nó veja em média 10 vezes a mensagem, visto que há 10 nós na rede e um TTL de 100, que é o que ocorre. Na busca em profundidade, como há ciclo na rede o nó vê a mensagem duas vezes, que é uma vez a mais do que o normal.

```

2
Digite a chave a ser buscada
a
Encaminhando mensagem "192.168.31.106:5009 4 100 SEARCH FL 5009 a 1" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5009 4 100 SEARCH FL 5009 a 1"
Encaminhando mensagem "192.168.31.106:5009 4 100 SEARCH FL 5009 a 1" para 192.168.31.106:5010
Mensagem recebida: "192.168.31.106:5009 4 99 SEARCH FL 5008 a 2"
Flooding: Mensagem repetida!
Envio feito com sucesso: "192.168.31.106:5009 4 100 SEARCH FL 5009 a 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "192.168.31.106:5009 4 99 SEARCH FL 5010 a 2"
Flooding: Mensagem repetida!
Mensagem recebida: "192.168.31.109:5003 3 100 VAL FL a above 4"
Valor encontrado! Chave: a valor: above

```

Busca em flooding do nó 9

```

3
Digite a chave a ser buscada
a
Valor na tabela local!
chave: a valor: above

```

Tentativa de busca em random walking do nó 9 por chave que já está na tabela

```

Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
4
Digite a chave a ser buscada
a
Valor na tabela local!
chave: a valor: above

```

Tentativa de busca em profundidade do nó 9 por chave que já está na tabela

```

5
Estatísticas
Total de mensagens de flooding vistas: 2
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: 4
Media de saltos ate encontrar destino por random walk: -nan
Media de saltos ate encontrar destino por busca em profundidade: -nan

```

Estatísticas do nó 9. Como as buscas em profundidade e em random walking não foram efetivamente realizadas, suas estatísticas não foram atualizadas. O total de mensagens de flooding vistas segue o padrão comentado no nó 10. A quantidade de saltos faz sentido visto que há 3 nós entre o remetente e o detentor da chave.

```

3
Digite a chave a ser buscada
a
Encaminhando mensagem "192.168.31.106:5009 3 100 SEARCH RW 5009 a 1" para 192.168.31.106:5010
Envio feito com sucesso: "192.168.31.106:5009 3 100 SEARCH RW 5009 a 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "192.168.31.109:5003 3 100 VAL RW a above 6"
Valor encontrado! Chave: a valor: above
5
Estatísticas
Total de mensagens de flooding vistas: 0
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 0
Media de saltos ate encontrar destino por flooding: -nan
Media de saltos ate encontrar destino por random walk: 6
Media de saltos ate encontrar destino por busca em profundidade: -nan

```

Para que pudéssemos ver a execução da rede para os outros tipos de busca, reiniciamos a rede e as realizamos. A busca em random walking está de acordo com o esperado, pois levou uma quantidade de saltos similar à distância entre nós.

```

4
Digite a chave a ser buscada
a
Encaminhando mensagem "192.168.31.106:5009 3 100 SEARCH BP 5009 a 1" para 192.168.31.106:5010
Envio feito com sucesso: "192.168.31.106:5009 3 100 SEARCH BP 5009 a 1"
Escolha o comando:
[0] Listar vizinhos
[1] HELLO
[2] SEARCH (flooding)
[3] SEARCH (random walk)
[4] SEARCH (busca em profundidade)
[5] Estatísticas
[6] Alterar valor padrao de TTL
[9] Sair
Mensagem recebida: "192.168.31.106:5009 3 98 SEARCH BP 5008 a 3"
BP: Ciclo detectado, devolvendo a mensagem...
Encaminhando mensagem "192.168.31.106:5009 3 97 SEARCH BP 5009 a 4" para 192.168.31.106:5008
Envio feito com sucesso: "192.168.31.106:5009 3 97 SEARCH BP 5009 a 4"
Mensagem recebida: "192.168.31.109:5003 3 100 VAL BP a above 8"
Valor encontrado! Chave: a valor: above
5
Estatísticas
Total de mensagens de flooding vistas: 0
Total de mensagens de random walk vistas: 0
Total de mensagens de busca em profundidade vistas: 1
Media de saltos ate encontrar destino por flooding: -nan
Media de saltos ate encontrar destino por random walk: -nan
Media de saltos ate encontrar destino por busca em profundidade: 8

```

A busca em profundidade está de acordo com o esperado. O ciclo que nosso nó está inserido fez com que a mensagem desse mais saltos do que o que seria necessário.