# Roteamento IP

O objetivo do trabalho é implementar um sistema que simula o encaminhamento de pacotes em uma rede, passando por vários roteadores.

# Autor

- Jeferson Urbieta da Silva Neto

# O problema

Você desenvolverá um sistema simples de roteamento de mensagens, sem bibliotecas especiais, utilizando comunicação via protocolo UDP.

Dois programas devem ser desenvolvidos: um emissor, para envio de mensagens na rede, e um programa “roteador” que realiza o encaminhamento e das mensagens recebidas. Por simplicidade este último também fará o papel do destinatário, exibindo a mensagem recebida quando for o caso.

Serão inicializados vários programas roteadores e emissores. Cada mensagem será enviada a algum roteador, que a repassa para o próximo roteador, e assim sucessivamente até ser entregue ao destinatário.

Para mais informações sobre o problema e a descrição do trabalho veja o arquivo **descricao_trabalho.pdf**

## Programa

Foi implementado um sistema em java que faz o envio de mensagens UDP, e as recebe através de uma Thread sendo executada em paralelo para cada roteador iniciado.

Todos os pacotes são codificados em forma de texto para serem enviados ficando dessa forma

sourceAddress###destinationAddress###routerPort###message###tll###

ao ser recebido o pacote e decodificado e processado.

### Funcionalidades

Na implementação foi feita uma interação por linha de comando, ao iniciar o programa é possivel executar comandos de acordo com as funcionalidades desenvolvidas.

As funcionalidades implementadas podem ser vistas e acessadas através da ajuda que é printada ao iniciar o programa.

```
+---------+---------------+-------------------------------------------------------
------------+
| Command | Name          | Description
|
+---------+---------------+-------------------------------------------------------
------------+
| 0       | HELP          | Mostra os comandos implemetados no aplicativo
|
| 1       | EMIT_MESSAGE  | Emite uma messagem com os dados informados
|
| 2       | CREATE_ROUTER | Cria um roteador com os dados informados
|
| 3       | ROUTER_LIST   | Lista todos os Roteadores ativos na maquina
|
| 4       | ROUTER_DETAIL | Mostra as informações do roteador como a tabela de
roteamento     |
| 5       | STOP_ROUTER   | Para a execução de um roteador
|
+---------+---------------+-------------------------------------------------------
------------+
```

## Arquitetura

Na arquitetura do código ficou dividida em algumas services e objetos de negocio cada uma com seu objetivo descrito abaixo.

### Model

- Connection: Reponsalve por armazenar uma coneção aberta
- Package: Pacote enviado atraves da rede
- Redirection: Caminho na tabela de roteamento de um roteador
- Router: Objeto com o objetivo de representar um roteador
- RoutingTable: Tabela de roteamento de um roteador

### Service

- CommandService: Responsável por controlar a interação com o programa
- ConnectionService: Responsável por controlar a criação das conexões
- EmitterService: Responsável por emitir pacotes na rede
- RouterService: Responsável por criar, listar, detalhar e parar roteadores

## Emissor

No emissor é possível enviar um pacote ao para o destino especificado, sendo possível fazer o envido de 2 formas, através do comando **emissor** ou colocar o comando 3 e ir colocando as informações separadamente de acordo com que é solicitado.

## Roteador

No roteador é possível se criar, listar, detalhar e parar um roteador com os comandos específicos.

A criação assim como a emissão pode ser feita de 2 formas, através do comando **emissor** ou colocar o comando 3 e ir colocando as informações separadamente de acordo com que é solicitado.

Ao se criar um roteador na porta especifica o programa inicia uma thread que fica aguardando para receber algum pacote.

Cada roteador tem uma tabela de roteamento que fica responsável por armazenar os caminhos para o redirecionamento dos pacotes e por processar o recebimento de um pacote para encontrar a rota para que ele seja encaminhado.

## Tecnologias

- Java
- log4j

## Execução

Para a execução execute o método main da classe Main localizada em **src/main/java/br/com/urbieta/jeferson/ApplicationMain.java**

## Teste

Alguns do casos de testes executados

## Criando os roteadores

### Criando o roteador para porta 1111

```
roteador 1111 10.0.0.0/255.0.0.0/0.0.0.0/0 20.20.0.0/255.255.0.0/0.0.0.0/0
30.1.2.0/255.255.255.0/127.0.0.1/2222
```

Tabela de roteamento criada

```
+-----------------+----------------+----------------+-----------------+
| Destiny         | Mask           | Gateway        | Interface Output|
+-----------------+----------------+----------------+-----------------+
| 10.0.0.0        | 255.0.0.0      | 0.0.0.0        | 0               |
| 20.20.0.0       | 255.255.0.0    | 0.0.0.0        | 0               |
| 30.1.2.0        | 255.255.255.0  | 127.0.0.1      | 2222            |
+-----------------+----------------+----------------+-----------------+
```

### Criando o roteador para porta 2222

```
roteador 2222 10.0.0.0/255.0.0.0/127.0.0.1/3333 20.20.0.0/255.255.0.0/0.0.0.0/0
0.0.0.0/0.0.0.0/127.0.0.1/3333 30.1.2.0/255.255.255.0/0.0.0.0/0
```

Tabela de roteamento criada

```
+-----------------+----------------+----------------+-----------------+
| Destiny         | Mask           | Gateway        | Interface Output|
+-----------------+----------------+----------------+-----------------+
| 10.0.0.0        | 255.0.0.0      | 127.0.0.1      | 3333            |
| 20.20.0.0       | 255.255.0.0    | 0.0.0.0        | 0               |
| 0.0.0.0         | 0.0.0.0        | 127.0.0.1      | 3333            |
| 30.1.2.0        | 255.255.255.0  | 0.0.0.0        | 0               |
+-----------------+----------------+----------------+-----------------+
```

### Criando o roteador para porta 3333

```
roteador 3333 10.0.0.0/255.0.0.0/127.0.0.1/2222
20.20.0.0/255.255.0.0/127.0.0.1/2222 30.1.2.0/255.255.255.0/0.0.0.0/0
0.0.0.0/0.0.0.0/127.0.0.1/4444
```

Tabela de roteamento criada

```
+-----------------+----------------+----------------+-----------------+
| Destiny         | Mask           | Gateway        | Interface Output|
+-----------------+----------------+----------------+-----------------+
| 10.0.0.0        | 255.0.0.0      | 127.0.0.1      | 2222            |
| 20.20.0.0       | 255.255.0.0    | 127.0.0.1      | 2222            |
| 30.1.2.0        | 255.255.255.0  | 0.0.0.0        | 0               |
| 0.0.0.0         | 0.0.0.0        | 127.0.0.1      | 4444            |
+-----------------+----------------+----------------+-----------------+
```

## Emitindo pacotes para os roteadores

### Teste rota direta

Esperado: Imprimir que chegou no R1111 ````` emissor 127.0.0.1 1111 1.1.1.1 10.0.0.5 Cheguei_R1?!

```
Resultado
```

Router 1111| Destination reached. From 1.1.1.1 to 10.0.0.5 : Cheguei_R1?!

```
###  Teste rota indireta

Esperado:  Imprimir que  passa em R1111, chega no R2222
  ````
emissor 127.0.0.1 1111 1.1.1.1 30.1.2.10 Cheguei_R2!?
```
```

Resultado

```
Router 1111| Forwarding packet for 30.1.2.10 to next hop 127.0.0.1 over interface
2222
Router 2222| Destination reached. From 1.1.1.1 to 30.1.2.10 : Cheguei_R2!?
```

## Teste rota default

Esperado: Imprimir que passa por R2222 e passa por R3333 ```` emissor 127.0.0.1 2222 1.1.1.1 20.0.2.1 Repassado_a_4444?!

```
Resultado
```

Router 2222| Forwarding packet for 20.0.2.1 to next hop 127.0.0.1 over interface 3333 Router 3333| Forwarding packet for 20.0.2.1 to next hop 127.0.0.1 over interface 4444

```
###  Teste rota inexistente
Esperado:  R1111 descarta pacote
  ````
emissor 127.0.0.1 1111 2.2.2.2 40.0.40.1 Descartado!!!
```

Resultado

```
Router 1111| Destination 40.0.40.1 not found in routing table, dropping packet
```

## Teste Loop em Rota

Esperado: R3333 passa a R2222 que devolve a R3333 ... ````` emissor 127.0.0.1 3333 2.2.2.2 10.10.10.10 TTL_excedido!!!

```
Resultado
```

Router 3333| Forwarding packet for 10.10.10.10 to next hop 127.0.0.1 over interface 2222 Router 2222| Forwarding packet for 10.10.10.10 to next hop 127.0.0.1 over interface 3333 Router 3333| Forwarding packet for 10.10.10.10 to next hop 127.0.0.1 over interface 2222 Router 2222| Forwarding packet for 10.10.10.10 to next hop 127.0.0.1 over interface 3333 Router 3333| Time to Live exceeded in Transit, dropping packet for 10.10.10.10 `````
```