

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
CURSO DE ENGENHARIA DE SOFTWARE
REDE DE COMPUTADORES

FELIPE ALVES MATOS CAGGI
HUELIGTON PEREIRA MELO

RELATÓRIO DE DESENVOLVIMENTO
ROTEAMENTO IP

CAMPO GRANDE
2019

Tabela de Conteúdos

1	INFORMAÇÃO GERAL.....	3
1.1	Visão Geral do Sistema.....	3
2	ARQUITETURA	4
2.1	Emissor	5
2.1.1	Inicialização do Emissor	5
2.1.2	Construção do pacote.....	5
2.1.3	Envio do pacote.....	5
2.2	Roteador	6
2.2.1	Inicialização do Roteador	6
2.2.2	Carregamento da Tabela de Roteamento	6
2.2.3	Abertura de Conexão.....	6
2.2.4	Manipulação de Requisições.....	6
2.2.5	Decremento de TTL	6
2.2.6	Verificação de Match com Tabela de Roteamento	6
3	ESTRUTURAS DE DADOS	7
3.1	Datagram Socket e Datagram Packet.....	7

1 INFORMAÇÃO GERAL

1.1 Visão Geral do Sistema

O roteamento ip desenvolvido é um simulador de encaminhamento de pacotes em uma rede de roteadores.

O sistema apresenta duas classes principais, Emissor e Roteador, que podem receber várias instâncias de execução concorrentes, simulando emissores e roteadores em uma rede. Os emissores estão responsáveis por criar pacotes e enviar ao roteador. E os roteadores são responsáveis pelo encaminhamento de pacotes dentro da rede.

2 ARQUITETURA

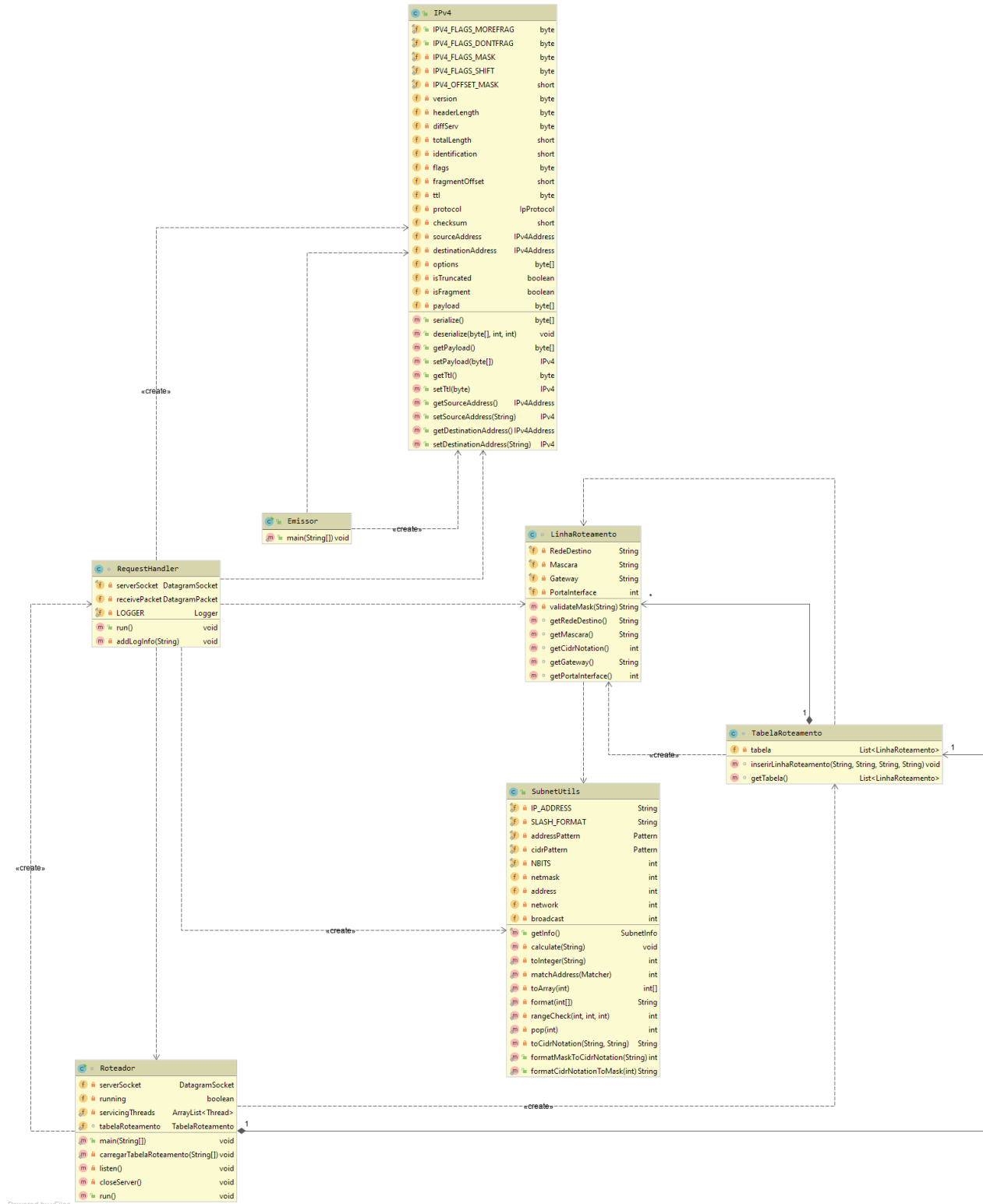


Figura 1 - Diagrama de classe do Emissor e Roteador

2.1 Emissor

2.1.1 Inicialização do Emissor

O programa Emissor é inicializado sempre que um pacote precisa ser enviado, recebendo como argumento as informações de endereçamento e a mensagem(dados) a ser enviada.

Segue abaixo a lista de argumentos utilizados para criação de pacote de dados pelo Emissor:

arg[0].	Endereço IP do roteador default(inicial)
arg[1].	Porta do roteador default(inicial)
arg[2].	Endereço de origem
arg[3].	Endereço de destino
arg[4].	Dados da mensagem

2.1.2 Construção do pacote

A construção do pacote é realizada através da criação de uma instância da classe IPv4, especificando os endereços de origem, destino, juntamente com o *Time To Live (TTL)*¹ do pacote e os dados da mensagem;

2.1.3 Envio do pacote

Para envio é utilizado um Datagram Socket², classe onde cada pacote enviado ou recebido em um soquete de datagrama é endereçado e roteado individualmente. Vários pacotes enviados de uma máquina para outra podem ser roteados de maneira diferente e podem chegar em qualquer ordem.

¹ Um dos requisitos de implementação deste projeto define tempo 5 do TTL durante a criação de pacotes. Dessa forma, o TTL do pacote é sempre definido em 5 durante a construção do pacote pelo programa Emissor implementado neste projeto.

² <https://docs.oracle.com/javase/7/docs/api/java/net/DatagramSocket.html>

2.2 Roteador

2.2.1 Inicialização do Roteador

O programa Roteador recebe como argumento a porta de execução e informações da tabela de roteamento.

Segue abaixo a lista de argumentos utilizados durante a inicialização do Roteador:

arg[0].	Interface de execução do Roteador
arg[1].	Entrada 1 da tabela de roteamento
arg[2].	Entrada 2 da tabela de roteamento
arg[3].	...

Cada entrada da tabela de roteamento é especificada no formato rede-destino/máscara/gateway/interface .

2.2.2 Carregamento da Tabela de Roteamento

Antes de disponibilizar a conexão ao Emissor, a tabela de carregamento deve ser construída. Para isso, o roteador possui em seu ciclo de vida uma instância da classe TabelaRoteamento, que armazenará uma List com instâncias da classe LinhaRoteamento, que representa cada argumento de entrada na tabela, possuindo os campos equivalentes ao formato rede-destino/máscara/gateway/interface.

2.2.3 Abertura de Conexão

Finalizado o carregamento da tabela de roteamento, o roteador receberá uma instância de um DatagramSocket com a porta de execução especificado no argumento de inicialização do programa, estando assim pronto para receber requisições do cliente.

2.2.4 Manipulação de Requisições

Uma thread é criada para cada requisição recebida, recebendo como parâmetro de construção uma instância da classe RequestHandler. Classe esta responsável por manipular o pacotes IPv4.

2.2.5 Decremento de TTL

Como primeiro passo de manipulação de requisição, é realizado o decremento do TTL do pacote e em seguida verificado se o valor do TTL é maior do que 0. Se o TTL for igual a 0, é feito o descarte do pacote por tempo de vida excedido em trânsito.

2.2.6 Verificação de Match com Tabela de Roteamento

É verificando se o destino do pacote se encontra na tabela de roteamento. Em casos negativos, o pacote é descartado. E para casos positivos é verificado o próximo destino do pacote, fazendo a entrega ou encaminhamento ao próximo roteador.

3 ESTRUTURAS DE DADOS

3.1 Datagram Socket e Datagram Packet

As classes Java DatagramSocket e DatagramPacket são usadas para programação de soquete sem conexão. A classe Java DatagramSocket representa um soquete sem conexão para enviar e receber pacotes de datagramas.

Um datagrama é basicamente uma informação, mas não há garantia de seu conteúdo, chegada ou hora de chegada.

Java DatagramPacket é uma mensagem que pode ser enviada ou recebida. Se você enviar vários pacotes, ele poderá chegar em qualquer ordem. Além disso, a entrega de pacotes não é garantida.

Construtores comumente usados da classe DatagramPacket (byte [] barr, int length): cria um pacote de datagrama.

O construtor DatagramSocket () usados da classe DatagramSocket cria um soquete de datagrama e o vincula ao número da porta disponível na máquina localhost.

O construtor DatagramPacket (byte [] barr, comprimento int, endereço InetAddress, porta int) da classe DatagramPacket é usado para enviar os pacotes.