

# **Trabalho Conjunto entre as Disciplinas de Sistemas Operacionais e Fundamentos de Redes de Computadores**

**Professores: Cristina Nunes, Carlos R. Moratelli**

**Entrega 06/06/2017**

## **Roteiro**

### **1. Objetivos**

1. Utilizar programação multithreading;
2. Praticar técnicas de sincronização de processos;
3. Aprofundar conceitos de endereçamento de rede;
4. Troca de mensagens entre processos;
5. Roteamento de Rede;
6. Explorar o conceito de interoperabilidade.

### **2. Descrição**

Desenvolver uma aplicação que simule um roteador de rede e que faça a troca de tabelas de roteamento, semelhante ao que foi apresentado em aula, utilizando sockets UDP para a comunicação. A aplicação deve implementar o protocolo descrito na seção 3.

Inicialmente, deverá ser informado, na aplicação, os endereços IPs dos *roteadores vizinhos* para que várias topologias diferentes possam ser simuladas. Cada roteador vizinho é uma instância do roteador implementado executando em outra máquina física. Esses endereços IPs deverão ser cadastrados em uma tabela de roteamento com métrica 1 e saída direta. Três campos deverão estar presentes na tabela de roteamento: IP de Destino, Métrica e IP de Saída.

As tabelas de roteamento (apenas os campos IP de Destino e Métrica) deverão ser trocadas entre os roteadores vizinhos a cada 10 segundos. Ao receber a tabela de roteamento de seus vizinhos, a aplicação deverá verificar as rotas recebidas e fazer as atualizações necessárias na tabela de roteamento local. Uma atualização deverá ser feita sempre que:

- for recebido um IP de Destino não presente na tabela local. Neste caso a rota deve ser adicionada, a Métrica deve ser incrementada em 1 e o IP de Saída deve ser o endereço do roteador que ensinou esta informação;

- for recebida uma Métrica menor para um IP Destino presente na tabela local. Neste caso, a Métrica e o IP de Saída devem ser atualizadas;
- um IP Destino deixar de ser divulgado. Neste caso, a rota deve ser retirada da tabela de roteamento.

Um roteador pode sair da rede a qualquer momento. Isso significa que seus vizinhos não receberão mais anúncios de rotas. Assim, depois de 30 segundos sem receber mensagens do roteador vizinho em questão, as rotas que passam por ele devem ser esquecidas.

Periodicamente, a tabela de roteamento local deverá ser apresentada para o usuário. Além disso, alterações na tabela de roteamento deverão ser informadas para os usuários (através de prints na saída padrão).

A aplicação deverá rodar sobre o **protocolo UDP**.

### 3. Protocolo de comunicação

A implementação deve respeitar fielmente o formato de mensagens descrito a seguir. A aplicação resultante deve ser **interoperável**, ou seja, implementações de diferentes grupos devem ser capazes de se comunicar entre si. Desta maneira, poderá ser construída uma topologia com roteadores implementados por diferentes equipes.

O protocolo consiste em apenas duas mensagens, conforme descrito a seguir.

#### Mensagem 1 - Anúncio de rotas

Esta mensagem deve ser enviada aos vizinhos a cada 10 segundos e consiste no envio da tabela de roteamento para os roteadores vizinhos. A mensagem consiste em uma lista de endereços IP's em formato string. Por exemplo, se a tabela for:

IP	Métrica	Saída
192.168.1.2	1	192.168.1.1
192.168.1.3	1	192.168.1.1

A mensagem enviada será:

\*192.168.1.2;1\*192.168.1.3;1

Ou seja, “\*” (asterisco) indica uma tupla, IP de Destino e Métrica. A métrica é separada do IP por um “;” (ponto e vírgula).

O protocolo não prevê confirmação de recebimento de mensagem, pois a tabela deverá ser reenviada a cada 10 segundos. Contudo, caso o recebimento de uma mensagem de anúncio de rotas cause a alteração da tabela de roteamento, o roteador deve enviar sua tabela imediatamente para seus vizinhos.

## Mensagem 2 - Anúncio de roteador

Esta mensagem deverá ser enviada aos vizinhos a cada 10 segundos somente se a tabela de roteamento estiver vazia. Assim, logo que um roteador entrar na rede e não tiver nenhuma rota pré-configurada, deverá anunciar-se para os vizinhos.

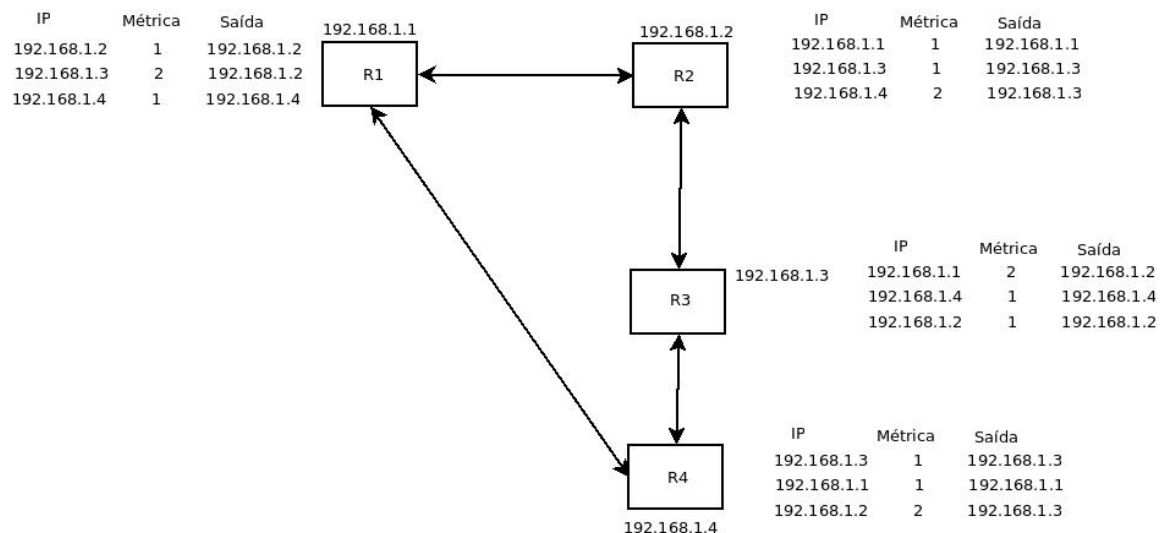
A mensagem enviada será:

!

Ou seja, apenas um ponto de exclamação para anunciar sua entrada.

## 4. Exemplo de topologia com 4 roteadores

A figura abaixo ilustra 4 roteadores e suas respectivas tabelas de roteamento depois de



algumas iterações para troca de tabelas. As flechas indicam roteadores vizinhos.

## **5. Considerações sobre a implementação**

A aplicação deve ser multithread, ou seja, devem existir ao menos duas threads: uma para receber mensagens dos vizinhos e atualizar a tabela de roteamento e outra para enviar a tabela de roteamento para os vizinhos a cada 10 segundos, ou quando a mesma for alterada. Deve-se evitar condições de corrida ao acesso (leitura/escrita) na tabela de roteamento.

Os endereços IP dos roteadores vizinho devem ser informados no arquivo denominado IPVizinhos.txt (um por linha), conforme implementação de referência.

A interoperabilidade é uma questão fundamental em redes de computadores. Ela permite que equipamentos de diferentes fabricantes operem em harmonia na rede. Para isso, as especificações dos protocolos devem ser rigidamente implementadas. Espera-se neste trabalho, que as diversas equipes apresentem implementações coerentes com a especificação e que permita a interoperabilidade com as implementações das outras equipes.

Como base para o início da implementação, utilize o esqueleto de código fornecido pelos professores. Tal implementação ilustra a comunicação através de sockets UDP. As mensagens de anúncio de rotas devem ser enviadas para a porta 5000 do roteador vizinho.

## **6. Entrega**

O trabalho pode ser realizado em grupos com no máximo 3 alunos. É permitido realizar o trabalho individualmente ou em grupos de 2 alunos. Junto com a implementação, em linguagem Java, deve ser entregue um relatório de no máximo 3 páginas explicando detalhes da implementação e dificuldades encontradas. A entrega do trabalho deve ser realizada pelo Moodle (código fonte e relatório) e o mesmo deve ser apresentado em sala de aula na data indicada no cronograma da disciplina. Para a apresentação, os alunos devem construir topologias com no mínimo 4 roteadores. Ainda, na apresentação, as equipes serão convidadas a colocar seus roteadores em uma topologia definida pelo professor e que envolverá diversas equipes.