



Pontifícia Universidade Católica do Rio Grande do Sul

Escola Politécnica

Fundamentos de Redes de Computadores

Definição do Trabalho Final

Objetivo

O objetivo do trabalho é desenvolver uma aplicação que simule o comportamento básico de um protocolo de roteamento dinâmico de vetor de distância, utilizando conceitos de tabelas de roteamento, métricas, atualização periódica de rotas e técnicas para evitar loops de roteamento, como o *Split Horizon*. O trabalho deverá ser implementado em linguagem de programação escolhida pelo grupo e testado em ambiente de rede local.

A aplicação deve possuir as seguintes partes, conforme descrito a seguir.

Parte 1 – Inicialização e Tabela de Roteamento

Para a implementação desta parte, os computadores de uma mesma rede deverão assumir o papel de roteadores. Quando um roteador iniciar, ele deverá ler um arquivo de configuração contendo os endereços IP de seus vizinhos diretos. Neste momento, os endereços IP informados deverão ser cadastrados em uma tabela de roteamento inicial, com métrica igual a 1 e IP de saída correspondente ao roteador vizinho.

A tabela de roteamento deve conter os seguintes campos: IP de Destino, Métrica e IP de Saída

Além disso, o roteador deve implementar a técnica de Split Horizon, que impede o envio de informações de rota de volta para o roteador de onde elas foram aprendidas, reduzindo assim a possibilidade de loops de roteamento.

Parte 2 – Atualização de Rotas

Cada roteador deve enviar, a cada 10 segundos, mensagens de atualização com o conteúdo da sua tabela de roteamento (apenas os campos IP de Destino e Métrica) para seus vizinhos.

Ao receber uma mensagem de atualização, o roteador deverá:

1. Verificar se há novas rotas ou melhorias em rotas já existentes (menor métrica).
2. Atualizar sua tabela de roteamento quando necessário.
3. Aplicar novamente o *Split Horizon* ao reenviar informações para os vizinhos.
4. Exibir no terminal, a cada atualização, a tabela de roteamento atualizada, destacando as alterações realizadas.
5. Informar a mudança aos demais vizinhos.

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.

Parte 3 – Detecção de Falhas

O sistema deve prever a perda de conectividade com algum vizinho. Caso um roteador deixe de receber mensagens de atualização de um vizinho, depois de 15 segundos, ele deve:

- Considerar o vizinho como inativo;
- Remover as rotas associadas a esse vizinho (**rota para ele e que passam por ele devem ser esquecidas**);
- Atualizar sua tabela de roteamento.

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).

Parte 4 - 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 grupos.

O protocolo de comunicação 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 em formato string. Por exemplo, se a tabela for:

IP	Métrica	Saída
192.168.1.2	1	192.168.1.2
192.168.1.3	1	192.168.1.3

A mensagem enviada será:

***192.168.1.2;1*192.168.1.3;1**

Ou seja, “*” 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 mensagens de roteamento, pois a tabela será 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. **Os roteadores não devem incluir rotas para eles mesmos.**

Mensagem 2 - Anúncio de roteador

Esta mensagem será enviada aos vizinhos (endereços IP que foram informados pelo usuário no arquivo de configuração) **somente quando o roteador entrar em uma rede já existente**. Neste caso, o roteador deverá anunciar-se para os vizinhos para que estes os incluam em suas tabelas de roteamento com métrica 1.

A mensagem enviada será:

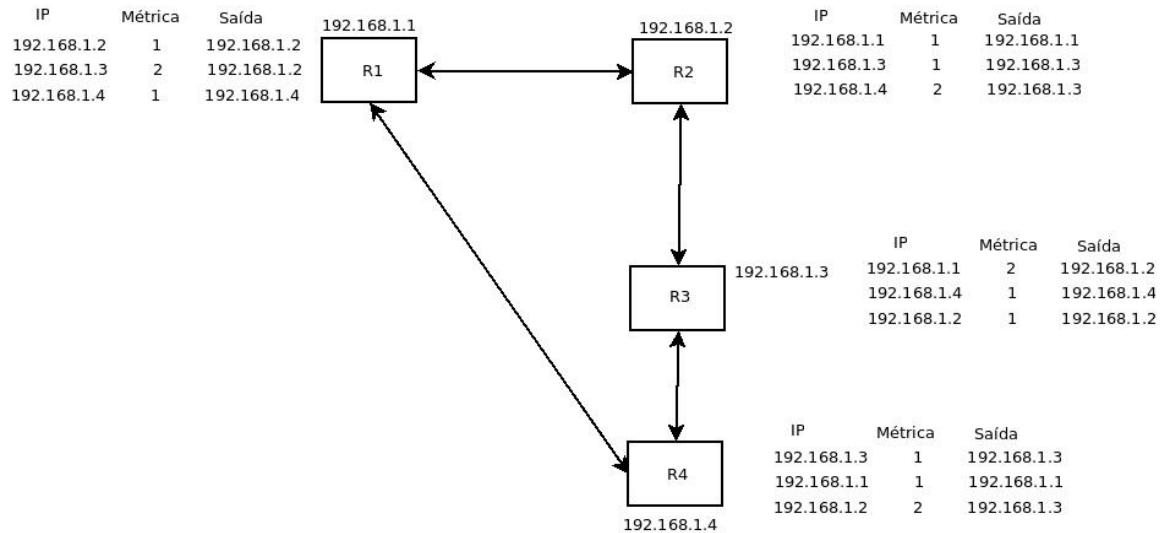
@192.168.1.1

Ou seja, um **@** (arroba) seguido do próprio endereço IP do roteador que entrou na rede.

Após o seu anúncio, o roteador que entrou na rede já existente deve criar a sua tabela de roteamento com os endereços IP informados pelo usuário e seguir o funcionamento do protocolo já especificado.

Parte 5 - 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.



Parte 6 – Envio de Mensagens entre os roteadores

A implementação também deve possibilitar o **envio de mensagens de texto entre os roteadores**, sendo que as mensagens devem ser roteadas de acordo com as tabelas de roteamento das máquinas.

A qualquer momento o usuário da aplicação pode enviar **mensagens de texto** para os roteadores. Neste caso, será necessário informar o **endereço IP do roteador destino e a mensagem de texto**.

O roteador deve aceitar apenas mensagens de texto direcionadas para ele. Caso a mensagem não seja para ele, será necessário roteá-la para o próximo roteador do caminho ou para o destino, conforme está especificado em sua tabela de roteamento.

Ao receber uma mensagem de texto, o roteador deve **imprimir a mensagem recebida, bem como o endereço IP da origem e o IP do destino e indicar se a mensagem está sendo repassada adiante ou se chegou ao destino**.

Uma mensagem de texto deve ter o seguinte formato:

!192.168.1.2;192.168.1.1;Oi tudo bem?

Ou seja, “!” indica que uma mensagem de texto foi recebida. O primeiro endereço é o IP da origem, o segundo é o IP de destino e a seguir vem a mensagem de texto. Cada informação é separada um “;” (ponto e vírgula).

Parte 7 - Considerações sobre a implementação

Os endereços IP dos roteadores vizinhos devem ser informados no arquivo denominado **roteadores.txt**, sendo especificado um endereço IP por linha.

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 os diversos grupos apresentem implementações coerentes com a especificação e que permita a interoperabilidade com as implementações dos outros grupos.

A implementação deve utilizar socket **UDP** e as mensagens devem ser enviadas para a porta **6000** do roteador vizinho.

Regras Gerais

Grupos: Até 4 componentes

Data de entrega: 12/11

Obs.: Todos os participantes devem estar presentes

Visualização dos Resultados:

- A demonstração deverá acontecer, no mínimo, em 4 máquinas. Os grupos serão convidados a colocar seus roteadores em uma topologia definida pelo professor e que envolverá diversos grupos.

IMPORTANTE: Não serão aceitos trabalhos entregues fora do prazo. Trabalhos que não compilam ou que não executam não serão avaliados. Todos os trabalhos serão analisados e comparados. Caso seja identificada cópia de trabalhos, todos os trabalhos envolvidos receberão nota ZERO.