

MAC0448/5910 - Programação para Redes de Computadores

EP3

Datas de Entrega: 16/10/2012

Prof. Daniel Macêdo Batista

1 Problema

Neste EP você deverá implementar um simulador de protocolos de roteamento. O simulador receberá como entrada a topologia de uma rede, processará as informações desta topologia e construirá as tabelas de roteamento de cada roteador. Após esta etapa, será apresentado um *prompt* para que o usuário possa solicitar as rotas entre dois roteadores quaisquer da rede simulada.

A implementação dos protocolos de roteamento deverá ser feita do ponto de vista de cada roteador separado, como seria feito em um roteador real.

2 Requisitos

2.1 Arquitetura do simulador

O simulador deve receber na linha de comando apenas um argumento, que será o nome de um arquivo texto contendo a topologia da rede que será simulada. Esta topologia estará representada como uma matriz de adjacências. A rede da Figura 1 por exemplo é representada pelo arquivo a seguir:

```
-1 2 5 1 -1 -1
2 -1 3.5 2 -1 -1
5 3.5 -1 3 1 3
1 2 3 -1 1 -1
-1 -1 1 1 -1 2
-1 -1 3 -1 2 -1
```

Os números apresentados à direita dos roteadores na Figura 1 são os identificadores dos roteadores. Os números em vermelho perto dos enlaces representam o atraso em milissegundos entre os roteadores.

Antes de aceitar comandos do usuário o programa deverá exibir todas as trocas de mensagens entre roteadores (não tem problema se isso deixar a saída poluída) para a montagem das tabelas de roteamento dos dois algoritmos de roteamento que serão especificados a seguir.

Após a exibição das trocas de mensagens deve ser exibido um prompt para o usuário. Esse prompt deverá receber comandos deste tipo:

```
<algoritmo> <origem> <destino> <metrica>
```

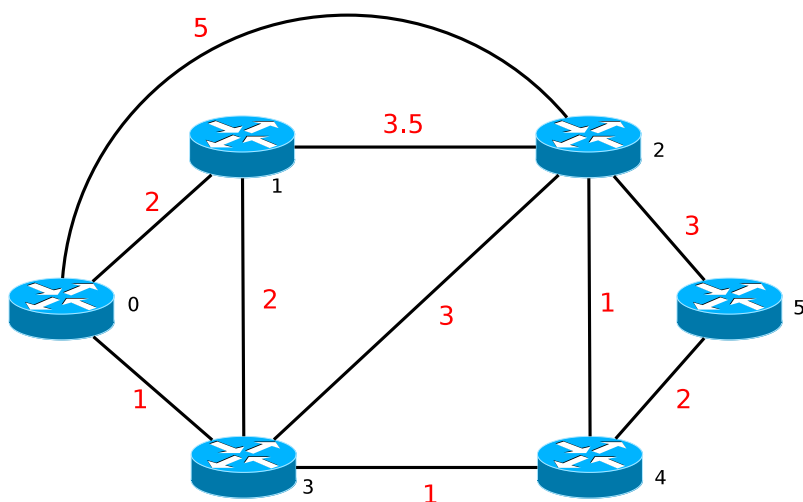


Figura 1: Exemplo de topologia de rede (Baseado em exemplo do livro do Kurose, página 271, quinta edição em português)

<algoritmo> pode ser ee ou vd. Caso o usuário digite ee, o algoritmo de roteamento considerado deve ser o algoritmo de estado de enlace apresentado em sala de aula (páginas 273 a 276 da quinta edição em português do livro do Kurose). Caso o usuário digite vd, o algoritmo de roteamento considerado deve ser o algoritmo de vetor de distâncias visto em sala de aula (páginas 276 a 279 do livro da quinta edição em português do livro do Kurose).

origem e destino são identificadores dos roteadores da rede. A rota deverá ser traçada entre origem e destino.

metrica pode ser h ou a. Caso o usuário digite h, o cálculo do caminho entre os roteadores deve considerar o número de *hops*. Caso o usuário digite a, o cálculo do caminho entre os roteadores deve considerar o atraso dos enlaces.

O retorno do algoritmo para cada comando deverá ser a rota entre os roteadores informados pelo usuário e o custo desta rota (em número de *hops* ou em termos do atraso do caminho). Abaixo há dois exemplos de como deve ser a saída:

```
0 1 3 10 25 (122 milisegundos)
```

```
4 3 22 (3 hops)
```

Além dos comandos solicitando as rotas, o prompt também deverá receber a mensagem `quit`. Esta mensagem encerrará a execução do simulador.

2.2 Implementação

A sua implementação não pode considerar uma entidade central na rede que conhece toda a rede e constrói as tabelas de roteamento de todos os roteadores. Você deve implementar seu algoritmo de modo que ele seja executado em cada roteador separado com as informações que cada roteador possui, já que é assim que acontece na Internet. Note que para isto funcionar é necessário que os roteadores troquem mensagens entre eles para as construções das tabelas. Você terá que explicar as mensagens que você definiu para o seu algoritmo. Essa explicação deve estar presente no LEIAME.

Cada roteador deve ser uma entidade distinta no seu código (a instância de uma classe, uma *thread*, um processo, etc...). Todos os roteadores devem ser implementados da mesma forma.

2.3 Linguagem

Os códigos podem ser escritos em qualquer linguagem, desde que os algoritmos sejam de fato implementados pelos integrantes do grupo. A utilização de uma biblioteca que já implementa o algoritmo implicará em nota ZERO no EP.

3 Entrega

Você deverá entregar um arquivo .tar.gz contendo os seguintes itens:

- fonte;
- Makefile (ou similar);
- arquivo LEIAME. Você deverá explicar no LEIAME o protocolo de trocas de mensagens que você definiu entre os roteadores durante a fase de construção das tabelas de roteamento.

O desempacotamento do arquivo .tar.gz deve produzir um diretório contendo os itens. O nome do diretório deve ser ep3-membros_da_equipe. Por exemplo: ep3-joao-maria.

A entrega do .tar.gz deve ser feita através do PACA.

O EP pode ser feito individualmente ou em dupla.

Obs.: Serão descontados pontos de EPs que não estejam nomeados como solicitado, que não criem o diretório com o nome correto após serem descompactados e que não contenham todos os arquivos necessários.

Obs.: O prazo de entrega expira às 08:00 do dia 16/10/2012. EPs entregues com atraso terão -1,0 por cada hora de atraso.

4 Avaliação

80% da nota será dada pela implementação e 20% pelo LEIAME. Os critérios detalhados da correção serão disponibilizados apenas quando as notas forem liberadas.

5 Referências úteis dos livros

- Kurose, 5a. edição em português:
 - Páginas 229 a 234 (Capítulo 4, seção “Introdução”), 270 a 282 (Capítulo 4, seção “Algoritmos de Roteamento”)