MAC0448/5910 - Programação para Redes de Computadores
EP3
Datas de Entrega: 20/11/2012

Prof. Daniel Macêdo Batista

1 Problema

Neste EP você deverá implementar um simulador de protocolos de roteamento multicast. O simulador receberá como entrada a topologia de uma rede, processará as informações desta topologia e construirá as árvores multicast necessárias para a comunicação entre os hospedeiros à medida que o usuário solicitar. O seu programa deverá apresentar um *prompt* para que o usuário possa criar um novo grupo multicast ou solicitar a inclusão ou a remoção de um computador de uma subrede em um dado grupo multicast.

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 dois argumentos. O primeiro 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 milisegundos entre os roteadores.

O segundo argumento passado para o programa será o algoritmo utilizado para gerenciar a árvore multicast. Pode ser `shared` caso seja utilizado o roteamento multicast usando uma árvore compartilhada pelo grupo ou `source` caso seja utilizado o roteamento multicast usando uma árvore baseada na fonte (Páginas 303 e 304 da quinta edição em português do livro do Kurose – Capítulo 4, seção “Algoritmos de roteamento multicast”)

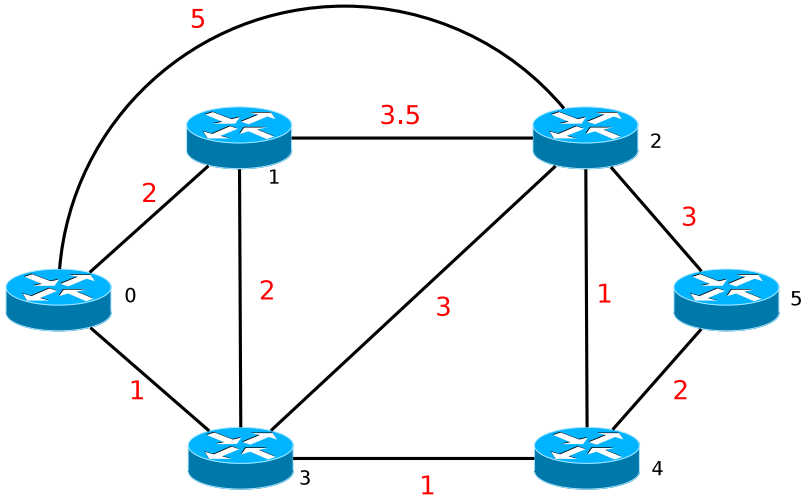

Figura 1: Exemplo de topologia de rede (Baseado em exemplo do livro do Kurose, página 271, quinta edição em português)

Após processar a topologia de entrada seu programa deverá exibir um prompt que aceitará os comandos abaixo:

```
send <netid>
join <netid> <group>
leave <netid> <group>
```

- `send`: comando inicial que deverá ser executado para a criação de um novo grupo multicast. Esse comando simulará o desejo de um usuário de iniciar uma transmissão de vídeo via multicast por exemplo. Seu simulador deverá armazenar quem é a origem deste grupo multicast e devolver no prompt o id do grupo multicast criado. O parâmetro `<netid>` é o identificador do roteador onde o host está conectado;

- `join`: comando executado por hosts da rede para fazerem parte de um grupo multicast. Seu simulador deverá armazenar todos os computadores que fazem parte de um grupo multicast e deverá atualizar a árvore multicast necessária para que os dados sejam transferidos para todos. O parâmetro `<netid>` é o identificador do roteador onde o host que solicitou o `join` está conectado. O parâmetro `<group>` é o identificador do grupo multicast no qual o host vai se conectar.

- `leave`: comando executado por hosts da rede para deixarem um grupo multicast. Seu simulador deverá atualizar a lista de computadores que fazem parte de um grupo multicast e deverá atualizar a árvore multicast sempre que um comando de `leave` for executado. O parâmetro `<netid>` é o identificador do roteador onde o host que solicitou o `leave` está conectado. O parâmetro `<group>` é o grupo do qual o host vai sair.

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

Após cada comando `join` e `leave` é necessário que sejam listados todos os grupos multicast existentes no momento. Os roteadores sendo usados por cada grupo também devem ser apresentados como no exemplo abaixo:

```
group 0: netid 1 eh a fonte dos dados
         netid 3 tem 1 receptor dos dados
         netid 4 tem 3 receptores dos dados
         netid 6 tem 5 receptores dos dados
         netid 7 tem 1 receptor dos dados
         ---
         arvore raiz: netid 1
         arvore nivel 2: netid 2, netid 3
         arvore nivel 3: netid 4, netid 7
         arvore nivel 4: netid 6
```

Após os comandos `join` e `leave` também é necessário que seu simulador exiba as informações sobre o tráfego de mensagens entre roteadores para a atualização das árvores multicast. Não tem problema se a exibição dessas mensagens deixar a saída poluída.

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 os algoritmos de modo que eles sejam executados 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. 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 ep4-membros_da_equipe. Por exemplo: ep4-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 300 a 304 (Capítulo 4, seção “Multicast”),