Algoritmia e Desempenho em Redes de Computadores

2º Mini Projeto - *Inter-domain routing*

Bernardo Gomes, 75573 Tomás Falcato, 75876

11 de Novembro de 2015

1 Descrição do problema

Neste mini-projecto, pretende-se a implementação de diversas funções relacionadas com o tipo de rota comercial eleita de cada nó de uma dada rede para um destino eleito.

Assim, a função *Dijkstra* elege o tipo de rota e calcula o número de saltos necessários para chegar ao destino.

O cálculo das estatísticas para as diferentes rotas e para os números de saltos até ao destino, é feito depois da função anterior ser corrida para todos os destinos possíveis no grafo. Este é feito pelas funções *paths_statistics* e *number_hops_statistics*.

2 Abordagem ao problema

Decidimos utilizar um algoritmo baseado no algoritmo de *Dijkstra* para a computação das rotas a utilizar dos diversos nós para um destino eleito. Como tal, foi necessário que o programa interpretasse os dados de entrada provenientes de um ficheiro de texto por forma a ficar com a topologia da rede. A topologia é armazenada numa lista de adjacências.

Por questões de tempo de resolução do problema e tendo sido assegurado que o número de nós dos grafos não excederia o valor máximo de alocação de memória, a lista de adjacências criada tem por base um vector de ponteiros em que cada posição contém uma lista para as adjacências de cada nó.

Sendo as estatísticas calculadas depois de cada nó saber os caminhos a escolher tendo por base todos os destinos possíveis, após a leitura do ficheiro, o programa procede à realização da nossa função *Dijkstra* para todos os destinos, no fim de cada iteração atualiza-se o número de rotas utilizadas de cada tipo e o número de saltos utilizados para chegar ao destino. No fim de todos os cálculos são então feitas as estatísticas, sendo posteriormente apresentadas ao utilizador.

3 Função *Dijkstra*

3.1 Função *Insert*

4 Estatísticas

Para o cálculo das estatísticas pedidas, utilizámos as variáveis *ones*, *twos*, *threes* e o vector *stat_hops*, que estão constantemente a ser atualizados no fim de cada iteração, recalculando o número de rotas *customer*, *peer*, *provider* e o número de vezes que se chega ao destino em i saltos, sendo i o índice do vector *stat_hops*.

4.1 Função *paths_ statistics*

O pseudo-código da função apresentada é o seguinte:
Result: estatísticas das rotas utilizadas
stat_customer=ones/total_routes;
stat_peer=twos/total_routes;
stat_peer=threes/total_routes;

Algorithm 1: *paths_ statistics*

4.2 Função *number_ hops_ statistics*

O pseudo-código da função apresentada é o seguinte:
Result: estatísticas para cada número de saltos
counts total number of hops;
for *each hop number* **do**
  prints ocurrencies/total;
end

Algorithm 2: *number_ hops_ statistics*

5 Considerações finais