



UNIVERSIDADE DE AVEIRO

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E
INFORMÁTICA

47064- DESEMPENHO E DIMENSIONAMENTO DE REDES

Network Traffic Engineering

8240 - MESTRADO INTEGRADO EM ENGENHARIA DE
COMPUTADORES E TELEMÁTICA

António Rafael da
Costa Ferreira
NMec: 67405

Rodrigo Lopes
da Cunha
NMec: 67800

Docentes: Paulo Salvador,
Susana Sargento

Maio de 2016
2015-2016

Conteúdos

1	Exercício 1	2
2	Exercício 2, 3 e 4	2
3	Exercício 5, 6 e 7	4
4	Exercício 8, 9 e 10	5

1 Exercício 1

Para iniciar este trabalho foi-nos dado vários ficheiros de base. O ficheiro NetGen.py é responsável por gerar uma rede, com vários nós que inclui o nome e localização geográfica, e as ligações inter-nó, matrizes de tráfego que define os fluxos de tráfego entre todas as cidades/ nós.

A localização geográfica (pos) é obtida dinamicamente a partir do nome do nó (nome da cidade) usando a API do Google Maps, ficheiro getGeo.py.

A matriz de tráfego (TM) é gerado aleatoriamente. É possível também guardar o resultado num ficheiro .dat e passando como argumento o parâmetro -f, net.dat.



Figura 1: Rede pequena de teste e rede grande

2 Exercício 2, 3 e 4

No exercício 2 é usado o caminho mais curto como escolha para o caminho entre pontos, sendo que isto é dado pela soma das conexões e usando o algoritmo Greedy.

No exercício 3, foi calculado o "average one-way delay" e a carga em todas as direções em todos os links.

Para isso, para calcular o "average one-way delay", foi usada a seguinte fórmula baseada na aproximação Kleinrock:

$$\mu = 1e9 / 8000, \text{ é igual ao link speed em pkts/sec (1Gbps)}$$

$$W = 1e6 \times \left[\frac{1}{(\mu - \text{atraso})} \right]$$

Para calcular o atraso, teve de se criar um ciclo de forma a percorrer todos os links e criar uma lista com os atrasos.

```
for pair in allpairs:
    path = sol[pair]
    for i in range(0, len(path) - 1):
```

$$ws_delay[pair] = 1e6 / (\mu - \backslash \\ net[path[i]][path[i + 1]]['load '])$$

Após isso foi possível apresentar a seguinte tabela para a rede pequena:

Origem	Destino	Saltos	Carga (pkts/sec)	Atraso (micro/sec)
Lisboa	Viseu	Lisboa, Viseu	31271	10.67
Porto	Lisboa	Porto, Aveiro, Lisboa	Indisponível	15.95
Viseu	Porto	Viseu, Porto	31401	10.68
Lisboa	Aveiro	Lisboa, Aveiro	62675	16.04
Aveiro	Viseu	Aveiro, Viseu	31199	10.66
Viseu	Aveiro	Viseu, Aveiro	31378	10.68
Aveiro	Porto	Aveiro, Porto	63050	16.14
Porto	Viseu	Porto, Viseu	31396	10.68
Porto	Aveiro	Porto, Aveiro	62129	15.91
Lisboa	Porto	Lisboa, Aveiro, Porto	Indisponível	16.14
Viseu	Lisboa	Viseu, Lisboa	31171	10.66
Aveiro	Lisboa	Aveiro, Lisboa	62304	15.95

Tabela 1: Solução obtida, carga nos links e atraso

Analisando a tabela obtida conseguimos perceber que o link Aveiro-Porto tem o máximo delay de 16.14 micro segundos e a carga máxima está de Aveiro ao Porto com 63050 pacotes/ segundo.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Aveiro-Porto	16.1420500404	13.3475508848

Tabela 2: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Aveiro-Porto	63050.00 pkts/sec	43797.40 pkts/sec

Tabela 3: Carga

3 Exercício 5, 6 e 7

No exercício 5 foi pedido que o routing agora fosse feito usando como parâmetro de escolha a menor carga possível, otimizando assim a largura de banda disponível ao longo do caminho.

A nível de código, a única mudança efetuada importante foi:

```
path = nx.shortest_path(net, pair[0], pair[1], weight='load')
```

Pois agora o critério para escolha do caminho mais curto é a carga.

Para a rede pequena o resultado obtido foi:

Origem	Destino	Saltos	Carga (pkts/sec)	Atraso (micro/sec)
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Viseu, Lisboa	Indisponível	15.95
Viseu	Porto	Viseu, Porto	62731	16.06
Lisboa	Aveiro	Lisboa, Aveiro	31345	10.68
Aveiro	Viseu	Aveiro, Viseu	31199	10.66
Viseu	Aveiro	Viseu, Aveiro	31378	10.68
Aveiro	Porto	Aveiro, Porto	31720	10.72
Porto	Viseu	Porto, Viseu	62512	16.00
Porto	Aveiro	Porto, Aveiro	31013	10.64
Lisboa	Porto	Lisboa, Viseu, Porto	Indisponível	16.06
Viseu	Lisboa	Viseu, Lisboa	62287	15.95
Aveiro	Lisboa	Aveiro, Lisboa	31188	10.66

Tabela 4: Solução obtida, carga nos links e atraso

Analisando a tabela e os resultados obtidos conseguimos perceber que houve mudança na distribuição dos caminhos mais curtos, por exemplo, Porto > Lisboa agora o caminho é feito por Porto, Viseu e Lisboa, já de Lisboa > Porto é feito por Lisboa, Viseu, Porto.

Este resultado é explicado porque a lista de pares da rede pequena está distribuída da seguinte forma:

```
[('Lisboa', 'Viseu'), ('Lisboa', 'Aveiro'),
 ('Lisboa', 'Porto'), ('Viseu', 'Lisboa'), ('Viseu', 'Aveiro'),
 ('Viseu', 'Porto'), ('Aveiro', 'Lisboa'), ('Aveiro', 'Viseu'),
 ('Aveiro', 'Porto'), ('Porto', 'Lisboa'), ('Porto', 'Viseu'),
 ('Porto', 'Aveiro')]
```

Inicialmente, irá testar a ligação Lisboa > Viseu, à qual será atribuída, depois Lisboa > Aveiro e será também atribuída, de seguida, Lisboa > Porto, como o somatório das cargas entre as ligações Lisboa > Viseu (31271) + Viseu > Porto (0) tem menos carga do que a Lisboa > Aveiro (31345) + Aveiro

> Porto (0) então seleciona ir por Viseu, daí a diferença nos resultados em relação ao primeiro exercício.

Na diferença de Porto > Lisboa, temos Porto > Aveiro (0) + Aveiro > Lisboa (31188) com mais carga do que Porto > Viseu (0) + Viseu > Lisboa (31171) no momento de atribuição, o que faz com que seja selecionado o caminho: Porto > Viseu > Lisboa.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Viseu-Porto	16.0593553775	13.3398664587

Tabela 5: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Viseu-Porto	62731.00 pkts/sec	43797.40 pkts/sec

Tabela 6: Carga

Com este exercício, mudando o critério para carga nos links na escolha do caminho mais curto, conseguimos melhorias no "maximum one-way delay" (-0,083 microseg) de e no "maximum one-way load" (-319 pktssec).

4 Exercício 8, 9 e 10

No exercício 8 pretende-se que seja escolhido o caminho mais curto tendo em conta o atraso, minimizando assim o "average one-way delay". Este por sua vez irá sempre escolher de forma sequencial, o que fará com que seja sempre na mesma ordem.

A nível de código, a única mudança efetuada importante foi:

```
path = nx.shortest_path(net, pair[0], pair[1],
                        weight='delay')
...
net[path[i]][path[i + 1]]['delay'] = 1e6 /
(mu - net[path[i]][path[i + 1]]['load'])
```

O critério agora é o atraso, e para calcular o atraso usou-se a aproximação $M/M/1$.

Para a rede pequena o resultado obtido foi:

Origem	Destino	Saltos	Carga (pkts/sec)	Atraso (micro/sec)
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Viseu, Lisboa	Indisponível	15.95
Viseu	Porto	Viseu, Porto	62731	16.06
Lisboa	Aveiro	Lisboa, Aveiro	31345	10.68
Aveiro	Viseu	Aveiro, Viseu	31199	10.66
Viseu	Aveiro	Viseu, Aveiro	31378	10.68
Aveiro	Porto	Aveiro, Porto	31720	10.72
Porto	Viseu	Porto, Viseu	62512	16.00
Porto	Aveiro	Porto, Aveiro	31013	10.64
Lisboa	Porto	Lisboa, Viseu, Porto	Indisponível	16.06
Viseu	Lisboa	Viseu, Lisboa	62287	15.95
Aveiro	Lisboa	Aveiro, Lisboa	31188	10.66

Tabela 7: Solução obtida, carga nos links e atraso

Nesta tabela em relação ao exercício anterior não houve mudanças, isto devido à ordem ser a mesma de atribuição.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Viseu-Porto	16.0593553775	13.3398664587

Tabela 8: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Viseu-Porto	62731.00 pkts/sec	43797.40 pkts/sec

Tabela 9: Carga

Como se pode observar, para a rede pequena manteve-se igual.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Portimao-Porto	11.1610879829	9.92986401539

Tabela 10: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Aveiro-Porto	11.16 pkts/sec	9.82 pkts/sec

Tabela 11: Carga