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

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 no (nome da cidade) usando a API do Google Maps, ficheiro getGeo.py.

A matriz de tráfego (TM) é gerado aleatóriamente. 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:

μ = 1e9 / 8000 , é igual ao link speed em pkts/sec (1Gbps)

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

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 - \
net[path[i]][path[i + 1]]['load'])
```

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

Origem	**Destino**	**Caminho**	**Carga (pkts/sec)**	**Atraso (micro/sec)**
Lisboa	Viseu	Lisboa, Viseu	31271	10.67
Porto	Lisboa	Porto, Aveiro, Lisboa	Indisponível	31.86
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	32.19
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 caminho Lisboa-Porto tem o máximo delay de 32.19 micro segundos, carga média de 16.01 micro segundos, a carga máxima está de Aveiro ao Porto com 63050 pacotes/ segundo e carga média de 43797.40 pacotes/ segundo.

Maximum one-way delay flow	**Maximum one-way delay**	**Mean one-way delay**
Lisboa-Porto	32.1869758326	16.010093405

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

Para a rede grande, obteve-se o máximo delay de 83.61 micro segundos para o caminho entre Granada e a Corunha, atraso médio de 30.44 micro segundos, e a carga máxima de Aveiro-Porto com 63497 pacotes/ segundo e a carga média de 22735.10 pacotes/segundo.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Granada-Corunha	83.6110516972	30.439503777

Tabela 4: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Aveiro-Porto	63497.00 pkts/sec	22735.10 pkts/sec

Tabela 5: Carga

3 Exercício 5, 6 e 7

No exercício 5 foi pedido que o routing agora fosse feito usando como parâmetro de escolha do caminho mais curto 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	Caminho	Carga (pkts/sec)	Atraso (micro/sec)
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Viseu, Lisboa	Indisponível	31.95
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	32.09
Viseu	Lisboa	Viseu, Lisboa	62287	15.95
Aveiro	Lisboa	Aveiro, Lisboa	31188	10.66

Tabela 6: 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**
Lisboa-Porto	32.0852532284	16.0089473288

Tabela 7: Atraso

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

Tabela 8: Carga

Com este exercício, mudando o critério para carga nos links na escolha do caminho mais curto, para a rede pequena, conseguiu-se melhorias no "maximum one-way delay" de -0,102 micro segundos, no atraso médio de -0,0012 micro segundos, no "maximum one-way load" de -319 pkts/sec e a carga média manteve-se como esperado.

Dado que agora o pretendido era querer-se melhorias em termos de carga, estas foram obtidas para a rede pequena.

Para a rede grande, obteve-se o máximo delay de 80.19 micro segundos para o caminho entre Valencia e Badajoz, o que significa uma melhoria de 3,42 micro segundos, já atraso médio de desceu 1,80 micro segundos em relação ao exercício anterior, e obteve-se a carga máxima de Aveiro-Porto com 35315 pacotes/ segundo (melhoria de 28182 pacotes/segundo) e a carga média de 23310.10 pacotes/segundo (melhoria de 575,1 pacotes/segundo).

Maximum one-way delay flow	**Maximum one-way delay**	**Mean one-way delay**
Valencia-Badajoz	80.1875239088	28.6383060926

Tabela 9: Atraso

Max load flow	**Maximum one-way load**	**Mean one-way load**
Aveiro-Porto	35315.00 pkts/sec	23310.48 pkts/sec

Tabela 10: Carga

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	Caminho	Carga (pkts/sec)	Atraso (micro/sec)
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Viseu, Lisboa	Indisponível	31.95
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	32.09
Viseu	Lisboa	Viseu, Lisboa	62287	15.95
Aveiro	Lisboa	Aveiro, Lisboa	31188	10.66

Tabela 11: 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
Lisboa-Porto	32.0852532284	16.0089473288

Tabela 12: Atraso

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

Tabela 13: Carga

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

Já para a rede grande, como o critério agora é o atraso, espera-se que se tenha obtido melhorias, comparando com o exercício anterior, existiu melhorias, no máximo delay de 0,97 micro segundos para o mesmo caminho do exercício anterior, já o atraso médio desceu também 0,72 micro segundos e obteve-se a carga máxima também de Aveiro ao Porto, piorando em +88 pacotes por segundo sendo que a carga média diminuiu em 557,67 pacotes/segundo.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Valencia-Badajoz	79.2209331482	27.922949817

Tabela 14: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Aveiro-Porto	35403.00 pkts/sec	22752.81 pkts/sec

Tabela 15: Carga

5 Exercício 11 e 12

No exercício 11 e 12, foi pedido que usando um método que aleatoriamente trocasse os elementos de uma lista e fazendo isto para um número de vezes finito, de forma a obter sempre os melhores resultados tendo em conta o atraso. Para isto, foi usado o código do exercício anterior, adicionado um for para iterar e mais umas pequenas mudanças.

```
# antes de iterar e preciso declarar os dicionarios,
# listas e rede melhores
allpairs_best = []
sol_best = {}
ws_delay_best = {}
liststats_result = None
net_best = nx.DiGraph()

# iterar 10000x
for i in range(0, 10000):
        ... # igual ao exercicio anterior

        # ver se e a melhor solucao
        tmp_stats = listStats(ws_delay)

    # best solution
    if liststats_result is None or
            liststats_result[0] > tmp_stats[0]:
        allpairs_best = allpairs
        sol_best = sol.copy()
        ws_delay_best = ws_delay.copy()
        liststats_result = tmp_stats
        net_best = net_tmp.copy()
```

Para a rede pequena o resultado obtido foi:

Origem	Destino	Caminho	Carga (pkts/sec)	Atraso (micro/sec)
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Aveiro, Lisboa	Indisponível	31.86
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	31396	10.68
Porto	Aveiro	Porto, Aveiro	62129	15.91
Lisboa	Porto	Lisboa, Viseu, Porto	Indisponível	32.09
Viseu	Lisboa	Viseu, Lisboa	31171	10.66
Aveiro	Lisboa	Aveiro, Lisboa	62304	15.95

Tabela 16: Solução obtida, carga nos links e atraso

Em relação aos exercícios anteriores, esta solução é diferente. Dado que agora a lista é re-ordenada de forma aleatória 10 mil vezes, possivelmente terão sido testadas todas as possibilidades para a rede pequena.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Lisboa-Porto	32.0852532284	15.9968868727

Tabela 17: Atraso

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

Tabela 18: Carga

Analisando os resultados obtidos, em relação ao exercício anterior, conseguiu-se uma melhoria no atraso médio de -0,012 micro segundos. Para os restantes parâmetros, os resultados foram iguais.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Vigo-Granada	60.0122049529	26.934401214

Tabela 19: Atraso

As melhorias também apareceram na rede grande, como agora se fez muitas repetições e pretende-se melhorar o atraso médio, comparando com o

Max load flow	Maximum one-way load	Mean one-way load
Vilar.Formoso-Braganca	30905.00 pkts/sec	22172.22 pkts/sec

Tabela 20: Carga

exercício anterior, existiu melhorias, no máximo delay de 19,21 micro segundos para o mesmo caminho do exercício anterior, já o atraso médio desceu também 0,99 micro segundos e obteve-se a carga máxima também de Aveiro ao Porto, melhorando em -4498 pacotes por segundo, sendo que a carga média diminuiu também em 580,59 pacotes/segundo.

6 Exercício 13

Neste exercício, é pedido que se implemente um algoritmo de procura local de forma a obter a melhor solução minimizando o atraso médio da rede.

Pretende-se que usando o código desenvolvido anteriormente se encontre uma solução inicial que depois será usada pelo algoritmo de procura local para encontrar uma melhor solução, sempre que essa solução encontrada for melhor do que a solução melhor atual esta será substituída pela nova solução.

Portanto o código inicial é igual ao do exercício 8, depois é guardada esta solução que será usada como referência pelo algoritmo de procura local.

```
...

# antes de iterar e preciso declarar os dicionarios,
# listas e rede melhores encontradas
allpairs_best = []
sol_best = {}
ws_delay_best = {}
liststats_result = None
net_best = nx.DiGraph()

# iterar pela lista de pares
for pair in allpairs:
        # copiar a solucao e a net encontrada anteriormente
        sol_tmp = sol.copy()
    net_tmp = net.copy()

    # coloca-se o dicionario do atraso vazio
    ws_delay = {}

        # caminho para o par selecionado
```

```
path = sol_tmp[pair]
# apagar da solucao o par selecionado
del sol_tmp[pair]

# para o caminho para o par selecionado
for i in range(0, len(path) - 1):
    # remover a carga
    net_tmp[path[i]][path[i + 1]]['load'] -= tm[pair[0]][pair[1]]
    # recalcular o atraso
    net_tmp[path[i]][path[i + 1]]['delay'] = 1e6 / (mu - \
            net_tmp[path[i]][path[i + 1]]['load'])

    # faz-se a procura do caminho mais pequeno de novo
    path = nx.shortest_path(net_tmp, pair[0], pair[1],
                                        weight='delay')
    # grava-se esse caminho na nova solucao
    sol_tmp.update({pair: path})

    ....
    # repoe-se a carga para o caminho novo encontrado
    # recalcula-se o atraso

    # calula-se os atrasos (ws_delay)

    # verifica-se se e a melhor solucao e se sim, substitui-se
```

Para a rede pequena o resultado obtido foi:

Origem	**Destino**	**Caminho**	**Carga (pkts/sec)**	**Atraso (micro/sec)**
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Aveiro, Lisboa	Indisponível	31.86
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	31396	10.68
Porto	Aveiro	Porto, Aveiro	62129	15.91
Lisboa	Porto	Lisboa, Viseu, Porto	Indisponível	32.09
Viseu	Lisboa	Viseu, Lisboa	31171	10.66
Aveiro	Lisboa	Aveiro, Lisboa	62304	15.95

Tabela 21: Solução obtida, carga nos links e atraso

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Lisboa-Porto	32.0852532284	15.9968868727

Tabela 22: Atraso

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

Tabela 23: Carga

Analisando os resultados obtidos, em relação ao exercício anterior, mantiveram-se todos os resultados.

Já para a rede grande, neste exercício piorou em todos os resultados em relação ao exercício anterior, isto deve-se pelo que foi explicado anteriormente para a análise da rede pequena. Com a melhoria introduzida no próximo exercício espera-se que os resultados sejam os melhores encontrados.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Palma-Vigo	60.3428511256	27.6466866517

Tabela 24: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Aveiro-Porto	35403.00 pkts/sec	22573.59 pkts/sec

Tabela 25: Carga

7 Exercício 14

Para este exercício, pretende-se fazer 10 mil repetições do exercício anterior, fazendo-se permutações da lista de pares tal como o exercício 8, 9 e 10 e fazendo também procura local como o exercício anterior, procurando sempre a melhor solução.

Para a rede pequena o resultado obtido foi:

Origem	**Destino**	**Caminho**	**Carga (pkts/sec)**	**Atraso (micro/sec)**
Lisboa	Viseu	Lisboa, Viseu	62601	16.03
Porto	Lisboa	Porto, Aveiro, Lisboa	Indisponível	31.86
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	31396	10.68
Porto	Aveiro	Porto, Aveiro	62129	15.91
Lisboa	Porto	Lisboa, Viseu, Porto	Indisponível	32.09
Viseu	Lisboa	Viseu, Lisboa	31171	10.66
Aveiro	Lisboa	Aveiro, Lisboa	62304	15.95

Tabela 26: Solução obtida, carga nos links e atraso

Maximum one-way delay flow	**Maximum one-way delay**	**Mean one-way delay**
Lisboa-Porto	32.0852532284	15.9968868727

Tabela 27: Atraso

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

Tabela 28: Carga

Analisando os resultados obtidos, em relação ao exercício anterior, mantiveram-se todos os resultados.

Já para a rede grande, em comparação com o exercício 11 e 12, piorou o atraso máximo em 0.124 micro segundos e a carga máxima em 2988 pacotes/segundo. Já o atraso médio e a carga média foi a melhor conseguido entre todos os exercícios, em comparação também, foi uma melhoria 0.072 micro segundos e para a carga média de 80,93 pacotes/segundos.

Maximum one-way delay flow	Maximum one-way delay	Mean one-way delay
Granada-Vigo	60.1357390951	26.8627989428

Tabela 29: Atraso

Max load flow	Maximum one-way load	Mean one-way load
Vilar.Formoso-Braganca	33893.00 pkts/sec	22091.29 pkts/sec

Tabela 30: Carga