

Trabalho de Aprofundamento 2

UA

Miguel Cabral, David Bicho



Trabalho de Aprofundamento 2

Departamento de Eletrónica, Telecomunicações e
Informática

UA

Miguel Cabral, David Bicho
(93091) miguel.f.cabral@ua.pt, (93215) david23@ua.pt

Abril de 2019

Agradecimentos

Neste trabalho queríamos agradecer aos docentes responsáveis pela **uc!** (**uc!**) respetiva de cada aluno deste grupo (A professora Pétia Georgieva e o Professor António Adrego) que lecionaram as aulas deste 2º semestre.

Conteúdo

Capítulo 1

Introdução

Este trabalho foi realizado no âmbito da disciplina de Laboratórios de Informática. Este trabalho consiste em criar um programa cliente em python que fosse capaz de avaliar a qualidade de ligação ,especialmente latência e largura de banda de Download e Upload.

O cliente irá aceder aos servidores que estão num ficheiro **JSON! (JSON!)** que foi fornecido pelos docentes.

O cliente deverá efetuar testes periódicos usando múltiplos servidores. Ao longo da sua execução, deverá emitir um relatório, em formato **CSV! (CSV!)**, contendo os dados obtidos (report.csv).

Os servidores utilizam o protocolo que opera sobre ligações **TCP! (TCP!)**,efetuadas para o endereço e porta especificados no ficheiro servers.json.

As mensagens trocadas são simples mensagens de texto terminadas pelo caractere'n'.

Este documento está dividido em quatro capítulos. Depois desta introdução, no ?? é apresentada a metodologia seguida, no ?? são apresentados os resultados obtidos, sendo estes discutidos no ??. Finalmente, no ?? são apresentadas as conclusões do trabalho.

Capítulo 2

Metodologia

Descrição da forma de realização do trabalho e apresentação dos resultados.

2.1 client.py

2.1.1 Funcionalidades

O programa client.py tem como função de comunicar com o servidor através da leitura do ficheiro **JSON!** e criar um ficheiro **CSV!** com os resultados.

2.1.2 Função valid()

Esta função tem como principal objetivo verificar se os argumentos introduzidos no terminal são corretos. Esta função vai realizar os seguintes passos:

- Verificar se o cliente inicia o programa com um número de argumentos inferior a 3, se tal acontecer imprime no terminal uma mensagem.
- Verifica se é iniciado com argumentos inválidos (tipo errado, valor incorreto ou não encontrado),e apresenta uma mensagem de erro respetiva ao argumento que gera o erro.

2.1.3 host()

Esta função tem como objetivo:

- Verificar se o 3 argumento introduzido é texto e se assim o for realizar o teste para qualquer servidor do país introduzido.
- Caso o 3 argumento for um número inteiro ,deverá realizar o teste para o servidor com esse identificador.

2.1.4 **main()**

Esta função realiza o seguinte:

- O primeiro argumento introduzido deverá ser um número inteiro que irá indicar o intervalo em segundos entre cada teste.
- O segundo argumento deverá ser um número inteiro que irá indicar o número de testes que irão ser realizados.

2.1.5 **download()**

Esta função realiza o seguinte:

- O cliente realiza uma descarga entre 10MB e 100MB ou até que passem 10 segundos.
- A taxa de largura de banda é calculada através do número de octetos recebidos sobre o tempo decorrido, após ter sido obtido 1MB.

2.1.6 **Ping()**

Esta função realiza o seguinte:

- A latência deve ser calculada pelo tempo médio de 10 transações de PING/PONG.

2.1.7 **readJSON()**

Esta função realiza o seguinte:

- Recebe o host e retorna o id desse host.

2.1.8 **Main()**

A Função Main realiza o seguinte:

- Abre o ficheiro **CSV!** e, enquanto o número de testes for inferior ao argumento introduzido, escreve neste os resultados obtidos.

Capítulo 3

Resultados

Neste capítulo vamos mostrar alguns testes que foram realizados com erros induzidos por nós ,para mostrar o resultado no terminal esperado.

3.1 Erro de argumentos

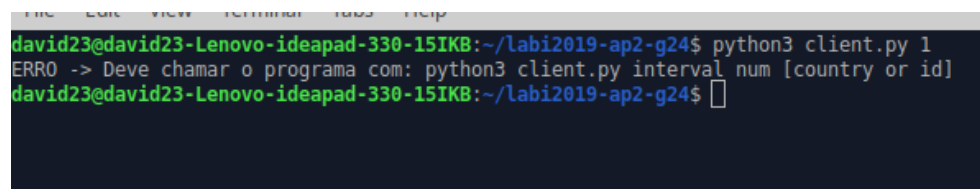
Quando o número de argumentos não é o correto ou não é adequado o resultado no terminal é o mostrado na figura 3.1 :

3.2 Comparação de Ping

Nas figuras 3.2 podemos comparar o PING em Portugal e na China.O resultado é o esperado visto que o PING é muito superior na China do que em Portugal,visto que o servidor está mais distante.

3.3 País ou Id inválido

Quando o país ou o Id introduzido como argumento é inválido o resultado no Output é o mostrado na figura 3.3 :

A terminal window screenshot showing a command prompt on a Linux system. The prompt is 'david23@david23-Lenovo-ideapad-330-15IKB:~/labi2019-ap2-g24\$'. The user has entered 'python3 client.py 1'. The terminal output shows an error message: 'ERRO -> Deve chamar o programa com: python3 client.py interval num [country or id]'. The prompt is now 'david23@david23-Lenovo-ideapad-330-15IKB:~/labi2019-ap2-g24\$' followed by a cursor.

```
david23@david23-Lenovo-ideapad-330-15IKB:~/labi2019-ap2-g24$ python3 client.py 1
ERRO -> Deve chamar o programa com: python3 client.py interval num [country or id]
david23@david23-Lenovo-ideapad-330-15IKB:~/labi2019-ap2-g24$
```

Figura 3.1: Erro de argumentos


```
miguel@miguel-Lenovo-ideapad-300-15ISK:~/labi2019-ap2-g24$ python3 client.py 3 1 China
Download -> 0.05604841780131421 MB
Ping -> 359.1708984375 ms
Teste realizado com sucesso, faltam 0 para terminar

miguel@miguel-Lenovo-ideapad-300-15ISK:~/labi2019-ap2-g24$ python3 client.py 3 1 Portugal
Download -> 10.980191602623822 MB
Ping -> 20.27685546875 ms
Teste realizado com sucesso, faltam 0 para terminar
```

Figura 3.2: Comparação de Ping

```
miguel@miguel-Lenovo-ideapad-300-15ISK:~/labi2019-ap2-g24$ python3 client.py 3 1
12345
id inexistente
miguel@miguel-Lenovo-ideapad-300-15ISK:~/labi2019-ap2-g24$ python3 client.py 1 1
Gronelândia
País inexistente
```

Figura 3.3: figura 3.4

3.4 Report.csv

Quando todos os argumentos introduzidos são os esperados ,é escrito os valores do Contador,id do servidor,data e hora no formato ISO,latência,largura de banda,check,como mostra a figura 3.4 :

```
contador,id,data,latencia,largura de banda,check
1,5874,2019-04-26T20:59:37.864962,29.766357421875,2.50015820310015,158742019-04-26T20:59:37.88641829.7663574218752.50015820310015
2,9729,2019-04-26T21:01:18.696623,13.004638671875,0.586375813934255,297292019-04-26T21:01:18.71728813.0046386718750.586375813934255
3,10363,2019-04-26T21:02:55.347890,14.166748046875,0.3987953494541553,3103632019-04-26T21:02:55.37024614.1667480468750.3987953494541553
4,8160,2019-04-26T21:03:01.383716,11.92236328125,10.756349848760852,481602019-04-26T21:03:01.40453511.9223632812510.756349848760852
5,4440,2019-04-26T21:04:40.315721,11.092041015625,0.4216440583652695,544402019-04-26T21:04:40.33719511.0920410156250.4216440583652695
```

Figura 3.4: figura 3.4

Capítulo 4

Conclusões

Este trabalho permitiu ao grupo aprofundar os conhecimentos sobre python e sobre a implementação de sockets e manipulação de ficheiros **JSON!**.

Contribuições dos autores

David Bicho 50% Miguel Cabral 50%

Acrónimos

CSV Comma Separated Values

TCP Transmission Control Protocol

JSON JavaScript Object Notation

uc Unidade Curricular