

## Tarefa de Programação

Implementação de um cliente PING usando protocolo UDP, simulando perdas de pacotes. A Tecnologia utilizada para desenvolver foi Python, uma linguagem de programação multi paradigma multi plataforma de sintaxe simples e poderosa.

Para realizar a execução do projeto, primeiramente precisa rodar o servidor feito em Java ja disponibilizado. O servidor obrigatoriamente precisa estar em uma maquina (ou na mesma maquina) visível do computador que está executando o cliente.

O projeto não necessita de nenhuma biblioteca adicional, somente é necessário que a maquina possuir python instalado. Para saber se existe o Python instalado basta ir no terminal (linha de comando) e executar o comando "Python". Se abrir o terminal abaixo é porque está tudo certo.

```
Python 2.7.3 (default, Apr 20 2012, 22:39:59)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Existem dois modos de usar o progrma. O primeiro contempla a resolução do exercicio e os execicios opcionais 1 e 2. A segunda é para executar o execicio opcional 3.

Modo de execução 1:

```
python PingClient.py [ip servidor] [porta servidor]
```

Ex:

```
$ python PingClient.py 127.0.0.1 50000
```

Nesse modo ele vai realizar o envio de 10 pacotes para o servidor (resolução do exercicio). Mostrando uma analize dos dados, tempos de envio e perda de pacotes (opcional 1). Em um intervalo de 1 seg. entre o envio de cada pacote, sendo que cada pacote se demorar mais que 1 seg. será considerado perdido (opcional 2).

Ex:

```
$ python PingClient.py 127.0.0.1 50000
```

A Conexao nao foi integra

Sequencia 0 perdida

Sequencia 9 perdida

##### - Dados finais - #####

Sequencia 1 com 179 milisegundos

Sequencia 2 com 68 milisegundos

Sequencia 3 com 133 milisegundos

Sequencia 4 com 64 milisegundos

Sequencia 5 com 96 milisegundos

Sequencia 6 com 86 milisegundos

Sequencia 7 com 196 milisegundos

Sequencia 8 com 167 milisegundos

Maior tempo:  
Sequencia 7 com 196 milisegundos

Menor tempo:  
Sequencia 4 com 64 milisegundos

Media de tempo:  
Media em milisegundos ~ 123.63

Modo de execução 2:

python PingClient.py [ip servidor] [porta servidor] [opcao]

Ex:

\$ python PingClient.py 127.0.0.1 50000 -d

Neste modo o sistema contempla o exercicio opcional 3, que consiste em enviar dados para o servidor e receber, caso ouiver perdas ele vai reenviar os dados faltantes até que todos os dados sejam recebidos pelo cliente.

\$ python PingClient.py 127.0.0.1 50000 -d  
foram enviados 13 pacotes

##### - Dados finais - #####

Sequencia 1 com 118 milisegundos  
Sequencia 2 com 149 milisegundos  
Sequencia 3 com 143 milisegundos  
Sequencia 4 com 86 milisegundos  
Sequencia 5 com 133 milisegundos  
Sequencia 6 com 139 milisegundos  
Sequencia 7 com 92 milisegundos  
Sequencia 8 com 41 milisegundos  
Sequencia 9 com 4 milisegundos

Maior tempo:  
Sequencia 2 com 149 milisegundos

Menor tempo:  
Sequencia 9 com 4 milisegundos

Media de tempo:  
Media em milisegundos ~ 100.56

Informações adicionais.

Para estrutura de controle e armazenamento de pacotes foram utilizados dicionários (<http://docs.python.org/tutorial/datastructures.html>), pois a facilidade de trabalhar facilitou a resolução do problema.

Para recebimento dos dados do servidor foi implementado uma thread (<http://docs.python.org/library/threading.html>) de escuta, que valida e somente aceita pacotes recebidos dos

servidor:porta informada por parametro ao iniciar o programa.

Mais informações sobre a linguagem python pode ser encontradas no site <http://www.python.org/>, a documentação do funcionamentos dos sockets pode ser encontrada no site <http://docs.python.org/library/socket.html>. Informações sobre instalação da linguagem python pode ser encontrada em <http://www.python.org/download/>.