MOPS Power Control

User’s guide

Contents

1 Introduction

1.1 What is MOPS Power Control?

MOPS Power Control is an application that adjusts the transmitting power of Base Transceiver Station (BTS) and Mobile Station (MS) based on measurement reports of BTS and MS.

1.2 Purposes of the MOPS Power Control

Main purposes of MOPS Power Control application are saving power of MS, BTS and also increasing the quality of the network connection by reducing the interference of the network.

1.3 Features

MOPS Power Control offers some extra features, allowing the user to:

- Override default configuration (flag: ‘-c’)
- Turn on/off handover algorithm (flag: ‘-h’)
- Create log file for debugging purposes (flag: ‘-d’)
- Write measurements to database (flag: ‘-db’)
- HTTP communication
- Plotter

In order to use extra features, additional flags provided at terminal launching are required. See Section 3.5. "Additional features" for details

2 Installation of MOPS Power Control

Before installing MOPS Power Control make sure that your operating system is Linux

2.1 System requirements

For the application to work correctly you should have Python 3.6.1 interpreter installed on your machine and Linux operating system.

2.2 Where to get MOPS Power Control?

https://bitbucket.org/pat049b/mops_power_control

2.3 Installation instructions

Download all the files from link provided in previous section. Unzip them all into one directory, shell command for unzipping:

```
unzip <file name>.zip -d <directory>
```

If unzip library is not installed on your system, run:

```
sudo apt-get install unzip.
```

Files being in another directory may cause necessity to provide absolute path to them.
No installation is required, script is interpreted by shell built-in interpreter. Program can be run from terminal.

2.3.1 Required libraries

- **`Matplotlib`** - library is required for grapher.py to draw charts. Installation can be run by:
  `pip-python3 matplotlib`
  or
  `apt-get install python3-matplotlib`
- **`Requests`** - required for http server. Installation instruction on: http://docs.python-requests.org/en/master/

3 User interface

3.1 Introduction

By now you should have MOPS Power Control application ready to work on your computer. In the next chapters we will explore:

- How to use basic MOPS Power Control features.
- How to change configuration of MOPS Power Control to set different rules for working algorithm

3.2 Start MOPS Power Control

You can start MOPS Power Control application from your shell using command:

```
cat input.txt | python pc.py
```

TIP

- When starting MOPS Power Control it is possible to specify optional settings using a command line. See Section 3.5. "Additional features" for details

3.3 Input

Algorithm takes input through stdin. Required input line is structured this way:

```
XL   XX   XXXX num1 num2
```

Where:
XL - indicates direction of transmission (UL or DL)
XX - BTS number (S0 for current cell, N1-6 for neighbour)
XXXX - Mobile station name (any string without special signs)
Num1 - power level
Num2 - quality level

3.3.1 Configuration

Algorithm has default configuration parameters:

```
# target:-75    --> Setting target power in dBm
# hister:3      --> histeresy threshold
# maxInc:8      --> Maximum power increase
# maxIncHist:1  --> Maximum power increase inside hysteresis area
# maxDec:4      --> Maximum power decrease
# maxDecHist:1  --> Maximum power decrease inside hysteresis area
# changeThresh:1--> Threshold of change
# maxMissing:3  --> Maximum number of missing signals before\ launching
MaxPower mode
# window:8      --> Number of measurements included in calculations
# offset:3      --> Minimum difference between current cell power\ and
neighbour
# minAmount:4   --> Minimum amount of measurements to start PC
```

These settings can be customized. Customization process is described in section 3.5. "Additional Features".

3.4 Output

There are few options for getting an output. Script output is printed by default in standard output in a terminal:

```
>>> cat input.txt | python3 pc.py
```

```
Loading default configuration
DL    S0      MS222    NCH
UL    S0      MS222    NCH
```

3.4.1 Printing the output to a file

Commands are always printed in terminal, they can be also redirected to a file by using
`> file_name` at the end of a terminal command:

```
>>> cat input.txt | python pc.py > output.txt
```

3.4.2 Database

Flag: ‘-db’ is required to send data to database
Measurement data is stored in database directly, or through http server if one is set up. See section 3.5.5. “HTTP communication” for details.

3.4.3 Graphic output

There is a possibility to visualize measurement data for specified mobile station, See section 3.5.4. “Plotter” for details.

3.5 Additional features

3.5.1 User configuration

User is able to override default configuration of an algorithm. To do so, additional flag should be

Flag: **`-c`**

Invoke example:

```
cat input.txt | python3 pc.py -h -c
cat input.txt | python3 pc.py -c
```

Feature description: Overrides default configuration with one specified in conf.cfg file. Detailed information related to changing configurable parameters are included inside this file.

3.5.2 Handover

Handover algorithm:

Flag: **-h**

Invoke example:

```
cat input.txt | python3 pc.py -h -c
cat input.txt | python3 pc.py -h
```

Feature description: When neighbour cell measurement received, algorithm is comparing signal power of S0 cell with signal power of neighbour. Sends HOBC (Handover Better Cell) signal if handover is profitable.

3.5.3 Debugger

Flag: **-d**

Invoke example:

```
cat input.txt | python3 pc.py -d -h
cat input.txt | python3 pc.py -d
```

Feature description: Creates log for debugging purposes in `logdeb.txt`. Debugger appends information to this file. In order to wipe logs out, manual deletion of file is required. Example log from one input line:

```
Current input line  UL S0 MS776 -78 2
Line correct
Current power history [-78, -75, -70, -70, -78]
Current quality history ['2', '2', '1', '1', '2']
Consecutive missings 0
Enough data to take an action
Average power: -74.71
Average quality:1.61
Command sent: UL   S0 MS776  NCH
```

3.5.4 Plotter

Flag: none - plotter is a separate application, using data in database to visualize history of power levels downlink and uplink for certain mobile station in specified time window.

Invoke example:

```
python3 grapher.py '2017-07-20 20:00' '2017-07-21 22:00' 'MS111'
```

After name of script (**`grapher.py`**), two following dates are beginning and end of time window.
Date format: '**`YYYY-MM-DD HH:MiMi:SS.SsSsSs`**' where:

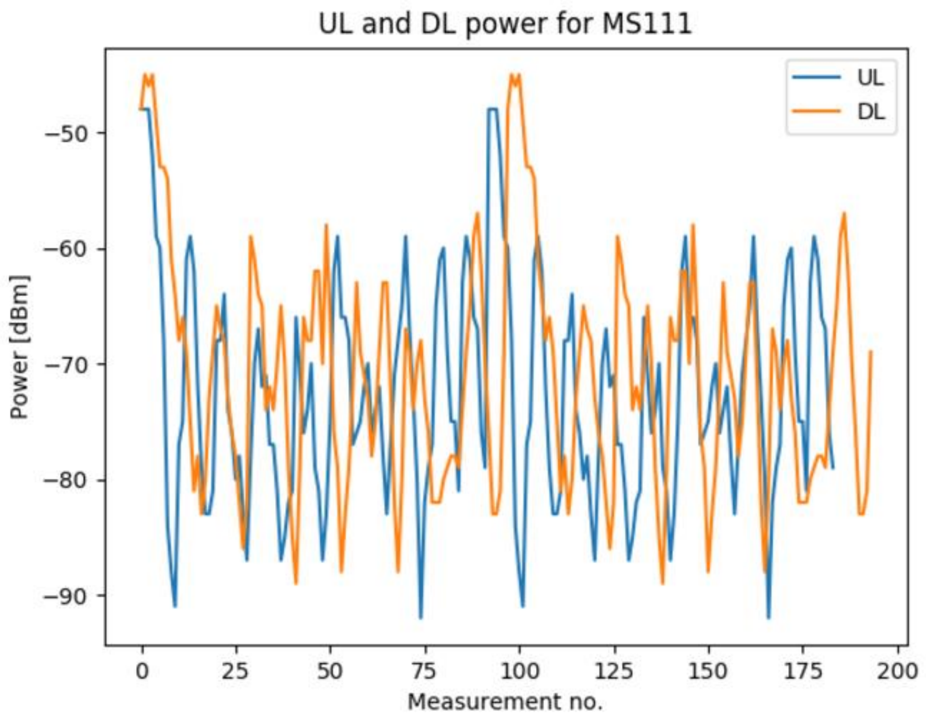
`Y` - year, **`M`** - month, **`D`** - day

`H` - hour, **`Mi`** - minute, **`S`** - seconds, **`Ss`** - parts of second

Provided dates accuracy does not matter.
The last parameter is a name of mobile station to be displayed.

Plotter uses **`matplotlib`** library, which must be downloaded previously. Installation on Ubuntu/Debian systems can be executed by command:
`sudo apt-get install python3-matplotlib` or
`pip-python3 matplotlib`

Example figure:



3.5.5 HTTP communication

To establish Http server user need to launch it before launching whole application.
Steps to do that:

1. Go to project directory
2. Launch **`http_server.py`** in python3 interpreter in separate console by command:

```
python3 http_server.py
```

3) Now your server is working in separate console.
4) Run your program in another console and communication with HTTP in server is established now. If you skip steps 1-3 your program is sending data to database directly.

It may happen that your environment **doesn't have requests lib** which is necessary for properly working HTTP communication. **To get requests lib** please find attached producent website: http://docs.python-requests.org/en/master/

3.5.6 Parser to analyze TCPDUMP

Flag: none - parser is a separate application, using data from tcpdump aplication to visualize history of communication with HTTP server.

Steps to start:

1. Start HTTP server - see 9.6 HTTP communication

2. Start a new terminal:

   Invoke example:

   1. `tcpdump -A -v -i <interface> > <file_name>`
   2. `python3 ana.py <file_name>`

4 Reporting problems

If you encounter any issues with MOPS Power Control, please report them to our project manager at: **patryk.bogusz@ust-global.com**