

# api

## API Documentation

September 11, 2019

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package command_ap</b>	<b>2</b>
1.1 Modules . . . . .	2
1.2 Variables . . . . .	2
<b>2 Package command_ap.cmd</b>	<b>3</b>
2.1 Modules . . . . .	3
2.2 Variables . . . . .	3
<b>3 Module command_ap.cmd.command_ap</b>	<b>4</b>
3.1 Functions . . . . .	4
3.2 Variables . . . . .	8
<b>4 Module command_ap.cmd.ifconfig</b>	<b>9</b>
4.1 Functions . . . . .	9
4.2 Variables . . . . .	9
<b>5 Module command_ap.cmd.iwconfig</b>	<b>10</b>
5.1 Functions . . . . .	10
5.2 Variables . . . . .	10
<b>6 Module command_ap.cmd.scan</b>	<b>11</b>
6.1 Functions . . . . .	11
6.2 Variables . . . . .	11
<b>7 Module command_ap.cmd.station</b>	<b>12</b>
7.1 Functions . . . . .	12
7.2 Variables . . . . .	13
<b>8 Module command_ap.cmd.survey</b>	<b>14</b>
8.1 Functions . . . . .	14
8.2 Variables . . . . .	14
<b>9 Module command_ap.cmd.xmit</b>	<b>15</b>
9.1 Functions . . . . .	15
9.2 Variables . . . . .	15

<b>10 Package command_ap.get_set</b>	<b>16</b>
10.1 Modules . . . . .	16
10.2 Variables . . . . .	16
<b>11 Module command_ap.get_set.client</b>	<b>17</b>
11.1 Variables . . . . .	17
<b>12 Module command_ap.get_set.server</b>	<b>18</b>
12.1 Functions . . . . .	18
12.2 Variables . . . . .	18
12.3 Class myHandler . . . . .	19
12.3.1 Methods . . . . .	19
<b>13 Module command_ap.get_set.server_ffox</b>	<b>24</b>
13.1 Functions . . . . .	24
13.2 Variables . . . . .	24
13.3 Class FirefoxDataMemory . . . . .	24
13.3.1 Methods . . . . .	25
13.3.2 Properties . . . . .	25
13.4 Class SrvPosts . . . . .	25
13.4.1 Methods . . . . .	25
<b>14 Module command_ap.get_set.teste</b>	<b>26</b>
14.1 Functions . . . . .	26
<b>15 Package command_ap.publisher_subscriber</b>	<b>27</b>
15.1 Modules . . . . .	27
15.2 Variables . . . . .	27
<b>16 Module command_ap.publisher_subscriber.publisher</b>	<b>28</b>
16.1 Variables . . . . .	28
<b>17 Module command_ap.publisher_subscriber.subscriber</b>	<b>29</b>
17.1 Variables . . . . .	29
<b>18 Package command_ap.rl</b>	<b>30</b>
18.1 Modules . . . . .	30
18.2 Variables . . . . .	30
<b>19 Module command_ap.rl.agent</b>	<b>31</b>
19.1 Functions . . . . .	31
19.2 Variables . . . . .	32
19.3 Class MABAgent . . . . .	33
19.3.1 Methods . . . . .	33
19.3.2 Properties . . . . .	34
<b>20 Module command_ap.rl.app1</b>	<b>35</b>
<b>21 Package command_ap.rl.basic</b>	<b>36</b>
21.1 Modules . . . . .	36
<b>22 Module command_ap.rl.basic.environment</b>	<b>37</b>
22.1 Class environment . . . . .	37

22.1.1	Methods . . . . .	37
22.1.2	Properties . . . . .	37
<b>23</b>	<b>Module <code>command_ap.rl.mab</code></b>	<b>38</b>
23.1	Functions . . . . .	38
23.2	Variables . . . . .	38
23.3	Class MAB . . . . .	38
23.3.1	Methods . . . . .	39
23.3.2	Properties . . . . .	40
23.4	Class RandomAbstract . . . . .	40
23.4.1	Methods . . . . .	40
23.4.2	Properties . . . . .	40
23.5	Class EpsilonGreedyAbstract . . . . .	41
23.5.1	Methods . . . . .	41
23.5.2	Properties . . . . .	42
23.6	Class UCBAbstract . . . . .	42
23.6.1	Methods . . . . .	42
23.6.2	Properties . . . . .	43
23.7	Class Boltzmann . . . . .	43
23.7.1	Methods . . . . .	43
23.7.2	Properties . . . . .	44
<b>24</b>	<b>Module <code>command_ap.rl.model</code></b>	<b>45</b>
24.1	Functions . . . . .	45
<b>25</b>	<b>Module <code>command_ap.rl.reward</code></b>	<b>46</b>
25.1	Functions . . . . .	46
25.2	Variables . . . . .	46
<b>26</b>	<b>Script <code>script-hostapd_conf</code></b>	<b>47</b>
26.1	Variables . . . . .	47
<b>27</b>	<b>Script <code>script-setup_cfg</code></b>	<b>48</b>
27.1	Functions . . . . .	48
27.2	Variables . . . . .	48

# 1 Package `command_ap`

## 1.1 Modules

- **cmd** (Section 2, p. 3)
  - **command\_ap** (Section 3, p. 4)
  - **ifconfig**: converts the output of ifconfig into a dictionary (Section 4, p. 9)
  - **iwconfig**: convert the output of iwconfig into a dictionary (Section 5, p. 10)
  - **scan**: convert the output of iw dev station dump into a dictionary (Section 6, p. 11)
  - **station**: convert the output of iw dev station dump into a dictionary (Section 7, p. 12)
  - **survey**: convert the output of iw dev station dump into a dictionary (Section 8, p. 14)
  - **xmit**: Module xmit (Section 9, p. 15)
- **get\_set** (Section 10, p. 16)
  - **client**: the server accepts requests from an http client. (Section 11, p. 17)
  - **server**: server that accepts requests from an http client used to send commands to the AP (Section 12, p. 18)
  - **server\_ffox**: The client (firefox) sends the following json data: (Section 13, p. 24)
  - **teste**: Test to get the data to compute: MOS client, hybrid and AP (Section 14, p. 26)
- **publisher\_subscriber** (Section 15, p. 27)
  - **publisher** (Section 16, p. 28)
  - **subscriber** (Section 17, p. 29)
- **rl** (Section 18, p. 30)
  - **agent**: runs the agent: python3 agent.py (Section 19, p. 31)
  - **app1** (Section 20, p. 35)
  - **basic** (Section 21, p. 36)
    - \* **environment** (Section 22, p. 37)
  - **mab**: This module define three abstract MAB agents: \* RandomAbstract: select random actions \* EpsilonGreedyAbstract: select action using an epsilon-greedy policy \* UCBAbstract: selects actions based on the UCB policy (Section 23, p. 38)
  - **model**: This module calculates the QoS based on the features (Section 24, p. 45)
  - **reward**: runs the agent: python3 agent.py (Section 25, p. 46)

## 1.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

## 2 Package command\_ap.cmd

### 2.1 Modules

- **command\_ap** (*Section 3, p. 4*)
- **ifconfig**: converts the output of ifconfig into a dictionary (*Section 4, p. 9*)
- **iwconfig**: convert the output of iwconfig into a dictionary (*Section 5, p. 10*)
- **scan**: convert the output of iw dev station dump into a dictionary (*Section 6, p. 11*)
- **station**: convert the output of iw dev station dump into a dictionary (*Section 7, p. 12*)
- **survey**: convert the output of iw dev station dump into a dictionary (*Section 8, p. 14*)
- **xmit**: Module xmit (*Section 9, p. 15*)

### 2.2 Variables

Name	Description
__package__	<b>Value:</b> None

### 3 Module `command_ap.cmd.command_ap`

#### 3.1 Functions

**get\_xmit**(*phy\_iface*='phy0')

get data from the xmit file. looks for it in `/sys/kernel/debug/ieee80211/ath*/xmit`

**Return Value**

the xmit fields

(*type=dict*)

**get\_ifconfig**(*interface*, *path\_ifconfig*=`__PATH_IFCONFIG`)

get data from ifconfig `<interface>`.

**Parameters**

**interface:** the wireless interface name, e.g. wlan0

**path\_ifconfig:** path to ifconfig

**Return Value**

the ifconfig fields

(*type=dict*)

**get\_iw\_stations**(*interface*, *path\_iw*=`__DEFAULT_IW_PATH`)

executes "iw station dump"

**Parameters**

**interface:** the wireless interface name, e.g. wlan0

**path\_iw:** path to iw

**Return Value**

the command fields

(*type=dict*)

**get\_status**(*path\_hostapd\_cli*=`__DEFAULT_HOSTAPD_CLI_PATH`)

get information from "hostapd\_cli status" TODO: what if the interface has multiple SSIDs  
???

**Parameters**

**path\_hostapd\_cli:** path to hostapd\_cli

**Return Value**

the returned command fields

(*type=dict*)

```
change_channel(interface, new_channel, count=1, ht_type=None,
path_hostapd_cli=__DEFAULT_HOSTAPD_CLI_PATH)
```

set the AP's channel using "hostapd\_cli chan\_switch" command.

TODO: add other optional parameters

```
[sec_channel_offset=] [center_freq1=] [center_freq2=] [bandwidth=] [blocktx]
```

@param *interface*: the wireless interface name, e.g. wlan0

@param *new\_channel*: the new channel number. Trying to change to the current channel returns an error

@param *ht\_type*: Valid values are ['', 'ht', 'vht']. Defines the type of channel. Invalid type returns an error

@param *path\_hostapd\_cli*: path to hostapd\_cli

@return: the ifconfig fields

@rtype: dict

```
get_stations(path_hostapd_cli=__DEFAULT_HOSTAPD_CLI_PATH)
```

returns information about all connected stations

#### Parameters

*path\_hostapd\_cli*: path to hostapd\_cli

#### Return Value

dictionary of dictionary

```
get_iw_info(interface, path_iw=__DEFAULT_IW_PATH)
```

executes "iw dev info"

#### Parameters

*interface*: the wireless interface name, e.g. wlan0

*path\_iw*: path to iw

#### Return Value

the command fields

(*type=dict*)

```
get_iwconfig_info(interface, path_iwconfig=__DEFAULT_IWCONFIG_PATH)
```

get the return from "iwconfig <interface>" NOTE: this method only supports (tested) two modes = Managed and Master

#### Parameters

*interface*: interface to change

*path\_iwconfig*: path to iwconfig

#### Return Value

the command fields

(*type=dict*)

---

```
get_power(interface, path_iw=__DEFAULT_IW_PATH,
path_iwconfig=__DEFAULT_IWCONFIG_PATH)
```

---

get the power in the interface (from a station or AP)

**Parameters**

**interface:** interface to change

**path\_iw:** path to iw

**Return Value**

the command fields

(*type=dict*)

---

```
set_iw_power(interface, new_power, path_iw=__DEFAULT_IW_PATH)
```

---

command dev <devname> set txpower <auto|fixed|limit> [<tx power in mBm>] NOTE: this module needs to run as superuser to set the power

**Parameters**

**interface:** interface to change

**new\_power:** can be a string 'auto', or a number (int or float) that represents the new power in dBm

**path\_iw:** path to iw

**Return Value**

if the command succeeded

---

```
disassociate_sta(mac_sta, path_hostapd_cli=__DEFAULT_HOSTAPD_CLI_PATH)
```

---

sends the command to disassociate a station

**Parameters**

**mac\_sta:** the MAC address of the station we want to disassociate

**Return Value**

if the command succeeded

(*type=bool*)

---

```
get_config(path_hostapd_cli=__DEFAULT_HOSTAPD_CLI_PATH)
```

---

executes "hostapd\_cli get\_config"

**Parameters**

**path\_hostapd\_cli:** path to hostapd\_cli

**Return Value**

dictionary {'ssid': 'ethanolQL1', 'bssid': 'b0:aa:ab:ab:ac:11',  
'rsn\_pairwise\_cipher': 'CCMP', 'group\_cipher': 'CCMP', 'key\_mgmt':  
'WPA-PSK', 'wpa': '2', 'wps\_state': 'disabled'}



---

**get\_iw\_survey**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

---

executes command "iw dev <interface> survey dump"

**Parameters**

**interface:** interface to change

**path\_iw:** path to iw

**Return Value**

decoded information from survey

---

**get\_scan**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

---

helper function that commands iw dev <interface> scan dump or scan ap-force. some APs only accept scan ap-force. used by get\_iw\_scan\_full(), get\_iw\_scan\_mac() and get\_iw\_scan().

**Parameters**

**interface:** interface to scan

**path\_iw:** path to iw

**Return Value**

return the output of the command

---

**get\_iw\_scan\_full**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

---

execute command "iw dev <interface> scan dump"

**Parameters**

**interface:** interface to change

**path\_iw:** path to iw

**Return Value**

decoded information from scan dump

---

**get\_iw\_scan\_mac**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

---

executes the command "iw dev <interface> scan dump"

**Parameters**

**interface:** interface to scan

**path\_iw:** path to iw

**Return Value**

decoded information from scan dump, only the detected MACs

---

**get\_iw\_scan**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

---

command dev <interface> scan dump

**Parameters**

**interface:** interface to scan

**path\_iw:** path to iw

**Return Value**

decoded information from scan dump, only the detected MACs

**trigger\_scan**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

command dev <interface> scan trigger it is necessary to call this method before call any method with 'scan', it forces the AP to scan all valid channels, and populate the statistics

**Parameters**

**interface:** interface to scan

**path\_iw:** path to iw

**Return Value**

nothing

**get\_phy\_with\_wlan**(*interface*, *path\_iw*=\_\_DEFAULT\_IW\_PATH)

**Parameters**

**interface:** the name of the interface, e.g. 'wlan0'

**Return Value**

a string with the phy interface name

## 3.2 Variables

Name	Description
LOG	<b>Value:</b> <code>logging.getLogger('CMD')</code>
valid_frequencies	<b>Value:</b> <code>[2412+ i* 5 for i in range(13)]</code>

## 4 Module `command_ap.cmd.ifconfig`

converts the output of ifconfig into a dictionary

### 4.1 Functions

**decode\_ifconfig(*data*)**

read ifconfig's output and returns a dictionary with the data

**Parameters**

**data:** is the captured screen from ifconfig output

**Return Value**

dictionary with decoded ifconfig output

### 4.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> <code>'command_ap.cmd'</code>

## 5 Module `command_ap.cmd.iwconfig`

convert the output of `iwconfig` into a dictionary

### 5.1 Functions

**`grab_first(x, k, type=None)`**

helper function to decode `iwconfig`. grabs the first element of the split given by key `k`

**Parameters**

**x:** string to be splitted by 'espaces'  
**k:** position of the splitted result to be returned  
**type:** valid values are [int, float, None]. If None, return the str, else try to convert to the specified type

**Return Value**

the element 'k'  
*(type=type)*

**`decode_iwconfig(data)`**

get the output of `iwconfig` and convert it into a dictionary

**Parameters**

**data:** output of `iwconfig` captured by the system

**Return Value**

a dictionary with `iwconfig` fields

### 5.2 Variables

Name	Description
<code>cmds_iwconfig</code>	<b>Value:</b> {'AP': <__builtin__.function object>, 'Bit Rate': <__buil...
<code>__package__</code>	<b>Value:</b> None

## 6 Module command\_ap.cmd.scan

convert the output of iw dev station dump into a dictionary

### 6.1 Functions

**find\_in\_cmd**(*line*)

searches the line against the text in 'cmds' returns the data in a simple dictionary

**get\_subitems**(*\_l, lines*)

**decode\_scan**(*data*)

decodes all the information returned by 'scan dump' TODO: finish all fields

**Parameters**

**data:** the output of scan dump

**Return Value**

dictionary containing the data

**decode\_scan\_mac**(*data*)

get the list of APs in range

**Parameters**

**data:** the output of scan dump

**Return Value**

list with the macs detected

**decode\_scan\_basic**(*data*)

get the list of APs in range

**Parameters**

**data:** the output of scan dump

**Return Value**

list with the macs detected

### 6.2 Variables

Name	Description
cmds	<b>Value:</b> ['TSF', 'freq', 'beacon interval', 'capability', 'signal'...]
cmds_sub	<b>Value:</b> ['RSN', 'WMM', 'BSS Load', 'HT operation', 'Overlapping B...]
__package__	<b>Value:</b> 'command_ap.cmd'

## 7 Module `command_ap.cmd.station`

convert the output of `iw dev station dump` into a dictionary

### 7.1 Functions

#### `decode_iw_station(data)`

return the data from "iw dev station dump"

##### Parameters

`data`: output from "iw dev station dump"

##### Return Value

#### `decode_hostapd_status(data)`

decodes "hostapd\_cli status"'s output

@param `data`: output from `hostapd_cli status`

@return: dictionary containing

```
{olbc_ht : 1
  cac_time_left_seconds : N/A
  num_sta_no_short_slot_time : 0
  olbc : 0
  num_sta_non_erp : 0
  ht_op_mode : 0x15
  state : ENABLED
  num_sta_ht40_intolerant : 0
  channel : 6
  bssid[0] : b0:aa:ab:ab:ac:11
  ieee80211n : 1
  cac_time_seconds : 0
  num_sta[0] : 2
  ieee80211ac : 0
  phy : phy0
  num_sta_ht_no_gf : 1
  freq : 2437
  num_sta_ht_20_mhz : 2
  num_sta_no_short_preamble : 0
  secondary_channel : 0
  ssid[0] : ethanolQL1
  num_sta_no_ht : 0
  bss[0] : wlan0
}
```

**is\_mac(s)**

verifies if 's' contains a MAC address

**Return Value**

the mac address found or None

(*type=*str)

**decode\_hostapd\_station(data)**

@param data: output from hostapd\_cli all\_sta

@return: dictionary of dictionary

```
{station1_mac: {'dot11RSNAStatsSelectedPairwiseCipher': '00-0f-ac-4',
                'rx_packets': '164',
                'dot11RSNAStatsTKIPLocalMICFailures': '0',
                'rx_bytes': '5420',
                'inactive_msec': '11828',
                'connected_time': '3402',
                'hostapdWPAPTKState': '11',
                'tx_bytes': '1340',
                'dot11RSNAStatsVersion': '1',
                'tx_packets': '10',
                'hostapdWPAPTKGroupState': '0',
                'dot11RSNAStatsTKIPRemoteMICFailures': '0'},
}
```

## 7.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> 'command_ap.cmd'

## 8 Module `command_ap.cmd.survey`

convert the output of `iw dev station dump` into a dictionary

### 8.1 Functions

```
decode_survey(data)
```

decodes the data provided by "iw survey dump"

@param *data*: output from iw dev survey dump

@return: dictionary of dictionary

```
{2432: {'noise': '-95 dBm',
        'in use': True,
        'channel transmit time': '713 ms',
        'channel busy time': '9479 ms',
        'channel active time': '54259 ms',
        'channel receive time': '8279 ms'},
 2467: {},
}
```

### 8.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> <code>'command_ap.cmd'</code>



## 9 Module `command_ap.cmd.xmit`

Module `xmit`

This module decodes the "xmit" file. Returns a dictionary with all decoded fields.

### 9.1 Functions

**check**(*line*, *items*)

helper function: test if one of the items in items exists in line

**Parameters**

**line:** the line to check

**items:** list of items

**Return Value**

true if the item in items exists in line

**decode\_xmit**(*filename*)

reads the ath\*k/xmit file, if file not found returns an empty dictionary otherwise decodes the file and returns a dictionary with its contents

**Parameters**

**filename:** full path to xmit

**Return Value**

a dictionary with xmit's content

### 9.2 Variables

Name	Description
<code>lines_with_queue_data</code>	<b>Value:</b> ['MPDUs Queued', 'MPDUs Completed', 'MPDUs XRetried', 'Ag...']
<code>__package__</code>	<b>Value:</b> 'command_ap.cmd'

## 10 Package command\_\_ap.get\_\_set

### 10.1 Modules

- **client**: the server accepts requests from an http client.  
(Section 11, p. 17)
- **server**: server that accepts requests from an http client used to send commands to the AP  
(Section 12, p. 18)
- **server\_ffox**: The client (firefox) sends the following json data:  
(Section 13, p. 24)
- **teste**: Test to get the data to compute: MOS client, hybrid and AP  
(Section 14, p. 26)

### 10.2 Variables

Name	Description
__package__	<b>Value:</b> None

## 11 Module `command_ap.get_set.client`

the server accepts requests from an http client. this module is uses to send commands to the AP, for testing purposes.

Usage: `python3 server.py [-port 8080]`

### 11.1 Variables

Name	Description
<code>valid_urls</code>	<b>Value:</b> <code>['/', '/test', '/get_info', '/get_power', '/set_power', '...]</code>

## 12 Module `command_ap.get_set.server`

server that accepts requests from an http client  
used to send commands to the AP

Usage from command line:

-----

```
python3 -m get_set.server.py [--port 8080]
```

Usage from program:

-----

```
import get_set.server
server.run(port)
```

Requirements

-----

iw 4.9+ (<https://git.kernel.org/pub/scm/linux/kernel/git/jberg/iw.git/snapshot/iw-4.9.tar.gz>)  
iwconfig version 30

### 12.1 Functions

<code>run(port=8080)</code>
-----------------------------

<code>collect(port)</code>
----------------------------

creates an HTTP server that receives POST requests from the client save the BODY as JSON in a file
----------------------------------------------------------------------------------------------------

<b>Parameters</b>
-------------------

<b>port:</b> number of the server port. Required.
---------------------------------------------------

### 12.2 Variables

Name	Description
LOG	<b>Value:</b> <code>logging.getLogger('REST_SERVER')</code>
httpd	<b>Value:</b> None
last_rt	<b>Value:</b> <code>dict()</code>
last_tx_bytes	<b>Value:</b> None
last_ampdu	<b>Value:</b> None
MAX_REPORTED_BITRATE	<b>Value:</b> 20000.0
MAXIMUM_TX_BITRATE	<b>Value:</b> 54.0
MAX_TX_BYTES_WIFI	<b>Value:</b> <code>MAXIMUM_TX_BITRATE* 1024* 1024</code>

### 12.3 Class myHandler

[illegible]

"This class will handles any incoming request from the browser

### 12.3.1 Methods

```
init (self, request, client address, server)
```

```
query(self)
```

---

parses the HTML query field

```
send_error(self)
```

returns to the web client a 404 error

```
send_dictionary(self, d)
```

returns to the web client a dictionary containing the data. the client should use `pickle.loads()` to reconvert the data to a python object

**info**(*self*)

```
process /get_info
```

```
@return: dictionary
```

```
{'wiphy': '0', 'Interface': 'wlan0', 'addr': 'b0:aa:ab:ab:ac:11',  
  'width': '20MHz,', 'channel': '6',  
  'txpower': '1.00 dBm', 'ssid': 'ethanolQL1', 'type': 'AP',  
  'ifindex': '3', 'frequency': '2437MHz,',  
  'wdev': '0x1', 'center1': '2437MHz'}
```

```
@rtype: dict
```

iwconfig(*self*)

```
process /get_iwconfig
```

```
@return: dictionary
```

```
{'Power Management': 'off', 'RTS thr': 'off', 'IEEE': '802.11bgn',  
'Mode': 'Master', 'Retry short limit': 7, 'Fragment thr': 'off',  
'interface': 'wlan0'}
```

```
{'iface': 'wlan0',  
  'rx_bytes': '2986426585', 'rx_overruns': '0', 'rx_dropped': '0',  
  'rx_packets': '30257063', 'rx_scale_bytes': '2.9', 'rx_errors': '0',  
  'tx_scale_bytes': '53.9', 'tx_bytes': '53923422941', 'tx_dropped': '0',  
  'tx_packets': '43083207', 'tx_overruns': '0', 'tx_errors': '0',  
  'collisions': '0', 'frame': '0',  
  'txqueuelen': '1000',  
  'carrier': '0',  
}
```

the tx power of iface

```
set the tx power of iface to new_power
```

(type=dict)

```
@rtype: dict
```

**get\_stations(self)**

process /num\_stations

@return:

```
{'54:e6:fc:da:ff:34': {'short slot time': 'yes', 'DTIM period': 2.0,
                        'authorized': 'yes',
                        'tx bitrate': 1.0,
                        'tx bytes': 322.0, 'tx packets': 2.0, 'tx failed': 0.0,
                        'rx bitrate': 1.0
                        'rx bytes': 288.0, 'rx drop misc': 1.0, 'rx packets': 2.0,
                        'preamble': 'short',
                        'WMM/WME': 'yes',
                        'signal avg': 58.0, 'MFP': 'no',
                        'beacon interval': 100.0, 'signal': 57.0,
                        'tx retries': 1.0,
                        'authenticated': 'yes', 'TDLS peer': 'no',
                        'connected time': 0.0, 'inactive time': 4.0, 'associated': 'yes',
                        }
}
```

@rtype: dict

**get\_num\_stations(self)**

process /get\_num\_stations

**Return Value**

number of stations

(type=int)

**get\_survey(self)**

@return:

```
{2432: {'channel busy time': 394.0, 'channel receive time': 285.0, 'channel transmit time': 81
2437: {'in use': True, 'channel receive time': 1073537372.0, 'noise': 80.0, 'channel busy tim
2442: {'channel busy time': 682.0, 'channel receive time': 336.0, 'channel transmit time': 31
2467: {}},
2472: {}},
}
```

@rtype: dict

**get\_scan(self)**

returns the partial results from iw scan dump

```
{'50:c7:bf:3b:db:37': {'channel': '1',
                      'SSID': 'LAC',
                      'TSF': '0d, 05:19:27',
                      'last seen': 104,
                      'freq': 2412,
                      'signal': -54.0,
                      'beacon interval': 100},
 '84:b8:02:44:07:d2': {'channel': '1',
                      'SSID': 'DCC-usuarios',
                      'TSF': '27d, 03:24:26',
                      'last seen': 1024,
                      'freq': 2412,
                      'signal': -58.0,
                      'beacon interval': 102}
}
```

**get\_scan\_mac(self)**

return the result from iw scan dump

**Return Value**

list[str] each entry is a detected mac

**get\_config(self)**

return the result from hostapd\_cli get\_config

**Return Value**

```
{'group_cipher': 'CCMP', 'key_mgmt': 'WPA-PSK ', 'rsn_pairwise_cipher':
 'CCMP', 'ssid': 'ethanolQL1', 'bssid': 'b0:aa:ab:ab:ac:11', 'wps_state':
 'disabled'}
```

(type=dict)

**hello(self)**

standard hello response. white page with 200 code

**do\_GET(self)**

self.path is the command the client wants to execute

function\_handler is a dictionary that contains {url : function responds to the command}



---

**fill\_feature\_results**(self, survey, station, k, stations, iface)

---

function that returns the features of a station.

**Parameters**

survey: data from iw survey dump  
station: the station data selected from the result of "iw station dump"  
k: the k-th value of the survey  
stations: data from iw station dump  
iface: wireless interface name

---



---

**get\_features**(self)

---

process /get\_features

here we collect all features necessary to train the QoS predictor

@return: dictionary

```
{'54:e6:fc:da:ff:34': {'tx_bitrate': 1.0, 'rx_bitrate': 1.0,
                        'tx_power': 1.0, 'avg_signal': 54.0,
                        'rxdrop': 16.0, 'rxb': 1232.0, 'rxp': 32.0,
                        'txr': 0.0, 'txp': 3.0, 'txf': 0.0, 'txb': 487.0,
                        'crt': 1073085286.0, 'cbt': 1163082876.0,
                        'ctt': 60749755.0, 'cat': 3626867638.0,
                        'num_stations': 1
                       }
}
```

---



---

**get\_mos\_hybrid**(self)

---

**Return Value**

[[timestamp, FR, frame\_loss, SBR, PLR], ...]

---



---

**get\_mos\_ap**(self)

---

**Return Value**

[num\_stations, BER, AMPDU, traffic\_load] needed to compute the MOS\_AP

---



---

**get\_mos\_client**(self)

---

read from local memory is filled using an node.js server  
this server receives connections from the clients, and then stores  
the values in a local json file

- r[t] = reportedBitrate in time [t] / max\_bitrate
- srt = not\_running\_time / (not\_running\_time + execution\_time)
- r[t-1] is obtained from a saved variable: self.last\_rt[client\_ip]

@ return: [rt, rt\_1, srt, sta]

---

## 13 Module command\_ap.get\_set.server\_ffox

The client (firefox) sends the following json data:

```
{'chunkData[resolution][]': '768', 'chunkData[start]': '32', 'chunkData[filename]': '7-16.video', 'chunkData[index]': '16', 'chunkData[quality]': '6', 'chunkData[endFragment]': 'true', 'chunkData[bandwidth]': '976342', 'chunkData[segmentType]': 'MediaSegment', 'playing[quality]': '6', 'playing[time]': '31.607175', 'playing[paused]': 'false', 'chunkData[representationId]': '7', 'chunkData[end]': '34', 'chunkData[codec]': 'video/mp4;codecs="avc3.64000C"', 'index': 6, 'latency': {'avg': 0.04, 'low': 0.08, 'high': 0.06}, 'droppedFPS': 15, 'maxIndex': 19, 'reportedBitrate': 976, 'calculatedBitrate': 810, 'video_ratio': {'avg': 11.63, 'low': 17.24, 'high': 13.63}, 'bufferLevel': 2.4, 'download': {'avg': 0.12, 'low': 0.17, 'high': 0.15},
```

### 13.1 Functions

#### **decode3field(*x*)**

##### **Parameters**

**x**: a string formatted as "float | float | float" to be converted to a list

##### **Return Value**

a list with 3 floats

(*type=list(float)*)

#### **decodeInt(*x*)**

##### **Parameters**

**x**: a string to be converted to int

##### **Return Value**

an int, or np.nan

(*type=int*)

### 13.2 Variables

Name	Description
LOG	<b>Value:</b> logging.getLogger('SERVER_FFOX')
funcs	<b>Value:</b> {'droppedFPS': lambda x: decodeInt(x), 'index': lambda x: ...}
map_ip_to_sta	<b>Value:</b> {'192.168.0.11': 'cloud', '150.164.10.50': 'storm', '192....
ffox_memory	<b>Value:</b> FirefoxDataMemory()

### 13.3 Class FirefoxDataMemory

object —  
 command\_ap.get\_set.server\_ffox.FirefoxDataMemory

### 13.3.1 Methods

```
__init__(self)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

```
push(self, data)
```

```
pop(self)
```

#### *Inherited from object*

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),  
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),  
__str__(), __subclasshook__()
```

### 13.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 13.4 Class SrvPosts

`http.server.BaseHTTPRequestHandler`

`command_ap.get_set.server_ffox.SrvPosts`

receives posts from the client (firefox), and saves the data into a json file

### 13.4.1 Methods

```
do_POST(self)
```

get the data, and save it into memory (a global variable called `ffox_memory`)

## 14 Module `command_ap.get_set.teste`

Test to get the data to compute: MOS client, hybrid and AP

### 14.1 Functions

<b>call</b> ( <i>cmd</i> , <i>ap_name</i> ='gnu-nb3.winet.dcc.ufmg.br')
-------------------------------------------------------------------------

calls the AP
--------------

<b>Parameters</b>
-------------------

<b>cmd</b> : valid values are ['/get_mos_hybrid', '/get_mos_ap', '/get_mos_client']
----------------------------------------------------------------------------------------

<b>call_h</b> ( <i>cmd</i> ='/get_mos_hybrid')
------------------------------------------------

get MOS hybrid data
---------------------

<b>call_a</b> ( <i>cmd</i> ='/get_mos_ap')
--------------------------------------------

get MOS AP data
-----------------

<b>call_c</b> ( <i>cmd</i> ='/get_mos_client')
------------------------------------------------

get MOS client data
---------------------

## 15 Package command\_ap.publisher\_subscriber

### 15.1 Modules

- **publisher** (*Section 16, p. 28*)
- **subscriber** (*Section 17, p. 29*)

### 15.2 Variables

Name	Description
__package__	<b>Value:</b> None

## 16 Module `command_ap.publisher_subscriber.publisher`

### 16.1 Variables

Name	Description
<code>port</code>	<b>Value:</b> <code>sys.argv [1]</code>
<code>context</code>	<b>Value:</b> <code>zmq.Context()</code>
<code>socket</code>	<b>Value:</b> <code>context.socket(zmq.PUB)</code>

## 17 Module `command_ap.publisher_subscriber.subscriber`

### 17.1 Variables

Name	Description
<code>port</code>	<b>Value:</b> <code>sys.argv [1]</code>
<code>port1</code>	<b>Value:</b> <code>sys.argv [2]</code>
<code>context</code>	<b>Value:</b> <code>zmq.Context()</code>
<code>socket</code>	<b>Value:</b> <code>context.socket(zmq.SUB)</code>
<code>topicfilter</code>	<b>Value:</b> <code>"10001"</code>
<code>total_value</code>	<b>Value:</b> <code>0</code>

## 18 Package command\_ap.rl

### 18.1 Modules

- **agent**: runs the agent: python3 agent.py  
(Section 19, p. 31)
- **app1** (Section 20, p. 35)
- **basic** (Section 21, p. 36)
  - **environment** (Section 22, p. 37)
- **mab**: This module define three abstract MAB agents: \* RandomAbstract: select random actions \* EpsilonGreedyAbstract: select action using an epsilon-greedy policy \* UCBAbstract: selects actions based on the UCB policy  
(Section 23, p. 38)
- **model**: This module calculates the QoS based on the features  
(Section 24, p. 45)
- **reward**: runs the agent: python3 agent.py  
(Section 25, p. 46)

### 18.2 Variables

Name	Description
curr	<b>Value:</b> <code>os.getcwd()</code>



## 19 Module `command_ap.rl.agent`

runs the agent: `python3 agent.py`

the `-double-trick` parameter uses the trick suggested by xxx, since MAB was not meant to run forever. If it is active, time periods of  $T$  iterations will be considered, and for each  $T$  iterations this period is increased to  $2T$ . `-T` define the initial period.

**Version:** 2.0

**Author:** Henrique Moura

**Copyright:** Copyright 2018, h3dema

**License:** GPL

### 19.1 Functions

<b><code>send_command(server, port, interface, cmd)</code></b>
----------------------------------------------------------------

send a command to the AP, using the REST API
----------------------------------------------

<b>Parameters</b>
-------------------

<b>server:</b>	server name or IP
<b>port:</b>	socket port
<b>interface:</b>	name of the wireless interface, e.g. 'wlan0'
<b>cmd:</b>	the <code>/command[?query]</code>

<b><code>set_power(server, port, interface, new_power)</code></b>
-------------------------------------------------------------------

set the AP's transmission power
---------------------------------

<b>Parameters</b>
-------------------

<b>server:</b>	server name or IP
<b>port:</b>	socket port
<b>interface:</b>	name of the wireless interface, e.g. 'wlan0'
<b>new_power:</b>	the new transmission power in dBm [1, 15] ( <i>type=int</i> )

**get\_power**(*server, port, interface*)

get the AP's transmission power

**Parameters**

**server:** server name or IP

**port:** socket port

**interface:** name of the wireless interface, e.g. 'wlan0'

**Return Value**

the transmission power in dBm [1, 15]

(*type=int*)

**get\_features**(*server, port, interface*)

get the AP's features necessary to calculate the QoS

**Parameters**

**server:** server name or IP

**port:** socket port

**interface:** name of the wireless interface, e.g. 'wlan0'

**Return Value**

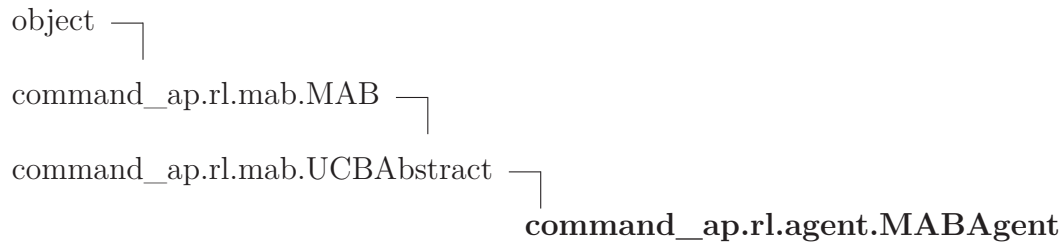
the features

(*type=dict*)

## 19.2 Variables

Name	Description
<code>__credits__</code>	<b>Value:</b> ["Henrique Moura"]
<code>__maintainer__</code>	<b>Value:</b> "Henrique Moura"
<code>__email__</code>	<b>Value:</b> "h3dema@gmail.com"
<code>__status__</code>	<b>Value:</b> "Production"
<code>LOG</code>	<b>Value:</b> <code>logging.getLogger('AGENT')</code>
<code>f_handler</code>	<b>Value:</b> <code>logging.FileHandler('Log_Qos.log')</code>
<code>f_format</code>	<b>Value:</b> <code>logging.Formatter('%(message)s')</code>

### 19.3 Class MABAgent



this is the real class. it implements the abstract methods from UCBAbstract you should implement only the `run_action` method this method interacts with the environment, performing the action and collection the reward it returns if the agent was able to perform the action

#### 19.3.1 Methods

<b><code>__init__</code></b> ( <i>self</i> , <i>n_actions</i> , <i>server</i> , <i>port</i> , <i>interface</i> )
the defaults of C and b define a UCB1 policy
<b>Parameters</b>
<i>n_actions</i> : number of actions the agent can perform from [0, <i>n_actions</i> - 1] ( <i>type=int</i> )
<i>server</i> : server name or IP
<i>port</i> : socket port
<i>interface</i> : name of the wireless interface, e.g. 'wlan0'
Overrides: <code>object.__init__</code>

<b><code>run_action</code></b> ( <i>self</i> , <i>action</i> )
:return r: the reward of the action taken :return success: boolean value indicating if the agent could perform the action or not
Overrides: <code>command_ap.rl.mab.MAB.run_action</code>

*Inherited from `command_ap.rl.mab.UCBAbstract`(Section 23.6)*

`get_action()`, `get_prob()`, `w()`

*Inherited from `command_ap.rl.mab.MAB`(Section 23.3)*

`name()`, `reset_pulls()`, `update()`

*Inherited from `object`*

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),  
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),  
__str__(), __subclasshook__()
```

### 19.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

## 20 Module command\_ap.rl.app1

## 21 Package command\_ap.rl.basic

### 21.1 Modules

- **environment** (*Section 22, p. 37*)

## 22 Module `command_ap.rl.basic.environment`

### 22.1 Class environment

object —  
     `command_ap.rl.basic.environment.environment`

#### 22.1.1 Methods

<code>__init__(self, **kwargs)</code> x. <code>__init__</code> (...) initializes x; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<code>get_reward(self, **kwargs)</code>
-----------------------------------------

<code>take_action(self, **kwargs)</code>
------------------------------------------

#### *Inherited from `object`*

`__delattr__`(), `__format__`(), `__getattr__`(), `__hash__`(), `__new__`(),  
`__reduce__`(), `__reduce_ex__`(), `__repr__`(), `__setattr__`(), `__sizeof__`(),  
`__str__`(), `__subclasshook__`()

#### 22.1.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

## 23 Module `command_ap.rl.mab`

This module define three abstract MAB agents: \* `RandomAbstract`: select random actions \* `EpsilonGreedyAbstract`: select action using an epsilon-greedy policy \* `UCBAbstract`: selects actions based on the UCB policy

**Version:** 2.0

**Author:** Henrique Moura

**Copyright:** Copyright 2018, h3dema

**License:** GPL

### 23.1 Functions

<b><code>softmax(x)</code></b>
returns the softmax function (probabilities) given an array x
<b>Parameters</b>
x: float
<b>Return Value</b>
softmax(x)
( <i>type=float</i> )

### 23.2 Variables

Name	Description
<code>__credits__</code>	<b>Value:</b> ["Henrique Moura"]
<code>__maintainer__</code>	<b>Value:</b> "Henrique Moura"
<code>__email__</code>	<b>Value:</b> "h3dema@gmail.com"
<code>__status__</code>	<b>Value:</b> "Production"
<code>LOG</code>	<b>Value:</b> <code>logging.getLogger('MAB')</code>

### 23.3 Class MAB

```

object └─
          └─ command_ap.rl.mab.MAB

```



### 23.3.1 Methods

<b>__init__</b> ( <i>self</i> , <i>n_actions</i> )
creates the agent handler
<b>Parameters</b>
<i>n_actions</i> : number of actions the agent can perform from [0, <i>n_actions</i> - 1]
( <i>type=int</i> )
Overrides: object.__init__

<b>get_action</b> ( <i>self</i> )
Get current best action
<b>Return Value</b>
the selected action according to the policy
( <i>type=int</i> )

<b>run_action</b> ( <i>self</i> , <i>action</i> )
@return: <i>r</i> , <i>success</i>
' <i>r</i> ' is the reward of the action taken, and ' <i>success</i> ' is boolean value indicating if the action was successful
@rtype float, bool

<b>reset_pulls</b> ( <i>self</i> )
reset the number of executions to zero.

<b>update</b> ( <i>self</i> , <i>action</i> , <i>reward</i> )
observe the reward from action and update agent's internal parameters
<b>Parameters</b>
<i>action</i> : the action 'a' taken
<i>reward</i> : the reward obtained by taking action 'a'

<b>name</b> ( <i>self</i> )
name of the class

#### *Inherited from object*

\_\_delattr\_\_(), \_\_format\_\_(), \_\_getattr\_\_(), \_\_hash\_\_(), \_\_new\_\_(),  
\_\_reduce\_\_(), \_\_reduce\_ex\_\_(), \_\_repr\_\_(), \_\_setattr\_\_(), \_\_sizeof\_\_(),

`__str__()`, `__subclasshook__()`

### 23.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 23.4 Class RandomAbstract



this class implements the random policy

### 23.4.1 Methods

<b><code>get_action(self)</code></b>
returns a random action
<b>Return Value</b>
the selected action
( <i>type=int</i> )
Overrides: <code>command_ap.rl.mab.MAB.get_action</code>

*Inherited from `command_ap.rl.mab.MAB` (Section 23.3)*

`__init__()`, `name()`, `reset_pulls()`, `run_action()`, `update()`

*Inherited from object*

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,  
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,  
`__str__()`, `__subclasshook__()`

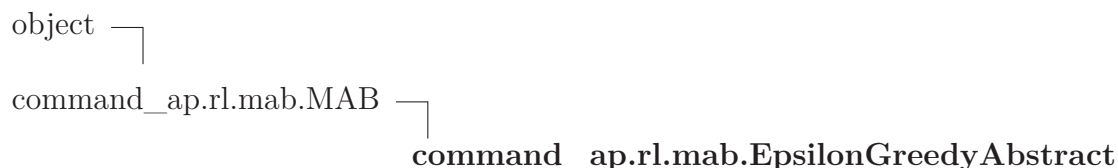
### 23.4.2 Properties

Name	Description
<i>Inherited from object</i>	

*continued on next page*

Name	Description
<code>__class__</code>	

## 23.5 Class *EpsilonGreedyAbstract*



this class implements the epsilon-greedy policy

### 23.5.1 Methods

<b><code>__init__</code></b> ( <i>self</i> , <i>n_actions</i> , <i>epsilon</i> =0.01) creates the agent handler <b>Parameters</b> <i>n_actions</i> : number of actions the agent can perform from [0, <i>n_actions</i> - 1] Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)
<b><code>get_action</code></b> ( <i>self</i> ) selects the action to be taken based on the epsilon-greedy policy <b>Return Value</b> the selected action ( <i>type=int</i> ) Overrides: <code>command_ap.rl.mab.MAB.get_action</code>

*Inherited from `command_ap.rl.mab.MAB`(Section 23.3)*

`name()`, `reset_pulls()`, `run_action()`, `update()`

*Inherited from `object`*

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`,  
`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,  
`__str__()`, `__subclasshook__()`

### 23.5.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

## 23.6 Class UCBAbstract



### 23.6.1 Methods

<b>__init__</b> ( <i>self</i> , <i>n_actions</i> , <i>C</i> =1, <i>b</i> =2)
the defaults of C and b define a UCB1 policy
<b>Parameters</b>
<i>n_actions</i> : number of actions the agent can perform from [0, <i>n_actions</i> - 1]
Overrides: object.__init__

<b>w</b> ( <i>self</i> )
--------------------------

<b>get_prob</b> ( <i>self</i> )
returns the probability of each action using SOFTMAX

<b>get_action</b> ( <i>self</i> )
selects the action to be taken based on the UCB policy
<b>Return Value</b>
the selected action
( <i>type=int</i> )
Overrides: command_ap.rl.mab.MAB.get_action

*Inherited from command\_ap.rl.mab.MAB(Section 23.3)*

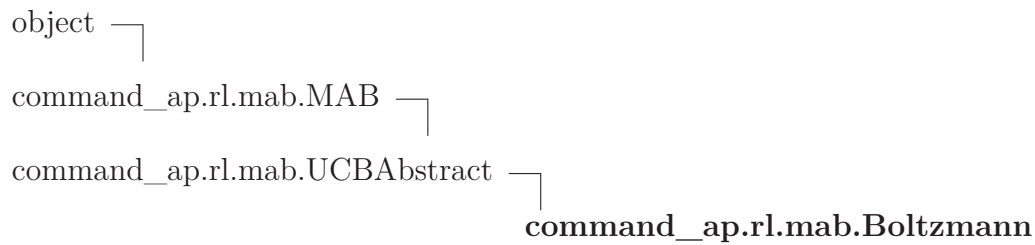
name(), reset\_pulls(), run\_action(), update()

***Inherited from object***

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(),
__str__(), __subclasshook__()
```

**23.6.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

**23.7 Class Boltzmann****23.7.1 Methods**

<b><code>get_action(self)</code></b>
selects the action to be taken based on the softmax policy
<b>Return Value</b>
the selected action
<i>(type=int)</i>
Overrides: <code>command_ap.rl.mab.MAB.get_action</code>

***Inherited from `command_ap.rl.mab.UCBAbstract`(Section 23.6)***

```
__init__(), get_prob(), w()
```

***Inherited from `command_ap.rl.mab.MAB`(Section 23.3)***

```
name(), reset_pulls(), run_action(), update()
```

***Inherited from object***

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(),
```

`__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`,  
`__str__()`, `__subclasshook__()`

### 23.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

## 24 Module `command_ap.rl.model`

This module calculates the QoS based on the features

### 24.1 Functions

<code>get_QoS(<i>model</i>, <i>features</i>)</code>
-----------------------------------------------------

TODO: implement this, using the model and the features use keras model to predict the QoS based on the features
-----------------------------------------------------------------------------------------------------------------

<code>create_window(<i>data_values</i>, <i>timesteps</i>)</code>
------------------------------------------------------------------

convert the <code>data_values</code> into the format needed by the keras model
--------------------------------------------------------------------------------

<b>Parameters</b>
-------------------

<code>data_values:</code>
---------------------------

<code>timesteps:</code> number of time steps
----------------------------------------------

## 25 Module `command_ap.rl.reward`

runs the agent: `python3 agent.py`

the `-double-trick` parameter uses the trick suggested by xxx, since MAB was not meant to run forever. If it is active, time periods of  $T$  iterations will be considered, and for each  $T$  iterations this period is increased to  $2T$ .  $-T$  define the initial period.

**Version:** 2.0

**Author:** Henrique Moura

**Copyright:** Copyright 2018, h3dema

**License:** GPL

### 25.1 Functions

<code>calc_reward(<i>qos</i>, <i>power</i>)</code>
----------------------------------------------------

this function goes to the agent it receives two scaled parameters (between 0 and 1), and returns the reward between 0 and 1
-----------------------------------------------------------------------------------------------------------------------------

### 25.2 Variables

Name	Description
<code>__credits__</code>	<b>Value:</b> ["Henrique Moura"]
<code>__maintainer__</code>	<b>Value:</b> "Henrique Moura"
<code>__email__</code>	<b>Value:</b> "h3dema@gmail.com"
<code>__status__</code>	<b>Value:</b> "Production"



## 26 Script script-hostapd\_conf

### 26.1 Variables

Name	Description
interface	Value: wlan0
bssid	Value: aa:
ssid	Value: my_wifi
driver	Value: nl80211
ignore_broadcast_ssid	Value: 0
channel	Value: 6
hw_mode	Value: g
wmm_enabled	Value: 1
ieee80211n	Value: 1
wpa	Value: 2
wpa_passphrase	Value: password
wpa_pairwise	Value: TKIP
rsn_pairwise	Value: CCMP
auth_algs	Value: 1
macaddr_acl	Value: 0
ctrl_interface	Value: / var/ run/ hostapd
logger_syslog	Value: -1
logger_syslog_level	Value: 0
logger_stdout	Value: -1
logger_stdout_level	Value: 0

## 27 Script `script-setup_cfg`

### 27.1 Functions

**platform**(*iterable*)

Return True if `bool(x)` is True for any `x` in the iterable. If the iterable is empty, return False.

**Return Value**

bool

### 27.2 Variables

Name	Description
name	<b>Value:</b> Command-the-ap
version	<b>Value:</b> 1.0.0
author	<b>Value:</b> Henrique Moura
description	<b>Value:</b> This group of python modules allows to send commands from...
license	<b>Value:</b> GNU
keywords	<b>Value:</b> wireless
classifiers	<b>Value:</b> Development Status:
zip_safe	<b>Value:</b> false
python_requires	<b>Value:</b> >= 3.0

## Index

- command\_ap (*package*), 2
  - command\_ap.cmd (*package*), 3
    - command\_ap.cmd.command\_ap (*module*), 4–8
    - command\_ap.cmd.ifconfig (*module*), 9
    - command\_ap.cmd.iwconfig (*module*), 10
    - command\_ap.cmd.scan (*module*), 11
    - command\_ap.cmd.station (*module*), 12–13
    - command\_ap.cmd.survey (*module*), 14
    - command\_ap.cmd.xmit (*module*), 15
  - command\_ap.get\_set (*package*), 16
    - command\_ap.get\_set.client (*module*), 17
    - command\_ap.get\_set.server (*module*), 18–23
    - command\_ap.get\_set.server\_ffox (*module*), 24–25
    - command\_ap.get\_set.teste (*module*), 26
  - command\_ap.publisher\_subscriber (*package*), 27
    - command\_ap.publisher\_subscriber.publisher (*module*), 28
    - command\_ap.publisher\_subscriber.subscriber (*module*), 29
  - command\_ap.rl (*package*), 30
    - command\_ap.rl.agent (*module*), 31–34
    - command\_ap.rl.app1 (*module*), 35
    - command\_ap.rl.basic (*package*), 36
    - command\_ap.rl.mab (*module*), 38–44
    - command\_ap.rl.model (*module*), 45
    - command\_ap.rl.reward (*module*), 46
- script-hostapd\_conf (*script*), 47
- script-setup\_cfg (*script*), 48
  - script-setup\_cfg.platform (*function*), 48