api

API Documentation

November 13, 2019

Contents

1 Package command_ap

1.1 Modules

1.2 Variables

Name	Description
___package___	**Value:** `None`

2 Package command_ap.cmd

2.1 Modules

2.2 Variables

Name	Description
___package___	**Value:** `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 err
@param ht_type: Valid values are ['', 'ht', 'vht']. Defines the type of channel. Invalid type retu
@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_channel(*interface*, *path_iw*=__DEFAULT_IW_PATH)

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 succeded

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 succeded

(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	**Value:** `logging.getLogger('CMD')`
valid_frequencies	**Value:** `[2412+ i* 5 for i in range(13)]`

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
___package___	**Value:** `'command_ap.cmd'`

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
cmds_iwconfig	**Value:** `{'AP': <__builtin__.function object>, 'Bit Rate': <__buil...`
___package___	**Value:** `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	**Value:** `['TSF', 'freq', 'beacon interval', 'capability', 'signal'...`
cmds_sub	**Value:** `['RSN', 'WMM', 'BSS Load', 'HT operation', 'Overlapping B...`
___package___	**Value:** `'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
___package___	**Value:** `'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
___package___	**Value:** `'command_ap.cmd'`

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
lines_with_queue_data	**Value:** `['MPDUs Queued', 'MPDUs Completed', 'MPDUs XRetried', 'Ag...`
___package___	**Value:** `'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)

10.2 Variables

Name	Description
___package___	**Value:** `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
valid_urls	**Value:** `['/', '/test', '/get_info', '/get_power', '/set_power', '...`

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

run(*port*=8080)

12.2 Variables

Name	Description
LOG	**Value:** `logging.getLogger('REST_SERVER')`
httpd	**Value:** `None`
last_rt	**Value:** `dict()`
last_tx_bytes	**Value:** `None`
last_ampdu	**Value:** `None`

12.3 Class myHandler

http.server.BaseHTTPRequestHandler

command_ap.get_set.server.myHandler

"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'}
```

ifconfig(*self*)

```
process /get_ifconfig

@return:
    {'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',
     }
```

get_power(*self*)

process /get_power

Return Value

the tx power of iface

set_power(*self*)

process /set_power

Return Value

set the tx power of iface to new_power

set_channel(*self*)

process /set_channel

Return Value

new channel in a dictionary format {'channel': new_channel}

(type=dict)

xmit(*self*)

```
process /get_xmit

@return: dictionary
{'TXOP Exceeded_VO': '0', 'TX-Pkts-All_VO': '4441336', 'FIFO Underrun_BK': '0',
 'HW-put-tx-buf_BK': '0', 'DELIM Underrun_VI': '0', 'MPDUs Queued_BE': '866',
 'DESC CFG Error_VO': '0', 'Aggregates_BK': '0', 'FIFO Underrun_VO': '0',
 'DESC CFG Error_VI': '0', 'AMPDUs Queued HW_VI': '0', 'TX-Pkts-All_BE': '42978693', 'TX-Pkts-All_
@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
                           }
    }
```

Index