

## LANforge Scripts Users Guide

## Script Table of Contents

py-scripts/lf\_dataplane\_test.py

**Example report: dataplane.pdf**

The Candela Tech WiFi data plane test is designed to conduct an automatic testing of all combinations of station types, MIMO types, Channel Bandwidths, Traffic types, Traffic direction, Frame sizes etc... It will run a quick throughput test at every combination of these test variables and plot all the results in a set of charts to compare performance. The user is allowed to define an intended load as a percentage of the max theoretical PHY rate for every test combination. The expected behavior is that for every test combination the achieved throughput should be at least 70% of the theoretical max PHY rate under ideal test conditions. This test provides a way to go through hundreds of combinations in a fully automated fashion and very easily find patterns and problem areas which can be further debugged using more specific testing.

py-scripts/lf\_interop\_ping.py

Example report: interop\_rings.pdf

The Candela Tech ping test is to evaluate network connectivity and measure the round-trip time taken for data packets to travel from the source to the destination and back. It helps assess the reliability and latency of the network, identifying any packet loss, delays, or variations in response times. The test aims to ensure that devices can communicate effectively over the network and pinpoint potential issues affecting connectivity.

py-scripts/if\_rvr test.py

Example report: rate vs range.pdf

The **Candela Rate vs Range Test** measures the performance over distance of the Device Under Test. Distance is emulated using programmable attenuation and a throughput test is run at each distance/RSSI step and plotted on a chart. The test allows the user to plot RSSI curves both upstream and downstream for different types of traffic and different station types.

## py-scripts>If\_wifi\_capacity\_test.py

### Example report: wifi\_capacity.pdf

The Candela WiFi Capacity test is designed to measure performance of an Access Point when handling different amounts of WiFi Stations. The test allows the user to increase the number of stations in user defined steps for each test iteration and measure the per station and the overall throughput for each trial. Along with throughput other measurements made are client connection times, Fairness, % packet loss, DHCP times and more. The expected behavior is for the AP to be able to handle several stations (within the limitations of the AP specs) and make sure all stations get a fair amount of airtime both in the upstream and downstream. An AP that scales well will not show a significant over-all throughput decrease as more stations are added.

py-scripts/lf tr398v4 test.py

Automate running TR398 issue 4 tests. See `cv_examples/run_tr398_71.bash` for example of how to use this in a larger context.

py-scripts/raw\_cli.py

Utility script intended to be used from shell scripts in order to send commands to a LANforge system through the REST API. This script can send a one-line preformatted command like the kind found in a /home/lanforge/DB directory, or can assemble a command using arguments.

py-scripts/sta\_connect2.py

Test will create a station, create TCP and UDP traffic, run it a short amount of time, and verify whether traffic was sent and received. It also verifies the station connected to the requested BSSID if bssid is specified as an argument. The script will clean up the station and connections at the end of the test.

py-scripts/test\_l3.py

Example report: test\_I3.pdf

The Layer 3 Traffic Generation Test is designed to test the performance of the Access Point by running layer 3 TCP and/or UDP Traffic. Layer-3 Cross-Connects represent a stream of data flowing through the system under test. A Cross-Connect (CX) is composed of two Endpoints, each of which is associated with a particular Port (physical or virtual interface).

The test will create stations, create CX traffic between upstream port and stations, run traffic and generate a report.

## py-scripts/test\_l3\_longevity.py

The Layer 3 Traffic Generation Test is designed to test the performance of the Access Point by running layer 3 TCP and/or UDP Traffic. Layer-3 Cross-Connects represent a stream of data flowing through the system under test. A Cross-Connect (CX) is composed of two Endpoints, each of which is associated with a particular Port (physical or virtual interface).

The test will create stations, create CX traffic between upstream port and stations, run traffic and generate a report.

## Script Information

py-scripts/lf\_dataplane\_test.py

Example report: dataplane.pdf

```

[--influx_tag INFLUX_TAG INFLUX_TAG] [--json JSON]
[--influx_json INFLUX_JSON] [-u UPSTREAM]
[--station STATION] [--dut DUT]
[--download_speed DOWNLOAD_SPEED]
[--upload_speed UPLOAD_SPEED] [--duration DURATION]
[--verbosity VERBOSITY] [--graph_groups GRAPH_GROUPS]
[--local_lf_report_dir LOCAL_LF_REPORT_DIR]
[--lf_logger_config_json LF_LOGGER_CONFIG_JSON]
[--help_summary]

NAME: lf_dataplane_test.py

PURPOSE: This script is designed to run dataplane tests under various scenarios.

EXAMPLE:
# Sample cli to test Dataplane Test :

./lf_dataplane_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge
--instance_name dataplane-instance --config_name test_con --upstream 1.1.eth1 --dut LISP_VAP_DUT
--duration 30s --station 1.1.wlan0 --download_speed 85% --upload_speed 0 --raw_line 'pkts: 60'
--raw_line 'cust_pkt_sz: 88 1200' --raw_line 'directions: DUT Transmit' --raw_line 'traffic_types: UDP'
--raw_line 'bandw_options: 20' --raw_line 'spatial_streams: 2' --raw_line 'modes: 802.11bgn-AX' --pull_report

# Sample cli to test Dataplane Test with influx db (Optional):

./lf_dataplane_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge
--instance_name dataplane-instance --config_name test_con --upstream 1.1.eth2 --dut linksys-8450
--duration 15s --station 1.1.sta01500 --download_speed 85% --upload_speed 0
--raw_line 'pkts: Custom;60;142;256;512;1024;MTU' --raw_line 'cust_pkt_sz: 88 1200'
--raw_line 'directions: DUT Transmit;DUT Receive' --raw_line 'traffic_types: UDP;TCP'
--test_rig Testbed-01 --pull_report
--influx_host c7-graphana --influx_port 8086 --influx_org Candela
--influx_token=_Wd-L8o992701F0c5UmqEp7wZ7YOMaLlxOMgmHfATJGnQbbmYyNxHBR9PgD6taM_tcxqJl6U8DjU1xINFQ==
--influx_bucket ben --influx_tag testbed Ferndale-01

Example 2:

# Sample cli to test Dataplane Test with <_dp_cli_config_>.json :

./lf_dataplane_test.py --json <name>.json

The Example/Sample json file should be :

"lf_dataplane_config.json"

Sample <name>.json between using eth1 and eth2
{
    "mgr": "192.168.0.101",
    "port": "8080",
    "lf_user": "lanforge",
    "lf_password": "lanforge",
    "instance_name": "dataplane-instance",
    "config_name": "test_con",
    "upstream": "1.1.eth1",
    "dut": "asus_5g",
    "duration": "15s",
    "station": "1.1.eth2",
    "download_speed": "85%",
    "upload_speed": "0",
    "raw_line": ["pkts: Custom;60;MTU", "cust_pkt_sz: 88 1200", "directions: DUT Transmit",
    "traffic_types: UDP", "bandw_options: 20", "spatial_streams: 1"]
}

Sample <name>.json between using eth1 and station 1.1.sta0002
{
    "mgr": "192.168.0.101",
    "port": "8080",
    "lf_user": "lanforge",
    "lf_password": "lanforge",
    "instance_name": "dataplane-instance",
    "config_name": "test_con",
    "upstream": "1.1.eth1",
    "dut": "asus_5g",
    "duration": "15s",
    "station": "1.1.sta0002",
    "download_speed": "85%",
    "upload_speed": "0",
    "raw_line": ["pkts: Custom;60;MTU", "cust_pkt_sz: 88 1200", "directions: DUT Transmit",
    "traffic_types: UDP", "bandw_options: 20", "spatial_streams: 1"]
}

SCRIPT_CLASSIFICATION: Test

SCRIPT_CATEGORIES: Performance, Functional, KPI Generation, Report Generation

NOTES:
This script is used to automate running Dataplane tests. You may need to view a Dataplane test
configured through the GUI to understand the options and how best to input data.

Note :
To Run this script gui should be opened with

path: cd LANforgeGUI_5.4.3 (5.4.3 can be changed with GUI version)
pwd (Output : /home/lanforge/LANforgeGUI_5.4.3)
./lfclient.bash -cli-socket 3990

---> lf_dataplane_test.py is designed to run dataplane tests under various scenarios.

./lf_dataplane_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge
--instance_name <instance name> --config_name test_con --upstream <upstream port> --dut <dut name>
--duration <test duration> --station <station name> --download_speed <download rate> --upload_speed <Opposit rate>
--raw_line 'pkts: 60' --raw_line 'cust_pkt_sz: 88 1200' --raw_line 'directions: DUT Transmit'
--raw_line 'traffic_types: UDP' --raw_line 'bandw_options: 20' --raw_line 'spatial_streams: 2'
--raw_line 'modes: 802.11bgn-AX' --pull_report

* --raw_line : 'line contents' will add any setting to the test config. This is useful way to support
any options not specifically enabled by the command options.

```

```
* --set modifications will be applied after the other config has happened, so it can be used to  
override any other config.
```

```
Example of raw text config for Dataplane, to show other possible options:
```

```
show_events: 1  
show_log: 0  
port_sorting: 0  
kpi_id: Dataplane Pkt-Size  
notes0: ec5211 in bridge mode, wpa2 auth.  
bg: 0xE0ECF8  
test_rig:  
show_scan: 1  
auto_helper: 0  
skip_2: 0  
skip_5: 0  
skip_5b: 1  
skip_dual: 0  
skip_tris: 1  
selected_dut: ea8300  
duration: 15000  
traffic_port: 1.1.157 sta01500  
upstream_port: 1.1.2 eth2  
path_loss: 10  
speed: 85%  
speed2: 0Kbps  
min_rssi_bound: -150  
max_rssi_bound: 0  
channels: AUTO  
modes: Auto  
pkts: Custom;60;142;256;512;1024;MTU  
spatial_streams: AUTO  
security_options: AUTO  
bandw_options: AUTO  
traffic_types: UDP;TCP  
directions: DUT Transmit;DUT Receive  
txo_preamble: OFDM  
txo_mcs: 0 CCK, OFDM, HT, VHT  
txo_retries: No Retry  
txo_sgi: OFF  
txo_txpower: 15  
attenuator: 0  
attenuator2: 0  
attenuator_mod: 255  
attenuator_mod2: 255  
attenuations: 0..+50..950  
attenuations2: 0..+50..950  
chamber: 0  
tt_deg: 0..+45..359  
cust_pkt_sz: 88 1200  
show_bar_labels: 1  
show_prct_tput: 0  
show_3s: 0  
show_ll_graphs: 0  
show_gp_graphs: 1  
show_lm: 1  
pause_iter: 0  
outer_loop_atten: 0  
show_realtime: 1  
operator:  
mconn: 1  
mpkt: 1000  
tos: 0  
loop_iterations: 1
```

```
STATUS: Functional
```

```
VERIFIED_ON: 11-MAY-2023,  
           GUI Version: 5.4.6  
           Kernel Version: 6.2.14+
```

```
LICENSE:
```

```
Free to distribute and modify. LANforge systems must be licensed.  
Copyright 2023 Candela Technologies Inc
```

```
INCLUDE_IN_README: False
```

```
options:  
-h, --help      show this help message and exit  
-m MGR, --mgr MGR  address of the LANforge GUI machine (localhost is default)  
-o PORT, --port PORT IP Port the LANforge GUI is listening on (8080 is default)  
--if_user LF_USER  LANforge username to pull reports  
--lf_password LF_PASSWORD  LANforge Password to pull reports  
-i INSTANCE_NAME, --instance_name INSTANCE_NAME  create test instance  
-c CONFIG_NAME, --config_name CONFIG_NAME  Config file name  
-r, --pull_report  pull reports from lanforge (by default: False)  
--load_old_cfg  Should we first load defaults from previous run of the capacity test? Default is False  
--enable ENABLE  Specify options to enable (set cfg-file value to 1). See example raw text config for possible options. May be sp  
--disable DISABLE  Specify options to disable (set value to 0). See example raw text config for possible options. May be sp  
--set SET SET  Specify options to set values based on their label in the GUI. Example: --set 'Basic Client Connectivity'  
--raw_line RAW_LINE  Specify lines of the raw config file. Example: --raw_line 'test_rig: Ferndale-01-Basic' See example raw  
--raw_lines_file RAW_LINES_FILE  Specify a file of raw lines to apply.  
--test_rig TEST_RIG  Specify the test rig info for reporting purposes, for instance: testbed-01  
--test_tag TEST_TAG  Specify the test tag info for reporting purposes, for instance: testbed-01  
--influx_host INFLUX_HOST  Hostname for the Influx database  
--influx_port INFLUX_PORT  IP Port for the Influx database  
--influx_org INFLUX_ORG  Organization for the Influx database
```

```

--influx_token INFLUX_TOKEN
    Token for the Influx database
--influx_bucket INFLUX_BUCKET
    Name of the Influx bucket
--influx_tag INFLUX_TAG INFLUX_TAG
    --influx_tag <key> <val> Can add more than one of these.
--json JSON
    Path to JSON configuration file for test. When specified, JSON takes precedence over command line args.
--influx_json INFLUX_JSON
    Path to Influx JSON configuration
-u UPSTREAM, --upstream UPSTREAM
    Upstream port used in test. For example, '1.1.eth2'
--station STATION
    Station used in test. Example: '1.1.sta01500'
--dut DUT
    Name of DUT used in test. Assumes DUT is already configured in LANforge. Example: 'linksys-8450'
--download_speed DOWNLOAD_SPEED
    Requested download speed used in test. Percentage of theoretical is also supported. Default: 85%.
--upload_speed UPLOAD_SPEED
    Requested upload speed used in test. Percentage of theoretical is also supported. Default: 0
--duration DURATION
    Duration of each traffic run
--verbosity VERTBOSITY
    Verbosity of the report specified as single value in 1 - 11 range (whole numbers).
    The larger the number, the more verbose. Default: 5
--graph_groups GRAPH_GROUPS
    Path to file to save graph_groups to on local system
--local_lf_report_dir LOCAL_LF_REPORT_DIR
    Path to directory to pull remote report data to on local system
--lf_logger_config_json LF_LOGGER_CONFIG_JSON
    Path to logger JSON configuration
--help_summary
    Show summary of what this script does

Data Plane Test

```

#### py-scripts/lf\_interop\_ping.py

Example report: interop\_ping.pdf

```

usage: interop_ping.py [-h] [--mgr MGR] [--target TARGET]
                      [--ping_interval PING_INTERVAL]
                      [--ping_duration PING_DURATION] [--ssid SSID]
                      [--mgr_port MGR_PORT] [--mgr_passwd MGR_PASSWD]
                      [--server_ip SERVER_IP] [--security SECURITY]
                      [--passwd PASSWD] [--virtual] [--num_sta NUM_STA]
                      [--radio RADIO] [--real] [--use_default_config]
                      [--debug] [--log_level LOG_LEVEL]
                      [--lf_logger_config_json LF_LOGGER_CONFIG_JSON]
                      [--help_summary]

NAME: lf_interop_ping.py

PURPOSE: lf_interop_ping.py will let the user select real devices, virtual devices or both and then allows them to run ping test for user given duration and packet interval on the given target IP or domain name.

EXAMPLE-1:
Command Line Interface to run ping test with only virtual clients
python3 lf_interop_ping.py --mgr 192.168.200.103 --target 192.168.1.3 --virtual --num_sta 1 --radio 1.1.wiphy2 --ssid RDT
--passwd OpenWifi --ping_interval 1 --ping_duration 1 --server_ip 192.168.1.61 --debug

EXAMPLE-2:
Command Line Interface to run ping test with only real clients
python3 lf_interop_ping.py --mgr 192.168.200.103 --real --target 192.168.1.3 --ping_interval 1 --ping_duration 1 --server_
--security wpa2_personal --passwd OpenWifi

EXAMPLE-3:
Command Line Interface to run ping test with both real and virtual clients
python3 lf_interop_ping.py --mgr 192.168.200.103 --target 192.168.1.3 --real --virtual --num_sta 1 --radio 1.1.wiphy2 --ss
--passwd OpenWifi --ping_interval 1 --ping_duration 1 --server_ip 192.168.1.61

EXAMPLE-4:
Command Line Interface to run ping test with existing Wi-Fi configuration on the real devices
python3 lf_interop_ping.py --mgr 192.168.200.63 --real --target 192.168.1.61 --ping_interval 5 --ping_duration 1 --passwd

SCRIPT_CLASSIFICATION : Test

SCRIPT_CATEGORIES: Performance, Functional, Report Generation

NOTES:
1.Use './lf_interop_ping.py --help' to see command line usage and options
2.Please pass ping_duration in minutes
3.Please pass ping_interval in seconds
4.After passing the cli, if --real flag is selected, then a list of available real devices will be displayed on the termin
5.Enter the real device resource numbers seperated by commas (,)

STATUS: BETA RELEASE

VERIFIED_ON:
Working date - 20/09/2023
Build version - 5.4.7
kernel version - 6.2.16+

License: Free to distribute and modify. LANforge systems must be licensed.
Copyright 2023 Candela Technologies Inc.

options:
-h, --help      show this help message and exit
--log_level LOG_LEVEL
    Set logging level: debug | info | warning | error | critical
--lf_logger_config_json LF_LOGGER_CONFIG_JSON
    --lf_logger_config_json <json file> , json configuration of logger
--help_summary
    Show summary of what this script does

Optional arguments:
--mgr MGR        hostname where LANforge GUI is running
--target TARGET   Target URL or port for ping test
--ping_interval PING_INTERVAL
    Interval (in seconds) between the echo requests
--ping_duration PING_DURATION
    Duration (in minutes) to run the ping test

```

```

--ssid SSID           SSID for connecting the stations
--mgr_port MGR_PORT  port on which LANforge HTTP service is running
--mgr_passwd MGR_PASSWD
                        Password to connect to LANforge GUI

--server_ip SERVER_IP
                        Upstream for configuring the Interop App

--security SECURITY   Security protocol for the specified SSID: <open | wep | wpa | wpa2 | wpa3>
--passwd PASSWD       passphrase for the specified SSID
--virtual              specify this flag if the test should run on virtual clients
--num_sta NUM_STA     specify the number of virtual stations to be created.
--radio RADIO          specify the radio to create the virtual stations
--real                 specify this flag if the test should run on real clients
--use_default_config  specify this flag if wanted to proceed with existing Wi-Fi configuration of the devices
--debug               Enable debugging


```

Allows user to run the ping test on a target IP or port for the given duration and packet interval with either selected number of virtual stations or provides the list of available real devices and allows the user to select the real devices and run ping test on them.

#### py-scripts/lf\_rvr\_test.py

Example report: rate\_vs\_range.pdf

```

usage: lf_rvr_test.py [-h] [-o PORT] [--lf_user LF_USER]
                      [--lf_password LF_PASSWORD] [-i INSTANCE_NAME]
                      [-c CONFIG_NAME] [-r] [--load_old_cfg] [--enable ENABLE]
                      [--disable DISABLE] [--set SET SET]
                      [--raw_line RAW_LINE] [--raw_lines_file RAW_LINES_FILE]
                      [--test_rig TEST_RIG] [--test_tag TEST_TAG]
                      [--influx_host INFLUX_HOST] [--influx_port INFLUX_PORT]
                      [--influx_org INFLUX_ORG] [--influx_token INFLUX_TOKEN]
                      [--influx_bucket INFLUX_BUCKET]
                      [--influx_tag INFLUX_TAG INFLUX_TAG] [-u UPSTREAM]
                      [--station STATION]
                      [--band {5g,24g,6g,dual_band_5g,dual_band_6g}]
                      [--radio RADIO] [--create_station] [-ssid SSID]
                      [--ssidpw SSIDPW] [--bssid BSSID] [--security SECURITY]
                      [--wifi_mode WIFI_MODE] [--vht160]
                      [--ieee80211w IEEE80211W] [--dut DUT]
                      [--download_speed DOWNLOAD_SPEED]
                      [--upload_speed UPLOAD_SPEED] [--duration DURATION]
                      [--verbosity VERBOSITY] [--graph_groups GRAPH_GROUPS]
                      [--report_dir REPORT_DIR]
                      [--local_lf_report_dir LOCAL_LF_REPORT_DIR]
                      [--log_level LOG_LEVEL]
                      [--lf_logger_config_json LF_LOGGER_CONFIG_JSON]
                      [--help_summary]

TEST NAME      : Rate vs Range Test
TEST SCRIPT    : lf_rvr_test.py
PURPOSE        : The Purpose of this script is to calculate the Throughput rate with increasing attenuation with one emula
NOTES          : To Run this script gui should be opened with

path: cd LANforgeGUI_5.4.3 (5.4.3 can be changed with GUI version)
pwd (Output : /home/lanforge/LANforgeGUI_5.4.3)
./lfclient.bash -cli-socket 3990

This script is used to automate running Rate-vs-Range tests. You
may need to view a Rate-vs-Range test configured through the GUI to understand
the options and how best to input data.

EXAMPLE-1      :
./lf_rvr_test.py --mgr 192.168.100.205 --lf_user lanforge --lf_password lanforge
--instance_name rvr-instance --config_name test_con --upstream 1.2.vap0000
--dut routed-AP --duration 1m --station 1.1.sta0000 --download_speed 85%
--upload_speed 56Kbps --raw_line 'pkts: MTU' --raw_line 'directions: DUT Transmit'
--raw_line 'traffic_types: TCP' --raw_line 'attenuator: 1.1.3219' --raw_line 'attenuations: 0..+50..95'
--raw_line 'attenuator_mod: 243' --ssid rvr_2g --ssidpw Password@123 --security wpa2 --radio wiphy0
--bssid DEFAULT --create_station

EXAMPLE-2      :
./lf_rvr_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge \
--instance_name rvr-instance --config_name test_con --upstream 1.1.eth1 \
--dut RootAP --duration 15s --station 1.1.wlan0 \
--download_speed 85% --upload_speed 56Kbps \
--raw_line 'pkts: MTU' \
--raw_line 'directions: DUT Transmit' \
--raw_line 'traffic_types: TCP' \
--test_rig Ferndale-Mesh-01 --pull_report \
--raw_line 'attenuator: 1.1.1040' \
--raw_line 'attenuations: 0..+50..950' \
--raw_line 'attenuator_mod: 3' \
--influx_host c7-graphana --influx_port 8086 --influx_org Candela \
--influx_token=u_Wd-L8o992701QF0c5UmqBp7w727YOMaWLxOMgmHATJGnQbbmYyNxHBR9PgD6taM_tcxqJ16U8DjUlxFNQ \
--influx_bucket ben \
--influx_tag testbed Ferndale-Mesh

SCRIPT_CLASSIFICATION: Test
SCRIPT_CATEGORIES: Performance, Functional, KPI Generation, Report Generation
NOTES:
attenuator_mod: selects the attenuator modules, bit-field.
This example uses 3, which is first two attenuator modules on Attenuator ID 1040.

--raw_line 'line contents' will add any setting to the test config. This is
useful way to support any options not specifically enabled by the
command options.
--set modifications will be applied after the other config has happened,
so it can be used to override any other config.
sel_port=0: 1.1.wlan0
show_events: 1
show_log: 0
port_sorting: 0

```

```

kpi_id: Rate vs Range
bg: 0x0E0CF8
test_rig:
show_scan: 1
auto_helper: 0
skip_2: 0
skip_5: 0
skip_5b: 1
skip_dual: 0
skip_tri: 1
selected_dut: RootAP
duration: 15000
traffic_port: 1.1.6 wlan0
upstream_port: 1.1.1 eth1
path_loss: 10
speed: 85%
speed2: 56Kbps
min_rssi_bound: -150
max_rssi_bound: 0
channels: AUTO
modes: Auto
pkts: MTU
spatial_streams: AUTO
security_options: AUTO
bandw_options: AUTO
traffic_types: TCP
directions: DUT Transmit
txo_preamble: OFDM
txo_mcs: 0 CCK, OFDM, HT, VHT
txo_retries: No Retry
txo_sgi: OFF
txo_txpower: 15
attenuator: 1.1.1040
attenuator2: 0
attenuator_mod: 243
attenuator_mod2: 255
attenuations: 0..+50..950
attenuations2: 0..+50..950
chamber: 0
tt_deg: 0..+45..359
cust_pkt_sz:
show_bar_labels: 1
show_prcnt_tput: 0
show_3s: 0
show_ll_graphs: 0
show_gp_graphs: 1
show_lm: 1
pause_iter: 0
outer_loop_atten: 0
show_realtime: 1
operator:
mconn: 1
mpkt: 1000
tos: 0
loop_iterations: 1

```

STATUS: BETA RELEASE

#### VERIFIED\_ON:

```

12th May 2023
GUI Version : 5.4.6
Kernel Version : 5.19.17+

```

#### LICENSEE:

```

Free to distribute and modify. LANforge systems must be licensed.
Copyright 2022 Candela Technologies Inc

```

#### INCLUDE\_IN\_README: False

Example of raw text config for Rate-vsRange, to show other possible options:

```

options:
-h, --help      show this help message and exit
-m MGR, --mgr MGR    address of the LANforge GUI machine (localhost is default)
-o PORT, --port PORT   IP Port the LANforge GUI is listening on (8080 is default)
--if_user LF_USER    LANforge username to pull reports
--if_password LF_PASSWORD
                      LANforge Password to pull reports
-i INSTANCE_NAME, --instance_name INSTANCE_NAME
                      create test instance
-c CONFIG_NAME, --config_name CONFIG_NAME
                      Config file name
-r, --pull_report
                      pull reports from lanforge (by default: False)
--load_old_cfg
                      Should we first load defaults from previous run of the capacity test? Default is False
--enable ENABLE
                      Specify options to enable (set cfg-file value to 1). See example raw text config for possible options. M
--disable DISABLE
                      Specify options to disable (set value to 0). See example raw text config for possible options. May be sp
--set SET SET
                      Specify options to set values based on their label in the GUI. Example: --set 'Basic Client Connectivity'
--raw_line RAW_LINE
                      Specify lines of the raw config file. Example: --raw_line 'test_rig: Ferndale-01-Basic' See example raw
--raw_lines_file RAW_LINES_FILE
                      Specify a file of raw lines to apply.
--test_rig TEST_RIG
                      Specify the test rig info for reporting purposes, for instance: testbed-01
--test_tag TEST_TAG
                      Specify the test tag info for reporting purposes, for instance: testbed-01
--influx_host INFLUX_HOST
                      Hostname for the Influx database
--influx_port INFLUX_PORT
                      IP Port for the Influx database
--influx_org INFLUX_ORG
                      Organization for the Influx database
--influx_token INFLUX_TOKEN
                      Token for the Influx database
--influx_bucket INFLUX_BUCKET
                      Name of the Influx bucket
--influx_tag INFLUX_TAG INFLUX_TAG
                      --influx_tag <key> <val> Can add more than one of these.
-u UPSTREAM, --upstream UPSTREAM

```

```

Upstream port for wifi capacity test ex. 1.1.eth2
--station STATION Station to be used in this test, example: 1.1.sta01500
--band {5g,24g,6g,dual_band_5g,dual_band_6g}
band testing --band 6g
--radio RADIO [LANforge station configuration] LANforge radio station created on --radio wiphy0
--create_station [LANforge station configuration] create LANforge station at the beginning of the test
--ssid SSID [station configuration] station ssid, ssid of station must match the wlan created --ssid 6G-wpa3-AP3
--ssidpw SSIDPW, --security_key SSIDPW
[station configuration] station security key --ssidpw hello123
--bssid BSSID, --ap_bssid BSSID
[station configuration] station AP bssid
--security SECURITY [station configuration] security type open wpa wpa2 wpa3
--wifi_mode WIFI_MODE
[station configuration] --wifi_mode auto types auto|a|abg|abgn|abgnAC|abgnAX|an|anAC|anAX|b|bg|bgn|bgnAC
--vht160 [station configuration] --vht160 , Enable VHT160 in lanforge
--ieee80211w IEEE80211W
[station configuration] --ieee80211w 0 (Disabled) 1 (Optional) 2 (Required) (Required needs to be set to R)
--dut DUT Specify DUT used by this test, example: linksys-8450
--download_speed DOWNLOAD_SPEED
Specify requested download speed. Percentage of theoretical is also supported. Default: 85
--upload_speed UPLOAD_SPEED
Specify requested upload speed. Percentage of theoretical is also supported. Default: 0
--duration DURATION Specify duration of each traffic run
--verbosity VERBOSITY
Specify verbosity of the report values 1 - 11 default 5
--graph_groups GRAPH_GROUPS
File to save graph_groups to
--report_dir REPORT_DIR
--local_lf_report_dir LOCAL_LF_REPORT_DIR
--local_lf_report_dir <where to pull reports to> default '' put where dataplane script run from
--log_level LOG_LEVEL
Set logging level: debug | info | warning | error | critical
--lf_logger_config_json LF_LOGGER_CONFIG_JSON
--lf_logger_config_json <json file> , json configuration of logger
--help_summary Show summary of what this script does

```

py-scripts/lf\_wifi\_capacity\_test.py

Example report: wifi\_capacity.pdf

```

usage: lf_wifi_capacity_test.py [-h] [-m MGR] [-o PORT] [--lf_user LF_USER]
[--lf_password LF_PASSWORD] [-i INSTANCE_NAME]
[-c CONFIG_NAME] [-r] [-load_old_cfg]
[--enable ENABLE] [--disable DISABLE]
[--set SET SET] [--raw_line RAW_LINE]
[--raw_lines_file RAW_LINES_FILE]
[--test_rig TEST_RIG] [--test_tag TEST_TAG]
[--influx_host INFLUX_HOST]
[--influx_port INFLUX_PORT]
[--influx_org INFLUX_ORG]
[--influx_token INFLUX_TOKEN]
[--influx_bucket INFLUX_BUCKET]
[--influx_tag INFLUX_TAG INFLUX_TAG]
[-u UPSTREAM] [-b BATCH_SIZE] [-l LOOP_ITER]
[-p PROTOCOL] [-d DURATION]
[--verbosity VERBOSITY]
[--download_rate DOWNLOAD_RATE]
[--upload_rate UPLOAD_RATE] [--sort SORT]
[-s STATIONS] [-cs] [-radio RADIO]
[-ssid SSID] [-security SECURITY]
[-paswd PASWD] [--report_dir REPORT_DIR]
[--scenario SCENARIO]
[--graph_groups GRAPH_GROUPS]
[--local_lf_report_dir LOCAL_LF_REPORT_DIR]
[--lf_logger_config_json LF_LOGGER_CONFIG_JSON]
[--num_stations NUM_STATIONS]
[--start_id START_ID] [--log_level LOG_LEVEL]
[--help_summary]

```

NAME: lf\_wifi\_capacity\_test.py

PURPOSE: This script runs wifi capacity test on the existing stations or runs on the stations specified (if --stations argument is mentioned or stations can be created using -cs with stations names mentioned with --stations) by creating layer3 cross connects and generates html and pdf report.

EXAMPLE:

```

example 1:
./lf_wifi_capacity_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge --instance_name wct_instance --co

example 2:
./lf_wifi_capacity_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge --instance_name wct_instance --co

example 3:
./lf_wifi_capacity_test.py --mgr 192.168.200.165 --upstream 1.1.eth1 --batch_size 1,5 --protocol UDP-IPv4 --duration 30000
--upload_rate 1Gbps --download_rate 1Gbps --raw_line 'ip_tos: 128' --raw_line 'do_pf: 1' --raw_line 'pf_min_period_dl: 100'
--raw_line 'pf_min_period_ul: 300' --raw_line 'pf_max_reconnects: 3' --num_stations 5 --start_id 333 --create_stations
--radio wiphy0 --ssid Netgear-5g --security wpa2 --paswd sharedsecret --test_rig Testbed-01 --set DUT_NAME linksys-8450
--pull_report

```

SCRIPT\_CLASSIFICATION : Test

SCRIPT\_CATEGORIES: Performance, Functional, KPI Generation, Report Generation

NOTES: This script is used to automate wifi capacity tests. You need a configured upstream to run the script.  
To Run this script gui should be opened with

192.168.200.147:1

```

path: cd LANforgeGUI_5.4.3 (5.4.3 can be changed with GUI version)
pwd (Output : /home/lanforge/LANforgeGUI_5.4.3)
./lfclient.bash -cli-socket 3990

```

This is a test file which will run a wifi capacity test.

ex. on how to run this script (if stations are available in lanforge):  
The influx part can be skipped if you are not using influx/graphana.

```

./lf_wifi_capacity_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge
--pull_report == If specified, this will pull reports from lanforge to your code directory,
from where you are running this code

```

--stations == Enter stations to use for wifi capacity

--instance\_name

```

--set DUT_NAME XXXX == Determines which DUT the wifi capacity test should use to get details on
STATUS: BETA RELEASE

VERIFIED_ON:
Working date - 11/05/2023
Build version - 5.4.6
kernel version - 6.2.14+

LICENSE:
Free to distribute and modify. LANforge systems must be licensed.
Copyright 2023 Candela Technologies Inc

INCLUDE_IN_README: False

options:
-h, --help      show this help message and exit
-m MGR, --mgr MGR    address of the LANforge GUI machine (localhost is default)
-o PORT, --port PORT   IP Port the LANforge GUI is listening on (8080 is default)
--lf_user LF_USER    LANforge username to pull reports
--lf_password LF_PASSWORD    LANforge Password to pull reports
-i INSTANCE_NAME, --instance_name INSTANCE_NAME    create test instance
-c CONFIG_NAME, --config_name CONFIG_NAME    Config file name
-r, --pull_report    pull reports from lanforge (by default: False)
--load_old_cfg    Should we first load defaults from previous run of the capacity test? Default is False
--enable ENABLE    Specify options to enable (set cfg-file value to 1). See example raw text config for possible options. M
--disable DISABLE    Specify options to disable (set value to 0). See example raw text config for possible options. May be sp
--set SET SET    Specify options to set values based on their label in the GUI. Example: --set 'Basic Client Connectivity'
--raw_line RAW_LINE    Specify lines of the raw config file. Example: --raw_line 'test_rig: Ferndale-01-Basic' See example raw
--raw_lines_file RAW_LINES_FILE    Specify a file of raw lines to apply.
--test_rig TEST_RIG    Specify the test rig info for reporting purposes, for instance: testbed-01
--test_tag TEST_TAG    Specify the test tag info for reporting purposes, for instance: testbed-01
--influx_host INFLUX_HOST    Hostname for the Influx database
--influx_port INFLUX_PORT    IP Port for the Influx database
--influx_org INFLUX_ORG    Organization for the Influx database
--influx_token INFLUX_TOKEN    Token for the Influx database
--influx_bucket INFLUX_BUCKET    Name of the Influx bucket
--influx_tag INFLUX_TAG INFLUX_TAG    --influx_tag <key> <val>  Can add more than one of these.
-u UPSTREAM, --upstream UPSTREAM    Upstream port for wifi capacity test ex. 1.1.eth1
-b BATCH_SIZE, --batch_size BATCH_SIZE    station increment ex. 1,2,3
-l LOOP_ITER, --loop_iter LOOP_ITER    Loop iteration ex. 1
-p PROTOCOL, --protocol PROTOCOL    Protocol ex.TCP-IPv4
-d DURATION, --duration DURATION    duration in ms. ex. 5000
--verbosity VERBOSITY    Specify verbosity of the report values 1 - 11 default 5
--download_rate DOWNLOAD_RATE    Select requested download rate. Kbps, Mbps, Gbps units supported. Default is 1Gbps
--upload_rate UPLOAD_RATE    Select requested upload rate. Kbps, Mbps, Gbps units supported. Default is 10Mbps
--sort SORT    Select station sorting behaviour: none | interleave | linear Default is interleave.
-s STATIONS, --stations STATIONS    If specified, these stations will be used. If not specified, all available stations will be selected. Ex
--cs, --create_stations    create stations in lanforge (by default: False)
--radio RADIO, --radio RADIO    create stations in lanforge at this radio (by default: wiphy0)
--ssid SSID, --ssid SSID    ssid name
--security SECURITY, --security SECURITY    ssid Security type
--passwd PASWD, --passwd PASWD, -passwd PASWD, --passwd PASWD    ssid Password
--report_dir REPORT_DIR
--scenario SCENARIO
--graph_groups GRAPH_GROUPS    File to save graph groups to
--local_lf_report_dir LOCAL_LF_REPORT_DIR    --local_lf_report_dir <where to pull reports to> default '' put where dataplane script run from
--lf_logger_config_json LF_LOGGER_CONFIG_JSON    --lf_logger_config_json <json file> , json configuration of logger
--num_stations NUM_STATIONS    Specify the number of stations need to be create.
--start_id START_ID    Specify the station starting id
--log_level LOG_LEVEL    Set logging level: debug | info | warning | error | critical
--help_summary    Show summary of what this script does

```

py-scripts/lf\_tr398v4\_test.py

```

usage:
Open this file in an editor and read the top notes for more details.

Example:
./lf_tr398v4_test.py --mgr localhost --port 8080 --lf_user lanforge --lf_password lanforge \
--instance_name tr398-instance --config_name test_con \
--upstream 1.2.eth2 \
--test_rig Testbed-01 --pull_report \

```

```

--local_lf_report_dir /tmp/my-report \
--dut6 'TR398-DUT-r750 ruckus-r750-5g 4c:b1:cd:18:e8:eb (3)' \
--dut5 'TR398-DUT-r750 ruckus-r750-5g 4c:b1:cd:18:e8:ec (1)' \
--dut2 'TR398-DUT-r750 ruckus-r750-2g 4c:b1:cd:18:e8:e8 (2)' \
--raw_lines_file example-configs/tr398v4-ferndale-be-cfg.txt \
--set 'Calibrate 802.11AX Attenuators' 0 \
--set 'Calibrate 802.11AC Attenuators' 0 \
--set '6.1.1 Receiver Sensitivity' 0 \
--set '6.2.1 Maximum Connection' 0 \
--set '6.2.2 Maximum Throughput' 1 \
--set '6.2.3 Airtime Fairness' 0 \
--set '6.2.4 Dual-Band Throughput' 0 \
--set '6.2.5 Bi-Directional Throughput' 0 \
--set '6.3.1 Range Versus Rate' 0 \
--set '6.3.2 Spatial Consistency' 0 \
--set '6.3.3 AX Peak Performance' 0 \
--set '6.4.1 Multiple STAs Performance' 0 \
--set '6.4.2 Multiple Assoc Stability' 0 \
--set '6.4.3 Downlink MU-MIMO' 0 \
--set '6.5.2 AP Coexistence' 0 \
--set '6.5.1 Long Term Stability' 0

./lf_tr398v4_test.py --mgr 192.168.100.105 --port 8080 --lf_user lanforge \
--lf_password lanforge --instance_name x \
--config_name testing --pull_report \
--local_lf_report_dir /tmp --dut5 'ASUS_70 ASUS_70 f0:2f:74:7c:a5:70 (1)' \
--dut2 'ASUS_70 ASUS_70 f0:2f:74:7c:a5:70 (1)' --raw_line "upstream_port: 1.1.eth2"

```

The contents of the 'raw\_lines\_file' argument can be obtained by manually configuring the TR398 issue 2 test in the LANforge GUI, then selecting 'Show Config' in the Advanced configuration tab, then highlighting and pasting that text into file. That file is the argument to the --raw\_lines\_file parameter.

Each TR398 test's setting values can be specified by the python script in multiple ways. For example, each test needs an upstream port. The python script can specify upstream port in several ways and below is the hierarchy of which upstream port will be the final one in the settings.

1. --upstream\_port argument
2. --raw\_lines argument in the command line
3. upstream port specified in the --raw\_lines\_file file.txt
4. upstream port loaded from the --config argument

```

[-h] [-m MGR] [-o PORT] [--lf_user LF_USER] [--lf_password LF_PASSWORD]
[-i INSTANCE_NAME] [-c CONFIG_NAME] [-r] [--load_old_cfg]
[--enable ENABLE] [--disable DISABLE] [--set SET SET]
[--raw_line RAW_LINE] [--raw_lines_file RAW_LINES_FILE]
[--test_rig TEST_RIG] [--test_tag TEST_TAG] [--influx_host INFLUX_HOST]
[--influx_port INFLUX_PORT] [--influx_org INFLUX_ORG]
[--influx_token INFLUX_TOKEN] [--influx_bucket INFLUX_BUCKET]
[--influx_tag INFLUX_TAG INFLUX_TAG] [-u UPSTREAM] [--dut2 DUT2]
[--dut5 DUT5] [--dut6 DUT6] [--local_lf_report_dir LOCAL_LF_REPORT_DIR]
[--help_summary]

options:
-h, --help      show this help message and exit
-m MGR, --mgr MGR    address of the LANforge GUI machine (localhost is default)
-o PORT, --port PORT    IP Port the LANforge GUI is listening on (8080 is default)
--lf_user LF_USER    LANforge username to pull reports
--lf_password LF_PASSWORD    LANforge Password to pull reports
-i INSTANCE_NAME, --instance_name INSTANCE_NAME    create test instance
-c CONFIG_NAME, --config_name CONFIG_NAME    Config file name
-r, --pull_report    pull reports from lanforge (by default: False)
--load_old_cfg    Should we first load defaults from previous run of the capacity test? Default is False
--enable ENABLE    Specify options to enable (set cfg-file value to 1). See example raw text config for possible options. May be specified multiple times. Most tests are enabled by default, except: longterm
--disable DISABLE    Specify options to disable (set value to 0). See example raw text config for possible options. May be specified multiple times.
--set SET SET    Specify options to set values based on their label in the GUI. Example: --set 'Basic Client Connectivity' 1 May be specified multiple times.
--raw_line RAW_LINE    Specify lines of the raw config file. Example: --raw_line 'test_rig: Ferndale-01-Basic' See example raw text config for possible options. This is catch-all for any options not available to be specified elsewhere. May be specified multiple times.
--raw_lines_file RAW_LINES_FILE    Specify a file of raw lines to apply.
--test_rig TEST_RIG    Specify the test rig info for reporting purposes, for instance: testbed-01
--test_tag TEST_TAG    Specify the test tag info for reporting purposes, for instance: testbed-01
--influx_host INFLUX_HOST    Hostname for the Influx database
--influx_port INFLUX_PORT    IP Port for the Influx database
--influx_org INFLUX_ORG    Organization for the Influx database
--influx_token INFLUX_TOKEN    Token for the Influx database
--influx_bucket INFLUX_BUCKET    Name of the Influx bucket
--influx_tag INFLUX_TAG INFLUX_TAG    --influx_tag <key> <val> Can add more than one of these.
-u UPSTREAM, --upstream UPSTREAM    Upstream port for wifi capacity test ex. 1.1.eth2
--dut2 DUT2    Specify 2Ghz DUT used by this test, example: 'TR398-DUT-r750 ruckus-r750-2g 4c:b1:cd:18:e8:e8 (2)'
```

```

--dut5 DUT5      Specify 5Ghz DUT used by this test, example:  

' TR398-DUT-r750 ruckus-r750-5g 4:c:b1:cd:18:e8:ec (1)'  

--dut6 DUT6      Specify 6Ghz DUT used by this test, example:  

' TR398-DUT-r750 ruckus-r750-6g 4:c:b1:cd:18:e8:eb (3)'  

--local_lf_report_dir LOCAL_LF_REPORT_DIR  

--local_lf_report_dir <where to pull reports to>  

default '' means put in current working directory  

--help_summary Show summary of what this script does

```

#### py-scripts/raw\_cli.py

```

usage: /home/candela/git/lanforge-scripts/py-scripts./raw_cli.py  

[-h] [-m MGR] [-o PORT] [-u USER] [-p PASSWD]  

[--resource RESOURCE]  

[--upstream_resource UPSTREAM_RESOURCE]  

[--upstream_port UPSTREAM_PORT] [--radio RADIO]  

[--sta_mode STA_MODE] [--dut_ssid DUT_SSID]  

[--dut_security DUT_SECURITY] [--dut_passwd DUT_PASSWD]  

[--dut_bssid DUT_BSSID] [--download_bps DOWNLOAD_BPS]  

[--upload_bps UPLOAD_BPS] [--side_a_pdu SIDE_A_PDU]  

[--side_b_pdu SIDE_B_PDU] [--runtime_sec RUNTIME_SEC]  

[--debug] [--prefix PREFIX]  

[--bringup_time BRINGUP_TIME]  

[--influx_user INFLUX_USER]  

[--influx_passwd INFLUX_PASSWORD] [--influx_db INFLUX_DB]  

[--influx_host INFLUX_HOST]  

[--monitor_interval MONITOR_INTERVAL]  

[--debug_log DEBUG_LOG] [--no_cleanup]  

[--local_lf_report_dir LOCAL_LF_REPORT_DIR]  

[--test_rig TEST_RIG] [--test_tag TEST_TAG]  

[--dut_hw_version DUT_HW_VERSION]  

[--dut_sw_version DUT_SW_VERSION]  

[--dut_model_num DUT_MODEL_NUM]  

[--dut_serial_num DUT_SERIAL_NUM]  

[--test_priority TEST_PRIORITY] [--test_id TEST_ID]  

[--csv_outfile CSV_OUTFILE] [--log_level LOG_LEVEL]  

[--lf_logger_config_json LF_LOGGER_CONFIG_JSON]  

[--help_summary]

```

#### py-scripts/sta\_connect2.py

```

usage: sta_connect2.py [-h] [-m MGR] [-o PORT] [-u USER] [-p PASSWD]  

[--resource RESOURCE]  

[--upstream_resource UPSTREAM_RESOURCE]  

[--upstream_port UPSTREAM_PORT] [--radio RADIO]  

[--sta_mode STA_MODE] [--dut_ssid DUT_SSID]  

[--dut_security DUT_SECURITY] [--dut_passwd DUT_PASSWD]  

[--dut_bssid DUT_BSSID] [--download_bps DOWNLOAD_BPS]  

[--upload_bps UPLOAD_BPS] [--side_a_pdu SIDE_A_PDU]  

[--side_b_pdu SIDE_B_PDU] [--runtime_sec RUNTIME_SEC]  

[--debug] [--prefix PREFIX]  

[--bringup_time BRINGUP_TIME]  

[--influx_user INFLUX_USER]  

[--influx_passwd INFLUX_PASSWORD] [--influx_db INFLUX_DB]  

[--influx_host INFLUX_HOST]  

[--monitor_interval MONITOR_INTERVAL]  

[--debug_log DEBUG_LOG] [--no_cleanup]  

[--local_lf_report_dir LOCAL_LF_REPORT_DIR]  

[--test_rig TEST_RIG] [--test_tag TEST_TAG]  

[--dut_hw_version DUT_HW_VERSION]  

[--dut_sw_version DUT_SW_VERSION]  

[--dut_model_num DUT_MODEL_NUM]  

[--dut_serial_num DUT_SERIAL_NUM]  

[--test_priority TEST_PRIORITY] [--test_id TEST_ID]  

[--csv_outfile CSV_OUTFILE] [--log_level LOG_LEVEL]  

[--lf_logger_config_json LF_LOGGER_CONFIG_JSON]  

[--help_summary]

-----
LANforge Unit Test: Connect Station to AP - sta_connect2.py
-----
Summary:  

This will create a station, create TCP and UDP traffic, run it a short amount of time, and verify whether traffic was sent and received. It also verifies the station connected to the requested BSSID if bssid is specified as an argument. The script will clean up the station and connections at the end of the test.
-----
CLI Example:  

./sta_connect2.py --mgr localhost --dut_ssid <ssid> --dut_passwd <passwd> --dut_security wpa2  

--upstream_port eth1 --radio wiphy1

CLI Example for kpi.csv report output:  

./sta_connect2.py --mgr localhost --dut_ssid <ssid> --dut_passwd <passwd> --dut_security wpa2  

--upstream_port eth2 --csv_outfile sta_connect2.csv --test_rig LF-Lab --test_tag L3 --dut_hw_version Linux  

--dut_model_num 1 --dut_sw_version 5.4.5 --dut_serial_num 1234

CLI Example for kpi.csv, variable tx/rx rates, and pdu size:  

./sta_connect2.py --mgr localhost --dut_ssid <ssid> --dut_passwd <passwd> --dut_security wpa2  

--upstream_port eth2 --download_bps 768000 --upload_bps 256000 --side_a_pdu 300 --side_b_pdu 750  

--csv_outfile sta_connect2.csv --test_rig LF-Lab --test_tag L3 --dut_hw_version Linux --dut_model_num 1  

--dut_sw_version 5.4.5 --dut_serial_num 1234

```

Note: --sta\_mode use values in second column

AUTO	0	# 802.11g
802.11a	1	# 802.11a
b	2	# 802.11b
g	3	# 802.11g
abg	4	# 802.11abg
abgn	5	# 802.11abgn
bgn	6	# 802.11bgn
bg	7	# 802.11bg
abgnAC	8	# 802.11abgn-AC
anAC	9	# 802.11an-AC
an	10	# 802.11an
bgnAC	11	# 802.11bgn-AC
abgnAX	12	# 802.11abgn-AX
		# a/b/g/n/AC/AX (dual-band AX) support
bgnAX	13	# 802.11bgn-AX
anAX	14	# 802.11an-AX
aAX	15	# 802.11a-AX (6E disables /n and /ac)

options:

```

-h, --help      show this help message and exit
-m MGR, --mgr MGR   address of the LANforge GUI machine (localhost is default)
-o PORT, --port PORT  IP Port the LANforge GUI is listening on (8080 is default)
-u USER, --user USER  TBD: credential login/username

```

```

-p PASSWD, --passwd PASSWD
    TBD: credential password
--resource RESOURCE  LANforge Station resource ID to use, default is 1
--upstream_resource UPSTREAM_RESOURCE
    LANforge Ethernet port resource ID to use, default is 1
--upstream_port UPSTREAM_PORT
    LANforge Ethernet port name, default is eth2
--radio RADIO  LANforge radio to use, default is wiphy0
--sta_mode STA_MODE  LANforge station-mode setting (see add_sta LANforge CLI documentation, default is 0 (auto))
--dut_ssid DUT_SSID  DUT SSID
--dut_security DUT_SECURITY
    DUT security: open, wpa, wpa2, wpa3, owe
--dut_passwd DUT_PASSWD
    DUT PSK password. Do not set for OPEN auth
--dut_bssid DUT_BSSID
    DUT BSSID to which we expect to connect.
--download_bps DOWNLOAD_BPS
    Set the minimum bps value on test endpoint A. Default: 25g000
--upload_bps UPLOAD_BPS
    Set the minimum bps value on test endpoint B. Default: 256000
--side_a_pdu SIDE_A_PDU
    Set the minimum pdu value on test endpoint A. Default: 1200
--side_b_pdu SIDE_B_PDU
    Set the minimum pdu value on test endpoint B. Default: 1500
--runtime_sec RUNTIME_SEC
    Set test duration time. Default: 60 seconds
--debug
    enable debugging
--prefix PREFIX
    Station prefix. Default: 'sta'
--bringup_time BRINGUP_TIME
    Seconds to wait for stations to associate and aquire IP. Default: 300
--influx_user INFLUX_USER
    Username for your Influx database
--influx_passwd INFLUX_PASSWD
    Password for your Influx database
--influx_db INFLUX_DB
    Name of your Influx database
--influx_host INFLUX_HOST
    Host of your influx database if different from the system you are running on
--monitor_interval MONITOR_INTERVAL
    How frequently you want to append to your database
--debug_log DEBUG_LOG
    Specify a file to send debug output to
--no_cleanup
    Do not cleanup before exit
--local_lf_report_dir LOCAL_LF_REPORT_DIR
    --local_lf_report_dir override the report path, primary use when running test in test suite
--test_rig TEST_RIG
    test rig for kpi.csv, testbed that the tests are run on
--test_tag TEST_TAG
    test tag for kpi.csv, test specific information to differenciate the test
--dut_hw_version DUT_HW_VERSION
    dut hw version for kpi.csv, hardware version of the device under test
--dut_sw_version DUT_SW_VERSION
    dut sw version for kpi.csv, software version of the device under test
--dut_model_num DUT_MODEL_NUM
    dut model for kpi.csv, model number / name of the device under test
--dut_serial_num DUT_SERIAL_NUM
    dut serial for kpi.csv, serial number / serial number of the device under test
--test_priority TEST_PRIORITY
    dut model for kpi.csv, test-priority is arbitrary number
--test_id TEST_ID
    test-id for kpi.csv, script or test name
--csv_outfile CSV_OUTFILE
    --csv_outfile <prepend input to generated file for csv data>
--log_level LOG_LEVEL
    Set logging level: debug | info | warning | error | critical
--lf_logger_config_json LF_LOGGER_CONFIG_JSON
    --lf_logger_config_json <json file>, json configuration of logger
--help_summary
    Show summary of what this script does

-----
CLI Example:
./sta_connect2.py --mgr localhost --dut_ssid <ssid> --dut_passwd <passwd> --dut_security wpa2
--upstream_port eth2 --radio wiphy0

```

py-scripts/test\_I3.py

Example report: test\_I3.pdf

```

usage: test_I3.py [-h] [--local_lf_report_dir LOCAL_LF_REPORT_DIR]
                  [--results_dir_name RESULTS_DIR_NAME] [--test_rig TEST_RIG]
                  [--test_tag TEST_TAG] [--dut_hw_version DUT_HW_VERSION]
                  [--dut_sw_version DUT_SW_VERSION]
                  [--dut_model_num DUT_MODEL_NUM]
                  [--dut_serial_num DUT_SERIAL_NUM]
                  [--test_priority TEST_PRIORITY] [--test_id TEST_ID]
                  [-o CSV_OUTFILE] [--tty TTY] [--baud BAUD] [--mgr LFMGR]
                  [--mgr_port LFMGR_PORT] [--test_duration TEST_DURATION]
                  [--tos TOS] [--debug] [--log_level LOG_LEVEL]
                  [--interopt_mode] [-t ENDP_TYPE] [-u UPSTREAM_PORT]
                  [--downstream_port DOWNSTREAM_PORT]
                  [--polling_interval POLLING_INTERVAL] [-r RADIO]
                  [-amr SIDE_A_MIN_BPS] [-amp SIDE_A_MIN_PDU]
                  [-bmr SIDE_B_MIN_BPS] [-bpm SIDE_B_MIN_PDU]
                  [--rates_are_totals] [--multiconn MULTICONN]
                  [--attenuators ATTENUATORS] [--atten_vals ATTN_VALS]
                  [--wait WAIT] [--sta_start_offset STA_START_OFFSET]
                  [--no_pre_cleanup] [--no_cleanup] [--cleanup_cx]
                  [--csv_data_to_report] [--no_stop_traffic] [--quiesce_cx]
                  [--use_existing_station_list]
                  [--existing_station_list EXISTING_STATION_LIST]
                  [--wait_for_ip_sec WAIT_FOR_IP_SEC] [--exit_on_ip_acquired]
                  [--lf_logger_config_json LF_LOGGER_CONFIG_JSON] [--ap_read]
                  [--ap_module AP_MODULE] [--ap_test_mode AP_TEST_MODE]
                  [--ap_scheme {serial,telnet,ssh,mux_serial}]
                  [--ap_serial_port AP_SERIAL_PORT]
                  [--ap_serial_baud AP_SERIAL_BAUD] [--ap_ip AP_IP]
                  [--ap_ssh_port AP_SSH_PORT]
                  [--ap_telnet_port AP_TELNET_PORT] [--ap_user AP_USER]
                  [--ap_passwd AP_PASSWORD] [--ap_if_2g AP_IF_2G]
                  [--ap_if_5g AP_IF_5G] [--ap_if_6g AP_IF_6G]
                  [--ap_file AP_FILE] [--ap_band_list AP_BAND_LIST]

```

```

[--help_summary]

NAME: test_13.py

PURPOSE: The Layer 3 Traffic Generation Test is designed to test the performance of the Access Point by running layer-3
Cross-Connect Traffic. Layer-3 Cross-Connects represent a stream of data flowing through the system under test.
A Cross-Connect (CX) is composed of two Endpoints, each of which is associated with a particular Port (physical or virtual)

The test will create stations, create cx traffic between upstream port and stations, run traffic. Verify
the traffic is being transmitted and received

* Supports creating user-specified amount stations on multiple radios
* Supports configuring upload and download requested rates and PDU sizes.
* Supports generating connections with different ToS values.
* Supports generating tcp and/or UDP traffic types.
* Supports iterating over different PDU sizes
* Supports iterating over different requested tx rates (configurable as total or per-connection value)
* Supports iterating over attenuation values.
* Supports testing connection between two ethernet connection - L3 dataplane

Generic command layout:
-----
./test_13.py --mgr <ip_address> --test_duration <duration> --endp_type <traffic types> --upstream_port <port>
--radio "radio==<radio> stations==<number stations> ssid==<ssid> ssid_pw==<ssid password>
security==<security type: wpa2, open, wpa3>" --debug

EXAMPLE:

#####
# Examples
#####
Example running traffic with two radios
1. Test duration 30 minutes
2. Traffic IPv4 TCP, UDP
3. Upstream-port eth2
4. Radio #0 wiphy0 has 1 station, ssid = ssid_2g, ssid password = ssid_pw_2g security = wpa2
5. Radio #1 wiphy1 has 2 stations, ssid = ssid_5g, ssid password = BLANK security = open
6. Create connections with TOS of BK and VI

    # The script now supports multiple radios, each specified with an individual --radio switch.

    # Intercept example Creating stations
    Intercept testing creating stations
    ./test_13.py --lfmgr 192.168.0.103           --test_duration 60s          --polling_interval 5s      --upstr
    ./test_13.py --lfmgr 192.168.0.103           --local_lf_report_dir /home/lanforge/html-reports/ct_id_004      --

    # Intercept using existing stations
    Intercept testing creating stations
    ./test_13.py --lfmgr 192.168.91.50          --test_duration 60s          --polling_interval 5s      --upstr

    * UDP and TCP bi-directional test, no use of controller.
    ./test_13.py --mgr 192.168.200.83 --endp_type 'lf_udp,lf_tcp' --upstream_port 1.1.eth1
    --radio "radio==1.1.wiphy0 stations==5 ssid==Netgear2g ssid_pw==lanforge security==wpa2"
    --radio "radio==1.1.wiphy1 stations==1 ssid==Netgear5g ssid_pw==lanforge security==wpa2"
    --test_duration 60s

    * Port resets, chooses random value between min and max
    ./test_13.py --lfmgr 192.168.200.83 --test_duration 90s --polling_interval 10s --upstream_port eth1
    --radio 'radio==wiphy0,stations==4,ssid==Netgear2g,ssid_pw==lanforge,security==wpa2,reset_port_enable==TRUE,
    reset_port_time_min==10s,reset_port_time_max==20s' --endp_type lf_udp --rates_are_totals --side_a_min_bps=20000
    --side_b_min_bps=300000000

    # Command: (remove carriage returns)
    ./test_13.py --lfmgr 192.168.200.83 --test_duration 30s --endp_type "lf_tcp,lf_udp" --tos "BK VI" --upstream_port 1.1
    --radio "radio==1.1.wiphy0 stations==1 ssid==Netgear2g ssid_pw==lanforge security==wpa2"

    # Have the stations continue to run after the completion of the script
    ./test_13.py --lfmgr 192.168.200.83 --endp_type 'lf_udp,lf_tcp' --tos BK --upstream_port 1.1.eth1
    --radio 'radio==wiphy0 stations==2 ssid==Netgear2g ssid_pw==lanforge security==wpa2' --test_duration 30s
    --polling_interval 5s --side_a_min_bps 256000 --side_b_min_bps 102400000 --no_stop_traffic

    # Have script use existing stations from previous run where traffic was not stopped and also create new stations and leave
    ./test_13.py --lfmgr 192.168.200.83 --endp_type 'lf_udp,lf_tcp' --tos BK --upstream_port 1.1.eth1
    --radio 'radio==wiphy0 stations==2 ssid==Netgear2g ssid_pw==lanforge security==wpa2' --sta_start_offset 1000
    --test_duration 30s --polling_interval 5s --side_a_min_bps 256000 --side_b_min_bps 102400000 --use_existing_station_l
    --existing_station_list '1.1.sta0000,1.1.sta0001,1.1.sta0002' --no_stop_traffic

    # Have script use wifi_settings enable flags :: wifi_settings==wifi_settings,enable_flags==(ht160_enable&&wpa2_enable&
    ./test_13.py --lfmgr 192.168.200.83 --test_duration 20s --polling_interval 5s --upstream_port 1.1.eth1
    --radio 'radio==1.1.wiphy0,stations==1,ssid==Netgear2g,ssid_pw==lanforge,security==wpa2,wifi_mode==0,wifi_settings==w'
    --radio 'radio==1.1.wiphy1,stations==1,ssid==Netgear5g,ssid_pw==lanforge,security==wpa2,wifi_mode==0,wifi_settings==w'
    --endp_type lf_udp --rates_are_totals --side_a_min_bps=20000 --side_b_min_bps=300000000 --test_rig ID_003 --test_tag
    --dut_hw_version 1.0 --dut_serial_num 12345678 --log_level debug

    # Setting wifi_settings per radio
    ./test_13.py
    --lfmgr 192.168.100.116
    --local_lf_report_dir /home/lanforge/html-reports/
    --test_duration 15s
    --polling_interval 5s
    --upstream_port eth2
    --radio "radio==wiphy1 stations==4 ssid==asus1lax-5 ssid_pw==hello123 security==wpa2 mode==0 wifi_settings==wifi_sett
    --endp_type lf_udp
    --rates_are_totals
    --side_a_min_bps=20000
    --side_b_min_bps=300000000
    --test_rig CT-US-001
    --test_tag 'test_13'

    # Example : LAN-1927 WPA2-TLS-Configuration
    ./test_13.py
    --lfmgr 192.168.0.103
    --test_duration 20s
    --polling_interval 5s
    --upstream_port 1.1.eth2

```

```

--radio 'radio==wiphy1,stations==1,ssid==ax88u_5g,ssid_pw==[BLANK],security==wpa2,wifi_settings==wifi_settings,wifi_m
--endp_type lf_udp
--rates_are_totals
--side_a_min_bps=256000
--side_b_min_bps=300000000
--test_rig ID_003
--test_tag 'test_13'
--dut_model_num GT-AXE11000
--dut_sw_version 3.0.0.4.386_44266
--dut_hw_version 1.0
--dut_serial_num 12345678
--log_level debug

# Example : LAN-1927 WPA2-TTLS-Configuration
./test_13.py
--lfmgr 192.168.0.103
--test_duration 20s
--polling_interval 5s
--upstream_port 1.1.eth2
--radio 'radio==wiphy1,stations==1,ssid==ax88u_5g,ssid_pw==[BLANK],security==wpa2,wifi_settings==wifi_settings,wifi_m
--endp_type lf_udp
--rates_are_totals
--side_a_min_bps=256000
--side_b_min_bps=300000000
--test_rig ID_003
--test_tag 'test_13'
--dut_model_num GT-AXE11000
--dut_sw_version 3.0.0.4.386_44266
--dut_hw_version 1.0
--dut_serial_num 12345678
--log_level debug

# Example : LAN-1927 WPA3-TTLS-Configuration
./test_13.py
--lfmgr 192.168.0.103
--test_duration 20s
--polling_interval 5s
--upstream_port 1.1.eth2
--radio 'radio==wiphy1,stations==1,ssid==ax88u_5g,ssid_pw==[BLANK],security==wpa3,wifi_settings==wifi_settings,wifi_m
--endp_type lf_ud
--rates_are_totals
--side_a_min_bps=256000
--side_b_min_bps=300000000
--test_rig ID_003
--test_tag 'test_13'
--dut_model_num GT-AXE11000
--dut_sw_version 3.0.0.4.386_44266
--dut_hw_version 1.0
--dut_serial_num 12345678
--log_level debug

# Example : LAN-1927 WPA3-TLS-Configuration
./test_13.py
--lfmgr 192.168.0.103
--test_duration 20s
--polling_interval 5s
--upstream_port 1.1.eth2
--radio 'radio==wiphy1,stations==1,ssid==ax88u_5g,ssid_pw==[BLANK],security==wpa3,wifi_settings==wifi_settings,wifi_m
--endp_type lf_udp
--rates_are_totals
--side_a_min_bps=256000
--side_b_min_bps=300000000
--test_rig ID_003
--test_tag 'test_13'
--dut_model_num GT-AXE11000
--dut_sw_version 3.0.0.4.386_44266
--dut_hw_version 1.0
--dut_serial_num 12345678
--log_level debug

SCRIPT_CLASSIFICATION: Creation & Runs Traffic
SCRIPT_CATEGORIES: Performance, Functional, KPI Generation, Report Generation
NOTES:
#####
# Command switches
#####

--mgr <hostname for where LANforge GUI is running>,default='localhost'
-d / --test_duration <how long to run> example --time 5d (5 days) default: 3m options: number followed by d, h, m or s',default=
--tos: Support different ToS settings: BK | BE | VI | VO | numeric',default="BE"
--debug: Enable debugging',default=False
-t / --endp_type <types of traffic> example --endp_type "lf_udp lf_tcp mc_udp" Default: lf_udp , options: lf_udp, lf_udp6, lf_tc
	default='lf_udp', type=valid_endp_types
-u / --upstream_port <cross connect upstream_port> example: --upstream_port eth1',default='eth1'
-o / --outfile <Output file for csv data>", default='longevity_results'

<duration>: number followed by one of the following
d - days
h - hours
m - minutes
s - seconds

<traffic type>:
lf_udp : IPv4 UDP traffic
lf_tcp : IPv4 TCP traffic
lf_udp6 : IPv6 UDP traffic
lf_tcp6 : IPv6 TCP traffic
mc_udp : IPv4 multi cast UDP traffic
mc_udp6 : IPv6 multi cast UDP traffic

<tos>:
BK, BE, VI, VO: Optional wifi related Tos Settings. Or, use your preferred numeric values. Cross connects type of service
* Data 0 (Best Effort, BE): Medium priority queue, medium throughput and delay.

```

```

    Most traditional IP data is sent to this queue.
* Data 1 (Background, BK): Lowest priority queue, high throughput. Bulk data that requires maximum throughput and
    is not time-sensitive is sent to this queue (FTP data, for example).
* Data 2 (Video, VI): High priority queue, minimum delay. Time-sensitive data such as Video and other streaming
    media are automatically sent to this queue.
* Data 3 (Voice, VO): Highest priority queue, minimum delay. Time-sensitive data such as Voice over IP (VoIP)
    is automatically sent to this Queue.

```

```

<wifi_mode>:
    Input      : Enum Val   : Shown by nc_show_ports

    AUTO       | 0          # 802.11
    802.11a   | 1          # 802.11a
    b          | 2          # 802.11b
    g          | 3          # 802.11g
    abg        | 4          # 802.11abg
    abgn      | 5          # 802.11abgn
    bgn       | 6          # 802.11bgn
    bg         | 7          # 802.11bg
    abgnAC    | 8          # 802.11abgn-AC
    anAC      | 9          # 802.11an-AC
    an        | 10         # 802.11an
    bgnAC     | 11         # 802.11bgn-AC
    abgnAX    | 12         # 802.11abgn-AX
                           # a/b/g/n/AC/AX (dual-band AX) support
    bgnAX     | 13         # 802.11bgn-AX
    anAX      | 14         # 802.11an-AX
    aAX       | 15         # 802.11a-AX (6E disables /n and /ac)

wifi_settings flags are currently defined as:
    wpa_enable           | 0x10          # Enable WPA
    # Use Custom wpa_supplicant config file.
    custom_conf          | 0x20
    # Use wpa_supplicant configured for WEP encryption.
    wep_enable           | 0x200
    # Use wpa_supplicant configured for WPA2 encryption.
    wpa2_enable          | 0x400
    # Disable HT-40 even if hardware and AP support it.
    ht40_disable         | 0x800
    # Enable SCAN-SSID flag in wpa_supplicant.
    scan_ssid            | 0x1000
    # Use passive scanning (don't send probe requests).
    passive_scan         | 0x2000
    disable_sgi          | 0x4000          # Disable SGI (Short Guard Interval).
    # OK-To-Migrate (Allow station migration between LANforge radios)
    lf_sta_migrate      | 0x8000
    # Verbose-Debug: Increase debug info in wpa-suppliant and hostapd logs.
    verbose              | 0x10000
    # Enable 802.11u (Interworking) feature.
    80211u_enable        | 0x20000
    # Enable 802.11u (Interworking) Auto-internetworking feature. Always enabled currently.
    80211u_auto          | 0x40000
    # AP Provides access to internet (802.11u Interworking)
    80211u_gw            | 0x80000
    # AP requires additional step for access (802.11u Interworking)
    80211u_additional    | 0x100000
    # AP claims emergency services reachable (802.11u Interworking)
    80211u_e911          | 0x200000
    # AP provides Unauthenticated emergency services (802.11u Interworking)
    80211u_e911_unauth   | 0x400000
    # Enable Hotspot 2.0 (HS20) feature. Requires WPA-2.
    hs20_enable           | 0x800000
    # AP: Disable DGAF (used by HotSpot 2.0).
    disable_gdaf          | 0x10000000
    8021x_radius          | 0x20000000          # Use 802.1x (RADIUS for AP).
    # Enable opportunistic PMSKA caching for WPA2 (Related to 802.11r).
    80211r_pmska_cache   | 0x40000000
    # Disable HT80 (for AC chipset NICs only)
    disable_ht80          | 0x80000000
    ibss_mode             | 0x20000000          # Station should be in IBSS mode.
    # Enable OSEN protocol (OSU Server-only Authentication)
    osen_enable            | 0x40000000
    # Disable automatic station roaming based on scan results.
    disable_roam           | 0x80000000
    ht160_enable          | 0x1000000000          # Enable HT160 mode.
    # Disable fast_reauth option for virtual stations.
    disable_fast_reauth    | 0x2000000000
    mesh_mode              | 0x4000000000          # Station should be in MESH mode.
    # Station should enable power-save. May not work in all drivers/configurations.
    power_save_enable      | 0x8000000000
    create_admin_down       | 0x100000000000          # Station should be created admin-down.
    # WDS station (sort of like a lame mesh), not supported on ath10k
    wds-mode               | 0x200000000000
    # Do not include supported-oper-class-IE in assoc requests. May work around AP bugs.
    no-supp-op-class-ie   | 0x400000000000
    # Enable/disable tx-offloads, typically managed by set_wifi_txo command
    txo-enable             | 0x800000000000
    use-wpa3                | 0x100000000000          # Enable WPA-3 (SAE Personal) mode.
    use-bss-transition       | 0x800000000000          # Enable BSS transition.
    disable-twt              | 0x1000000000000000          # Disable TWT mode

For wifi_extra_keys syntax :
    telnet <lanforge ip> 4001
    type: help set_wifi_extra
    wifi_extra keys:
        key_mgmt (Key Management)
        pairwise (Pairwise Ciphers)
        group   (Group Ciphers)
        psk     (WPA PSK)
        wep_key
        ca_cert (CA Cert File)
        eap     (EAP Methods) EAP method: MD5, MSCHAPV2, OTP, GTC, TLS, PEAP, TTLS. (note different the GUI no appended EA
        identity (EAP Identity)
        anonymous_identity (EAP Anon Identity)
        phase1 (Phase-1)
        phase2 (Phase-2)
        passwd (EAP Password)

```

```

pin (EAP Pin)
pac_file (PAC file)
private_key (Private Key)
pk_password (PK Password)
hessid="00:00:00:00:00:00"
realm (Realm)
client_cert (Client Cert)
imsi (IMSI)
milenage (Milenage)
domain (Domain)
roaming_consortium (Consortium)
venue_group ()
network_type (Network Auth)
ipaddr_type_avail ()
network_auth_type ()
angp_3gpp_cell_net ()

ieee80211w : 0,1,2

Multicast traffic :
    Multicast traffic default IGMP Address in the range of 224.0.0.0 to 239.255.255.255,
    so I have provided 224.9.9.9 as IGMP address and IGMP Dest port as 9999 and MIN-IP PORT as 9999.
    these values must be same on the eth1(server side) and client side, then the traffic will run.

=====
** FURTHER INFORMATION **
Using the layer3_cols flag:

Currently the output function does not support inputting the columns in layer3_cols the way they are displayed in the GUI. This
certain columns in the GUI in your final report, please match the according GUI column display to it's counterpart to have the
your report.

GUI Column Display      Layer3_cols argument to type in (to print in report)

Name          | 'name'
EID           | 'eid'
Run           | 'run'
Mng           | 'mng'
Script         | 'script'
Tx Rate       | 'tx rate'
Tx Rate (1 min) | 'tx rate (1 min)'
Tx Rate (last)  | 'tx rate (last)'
Tx Rate LL     | 'tx rate ll'
Rx Rate        | 'rx rate'
Rx Rate (1 min) | 'rx rate (1 min)'
Rx Rate (last)  | 'rx rate (last)'
Rx Rate LL     | 'rx rate ll'
Rx Drop %      | 'rx drop %'
Tx PDUs        | 'tx pdus'
Tx Pkts LL     | 'tx pkts ll'
PDU/s TX       | 'pdu/s tx'
Pps TX LL      | 'pps tx ll'
Rx PDUs        | 'rx pdus'
Rx Pkts LL     | 'pps rx ll'
PDU/s RX       | 'pdu/s tx'
Pps RX LL      | 'pps rx ll'
Delay          | 'delay'
Dropped        | 'dropped'
Jitter          | 'jitter'
Tx Bytes        | 'tx bytes'
Rx Bytes        | 'rx bytes'
Replays         | 'replays'
TCP Rtx         | 'tcp rtx'
Dup Pkts        | 'dup pkts'
Rx Dup %        | 'rx dup %'
OOO Pkts        | 'ooo pkts'
Rx OOO %        | 'rx ooo %'
RX Wrong Dev   | 'rx wrong dev'
CRC Fail        | 'crc fail'
RX BER          | 'rx ber'
CX Active       | 'cx active'
CX Estab/s     | 'cx estab/s'
1st RX          | '1st rx'
CX TO           | 'cx to'
Pattern         | 'pattern'
Min PDU         | 'min pdu'
Max PDU         | 'max pdu'
Min Rate        | 'min rate'
Max Rate        | 'max rate'
Send Buf        | 'send buf'
Rcv Buf         | 'rcv buf'
CWND            | 'cwnd'
TCP MSS          | 'tcp mss'
Bursty          | 'bursty'
A/B             | 'a/b'
Elapsed          | 'elapsed'
Destination Addr | 'destination addr'
Source Addr     | 'source addr'

Using the port_mgr_cols flag:
'4way time (us)'
'activity'
'alias'
'angp time (us)'
'ap'
'beacon'
'bps rx'
'bps rx ll'
'bps tx'
'bps tx ll'
'bytes rx ll'
'bytes tx ll'
'channel'
'collisions'
'connections'
'crypt'
'cx ago'

```

```

'cx time (us)'
'device'
'dhcp (ms)'
'down'
'entity id'
'gateway ip'
'ip'
'ipv6 address'
'ipv6 gateway'
'key/phrase'
'login-fail'
'login-ok'
'logout-fail'
'logout-ok'
'mac'
'mask'
'misc'
'mode'
'mtu'
'no cx (us)'
'noise'
'parent dev'
'phantom'
'port'
'port type'
'pps rx'
'pps tx'
'qlen'
'reset'
'retry failed'
'rx bytes'
'rx crc'
'rx drop'
'rx errors'
'rx fifo'
'rx frame'
'rx length'
'rx miss'
'rx over'
'rx pkts'
'rx-rate'
'sec'
'signal'
'ssid'
'status'
'time-stamp'
'tx abort'
'tx bytes'
'tx crr'
'tx errors'
'tx fifo'
'tx hb'
'tx pkts'
'tx wind'
'tx-failed %'
'tx-rate'
'wifi retries'

```

Can't decide what columns to use? You can just use 'all' to select all available columns from both tables.

STATUS: Functional

VERIFIED\_ON: 18-JULY-2023,  
 GUI Version: 5.4.6  
 Kernel Version: 5.19.17+

LICENSE:

Free to distribute and modify. LANforge systems must be licensed.  
 Copyright 2023 Candela Technologies Inc

INCLUDE\_IN\_README: False

options:

-h, --help show this help message and exit  
 --help\_summary Show summary of what this script does

arguments defined in test\_l3.py file:

```

--local_lf_report_dir LOCAL_LF_REPORT_DIR
                    --local_lf_report_dir override the report path (lanforge/html-reports), primary used when making another d
--results_dir_name RESULTS_DIR_NAME
                    the name of the directory that contains the output from the test /lanforge/html-reports/<results_dir_name>
--test_rig TEST_RIG test rig for kpi.csv, testbed that the tests are run on
--test_tag TEST_TAG test tag for kpi.csv, test specific information to differentiate the test
--dut_hw_version DUT_HW_VERSION
                    dut hw version for kpi.csv, hardware version of the device under test
--dut_sw_version DUT_SW_VERSION
                    dut sw version for kpi.csv, software version of the device under test
--dut_model_num DUT_MODEL_NUM
                    dut model for kpi.csv, model number / name of the device under test
--dut_serial_num DUT_SERIAL_NUM
                    dut serial for kpi.csv, serial number / serial number of the device under test
--test_priority TEST_PRIORITY
                    dut model for kpi.csv, test-priority is arbitrary number
--test_id TEST_ID test-id for kpi.csv, script or test name
-o CSV_OUTFILE, --csv_outfile CSV_OUTFILE
                    --csv_outfile <output file for csv data>
--tty TTY
                    --tty "/dev/ttyUSB2" the serial interface to the AP
--baud BAUD
                    --baud "9600" AP baud rate for the serial interface
--mgr LFMGR, --lfmgr LFMGR
                    --lfmgr <hostname for where LANforge GUI is running>
--mgr_port LFMGR_PORT, --lfmgr_port LFMGR_PORT
                    --lfmgr_port <port LANforge GUI HTTP service is running on>
--test_duration TEST_DURATION
                    --test_duration <how long to run> example --time 5d (5 days) default: 3m options: number followed by d, h
--tos TOS
                    --tos: Support different ToS settings: BK,BE,VI,VO,numeric

```

```

--debug           --debug this will enable debugging in py-json method
--log_level LOG_LEVEL
                  Set logging level: debug | info | warning | error | critical
--interopt_mode
                  For Interopt continue to try running even if some clients do not get an IP.
-t ENDP_TYPE, --endp_type ENDP_TYPE
                  --endp_type <types of traffic> example --endp_type "lf_udp lf_tcp mc_udp" Default: lf_udp , options: lf_u
-u UPSTREAM_PORT, --upstream_port UPSTREAM_PORT
                  --upstream_port <cross connect upstream_port> example: --upstream_port ethl
--downstream_port DOWNSTREAM_PORT
                  --downstream_port <cross connect downstream_port> for use when downstream is ethernet (eth to eth connect
--polling_interval POLLING_INTERVAL
                  --polling_interval <seconds>
-r RADIO, --radio RADIO
                  --radio "radio==<number of wiphy stations==<number of stations> ssid==<ssid> ssid_pw==<ssid password> sec
-amr SIDE_A_MIN_BPS, --side_a_min_bps SIDE_A_MIN_BPS, --upload_min_bps SIDE_A_MIN_BPS
                  --side_a_min_bps, requested downstream min tx rate at stations / client, comma separated list for multiple
                  When running with tcp/udp traffic along with mcast , mcast will ignore the upload value
-amp SIDE_A_MIN_PDU, --side_a_min_pdu SIDE_A_MIN_PDU
                  --side_a_min_pdu, downstream pdu size, comma separated list for multiple iterations. Default MTU
-bmr SIDE_B_MIN_BPS, --download_min_bps SIDE_B_MIN_BPS, --side_b_min_bps SIDE_B_MIN_BPS, --do SIDE_B_MIN_BPS
                  --side_b_min_bps or --download_min_bps, requested upstream min tx rate, comma separated list for multiple
                  When runnign with tcp/udp and mcast will use this value
-bmp SIDE_B_MIN_PDU, --side_b_min_pdu SIDE_B_MIN_PDU
                  --side_b_min_pdu, upstream pdu size, comma separated list for multiple iterations. Default MTU
--rates_are_totals Treat configured rates as totals instead of using the un-modified rate for every connection.
--multiconn MULTICONN
                  Configure multi-conn setting for endpoints. Default is 1 (auto-helper is enabled by default as well).
--attenuators ATTENATORS
                  --attenuators, comma separated list of attenuator module eids: shelf.resource.attenu-serno.attenu-idx
--atten_vals ATTN_VALS
                  --atten_vals, comma separated list of attenuator settings in ddb units (1/10 of db)
--wait WAIT
                  --wait <time>, time to wait at the end of the test
--sta_start_offset STA_START_OFFSETSET
                  Station start offset for building stations
--no_pre_cleanup Do not pre cleanup stations on start
--no_cleanup Do not cleanup before exit
--cleanup_cx cleanup cx before exit
--csv_data_to_report collected interval data in csv for each cx will be put in report
--no_stop_traffic leave traffic running
--quiesce_cx --quiesce store true, allow the cx to drain then stop so as to not have rx drop pkts
--use_existing_station_list
                  --use_station_list ,full eid must be given,the script will use stations from the list, no configuration on
--existing_station_list EXISTING_STATION_LIST
                  --station_list [list of stations] , use the stations in the list , multiple station lists may be entered
--wait_for_ip_sec WAIT_FOR_IP_SEC
                  --wait_for_ip_sec <seconds> default : 120s
--exit_on_ip_acquired
                  --exit_on_ip_acquired store true
--lf_logger_config_json LF_LOGGER_CONFIG_JSON
                  --lf_logger_config_json <json file> , json configuration of logger
--ap_read
                  --ap_read flag present enable reading ap
--ap_module AP_MODULE
                  series module
--ap_test_mode AP_TEST_MODE
                  --ap_mode
--ap_scheme {serial,telnet,ssh,mux_serial}
                  --ap_scheme '/dev/ttyUSB0'
--ap_serial_port AP_SERIAL_PORT
                  --ap_serial_port '/dev/ttyUSB0'
--ap_serial_baud AP_SERIAL_BAUD
                  --ap_baud '115200', default='115200
--ap_ip AP_IP
                  --ap_ip
--ap_ssh_port AP_SSH_PORT
                  --ap_ssh_port
--ap_telnet_port AP_TELNET_PORT
                  --ap_telnet_port
--ap_user AP_USER
                  --ap_user , the user name for the ap, default = lanforge
--ap_passwd AP_PASSWD
                  --ap_passwd, the password for the ap default = lanforge
--ap_if_2g AP_IF_2G
--ap_if_5g AP_IF_5G
--ap_if_6g AP_IF_6G
--ap_file AP_FILE
--ap_band_list AP_BAND_LIST
                  --ap_band_list '2g,5g,6g' supported bands

      Useful Information:
      1. Polling interval for checking traffic is fixed at 1 minute
      2. The test will generate csv file
      3. The tx/rx rates are fixed at 256000 bits per second
      4. Maximum stations per radio based on radio

```

#### py-scripts/test\_l3\_longevity.py

```

usage: test_l3_longevity.py [-h] [--help_summary]
                           [--local_lf_report_dir LOCAL_LF_REPORT_DIR]
                           [--test_rig TEST_RIG] [--test_tag TEST_TAG]
                           [--dut_hw_version DUT_HW_VERSION]
                           [--dut_sw_version DUT_SW_VERSION]
                           [--dut_model_num DUT_MODEL_NUM]
                           [--dut_serial_num DUT_SERIAL_NUM]
                           [--test_priority TEST_PRIORITY]
                           [--test_id TEST_ID] [-o CSV_OUTFILE] [--tty TTY]
                           [--baud BAUD] [--mgr LFMGR]
                           [--mgr_port LFMGR_PORT]
                           [--test_duration TEST_DURATION] [--tos TOS]
                           [--debug] [-t ENDP_TYPE] [-u UPSTREAM_PORT]
                           [--downstream_port DOWNSTREAM_PORT]
                           [--polling_interval POLLING_INTERVAL] [-r RADIO]
                           [--collect_layer3_data] [--ap_read]
                           [--ap_scheme {serial,telnet,ssh,mux_serial}]
                           [--ap_port AP_PORT] [--ap_baud AP_BAUD]
                           [--ap_ip AP_IP] [--ap_ssh_port AP_SSH_PORT]
                           [--ap_user AP_USER] [--ap_passwd AP_PASSWD]
                           [--ap_if_2g AP_IF_2G] [--ap_if_5g AP_IF_5G]
                           [--ap_if_6g AP_IF_6G] [--ap_cmd_6g AP_CMD_6G]

```

```

[--ap_cmd_5g AP_CMD_5G] [--ap_cmd_2g AP_CMD_2G]
[--ap_cmd_6g AP_CMD_6G]
[--ap_cmd_5g AP_CMD_5G]
[--ap_cmd_2g AP_CMD_2G]
[--ap_chanim_cmd_6g AP_CHANIM_CMD_6G]
[--ap_chanim_cmd_5g AP_CHANIM_CMD_5G]
[--ap_chanim_cmd_2g AP_CHANIM_CMD_2G]
[--ap_scheduler_stats] [-ap_ofdma_stats]
[--ap_test_model [-amr SIDE_A_MIN_BPS]
[-amp SIDE_A_MIN_PDU] [-bmr SIDE_B_MIN_BPS]
[-bmp SIDE_B_MIN_PDU] [--rates_are_totals]
[--multiconn MULTICONN]
[--attenuators ATTENATORS]
[--atten_vals ATTEM_VALS]
[--influx_host INFLUX_HOST]
[--influx_port INFLUX_PORT]
[--influx_org INFLUX_ORG]
[--influx_token INFLUX_TOKEN]
[--influx_bucket INFLUX_BUCKET]
[--influx_tag INFLUX_TAG INFLUX_TAG]
[--cap_ctl_out] [--wait WAIT]
[--sta_start_offset STA_START_OFFSET]
[--no_pre_cleanup] [--no_cleanup]
[--no_stop_traffic] [-use_existing_station_list]
[--existing_station_list EXISTING_STATION_LIST]
[--wait_for_ip_sec WAIT_FOR_IP_SEC]
[--log_level LOG_LEVEL]
[--lf_logger_config_json LF_LOGGER_CONFIG_JSON]

test_13_longevity.py:
-----
Summary :
-----
The Layer 3 Traffic Generation Test is designed to test the performance of the Access Point by running layer 3 Cross-Connect Traffic. Layer-3 Cross-Connects represent a stream of data flowing through the system under test. A Cross-Connect (CX) is composed of two Endpoints, each of which is associated with a particular Port (physical or virtual interface).

The test will create stations, create cx traffic between upstream port and stations, run traffic and generate a report.

Generic command layout:
-----
./test_13_longevity.py --mgr <ip_address> --test_duration <duration> --endp_type <traffic types>
--upstream_port <shelf>.<resource>.<port>
--radio "radio==<shelf>.<resource>.<radio> stations==<number stations> ssid==<ssid> ssid_pw==<ssid password>
security==<security type: wpa2, open, wpa3>" --debug

Multiple radios may be entered with individual --radio switches

# UDP bi-directional test, no use of controller.
./test_13_longevity.py --mgr localhost --endp_type 'lf_udp lf_tcp' --upstream_port 1.1.eth1
--radio "radio==1.1.wiphy0 stations==10 ssid==ASUS_70 ssid_pw=[BLANK] security==open"
--radio "radio==1.1.wiphy2 stations==1 ssid==ASUS_70 ssid_pw=[BLANK] security==open" --test_duration 30s

# Port resets, chooses random value between min and max
./test_13_longevity.py --lfmigr LF_MGR_IP --test_duration 90s --polling_interval 10s --upstream_port 1.1.eth2
--radio 'radio==1.1.wiphy1,stations==4,ssid==SSID_USED,ssid_pw==SSID_PW_USED,security==SECURITY_USED,
reset_port_enable==TRUE,reset_port_time_min==10s,reset_port_time_max==20s'
--endp_type lf_udp --rates_are_totals --side_a_min_bps=20000 --side_b_min_bps=300000000"

<duration>: number followed by one of the following
d - days
h - hours
m - minutes
s - seconds

<traffic type>:
lf_udp : IPv4 UDP traffic
lf_tcp : IPv4 TCP traffic
lf_udp6 : IPv6 UDP traffic
lf_tcp6 : IPv6 TCP traffic
mc_udp : IPv4 multi cast UDP traffic
mc_udp6 : IPv6 multi cast UDP traffic

<tos>:
BK, BE, VI, VO: Optional wifi related Tos Settings. Or, use your preferred numeric values.

#####
# Command switches
#####

--mgr <hostname for where LANforge GUI is running>, default='localhost'
-d / --test_duration <how long to run> example --time 5d (5 days) default: 3m options: number followed by d, h, m or s', default=
--tos: Support different ToS settings: BK | BE | VI | VO | numeric', default="BE"
--debug: Enable debugging', default=False
-t / --endp_type <types of traffic> example --endp_type "lf_udp lf_tcp mc_udp" Default: lf_udp , options: lf_udp, lf_udp6, lf_tc
	default="lf_udp", type=valid_endp_types
-u / --upstream_port <cross connect upstream_port> example: --upstream_port eth1', default='eth1'
-o / --outfile <Output file for csv data>", default='longevity_results'

#####
# Examples
#####
Example #1 running traffic with two radios
1. Test duration 30 minutes
2. Traffic IPv4 TCP, UDP
3. Upstream-port eth2
4. Radio #0 wiphy0 has 1 station, ssid = ssid_2g, ssid password = ssid_pw_2g security = wpa2
5. Radio #1 wiphy1 has 2 stations, ssid = ssid_5g, ssid password = BLANK security = open

Command:
python3 ./test_13_longevity.py --test_duration 30s --endp_type "lf_tcp lf_udp" --tos "BK VI" --upstream_port eth2
--radio "radio==wiphy0 stations==1 ssid==ssid_2g ssid_pw==ssid_pw_2g security==wpa2"
--radio "radio==wiphy1 stations==2 ssid==ssid_5g ssid_pw==BLANK security==open"

```

```

Example : Have the stations continue to run after the completion of the script
./test_l3_longevity.py --lfmgr 192.168.0.101 --endp_type 'lf_udp,lf_tcp' --tos BK --upstream_port 1.1.eth2
--radio 'radio==wiphy1 stations==2 ssid==asus_2g ssid_pw==lf_asus_2g security==wpa2'
--test_duration 30s --polling_interval 5s
--side_a_min_bps 256000 --side_b_min_bps 102400000
--no_stop_traffic

Example : Have script use existing stations from previous run where traffic was not stopped and also create new stations and leave traffic running
./test_l3_longevity.py --lfmgr 192.168.0.101 --endp_type 'lf_udp,lf_tcp' --tos BK --upstream_port 1.1.eth2
--radio 'radio==wiphy0 stations==2 ssid==asus_5g ssid_pw==lf_asus_5g security==wpa2'
--sta_start_offset 1000
--test_duration 30s --polling_interval 5s
--side_a_min_bps 256000 --side_b_min_bps 102400000
--use_existing_station_list
--existing_station_list '1.1.sta0000,1.1.sta0001'
--no_stop_traffic

Example : Add the following switches to use ssh to access ASUS (both ssh and serial supported), the interfaces need to be provided
--ap_read
--ap_scheme ssh
--ap_ip 192.168.50.1
--ap_ssh_port 1025
--ap_user lanforge
--ap_passwd lanforge
--ap_if_2g eth6
--ap_if_5g eth7
--ap_if_6g eth8

Setting wifi_settings per radio
./test_l3_longevity.py --lfmgr 192.168.100.116 --local_lf_report_dir /home/lanforge/html-reports/ --test_duration 15s
--polling_interval 5s --upstream_port 1.1.eth2
--radio "radio==1.1.wiphy1 stations==4 ssid==asusllax-5 ssid_pw==hello123 security==wpa2 mode==0 wifi_settings==wifi_settings
enable_flags==('ht160_enable'|'wpa2_enable'|'80211u_enable'|'create_admin_down'|'ht160_enable')"
--endp_type lf_udp --rates_are_totals --side_a_min_bps=20000 --side_b_min_bps=300000000 --test_rig CT-US-001 --test_tag 'l3_longev

Note: for enable flags can us && as separator in vscode

wifi_mode
Input      : Enum Val   : Shown by nc_show_ports
              AUTO      | 0       # 802.11g
              802.11a | 1       # 802.11a
              b        | 2       # 802.11b
              g        | 3       # 802.11g
              abg     | 4       # 802.11abg
              abgn    | 5       # 802.11abgn
              bgn     | 6       # 802.11bgn
              bg      | 7       # 802.11bg
              abgnAC  | 8       # 802.11abgn-AC
              anAC    | 9       # 802.11an-AC
              an      | 10      # 802.11an
              bgnAC   | 11      # 802.11bgn-AC
              abgnAX  | 12      # 802.11abgn-AX
              #      a/b/g/n/AC/AX (dual-band AX) support
              bgnAX   | 13      # 802.11bgn-AX
              anAX    | 14      # 802.11an-AX
              aAX     | 15      # 802.11a-AX (6E disables /n and /ac)

wifi_settings flags are currently defined as:
wpa_enable      | 0x10      # Enable WPA
custom_conf     | 0x20      # Use Custom wpa_supplicant config file.
wep_enable      | 0x200     # Use wpa_supplicant configured for WEP encryption.
wpa2_enable     | 0x400     # Use wpa_supplicant configured for WPA2 encryption.
ht40_disable   | 0x800     # Disable HT-40 even if hardware and AP support it.
scan_ssid      | 0x1000    # Enable SCAN-SSID flag in wpa_supplicant.
passive_scan   | 0x2000    # Use passive scanning (don't send probe requests).
disable_sgi    | 0x4000    # Disable SGI (Short Guard Interval).
lf_sta_migrate | 0x8000    # OK-To-Migrate (Allow station migration between LANforge radios)
verbose        | 0x10000   # Verbbose-Debug: Increase debug info in wpa-suppliant and hostapd logs.
80211u_enable  | 0x20000   # Enable 802.11u (Interworking) feature.
80211u_auto   | 0x40000   # Enable 802.11u (Interworking) Auto-internetworking feature. Always enabled currentl
80211u_gw     | 0x80000   # AP Provides access to internet (802.11u Interworking)
80211u_additional | 0x100000  # AP requires additional step for access (802.11u Interworking)
80211u_e911   | 0x200000  # AP claims emergency services reachable (802.11u Interworking)
80211u_e911_unauth | 0x400000  # AP provides Unauthenticated emergency services (802.11u Interworking)
hs20_enable    | 0x800000  # Enable Hotspot 2.0 (HS20) feature. Requires WPA-2.
disable_gdaf   | 0x1000000 # AP: Disable DGF (used by HotSpot 2.0).
8021x_radius  | 0x2000000 # Use 802.1x (RADIUS for AP).
80211r_pmska_cache | 0x4000000 # Enable oportunistic PMSKA caching for WPA2 (Related to 802.11r).
disable_ht80   | 0x8000000 # Disable HT80 (for AC chipset NICs only)
ibss_mode     | 0x2000000 # Station should be in IBSS mode.
osen_enable   | 0x40000000 # Enable OSEN protocol (OSU Server-only Authentication)
disable_roam   | 0x80000000 # Disable automatic station roaming based on scan results.
ht160_enable  | 0x10000000 # Enable HT160 mode.
disable_fast_reauth | 0x200000000 # Disable fast_reauth option for virtual stations.
mesh_mode     | 0x400000000 # Station should be in MESH mode.
power_save_enable | 0x800000000 # Station should enable power-save. May not work in all drivers/configurations.
create_admin_down | 0x1000000000 # Station should be created admin-down.
wds-mode      | 0x2000000000 # WDS station (sort of like a lame mesh), not supported on ath10k
no-supp-op-class-ie | 0x4000000000 # Do not include supported-oper-class-IE in assoc requests. May work around AP bugs.
txo-enable    | 0x8000000000 # Enable/disable tx-offloads, typically managed by set_wifi_txo command
use-wpa3      | 0x10000000000 # Enable WPA-3 (SAE Personal) mode.
use-bss-transition | 0x80000000000 # Enable BSS transition.
disable-twt   | 0x10000000000 # Disable TWT mode

=====
** FURTHER INFORMATION ***
Using the layer3_cols flag:

Currently the output function does not support inputting the columns in layer3_cols the way they are displayed in the GUI. This certain columns in the GUI in your final report, please match the according GUI column display to its counterpart to have the your report.

GUI Column Display          Layer3_cols argument to type in (to print in report)

```

Name	'name'
EID	'eid'
Run	'run'
Mng	'mng'
Script	'script'
Tx Rate	'tx rate'
Tx Rate (1 min)	'tx rate (1 min)'
Tx Rate (last)	'tx rate (last)'
Tx Rate LL	'tx rate ll'
Rx Rate	'rx rate'
Rx Rate (1 min)	'rx rate (1 min)'
Rx Rate (last)	'rx rate (last)'
Rx Rate LL	'rx rate ll'
Rx Drop %	'rx drop %'
Tx PDUs	'tx pdus'
Tx Pkts LL	'tx pkts ll'
PDU/s TX	'pdu/s tx'
Pps TX LL	'pps tx ll'
Rx PDUs	'rx pdus'
Rx Pkts LL	'pps rx ll'
PDU/s RX	'pdu/s tx'
Pps RX LL	'pps rx ll'
Delay	'delay'
Dropped	'dropped'
Jitter	'jitter'
Tx Bytes	'tx bytes'
Rx Bytes	'rx bytes'
Replays	'replays'
TCP Rtx	'tcp rtx'
Dup Pkts	'dup pkts'
Rx Dup %	'rx dup %'
OOO Pkts	'ooo pkts'
Rx OOO %	'rx ooo %'
RX Wrong Dev	'rx wrong dev'
CRC Fail	'crc fail'
RX BER	'rx ber'
CX Active	'cx active'
CX Estab/s	'cx estab/s'
1st RX	'1st rx'
CX TO	'cx to'
Pattern	'pattern'
Min PDU	'min pdu'
Max PDU	'max pdu'
Min Rate	'min rate'
Max Rate	'max rate'
Send Buf	'send buf'
Rcv Buf	'rcv buf'
CWND	'cwnd'
TCP MSS	'tcp mss'
Bursty	'bursty'
A/B	'a/b'
Elapsed	'elapsed'
Destination Addr	'destination addr'
Source Addr	'source addr'

Using the port\_mgr\_cols flag:

```
'4way time (us)'
'activity'
'alias'
'anqp time (us)'
'ap'
'beacon'
'bps rx'
'bps rx ll'
'bps tx'
'bps tx ll'
'bytes rx ll'
'bytes tx ll'
'channel'
'collisions'
'connections'
'crypt'
'cx ago'
'cx time (us)'
'device'
'dhcp (ms)'
'down'
'entity id'
'gateway ip'
'ip'
'ipv6 address'
'ipv6 gateway'
'key/phrase'
'login-fail'
'login-ok'
'logout-fail'
'logout-ok'
'mac'
'mask'
'misc'
'mode'
'mtu'
'no cx (us)'
'noise'
'parent dev'
'phantom'
'port'
'port type'
'pps rx'
'pps tx'
'qlen'
'reset'
'retry failed'
'rx bytes'
'rx crc'
'rx drop'
'rx errors'
```

```

'rx fifo'
'rx frame'
'rx length'
'rx miss'
'rx over'
'rx pkts'
'rx-rate'
'sec'
'signal'
'ssid'
'status'
'time-stamp'
'tx abort'
'tx bytes'
'tx crr'
'tx errors'
'tx fifo'
'tx hb'
'tx pkts'
'tx wind'
'tx-failed %'
'tx-rate'
'wifi retries'

Can't decide what columns to use? You can just use 'all' to select all available columns from both tables.

options:
-h, --help      show this help message and exit
--help_summary  Show summary of what this script does
--local_lf_report_dir LOCAL_LF_REPORT_DIR
                --local_lf_report_dir override the report path, primary use when running test in test suite
--test_rig TEST_RIG  test rig for kpi.csv, testbed that the tests are run on
--test_tag TEST_TAG  test tag for kpi.csv, test specific information to differentiate the test
--dut_hw_version DUT_HW_VERSION
                dut hw version for kpi.csv, hardware version of the device under test
--dut_sw_version DUT_SW_VERSION
                dut sw version for kpi.csv, software version of the device under test
--dut_model_num DUT_MODEL_NUM
                dut model for kpi.csv, model number / name of the device under test
--dut_serial_num DUT_SERIAL_NUM
                dut serial for kpi.csv, serial number / serial number of the device under test
--test_priority TEST_PRIORITY
                dut model for kpi.csv, test-priority is arbitrary number
--test_id TEST_ID   test-id for kpi.csv, script or test name
-o CSV_OUTFILE, --csv_outfile CSV_OUTFILE
                --csv_outfile <Output file for csv data>
--tty TTY          --tty "/dev/ttyUSB2" the serial interface to the AP
--baud BAUD       --baud "9600" AP baud rate for the serial interface
--mgr LFMGR, --lfmgr LFMGR
                --lfmgr <hostname for where LANforge GUI is running>
--mgr_port LFMGR_PORT, --lfmgr_port LFMGR_PORT
                --lfmgr_port <port LANforge GUI HTTP service is running on>
--test_duration TEST_DURATION
                --test_duration <how long to run> example --time 5d (5 days) default: 3m options: number followed by d, h
--tos TOS          --tos: Support different ToS settings: BK | BE | VI | VO | numeric
--debug           --debug flag present debug on enable debugging
-t ENDP_TYPE, --endp_type ENDP_TYPE
                --endp_type <types of traffic> example --endp_type "lf_udp lf_tcp mc_udp" Default: lf_udp , options: lf_u
-u UPSTREAM_PORT, --upstream_port UPSTREAM_PORT
                --upstream_port <cross connect upstream_port> example: --upstream_port eth1
--downstream_port DOWNSTREAM_PORT
                --downstream_port <cross connect downstream_port> example: --downstream_port eth2
--polling_interval POLLING_INTERVAL
                --polling_interval <seconds>
-r RADIO, --radio RADIO
                --radio "radio==<number_of_wiphy stations=<=number of stations> ssid==<ssid> ssid_pw==<ssid password> sec
--collect_layer3_data
                --collect_layer3_data flag present creates csv files recording layer3 columns of cxs.
--ap_read          --ap_read flag present enable reading ap
--ap_scheme {serial,telnet,ssh,mux_serial}
                --ap_scheme '/dev/ttyUSB0'
--ap_port AP_PORT, --ap_port '/dev/ttyUSB0'
--ap_baud AP_BAUD, --ap_baud '115200', default='115200'
--ap_ip AP_IP, --ap_ip
--ap_ssh_port AP_SSH_PORT
                --ap_ssh_port
--ap_user AP_USER, --ap_user , the user name for the ap, default = lanforge
--ap_passwd AP_PASSWD
                --ap_passwd, the password for the ap default = lanforge
--ap_if_2g AP_IF_2G, --ap_if_2g eth6
--ap_if_5g AP_IF_5G, --ap_if_5g eth7
--ap_if_6g AP_IF_6G, --ap_if_6g eth8
--ap_cmd_6g AP_CMD_6G
                ap_cmd_6g 'wl -i wl2 bs_data'
--ap_cmd_5g AP_CMD_5G
                ap_cmd_5g 'wl -i wl1 bs_data'
--ap_cmd_2g AP_CMD_2G
                ap_cmd_2g 'wl -i w10 bs_data'
--ap_cmd_ul_6g AP_CMD_UL_6G
                ap_cmd_ul_6g 'wl -i wl2 rx_report'
--ap_cmd_ul_5g AP_CMD_UL_5G
                ap_cmd_ul_5g 'wl -i wl1 rx_report'
--ap_cmd_ul_2g AP_CMD_UL_2G
                ap_cmd_ul_2g 'wl -i w10 rx_report'
--ap_chanim_cmd_6g AP_CHANIM_CMD_6G
                ap_chanim_cmd_6g 'wl -i wl2 chanim_stats'
--ap_chanim_cmd_5g AP_CHANIM_CMD_5G
                ap_chanim_cmd_5g 'wl -i wl1 chanim_stats'
--ap_chanim_cmd_2g AP_CHANIM_CMD_2G
                ap_chanim_cmd_2g 'wl -i w10 chanim_stats'
--ap_scheduler_stats --ap_scheduler_stats flag to clear stats run test then dump ul and dl stats to file
--ap_ofdma_stats   --ap_ofdma_stats flag to clear stats run test then dumps wl -i wl1 muinfo -v and wl 0i wl0 muinfo -v to fi
--ap_test_mode     ap_test_mode flag present use ap canned data
-amr SIDE_A_MIN_BPS, --side_a_min_bps SIDE_A_MIN_BPS
                --side_a_min_bps, requested downstream min tx rate, comma separated list for multiple iterations. Default

```

```

--amp SIDE_A_MIN_PDU, --side_a_min_pdu SIDE_A_MIN_PDU
    --side_a_min_pdu, downstream pdu size, comma separated list for multiple iterations. Default MTU
--bmr SIDE_B_MIN_BPS, --side_b_min_bps SIDE_B_MIN_BPS
    --side_b_min_bps, requested upstream min tx rate, comma separated list for multiple iterations. Default 2
--bmp SIDE_B_MIN_PDU, --side_b_min_pdu SIDE_B_MIN_PDU
    --side_b_min_pdu, upstream pdu size, comma separated list for multiple iterations. Default MTU
--rates_are_totals Treat configured rates as totals instead of using the un-modified rate for every connection.
--multiconn MULTICONN
    Configure multi-conn setting for endpoints. Default is 1 (auto-helper is enabled by default as well).
--attenuators ATTENATORS
    --attenuators, comma separated list of attenuator module eids: shelf.resource.attenu-serno.attenu-idx
--atten_vals ATTEM_VALS
    --atten_vals, comma separated list of attenuator settings in ddb units (1/10 of db)
--influx_host INFLUX_HOST
    Hostname for the Influx database
--influx_port INFLUX_PORT
    IP Port for the Influx database
--influx_org INFLUX_ORG
    Organization for the Influx database
--influx_token INFLUX_TOKEN
    Token for the Influx database
--influx_bucket INFLUX_BUCKET
    Name of the Influx bucket
--influx_tag INFLUX_TAG INFLUX_TAG
    --influx_tag <key> <val> Can add more than one of these.
--cap_ctl_out --cap_ctl_out, switch the controller output will be captured
--wait WAIT
    --wait <time>, time to wait at the end of the test
--sta_start_offset STA_START_OFFSET
    Station start offset for building stations
--no_pre_cleanup Do not pre cleanup stations on start
--no_cleanup Do not cleanup before exit
--no_stop_traffic leave traffic running
--use_existing_station_list
    --use_station_list ,full eid must be given, the script will use stations from the list, no configuration on
--existing_station_list EXISTING_STATION_LIST
    --station_list [list of stations] , use the stations in the list , multiple station lists may be entered
--wait_for_ip_sec WAIT_FOR_IP_SEC
    --wait_for_ip_sec <seconds> default : 120s
--log_level LOG_LEVEL
    Set logging level: debug | info | warning | error | critical
--lf_logger_config_json LF_LOGGER_CONFIG_JSON
    --lf_logger_config_json <json file> , json configuration of logger

    Useful Information:
        1. Polling interval for checking traffic is fixed at 1 minute
        2. The test will generate csv file
        3. The tx/rx rates are fixed at 256000 bits per second
        4. Maximum stations per radio based on radio

```

Candela Technologies, Inc., 2417 Main Street, Suite 201, Ferndale, WA 98248, USA  
[www.candlatech.com](http://www.candlatech.com) | [sales@candlatech.com](mailto:sales@candlatech.com) | +1.360.380.1618