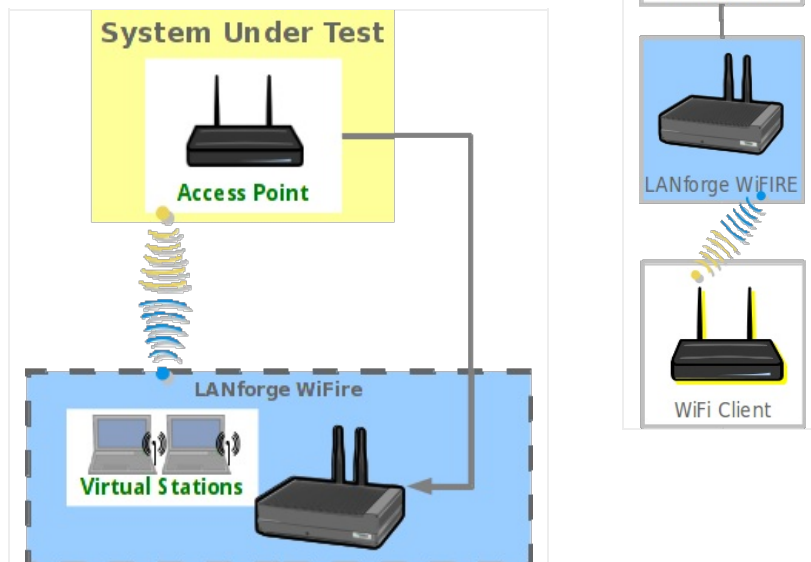


## Using LANforge python script to test QoS on WiFi Networks

**Goal:** Create data connections with different QoS settings and report on throughput and latency

This script automates creating data connections with BK, BE, VI, VO and/or other QoS settings. It then runs the tests for a specified amount of time and reports on latency and throughput. The report is an xlsx spreadsheet that can be opened in your favorite spread-sheet tool. This script requires LANforge 5.4.2 or higher.



1. Create and configure stations to talk to your DUT. You may create these through Chamber View scenarios or other methods. Please refer to introductory cookbooks for more information on this if you have questions.
2. Run lf\_tos\_test.py script.
  - A. Change directory to the /home/lanforge/scripts directory (or other location if you have installed scripts elsewhere), and run the lf\_tos\_tst.py script with the --help argument to understand your options..

```

greearb@ben-home:~/btbits/x64_btbits/server/lf_scripts
File Edit View Search Terminal Help
[greearb@ben-home lf_scripts]$ ./lf_tos_test.py --help
usage: lf_tos_test.py [-h] [--upstream_port UPSTREAM_PORT] [--station STATION]
                    [--lfmgr LFMGR] [--outfile OUTFILE] [--tos TOS]
                    [--speed_mbps SPEED_MBPS] [--duration DURATION]
                    [--proto PROTO]

ToS report Script

optional arguments:
  -h, --help            show this help message and exit
  --upstream_port UPSTREAM_PORT
                        LANforge upsteram-port to use (1.eth1, etc)
  --station STATION     LANforge stations to use (1.wlan0 1.wlan1 etc)
  --lfmgr LFMGR         LANforge Manager IP address
  --outfile OUTFILE     Output file for csv data
  --tos TOS             IP Type of Service: BK BE VI VO
  --speed_mbps SPEED_MBPS
                        Total requested transmit speed, in Mbps
  --duration DURATION   Duration to run traffic, in minutes
  --proto PROTO         List of protocols (udp tcp)
[greearb@ben-home lf_scripts]$

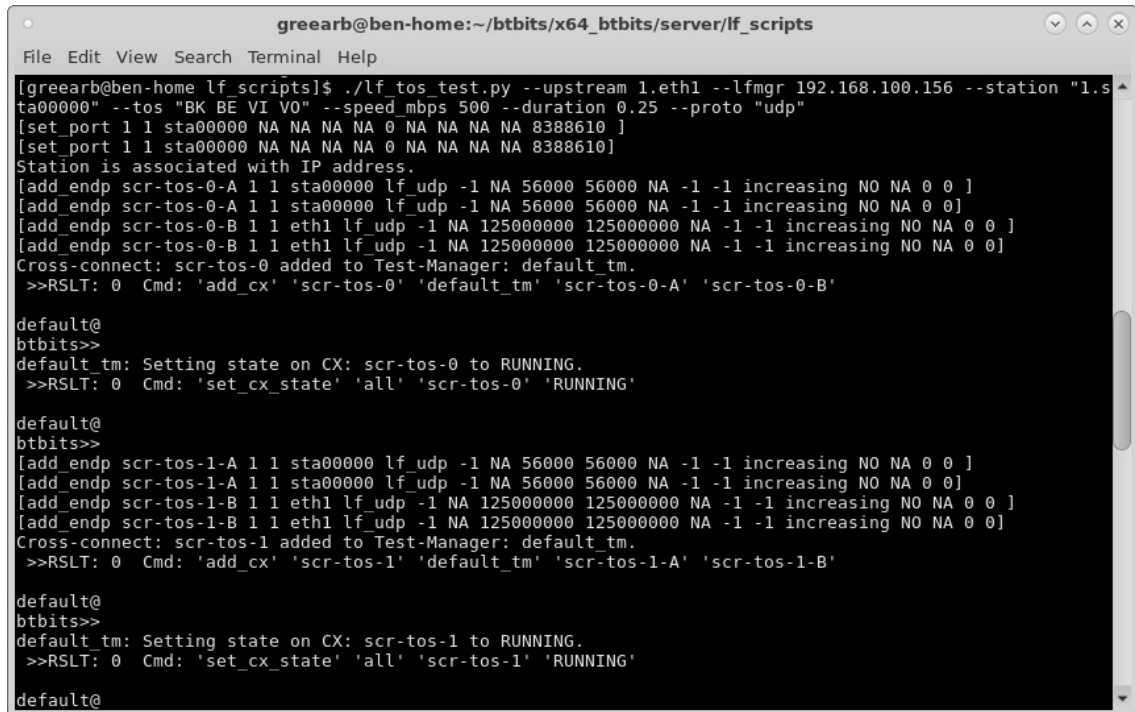
```

- B. Run the script with arguments for your test case. The basic idea is to provide a list of stations you wish to use for traffic generation, the upstream port, the type of service settings to use, requested bandwidth, etc.

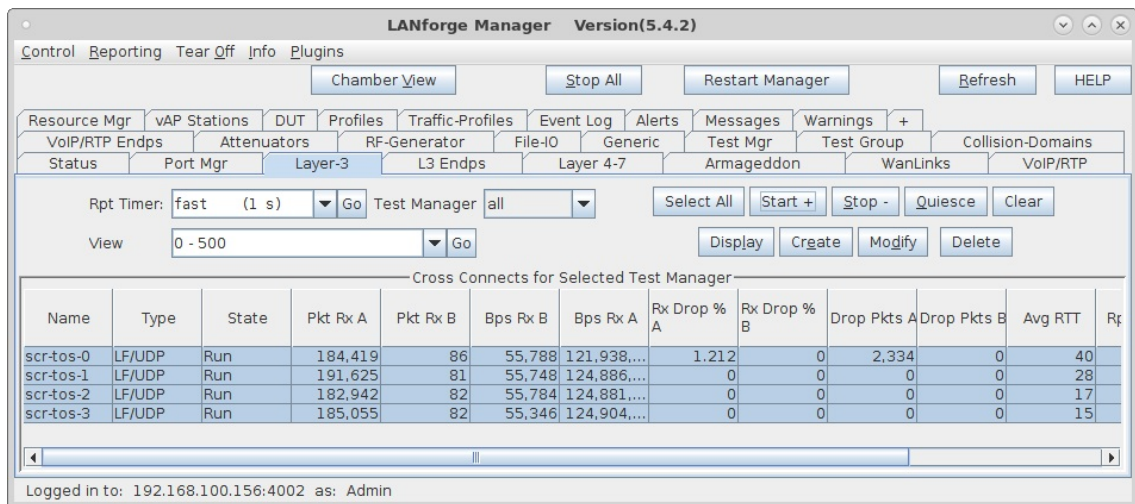
Examples

Run BE, BK, VI VO traffic between eth1 and sta0000, 500Mbps total traffic requested, with 15 second duration:

```
./lf_tos_test.py --upstream 1.eth1 --lfmgr 192.168.100.156 --station "1.sta00000" \
--tos "BK BE VI VO" --speed_mbps 500 --duration 0.25 --proto "udp"
```



- C. The auto-created data connections can be found in the Layer-3 tab in the LANforge GUI.





- G. The more interesting part of the report is the throughput and latency results. The latency-range columns show the amount of packets received in certain latency ranges. This gives you an idea of jitter and other latency related information. The average latency is a good way to get a quick idea of the latency.

The screenshot shows a LibreOffice Calc spreadsheet titled 'tos\_results.xlsx'. The spreadsheet contains a table with 14 columns and 12 rows. The columns are: Cx Rx Throughput, Avg Latency, Min Latency, Max Latency, Latency Range 0, Latency Range 1, Latency Range 2-3, Latency Range 4-7, Latency Range 8-15, Latency Range 16-31, Latency Range 32-63, Latency Range 64-127, Latency Range 128-255, Latency Range 256-511, and Latency Range 512-1023. The rows contain numerical data for each of these metrics. The first row (row 1) is the header, and rows 2 through 12 contain the data. The spreadsheet interface includes a menu bar, a toolbar, and a status bar at the bottom showing 'Average: ; Sum: 0' and '100%' zoom.

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	Cx Rx Throughput	Avg Latency	Min Latency	Max Latency	Latency Range 0	Latency Range 1	Latency Range 2-3	Latency Range 4-7	Latency Range 8-15	Latency Range 16-31	Latency Range 32-63	Latency Range 64-127	Latency Range 128-255	Latency Range 256-511	Latency Range 512-1023
2	46.48	81	0	525	3540	6010	13747	27668	21381	3720	7526	34040	32674	85409	15
3	46.48	39	1	223	0	9	3	7	11	16	16	20	18	5	0
4	44.84	14	0	149	2537	3982	10384	29199	54870	68519	48288	12679	863	0	0
5	44.84	22	1	156	0	6	3	10	20	21	27	10	3	3	0
6	42.79	19	0	56	781	1961	9172	39919	90797	68661	7558	0	0	0	0
7	42.79	10	1	54	0	11	11	22	18	24	9	3	0	0	0
8	40.48	6	1	41	0	377	3071	24264	82920	87847	8397	91	0	0	0
9	40.48	8	2	27	0	0	19	21	24	23	5	0	0	0	0
10															
11															
12															

- H. For more details, see the [spread-sheet result from this test](#). There is also a more advanced script that does similar work, see: `./lf_tos_plus_test.py --help`, [example output](#).