Candela
TECHNOLOGIES
Network Testing and Emulation Solutions

sales@candelatech.com
support@candelatech.com
+1 (360) 380-1618 [PST, GMT -8]

Using Flent to Generate Traffic

Goal: Set up virtual stations using a LANforge system, connect them to an AP under test, set up Flent, and run tests.

In this test scenario a LANforge system is used to create both the wireless stations and Flent test setup. The tests can then be configured to use Flent to generate upload or download traffic.

FLENT - **FLE**xible **N**etwork **T**ester is a set of python scripts that use various open-source tools to generate traffic and collect test data. Flent is developed and maintained by Toke Høiland-Jørgensen (@toke_dk)

One advantage to using Flent on LANforge is that a single LANforge box can send traffic to itself which eliminates the need for multiple systems to run client/server test traffic against a DUT.
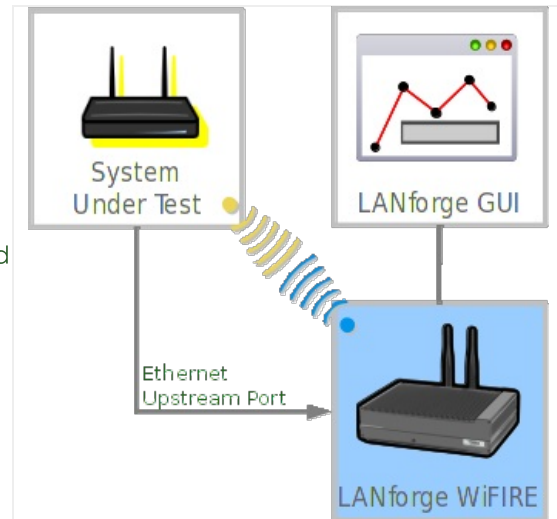
1. **Install packages required to run Flent on Fedora-30 (also works on Fedora-36).**
   Additional packages may need to be installed. See flent.org for more info.

   A. Install automake and texinfo
   ```
   sudo dnf install automake texinfo autoconf autotools-dev kernel-devel
   ```

   B. Flent
   ```
   git clone https://github.com/tohojo/flent.git
   cd flent
   git checkout v1.3.2
   sudo python3 setup.py install
   ```

   C. Flent GUI - **only necessary to view flent output graphs**
   ```
   sudo dnf install python3-matplotlib python3-matplotlib-qt5
   sudo dnf install python3-qt5 python3-PyQt5 python3-QtPy
   ```

   D. Netperf/Netserver
   ```
   git clone https://github.com/HewlettPackard/netperf.git
   cd netperf
   sudo ./autogen.sh
   sudo ./configure --enable-demo=yes
   sudo make
   sudo make install
   ```

   E. Fping
   ```
   git clone https://github.com/schweikert/fping.com
   cd fping
   git checkout 3.6
   sudo ./autogen.sh
   sudo ./configure
   sudo make
   sudo make install
   ```

F. Iperf2 - **Required for the --enhanced option which will display enhanced output in reports.**
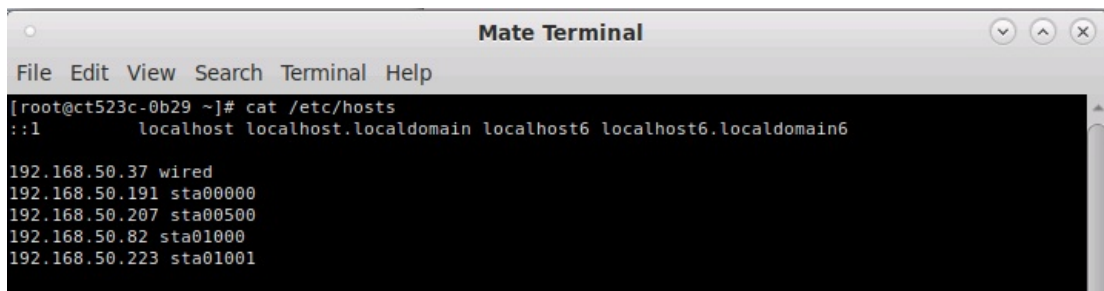
```
dnf install iperf
```

2. **To view a list of possible Flent tests, or run Flent by command line, open a terminal window.**

```
[lanforge@ct523c-0b29 ~]$ flent --list-tests
Available tests:
bursts                    :  Latency measurements under intermittent UDP bu
bursts_11e                :  802.11e Latency measurements under intermitten
cisco_5tcpup              :  RTT Fair Realtime Response Under Load
cisco_5tcpup_2udpflood    :  Cisco 5TCP up + 2 6Mbit UDP
cubic_bbr                 :  Cubic VS BBR smackdown
cubic_cdg                 :  Cubic VS CDG smackdown
cubic_dctcp               :  Cubic VS DCTCP smackdown
cubic_ledbat              :  Cubic VS Ledbat smackdown
cubic_ledbat_1            :  Cubic vs LEDBAT upload streams w/ping
cubic_reno                :  Cubic VS Reno smackdown
cubic_westwood            :  Cubic VS Westwood
dslreports_8dn            :  8 down - dslreports dsl test equivalent
http                      :  HTTP latency test
http-1down                :  HTTP get latency with competing TCP download s
http-1up                  :  HTTP get latency with competing TCP upload str
http-rrul                 :  HTTP get latency with competing RRUL test
iterated_bidirectional    :  Iterated TCP bidirectional transfers example
ledbat_cubic_1            :  Cubic vs LEDBAT upload streams w/ping
```

3. **Enter interface names to be used in Flent tests into the /etc/hosts file.**
   This is necessary for some of the Flent tools that can only use an interface name rather than IP address.

```
Mate Terminal
File  Edit  View  Search  Terminal  Help
[root@ct523c-0b29 ~]# cat /etc/hosts
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.50.37 wired
192.168.50.191 sta00000
192.168.50.207 sta00500
192.168.50.82 sta01000
192.168.50.223 sta01001
```

4. **To run Flent tests by command line:**

   A. Run netserver. This will start the netserver program which is the traffic server for the other Flent tools specifically netperf.

   ```
   [lanforge@ct523c-0b29 ~]$ netserver
   Starting netserver with host 'IN(6)ADDR_ANY' port '12865' and family AF_UNSPEC
   ```

   B. Run one of the available tests. Because LANforge is using vrf, use the **vrf_exec.bash** script to specify which interface to run the test. The direction of the test is from the perspective of the wired interface. Use the **--swap-up-down** option to change the direction to the host interface which is the wireless station sta00000 in this example.

   ```
   [lanforge@ct523c-0b29 ~]$ sudo ./vrf_exec.bash eth2 flent -x -H sta00000 tcp_download --swap-up-down
   [sudo] password for lanforge:
   Started Flent 1.9.9-git-d06e0b6 using Python 3.7.6.
   Starting tcp_download test. Expected run time: 70 seconds.
   Data file written to ./tcp_download-2020-04-15T113910.591355.flent.gz

   Summary of tcp_download test run from 2020-04-15 18:39:10.591355

                       avg          median        # data pts
   Ping (ms) ICMP :    23.20        23.10 ms         350
   TCP download   :    671.49       672.59 Mbits/s   350
   [lanforge@ct523c-0b29 ~]$
   ```
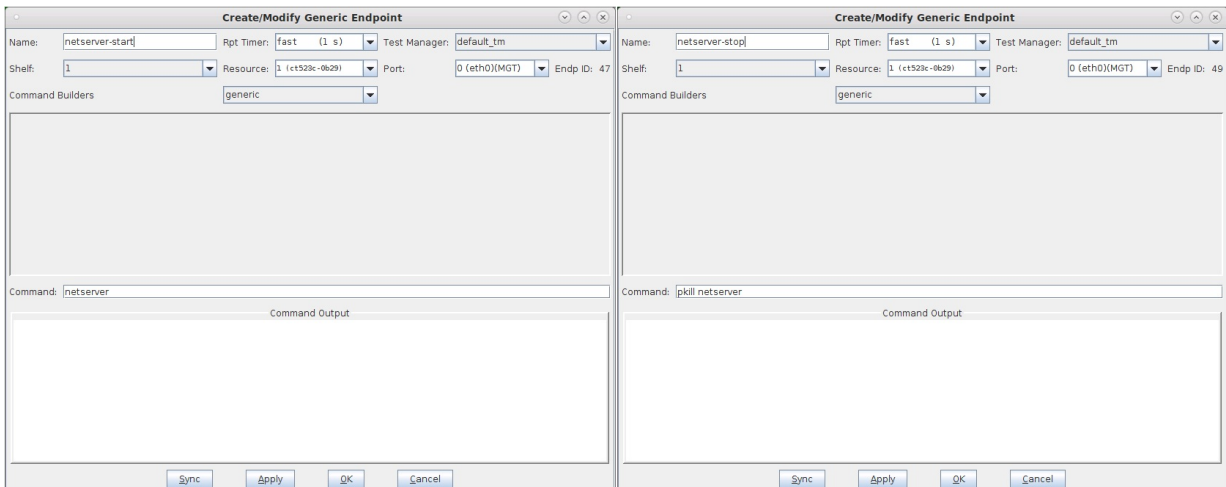
5. **To run Flent tests from the LANforge GUI, select the Generic tab from the main GUI window.**
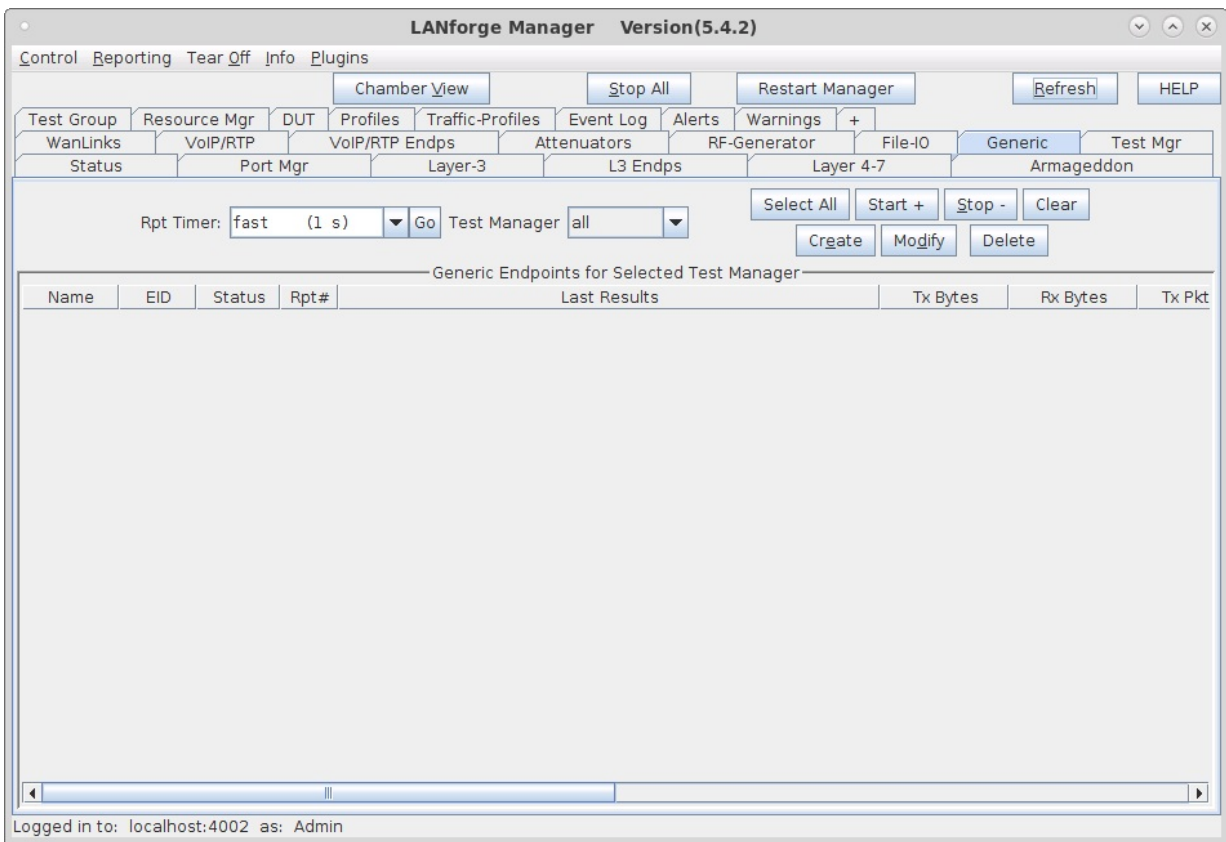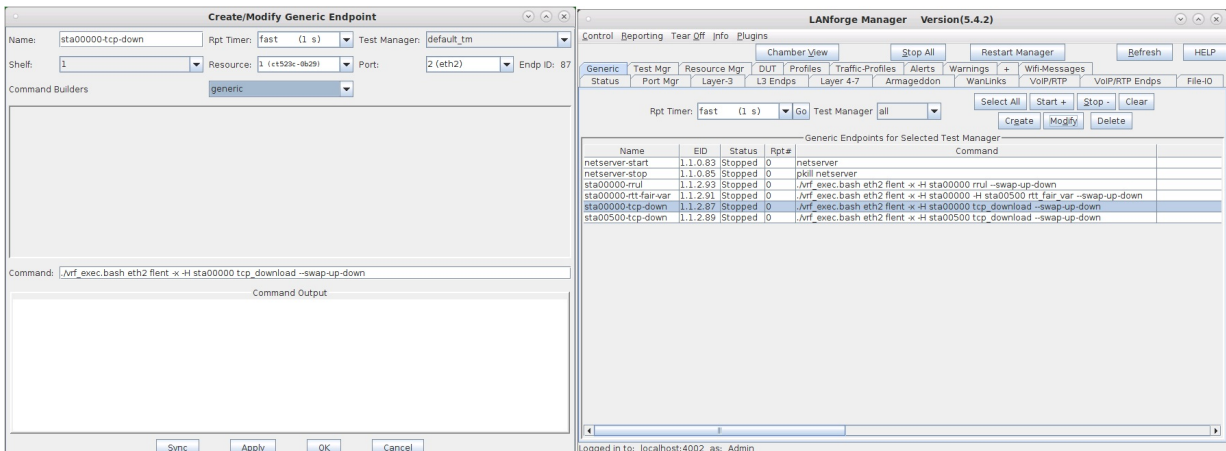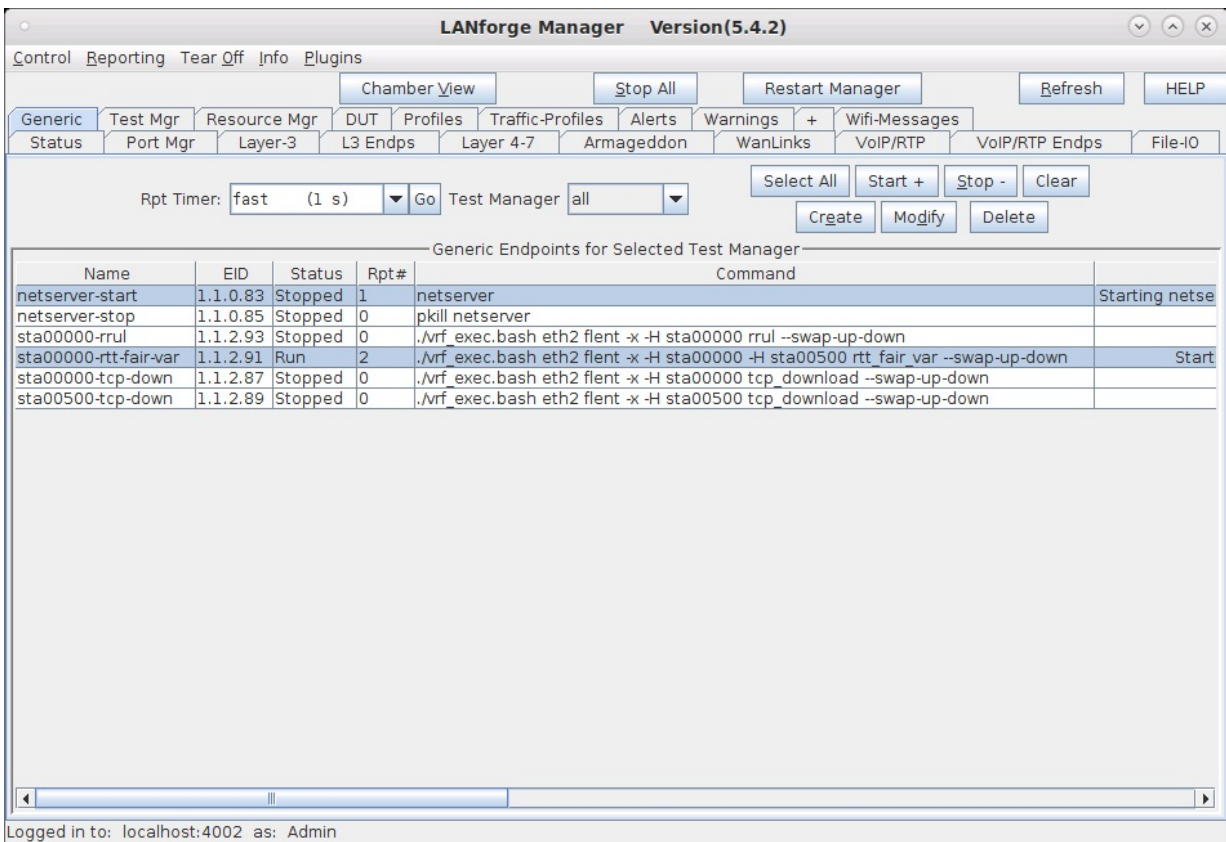
6. **Create the Flent netserver.**

   Here it is useful to create one Generic endpoint to start netserver and one to stop it because the netserver program runs in the background once it is started.

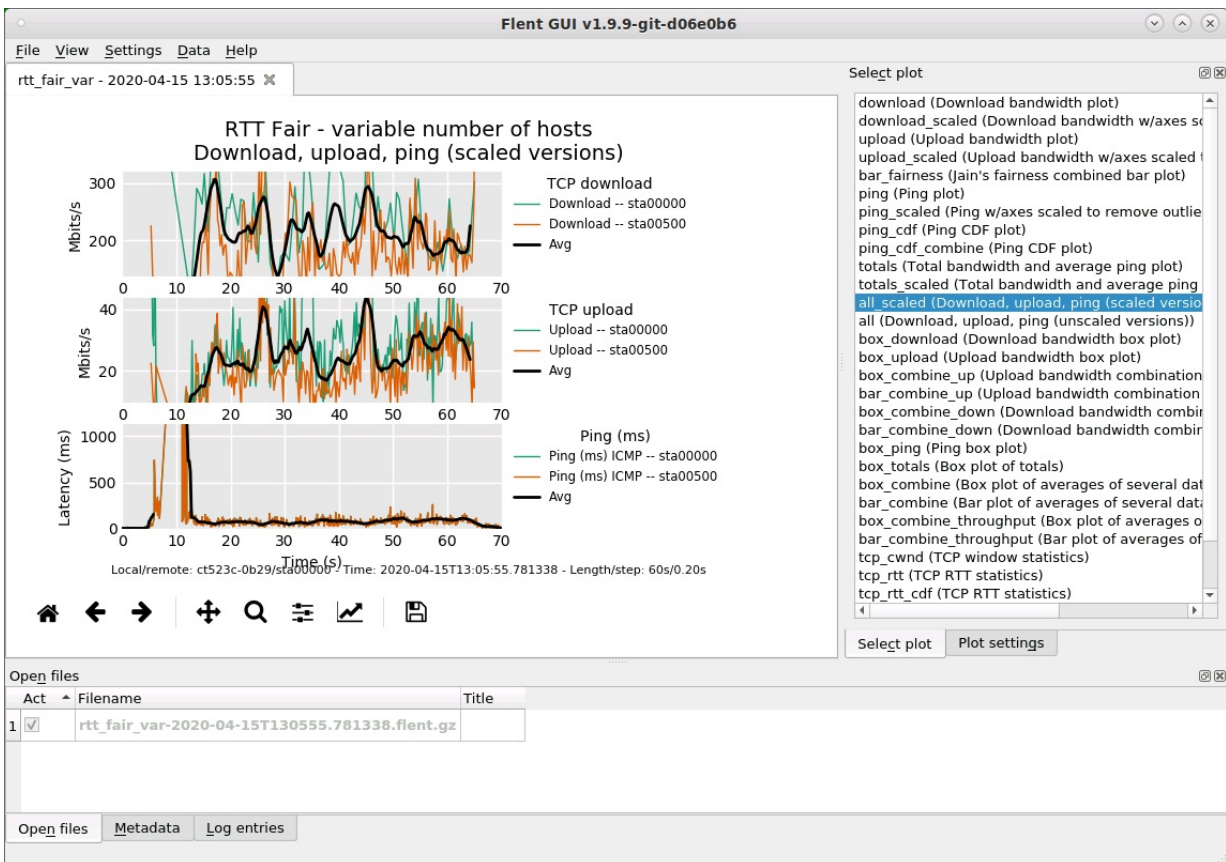
7. **Create the Flent test or tests.**

8. **To start any Flent test, start the netserver first, then start the tests one at a time.**
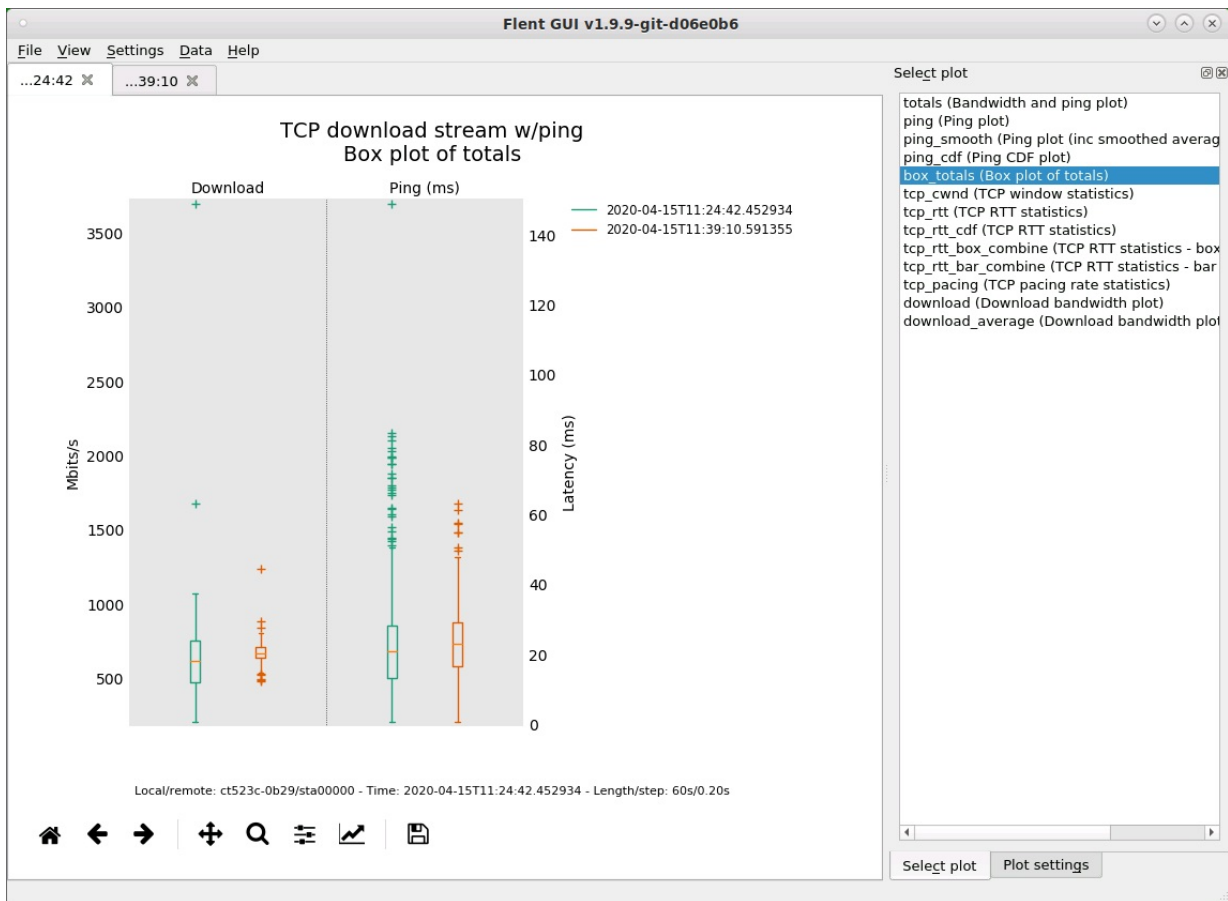

9. **Use the Flent GUI to view Flent test results.**

```
[lanforge@ct523c-0b29 ~]$ flent-gui rtt_fair_var-2020-04-15T130555.781338.flent.gz
```


10. **Use the Flent GUI to compare Flent test results.**

```
[lanforge@ct523c-0b29 ~]$ flent-gui tcp_download-2020-04-15T112442.452934.flent.gz tcp_
```

11. Flent batch files can be setup to run consecutive and repeated tests.

```
[lanforge@ct523c-0b29 ~]$ cat flent-batch.txt

[Batch::tcp-down]
test_name = tcp_download
hosts = sta00000
title = sta00000-tcp-down rep:${repetition}
filename_extra = sta00000-tcp-down-${repetition}
repetitions = 5
[Batch::tcp-up]
test_name = tcp_upload
hosts = sta00000
title = sta00000-tcp-up rep:${repetition}
filename_extra = sta00000-tcp-up-${repetition}
repetitions = 5
[Batch::rrul]
test_name = rrul
hosts = sta00000
title = sta00000-rrul rep:${repetition}
filename_extra = sta00000-rrul-${repetition}
repetitions = 5\
```

Create/Modify Generic Endpoint

Name: sta00000-batch   Rpt Timer: fast   (1 s)   Test Manager: default_tm

Shelf: 1   Resource: 1 (ct523c-0b29)   Port: 2 (eth2)   Endp ID: 95

Command Builders   generic

Command: ./vrf_exec.bash eth2 flent -x -H sta00000 -b tcp-down -b tcp-up -b rrul -B flent-batch.txt --swap-up-down

Command Output

Sync   Apply   OK   Cancel

12. **Flent batch results for rrul.**
The **rrul** test is the Realtime Response Under Load test which attempts to completely use up the link with four upstream TCP connections and four downstream TCP connections each with a different IP ToS setting plus ping and UDP to measure latency all running simultaneously for the purpose of exposing bufferbloat in the network under test.
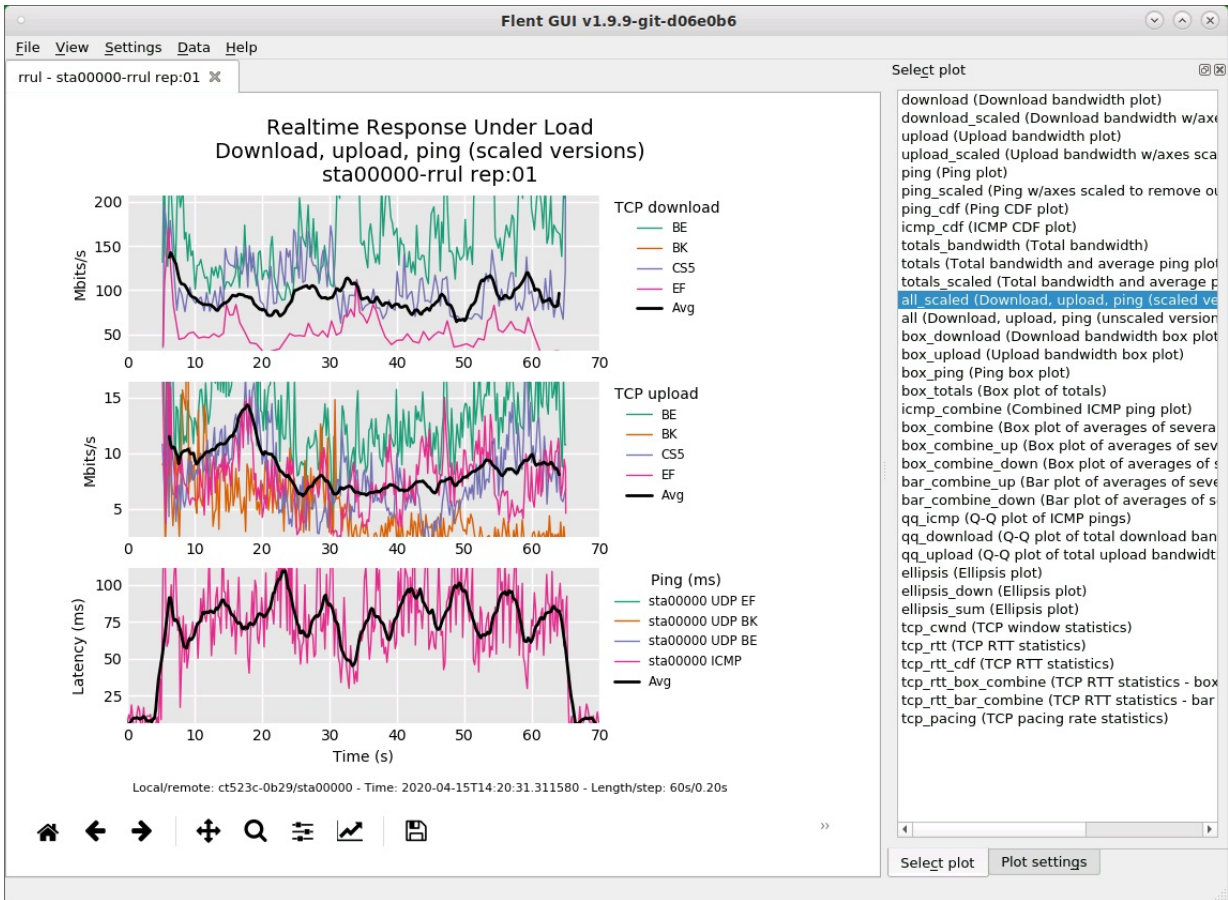
13. **Flent batch results for five rrul repetitions.**