Stateless mode

<https://trex-tgn.cisco.com/trex/doc/trex_manual.html>

Config file: /etc/trex\_cfg.yaml

cp cfg/simple\_cfg.yaml /etc/trex\_cfg.yaml

./dpdk\_setup\_ports.py -s

./dpdk\_setup\_ports.py -i

./dpdk\_setup\_ports.py -t

./dpdk\_setup\_ports.py -c <TRex interface 0> <TRex interface 1> ..

./t-rex-64 -f cap2/dns.yaml **--lm 1 --lo** -l 1000 -d 100

Dummy ports:

interfaces: ['07:00.0', 'dummy', 'dummy', '8a:00.0']

./t-rex-64 -f cap2/dns.yaml -c 4 -m 1 -d 10

./bp-sim-64-debug -f cap2/dns.yaml -o my.erf -v 3

wireshark my.erf

**TRex command line examples**

**Simple HTTP 1Gb/sec for 100 sec**

[bash]>sudo ./t-rex-64 -f cap2/simple\_http.yaml -c 4 -m 100 -d 100

**Simple HTTP 1Gb/sec with latency for 100 sec**

[bash]>sudo ./t-rex-64 -f cap2/simple\_http.yaml -c 4 -m 100 -d 100 -l 1000

**SFR 35Gb/sec traffic**

[bash]>sudo ./t-rex-64 -f avl/sfr\_delay\_10\_1g.yaml -c 4 -m 35 -d 100 -p

**SFR 20Gb/sec traffic with latency**

[bash]>sudo ./t-rex-64 -f avl/sfr\_delay\_10\_1g.yaml -c 4 -m 20 -d 100 -l 1000

**SFR ipv6 20Gb/sec traffic with latency**

[bash]>sudo ./t-rex-64 -f avl/sfr\_delay\_10\_1g\_no\_bundeling.yaml -c 4 -m 20 -d 100 -l 1000 --ipv6

**Simple HTTP 1Gb/sec with NAT translation support**

[bash]>sudo ./t-rex-64 -f cap2/simple\_http.yaml -c 4 -m 100 -d 100 -l 1000 --learn-mode 1

**IMIX 1G/sec ,1600 flows**

[bash]>sudo ./t-rex-64 -f cap2/imix\_fast\_1g.yaml -c 4 -m 1 -d 100 -l 1000

**IMIX 1Gb/sec,100K flows**

[bash]>sudo ./t-rex-64 -f cap2/imix\_fast\_1g\_100k.yaml -c 4 -m 1 -d 100 -l 1000

**64bytes ~1Gb/sec,1600 flows**

[bash]>sudo ./t-rex-64 -f cap2/imix\_64.yaml -c 4 -m 1 -d 100 -l 1000

**IMIX support:**

[bash]>sudo ./t-rex-64 -f cap2/imix\_64.yaml -d 1000 -m 40000 -c 4 -p

**Tutorial: Python automation**

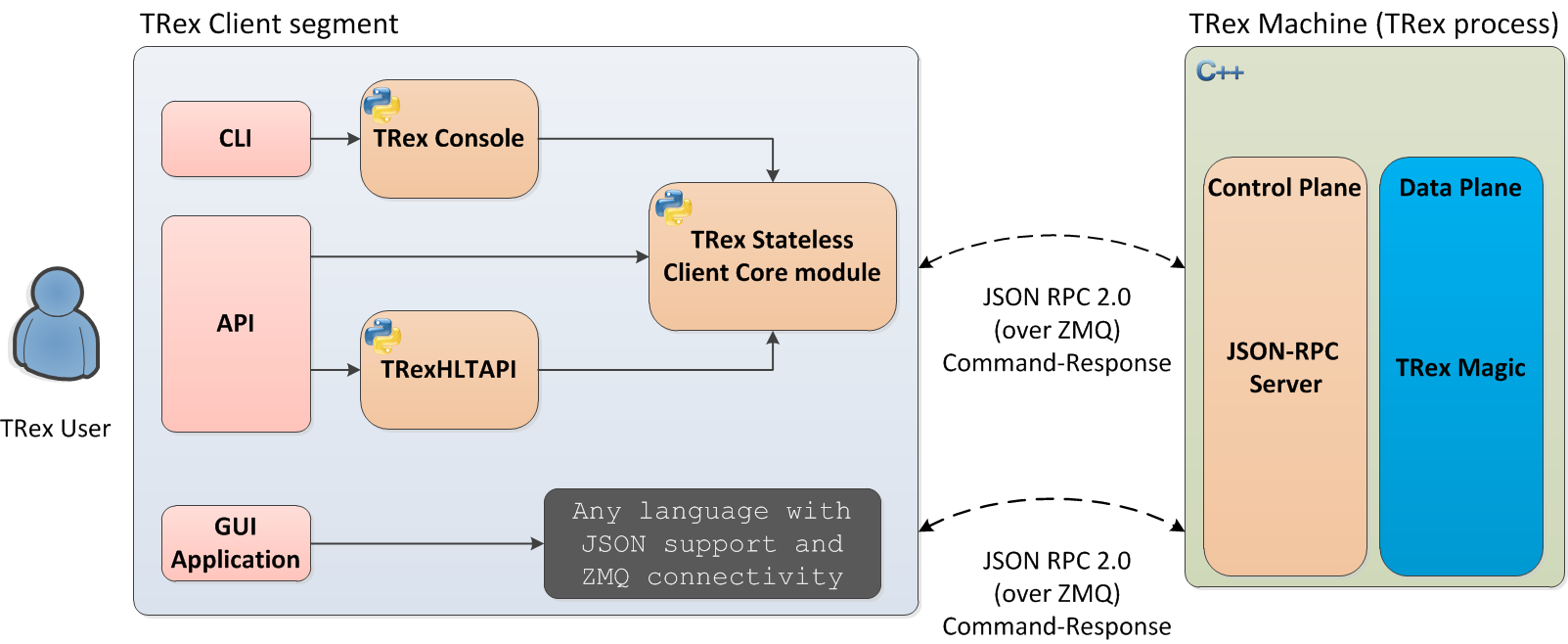
**Goal**

Simple automation test using Python from a local or remote machine.

**Directories**

Python API examples: automation/trex\_control\_plane/interactive/trex/examples/stl

Python API library: automation/trex\_control\_plane/interactive/trex/stl



Stateless API Reference

<https://trex-tgn.cisco.com/trex/doc/cp_stl_docs/api/client_code.html>

<https://trex-tgn.cisco.com/trex/doc/cp_stl_docs/index.html>

export PYTHONPATH=$PYTHONPATH: /usr/local/lib/python3.9/site-packages/trex/automation/trex\_control\_plane/interactive/trex/examples/stl/

export TREX\_PATH=/usr/local/lib/python3.9/site-packages/trex/automation/trex\_control\_plane/interactive/

export TREX\_EXT\_LIBS=/usr/local/lib/python3.9/site-packages/trex/trex\_client/external\_libs

STL\_PROFILES\_PATH: /opt/trex/v2.89/stl

EXT\_LIBS\_PATH: /opt/trex/v2.89/external\_libs

TREX\_EXT\_LIBS /opt/trex/v2.89/trex\_client/external\_libs

#### HLT Python API

Example: <https://github.com/cisco-system-traffic-generator/trex-core/tree/master/scripts/automation/trex_control_plane/interactive/trex/examples/stl/hlt_udp_simple.py>

APIs: <https://trex-tgn.cisco.com/trex/doc/trex_stateless.html#_hlt_supported_arguments_a_id_altapi_support_a>

./stl-sim -f stl/udp\_1pkt\_simple.py -o b.pcap -l 10

./stl-sim -f stl/udp\_1pkt\_simple.py --json

./stl-sim -f stl/udp\_1pkt\_simple.py --yaml

./stl-sim -f stl/udp\_1pkt\_simple.py --pkt

./stl-sim -f my\_yaml.yaml --native