A YANG Data Model for Network Interconnect Tester Management

draft-vassilev-bmwg-network-interconnect-tester-02

Author: Vladimir Vassilev <vladimir@lightside-instruments.com>

What is NOT available in network interconnect testers today?

1. Standard based management interface
2. Multi-vendor interoperability
3. Transactional Model definition - scalablity, simulatable (e.g. OMNeT++ model based on the YANG model)
4. Standard based report – NETCONF session recording

How does IETF YANG Data Model for Network Interconnect Tester Management solve these problems?

Multi-vendor interoperability

1. Tools
   All NETCONF/YANG tools will be available to validation and benchmark test developers as well as the network interconnect tester developers

2. Test programs
   Validation and benchmarking test suites can be developed in reusable way.

3. Self-tests
   A private case of the above point is the reusable network tester validation self-test

Transactional Model definition

1. NETCONF <commit> transactions to trigger reconfiguration independent of target

2. Scalable test framework

3. Easy to document

4. Support in discrete event simulation tools

Standard-based report generation from each test

1. NETCONF session data is human readable document.

Hackathon Plan

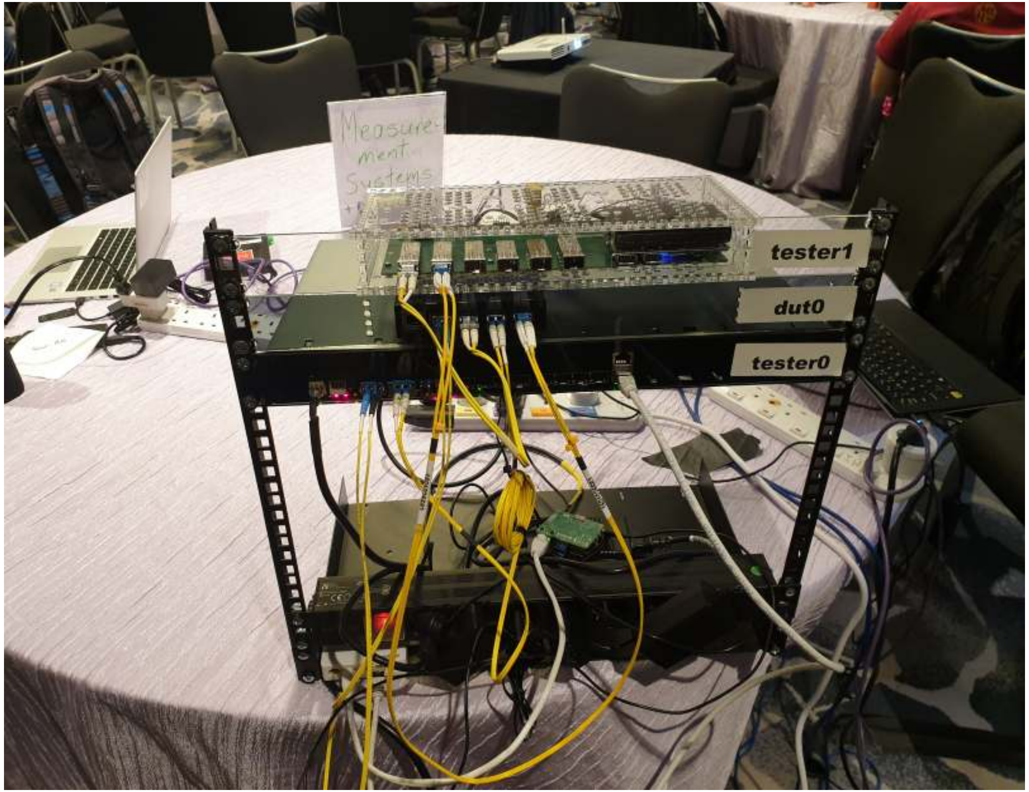
RFC2544 test using testers implementing the model specified in the draft. Test against DUT with NETCONF/YANG interface:

```
                     sfp-sfpplus1  sfp-sfpplus2
 +-------------+          +------------+        +------------+
 |             | xe0      |            |     xe0|            |
 | tester0   TG|>-------->|    dut0    |>------>|TA  tester1 |
 |             |          |            |        |            |
 +-------------+          +------------+        +------------+
```

```
$ python run-rfc2544.py config.xml | tee result.xml
```

Running hardware

>



Published code and reports

https://github.com/vlvassilev/litenc/blob/master/tntapi\
/example/ietf-network-interconnect-tester :

1. README
2. config-1.xml
3. report-bandwidth-1-64-10240.txt
4. report-bandwidth-1-64-9216.txt
5. test-network-interconnect-tester.py
6. test-rfc2544-throughput.py

Test run with losses

```
$ set-net config-1.xml
$ python ./test-rfc2544-throughput.py --config=config-1.xml \
--frame-size=64 --interframe-gap=9216 --tx-node=tester0 \
--tx-node-port=xe0 --rx-node=tester0 --rx-node-port=xe1 \
--src-mac-address="00:00:00:00:00:00" \
--dst-mac-address="00:00:00:00:00:01" > \
report-bandwidth-1-64-9216.txt
...
```

```
...
Transaction 5 started: 2019-11-20T01:36:39
Transaction 5 completed: 2019-11-20T01:36:39
Test time:                            60
Generated packets:               8092690
Lost packets:                     221575
Lost packets percent:           2.737965
Sequence errors:                   21009
Sequence errors percent:        0.259605
Latency Min[nanoseconds]:          12557
Latency Max[nanoseconds]:        9902196
```

Test run without losses

```
$ set-net config-1.xml
$ python ./test-rfc2544-throughput.py --config=config-1.xml \
--frame-size=64 --interframe-gap=10240 --tx-node=tester0 \
--tx-node-port=xe0 --rx-node=tester0 --rx-node-port=xe1 \
--src-mac-address="00:00:00:00:00:00" \
--dst-mac-address="00:00:00:00:00:01" | tee \
report-bandwidth-1-64-10240.txt
...
```

```
...
Transaction 5 started: 2019-11-20T01:34:44
Transaction 5 completed: 2019-11-20T01:34:44
Test time:                           60
Generated packets:              7289848
Lost packets:                         0
Lost packets percent:          0.000000
Sequence errors:                      0
Sequence errors percent:       0.000000
Latency Min[nanoseconds]:          9299
Latency Max[nanoseconds]:       1222324
```

References

1. Code and reports added as example to the tntapi project (Transactional Network Test API):

* https://github.com/vlvassilev/litenc/tree/master/tntapi\
/example/ietf-network-interconnect-tester

2. Toolchain with netconfd, yangcli with library for python scriptiing:

* https://yuma123.org/wiki

3. OMNeT++ use case:

* https://github.com/IETF-Hackathon\
/ietf104-project-presentations/blob/master\
/bmwg-network-interconnect-tester-hackathon104.pdf