



YANG model for management of Network Tester

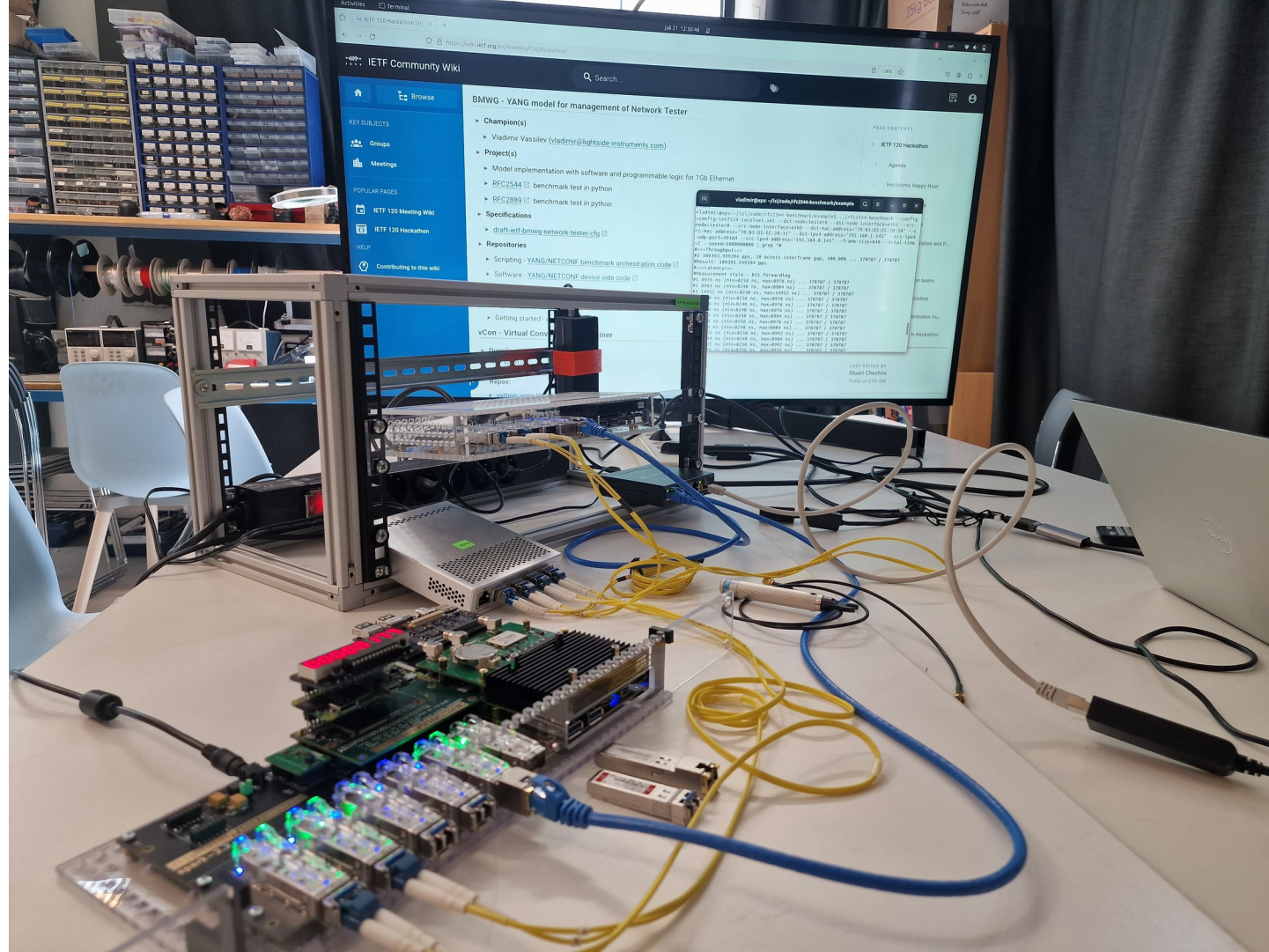
**IETF 120
20–21 July 2024
Vancouver, Canada**

Hackathon Plan

- validate draft-ietf-bmwg-network-tester-cfg-04 modules
- test latest reference implementation
- make implementation accessible to the IETF community with a public network address for the duration of IETF120 as sandbox e.g.

```
sudo apt install yangcli
```

```
yangcli --server=sandbox0.lightside-instruments.com --user=user --password=ietf120 --ncport=22006
```

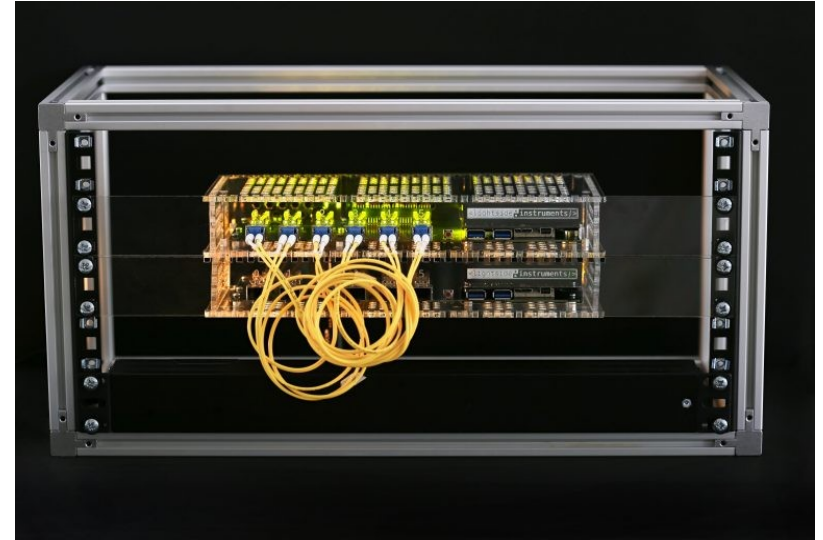


Quiz: Which 2 of the 5 network testers below have standard based management interfaces and protocols?

A



D



B



C



E



Network Tester Management Solutions

```

+-----+
eth0 |                               | eth1
+--<|TG    tester0    TA|<--+
|  |                               |  |
|  +-----+                       |
|          +-----+                |
+----->| DUT |>-----+
          +-----+

+-----+
eth0 |                               |
+--<|TG    tester0    |<-1PPS
|  |                               |<-10MHz
|  +-----+
+-----+
| DUT |
+-----+
|  +-----+
|  |                               |
+-->|TA    tester1    |<-1PPS
eth0 |                               |<-10MHz
+-----+

```

- Command line (SCPI) over TCP/IP or GPIB (IEEE 488)
- Cisco TRex (YAML,JSON-RPC)
- Keysight Open Traffic Generator APIs (REST) & Data Models (2023)
- High Level Test Application Programming Interface (HLTAPI)
- Other
- YANG (RFC7950) model and NETCONF(RFC6241) protocol implementation

Test & Measurement Instrument Automation Wish-list 1/2

1. have an interchangeable interface - IVI, YANG/NETCONF
2. be scalable instead of complex – combining L2-3 with >L4 devices is unnecessary
3. have transactional management model with hierarchical (tree-like) configuration and state representation - YANG/NETCONF
4. model that does not put constraints on the precision of the implementation (example: traffic spec with Pkt/sec, Bandwidth or interframe-gaps)
5. model that does not limit re-usability of implementations
have model with optional features and a mechanism to announce the set of implemented features – RFC8525 YANG Library)
6. management model has to have a compatible and simple command line manipulation syntax which does not require extra work to be developed – **yangcli**

Test & Measurement Instrument Automation Wish-list 2/2

7. model has to be as deterministic as it can be - avoid RFC 8342 NMDA delayed commit provision on test instruments
8. have model applicable to real devices, discrete event network simulation and gate level simulations
9. should generate report that is both human readable and machine readable – record the NETCONF session
10. bulk recording of state variables – NETCONF <get> with filter
11. be implemented based on open standards if such are available (ietf-network, ietf-interfaces)
12. processing requirements should be in the range of common embedded systems used for management of test instruments and configuration transaction should take less than 10 ms

Chronology 1/2

1965 HP develops „the interfacing of all future HP instruments“ and patents the Interface Bus (HP-IB) later known as GPIB

1975 IEEE488 Standard Digital Interface for Programmable Instrumentation is published

1980 RFC760 Internet Protocol is published

1999 RFC2544 Benchmarking Methodology for Network Interconnect Devices is published

2006 RFC4741 NETCONF Configuration Protocol is published

2010 RFC6020 YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF) is published

2014 Cisco acquires Tail-f (Sweden)

2015 TRex generator/analyzer released by Cisco

2016 First open-source YANG/NETCONF tool-chain is accepted into Debian (yuma123)

Chronology 2/2

2017 Keysight acquires Ixia

2018 Transpacket AS and ADVA demonstrate setup using YANG/NETCONF managed network traffic generators

2019 First draft version published to IETF BMWG

2020 Lightside Instruments AS publishes open-source hardware implementation of the draft (1Gb Ethernet)

2021 Lightside Instruments AS publishes open-source implementation of RFC2544 benchmark

2022 IETF BMWG adopts A YANG Data Model for Network Tester Management draft

2023 Teledyne acquires Xena Networks

2024 Keysight outbids Viavi and acquires Spirent

The project

Specification:

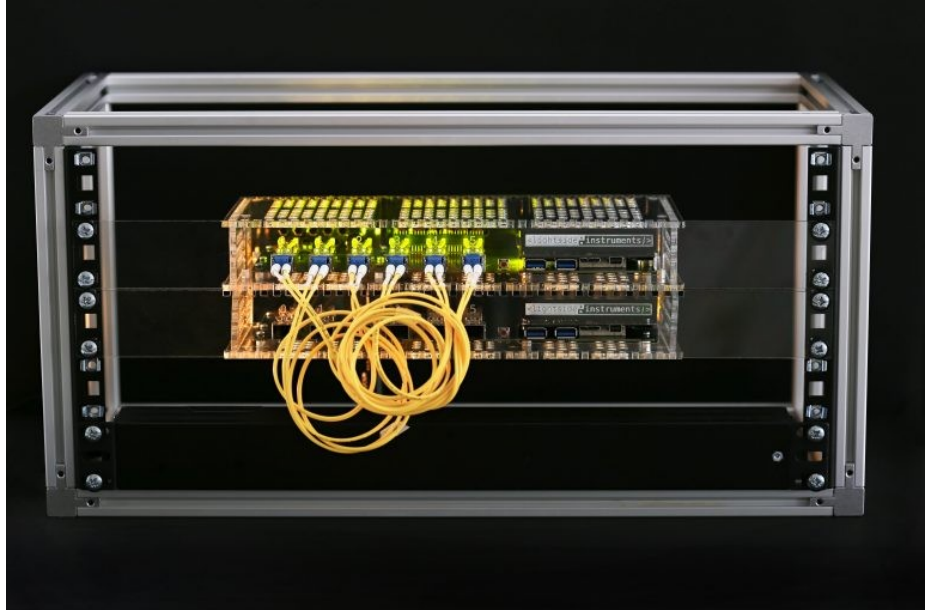
- * [draft-ietf-bmwg-network-tester-cfg-04](#)

Client side examples:

- * rfc2544-benchmark.py ([Python](#))

Device side:

- * Software - YANG/NETCONF server instrumentation code ([C](#))
- * Firmware - ([Verilog](#))
- * Hardware - off-the-shelf FPGA module Ultra96 + 6x SFP+ network programmability kit shield ([KiCAD](#), [Walk-through](#), OSHWA UIDs [NO000005](#), [NO000006](#))
- * Pre-silicon gate level [simulation](#) with cocotb/iverilog as alternative to target hardware



Open-source rfc2544-benchmark.py run - 1/5

* git repository -> [rfc2544-benchmark](#) 407 lines of python code

```
> rfc2544-benchmark.py --config=config.xml --dst-node=tester0 \  
--dst-node-interface=eth1 \  
--src-node=tester0 --src-node-interface=eth0 --dst-mac-address="70:B3:D5:EC:20:01" \  
--src-mac-address="70:B3:D5:EC:20:00" -dst-ipv4-address="192.0.2.2" \  
--src-ipv4-udp-port=49184 \  
--src-ipv4-address="192.0.2.1" --frame-size=64 --trial-time=120 --speed=1000000000 \  
| tee log.txt | grep "^#"  
...
```

Open-source rfc2544-benchmark.py run - 2/5

#===Throughput===

#1	1488095.238095	pps,	20	octets	interframe	gap,	100.00%	...	2001294	/	2976190
#2	744047.619048	pps,	104	octets	interframe	gap,	50.00%	...	1488095	/	1488095
#3	1116071.428571	pps,	48	octets	interframe	gap,	75.00%	...	2001280	/	2232142
#4	925925.925926	pps,	71	octets	interframe	gap,	62.22%	...	1851851	/	1851851
#5	1016260.162602	pps,	59	octets	interframe	gap,	68.29%	...	2001280	/	2032520
#6	968992.248062	pps,	65	octets	interframe	gap,	65.12%	...	1937984	/	1937984
#7	992063.492063	pps,	62	octets	interframe	gap,	66.67%	...	1984126	/	1984126
#8	1000000.000000	pps,	61	octets	interframe	gap,	67.20%	...	2000000	/	2000000
#9	1008064.516129	pps,	60	octets	interframe	gap,	67.74%	...	2001289	/	2016129
#10	1000000.000000	pps,	61	octets	interframe	gap,	67.20%	...	2000000	/	2000000

#Result: 1000000.000000 pps

...

Open-source rfc2544-benchmark.py run - 3/5

#===Latency===

#Measurement style - bit forwarding

#1 24624 ns (min=7024 ns, max=24624 ns) ... 2000000 / 2000000

#2 24472 ns (min=7016 ns, max=24472 ns) ... 2000000 / 2000000

#3 24624 ns (min=7024 ns, max=24624 ns) ... 2000000 / 2000000

#4 24360 ns (min=7016 ns, max=24360 ns) ... 2000000 / 2000000

...

#14 24360 ns (min=7024 ns, max=24360 ns) ... 2000000 / 2000000

#15 24224 ns (min=7016 ns, max=24224 ns) ... 2000000 / 2000000

#16 24208 ns (min=7016 ns, max=24208 ns) ... 2000000 / 2000000

#17 24704 ns (min=7016 ns, max=24704 ns) ... 2000000 / 2000000

#18 24752 ns (min=7016 ns, max=24752 ns) ... 2000000 / 2000000

#19 24760 ns (min=7016 ns, max=24760 ns) ... 2000000 / 2000000

#20 24176 ns (min=7016 ns, max=24176 ns) ... 2000000 / 2000000

#Result: 24440.400000 nanoseconds

...

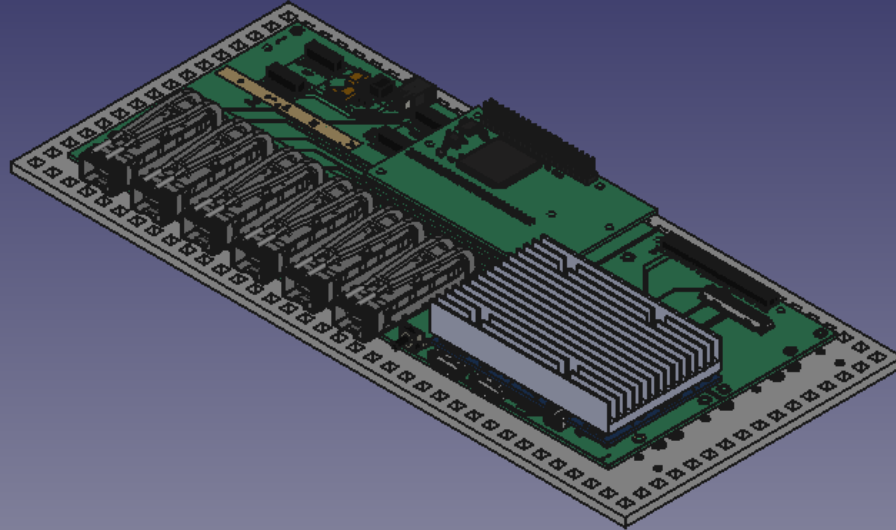
Open-source rfc2544-benchmark.py run - 4/5

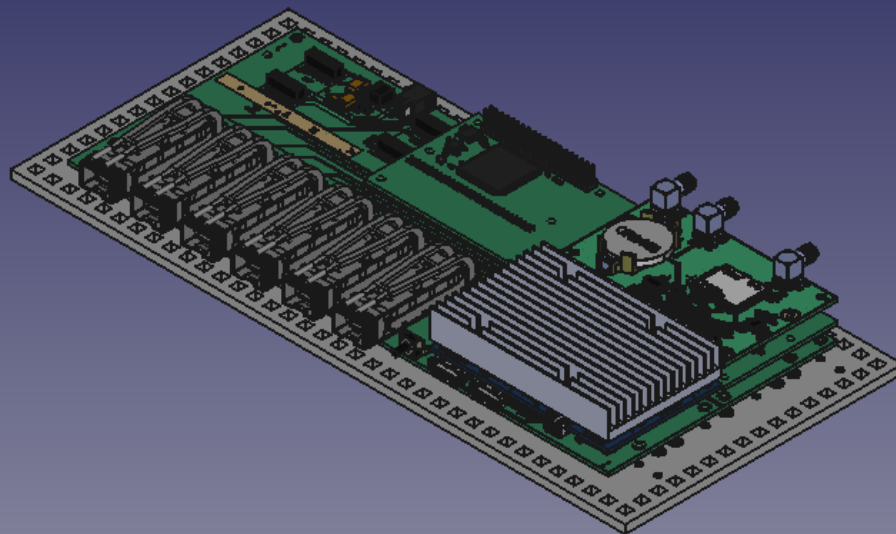
#===Frame loss rate===

```
#1 100% rate, 32% loss, (100.000000% rate actual), 1488095.238095 pps (1488095.238095 pps actual),  
  20 octets interframe gap ...      2001302 / 2976190  
#2 90% rate, 24% loss, (89.361702% rate actual),   1339285.714286 pps (1329787.234043 pps actual),  
  30 octets interframe gap ...      2001276 / 2659574  
#3 80% rate, 15% loss, (80.000000% rate actual),   1190476.190476 pps (1190476.190476 pps actual),  
  41 octets interframe gap ...      2001279 / 2380952  
#4 70% rate, 3% loss, (69.421488% rate actual),    1041666.666667 pps (1033057.851240 pps actual),  
  57 octets interframe gap ...      2001302 / 2066115  
#5 60% rate, 0% loss, (59.574468% rate actual),     892857.142857 pps ( 886524.822695 pps actual),  
  77 octets interframe gap ...      1773049 / 1773049  
#6 50% rate, 0% loss, (49.704142% rate actual),     744047.619048 pps ( 739644.970414 pps actual),  
 105 octets interframe gap ...      1479289 / 1479289  
...
```

Open-source rfc2544-benchmark.py run - 5/5

```
#===Back to back frames===  
#1 2 back-to-back frames ... 2 / 2  
#2 4 back-to-back frames ... 4 / 4  
...  
#10 1024 back-to-back frames ... 1024 / 1024  
#11 2048 back-to-back frames ... 1915 / 2048  
#12 1536 back-to-back frames ... 1536 / 1536  
#13 1792 back-to-back frames ... 1748 / 1792  
#14 1664 back-to-back frames ... 1664 / 1664  
#15 1728 back-to-back frames ... 1686 / 1728  
#16 1696 back-to-back frames ... 1683 / 1696  
#17 1680 back-to-back frames ... 1675 / 1680  
#18 1672 back-to-back frames ... 1672 / 1672  
#19 1676 back-to-back frames ... 1676 / 1676  
#20 1678 back-to-back frames ... 1678 / 1678  
#21 1679 back-to-back frames ... 1679 / 1679  
#Result: >= 1679
```

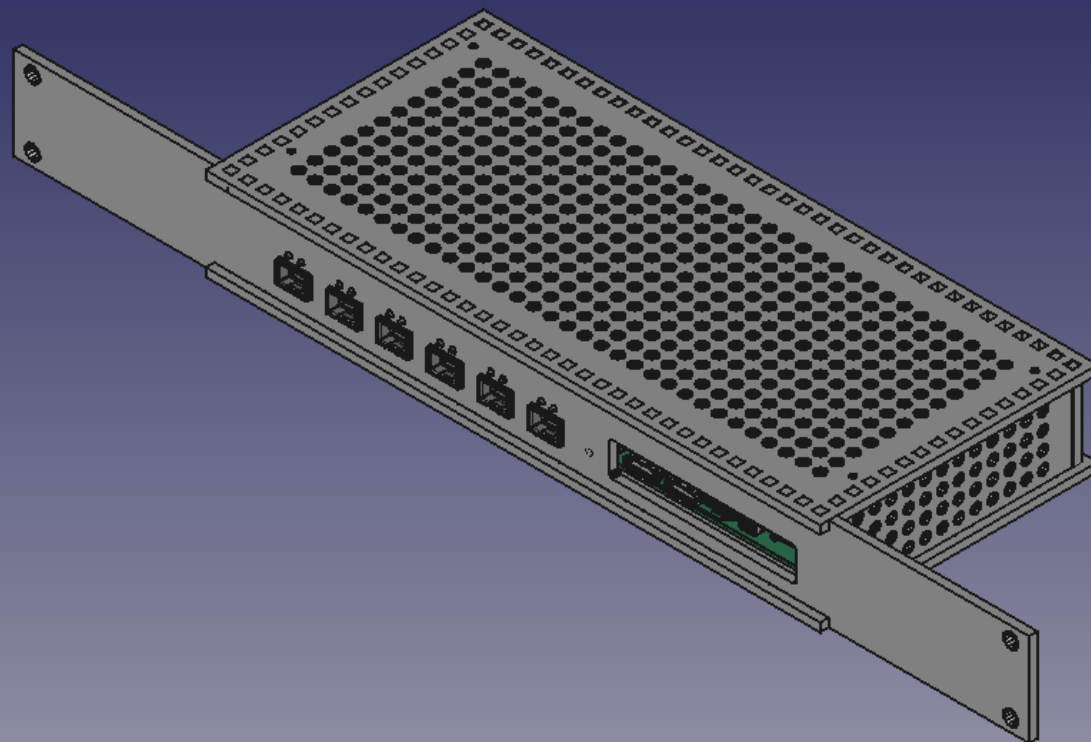




N0000005



N0000006



OSHw
N0000005

OSHw
N0000006

YANG tree diagram of the models

The following slide contains the complete YANG tree diagram of the ietf-traffic-generator.yang and ietf-traffic-analyzer.yang modules

```

module: ietf-traffic-analyzer
augment /if:interfaces/if:interface:
+--rw traffic-analyzer!
| +--rw testframe-filter! {testframe-filter}?
| | +--rw type identityref
| | +--rw mask? string
| | +--rw data? string
+--rw capture {capture}?
| +--rw start-trigger
| | +--rw (start-trigger)?
| | | +--:(frame-index)
| | | | +--rw frame-index? uint64
| | | +--:(testframe-index)
| | | +--rw testframe-index? uint64
| +--rw stop-trigger
| | +--rw (stop-trigger)?
| | | +--:(when-full)
| | | +--rw when-full? empty
+--ro state
+--ro pkts? yang:counter64
+--ro octets? yang:counter64
+--ro idle-octets? yang:counter64 {idle-octets-counter}?
+--ro errors? yang:counter64
+--ro testframe-stats
| +--ro testframe-pkts? yang:counter64
| +--ro sequence-errors? yang:counter64
| +--ro payload-errors? yang:counter64
| +--ro latency
| | +--ro samples? uint64
| | +--ro min? uint64
| | +--ro max? uint64
| | +--ro average? uint64
| | +--ro latest? uint64
+--ro capture {capture}?
+--ro frame* [sequence-number]
+--ro sequence-number uint64
+--ro timestamp? yang:date-and-time
+--ro length? uint32
+--ro data? string

```

```

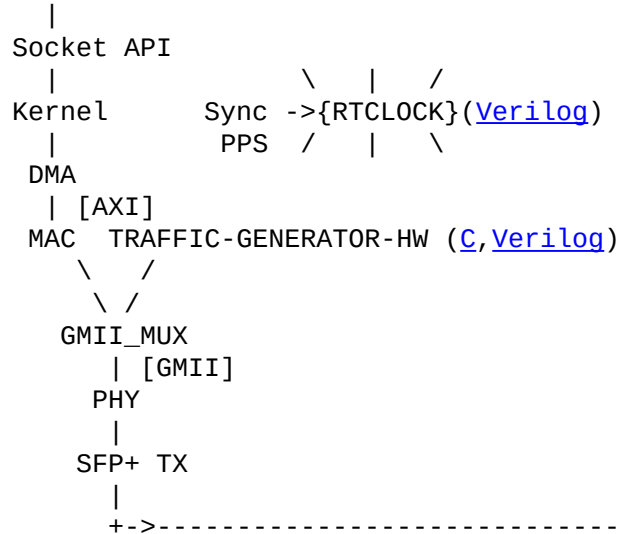
module: ietf-traffic-generator
augment /if:interfaces/if:interface:
+--rw traffic-generator
| +--rw (type)?
| | +--:(single-stream)
| | | +--rw testframe-type? identityref
| | | +--rw frame-size uint32
| | | +--rw frame-data? string
| | | +--rw interframe-gap uint32
| | | +--rw interburst-gap? uint32
| | | +--rw frames-per-burst? uint32
| | | +--rw modifiers
| | | | +--rw modifier* [id]
| | | | +--rw id uint32
| | | | +--rw action identityref
| | | | +--rw offset uint32
| | | | +--rw mask string
| | | | +--rw repetitions uint32
| | +--:(multi-stream)
| | | +--rw streams
| | | | +--rw stream* [id]
| | | | +--rw id uint32
| | | | +--rw testframe-type? identityref
| | | | +--rw frame-size uint32
| | | | +--rw frame-data? string
| | | | +--rw interframe-gap uint32
| | | | +--rw interburst-gap? uint32
| | | | +--rw frames-per-burst? uint32
| | | | +--rw frames-per-stream uint32
| | | | +--rw interstream-gap uint32
| | | | +--rw modifiers
| | | | | +--rw modifier* [id]
| | | | | +--rw id uint32
| | | | | +--rw action identityref
| | | | | +--rw offset uint32
| | | | | +--rw mask string
| | | | | +--rw repetitions uint32
+--rw realtime-epoch?
| yang:date-and-time {realtime-epoch}?
+--rw total-frames?

```

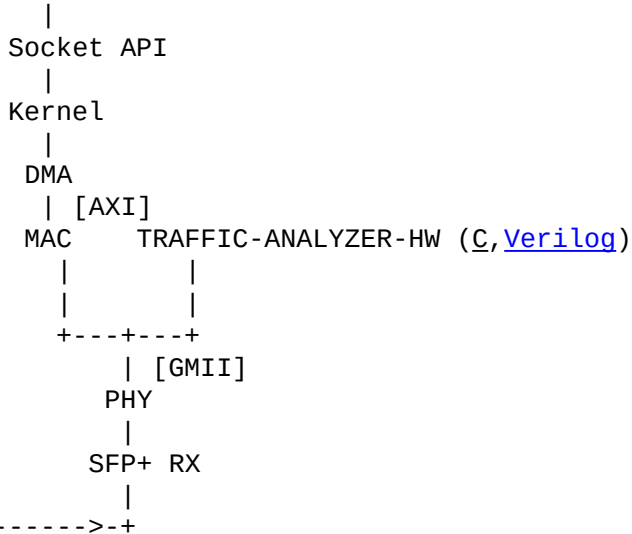
Design and implementation

NETCONF Server (Model ([YANG](#)), Implementation Generator module ([C](#)), Analyzer module ([C](#)))

TRAFFIC-GENERATOR-SW ([C](#))



TRAFFIC-ANALYZER-SW ([C](#))



* - underlined text has links to repositories

Some management transaction examples follow:

1. Configuration of 64 octet packet stream with dynamic timestamps with minial interframe gap on a traffic generator
2. Configuration of testframe filter with bitfield matching
3. Get counters and status information from the traffic anazlizer

* Notice the use of automated command line serialization with **yangcli**

1. Configure traffic generation:

```
yangcli user@192.168.4.145> create /interfaces/interface[name='eth0']/traffic-generator -- frame-size=64 interframe-gap=20 \
testframe-type=dynamic \
frame-data=123456789ABCDEF01234567808004500002E000000000A112CBCC0000201C0000202C0200007001A00000\
00102030405060708090A0B0C0D0E0F10119CD50E0F
```

```
<edit-config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <target>
    <candidate/>
  </target>
  <default-operation>merge</default-operation>
  <test-option>set</test-option>
  <config>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>eth0</name>
        <traffic-generator
          xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
          nc:operation="create"
          xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-generator">
          <testframe-type
            xmlns:nttg="urn:ietf:params:xml:ns:yang:ietf-traffic-generator">nttg:dynamic</testframe-type>
          <frame-size>64</frame-size>
          <frame-data>123456789ABCDEF01234567808004500002E000000000A112CBCC0000201C
0000202C0200007001A0000000102030405060708090A0B0C0D0E0F10119CD50E0F</frame-data>
          <interframe-gap>20</interframe-gap>
        </traffic-generator>
      </interface>
    </interfaces>
  </config>
</edit-config>
```

2. Configure test frame filter:

```
yangcli user@192.168.4.145> create /interfaces/interface[name='eth1']/traffic-analyzer/testframe-filter
-- type=bit-field-match data="123456789ABCDEF012345678" mask="000000000000FFFFFFFFFFFF"
```

```
<edit-config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <target>
    <candidate/>
  </target>
  <default-operation>merge</default-operation>
  <test-option>set</test-option>
  <config>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>eth1</name>
        <traffic-analyzer xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-analyzer">
          <testframe-filter
            xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
            nc:operation="create">
            <type
              xmlns:ntta="urn:ietf:params:xml:ns:yang:ietf-traffic-analyzer">ntta:bit-field-match</type>
            <mask>000000000000FFFFFFFFFFFF</mask>
            <data>123456789ABCDEF012345678</data>
          </testframe-filter>
        </traffic-analyzer>
      </interface>
    </interfaces>
  </config>
</edit-config>
```


3. Get status information:

```
yangcli user@192.168.4.145>
```

```
xget /interfaces/interface/traffic-analyzer/state
```

```
<get
```

```
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
```

```
  <filter type="xpath"
```

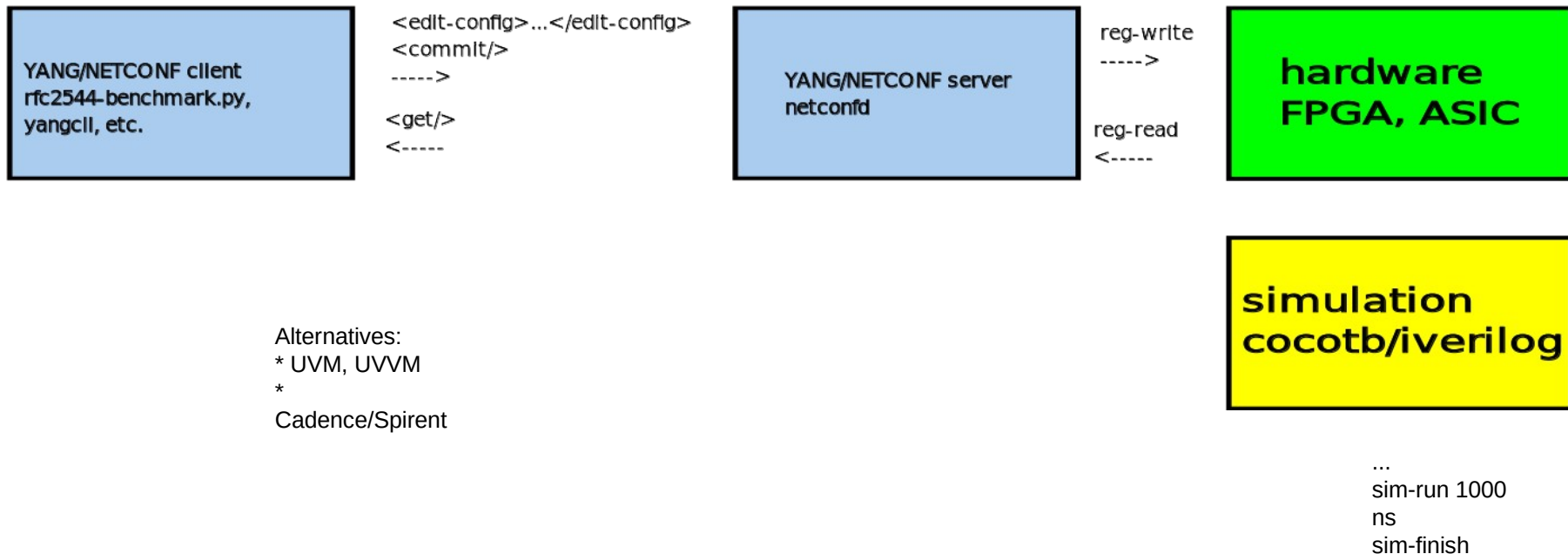
```
    select="/interfaces/interface/traffic-
```

```
analyzer/state"/>
```

```
</get>
```

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
      <interface>
        <name>eth1</name>
        <traffic-analyzer xmlns="urn:ietf:params:xml:ns:yang:ietf-traffic-analyzer">
          <state>
            <pks>43200851</pks>
            <octets>2764854464</octets>
            <octets-idle>562950384</octets-idle>
            <bad-crc-octets>0</bad-crc-octets>
            <bad-crc-pkts>0</bad-crc-pkts>
            <bad-preamble-octets>0</bad-preamble-octets>
            <bad-preamble-pkts>0</bad-preamble-pkts>
            <octets-total>3630210805</octets-total>
            <testframe-stats>
              <pks>43200851</pks>
              <sequence-errors>0</sequence-errors>
              <latency>
                <samples>43200851</samples>
                <min-sec>0</min-sec>
                <min>832</min>
                <max-sec>0</max-sec>
                <max>864</max>
                <last-sec>0</last-sec>
                <last>864</last>
              </latency>
            </testframe-stats>
            <capture>
              <timestamp>
                <nsec>902536272</nsec>
              </timestamp>
              <sequence-number>43200851</sequence-number>
              <data>5555555555555D5123456789ABCDEF01234567808004500002E000000000A112BCC00000201
C0000202C0200007001A0000000000000293315200000000005E735CB98F0345964C7</data>
            </capture>
          </state>
        </traffic-analyzer>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

Model defined pre-silicon testing environment



Pre-silicon testing

In the following example a RFC2544 benchmark against a **netconfd** server implementing the model by controlling a gate-level simulation of the synthesizable traffic-generator-gmii and traffic-analyzer-gmii cores in **cocotb** `sim_time_ns=1565330` (we used bogus 10 Kb Ethernet speed to actually simulate the dataplane in realtime).

We published the results in a [branch](#) :

- * Waveform trace (cocotb/iverilog gate-level generated)
- * Report (with verbose NETCONF transaction)

Some random screenshots taken during this process complete this presentation.

```
vladimir@xps: ~/ls/code/network-interconnect-tester-cores-...  
pps= 0, pps=0, pps2=0  
tic : time= 104000, sec= 0, nsec= 72, sec_next_  
pps= 0, pps=0, pps2=0  
110.00ns INFO cocotb.testers_loop.S_AXI Write complete addr: 0x0  
000000c prot: AxiProt.NONSECURE resp: AxiResp.OKAY length: 4  
110.00ns INFO cocotb.testers_loop.S_AXI Read start addr: 0x00000  
00c prot: AxiProt.NONSECURE length: 4  
tic : time= 112000, sec= 0, nsec= 80, sec_next_  
pps= 0, pps=0, pps2=0  
driving bus ...  
tic : time= 120000, sec= 0, nsec= 88, sec_next_  
pps= 0, pps=0, pps2=0  
tic : time= 128000, sec= 0, nsec= 96, sec_next_  
pps= 0, pps=0, pps2=0  
tic : time= 136000, sec= 0, nsec= 104, sec_next_  
pps= 0, pps=0, pps2=0  
tic : time= 144000, sec= 0, nsec= 112, sec_next_  
pps= 0, pps=0, pps2=0  
150.00ns INFO cocotb.testers_loop.S_AXI Read complete addr: 0x00  
00000c prot: AxiProt.NONSECURE resp: AxiResp.OKAY data: ed cb a9 87  
OK. Current value at REG_FLIP_ADDR is 0xEDCBA987 as expected  
Listening ...  
Accepting ...  
[ ]
```

```

edit-transaction 82703: operation replace on session 1 by y123@127.0.0.1
  at 2024-03-20T14:47:09Z on target 'running'
  data: /if:interfaces/1f:interface[1f:name='eth1']
traffic_analyzer_to_dictdelete:

  ntt:traffic-analyzer {
  }
ses: session 1 shut by remote peer
Session 1 closed
ses: session 2 shut by remote peer
Session 2 closed^C
Shutting down the netconfd server








vladimir@xps:~$ netconfd --superuser=y123 --module=ietf-traffic-generator --modu
le=ietf-traffic-analyzer --no-startup
Starting netconfd...
Copyright (c) 2008-2012, Andy Bierman, All Rights Reserved.
Copyright (c) 2013-2022, Vladimir Vassilev, All Rights Reserved.

agt: Startup configuration skipped due to no-startup CLI option

Running netconfd server (2.14-0)

```

```
vladimir@xps: ~/lsi/code/network-interconnect-tester-cores...  
vladimir@xps:~/lsi/code/network-interconnect-tester-cores-git/systems/simulation  
$ ./rfc2544-benchmark/rfc2544-benchmark --config=config.xml --dst-node=tester0 --  
--dst-node-interface=eth1 --src-node=tester0 --src-node-interface=eth0 --dst-mac-  
address="70:B3:D5:EC:20:10" --src-mac-address="70:B3:D5:EC:20:11" --dst-ipv4-ad-  
dress="192.168.1.145" --src-ipv4-udp-port=49184 --src-ipv4-address="192.168.0.145"  
--frame-size=64 --trial-time=2 --speed=10000 | tee rfc2544-benchmark-report-ve-  
rbose.txt | grep ^# | tee rfc2544-benchmark-report.txt
```

- 5: 
- 6: 
- 7: 
- 8: 
- 9: 
- 10: 
- 11: 

Device side:

- * Software - YANG/NETCONF
- * Firmware - ([Verilog](#))
- * Hardware – off-the-shelf FPGA kit shield ([KiCAD](#), [Walk-through](#))
- * Pre-silicon gate level simulation hardware

The screenshot shows the Beamer presentation editor interface. The main window displays a list of slides and shapes. The slides are numbered 3 through 11. Slide 3 contains shapes 1 through 6. Slide 4 contains shapes 1 through 6. Slide 5 contains shapes 1 through 3. Slide 7 contains shapes 1 through 2. Slide 8 contains shape 1. Slide 9 contains shapes 1 through 2. Slide 10 contains shapes 1 through 2. Slide 11 is currently selected and highlighted in orange. The right sidebar shows various icons for editing and navigation, including a search icon, a list icon, a zoom icon, a refresh icon, a delete icon, a copy icon, a paste icon, a link icon, a unlink icon, a zoom in icon, a zoom out icon, a full screen icon, a help icon, and a search icon.

gator

CustomShape 3

CustomShape 5

Shape 5 (Image with transparency)

CustomShape 6

Slide 3

CustomShape 1

CustomShape 2

CustomShape 3

CustomShape 4

Shape 5 (Image with transparency)

Shape 6 (Image)

Slide 4

Shape 1 (Image)

Shape 2 (Image with transparency)

Shape 3 (Image with transparency)

Slide 5

Shape 1 (Text Frame 'module: ...')

Shape 2 (Text Frame 'module: ...')

Slide 6

Shape 1 (Text Frame 'Tasks')

Shape 2 (Text Frame '* Implem...')

Slide 7

Shape 1 (Text Frame 'yangcli ...')

Slide 8

Shape 1 (Text Frame 'yangcli ...')

Slide 9

Shape 1 (Text Frame '<rpc-rep...')

Shape 2 (Text Frame 'yangcli ...')

Slide 10

Slide 11

1119-bmwf-network-interconnect-tester-model-00

```
vladimir@xps: ~/lsl/code/network-interconnect-tester-cores-...
Receiving ...
Received: write 0x20000010 0x00000001

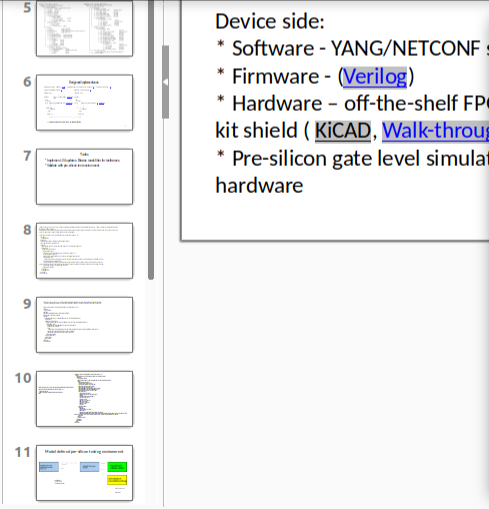
address=0x20000010, offset=0x00000010
1565290.00ns INFO cocotb.test_loop.S_AXI_TA Write start addr: 0x200
00010 prot: AxiProt.NONSECURE data: 01 00 00 00
tic : time= 1565296000, sec= 0, nsec= 1565264, sec_next_
pps= 0, pps=0, pps2=0
driving bus ...
driving bus ...
tic : time= 1565304000, sec= 0, nsec= 1565272, sec_next_
pps= 0, pps=0, pps2=0
tic : time= 1565312000, sec= 0, nsec= 1565280, sec_next_
pps= 0, pps=0, pps2=0
tic : time= 1565320000, sec= 0, nsec= 1565288, sec_next_
pps= 0, pps=0, pps2=0
tic : time= 1565328000, sec= 0, nsec= 1565296, sec_next_
pps= 0, pps=0, pps2=0
1565330.00ns INFO cocotb.test_loop.S_AXI_TA Write complete addr: 0x
20000010 prot: AxiProt.NONSECURE resp: AxiResp.OKAY length: 4
Receiving ...
Received:
Accepting ...
```

```
vladimir@xps: ~
Warning: 'latency' has no child node 'last-sec'. Using anyxml
Warning: 'latency' has no child node 'last'. Using anyxml
Warning: 'capture' has no child node 'timestamp0'. Using anyxml
Warning: 'capture' has no child node 'sequence-number'. Using anyxml
Warning: 'capture' has no child node 'data'. Using anyxml
edit-transaction 83000: operation delete on session 2 by y123@127.0.0.1
at 2024-03-20T14:56:05Z on target 'candidate'
data: /if:interfaces/if:interface[if:name='eth1']/ntta:traffic-analyzer

edit-transaction 83001: operation delete on session 1 by y123@127.0.0.1
at 2024-03-20T14:56:05Z on target 'running'
data: /if:interfaces/if:interface[if:name='eth1']/ntta:traffic-analyzer

edit-transaction 83001: operation replace on session 1 by y123@127.0.0.1
at 2024-03-20T14:56:05Z on target 'running'
data: /if:interfaces/if:interface[if:name='eth1']
traffic_analyzer_io_dictdelete:

ntta:traffic-analyzer {
}
ses: session 1 shut by remote peer
Session 1 closed
ses: session 2 shut by remote peer
Session 2 closed
```



Device side:

- * Software - YANG/NETCONF
- * Firmware - [Verilog](#)
- * Hardware - off-the-shelf FPC kit shield ([KICAD](#), [Walk-through](#))
- * Pre-silicon gate level simulation hardware

```
vladimir@xps: ~/lsl/code/network-interconnect-tester-cores-...
#16 8 ns (min=8 ns, max=8 ns) ... 29 / 29
#17 8 ns (min=8 ns, max=8 ns) ... 29 / 29
#18 8 ns (min=8 ns, max=8 ns) ... 29 / 29
#19 8 ns (min=8 ns, max=8 ns) ... 29 / 29
#20 8 ns (min=8 ns, max=8 ns) ... 29 / 29
#Result: 8.000000 nanoseconds
#===Frame loss rate===
#1 100% rate, 0% loss, (100.000000% rate actual), 14.880952 pps (14.880952 pps a
ctual), 20 octets interframe gap ... 29 / 29
#2 90% rate, 0% loss, (89.361702% rate actual), 13.392857 pps (13.297872 pps act
ual), 30 octets interframe gap ... 26 / 26
#===Back to back frames===
#1 2 back-to-back frames ... 2 / 2
#2 4 back-to-back frames ... 4 / 4
#3 8 back-to-back frames ... 8 / 8
#4 14 back-to-back frames ... 14 / 14
#The back to back search is limited to bursts below 1 second.
#Result: >= 14
#===System recovery===
#TODO
#===Reset===
#TODO
vladimir@xps:~/lsl/code/network-interconnect-tester-cores-git/systems/simulation
$
```

gator

- CustomShape 3
- CustomShape 5
- Shape 5 (Image with transparency)
- CustomShape 6
- Slide 3
- CustomShape 1
- CustomShape 2
- CustomShape 3
- CustomShape 4
- Shape 5 (Image with transparency)
- Shape 6 (Image)
- Slide 4
- Shape 1 (Image)
- Shape 2 (Image with transparency)
- Shape 3 (Image with transparency)
- Slide 5
- Shape 1 (Text Frame 'module: ...')
- Shape 2 (Text Frame 'module: ...')
- Slide 6
- Shape 1 (Text Frame 'Tasks')
- Shape 2 (Text Frame '* Implem...')
- Slide 7
- Shape 1 (Text Frame 'yangcli ...')
- Slide 8
- Shape 1 (Text Frame 'yangcli ...')
- Slide 9
- Shape 1 (Text Frame '<rcp-rep...')
- Shape 2 (Text Frame 'yangcli ...')
- Slide 10
- Slide 11

ietf119-bmwf-network-interconnect-tester-model-00

SST

- tester_loop
 - rtclock0 (rtclock)
 - traffic_analyzer_gmii0 (t
 - bram_io_inst (bram_i
 - ethernet_crc_8_check
 - opl_cpu_regs_inst (tr
 - testframe_parser_0 (
 - traffic_generator_gmii0
 - bram_io_inst (bram_i
 - ethernet_crc_8_0 (eth
 - opl_cpu_regs_inst (tr

Type	Signals
wire	preamble_ok
wire	resetrn
reg	run
wire	sec[47:0]
reg	sequence_errors[63:0]
integer	sequence_errors_delta
reg	sequence_errors_reg[63:0]
wire	sequence_num[63:0]
reg	state[1:0]
wire	testframe_filter_data[63:0]
wire	testframe_filter_mask[63:0]
wire	testframe_match
reg	testframe_pkts[63:0]
integer	testframe_pkts_delta
reg	testframe_pkts_reg[63:0]
reg	timestamp_nsec[31:0]
reg	timestamp_nsec_reg[63:0]
reg	timestamp_sec[63:0]
reg	timestamp_sec_reg[63:0]
wire	timestamp_tx_nsec[31:0]
wire	timestamp_tx_nsec_reg[63:0]

Filter:

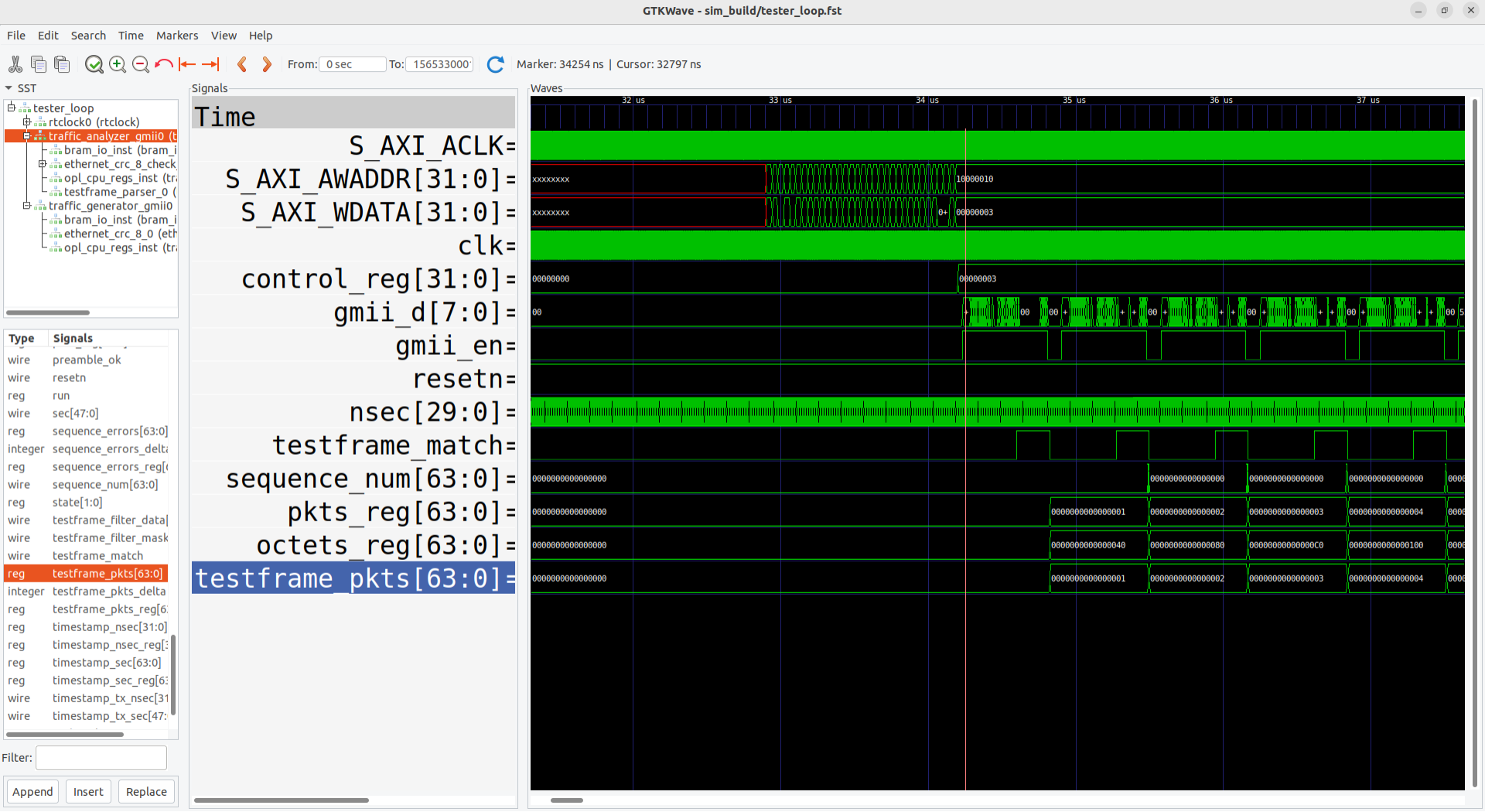
Append Insert Replace

Signals

Time

```
S_AXI_ACLK=
S_AXI_AWADDR[31:0]=
S_AXI_WDATA[31:0]=
clk=
control_reg[31:0]=
gmii_d[7:0]=
gmii_en=
resetrn=
nsec[29:0]=
testframe_match=
sequence_num[63:0]=
pkts_reg[63:0]=
octets_reg[63:0]=
testframe_pkts[63:0]=
```





- tester_loop
 - rtclock0 (rtclock)
 - traffic_analyzer_gmii0 (traffic_analyzer_gmii0) **selected**
 - bram_io_inst (bram_io_inst)
 - ethernet_crc_8_check (ethernet_crc_8_0)
 - opl_cpu_regs_inst (opl_cpu_regs_inst)
 - testframe_parser_0 (testframe_parser_0)
 - traffic_generator_gmii0 (traffic_generator_gmii0)
 - bram_io_inst (bram_io_inst)
 - ethernet_crc_8_0 (ethernet_crc_8_0)
 - opl_cpu_regs_inst (opl_cpu_regs_inst)

Append Insert Replace

The End