dknog

DKNOG14 | March 7-8th 2024

rtbrick BNG Blaster

Open Source Network Tester

Introduction

Christian Giese | Field Services and Software Engineer | RtBrick

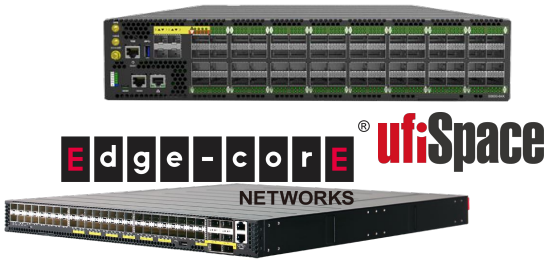
What do we do?

rtbrick Container

- **Interfaces**
  CLI, REST, TSDB ...
- **Micro Services**
  BGP, ISIS, OSPF, PPPoE, L2TP, ...
- **Brick Data Store (BDS)**
- **Linux (LXC Container)**
- **Linux (Host)**

Disaggregated Routing (Core) and Access (BNG)

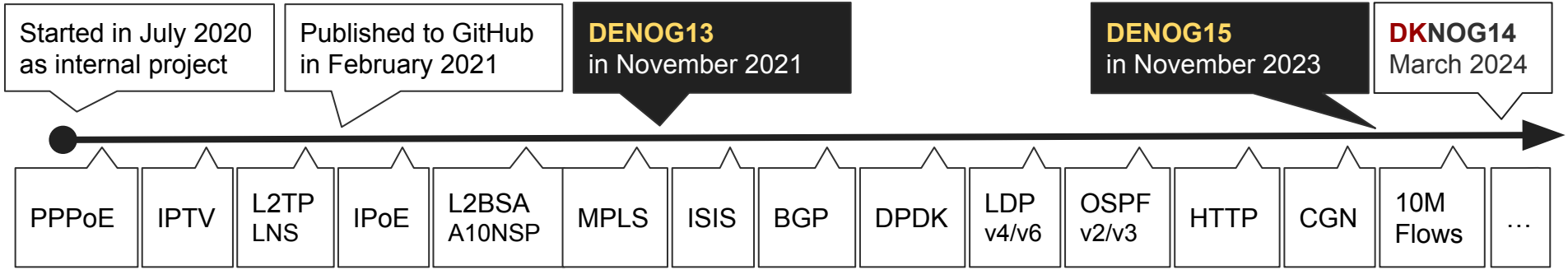
ISIS, OSPFv2/3, Segment Routing, LDP, BGP, PPPoE, L2TPv2, IPoE (DHCPv4/6), RADIUS, HA, HTTP Redirect, Carrier Grade NAT, …



https://www.rtbrick.com

BNG Blaster

- Open source network tester (BSD-3 license)
- Started as BNG access protocol tester
- Added support for traffic generator, routing protocols and applications
- Easily extendable
- Continuously improved and actively maintained
- Controller with automation friendly API
- …



>3 years of work, >1000 commits, >50K lines of C Code, …

Features

rtbrick

- Emulates massive sessions with low CPU and memory footprint
- Runs on every modern linux, virtual machines and containers
- All protocols implemented in user-space and optimized for performance
- Automation friendly API
- …

Access Protocols

- Emulate massive PPPoE and IPoE (DHCP) clients
- Emulate L2TPv2 LNS servers with different behaviors
- Emulate A10NSP interfaces for L2BSA testing
- Included multicast and IPTV test suite
- Verify legal interception (LI) traffic
- Emulate HTTP client/server for redirect and filter testing
- …

Routing Protocols

- Emulate ISIS and OSPF topologies with thousands of nodes
- Support for Segment Routing
- Setup thousands of BGP sessions with millions of prefixes
- Support for LDP
- Verify MPLS labels for millions of flows
- …

Traffic Generator

- Generate and track millions of traffic flows
- Verify your QoS configuration
- Verify all forwarding states
- Measure convergence times and loss
- Carrier Grade NAT (CGN)
- DPDK
- …

BNG Blaster Requirements

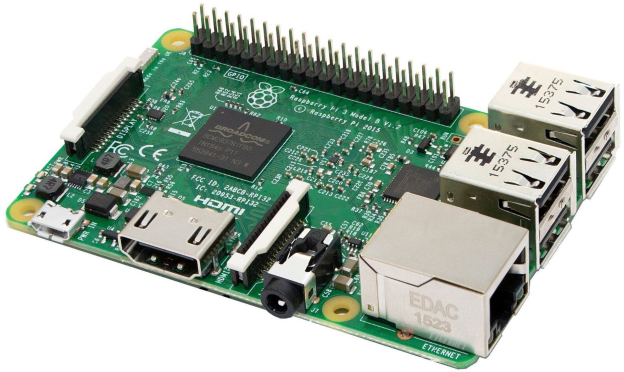
Minimum:

- Any modern 64 Bit Linux
- 1 x vCPU
- 1G RAM

Recommended:

- Ubuntu 22.04 LTS
- 4 x vCPU
- 4G RAM

The BNG Blaster runs almost everywhere, virtual machines, containers or even on a Raspberry Pi 3 Model B+.



Traffic Generator

- Different IO drivers (packet_mmap, raw and DPDK)
- > 10Gbps / 8M PPS / 10M flows
- ~200.000 PPS per thread / vCPU
- Traffic flows are automatically distributed over all threads based on PPS
- Traffic with sequence numbers and timestamps to calculate loss and jitter
- Traffic capture to PCAP file with optional filters
- Wireshark plugin available

Experimental DPDK Support:

- Ubuntu 22.04 (LTS)
- DPDK version 22.11.4 (LTS)
- Intel XL710 only
- > 40GBps / >10M PPS / 10M flows

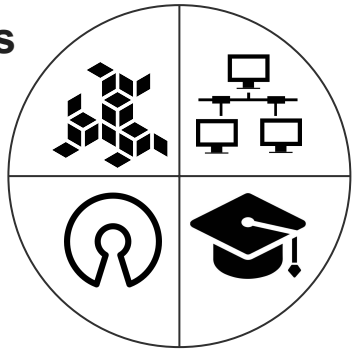
Details see official BNG Blaster performance guide:
https://rtbrick.github.io/bngblaster/performance.html

What can you do with BNG Blaster?

rtbrick

Network Hard and Software Vendors
Open Source Network Projects

- RFC conformance tests
- Interoperability tests
- Scaling tests
- Robustness tests
- Regression tests
- Reproduce customer bugs
- Customer demonstrations
- …



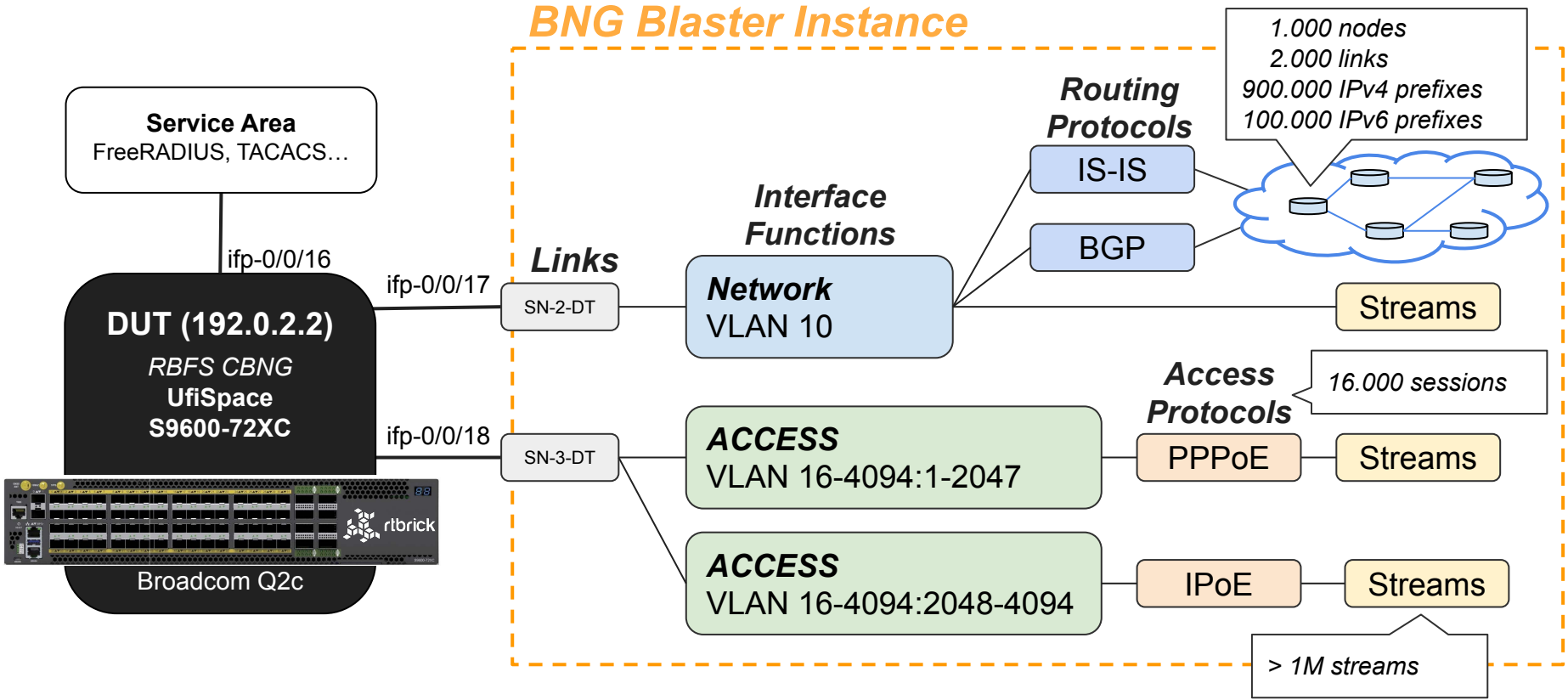
Network Providers and Integrators

- Configuration validation
- Software upgrade validation
- Network automation validation
- Integration testing
- Operator training
- Vendor comparison (tenders)
- …

Education

- Universities, Trainers, …
- Teach students on realistic environments
- …

DEMO SETUP


BNG Blaster Instance
1.000 nodes
2.000 links
900.000 IPv4 prefixes
100.000 IPv6 prefixes
Routing Protocols
IS-IS
BGP
Interface Functions
Links
Network
VLAN 10
Streams
Service Area
FreeRADIUS, TACACS…
ifp-0/0/16
DUT (192.0.2.2)
RBFS CBNG
UfiSpace
S9600-72XC
Broadcom Q2c
ifp-0/0/17
SN-2-DT
ifp-0/0/18
SN-3-DT
Access Protocols
16.000 sessions
ACCESS
VLAN 16-4094:1-2047
PPPoE
Streams
ACCESS
VLAN 16-4094:2048-4094
IPoE
Streams
> 1M streams

DEMO SETUP CONFIG

Interfaces:

```
"interfaces": {
    "network": [
        {
            "interface": "SN-2-DT",
            "address": "192.0.2.194/29",
            "gateway": "192.0.2.193",
            "address-ipv6": "2001:db8:2::2/64",
            "gateway-ipv6": "2001:db8:2::1",
            "vlan": 10
        }
    ],
    "access": [
        {
            "interface": "SN-3-DT",
            "outer-vlan-min": 16,
            "outer-vlan-max": 4094,
            "inner-vlan-min": 1,
            "inner-vlan-max": 2047,
            "type": "pppoe",
            "stream-group-id": 1
        }
    ]
},
```

Protocols:

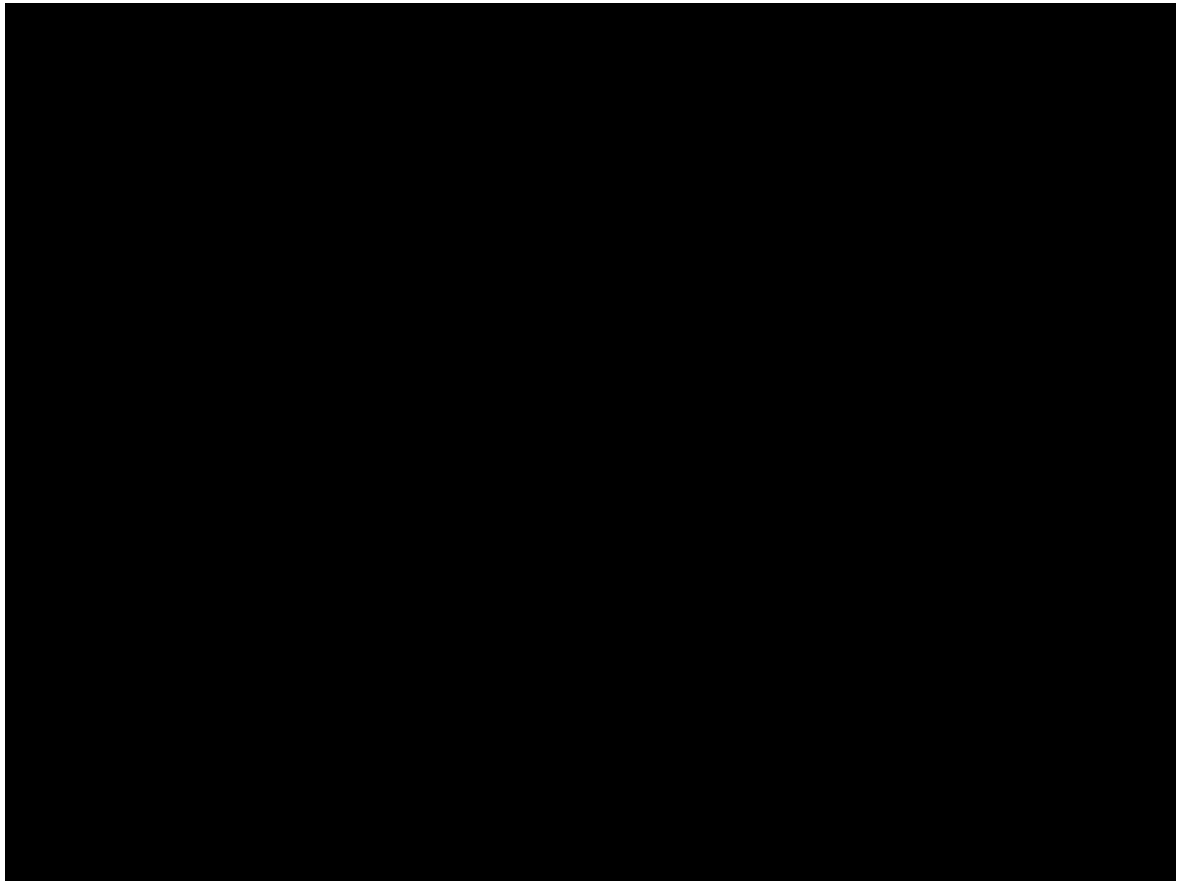
```
"pppoe": {
    "discovery-timeout": 3,
    "discovery-retry": 30,
    "host-uniq": true,
    "vlan-priority": 6
},
"ppp": {
    "mru": 1492,
    "authentication": {
        "username": "pppoe@rtbrick.local",
        "password": "test",
        "retry": 30
    }
},
```

Traffic

```
"session-traffic": {
    "autostart": true,
    "ipv4-pps": 1,
    "ipv6-pps": 1,
    "ipv6pd-pps": 1
},
"streams": [
    {
        "stream-group-id": 1,
        "name": "S1",
        "type": "ipv4",
        "direction": "both",
        "pps": 1000
    }
]
```

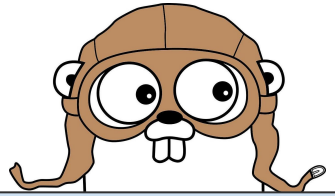
DEMO 01

rtbrick



BNG Blaster Controller

https://rtbrick.github.io/bngblaster/controller.html



rtbrick

Controller

{ REST }

- GET `/metrics` Metrics.
- GET `/api/v1/instances/{instance_name}` Status information of an instance.
- PUT `/api/v1/instances/{instance_name}` Create or update an instance
- DELETE `/api/v1/instances/{instance_name}` Delete an instance.
- POST `/api/v1/instances/{instance_name}/_start` Start an instance
- POST `/api/v1/instances/{instance_name}/_stop` Stop an instance
- POST `/api/v1/instances/{instance_name}/_kill` Kill an instance
- POST `/api/v1/instances/{instance_name}/_command` Send a command to the ctrl socket of the instance.
- GET `/api/v1/instances/{instance_name}/{file_name}` Download one of the output files.

Prometheus

test01

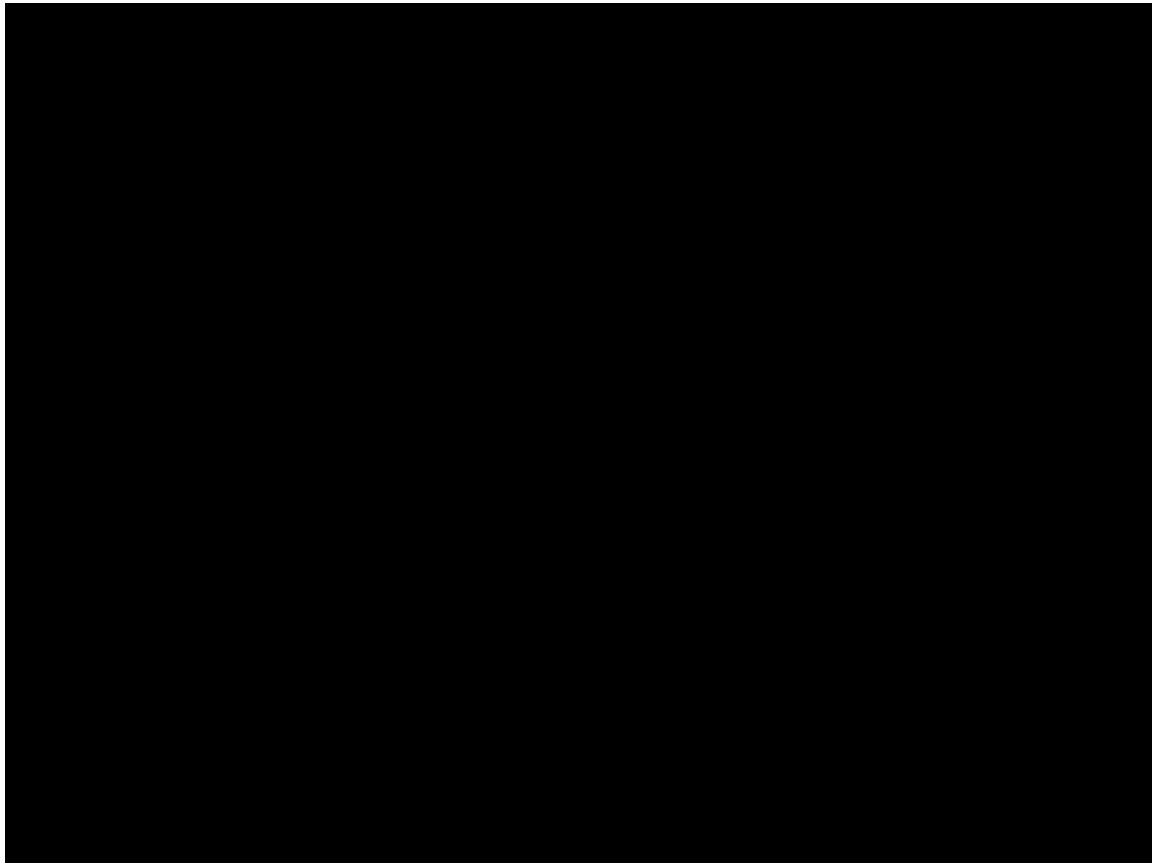
```
RUN: bngblaster -C config.json …

/var/bngblaster/test01/
 +    config.json: bngblaster configuration
 +    run.pid: bngblaster process ID (if running)
 +    run.json: bngblaster arguments
 +    run.log: bngblaster log file (if enabled)
 +    run_report.json: bngblaster report (if enabled)
 +    run.pcap: bngblaster traffic capture (if
      enabled)
 +    run.sock: bngblaster control socket
 +    run.stderr: bngblaster standard error
 +    run.stdout: bngblaster standard output
```

test02

test…

DEMO 02

rtbrick



Carrier Grade NAT

- Support for unidirectional and bidirectional traffic streams with NAT support
- RAW TCP Stream
  - UDP like traffic with TCP header
  - Used for NAT and ACL testing
  - > 10M streams supported

Config:

```
{
   "streams": [
       {
           "name": "NAT_S01",
           "stream-group-id": 1,
           "type": "ipv4",
           "setup-interval": 30,
           "direction": "both",
           "raw-tcp": true,
           "priority": 64,
           "source-port": 65001,
           "destination-port": 65056,
           "destination-ipv4-address": "10.0.0.1",
           "length": 100,
           "nat": true,
           "pps": 0.1
       }
   ]
}
```

Commands:

```
$ sudo bngblaster-cli run.sock stream-info flow-id 1
{
    "status": "ok",
    "code": 200,
    "stream-info": {
        "name": "NAT_S01",
        "type": "unicast",
        "sub-type": "ipv4",
        "direction": "upstream",
        "source-address": "100.64.0.2",
        "source-port": 65056,
        "destination-address": "10.0.0.1",
        "destination-port": 65056,
        "protocol": "tcp",  # udp or tcp
…
        "rx-source-ip": "192.0.2.8",
        "rx-source-port": 48523,
…
        "session-id": 1,
        "reverse-flow-id": 2
    }
}

$ sudo bngblaster-cli run.sock stream-update flow-id 1 tcp-flags fin
```

PUBLIC IP/PORT

rtbrick

BNG Blaster - RBFS CGNAT Demo (YouTube)

https://youtu.be/MoQrQc2R9AY?si=t0jxrEBfXzPsoqCe

HTTP Redirect

- HTTP client and server emulation
- PPPoE and IPoE
- Used for HTTP redirect and NAT testing
- lwIP TCP
- 200 HTTP clients with multiple sessions per client (no hard limit)

Config:

```
{
   "http-client": [
       {
           "http-client-group-id": 1,
           "name": "CLIENT-1",
           "url": "blaster.rtbrick.com",
           "destination-ipv4-address": "192.168.131.2",
           "destination-port": 80
       }
   ],
   "http-server": [
       {
           "name": "SERVER",
           "network-interface": "SN-3-DT",
           "ipv4-address": "192.168.131.2",
           "port": 80
       }
   ]
}
```

Commands:

```
$ sudo bngblaster-cli run.sock http-clients
{
    "status": "ok",
    "code": 200,
    "http-clients": [
        {
            "session-id": 1,
            "http-client-group-id": 1,
            "name": "CLIENT-1",
            "url": "blaster.rtbrick.com",
            "destination-address": "192.168.131.2",
            "destination-port": 80,
            "state": "closed",
            "response": {
                "minor-version": 1,
                "status": 200,
                "msg": "OK\r\nServer: BNG-Blaster\r\nX-...",
                "headers": [
                    {
                        "name": "Server",
                        "value": "BNG-Blaster"
                    },
                    {
                        "name": "X-Client-Ip",
                        "value": "100.100.100.1"
                    },
                    {
                        "name": "X-Client-Port",
                        "value": "1033"
                    }
                      …
```

PUBLIC IP/PORT

DPDK

rtbrick

```
F1: Select View  F7/F8: Start/Stop Traffic  F9: Terminate Sessions
F3: Select A10NSP Interface  Left/Right: Access Interface

Test Duration: 126s All Sessions Established: 117s

Sessions            4000 (0 PPPoE / 4000 IPoE)
  Established       4000 [############################################################]
  Outstanding          0 [                                                            ]
  Terminated           0 [                                                            ]
  DHCPv4       4000/4000 [############################################################]
  Setup Time        7108 ms
  Setup Rate      562.75 CPS (MIN: 354.92 AVG: 514.36 MAX: 564.97)
  Flapped              0

Traffic Flows Verified
  Stream 12800000/12800000 [############################################################]

A10NSP Interface ( 0000:01:00.0 )
  TX Packets                530322844 |9257885 PPS    8739443 Kbps 8.739 Gbps
  RX Packets                534947136 |9418927 PPS    8891467 Kbps 8.891 Gbps
  TX Stream Packets         530302844 |9257885 PPS
  RX Stream Packets         534927136 |9418927 PPS          0 Loss (0.000%)

Access Interface ( 0000:01:00.2 )
  TX Packets                534947348 |9418935 PPS    8891475 Kbps 8.891 Gbps
  RX Packets                530322547 |9257842 PPS    8739403 Kbps 8.739 Gbps
  TX Stream Packets         534927348 |9418935 PPS
  RX Stream Packets         530302547 |9257842 PPS          0 Loss (0.000%)
  RX Multicast Packets              0 |      0 PPS          0 Loss
```

```
Mar 04 18:37:59.446870 DPDK: init the EAL
Mar 04 18:38:01.846632 DPDK: version DPDK 22.11.4
Mar 04 18:38:01.846639 DPDK: 4 ports available
Mar 04 18:38:01.846644 DPDK: interface 0000:01:00.0 (0) driver net_i40e firmware 9.40 0x8000ecf8 0.0.0
Mar 04 18:38:01.846647 DPDK: interface 0000:01:00.0 (0) max queues rx 192 tx 192
Mar 04 18:38:01.846651 DPDK: interface 0000:01:00.1 (1) driver net_i40e firmware 9.40 0x8000ecf8 0.0.0
Mar 04 18:38:01.846653 DPDK: interface 0000:01:00.1 (1) max queues rx 192 tx 192
Mar 04 18:38:01.846655 DPDK: interface 0000:01:00.2 (2) driver net_i40e firmware 9.40 0x8000ecf8 0.0.0
Mar 04 18:38:01.846661 DPDK: interface 0000:01:00.2 (2) max queues rx 192 tx 192
Mar 04 18:38:01.846664 DPDK: interface 0000:01:00.3 (3) driver net_i40e firmware 9.40 0x8000ecf8 0.0.0
Mar 04 18:38:01.846666 DPDK: interface 0000:01:00.3 (3) max queues rx 192 tx 192
Mar 04 18:38:01.846703 DPDK: interface 0000:01:00.0 (0) MAC address 64:9d:99:b1:83:f0
Mar 04 18:38:01.958605 DPDK: interface 0000:01:00.0 (0) link up (speed 10000 Mbps full-duplex)
Mar 04 18:38:01.958652 DPDK: interface 0000:01:00.2 (2) MAC address 64:9d:99:b1:83:f2
Mar 04 18:38:02.070322 DPDK: interface 0000:01:00.2 (2) link up (speed 10000 Mbps full-duplex)
Mar 04 18:39:00.567684 Total PPS of all streams: 12800000.00
Mar 04 18:39:00.806349 Opened control socket run.sock
Mar 04 18:39:02.458960 Resolve network interfaces
Mar 04 18:39:02.459188 All network interfaces resolved
Mar 04 18:39:09.573700 ALL SESSIONS ESTABLISHED
Mar 04 18:39:12.664082 ALL STREAM TRAFFIC FLOWS VERIFIED
```

4000 IPoE Sessions
20 Gbps Bidirectional
20M PPS

```
{
   "interfaces": {
       "io-slots": 16384,
       "links": [
           {
               "interface": "0000:01:00.0",
               "io-mode": "dpdk",
               "rx-threads": 16,
               "rx-cpuset": [0,1,2,3],
               "tx-threads": 3,
               "tx-cpuset": [4,5,6]
           },
           {
               "interface": "0000:01:00.2",
               "io-mode": "dpdk",
               "rx-threads": 16,
               "rx-cpuset": [7,8,9,10],
               "tx-threads": 3,
               "tx-cpuset": [11,12,13]
           }
       ]
   }
}
```

Who is using BNG Blaster?

rtbrick

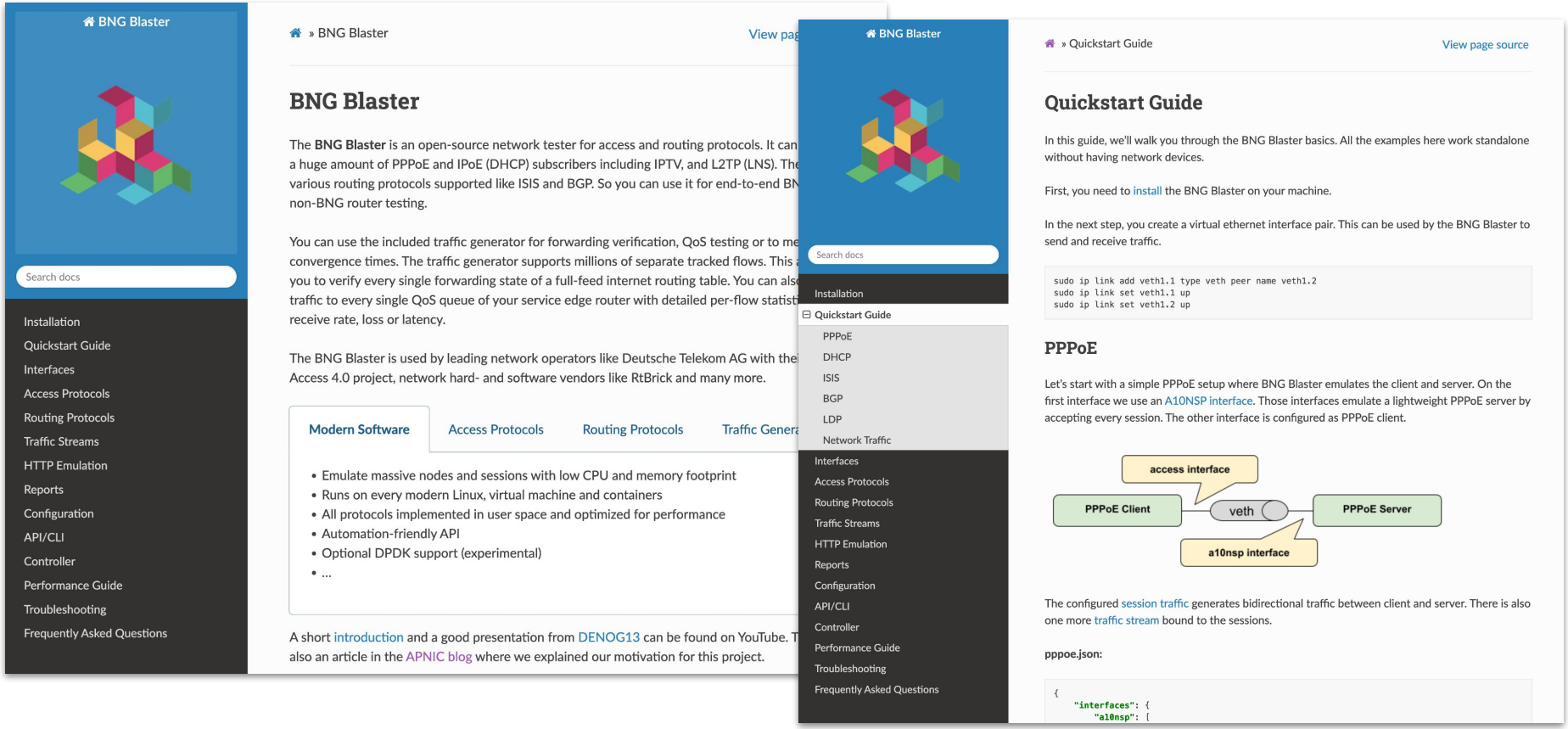
The BNG Blaster is used by leading network operators, network hardware and software vendors …

ACCESS 4.0

… and many more!

How to start with BNG Blaster?

https://rtbrick.github.io/bngblaster



We want you!

Contribute to the project and share your experience!

bngblaster@rtbrick.com

https://github.com/rtbrick/bngblaster

https://matrix.to/#/#bngblaster:matrix.org

Questions?