rtbrick

BNG Blaster

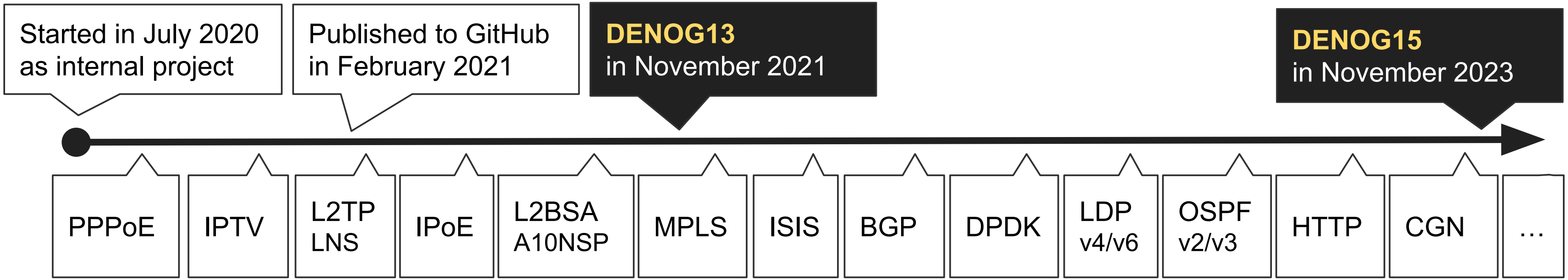
Open Source Network Tester

Routing Protocol Update

Christian Giese | Field Services and Software Engineer | RtBrick

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
- …



Started in July 2020 as internal project

Published to GitHub in February 2021

DENOG13 in November 2021

DENOG15 in November 2023

PPPoE | IPTV | L2TP LNS | IPoE | L2BSA A10NSP | MPLS | ISIS | BGP | DPDK | LDP v4/v6 | OSPF v2/v3 | HTTP | CGN | …

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

Features

- 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
- …

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
- …

Routing Protocols

- Emulate IS-IS and OSPFv2/v3 topologies with thousands of nodes
- Support for LSA/LSP flapping, refresh, purge, …
- Support for Segment Routing
- Setup thousands of BGP sessions with millions of prefixes
- Support for LDPv4/v6
- Verify MPLS labels for millions of flows
- …

Features

HTTP Redirect and 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
  - > 1M streams supported
- HTTP client and server
  - Emulate HTTP traffic over PPPoE or IPoE sessions
  - Used for HTTP redirect and NAT testing
- …

```
$ sudo bngblaster-cli run.sock stream-info flow-id 1
{
    "status": "ok",
    "code": 200,
    "stream-info": {
        "name": "UDP1",
        "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": "udp", # udp or tcp
…
        "rx-source-ip": "192.0.2.8",
        "rx-source-port": 48523,
…
        "session-id": 1,
        "reverse-flow-id": 2
    }
}
```

BNG Blaster Requirements

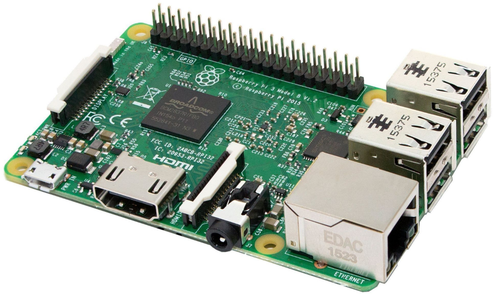
Minimum:

- 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 / 1M PPS
- 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:

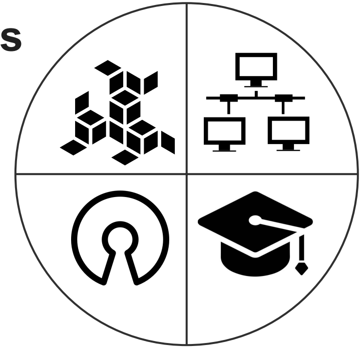
- Ubuntu 22.04 (LTS)
- DPDK version 21.11.2 (LTS)
- Intel XL710 only
- > 40GBps / 10M PPS

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

What can you do with BNG Blaster?

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
- …



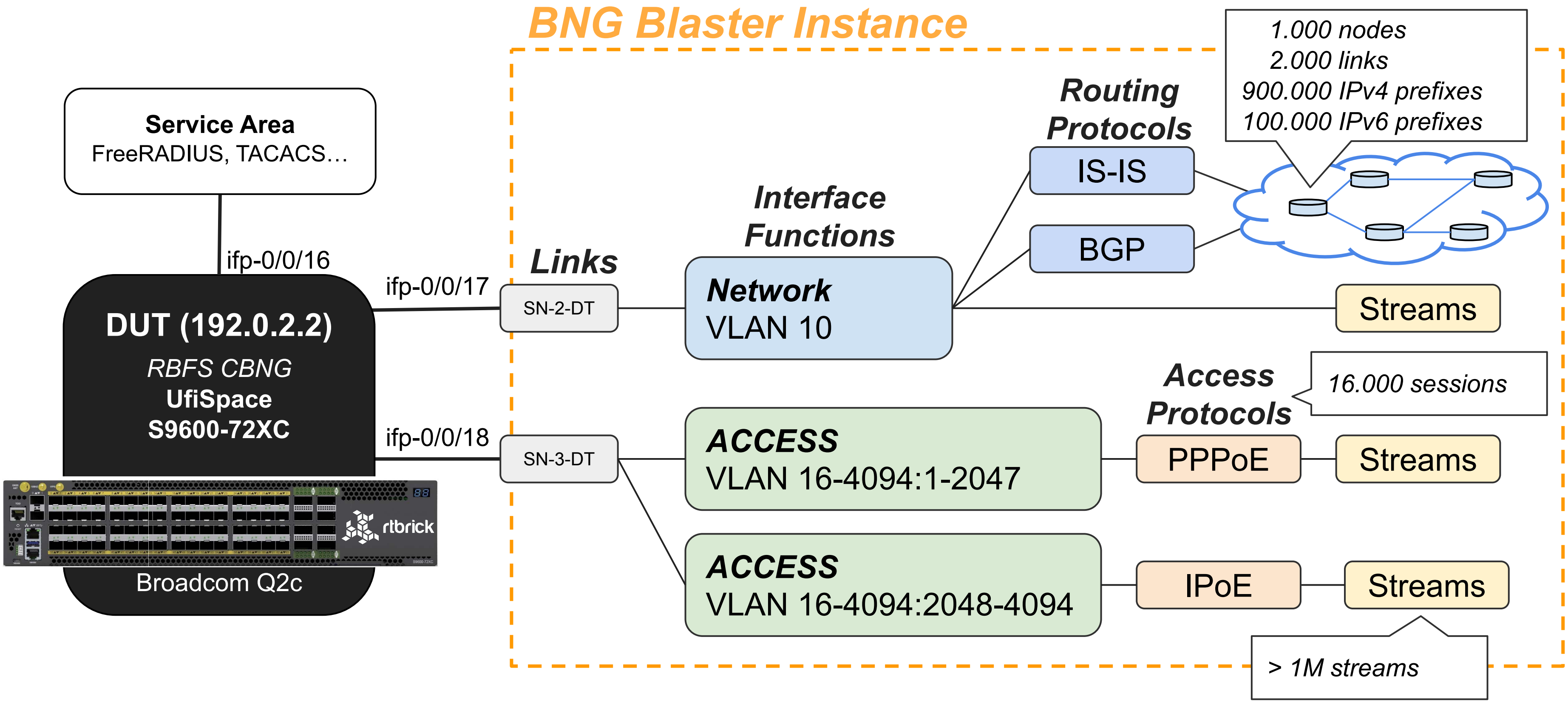
Network Providers and Integrators

- 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



Service Area
FreeRADIUS, TACACS…

ifp-0/0/16

DUT (192.0.2.2)
RBFS CBNG
UfiSpace
S9600-72XC
Broadcom Q2c

ifp-0/0/17

ifp-0/0/18

BNG Blaster Instance

Links

SN-2-DT

SN-3-DT

Interface Functions

Network
VLAN 10

ACCESS
VLAN 16-4094:1-2047

ACCESS
VLAN 16-4094:2048-4094

Routing Protocols

IS-IS

BGP

1.000 nodes
2.000 links
900.000 IPv4 prefixes
100.000 IPv6 prefixes

Streams

Access Protocols

16.000 sessions

PPPoE

Streams

IPoE

Streams

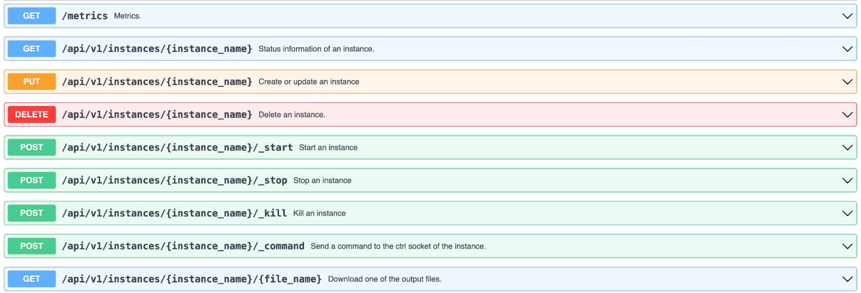
> 1M streams

DEMO 01

BNG Blaster Controller

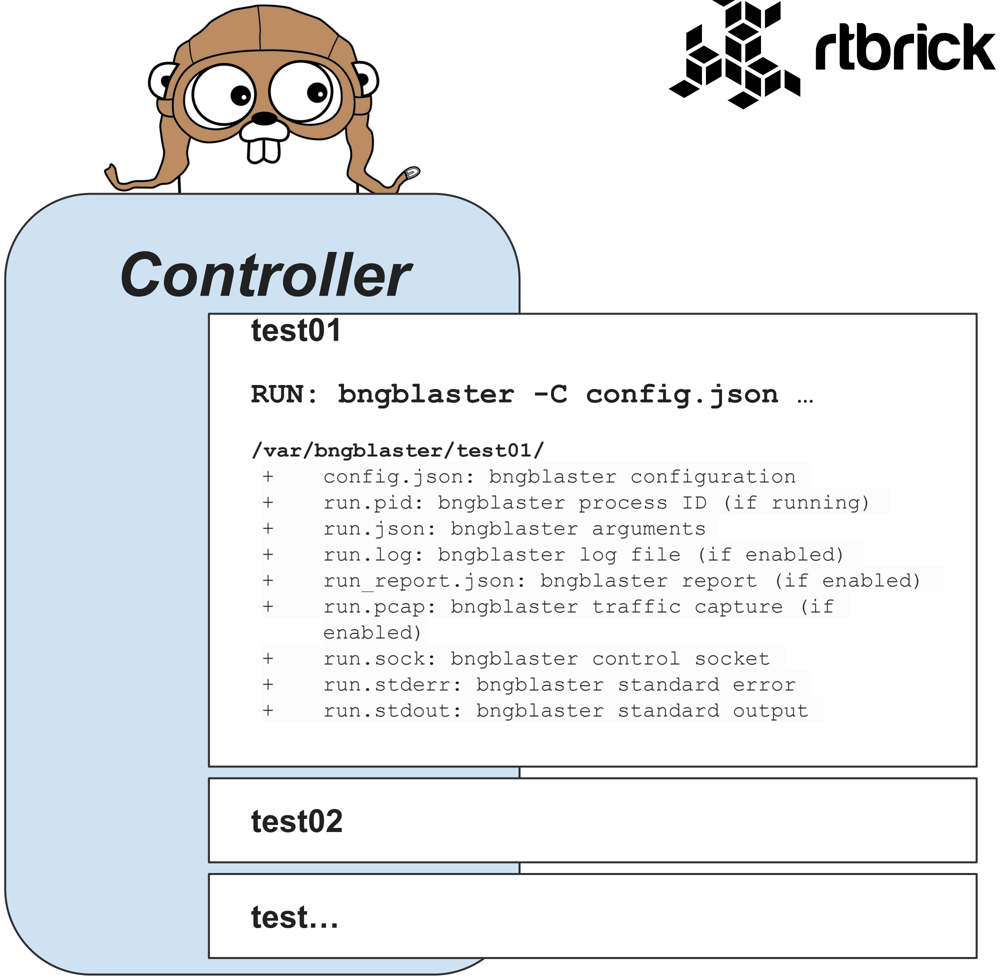
https://rtbrick.github.io/bngblaster/controller.html

{ 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



Controller

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

Who is using BNG Blaster?

rtbrick

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

rtbrick

Deutsche Telekom

ACCESS 4.0

WOBCOM

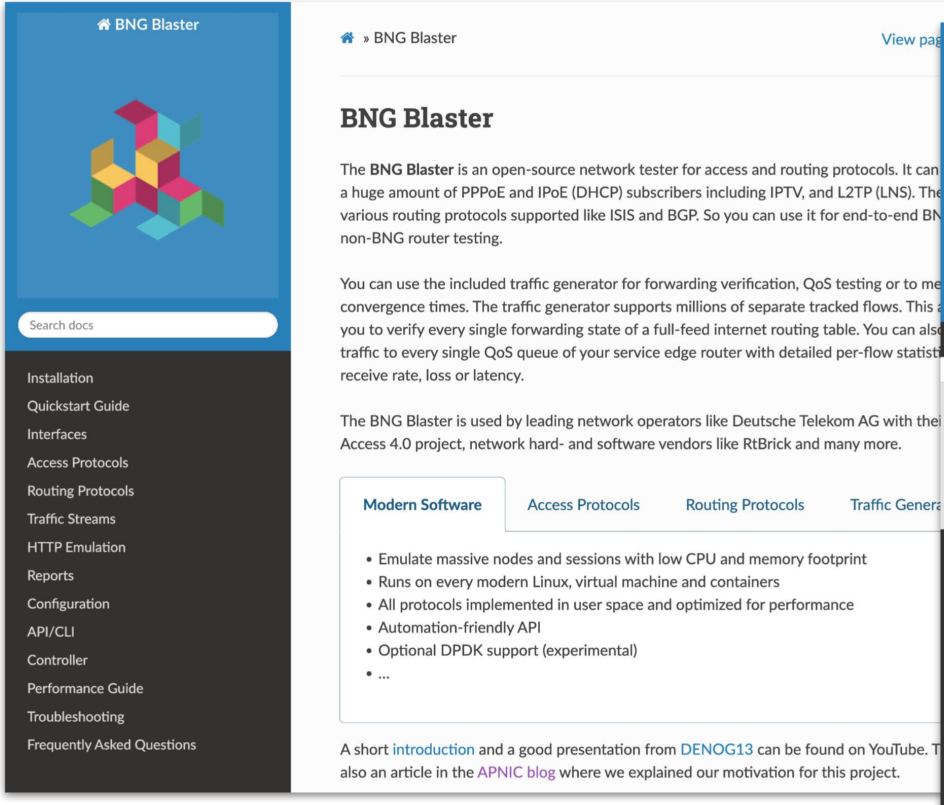
WDZ

glasfaser nordwest

… and many more!

How to start with BNG Blaster?

https://rtbrick.github.io/bngblaster



BNG Blaster

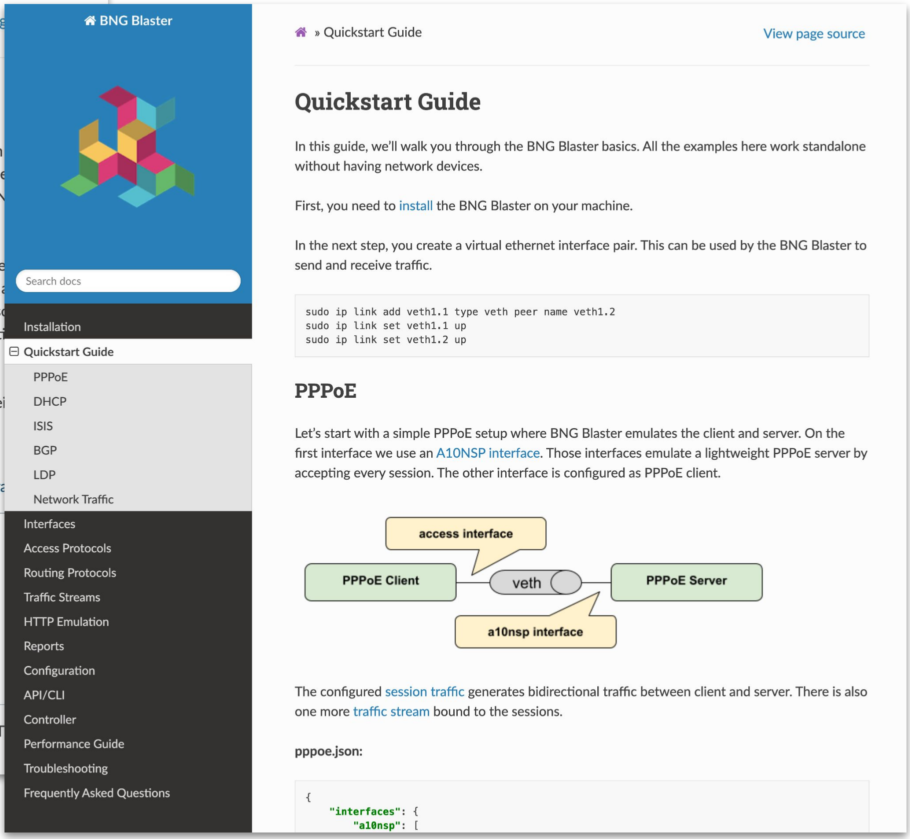
The **BNG Blaster** is an open-source network tester for access and routing protocols. It can a huge amount of PPPoE and IPoE (DHCP) subscribers including IPTV, and L2TP (LNS). The various routing protocols supported like ISIS and BGP. So you can use it for end-to-end BN non-BNG router testing.

You can use the included traffic generator for forwarding verification, QoS testing or to me convergence times. The traffic generator supports millions of separate tracked flows. This a you to verify every single forwarding state of a full-feed internet routing table. You can also traffic to every single QoS queue of your service edge router with detailed per-flow statisti receive rate, loss or latency.

The BNG Blaster is used by leading network operators like Deutsche Telekom AG with thei Access 4.0 project, network hard- and software vendors like RtBrick and many more.

- Emulate massive nodes and sessions with low CPU and memory footprint
- Runs on every modern Linux, virtual machine and containers
- All protocols implemented in user space and optimized for performance
- Automation-friendly API
- Optional DPDK support (experimental)
- ...

A short introduction and a good presentation from DENOG13 can be found on YouTube. T also an article in the APNIC blog where we explained our motivation for this project.



Quickstart Guide

In this guide, we'll walk you through the BNG Blaster basics. All the examples here work standalone without having network devices.

First, you need to install the BNG Blaster on your machine.

In the next step, you create a virtual ethernet interface pair. This can be used by the BNG Blaster to send and receive traffic.

```
sudo ip link add veth1.1 type veth peer name veth1.2
sudo ip link set veth1.1 up
sudo ip link set veth1.2 up
```

PPPoE

Let's start with a simple PPPoE setup where BNG Blaster emulates the client and server. On the first interface we use an A10NSP interface. Those interfaces emulate a lightweight PPPoE server by accepting every session. The other interface is configured as PPPoE client.

The configured session traffic generates bidirectional traffic between client and server. There is also one more traffic stream bound to the sessions.

pppoe.json:

```
{
    "interfaces": {
        "a10nsp": [
```

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?