



IETF #106 - BMWG

Methodology for VNF Benchmarking Automation **-05**

R. Rosa, C. Rothenberg, M. Peuster, H. Karl

Context, In version -04: Which were the major technical changes?

- Filtered only important concepts in Terminology
- Considerations on Benchmarking Procedures
 - ◆ Generic Phases (I to IV): Deployment, Configuration, Execution, Report
- **Refined VNF Benchmarking Descriptor (VNF-BD) structure (Sec. 6.1)**
 - ◆ Description Headers: VNF-BD versioning, authorship, description, etc
 - ◆ Target Information: VNF (SUT) descriptor (version, image, etc)
 - ◆ Experiments: Defines overall VNF-BD parameters: repetition of Trials, Tests, Method
 - ◆ Environment: Settings referring to components (e.g., orchestrator) to deploy scenario
 - ◆ Scenario: Topology for Tests
 - ◆ Proceedings: Agent(s)/Monitor(s) settings for (prober(s)/listener(s)) Test parameters
- VNF-BD Yang model updated
- Gym updated reference to open source repository

Why the draft was updated?

- Needed to unify the models to describe VNF bench. experiments (VNF-BD)
- Need of clear considerations regarding VNF Performance Profiles (Sec. 6.2)
- We did experimental analysis with VNF-BD and VNF-PP Yang models
 - ◆ Comparison factors (i.e., VNF-BD fully functional)
- Address comments in the mailing-list written by Luis M. Contreras
- More comments are coming in and are work in progress

Which are the major technical changes?

→ Refined VNF Performance Profile (VNF-PP) structure (Sec. 6.2)

- ◆ Description Headers: VNF-PP versioning, authorship, description, etc
- ◆ Reports: structure **Test** results from Agent(s)/Monitor(s)
 - Execution Environment: description of hardware/software specs of VNF-BD scenario
 - Snapshots: structured **Trial** results by each Agent/Monitor
 - Origin: Agent/Monitor identification (id, hostname, etc)
 - Evaluations: structured prober(s)/listener(s) result data (measurements)
 - ◆ Source: prober/listener identification (id, tool, version, command call, etc)
 - ◆ Metrics: list of metrics provided by prober/listener (name, unit, value, type)

→ VNF-BD and VNF-PP models updated and aligned with IETF Yang specs

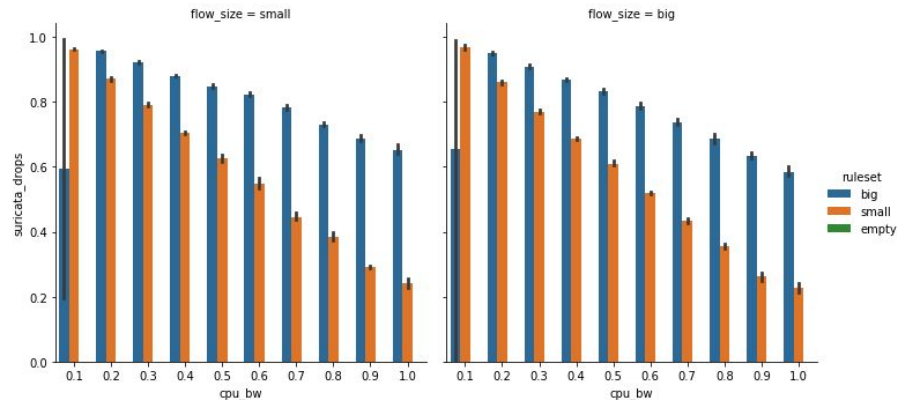
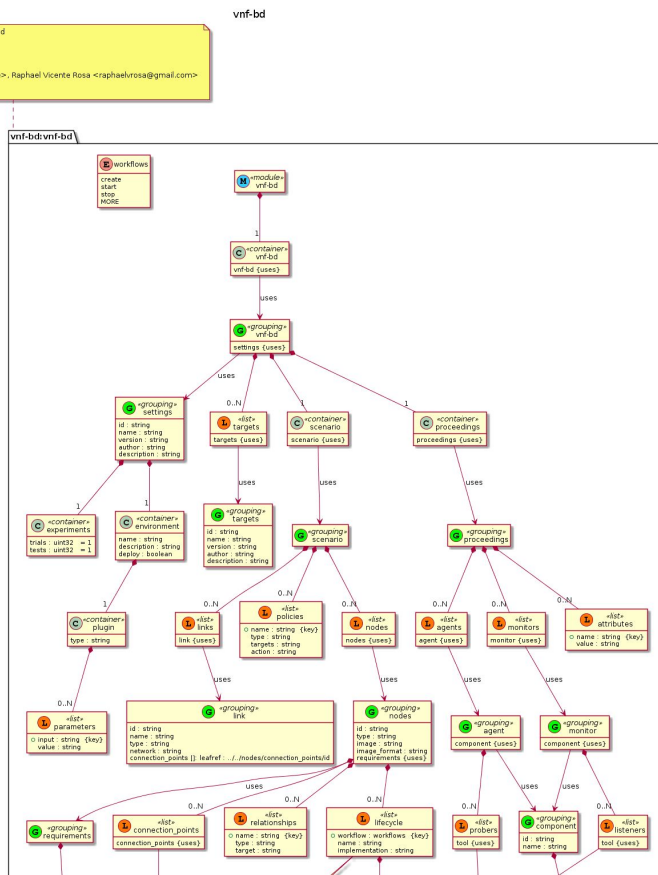
- ◆ <https://github.com/raphaelvrosa/vnf-bench-model/tree/master/vnf-br/yang>

→ Published Gym and Tng-bench comparison tests (ipynb)

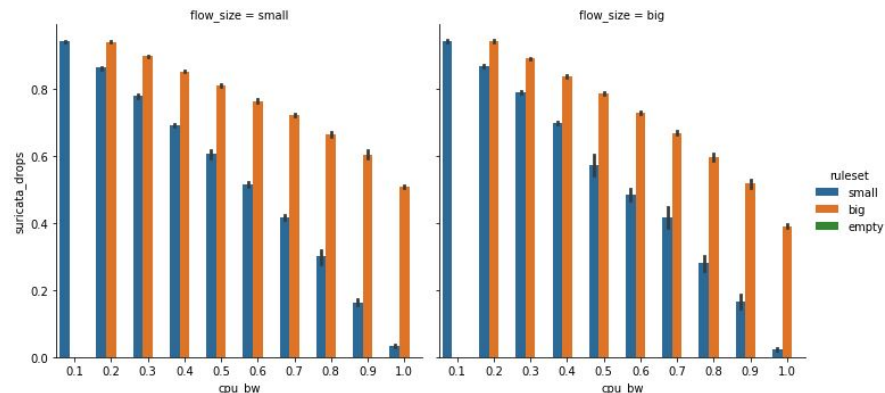
- ◆ <https://github.com/raphaelvrosa/vnf-bench-model/tree/master/experiments>

Models and Example Results

NameSpace: urn:ietf:params:xml:ns:yang:vnf-bd
 Prefix: vnf-bd
 Organization: IETF/BMG
 Contact: Manuel Feuster <peuster@mail.uni-paderborn.de>, Raphael Vicente Rosa <rphaelvrosa@gmail.com>
 Revision: 2019-08-13



Results recorded by Gym



Results recorded by tng-bench

Which issues are unresolved? Which issues needs further discussion.

- VNF Performance Profile in open source reference implementations
 - ◆ Fully comparison of VNF-BD and VNF-PP
- Structure VNF Benchmark Report
 - ◆ Establish yang model for VNF-BR
 - ◆ Useful/Summary joint information from VNF-BD and VNF-PP
 - e.g., `performance_metrics = F(traffic_workload, allocated_resources, VNF_configuration)`
- Synergies (alignment/collaboration) with BMWG related work
 - ◆ Considerations for Benchmarking Network Performance in Containerized Infrastructures
 - ◆ Considerations for Benchmarking Network Virtualization Platforms
 - ◆ **A YANG Data Model for Network Interconnect Tester Management**
 - ◆ ... others?

Final Remarks

- Considering:
 - Draft in version -05
 - Comments in mailing list addressed
 - VNF-BD and VNF-PP Yang models
 - Reference tools implementing draft methodology
- **We ask BMWG to adopt the draft**
 - **We have support from industry and academia (mailing-list comments)**
 - **We are going move forward on refining the draft**
 - **We are going to prototype the VNF-BR Yang model in the reference implementations**



Thank you!



Backup

- ❖ Why?
 - “If VNFs deployments can be fully automated, VNF benchmarking should be automated as well!”
 - Concept: Design and specify a **generic workflow to automatically execute arbitrary pre-defined VNF benchmarking experiments.**
- ❖ We define **how to automate** the benchmarking process, **not how to benchmark** → highly depends on the SUT
- ❖ **Two open-source reference implementations**
 - Gym [1][2]
 - 5GTANGO benchmarker “tng-bench” [3][4]

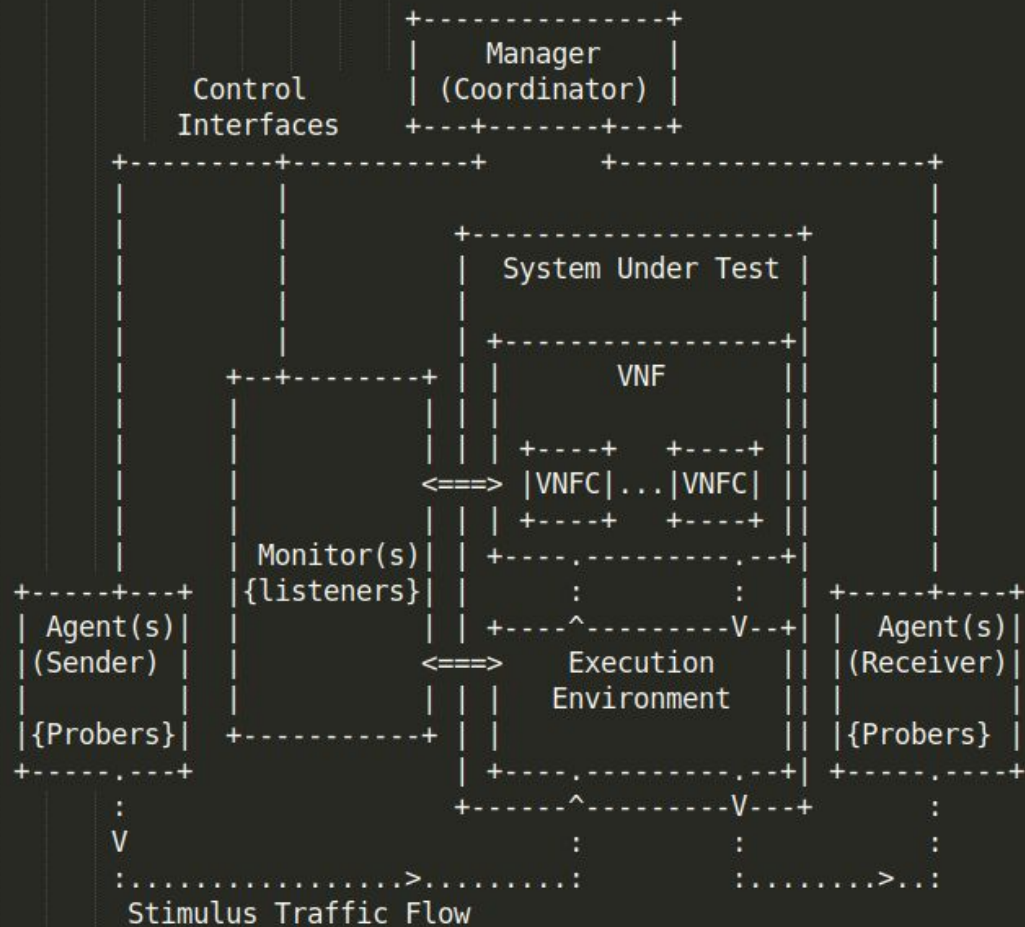


Figure 1: Generic VNF Benchmarking Setup

Backup

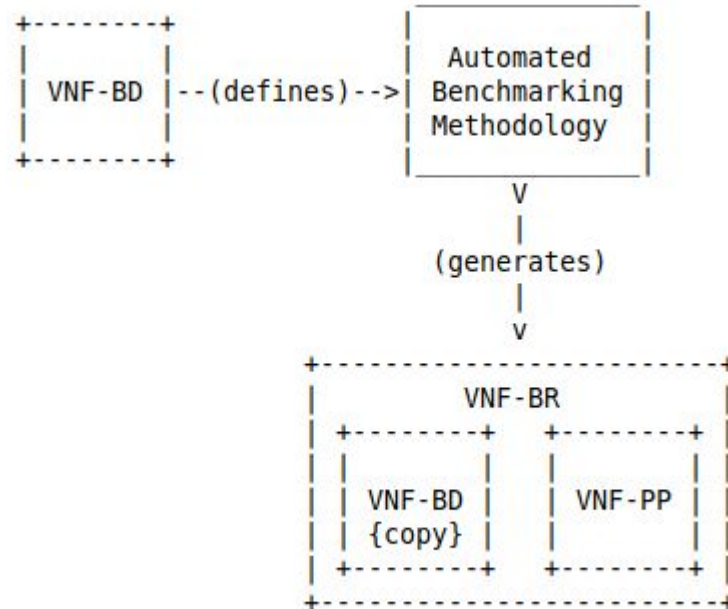


Figure 2: VNF benchmarking process inputs and outputs

Backup: Example Results

- SUT: Suricata IDS VNF deployed in a Docker container
- Parameters
 - Different IDS rulesets
 - Different number of vCPU cores
 - Different amounts of CPU bandwidth (CPU time)
 - Different memory limits
- Stimulation
 - Traffic traces with small and big flows
- Experiments executed without human interaction using benchmarking descriptors
- Everything open: <https://github.com/raphaelvrosa/vnf-bench-model>

References

- [1] R. Rosa, C. Bertoldo, C. Rothenberg, "Take your VNF to the Gym: A Testing Framework for Automated NFV Performance Benchmarking", IEEE Communications Magazine Testing Series , Sept 2017, <<http://ieeexplore.ieee.org/document/8030496>>.
- [2] "Gym Home Page", <<https://github.com/intrig-unicamp/gym>>.
- [3] M. Peuster, H. Karl, "Profile Your Chains, Not Functions: Automated Network Service Profiling in DevOps Environments", IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) , 2017, <<http://ieeexplore.ieee.org/document/8169826/>>.
- [4] "5GTANGO VNF/NS Benchmarking Framework", <<https://github.com/sonata-nfv/tng-sdk-benchmark>>.
- [5] YANG Models: <https://github.com/raphaelvrosa/vnf-bench-model/tree/master/vnf-br/yang>
- [6] Example Results: <https://github.com/raphaelvrosa/vnf-bench-model/tree/master/experiments>