BB-Gen: A Packet Crafter for P4 Target Evaluation

Fabricio Rodriguez
University of Campinas

P Gyanesh Kumar Patra
University of Campinas

Levente Csikor
University of Campinas

Christian Rothenberg
University of Campinas

Péter Vörös, Sándor Laki
ELTE Eötvös Loránd University

Gergely Pongrácz
Ericsson Research

ABSTRACT

With P4 gaining traction to define datapath pipelines along auto-generated control plane APIs, the protocol-independence and increased flexibility add non-trivial hazards when it comes to functional and in-depth performance evaluation. P4-dependent workload traces are needed along automated methods to populate the tables of the datapath under test accordingly. Without proper tools, manual efforts are required for tedious tasks such as creating appropriate PCAP traces, defining the distribution of field values, and inserting entries in the pipeline tables. To this end, we present BB-Gen, a packet crafter and table generator tool that given a P4 application and a corresponding user configuration results in packet and table traces to carry automated performance evaluation tasks. We demonstrate BB-Gen with P4 applications of increasing complexity (from L2 to VXLAN-based Data Center Gateway), using two different multi-architecture backend compilers (MACSAD, T4P4S) and different targets.

CCS CONCEPTS

• **Networks → Network performance analysis**; **Network measurement**; *Programmable networks*; • **Hardware →** *Emerging languages and compilers*;

KEYWORDS

P4, Software Defined Networking, Performance analysis

ACM Reference Format:
Fabricio Rodriguez, P Gyanesh Kumar Patra, Levente Csikor, Christian Rothenberg, Péter Vörös, Sándor Laki, and Gergely Pongrácz. 2018. BB-Gen: A Packet Crafter for P4 Target Evaluation. In *SIGCOMM Posters and Demos '18: ACM SIGCOMM 2018 Conference Posters and Demos, August 20–25, 2018, Budapest, Hungary.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3234200.3234229

1 INTRODUCTION

The advent of flexible fast path packet processing research in Software Defined Networking (SDN) infrastructures, most notably fueled by the protocol-independence DSL (Domain-Specific Language) approach of P4 [1], exacerbate the challenges of carrying proper performance evaluation of datapath devices. Due to the evolving traffic demands and communication patterns , different packet processing technologies (e.g., DPDK, Netmap), diverse communication topologies, and use cases turn answering the traditionally simple question *"How fast is this datapath device"* extraordinarily painstaking. Existent generic traffic generation and benchmarking tools are largely based on traditional, fixed definitions of flow traces providing limited synthetic means to generate and analyze test traffic, and configure the datapath.

During our last two years journey on multi-architecture P4 compilers [7, 10] running rich experimental evaluation of diverse P4 applications over heterogeneous targets (e.g., Netmap/DPDK/ODP, ARM, x86, 10G/100G NICs), we have experienced, first hand, the required efforts to carry such work due to the combinatorial factors involved. Given a P4 application, *flexibility* the blessing of P4 becomes a curse when generating corresponding traffic traces and P4 table entries for the target Device Under Test (DUT).

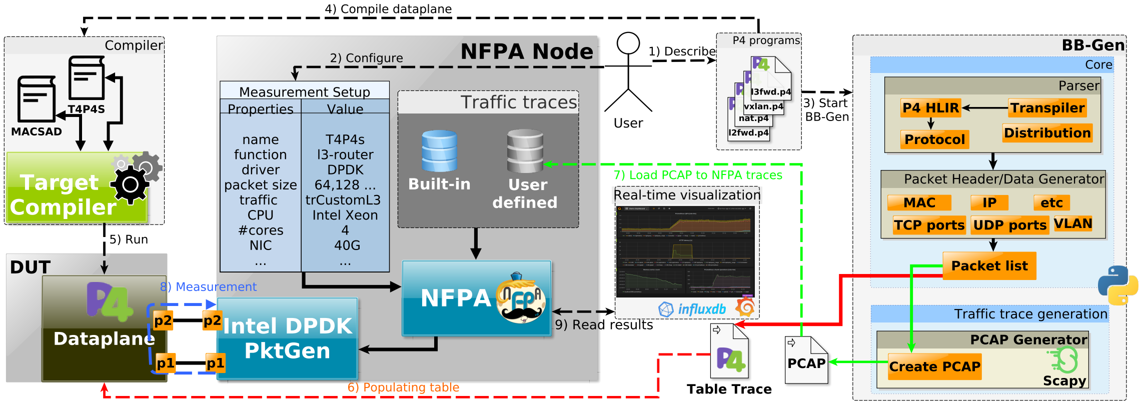
In this demo, we present BB-Gen [12–14] to overcome the aforementioned hazards of agile P4 performance evaluation, showcase it's simplicity and effectiveness to generate network traffic and P4 table entries for different P4 use cases with augmenting complexity while increasing number of tables (i.e., l2fwd & l3fwd, VXLAN, NAT), custom number of entries and packet sizes, target compilers, and platforms.

2 BB-GEN ARCHITECTURE & DEMO

Following a design approach prioritizing simplicity, component re-usability, and reproducibility, BB-Gen is a packet crafter that generates packet flows formatted as PCAP files

SIGCOMM Posters and Demos '18, August 20–25, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5915-3/18/08…$15.00
https://doi.org/10.1145/3234200.3234229



Measurement Setup	
Properties	Value
name	T4P4s
function	l3-router
driver	DPDK
packet size	64,128 ...
traffic	trCustomL3
CPU	Intel Xeon
#cores	4
NIC	40G
...	...

Figure 1: BB-Gen architecture and demo workflow featuring integration with NFPA and Intel DPDK.

from a given P4 file input. BB-Gen complements other benchmarking tools instead of trying to replace them; here, we chose Network Function Performance Analyzer (NFPA) [5] that is used for measuring performance matrices of programmable dataplanes and utilizes PktGen as the traffic generator supporting custom PCAP trace files.

The architecture is depicted in Fig. 1, where the dashed arrows with numbered labels show the workflow step-by-step: first, the user defines her P4 program and configures (step 2) the benchmarking tool accordingly (in our case, NFPA only requires the high-level configuration details, e.g., packet size). Then, the P4 program is fed into BB-gen (step 3), which contains two main modules: (*i*) *Core*: comprised by Parser and Packet Header/Data Generator components; first, the P4 code is parsed to obtain packet header information. According to the field value distributions (chosen by the user), the packet descriptors are generated; (*ii*) *Scapy* module: for assembling the packets included in the packet list descriptor resulting in the final PCAP file. The tool allows to use well-known standard protocols as well as easily set custom packet headers through template-based modifications. Users can define target PCAPs with different packet sizes (i.e., from 64 to 1500 Bytes) from best-case (i.e., fixed, single header fields) to worst-case (random, unique field values) workloads.

While the packets are generated, the DUT is compiled by our Target Compiler module (step 4) supporting multi-architecture P4 compilers, such as MACSAD [11] and T4P4S.

Once the DUT is running (step 5), BB-gen uses the available APIs to carefully populate the flow tables (step 6) using the P4 Table Trace derived from PCAP packet trace and Packet List. At the same time, BB-gen loads the generated PCAP file into NFPA (step 7), which handles practical measurements (step 8) conforming the standards (RFC 2544 [2]).

During the demo. Attendees will be presented with a choice of P4 programs (e.g., L2fwd, L3fwd IPv4/v6, GRE, VXLAN) [9] and two P4 compilers (MACSAD, T4P4S) in addition to further user-defined configuration options of NFPA to effectively shape the experimental scenarios showcasing BB-Gen's ability to generate P4 table traces and PCAP files, and populate the DUT flow tables accordingly. By means of a real-time visualization component developed as an extension to NFPA, the results can be shown on-the-fly.

3 CONCLUSIONS AND ONGOING WORK

This demo shows how BB-Gen, NFPA and "MACSAD & T4P4S" trio can work conjointly to illustrate how a single P4 program suffices to define the datapath pipeline, create match + action table traces to control the P4 DUT, and generate a suitable PCAP file used for performance evaluation. BB-Gen applicability to MACSAD and T4P4S projects for varying P4 use case complexities suggest that other practitioners could benefit from such a tool in their evaluation workflows, as well as the broader community striving for research reproducibility, e.g., by re-using traces, toolchains, and evaluation methodologies.

BB-Gen effectively complements other artifacts in the toolbox of P4 developers, such as `p4pktgen` [8] to validate a P4 program by generating test input packets and table entries, `p4app` [3] to perform functional tests for P4 program using BMV2 simple_switch in Mininet environment, and `P4benchmark` [6] to test target compilers by generating different P4 applications with variable complexities.

Future work. Integration with `P4Runtime` [4] to control the DUT, leveraging the possibility to configure the P4 table entries. BB-Gen will allow to rely on Scapy's extensibility features to create packets with arbitrary sequence of headers that might not conform to standards, but an unorthodox P4 program may require it. For faster packet crafting, we are considering Hexdump PCAP generation and NetFPGA/OSNT.

4 ACKNOWLEDGMENTS

This work was supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil under grant agreement UNI.63.

REFERENCES

[1] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: Programming Protocol-independent Packet Processors. *SIGCOMM Comput. Commun. Rev.* 44, 3 (July 2014), 87–95. https://doi.org/10.1145/2656877.2656890

[2] Scott Bradner and Jim McQuaid. 1999. *Benchmarking Methodology for Network Interconnect Devices*. RFC 2544. RFC Editor. http://www.rfc-editor.org/rfc/rfc2544.txt http://www.rfc-editor.org/rfc/rfc2544.txt.

[3] P4 Language Consortium. 2016. p4app. https://github.com/p4lang/p4app.

[4] P4 Language Consortium. 2017. P4 Runtime. https://p4.org/p4-runtime/.

[5] L. Csikor, M. Szalay, B. Sonkoly, and L. Toka. 2015. NFPA: Network function performance analyzer. In *2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*. 15–17. https://doi.org/10.1109/NFV-SDN.2015.7387395

[6] Huynh Tu Dang, Han Wang, Theo Jepsen, Gordon Brebner, Changhoon Kim, Jennifer Rexford, Robert Soulé, and Hakim Weatherspoon. 2017. Whippersnapper: A p4 language benchmark suite. In *Proceedings of the Symposium on SDN Research*. ACM, 95–101.

[7] Sándor Laki, Dániel Horpácsi, Péter Vörös, Róbert Kitlei, Dániel Leskó, and Máté Tejfel. 2016. High Speed Packet Forwarding Compiled from Protocol Independent Data Plane Specifications. In *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*. ACM, New York, NY, USA, 629–630. https://doi.org/10.1145/2934872.2959080

[8] Andres Nötzli, Jehandad Khan, Andy Fingerhut, Clark Barrett, and Peter Athanas. 2018. P4pktgen: Automated test case generation for p4 programs. In *Proceedings of the Symposium on SDN Research*. ACM, 5.

[9] P Gyanesh Patra, Fabricio Rodriguez, Daniel Feferman, Juan Sebastian Mejia, Christian Esteve Rothenberg, and Gergely Pongracz. 2018. MACSAD. https://github.com/intrig-unicamp/macsad.

[10] P Gyanesh Patra, Christian Esteve Rothenberg, and Gergely Pongrácz. 2016. MACSAD: Multi-Architecture Compiler System for Abstract Dataplanes (Aka Partnering P4 with ODP). In *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*. ACM, New York, NY, USA, 623–624. https://doi.org/10.1145/2934872.2959077

[11] P. G. Patra, C. E. Rothenberg, and G. Pongracz. 2017. MACSAD: High performance dataplane applications on the move. In *2017 IEEE 18th International Conference on High Performance Switching and Routing (HPSR)*. 1–6. https://doi.org/10.1109/HPSR.2017.7968684

[12] Fabricio Rodriguez and P Gyanesh Patra. 2018. BB-Gen: A Packet Crafter for P4 Target Evaluation - Sigcomm Demo. https://www.youtube.com/watch?v=430KjOsgJf0

[13] Fabricio Rodriguez, P Gyanesh Patra, and Christian Esteve Rothenberg. 2018. BB-Gen. https://github.com/intrig-unicamp/bb-gen.

[14] Fabricio Rodriguez Cesen, Pattam Gyanesh Kumar Patra, and Christian Esteve Rothenberg. 2018. BB-Gen: A Packet Crafter for Data Plane Evaluation. In *SBRC 2018 - Salão de Ferramentas*. http://www.sbrc2018.ufscar.br/wp-content/uploads/2018/04/180625_1.pdf