Ralf Kundel, Amr Rizk, Boris Koldehofe. *Microbursts in Software and Hardware-based Traffic Load Generation*
To appear in the Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE, 978-1-7281-4973-820, 2020.

Microbursts in Software and Hardware-based Traffic Load Generation

Ralf Kundel
Multimedia Communications Lab
TU Darmstadt, Germany
ralf.kundel@kom.tu-darmstadt.de

Amr Rizk
Universität Ulm, Germany
amr.rizk@uni-ulm.de

Boris Koldehofe
Multimedia Communications Lab
TU Darmstadt, Germany
boris.koldehofe@kom.tu-darmstadt.de

***Abstract*—Many software based traffic load generators suffer from packet rate variation which is known as rate jitter. In this Demo, we show how this varying rate burstiness can affect the device under test even if the generated average data rate seems constant. To this end, we compare a hardware rate shaping, which is implemented using a programmable P4-switch, and a conventional software load generator and show their impact on a software device under test. The results show, that microbursts within the test load significantly impact the experiment results. Our recommendation is to benchmark the traffic load generator before conducting measurement experiments especially when the device under test is sensitive to microbursts.**

***Index Terms*—Load Generation, Rate Jitter, P4, Microburst**

I. Introduction

Network applications are required to provide high flexibility while keeping costs as low as possible. A widely established approach to provide this flexibility at low cost is to deploy network functionality as software components on commodity compute hardware, named Virtual Network Function (VNF). Besides verifying the functional correctness of a VNF, it is crucial to test the performance characteristics of a VNF to make sure that it efficiently utilizes its resources in a given utilization spectrum. These characteristics include classical QoS-metrics as latency, throughput and packet loss.

Benchmarking VNFs, considered in the following as Device Under Test (DUT), usually requires the use of a load generator which can be implemented in software, e.g. *IPerf3*, in hardware, e.g. *Spirent TestCenter*, or as a combination of both. Previous related works have investigated different performance characteristics of software load generators: For example, the work of Botta *et al.* [1] focuses on the accuracy of software load generators in general. Emmerich *et al.* [2] investigated load generators with hardware assistance of the used Network Interface Card (NIC). Both works concluded, that there are strong differences between different implementations, specially regarding timestamping and rate limiting accuracy.

In this work, we closely examine one metric that is used for classifying the load generator rate limiting accuracy which is denoted as rate jitter. While the average rate of a load generator may well correspond to the configured load rate (rate limiting), the distribution of the sent packets over *windows of the same length*, as shown later in Figure 2, might not be constant. Note that rate limiting can be either traffic shaping or policing. Whereas policing marks/drops packet if the policer rate is exceeded, shaping queues the packets and sends them with a configured rate.

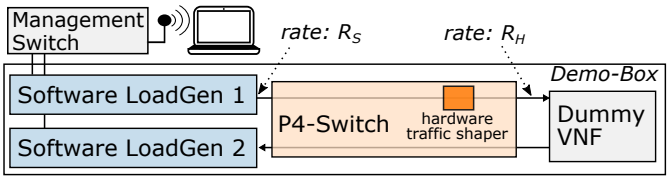


Fig. 1: Demo setup consisting of two commodity servers for sw-load generation and a P4-switch for hardware rate limiting.

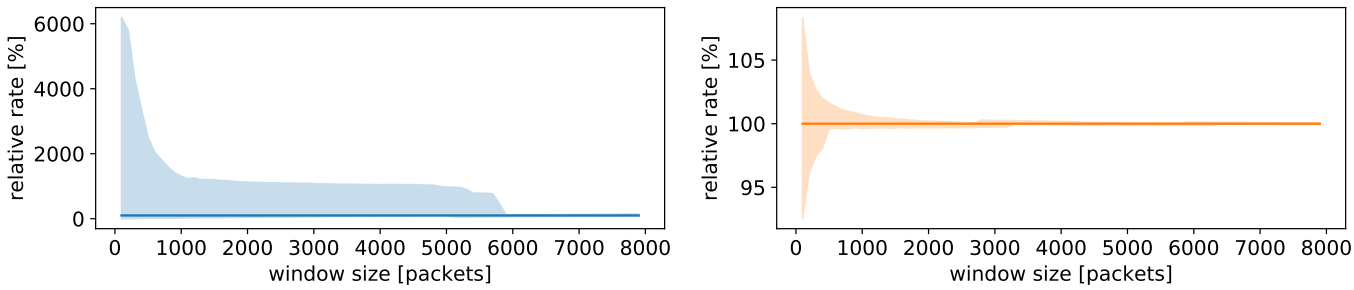
This demo shows (1) how to measure the rate jitter of a test load packet series. By that, we can show the differences in rate jitter between software and hardware load generation. Further, (2) we show the impact of rate jitter on a DUT, a dummy network function implemented in software. All illustrated measurements are performed in a testbed based on the P4STA-framework [3] which allows taking timestamps with nanosecond accuracy at line rate. Note that traffic shaping is performed in hardware whereas the rate limiting mechanism of the software load generator is realized in software. In the following, we focus on the rate jitter accuracy of the *IPerf3* software load generator and a P4-programmable switch as hardware shaper within the P4STA framework.

II. Demo Setup and Goals of this Demo

The introduction of the programming language P4 in 2014 ushered in a new era of reconfigurable networking devices. P4 enables the description of switch pipeline behavior, e.g. custom match tables and corresponding actions. Based on this powerful abstraction and corresponding hardware we provide a load generation and testing framework, called P4STA [3], which combines the flexibility of software traffic load generation and the accuracy of hardware packet timestamping.

The setup of this demo, depicted in Figure 1, consists of two load generation servers running IPerf3 server and client. Both servers are connected using a 40Gbit/s link to a P4-programmable switch, based on a Barefoot Tofino ASIC, which forwards all packets to the corresponding port of the DUT (a dummy VNF). This VNF, realized in software, has a simple functionality, i.e., it forwards all packets received on a certain port back to the P4 switch that finally relays the packets to the second server.

Ralf Kundel, Amr Rizk, Boris Koldehofe. *Microbursts in Software and Hardware-based Traffic Load Generation*
To appear in the Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE, 978-1-7281-4973-820, 2020.



(a) Rate Jitter of IPerf3 load generator ($rate\ R_S$).

(b) Rate Jitter of P4-programmable switch ($rate\ R_H$).

Fig. 2: Maximum, minimum and average measured sending rate, produced by (a) IPerf3 and (b) the P4-switch traffic shaper. The observation window size (x-Axis) is varied from 100 to 8000 packets in 100 packet steps. 100% corresponds to $640Mbit/s$.

The P4 switch takes a timestamp for each packet before entering and after exiting the VNF. Optionally, all packets to the VNF can be rate limited (shaped) inside the switch. Based on this setup, the following two measurement scenarios will be demonstrated:

- **Software load generation and rate limiting:** Load generation is performed using IPerf3 with rate limiting up to 640 Mbit/s link speed. The P4 switch in between is only used for timestamping.
- **Software load generation and hardware rate shaping:** In contrast to the first scenario, the IPerf3 load generation is not rate limited in software anymore. Instead, the P4 switch performs traffic shaping on the output queue with a rate of 640 Mbit/s and $1ms$ queue size.

III. Measurement Results

Measurements are performed with a test duration of $10s$ and IPerf3 as load generator. Timestamps, taken by the P4-switch for each packet, are exported as csv-file. The measurement accuracy and method will not be discussed here for space reasons, however, we refer to the corresponding work [3].

First, we want to focus on the accuracy of the rate generation, i.e., the rate jitter, for constant rate traffic configuration. Figure 2 depicts the observed sending rate for both scenarios. The x-Axis describes the time window which is considered for computing the rate. The horizontal line in the graph represents the average packet rate. For each window size, we compute the maximum, minimum and average observed rate as depicted by the flecked area below and above the average rate. Please note the different y-Axis scales of the two subplots.

The results of the IPerf3 load generator show that within a window size of 100 and 200 packets a rate of up to $38Gbit/s$ (6000%) occur. Even for quite larger window sizes a rate of nearly $10Gbit/s$ (1500%) will be sent by the load generator. Only for window sizes larger than 6000 packets, the measured rate approaches the expected rate which was configured. In contrast to this, the observed rate of the hardware shaper shows only very little deviation from the configured rate. Note that other software load generators might perform much better, e.g., by the use of kernel bypassing or Linux traffic control.

rate limiter	total data	packet loss in DUT	avg. latency of DUT	max. IPDV of DUT
IPerf3:	$803.1MB$	1.93%	$520.30\mu s$	$1750\mu s$
P4-ASIC:	$803.66MB$	0.47%	$425.74\mu s$	$706\mu s$

TABLE I: Observed performance characteristics for the DUT (dummy VNF) with 640 Mbit/s load and 10s test duration.

Secondly, we examine the impact on the performance characteristics of the DUT. Table I depicts the measured performance characteristics for both scenarios. As the test duration and configured rate is equal in both test cases, the total transmitted data is comparable. Indeed, the observed packet loss within the virtual network function (DUT) is four times higher in case of the IPerf3 rate limiting and the measured latency is increased by 25%. The maximum absolute Inter Packet Delay Variation (IPDV), describing the latency difference of two consecutive packets, is higher. Consequential, the dummy VNF performance is rate jitter sensitive.

Despite the fact that jitter is often considered as one cause of poor system performance in networked applications there is little work that deeply investigates the influence and causes of jitter. Together with the companion paper P4STA, we provide a system that enables an empirical investigation of the impact of traffic jitter of devices under test.

To summarize, this demo highlighted (1) the fact that there are strong differences in load generator rate limiting regarding rate jitter and (2) demonstrated the impact on a dummy VNF.

Acknowledgment

This work has been supported by Deutsche Telekom through the Dynamic Networks 8 project, and in parts by the German Research Foundation (DFG) as part of the projects C2 and B4 within the Collaborative Research Center (CRC) 1053 MAKI as well as the DFG grant SPINE.

References

[1] A. Botta, A. Dainotti, and A. Pescapé, "Do you trust your software-based traffic generator?" *IEEE Communications Magazine*, vol. 48, no. 9, pp. 158–165, Sep. 2010.

[2] P. Emmerich, S. Gallenmüller, G. Antichi, A. W. Moore, and G. Carle, "Mind the gap - a comparison of software packet generators," in *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, May 2017, pp. 191–203.

[3] R. Kundel, F. Siegmund, J. Blendin, A. Rizk, and B. Koldehofe, "P4sta: High performance packet timestamping with programmable packet processors," in *2020 IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Apr. 2020.