Estimation of logarithmic and exponential functions entirely in P4-programmable data planes

Damu Ding[1,2], Marco Savi[1] and Domenico Siracusa[1]
[1] Fondazione Bruno Kessler, CREATE-NET Research Center, Trento, Italy
[2] University of Bologna, Bologna, Italy
{ding, m.savi, dsiracusa}@fbk.eu

For the purpose of network management, network operators always need the support of advanced network monitoring functionalities, ranging from heavy-hitter detection [1], heavy change detection [2], traffic entropy estimation [3], distributed denial-of-service (DDoS) detection [4], to super-spreader detection [5]. However, both in SNMP-based [6] legacy networks or in more recent Openflow-based [7] Software-Defined Networks, all these advanced monitoring functionalities are executed in a centralized component (generally known as *monitoring collector*) requiring the transmission, storage and processing of a huge amount of monitoring information from the network data plane. This comes with two well-known drawbacks: (i) a significant communication overhead is generated and (ii) a significant processing capability is needed by the collector, with the risk of affecting performance of network and monitoring operations if involved parties are not well-dimensioned.

Recently, the advent of the so-called *programmable switches*, whose data plane can be programmed by means of the P4 language [8]), allows network operators to overcome the above-mentioned issues. In fact, by properly configuring the switch data plane pipeline, part of the mentioned monitoring functionalities can be implemented directly in the switch data plane and only already-processed information needs to be sent to the collector. However, the high-level and domain-specific P4 language comes with some limitations. For example, it does not support some arithmetic operations such as logarithm, exponential function, division, and any operations on floating numbers. In order to limit the processing time of each incoming packet in the switch, `for` loop is forbidden as well [9]. Additionally, the memory size of a programmable switch is small (for instance, Barefoot Tofino switch [10] has a memory of few tens of MB [11]). Due to these limitations, all unsupported arithmetic operations by the P4 language must be executed at the collector, meaning that most of the application-layer functionalities cannot be implemented entirely in the programmable data plane, thus reducing the benefits of data plane programmability. For instance, network traffic entropy cannot be estimated in the switch data plane, since it requires the computation of logarithms.

The goal of this work is to estimate logarithmic and exponential functions entirely in the switch programmable data plane. As a result, we propose *P4Log* algorithm (for the estimation of logarithms) and *P4Exp* algorithm (for the estimation of exponential functions), by only relying on arithmetic operations supported by the P4 language. We also implemented a prototype of the proposed algorithms in the P4 behavioral model [12] to validate them and to show that they are fully executable in a P4 emulated environment. We also evaluate P4Log and P4Exp by means of simulations to show their sensitivity to different tuning parameters. Results show that our algorithms require no memory occupation in programmable switches to ensure the same relative error of a state-of-the-art solution [13] that leverages on *ternary* and *exact* Match+Action tables with a overall memory consumption of 0.7 KB. Moreover, no communication overhead is generated between the switch and the collector by our algorithms, while the state-of-the-art strategy requires the collector to populate entries for the aforementioned tables.

To summarize, the main contributions of this work are as follows:

- We propose two new algorithms, named *P4Log* and *P4Exp*, used for the estimation of logarithmic and exponential functions by only using arithmetic operations supported by the P4 language.
- We implemented a prototype of the proposed algorithms in the P4 behavioral model [12], proving that they can be entirely executed in the programmable data plane.

References

[1] D Ding, M Savi, G Antichi, and D Siracusa. Incremental deployment of programmable switches for network-wide heavy-hitter detection. In *IEEE Conference on Network Softwarization (NetSoft 2019)*, 2019.

[2] Qun Huang and Patrick PC Lee. Ld-sketch: A distributed sketching design for accurate and scalable anomaly detection in network data streams. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1420–1428. IEEE, 2014.

[3] Ashwin Lall, Vyas Sekar, Mitsunori Ogihara, Jun Xu, and Hui Zhang. Data streaming algorithms for estimating entropy of network traffic. In *ACM SIGMETRICS Performance Evaluation Review*, volume 34, pages 145–156. ACM, 2006.

[4] Ângelo Cardoso Lapolli, Jonatas Adilson Marques, and Luciano Paschoal Gaspary. Offloading real-time ddos attack detection to programmable data planes. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 19–27. IEEE, 2019.

[5] Yang Liu, Wenji Chen, and Yong Guan. Identifying high-cardinality hosts from network-wide traffic measurements. *IEEE Transactions on Dependable and Secure Computing*, 13(5):547–558, 2016.

[6] Niels LM Van Adrichem, Christian Doerr, and Fernando A Kuipers. Opennetmon: Network monitoring in openflow software-defined networks. In *IEEE Network Operations and Management Symposium (NOMS)*, 2014.

[7] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling innovation in campus networks. *ACM Computer Communication Review*, 38(2), 2008.

[8] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95, 2014.

[9] P4-supported arithmetic operations. https://p4.org/p4-spec/p4-14/v1.0.5/tex/p4.pdf.

[10] Barefoot Tofino. https://www.barefootnetworks.com/products/brief-tofino/.

[11] Amedeo Sapio, Ibrahim Abdelaziz, Abdulla Aldilaijan, Marco Canini, and Panos Kalnis. In-network computation is a dumb idea whose time has come. In *ACM Workshop on Hot Topics in Networks (HotNets)*, 2017.

[12] P4 behavioral model. https://github.com/p4lang/behavioral-model.

[13] Naveen Kr Sharma, Antoine Kaufmann, Thomas Anderson, Arvind Krishnamurthy, Jacob Nelson, and Simon Peter. Evaluating the power of flexible packet processing for network resource allocation. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 67–82, 2017.