Leveraging in-network real-value computation for home network device recognition

Matthews Jose, Kahina Lazri, Jérôme François, Olivier Festor

Leveraging in-network real-value computation for home network device recognition

Matthews Jose*†, Kahina Lazri*, Jérôme François† and Olivier Festor†
*Orange, Chatillon, France, email: [matthews.jose,kahina.lazri]@orange.com
†Inria, LORIA, University of Lorraine, email: [jerome.francois,olivier.festor]@inria.fr

***Abstract*—Current generation of switches are highly programmable and able to support stateful. However, these switches cannot perform floating point operations. As a result several network applications have to be run on external servers or middleboxes in the network. We introduce InREC, a system that extends the capabilities of programmable switches to support in-network real-valued operations using the IEEE half-precision floating point representation. Our demo on a Barefoot Tofino switches demonstrates the efficiency of InREC for in-network computation by computing a logistic regression function to classify devices (IoT and laptop) in a simulated home environment.**

I. Introduction

With explosion in the internet of things (IoT) a wide of range of IoT devices are increasing being used in the home environment. Recognizing these devices connected to the home environment have several advantages including in terms of management and security. A lot of works has been done on IoT device detection by analysing the traffic patterns [1] generated and have proven to be quite accurate. But this requires external servers in the home-network to run. We demonstrate the capabilities of InRec [2] to help program a Tofino switch to perform one such detection algorithm completely in-network without the use for external servers.

Advances in programmable networking have provided new programming languages and stateful operations on the dataplane [3] and have resulted mounting interest for deploying basic stateful applications on the dataplane. Operating networks also involve the execution of complex operations such as data analytics, intrusion detection and encryption/decryption, currently available in dedicated appliances [4]. In the context of this demo, we are interested in performing in-network identification of IoT devices. The lack of native primitives for real-value operations on programmable switches have made it difficult to perform necessary computation to analyze traffic pattern with some computation. For instance, logistic regression is a well-known method to do classification. Even if it is not the more advanced ones in comparison to deep learning models, it requires real number operations, which are hard to achieve with a good accuracy in switches.

InREC is an automated pipeline generation tool that generates a switch specific pipeline for a specified real-valued function. To overcome the limited resources on programmable switches, InREC promotes the use of LUTs (Lookup Tables) to support fast match between inputs and outputs of a real-valued function. InREC is fully described in [2] and a first application scenario using logistic regression have been introduced but using arbitrary data. The goal of this demo is demonstrate the capabilities of InREC to do such operations in a realistic context, *i.e.* to perform device identification on the fly in the network.

II. InREC pipeline generation tool

InREC is and automated tool for programmable switches that computes a real-valued function by constructing an equivalent switch pipeline. It relies on a database of elementary operations and their equivalent pipeline representations including LUTs for direct computations. Each compound function is so decomposed into its elementary operations. Dependencies between the elementary operations are represented as a directed acyclic graph. The domain/range nodes (elementary operations/input variables) are modified based on its mathematical properties, input/output to and from the node and the nature of the operations (for example bounded values). The resulting nodes are optimized by performing aggregation where ever possible and intelligently substituting an appropriate pipeline representation from the database for each elementary operation. The resulting graph is then converted to P4 code that can be compiled and offloaded onto a switch.

III. Demonstration Scenario

We demonstrate InREC's capabilities by means of a device recognition use-case in a home environment. The setup (figure 3) consists in a Tofino based programmable connected to an IBM blade server. The server uses Tcpreplay to replay network traffic to the switch for various devices, namely a TP-link wireless camera, a Godrej direct connect bulb and laptop used for browsing the internet. This is a small environment representing a home network including IoT devices and a Laptop. The identification is done on the switch by applying a logistic regression function in-network using the pipeline provided by InREC. The model was trained using the python library scikit-learn and tensorflow. The regression function uses the following 4 attributes to identify a device:

- The size of the first N packets sent and received
- The N - 1 packet inter-arrival times between the first N packets sent and received

Starting with $N = 2$, N increases with the number of devices we are classifying.

In part 1 of the demo, we will introduce our setup and briefly explains the core components of InREC and highlight its application to the logistic regression function. The function

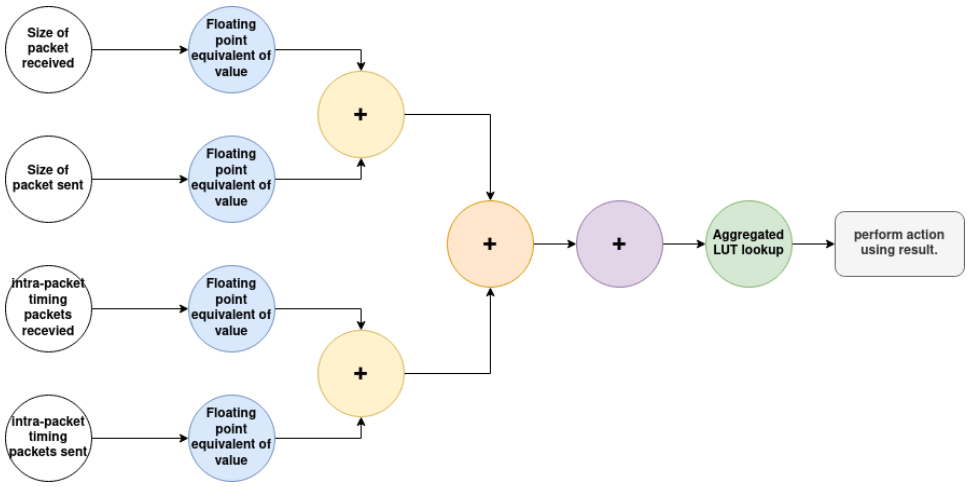

Fig. 1. Logistic regression optimized pipeline. The circles of the same color are computed in parallel and circles in green is the result of aggregation performed by InREC.

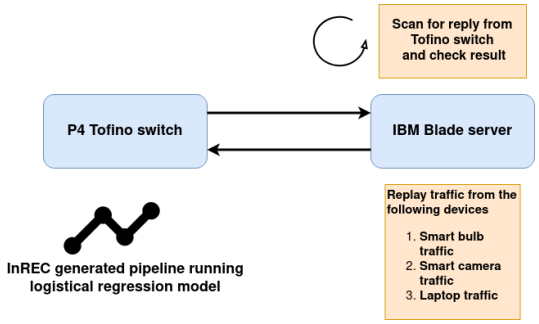

Fig. 2. Demo setup: the IBM blade server replays traffic, the Tofino switch performs the classification and sends back the results to the server

contains 4 addition operations, an exponentiation operations and a division operation.

```
Emulating traffic from smart camera ...
Device is not an IoT device, computed value: 0.811
Device is not an IoT device, computed value: 0.749
Device is not an IoT device: 0.749
Device is a IoT device, computed value: 0.338
```

Fig. 3. Traffic emulation of a smart camera from the server, the first 2 packets in the flow are computed as not being an IoT device until the state variables are updated in subsequent packets in the flow.

Running it through InREC produces the reduced switch pipeline graph (figure 1) that consists of 4 additions and a single lookup aggregating the exponentiation and division operations. The values for the model are extracted from flow packets using the switch metadata fields to track the packet size and the global timer with a register to calculate intra-packet times. All the values are converted from bitfield to half-precision floating point numbers via a lookup table before taken as an input to the generated pipeline. This entire construct is deployed via a controller.

In part 2 of the demo, the traffic is replayed from the server emulating each device sequentially, in bursts and in various combinations. The traffic is set at various speeds and in each scenario the switch is used to identify the device type (IoT device or not IoT device), tag it using a custom header that is prepended and then send it back to the source port. Then, a terminal displays in real-time the identified devices. For comparison, the traffic is mirrored to an external server for performing the same classification task. The latency of each case is shown demonstrating a significant reduction when performing classification in-network on the switch. Furthermore, performing the computation directly on the switch ensures a lower detection time since it doesn't rely on polling mechanisms to sample traffic.

IV. Conclusion

This demo presents a realistic scenario of an application of our InREC [2] prototype. The demo is performed on real P4 supported hardware to highlight its viability in a real environment. Beyond showing an application sceanrio, the aim of the demo is to show the performance of InREC for in-network real-number computations.

References

[1] P. Yadav, A. Feraudo, B. Arief, S. F. Shahandashti, and V. G. Vassilakis, "Position paper: A systematic framework for categorising iot device fingerprinting mechanisms," in *ACM International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, ser. AIChallengeIoT '20. Association for Computing Machinery, 2020.

[2] M. Jose, J. François, and K. Lazri. (2021) InREC: In-network REal Number Computation. Bordeaux, France. [Online]. Available: https://1drv.ms/b/s!AmSZGUizmAdemzayiMklMCAWAWt?e=oWGhQQ

[3] A. Sivaraman, A. Cheung, M. Budiu, C. Kim, M. Alizadeh, H. Balakrishnan, G. Varghese, N. McKeown, and S. Licking, "Packet transactions: High-level programming for line-rate switches," *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016.

[4] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang, "An untold story of middleboxes in cellular networks," *SIGCOMM Comput. Commun. Rev.*, vol. 41, Aug. 2011.