A perspective on P4-based data and control plane modularity for network automation

Ollora Zaballa, Eder; Franco, David; Berger, Michael Stübert; Higuero, Marivi

Published in:
Proceedings of the 3rd P4 Workshop in Europe

Link to article, DOI:
10.1145/3426744.3431330

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Ollora Zaballa, E., Franco, D., Berger, M. S., & Higuero, M. (2020). A perspective on P4-based data and control plane modularity for network automation. In *Proceedings of the 3rd P4 Workshop in Europe* (pp. 59-61). Association for Computing Machinery. https://doi.org/10.1145/3426744.3431330

A perspective on P4-based data and control plane modularity for network automation

Eder Ollora Zaballa
Technical University of Denmark
Fotonik
Lyngby, Denmark
eoza@fotonik.dtu.dk

David Franco
University of the Basque Country
Department of Communications Engineering
Bilbao, Vizcaya, Spain
david.franco@ehu.eus

Michael S. Berger
Technical University of Denmark
Fotonik
Lyngby, Denmark
msbe@fotonik.dtu.dk

Marivi Higuero
University of the Basque Country
Department of Communications Engineering
Bilbao, Vizcaya, Spain
marivi.higuero@ehu.eus

ABSTRACT

When Software-Defined Networks (SDN) gained popularity, different control plane applications could run together to manage vendor-agnostic OpenFlow data planes. With the adoption of P4, developers realized that the P4 logic developed for one particular model and target could not be modularly shared, preventing fast adoption. As a result, a pipeline's specific functionality can not be exported and incorporated into another pipeline without the necessary manual adaptations. As a result, we present the early work for *P4click*, a Next-Generation SDN automation platform. It aims to modularly increment data plane pipeline features, combine P4 entities such as headers, the parser/deparser, build the pipeline execution flow, and the appropriate control plane applications. This paper presents the design adoptions to segregate independent P4 logic features and the architecture for data and control plane modularization.

ACM Reference Format:
Eder Ollora Zaballa, David Franco, Michael S. Berger, and Marivi Higuero. 2020. A perspective on P4-based data and control plane modularity for network automation. In *3rd P4 Workshop in Europe (EuroP4 '20), December 1, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3426744.3431330

1 INTRODUCTION

The use of OpenFlow to manage vendor-agnostic SDN forwarding devices using the same controller has attracted a significant amount of research. Vendors incorporated OpenFlow to their forwarding devices and provided hardware for both academia and industry. The fact that a single control plane application could be used to manage different devices opened the possibility to achieve greater research output in the networking community. The use of control plane forwarding abstractions and vendor drivers that adapt the control to data plane message format allows users to specify the forwarding behavior without considering the OpenFlow pipeline. Following a similar path, P4 expanded the possibilities to provide a custom data plane that provides a broader range of use cases, such as incorporating new protocols on demand. The data plane's flexibility requires new Data-Controller Plane Interface (DCPI) protocols such as the P4Runtime protocol. However, data plane programming still requires new research to find novel approaches to reuse P4 code and find ways to merge pipelines.

In terms of related work, MPvisor [3] is a hypervisor that offers modular programmability. While MPVisor and P4click focus on data plane programmability, P4click covers data and control plane modularity by automating network deployment. Authors from P4I/O [2] developed an Intent Definition Language (IDL) based on Jinja2 templates. P4click supports blueprints and CLI runtime configuration that can build pipelines for different models and targets from scratch. In contrast, P4I/O only supports the addition of P4 entities on specific places of the pipeline.

2 PROBLEM STATEMENT

In SDN controllers like ONOS, it is possible to segregate control plane applications that perform specific functionalities into independent compressed archives. Therefore, each application comprises a different functionality. For instance, applications that handle ARP/NDP (proxying), DHCP (server), or packet in/out for IP traffic run along in the same controller. Still, the applications could also do it along with other third-party applications. *Could we add a similar procedure to P4 pipelines?* It is currently not possible to build a data plane pipeline from scratch by selecting the features to be incorporated. A particular pipeline programmed in P4 with different features such as L2/L3 forwarding, packet I/O, In-band Telemetry (INT) support, or ACL defines all pipeline entities with a strong correlation. It means that the features mentioned earlier can not be modularly added to other pipelines unless a manual pipeline modification is performed. While building a single pipeline with all possible features included, the approach is hardly scalable. In examples such as switch.p4, different elements can be compiled using preprocessor *#ifdef/#ifndef* directives, although the project code readability is poor. Besides, to develop P4 applications, it is

EuroP4 '20, December 1, 2020, Barcelona, Spain

ACM ISBN 978-1-4503-8181-9/20/12.
https://doi.org/10.1145/3426744.3431330

mandatory to create a particular control plane. Finding P4 references to build a new pipeline and adapting a control plane is a time-consuming task. This is also a reason why adopting data and control plane programmability is risky. Data plane features have already been developed, so new adopters could quickly shift networking technologies if these features were centrally available. Therefore, the lack of unified and modular data and control plane management and deployment automation triggered the development of P4click.

P4click [1] manages the control and data plane deployment. Both merging data plane features and also control plane applications to be installed in the controllers. The data plane modules are categorized by model and target, which means that only modules added for a particular model and target can be merged.

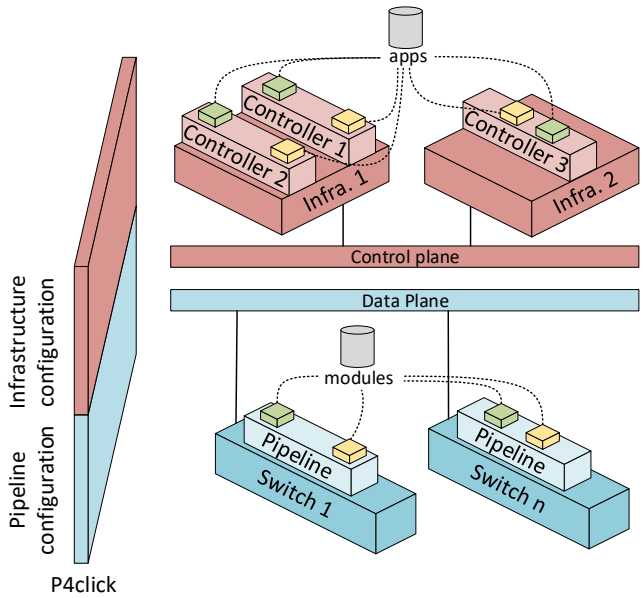

Figure 1: Abstraction of module and app integration within the control and data plane infrastructure.

P4click uses data and control plane repositories containing independent modules, although a particular data plane module will indeed impose the use of a specific control plane application. The control plane repository keeps applications that can be installed in one or more controllers, defined by the infrastructure configuration in P4click (Figure 1). There is a tight coupling between one data plane module and a control plane application to automate network deployment. In this way, data plane module deployment will trigger a compatible control plane that can expose northbound interfaces to third party applications.

3 IMPLEMENTATION

The data plane repository contains modules that represent features. These features include P4 code (actions, tables, etc.) that perform the feature behavior. Additionally, the module comprises headers, parsers, constants, typedefs, etc., as YAML configuration files (see Figure 2). It is also possible to keep data plane modules as a single-featured P4 pipeline. However, generating a modular pipeline would require parsing multiple single-featured pipelines, reaching an intermediate representation, and building a merged pipeline. To ease the process of building a merged pipeline, the logic of the feature is the only part written in P4. Selecting P4 modules for a particular model and target instructs P4click to build the P4 code for a unified pipeline. The platform [1] has been tested to individually gather modules from our repository, parse the configuration, and properly compile the merged pipeline for the BMv2 target (V1Model). However, to address hardware targets, future releases of P4click will have to consider limitations like the available resources. The platform also includes an SSH-based toolset (currently under development) to deploy control and data plane applications.

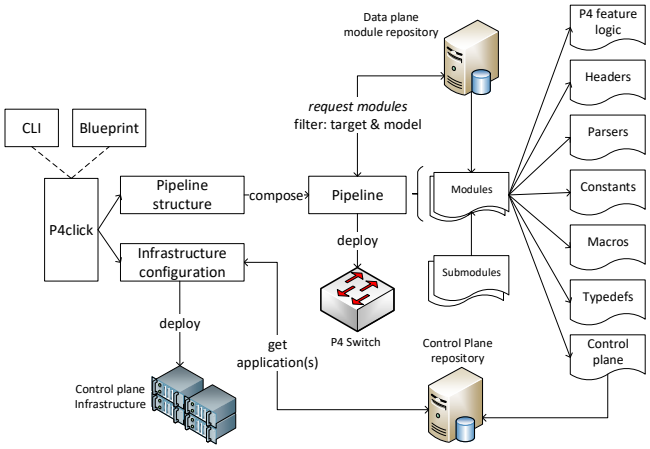

Figure 2: Integration of the modularity engine (P4click) with the control and data plane infrastructure.

4 CONCLUSION

One of the drawbacks of SDN is the Operating Expenditure (Opex) that requires a robust control plane programming. With Next-Generation SDN (NG-SDN), the Opex raises due to the data plane development. In order to increase the adoption of P4 and automate the deployment of data and control plane applications, ground-up modularity in P4 is essential. This paves the way for integrating P4-based devices into software-based network architectures, such as NFV, where OpenFlow is dominant. The seamless adoption of new features into programmable devices is the missing factor to widespread data plane network programmability.

ACKNOWLEDGMENTS

This research has been supported by the 5GCity Project funded by the Spanish Ministry of Economy and Competitiveness under Grant TEC2016-76795-C6-5R and by the Basque Government through the Elkartek Program (ref: KK-2020/00031). Finally, this research has also been supported by the Spanish Ministry of Science, Innovation and Universities within the project TEC2017-87061-C3-2-R (CIENCIA/AEI/FEDER, UE).

REFERENCES

[1] Eder Ollora Zaballa, David Franco. 2020. ederollora/EuroP4_20_p4click: Current status of the p4click as per publication. (Oct. 2020). https://doi.org/10.5281/zenodo.4147320

[2] M. Riftadi and F. Kuipers. 2019. P4I/O: Intent-Based Networking with P4. In *2019 IEEE Conference on Network Softwarization (NetSoft)*. 438–443.

[3] Cheng Zhang, Jun Bi, Yu Zhou, Abdul Basit Dogar, and Jianping Wu. 2017. MPVisor: A Modular Programmable Data Plane Hypervisor. In *Proceedings of the Symposium on SDN Research (SOSR '17)*. Association for Computing Machinery, New York, NY, USA, 179–180. https://doi.org/10.1145/3050220.3060600