Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Thesis B.Sc.

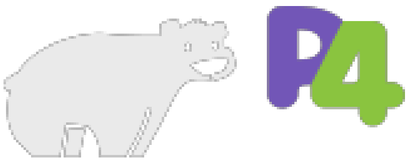
Thesis M.Sc.

IDP, Guided Research

P4 vs. eBPF: Comparison of Expressiveness, Use Cases and Performance

Motivation

P4 is a programming language intended to describe the behavior of packet processing systems. P4 was introduced in 2014 and can be used to define entirely new networks with new protocols which behave differently from the networks we currently use.


P4 programming language

Berkeley Packet Filters (eBPF) allows user-written extensions in the kernel processing path. The successor, extended BPF (eBPF), improves flexibility and is realized via a virtual machine featuring both a just-in-time (JIT) compiler and an interpreter running in the kernel. It executes custom eBPF programs supplied by the user, effectively moving kernel functionality into user space. eBPF has also become a language used to program hardware devices, e.g. the Netronome Agilio SmartNIC.
eBPF and P4 have similar, yet different use cases and functionality. While compilers for P4 to eBPF and eBPF to P4 exist [1], not all features of one language can be represented in the other.
Goal of this thesis is to analyse and evaluate the functionality of these compilers and different software and hardware targets. In particular, how expressive both approaches are and which functionality when translating between them is lost.

Tasks

The thesis requires diving into both P4 and eBPF, as well as different targets (Linux kernel, Smart NIC, ...).

- Get familiar with P4 and eBPF
- Evaluate different P4→eBPF/eBPF→P4 compiler
- Use sample applications (L2/3 forwarder, packet filtering, . . .) to analyse expressiveness, usability and performance

Prior experience with eBPF is recommended.

Sources

- [1] https://github.com/p4lang/p4c/tree/master/backends/ebpf

Contact

Dominik Scholz	scholz@net.in.tum.de
Sebastian Gallenmüller	gallenmu@net.in.tum.de
Henning Stubbe	stubbe@net.in.tum.de