

Thesis
M.Sc.

IDP

Compiling P4 to eBPF: Evaluation of Software-Based Packet Processing

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. Several compilers for P4 to eBPF exist [1].

Goal of this thesis is to analyse the p4c-ubpf compiler in terms of supported functionality and performance. In particular, what P4 programs can be compiled and what software or hardware targets are supported.

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 (p4c-ubpf)
 - Evaluate the p4c-ubpf compiler
 - Compile several programs with increasing complexity of individual P4 components
 - Analyse the performance on at least one software target
 - Model the resource consumption and performance
- Prior experience with eBPF and/or P4 is recommended.

- [1] <https://p4.org/p4/p4c-ubpf.html>

Sources

Contact

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

