Chair of Network Architectures and Services
School of Computation, Information and Technology
Technical University of Munich

Programmable Networks

The P4 Programming Language

Open-source language to specify packet processing logic

- Official website: `https://p4.org`

P4 goals

- Target independence
- Protocol independence
- High Performance
- Reconfigurability

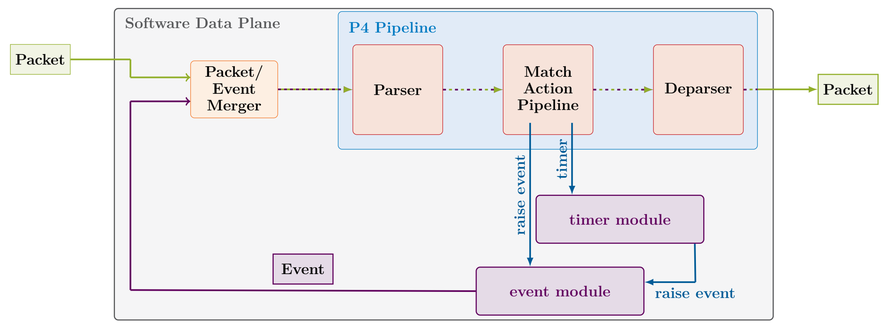
P4 supports multiple platforms

- Software platform
  - *DPDK*, *t4p4s*
- Field-Programmable Gate Array
  - *NetFPGA SUME*
- Network Processing Unit (NPU)
  - *Netronome SmartNIC*
- Application-specific IC (ASIC)
  - *Intel Tofino*

P4 programmable hardware switch based on the Intel Tofino switching ASIC

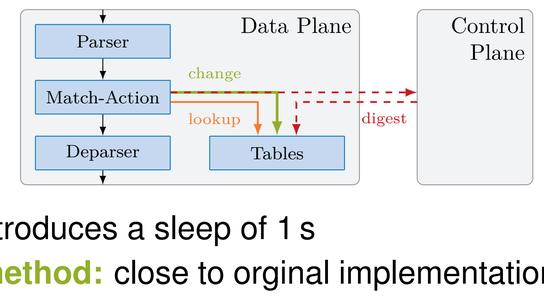
Event-based P4 Architectures

- P4 event architecture for *t4p4s* [4], similar to [2] for SUME NetFPGA
- Two pulling-based queues are processed in every iteration
- DPDK timers are used to trigger timing-based events
- Performance variables to investigate:
  - Timer update frequency
  - Cost of creating and checking events
  - Interference between packets and events



State Management in P4

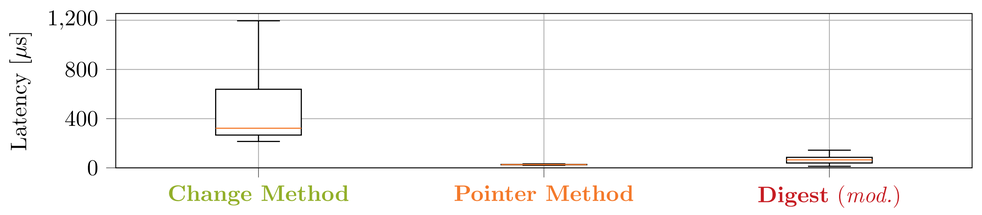
- State Keeping is essential for many applications, possible via:
  - *Registers*: limited functionality, may be fragmented
  - *Tables*: key matching, but only updatable by controller (before)
- Table Update Approaches



  - **Digest:** introduces a sleep of 1 s
  - **Change method:** close to orginal implementation, but avoid detour
  - **Pointer method:** directly changes entries using their pointers
- Integrated updatable tables entries using `@__ref` annotation into *t4p4s* [3]

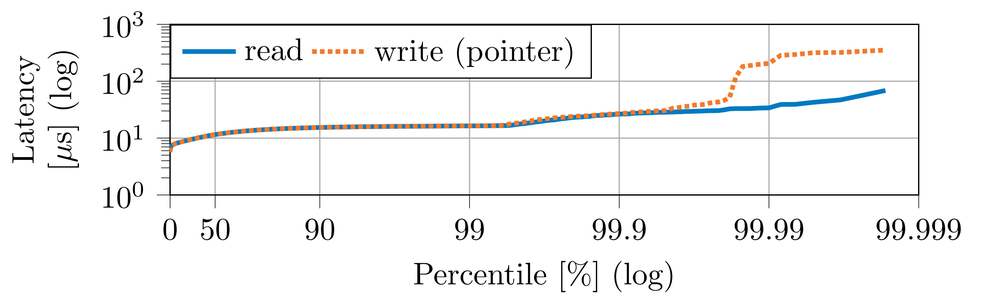
Table Update Performance

- Performance of Update Approaches



Median latencies: 322 µs, 26.5 µs, and 65.3 µs

- Updates are possible at line-rate, nearly as fast as lookups



Dynamic Network Functions - Idea

Idea

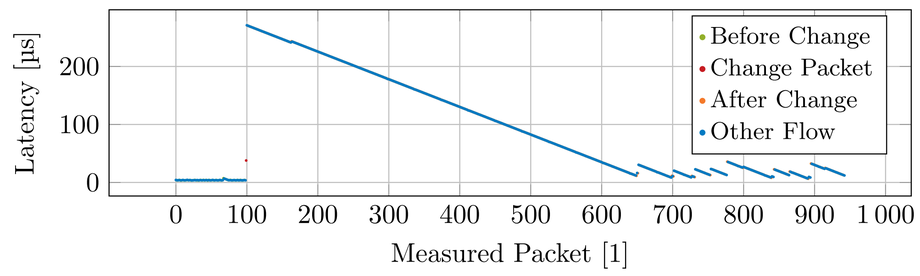
- Trusted parties can deliver code for network functions
- These will be applied to all further traffic of the flow on-the-fly
- E.g., for In-Network-Computing

Libmoon/Lua Prototype

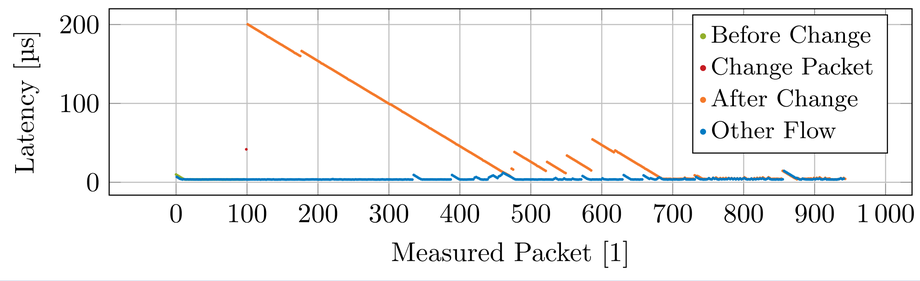
- Libmoon [1] is used for implementing a prototype
- LuaJIT uses just-in-time compilation for a high performance
- Hashtable for function matching
- `loadstring()` to compile new functions

Dynamic Network Functions - Evaluation

- Using one thread/task



- Using two threads/tasks



[1] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle. MoonGen: A Scriptable High-Speed Packet Generator. In *Internet Measurement Conference 2015 (IMC'15)*, Tokyo, Japan, Oct. 2015.
[2] S. Ibanez, G. Antichi, G. Brebner, and N. McKeown. Event-driven packet processing. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, HotNets '19, pages 133–140, New York, NY, USA, 2019. Association for Computing Machinery.
[3] M. Simon, H. Stubbe, D. Scholz, S. Gallenmüller, and G. Carle. High-Performance Match-Action Table Updates from within Programmable Software Data Planes. In *4th P4 Workshop in Europe (EUROP4)*, Virtual Event, Dec. 2021.
[4] P. Vörös, D. Horpacsi, R. Kitlei, D. Lesko, M. Tejfel, and S. Laki. T4P4S: A Target-independent Compiler for Protocol-independent Packet Processors. 06 2018.

Manuel Simon *simonm@net.in.tum.de*