P4 (and SDN)

Alberto Martínez Alba, Nemanja Đerić, Dominik Scholz

May 24, 2017

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

Problem with OpenFlow

• matching only on 40ish predefined fields
• new version, more fields
  → long development/update cycles
  → new/uncommon protocols not supported

Programming protocol-independent packet processing

• dumb hardware → program the header and processing
• highly flexible
• easily deployable
Targets

Software

- Open vSwitch (+DPDK): PISCES
- DPDK: ELTE@P4

Hardware

- NetFPGA
- Netronome Alterio 10GbE NIC
- Barefoot Capilano (coming soon... )
Sample Use Case

Complex stack, e.g. VXLAN encapsulation

- normal switch: not really capable of inspecting e.g. payload

With P4

- Define headers (and stack)
- Parse packets on ingress
- Process (payload, headers, ...)
- Deparse on egress
  → add/remove headers, modify fields
Other approaches?

Other languages?
- yes, but not as sophisticated
- not as much pushed (P4: Barefoot Networks)
- other goals

OpenWrt?
- replace firmware with Linux distribution
- OS in itself, more freedom
- different goal/use case?!

P4
- defined runtime environment
- defined program structure
What is still required?

Scheduling

- time sensitive networks
- quality of service
- proposal (see next slide)

Cryptography

- would complicate/bloat hardware architecture
- too expensive (price)
- not required for what P4 should be used (switch/router)
Interesting papers

  - extension to P4?
  - further extensions in upcoming years (P6)?

  - how to evaluate performance of targets
  - mainly delay and throughput
  - what else? (profiling)

  - based on Push-In First-Out (PIFO) queues