cesnet

P4-TO-VHDL

Pavel Benáček

CESNET

20.8.2018
P4 Tutorial
SIGCOMM Budapest

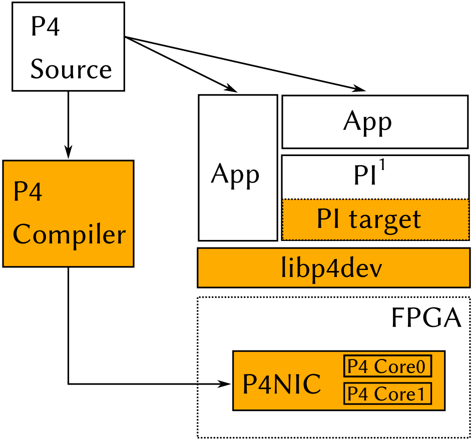
cesnet

Solution Overview

Solution Overview

- Project started as my Ph.D. work
  - Czech Technical University in Prague, Faculty of Information Technology
  - CESNET (**C**zech **E**ducational and **S**cienific **NET**work)
- Ecosystem includes:
  1. Compiler from P4 to VHDL
  2. General design of P4 ready NIC (P4NIC)
  3. Library for device configuration (lib4dev)
  4. Support of the pipeline in PI library (P4Runtime)
- We provide all parts required for rapid development
- The solution is capable to reach high-speeds

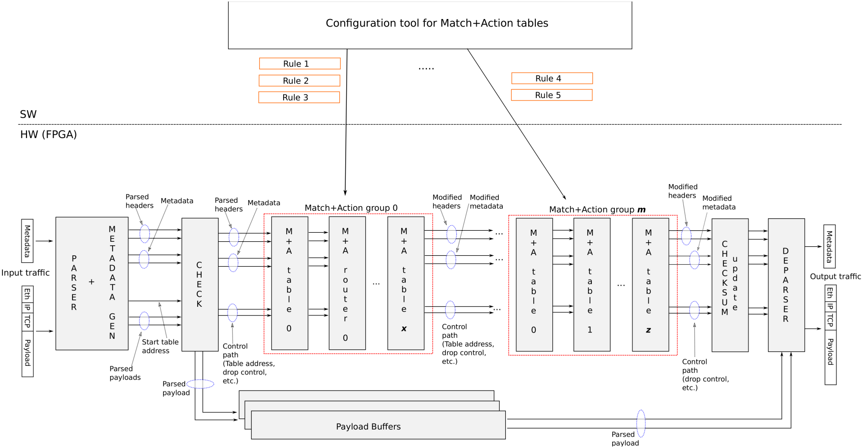
Solution Overview



1) p4lang/PI Project https://github.com/p4lang/PI

Architecture of P4 Pipeline

Architecture



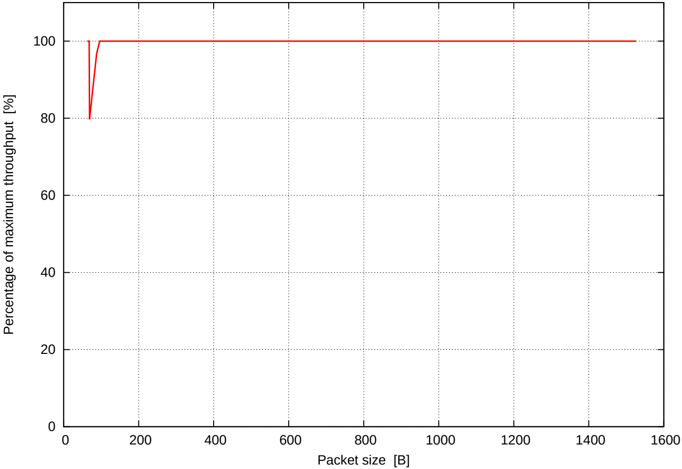
Size and Throughput I

1. **IPv4 Filter** — filtering of packets is based (IPv4 address), the non-IP traffic is dropped
2. **IPv4+IPv6 Filter** — extends the IPv4 Filter with support of IPv6 protocol
3. **Full Filter** — extends the IPv4+IPv6 Filter with support of tagging (VLAN and MPLS)

Project	P4 lines	Time [s]	Generated lines	Total lines
IPv4 Filter	91	1.574	6283	24791
IPv4+IPv6 Filter	129	1.818	9888	28396
Full Filter	212	1.929	13824	32332

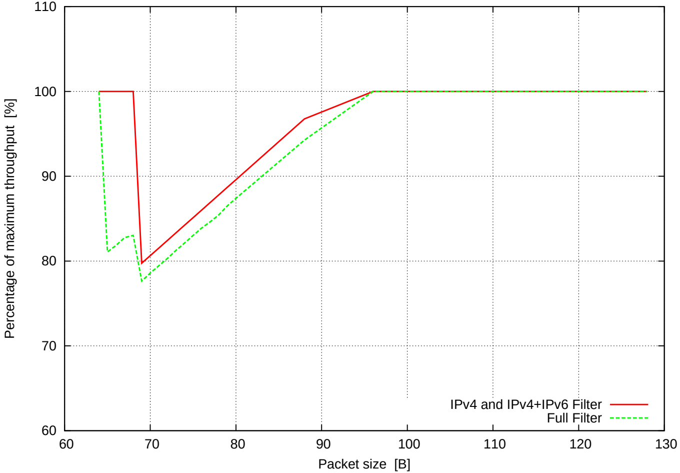
- **Generated lines** — expresses the effort of the generator
- **Total Lines** — the sum of generated lines and lines of library source code (FIFO, TCAM, and so on)

Size and Throughput II



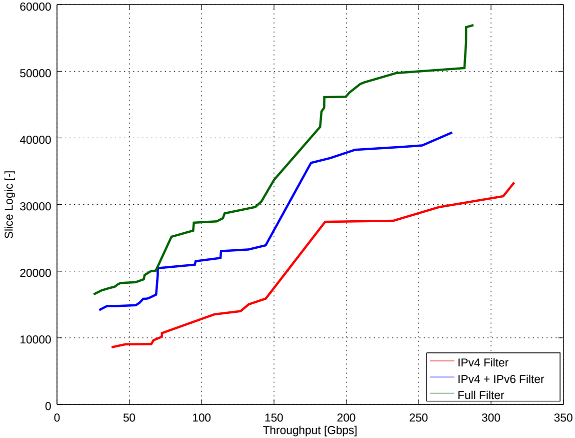
- Throughput of test line is 100Gbps

Size and Throughput III

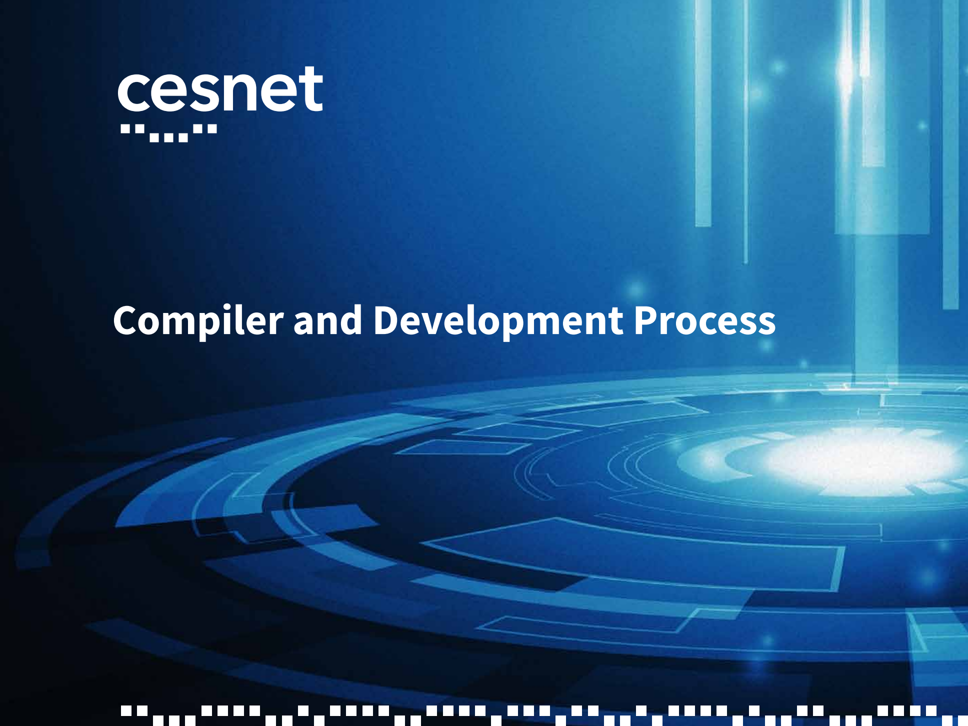


- Inefficiency is caused by the *Deparser* block

Size and Throughput IV



- Slice Logic = LUTs + Registers

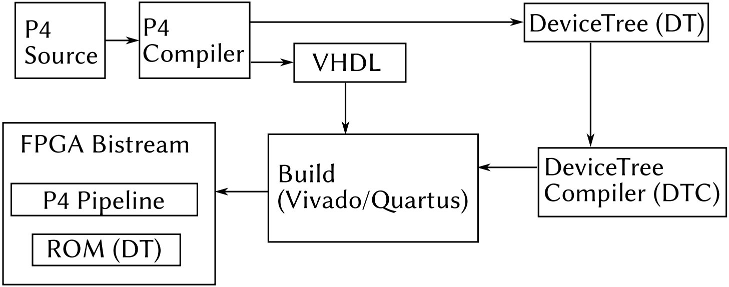


Compiler and Development Process

Compiler

- Currently supports $P4_{14}$
  - P4-HLIR, implemented in Python
  - Version with P4C frontend is under development
- Generated VHDL code is platform independent
- Supported FPGAs
  - Xilinx – Virtex 7, Virtex UltraScale+
  - Intel
- P4 $\xrightarrow{P4-to-VHDL}$ VHDL $\xrightarrow{Synthesis}$ bitstream
- No additional tool is needed
  - *p4fpga* – requires Bluespec compiler

Translation Process



- Each generated pipeline has a different structure
    - Keys structure, set of actions, number of tables
- DT = DeviceTree
    - Data structure for hardware description
    - `https://www.devicetree.org/`

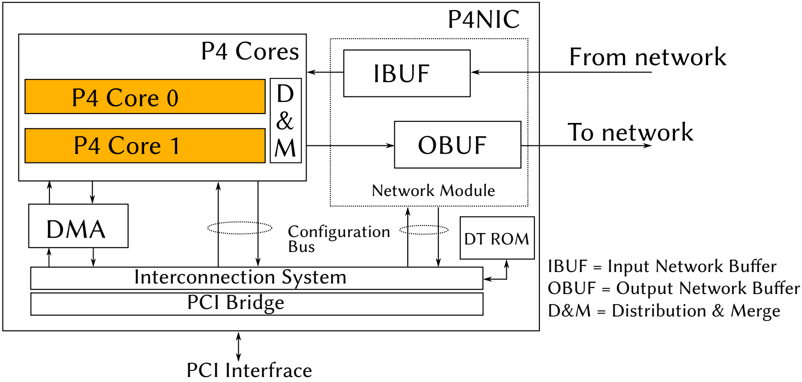
libp4dev

- Fast HW without comfort control == useless
- Multiplatform library in C language
  - Configuration M+A Tables with rules
  - Device control (Enable/Disable, Reset, etc.)
- No recompilation is needed for new P4 program
- All necessary data are stored in DT on a device
  - Structure of the P4 pipeline
  - Action name → opcode
  - Table name → address space offset
  - Structure of table search key
  - Structure of the pipeline's address space
- Allows fast and easy development

P4NIC I

- P4NIC = NIC with P4 generated cores
  - Basic infrastructure for transfers from/to SW to Ethernet/DMA
  - Doesn’t need to be modified because interface is sustained
  - Used for fast development of P4 accelerated applications
- Supported cards:
  - COMBO-100G (Virtex 7, 100 Gbps)
  - COMBO-100G2Q (Virtex 7, 100 Gbps)
  - COMBO-200G2QL(Virtex US+, 2 × 100 Gbps)

P4NIC II



IBUF = Input Network Buffer
OBUF = Output Network Buffer
D&M = Distribution & Merge

- Coloured parts are generated by the compiler
- White parts stay unchanged

Use Cases

Use Cases I

INT Demo

- P4 Worskhop 2017, CA, USA
- INT Sink at 100 Gbps using a single FPGA
- GUI - Flowmon Collector
  - Professional tool for flow monitoring
  - Extended to display delays of switches in INT network

Use Cases II

Open vSwitch (OVS) Acceleration

- Demonstrated on several conferences
  - P4 Workshop 2018,CA, USA
  - TNC18, Trondheim, Norway
- Partial OVS acceleration
  - Input traffic is parsed and assigned with Mark ID
  - Mark ID assignment is configured via DPDK's RTE Flow
- Accelerated version - 2 × higher packet rate

Use Cases III

Service Function Chaining

- P4 Workshop 2018, CA, USA
- Realization of NFV (Network Function Virtualization)
- Implemented using the IPv6 Segment routing
- Capable to process data at 100 Gbps

IPv7 Demo

- FPL 2017, Ghent, Belgium
- Accelerated decapsulation from non-existing IP protocol
- Capable to process data at 100 Gbps

Demo

- Demonstration of the project translation (**live**)
- Demonstration of OVS Acceleration (**video**)

Future Work and Conclusion

Conclusion

- Full P4 environment for rapid development
  - P4 compiler, configuratin layer, initial FPGA design (P4NIC), build system
- Capable to reach high-speeds (100 Gbps)
- More information about generated pipeline
  - Ph.D. Thesis: *Generation of High-Speed Network Device from High-Level Description*
- Future Work:
  - Remove all bottlenecks from the design
  - Full support of $P4_{16}$
  - Implementation of verification environment for generated RTLs

cesnet

Thank you for your attention.

www.liberouter.org