Automatic Generation of 100 Gbps Packet Parsers from P4 Description

Pavel Benáček [1] Viktor Puš [1] Hana Kubátová [2]

[1]Liberouter
CESNET

[2]Faculty of Information Technology
Czech Technical University in Prague

H^{2}RC Workshop, November 2015

Introduction

- OpenFlow is a very popular protocol for realization of SDN (**S**oftware **D**efined **N**etworking)
- Pros and Cons:
  - \+ Allows us to decouple control and data plane
  - \+ Provides a way to fill match tables of switches at runtime
  - \- Not possible to change set of supported protocols (parsers are fixed)
- Researchers are looking for a solution of this disadvantage
  - P4 language is the next step in the SDN concept realization
- Our paper introduces a generator which transforms P4 source to the FPGA parser's architecture

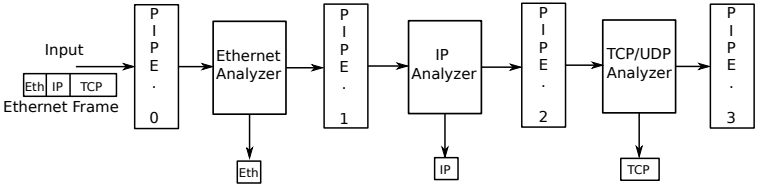
CESNET

P4

- **P**rogramming **P**rotocol-independent **P**acket **P**rocessors
- Language with relatively simple syntax
- Provides a way to define packet processing functionality of network devices
- Defines following aspects of packet processing:
  1. **Header Formats** } packet parser definition
  2. **Packet Parser** }
  3. Table Specification } match to action mapping
  4. Control Program }
  5. Action Specification

Parser structure

- Two basic types of modules are connected to pipeline
  1. Protocol Analyzer - understands one protocol from the protocol stack
     - ★ Data extraction
     - ★ Computes the next protocol type in the stack
     - ★ Computes the next protocol starting offset
  2. Pipeline - used to tune the final frequency and latency
- Unified interface for easy connection of modules



Transformation Algorithm

- Structure of protocol analyzer is generated from the P4's **Header Format**
- Parser's structure is inferred from P4's **Packet Parser**

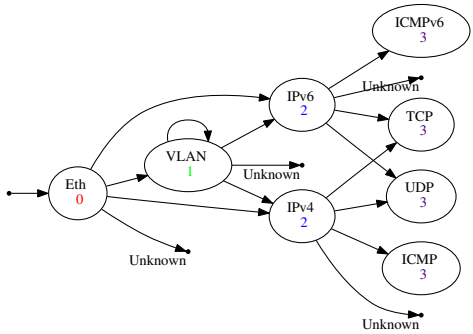
Basic idea of parser's structure identification

We have to identify the latest usage of the protocol in the parser.

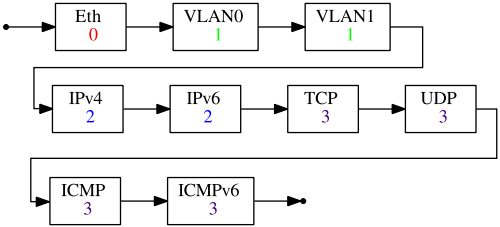
- Algorithm for the identification of the latest usage: DFS (Depth First Search)
- This can be done in **P**arser **G**raph **R**epresentation (PGR)
  - Directed acyclic graph
  - Represents relations between protocols
  - Created from P4 description

CESNET

Transformation Algorithm - Example



Logical model (PGR)



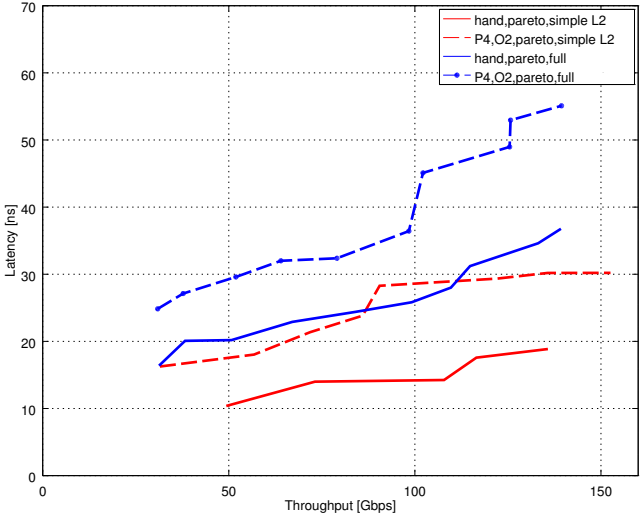
Physical model

Results

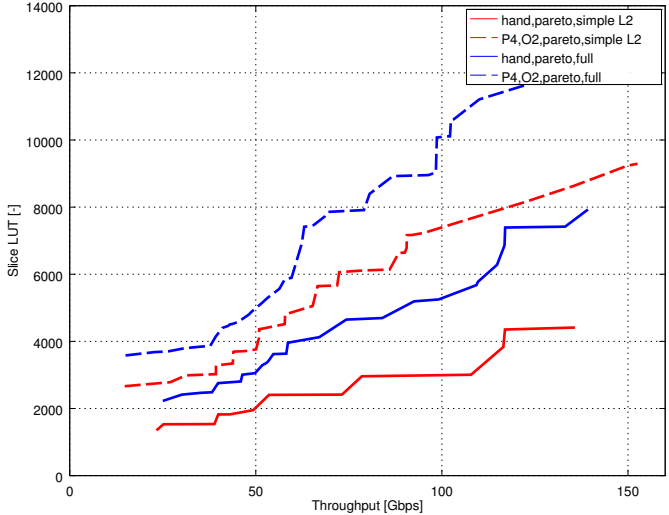
- We have tested two protocol stacks:
  - **Simple L2** - Ethernet, IPv4/IPv6, TCP/UDP, ICMP/ICMPv6
  - **Full** - Ethernet, 2x VLAN, 2x MPLS, IPv4/IPv6, TCP/UDP, ICMP/ICMPv6
- Possible optimizations:
  - **Automatic** - Optimizations of internal parser's structure
    - O1 - Offset bus optimization
    - O2 - Multiplexer optimization
  - **Manual** - Tweaked P4 program (O3)
- We performed synthesis for our implementation platform:
  - Equipped with Xilinx Virtex 7 FPGA
  - Suports 100 Gbps

CESNET

Results - Latency



Results - Resources



Conclusion

- We implemented and evaluated our generator of parsers
- Parsers are capable to process 100 Gbps
- From presented work we can infer following important results:
  1. Ability to generate parsers with equal functionality in shorter time
  2. Generated parsers aren't significantly worse than hand optimized versions created by a professional with many years of experience in HDL coding
- Future work:
  - Deparser - construction of packets
  - Match+Action tables - general processing of extracted data

CESNET

Thank you for your attention

Pavel Benáček

www.liberouter.org

@liberouter

Visit our business partner in booth #3011

CESNET

Protol Analyzer