

# Paper Summary

## HA 2.a

Damodar Sahasrabudhe

School of Computing, University of Utah

September 2, 2015

### 1 Title of paper: P4: Programming Protocol-Independent Packet Processors

Paper discussed in this summary is "P4: Programming Protocol-Independent Packet Processors"[1].

#### 1.1 First pass information

1. *Category:* Paper suggests a new Programming language and presents examples.
2. *Context:* Paper is in area of SDN. Refers to multiple other papers such as - Kangaroo [2], Portland[2], NOSIX[2] etc.
3. *Assumptions:*
  - Actions will be built with protocol independent elements.
  - For sake of explanation, paper assumes there will be no packet processing during configuration.
  - Underlying switch has capability to parse the header.
  - Compilers targeting every type of underlying hardware will be easily available.
4. *Contributions:* Paper suggests building of an programming language to hide underlying hardware and protocols and provide simple and uni-

form interface to programmers, where controllers could be easily programmed.

#### 1.2 Second pass information

*Summary:*

- Introduction: Over the period of time open flow headers are becoming more and more complex. It shows need to have flexible headers and need to add / remove header elements dynamically. This in turn needs more abstract programming interface which will be platform and protocol independent.
- Abstract Forwarding Model: Improving further upon concepts of OpenFlow, switch could be have inbuilt parser tables (to parse header fields), parallel match-action tables (against serialized operations of OpenFlow tables) and action set.
- Programming Language and Examples: Parallel operations of flow tables requires compiler should prepare Table Dependency Graph (TDG) and then map TDG to specific switch. P4 introduces new constructs for - header, parser, table, action and control. Each of them provides consistent way of configuring controller and/or switches irrespective of protocol or underlying hardware.

### 1.3 Third pass information

- *Strengths:*

- The idea, if implemented effectively, will make life of administrators easier.
- It will be easy to use P4 and write an application to gather statistics and provide Bird’s eye view.
- Snapshots of code makes it easier to understand the paper.
- Parallel processing of packets will give better performance than OpenFlow switch.

- *Weaknesses:*

- Packet tampering may cross boundaries of network protocol layers.
- Seems over complex to be implemented and tested.

- *Questions:* Is it practically possible to have compilers targeting so many protocols and underlying hardware?

- *Interesting citations:* Scalable, high performance ethernet forwarding with CuckooSwitch

- *Possible improvements:*

- No need of multiple match action tables. Single entry can use AND / OR conditions to check complex conditions.

- *Future work:* There should be some governing body to control custom protocols headers, otherwise there will be explosion of such private changes.

[3] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, PortLand: A scalable fault-tolerant layer 2 data center network fabric, in ACM SIGCOMM, pp. 3950, Aug. 2009.

[4] M. Raju, A. Wundsam, and M. Yu, NOSIX: A lightweight portability layer for the SDN OS, ACM SIGCOMM Computer Communications Review, 2014

## References

- [1] P4: Programming Protocol-Independent Packet Processors
- [2] C. Kozanitis, J. Huber, S. Singh, and G. Varghese, Leaping multiple headers in a single bound: Wire-speed parsing using the Kangaroo system, in IEEE INFOCOM, pp. 830838, 2010