CS 6480: Paper reading summary
HA 6.a

José Monterroso

School of Computing, University of Utah

September 14, 2020

1 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:*

   This paper is a description of a research prototype. The research prototype being discussed is P4.

2. *Context:*

   The technical area of this paper relates to software defined networking (SDN). Specifically how P4 can improve OpenFlow. This is the first paper we have read that relates specifically to SDN.

3. *Assumptions:*

   A small assumption the authors make is that they believe new header values that OpenFlow will need to match on will be constantly increasing. I agree, not only do the authors provided a table to back this assumption but it's a bit obvious that as time goes by new networking protocols are created and OpenFlow needs to adapt. Furthermore, the authors assume that configuration languages will lead to future switches that provide greater flexibility and unlock the potential of SDN. I'm not too familiar with configuration languages but their assumption seems logical to me, and could possibly be accurate.

4. *Contributions:*

   They do not explicitly state any claims or contributions but P4 claims to serve as a general interface between the controller and the switches. They appear to claim that this is a stepping stone for OpenFlow 2.0.

5. *Clarity:*

   From what I have read I do believe this paper is well written. I actually like the writing style and font a lot.

1.2 Second pass information

- *Summary:*

  P4 is a high-level language for programming protocol-independent packet processors. P4 aims to be a stepping stone for OpenFlow 2.0. Through the use of P4, programmers should be able to change the way switches process packets once they are deployed, switches should not be tied to any specific networking protocols, and programmers should be able to describe packet-processing functionality independently of the specifics of the underlying hardware. P4 follows an abstract forwarding model that supports a programmable parser to all new headers to be defined, supports match+action stages to be in parallel, and the actions are composed from protocol-independent primitives supported by the switch. Section 4 gets into a little of the behind the scenes details of the P4 language. Specifically how the the P4 language uses headers, parsers, tables, actions, and control programs to get everything to work and flow correctly. Lastly, for a network to run the P4 program we need a compiler to allocate the target's resources and generate appropriate configuration for the device.

1.3 Third pass information

- *Strengths:*

  I really liked how the Abstract stated their goals, and what P4 is and aims to be. Even though the paper is a bit complex with regards to the actually content, the paper is easy to follow and read. I also though the 'mTag' example that was used throughout section 4 was a strength because they applied their work to an example and didn't simply state all of the P4 information in one giant section that you had to read and decipher. Overall I really liked the readability and layout of the paper.

- *Weaknesses:*

  The Related Works has good content, but it is located in a weird place at the end of the introduction. I also though that section 5, the compiling section, lacked depth. I mean section 5 did offer good content but it wasn't up to par like the previous sections. I felt that once they define P4 they stopped caring about the rest, even though a specific complier is needed to be able to use P4.

- *Questions:*

  I'm a bit confused on the differences between ASIC switches and normal switches. Also what are ToR switches.

- *Interesting citations:*

  Most if not all the citations of this paper are fairly old. However, I found Click modular [2] to be a potential interesting read. The authors used Click as a base for P4 and added the ability to mirror the parse match-action pipelines in dedicated hardware.

- *Possible improvements:*

  I think they could have added more to the related works section to make it into its own separate section rather than being a small paragraph in the Introduction. Also their compiler section lacked content. I think they either need to buckle down and create a compiler that matches the P4 requirements or do a brief statement of what that compiler needs to do in order for P4 to work. I think what the authors gave us was a middle ground that lacked depth.

- *Future work:*

  Some future work that I can think about is using OpenFlow for SDN approaches. I'm not sure if P4 is still alive today, but I'd like to play around with it. It was mentioned that there is a need for a specific complier to run P4, I would like to do some research in this area. Furthermore, Click seems to be the inspiration for P4 so I would be interested in doing some future work related to Click.

References

[1] Bosshart, P., Daly, D. Gibb, G., Izzard, M., Mckeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and Walker, D. P4: programming protocol-independent packet proocessors. *ACM SIGCOM Computer Communication Review* (July 2014).

[2] Kohler, E., Morris, R., Chen, B., Jannotti, J., and Kaashoek, M. The click modular router. *ACM Transactions on Computer Systems* (Aug. 2000).