Whippersnapper: A P4 Language Benchmark Suite

Huynh Tu Dang[*], Han Wang[†], Theo Jepsen[*], Gordon Brebner[§], Changhoon Kim[°], Jennifer Rexford[¶], Robert Soulé[*], Hakim Weatherspoon[†]

[*]**Università della Svizzera italiana**, [†]Cornell University,
[§]Xilinx Inc., [°]Barefoot Networks, [¶]Princeton University

Outline

- Introduction
- Platform-Independent Suite
- Platform-Specific Suite
- Example Use Cases
- Conclusion

What Is P4?

- A programming language allowing the specification of packet processing logic

```
/* Header definition */

header_type ethernet_t {
   fields {
      dstAddr : 48;
      srcAddr : 48;
      etherType : 16;
   }
}
```

```
/* Parser */

parser parse_ethernet {
   extract(ethernet);
   return select(latest.etherType) {
      0x800 : parse_ipv4;
      0x806 : parse_arp;
      default : ingress;
   }
}
```

What Is P4?

- Based on a Match-Action Model

- Allows the automatic generation of APIs to configure the packet processing tables

```
action _drop()
{
  drop();
}

action next_hop(arg1, arg2, …)
{
  // instructions
}
```

```
table forwarding_tbl {
    read {
        ipv4.dstAddr : lpm;
    }
    actions {
        next_hop;
        _drop;
    }
 }
```

What Is P4?

- There are many open source and proprietary compilers:
  - Open source: BMV2, PISCES, P4FPGA, P4@ELTE
  - Proprietary: Xilinx SDNet, Netronome NFP, Barefoot Tofino

There is a need for a P4 benchmark suite

P4 Benchmark Suite Challenges

- Diverse target platforms
  - Diverse metrics
  - Difficult to collect target-specific metrics
- Representative applications and workloads

Whippersnapper:
P4 Benchmark Suite

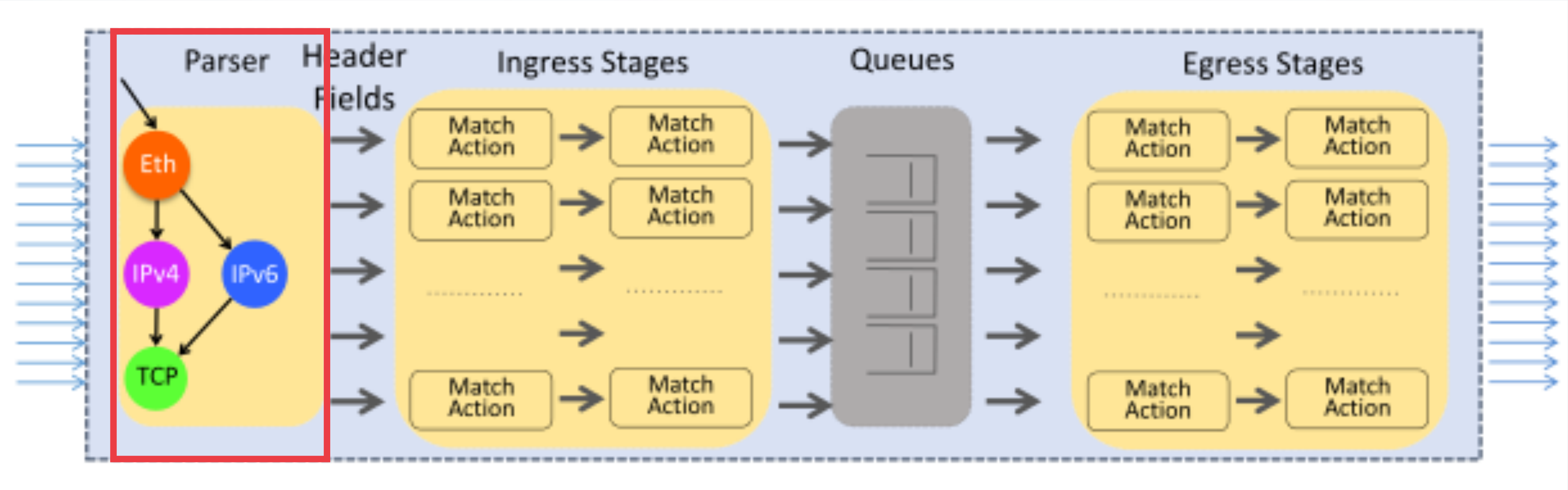
- Copes with target heterogeneity
  - Platform-Independent benchmark
  - Platform-Specific benchmark
  - Black-box benchmarking methodology
- Consists of artificial programs and workloads

Platform-Independent Benchmark

Feature	Parameter
Parsing	#Packet headers #Packet fields #Branches in parse graph
Processing	#Tables (no dependencies) Depth of pipeline Checksum on / off
State Accesses	#Writes to same/different registers #Reads to same/different registers
Packet Modification	#Header adds #Header removes

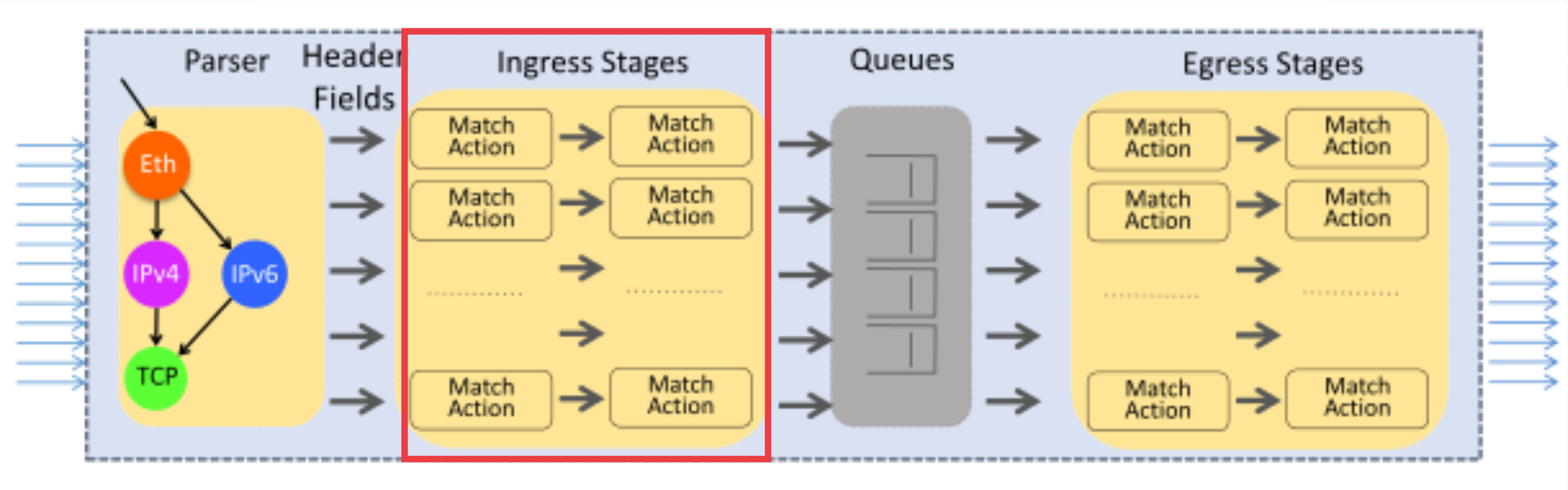
Parser: Topological Ordering

- Vertices correspond to parse states
- Edges correspond to transitions between states



Processing: Match-Action Tables

- Number of tables
- Depth of pipeline (Tables having dependencies)
- Checksum computation



Packet Operations

- State Access
  - Number of reads/writes to stateful memory (register)
  - P4 does not specify a concurrency model for state access
- Packet Modification
  - Number of headers to be added
  - Number of headers to be removed

Platform-Specific Benchmark

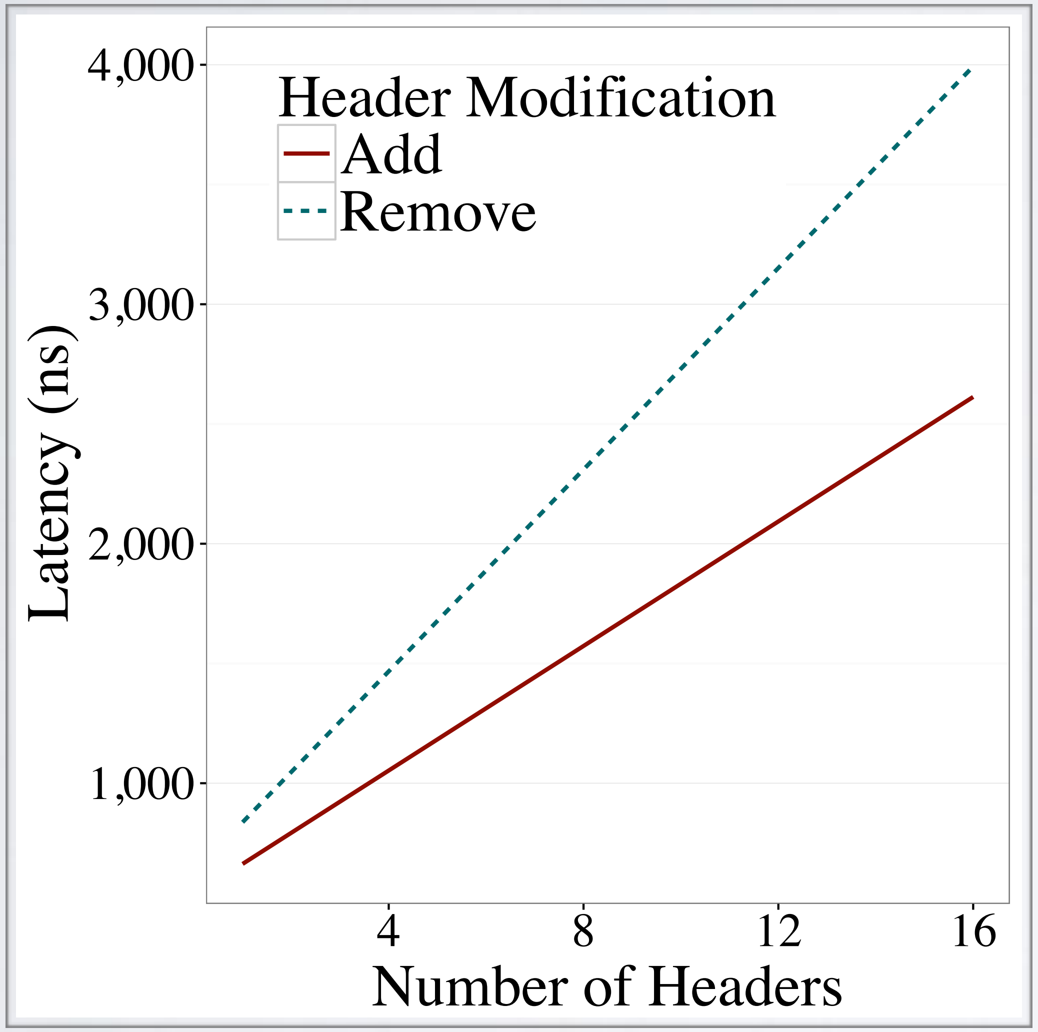
Feature	Metric	Parameter
CPU and NPU	Throughput Latency	Flow set Size of flows
FPGAs	Area Timing Resources	#Tables Size of tables
ASICs	Area Timing Resources Power	#Tables Size of tables #Depth of dependencies

Example Use Cases

Experiments with four P4 targets:

- P4c & Behavioral Model Switch (Bmv2)
- PISCES: customized OVS to support P4
- P4FPGA: compiled P4 for FPGAs (experimented with NetFPGA SUME board)
- Xilinx SDNet: compiled P4 for FPGAs (experimented with Virtex UltraScale+ XCVU13P board)

Packet Modification



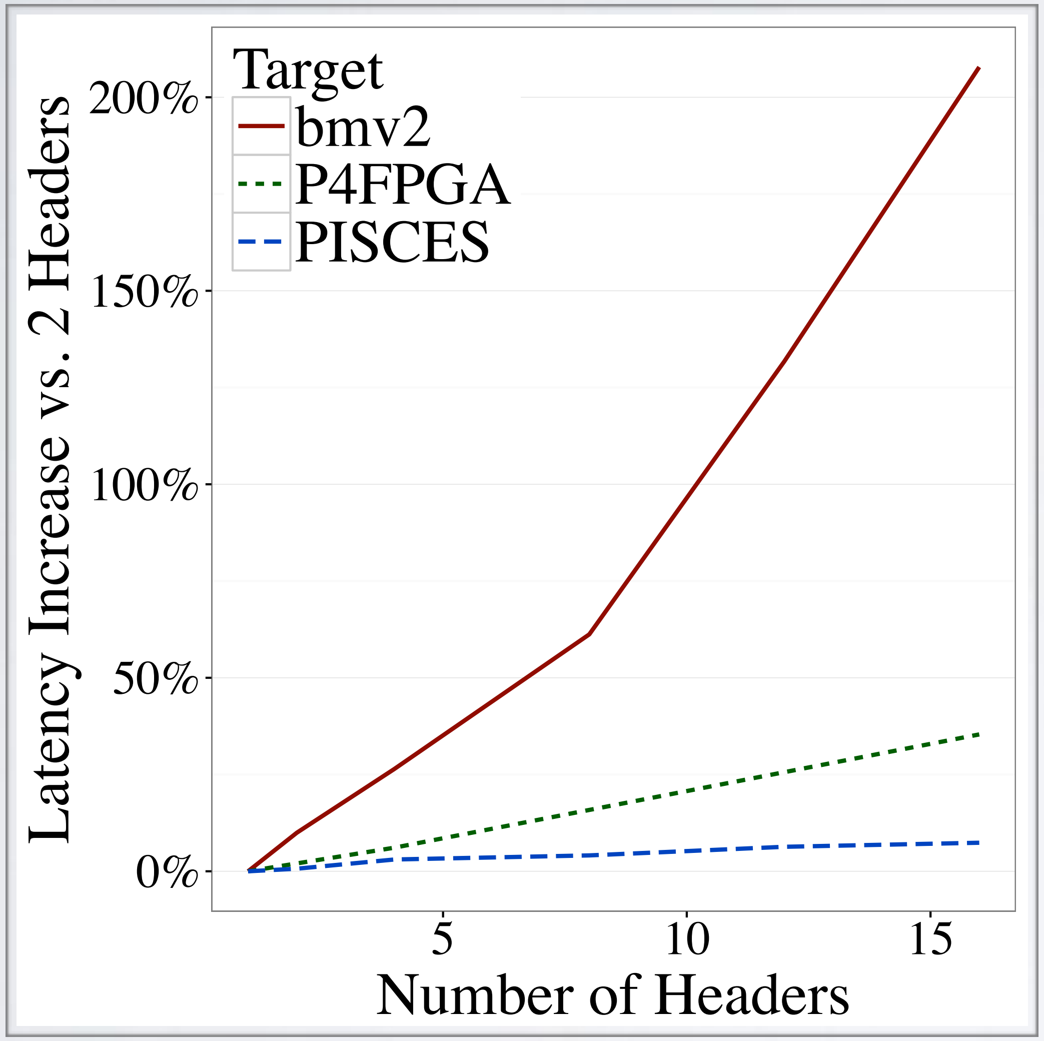
- Experimented with $P4_{14}$-to-PX Xilinx SDNet on XCVU13P board

- Each header removal adds one stage

- All header additions results in one stage

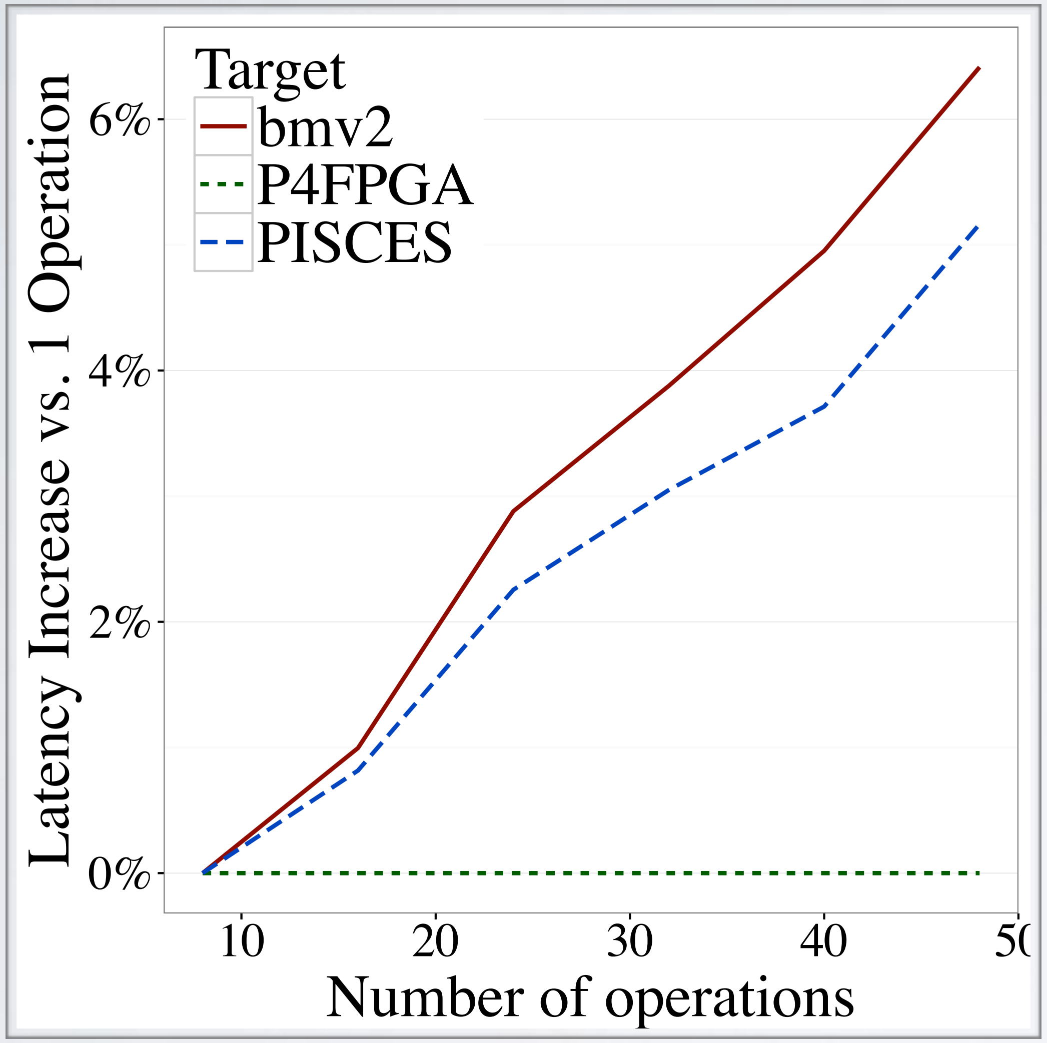
- This behavior doesn’t exist in $P4_{16}$-to-PX Xilinx SDNet

Parse Header



- All results are normalized to the latency of parsing 1 header for each particular targets
- 30% increase for P4FPGA
- 7% increase for PISCES

Action Complexity

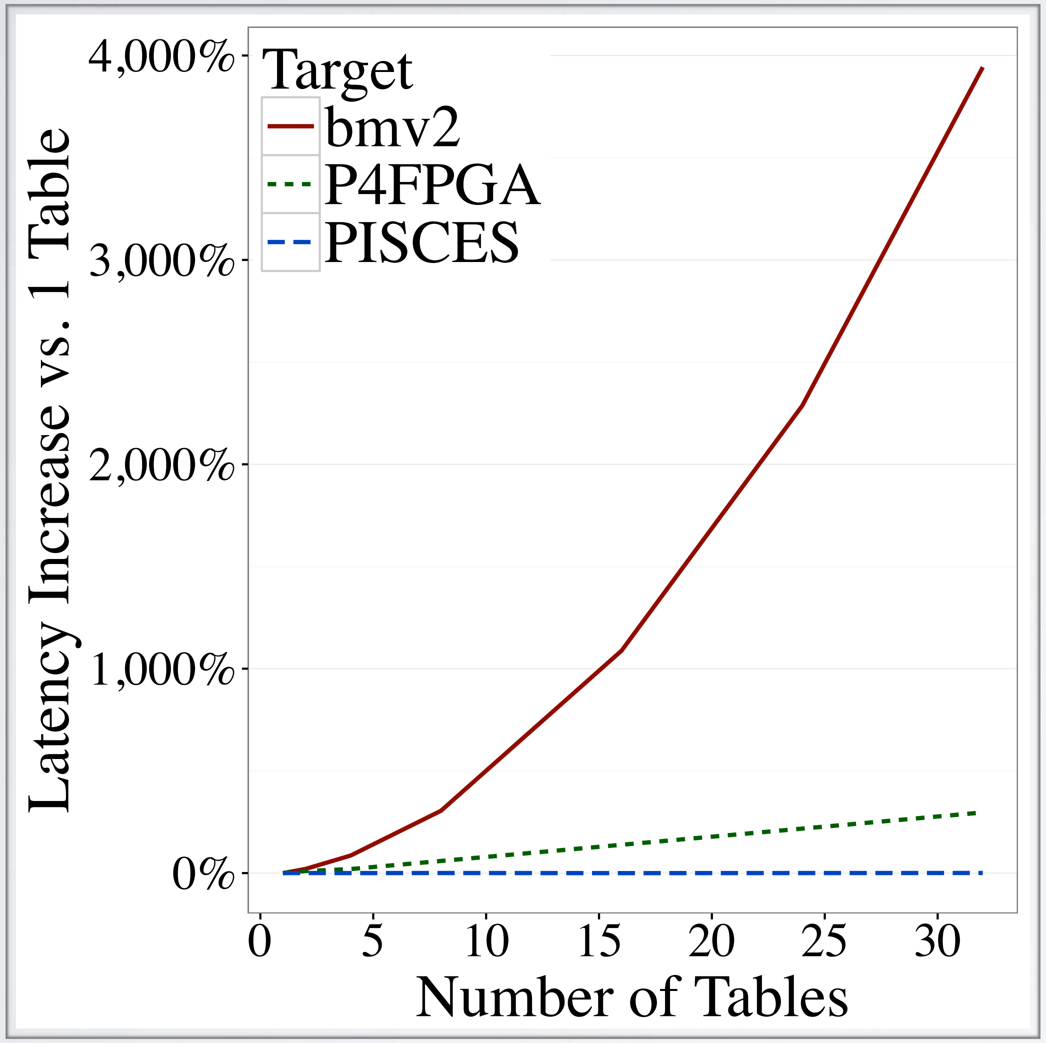


- All results are normalized to the latency of write to 1 header field for each particular targets

- P4FPGA schedules independent writes in 1 clock cycle

- Bmv2 and PISCES execute field write sequentially

Processing Pipeline



- All results are normalized to the latency of applying 1 table for each particular targets

- Tables in PISCES are converted to 1 big table

Conclusion

Whippersnapper:

- is a synthetic benchmark suite for P4
- provides a standard evaluation for heterogeneous P4 platforms
- helps spur innovations

Thank You For Listening
Q & A

https://github.com/usi-systems/p4benchmark