CHIMERA:
Fuzzing P4 Network Infrastructure for Multi-Plane Bug Detection and Vulnerability Discovery

Jiwon Kim[1], Dave (Jing) Tian[1], and Benjamin E. Ujcich[2]

[1]Purdue University

[2]Georgetown University

46th IEEE Symposium on Security and Privacy 2025

PURDUE UNIVERSITY

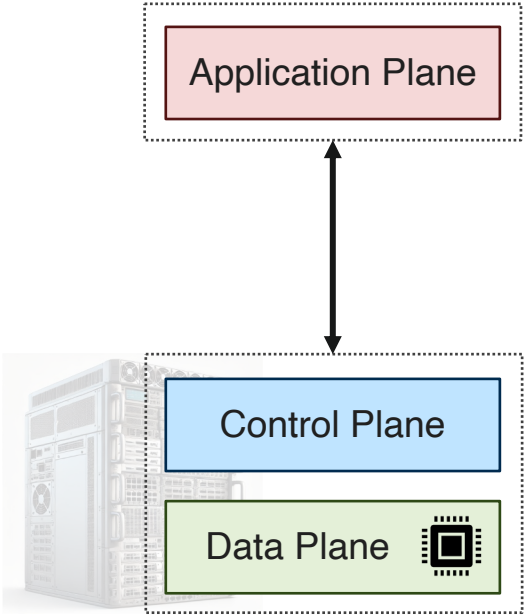
CERIAS

PurSec Lab

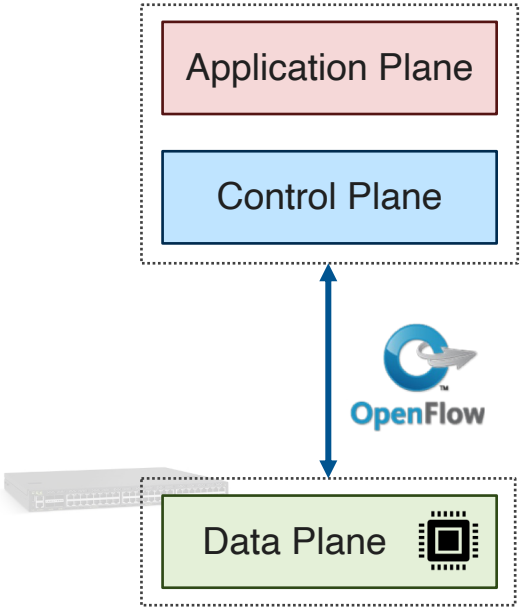
GEORGETOWN UNIVERSITY

GEORGETOWN SECURITY LAB

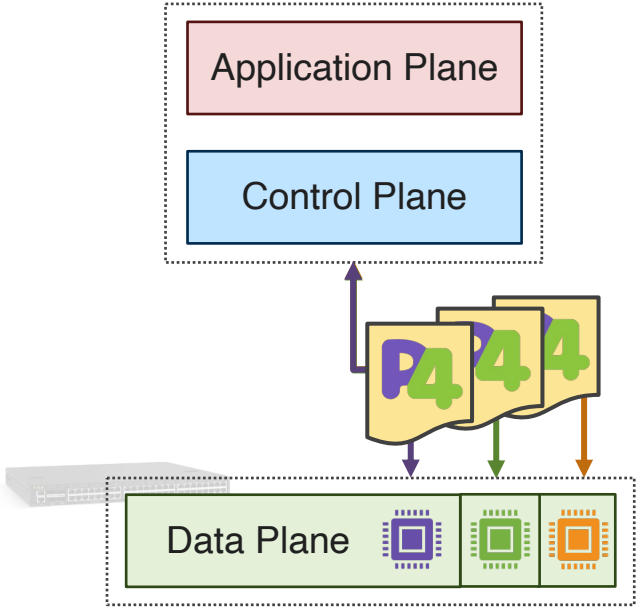
Software Defined Networking



Traditional Networking



Software-Defined Networking with Predefined Protocols



Programmable Networking with Flexible Protocols

How P4 Works

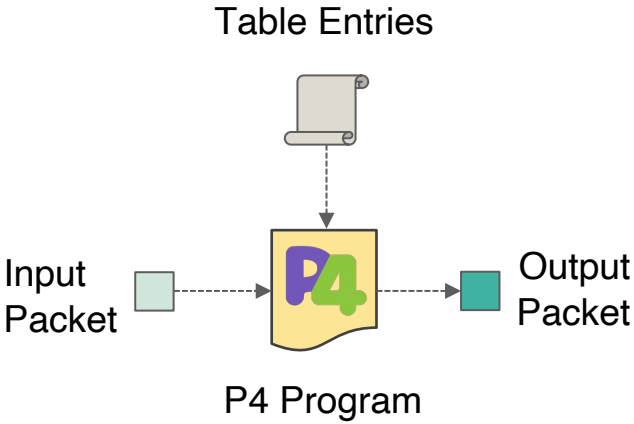
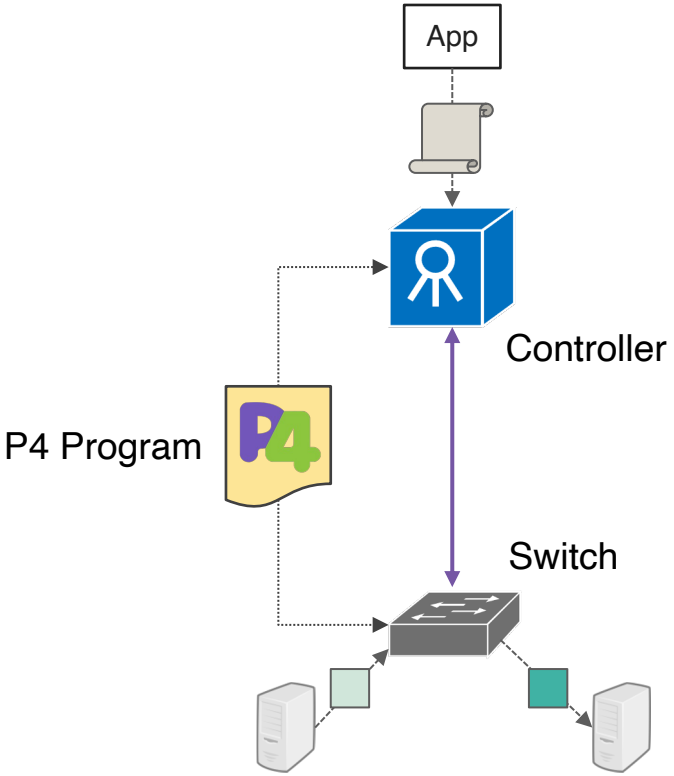
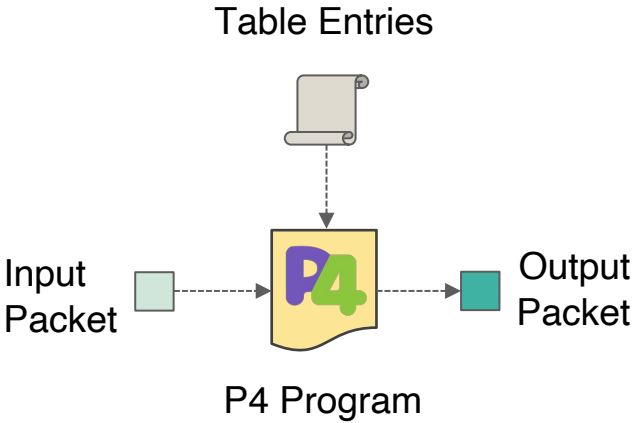
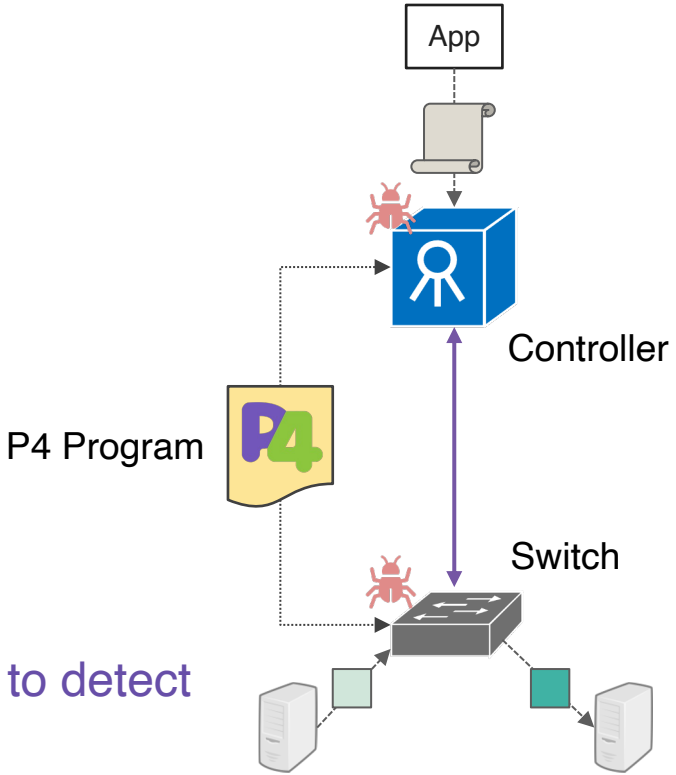


Table Entries
Input Packet
Output Packet
P4 Program


App
Controller
P4 Program
Switch

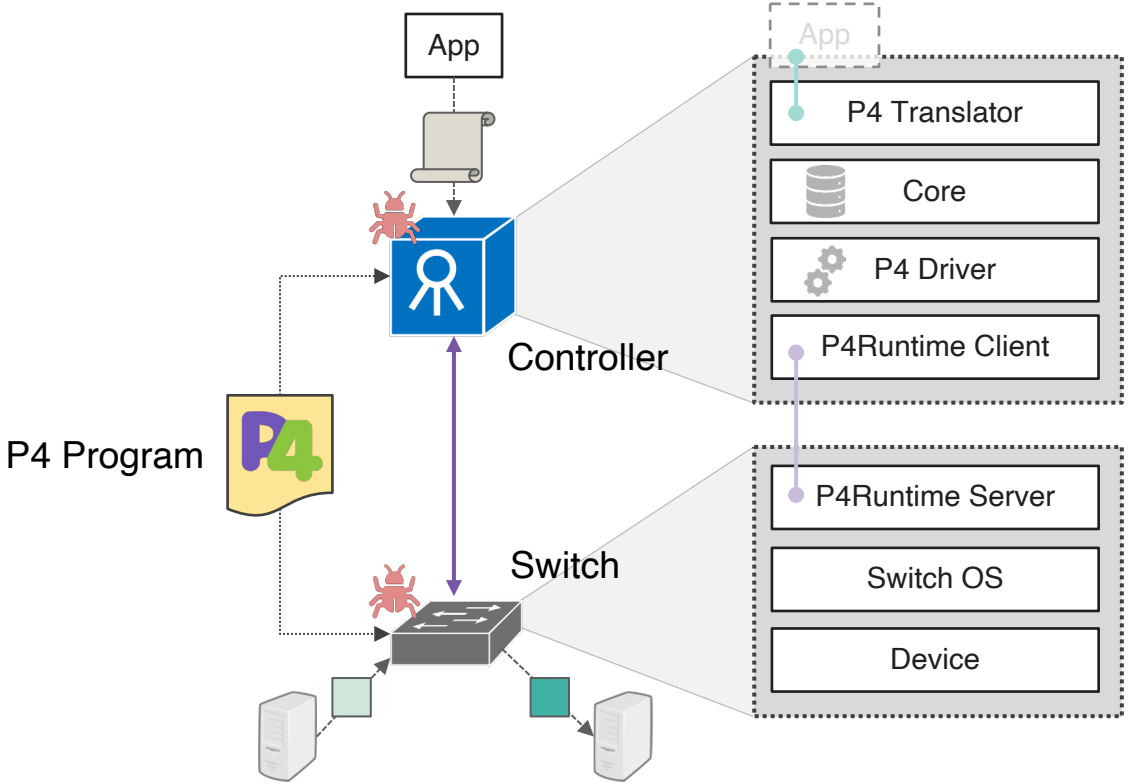
P4 Infrastructure Bugs



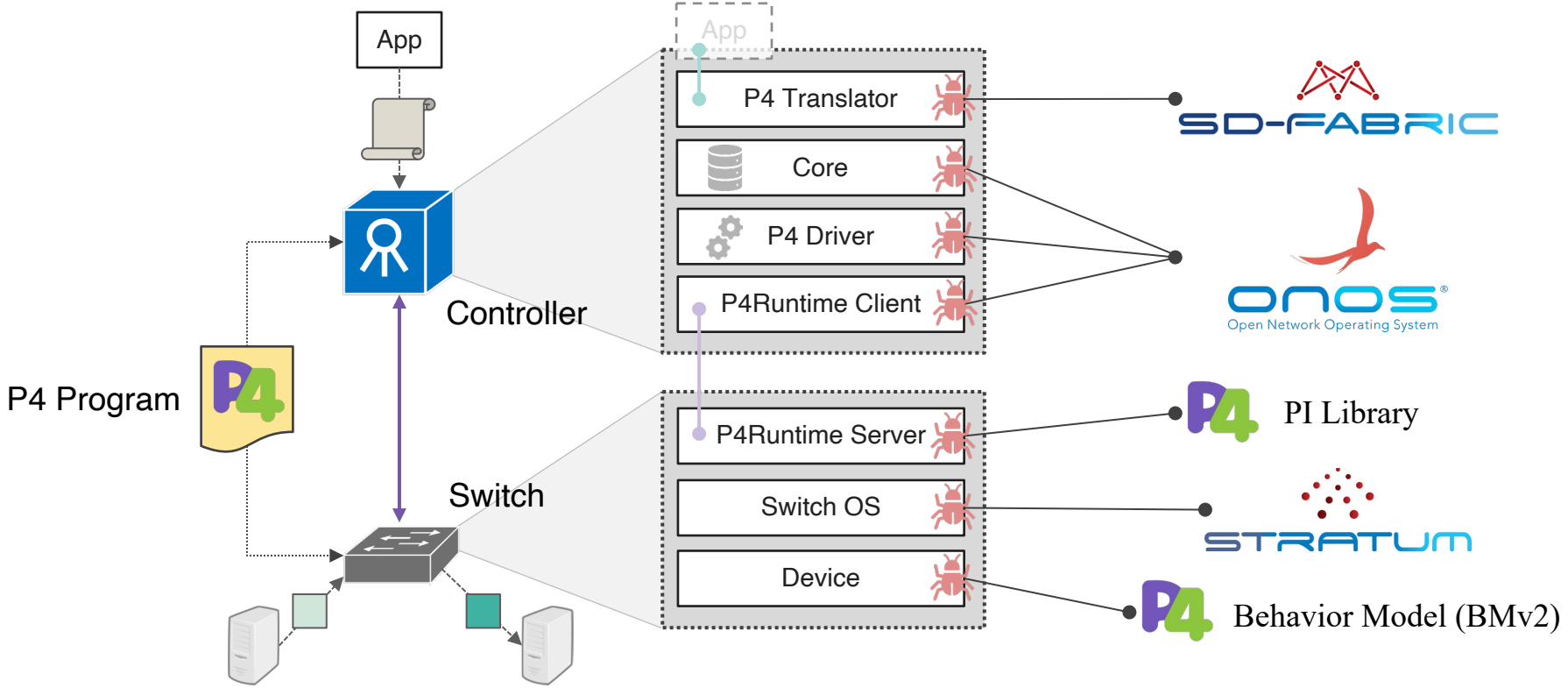


Finding 1. The P4 program can be a criterion to detect P4 infrastructure bugs.

Complex Programmable Networking Stack



Complex Programmable Networking Stack



Bug Study of Multi-Plane P4 Infrastructure

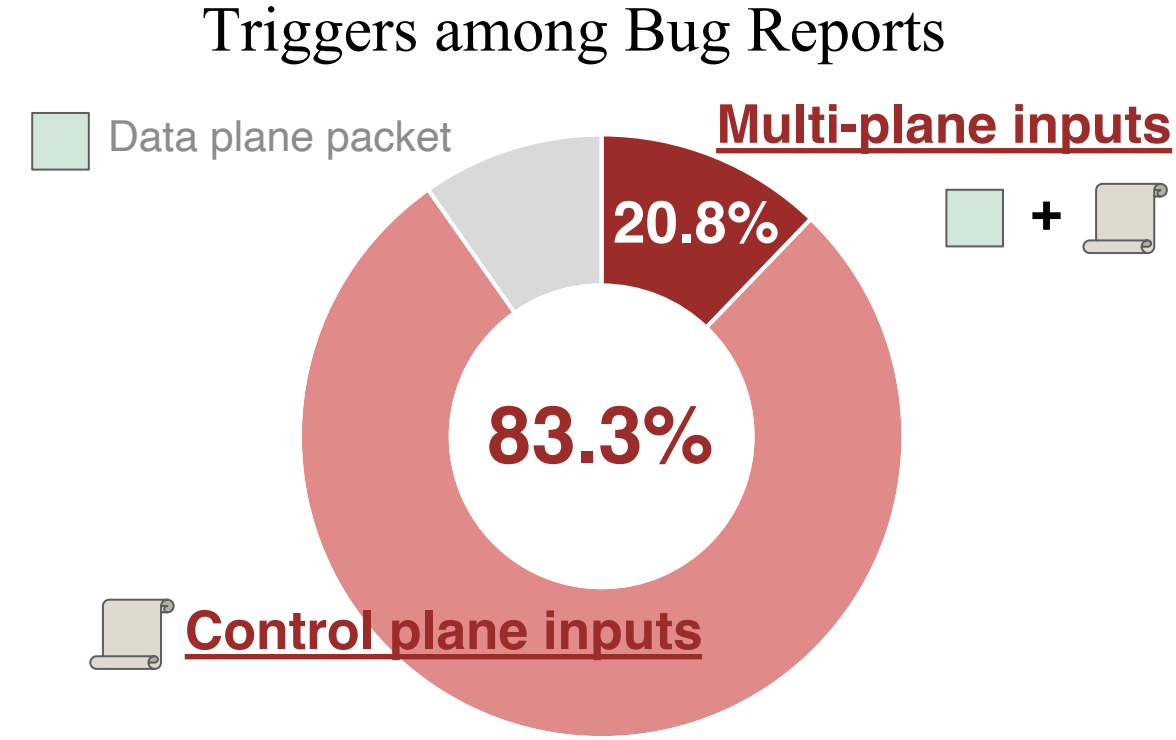
SD-FABRIC

ONOS Open Network Operating System

PI Library

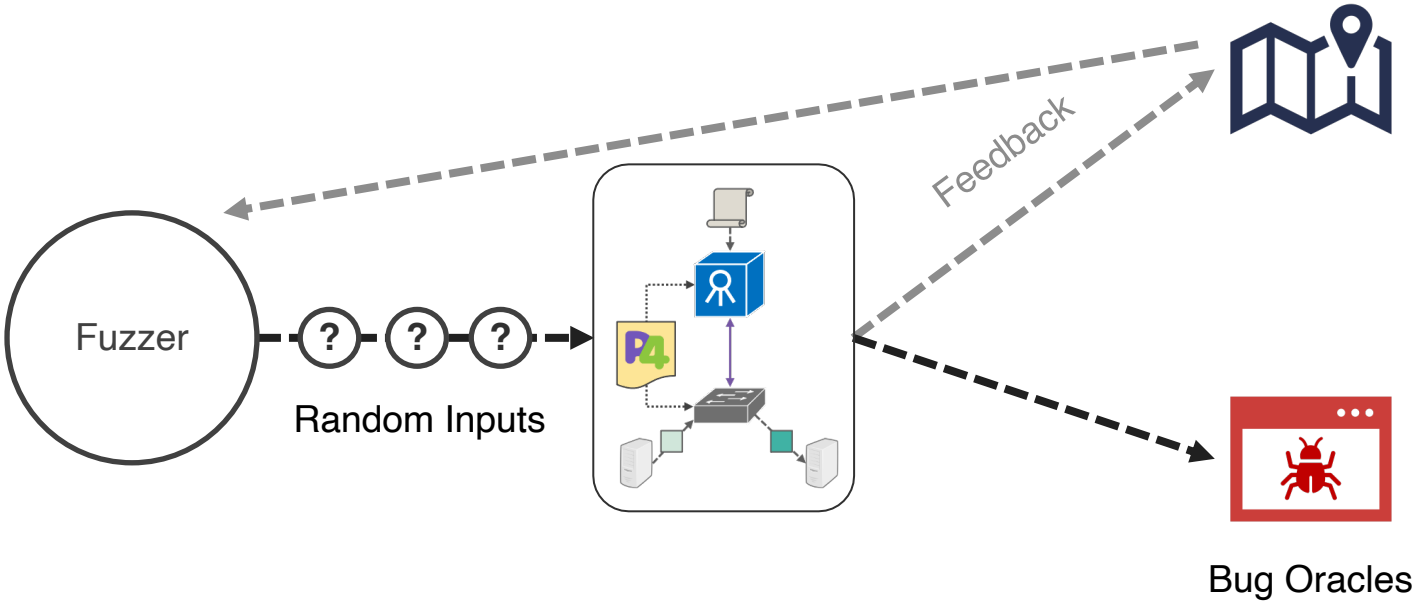
STRATUM

Behavior Model (BMv2)



Finding 2. To find more bugs, P4 fuzzer needs **multi-plane inputs**.

Objective: Fuzzing P4 Infrastructure



1 Inputs of Single-Plane SDN Fuzzing Tools

Data Plane Fuzzers

- Mutate data plane packets
- Use a fixed set of control plane rules

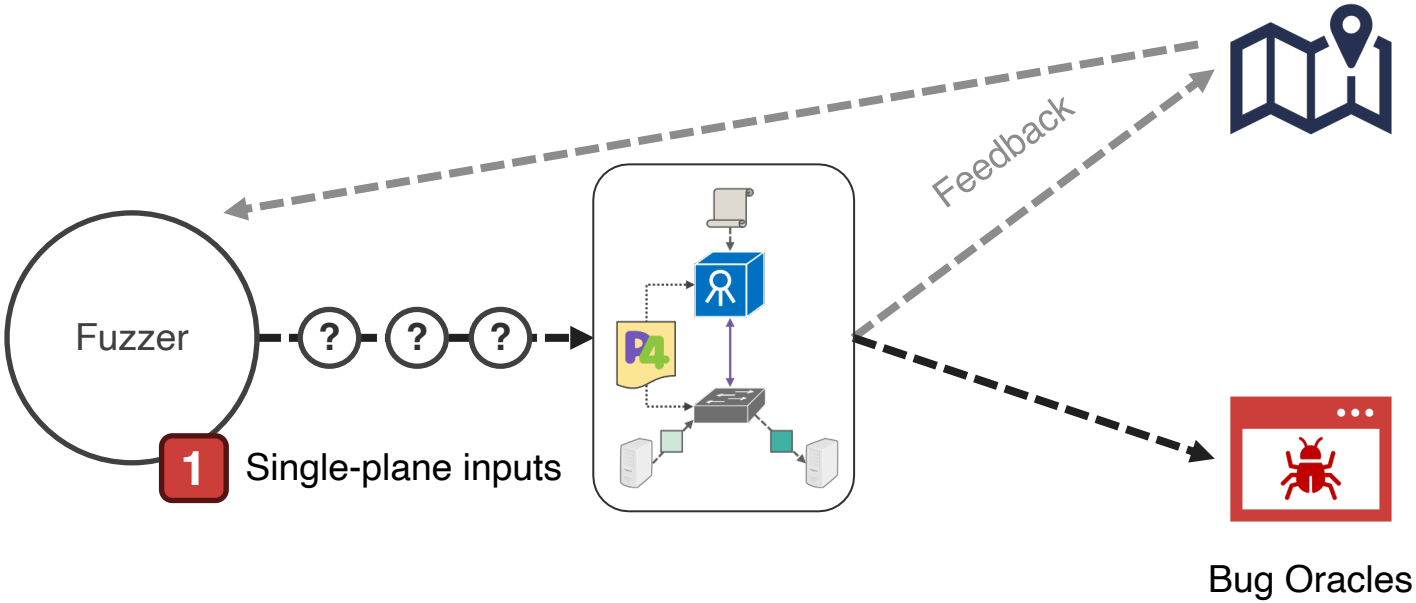
  e.g., FP4 [Arxiv '22], P6 [INFOCOM '21]

Control Plane Fuzzers

- Mutate control plane messages (e.g., flow rules)
- Generate a special data plane traffic

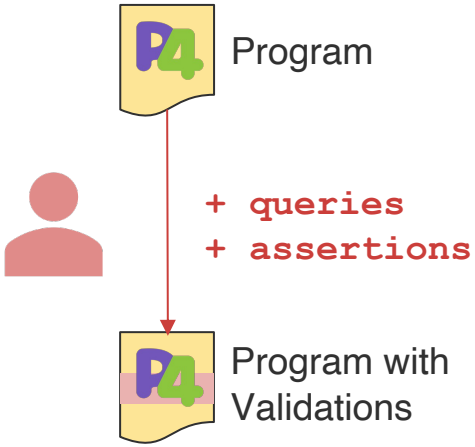
  e.g., Intender [Sec '23], AIM-SDN [CCS '18], DELTA [NDSS '17]

Limitations in Fuzzing P4 Infrastructure



2 Bug Discovery in Existing P4 Fuzzing Tools

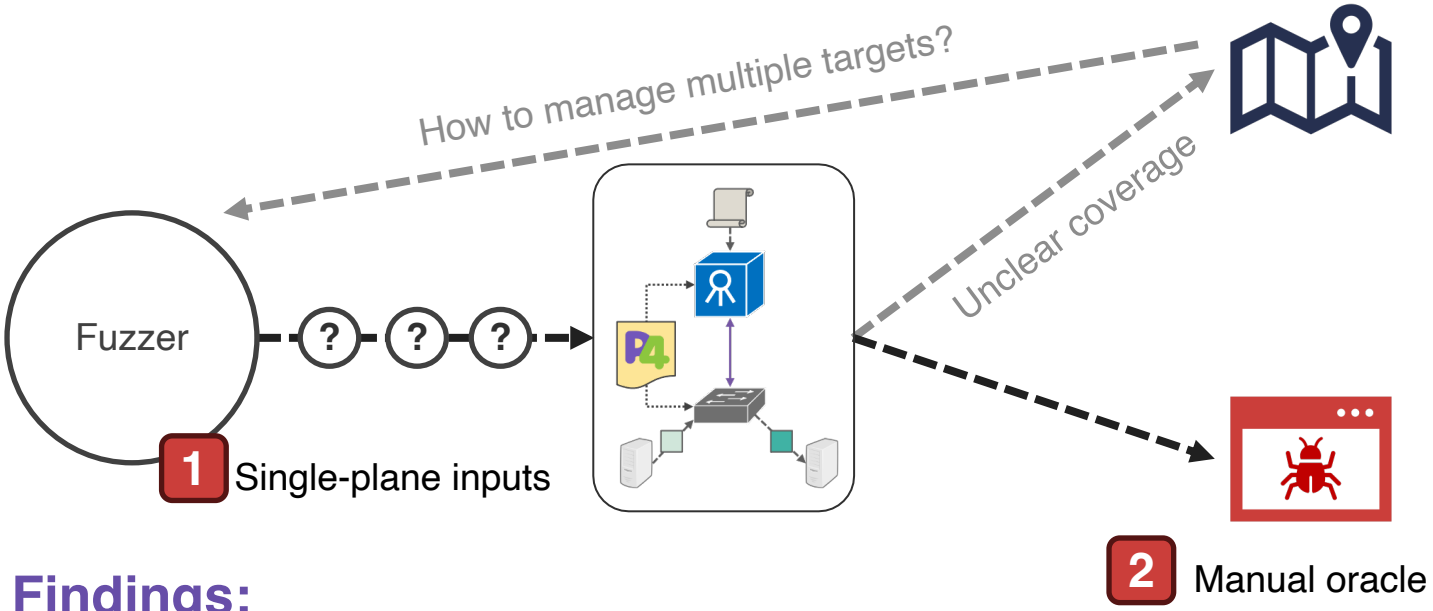
Manual Validation (e.g., FP4 [Arxiv '22], P6 [INFOCOM '21])



Non-trivial manual effort

✗ Incomplete and error-prone oracle

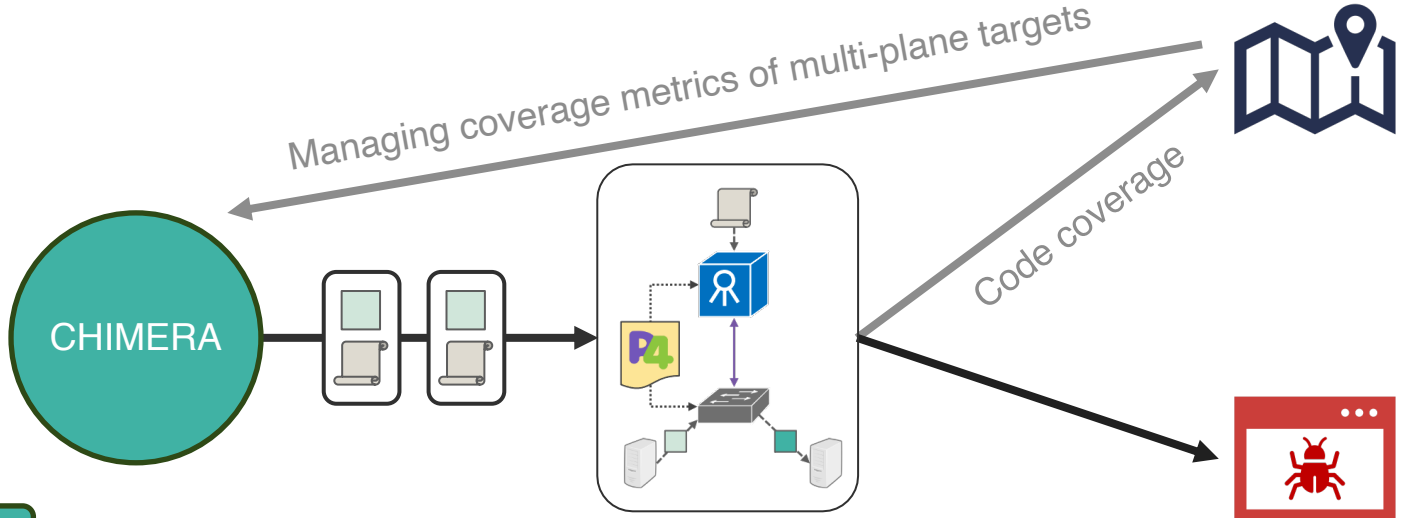
Limitations in Fuzzing P4 Infrastructure (cont'd)



Our Findings:

- To find more bugs, P4 fuzzer needs **multi-plane inputs**.
- The P4 program can be a **criterion** to detect P4 infra bugs.

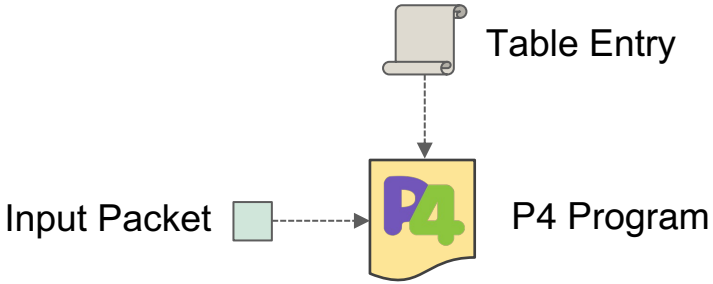
CHIMERA: Fuzzing P4 Infrastructure



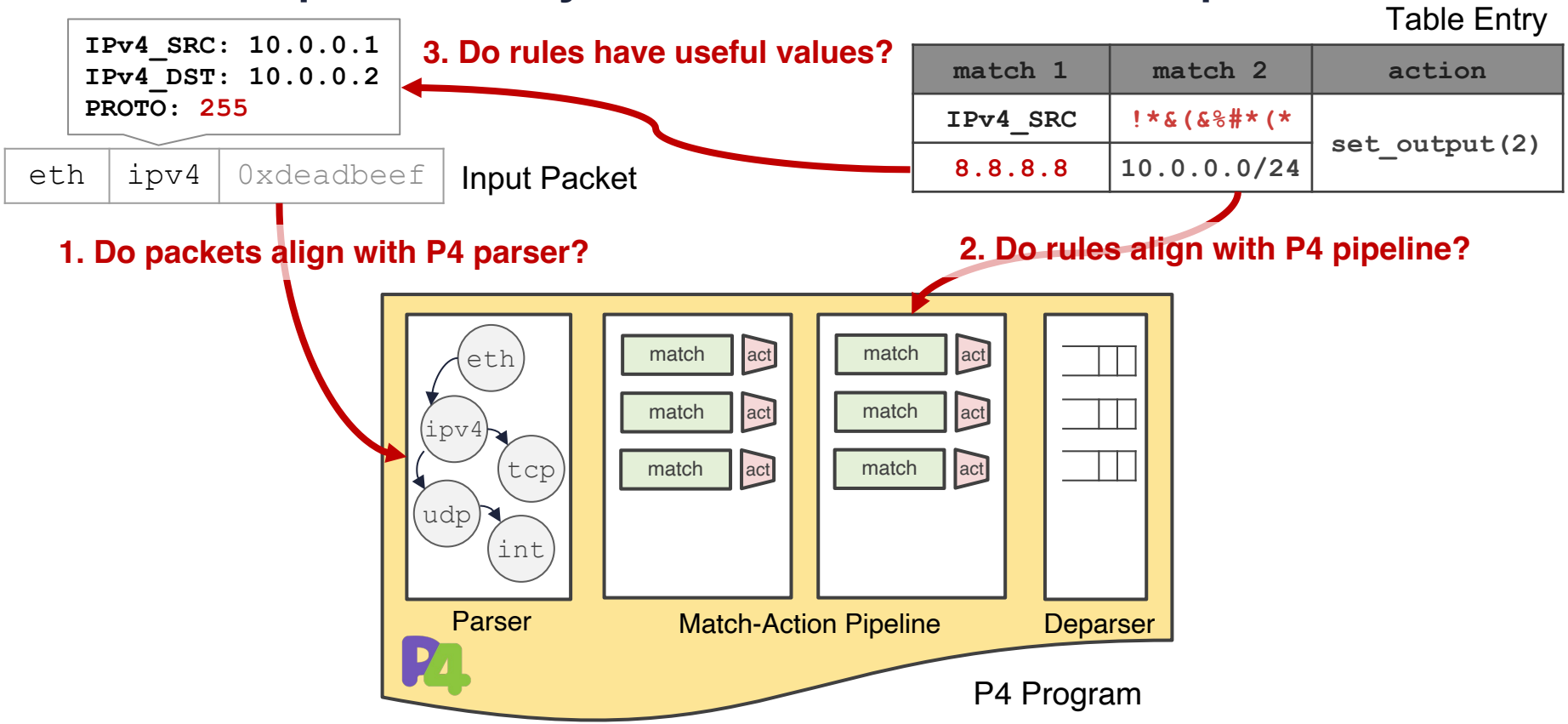
1 **Multi-plane testing with smart mutators**

2 **P4CE as a bug oracle**

Interdependency in Multi-Plane P4 Inputs



Interdependency in Multi-Plane P4 Inputs



1 Smart Mutators for P4

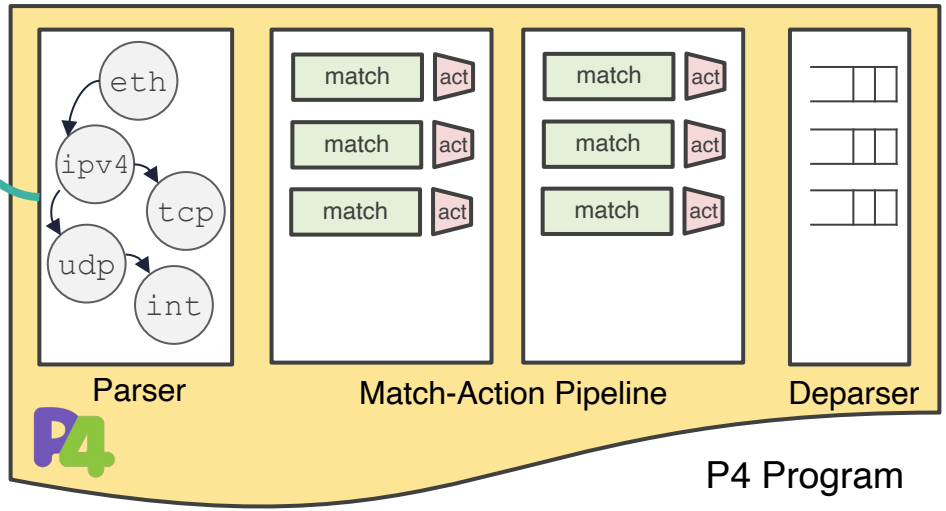
```
IPv4_SRC: 10.0.0.1
IPv4_DST: 10.0.0.2
PROTO: 255
```

eth	ipv4	0xdeadbeef

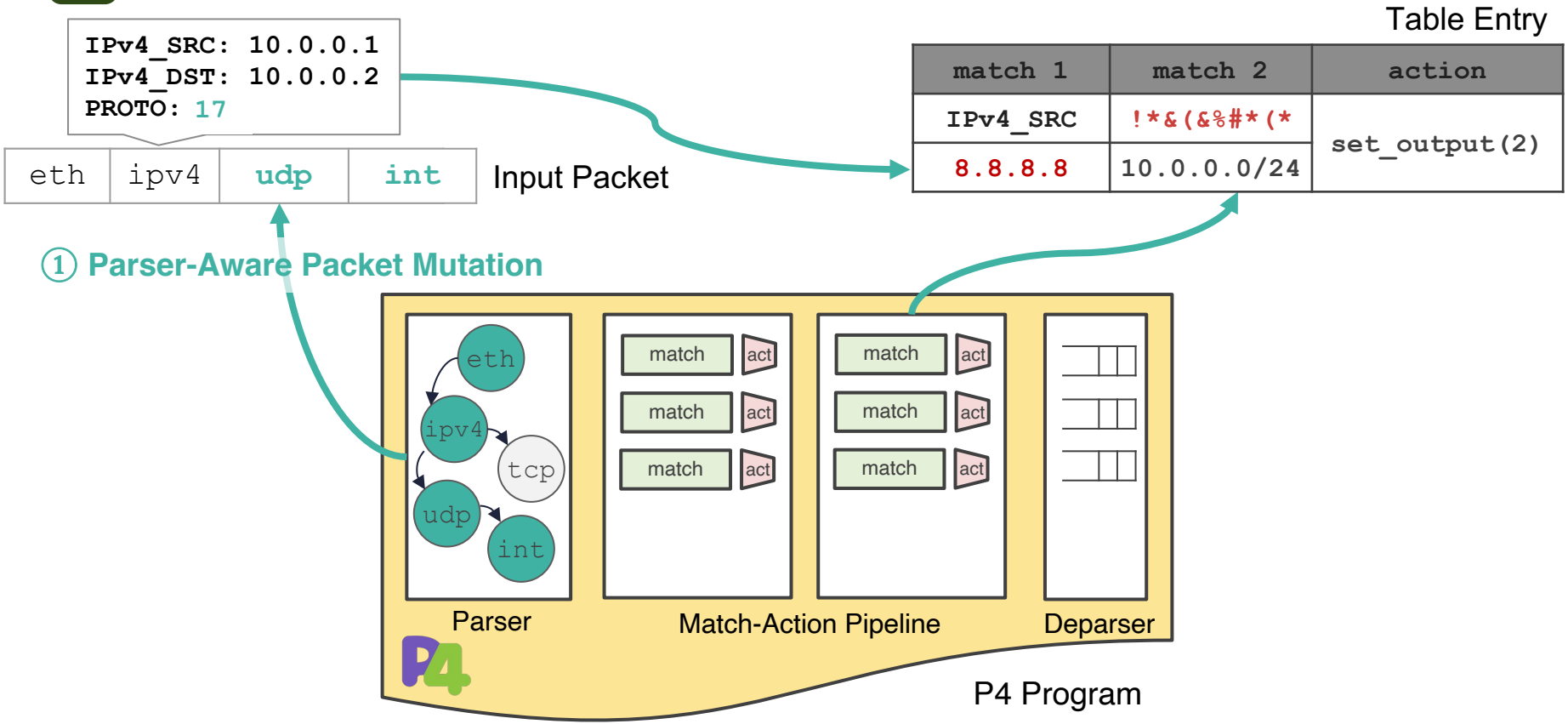
Input Packet

Table Entry

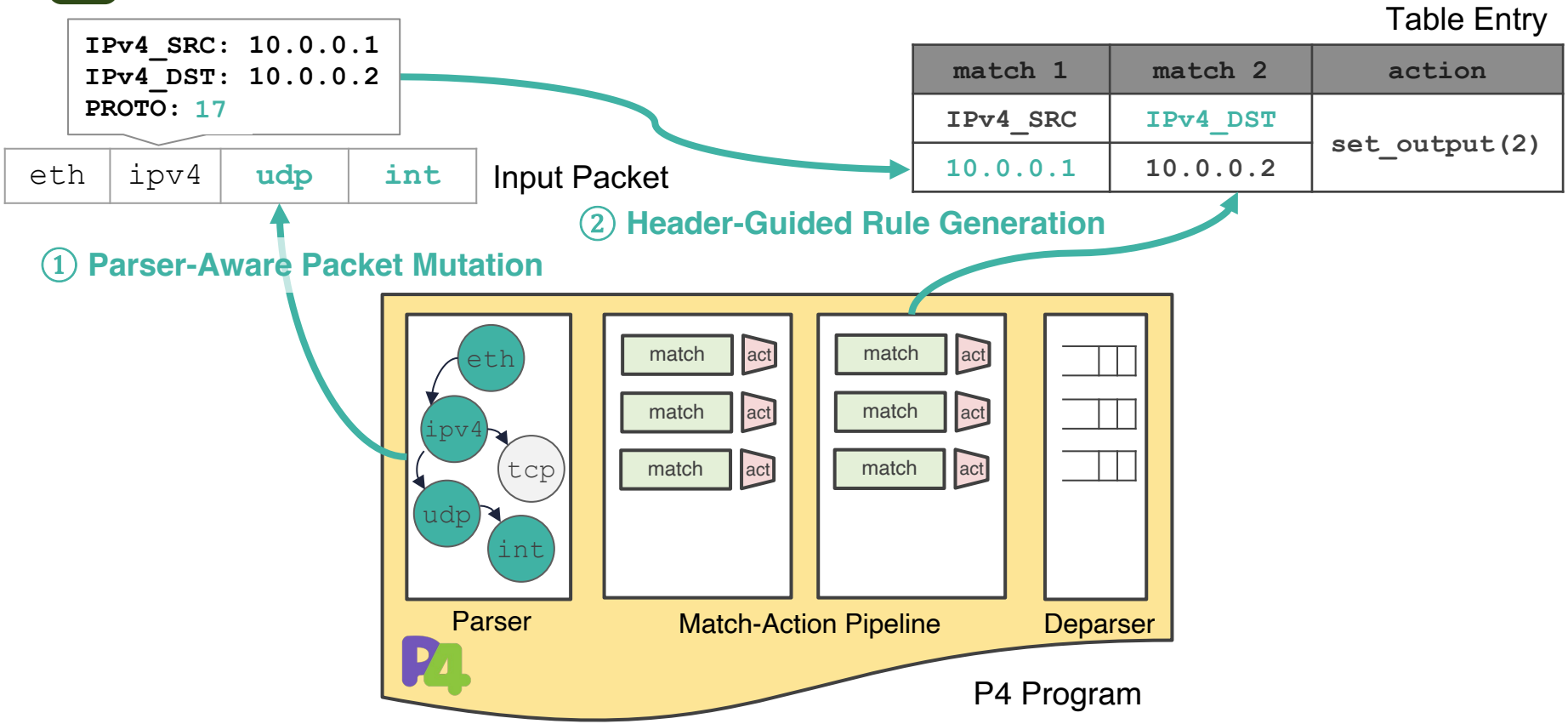
match 1	match 2	action
IPv4_SRC	!*&(&%#*(*	set_output(2)
8.8.8.8	10.0.0.0/24	



1 Smart Mutators for P4

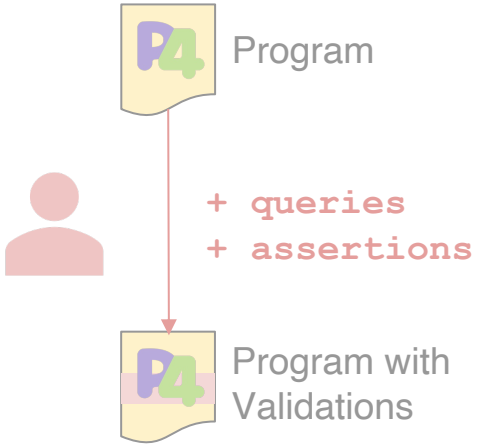


1 Smart Mutators for P4

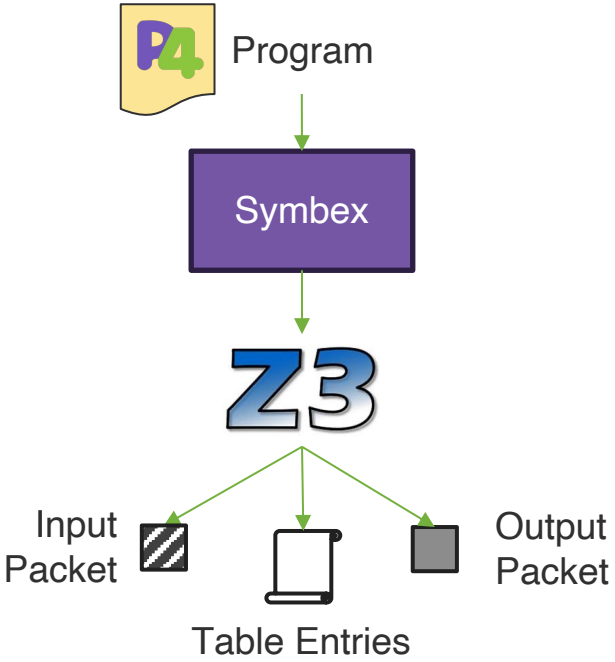


Existing Approaches in Bug Discovery

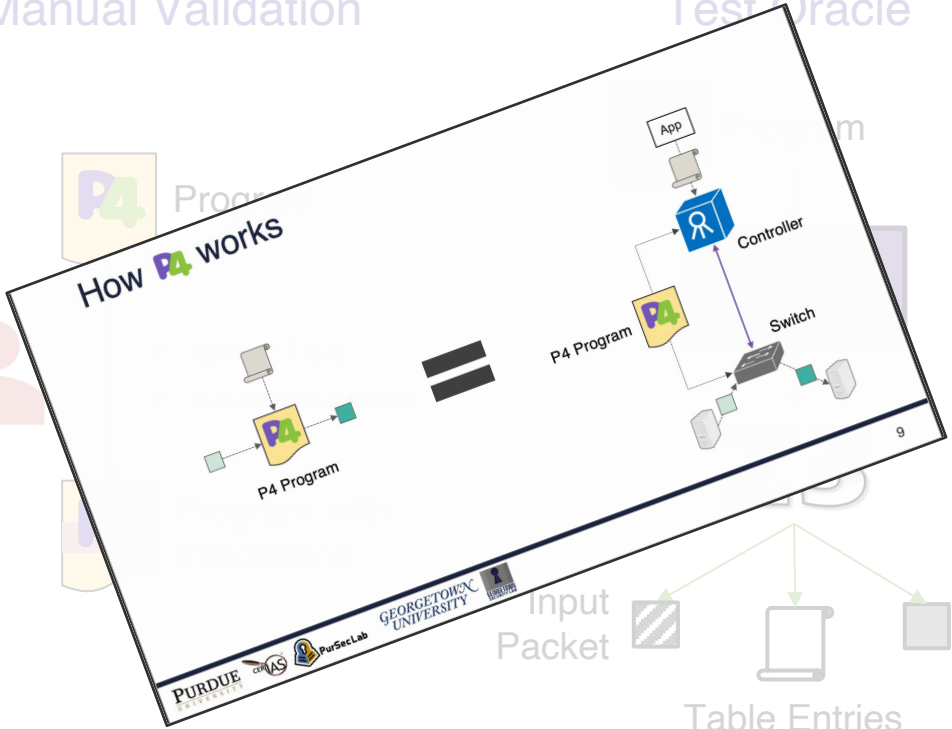
Manual Validation



Test Oracle



2 Concolic Execution as a Bug Oracle



Input Packet

Table Entries

Program

P4CE

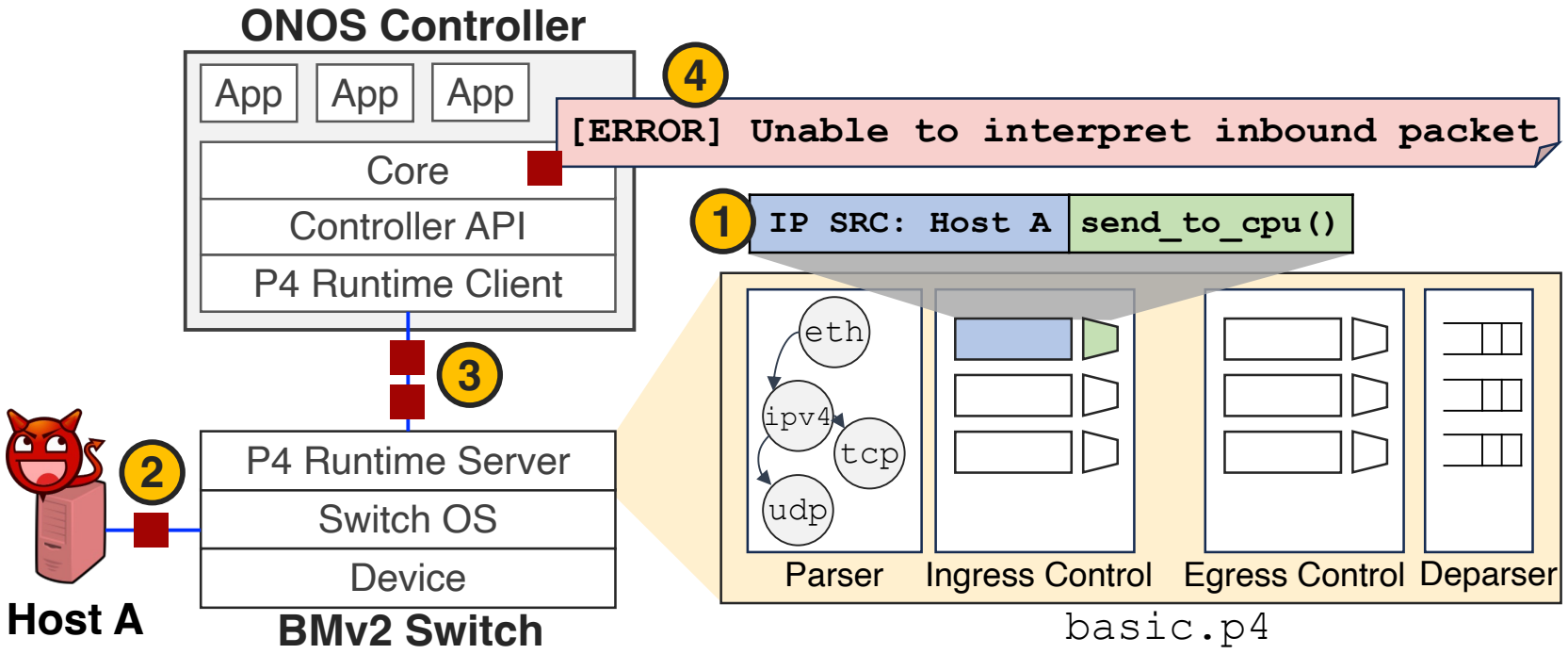
Output Packet

Evaluation

- Found **7 new bugs** with **2 CVEs assigned**
  - **2 bugs** triggered by multi-plane inputs
  - **5 bugs** detected by P4CE
  - Security impacts: network-wide **denial of service** & **tampering**
- Compare 2 existing fuzzers: FP4 and Intender
  - Highest coverage in both the data plane (**1.16x**) and the control plane (**1.1x**)
  - Up to **3.5x** higher bug detection rate

Case Study: CVE-2024-53423

Cross-plane attack to cause denial of service in controller



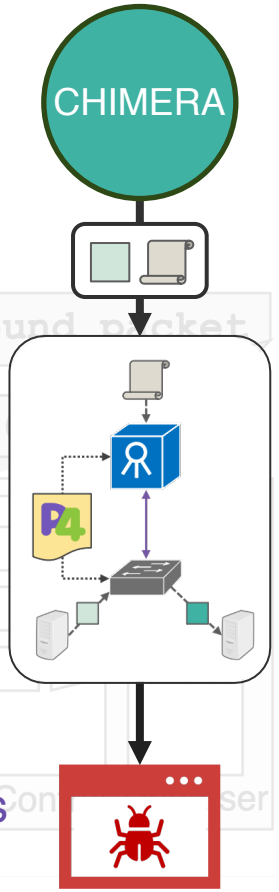
How does CHIMERA find this bug?

① Finding cross-plane bugs needs **multi-plane testing**

② **PAPM** guarantees packets specific to a P4 program

③ **HGRG** generates valid rules that match packets

④ **P4CE** can detect abnormal behavior of P4 infra different from P4 programs



Conclusion

- Investigated **41 bug reports** in 5 P4 open-source projects
- Designed new fuzzing techniques for P4 infrastructure
  - **P4CE** as a bug oracle
  - Parser-Aware Packet Mutation (**PAPM**)
  - Header-Guided Rule Generation (**HGRG**)
- Found **7 bugs (2 CVEs)** in P4 infrastructure

Thank you!

EMAIL

WEBSITE

CODE