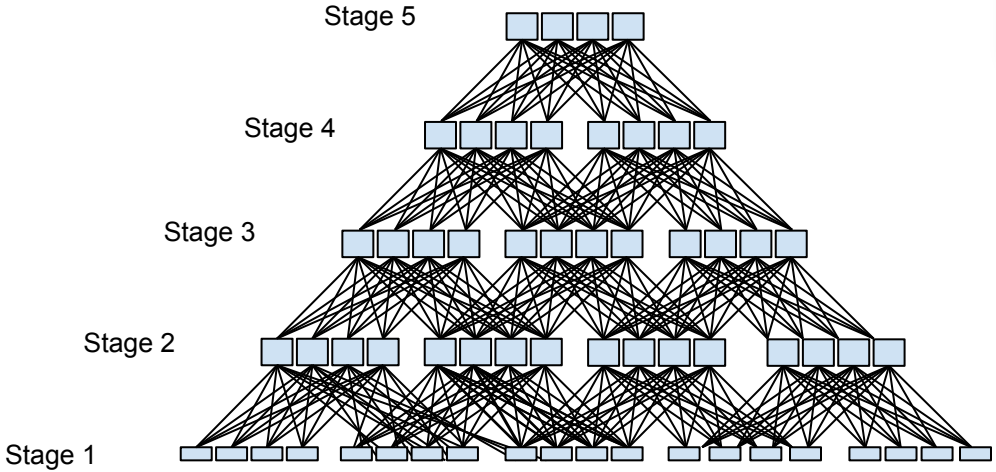
Google Cloud

Using P4 and P4Runtime for Optimal L3 Routing

Stefan Heule <heule@google.com>
Google, Network Infrastructure

P4 Expert Roundtable Series
April 28, 2020

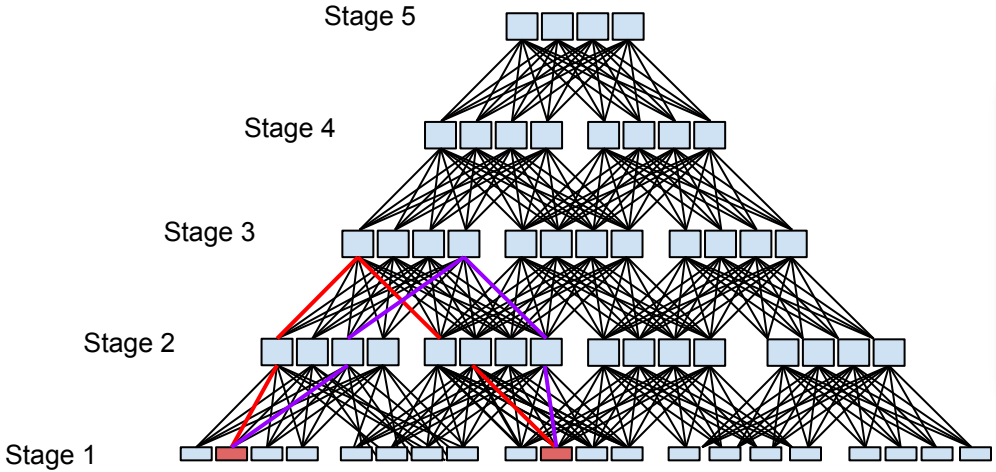
Clos Topologies



Large virtual switch built out of small commodity switches

[1] C. Clos, "A study of non-blocking switching networks" in The Bell System Technical Journal, Vol. 32

Clos Topologies



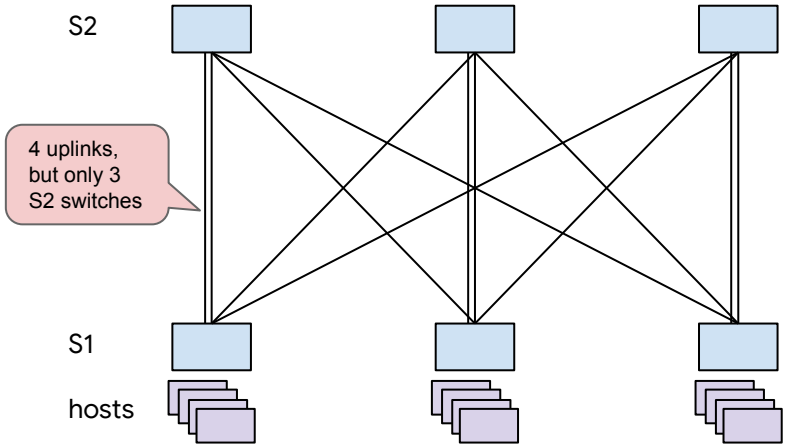
Large virtual switch built out of small commodity switches

Highly redundant paths between nodes ensures fault-resilience

ECMP to load-balance

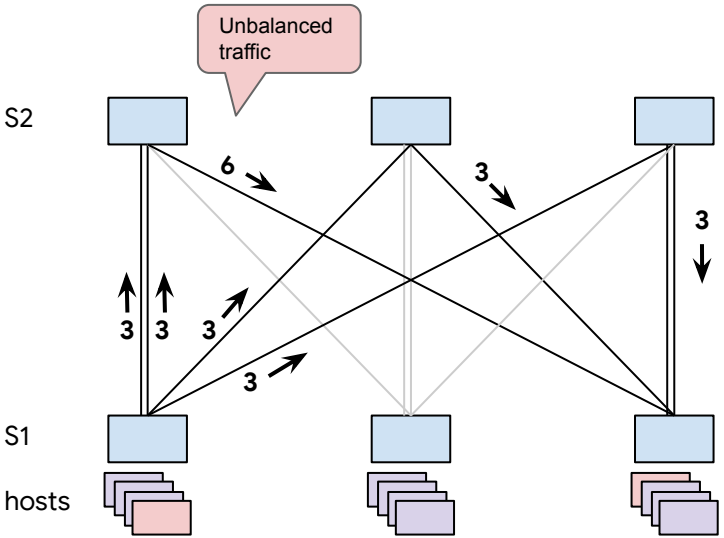
[1] C. Clos, "A study of non-blocking switching networks" in The Bell System Technical Journal, Vol. 32

Sources of Imbalance



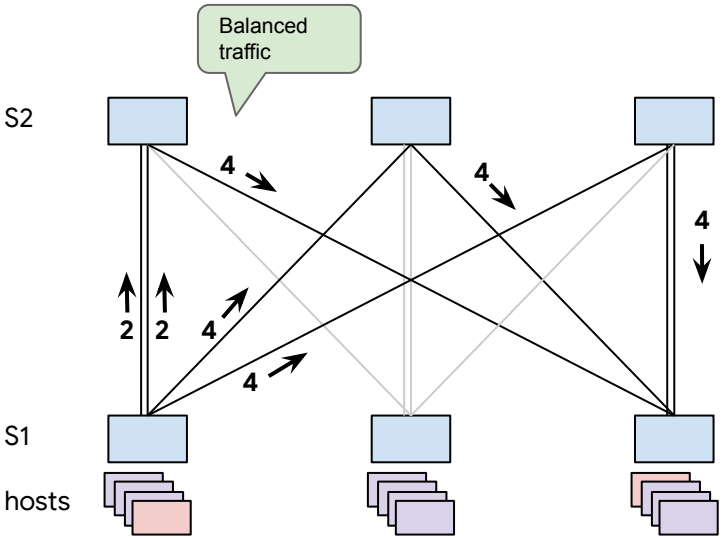
Number of uplinks is not integer multiple of number of switches per stage

Sources of Imbalance



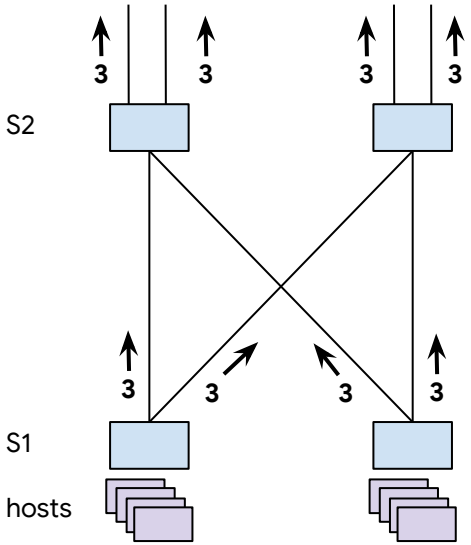
Number of uplinks is not integer multiple of number of switches per stage

Sources of Imbalance



Number of uplinks is not integer multiple of number of switches per stage

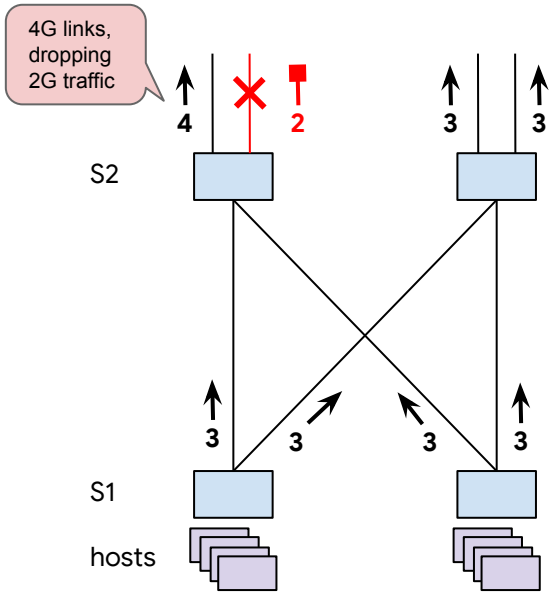
Sources of Imbalance



Number of uplinks is not integer multiple of number of switches per stage

Link and node failures

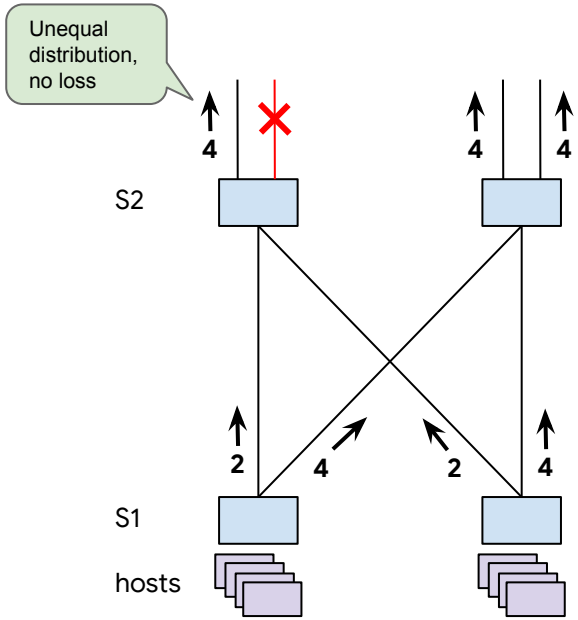
Sources of Imbalance



Number of uplinks is not integer multiple of number of switches per stage

Link and node failures

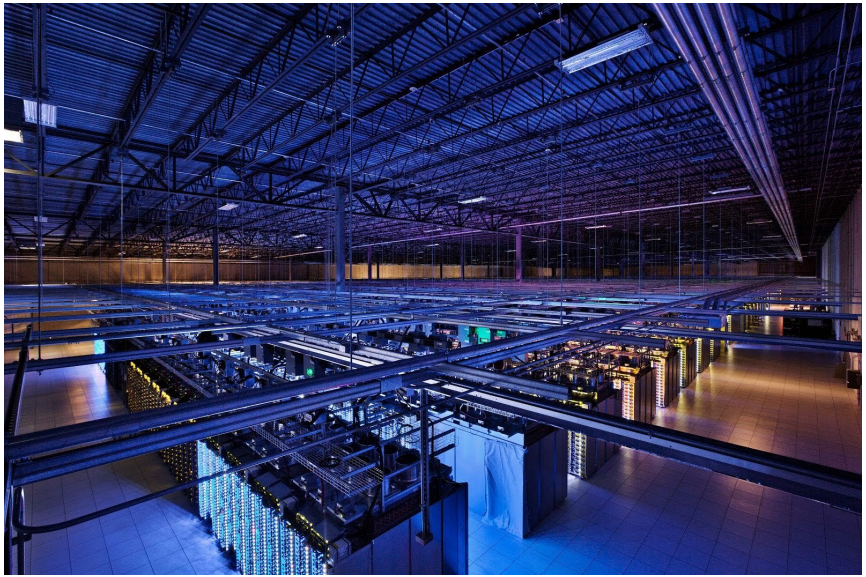
Sources of Imbalance



Number of uplinks is not integer multiple of number of switches per stage

Link and node failures

Sources of Imbalance

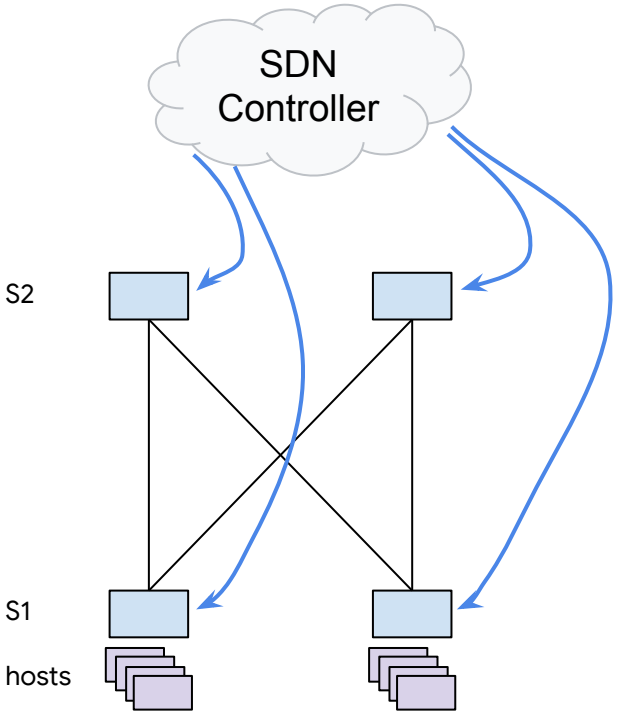


Number of uplinks is not integer multiple of number of switches per stage

Link and node failures

Uneven tree due to partially filled data-centers (allowing for future expansions)

Weighted Cost Multi Path

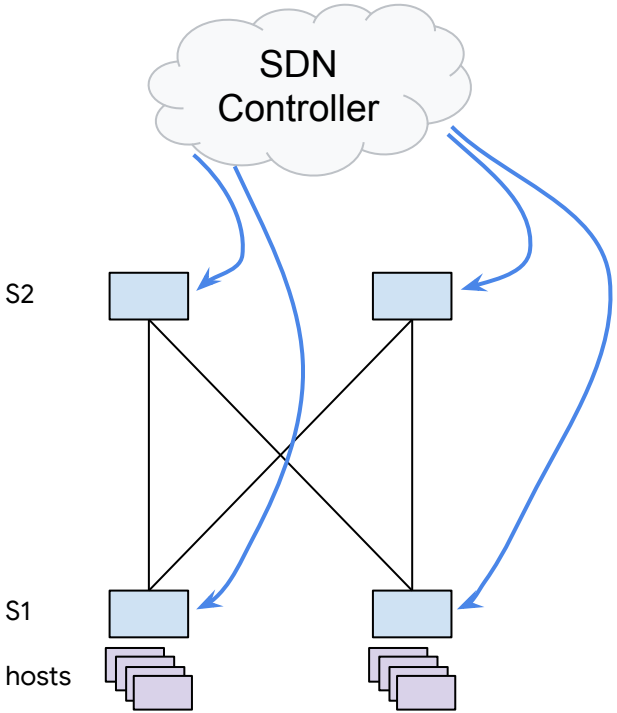


WCMP crucial to optimal utilization of network [2]

Central controller allows allocating optimal weights

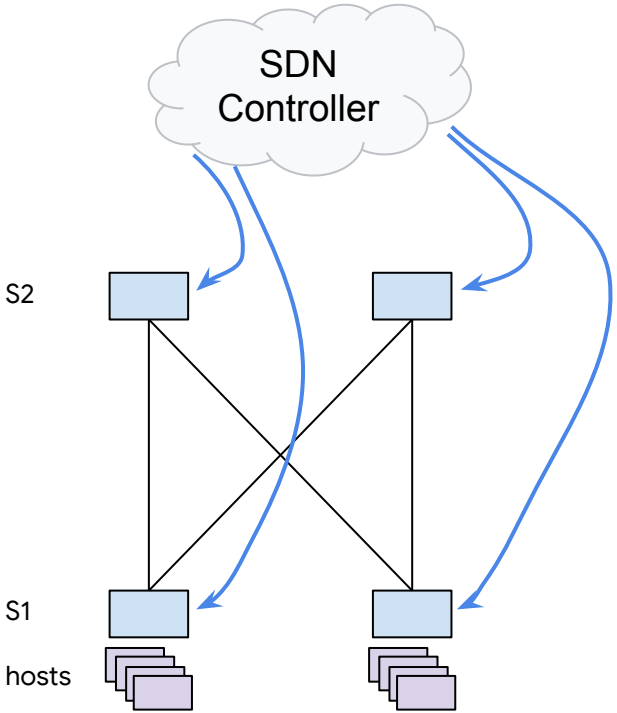
[2] J. Zhou et al., "WCMP: weighted cost multipathing for improved fairness in data centers" at EuroSys '14

Centralized Controller



Weights computed according to
- Available capacity
- Topology
- Current failures
- Policy decisions

Centralized Controller

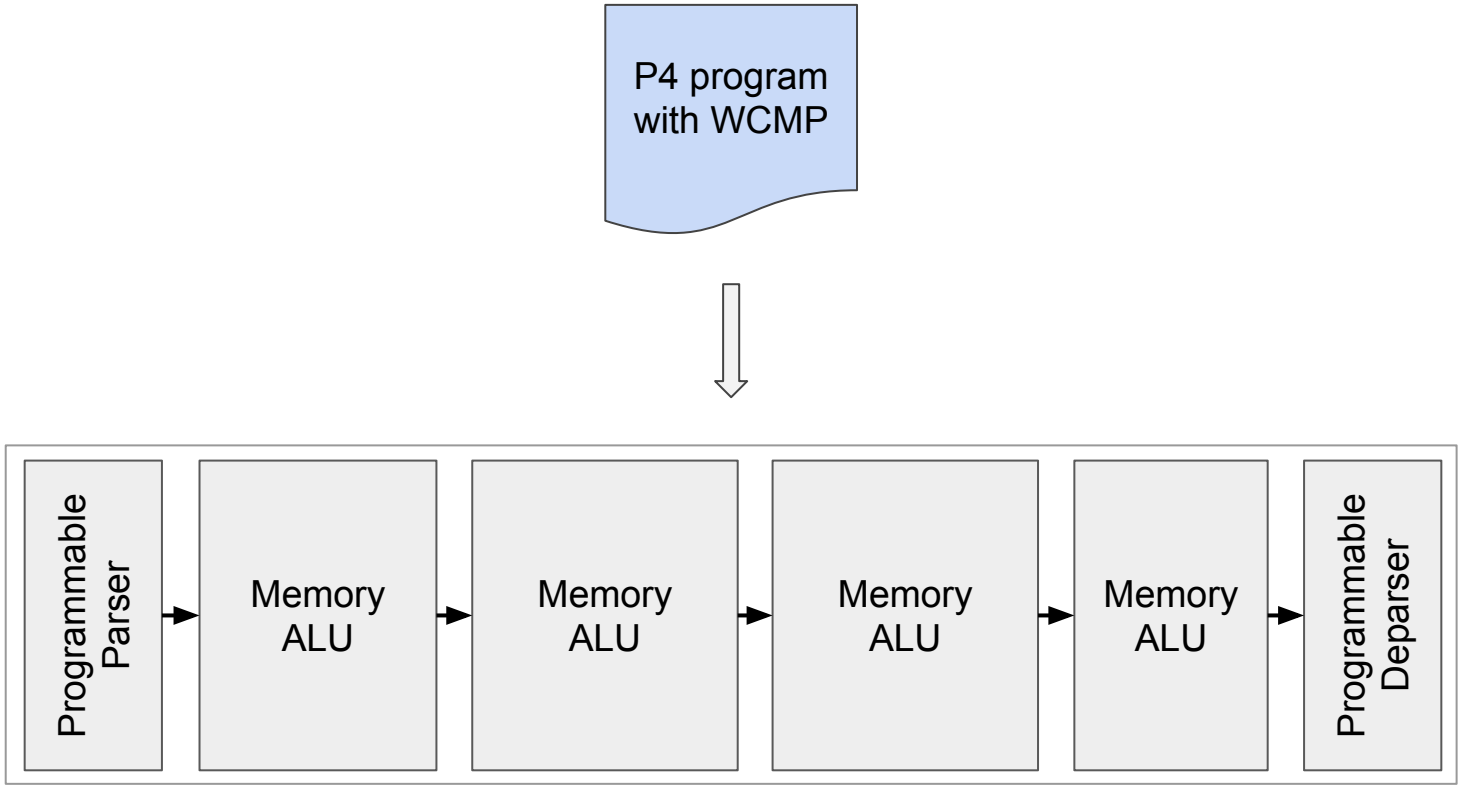


Weights computed according to
- Available capacity
- Topology
- Current failures
- Policy decisions

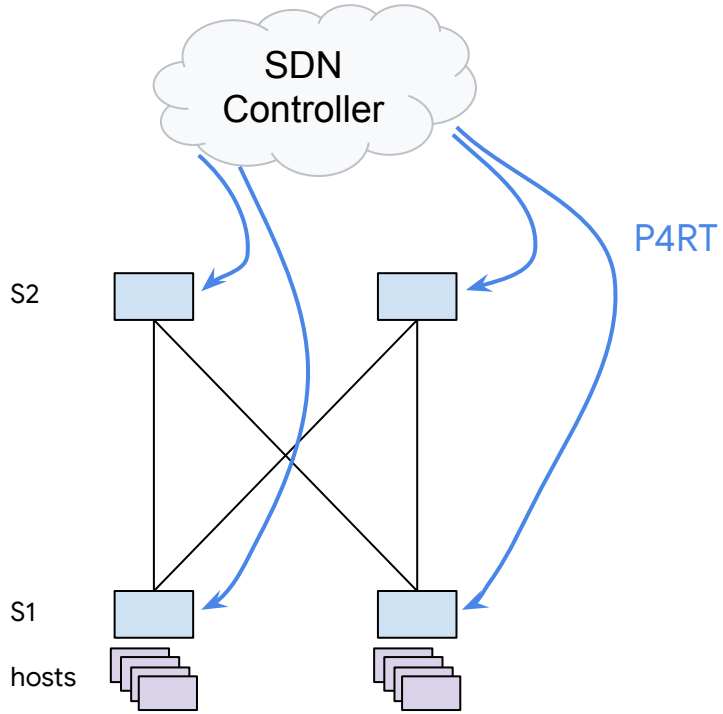
Often switches have maximum sum of weights
- Controller can use more precise weights for more important traffic

How does P4/P4Runtime help with WCMP?

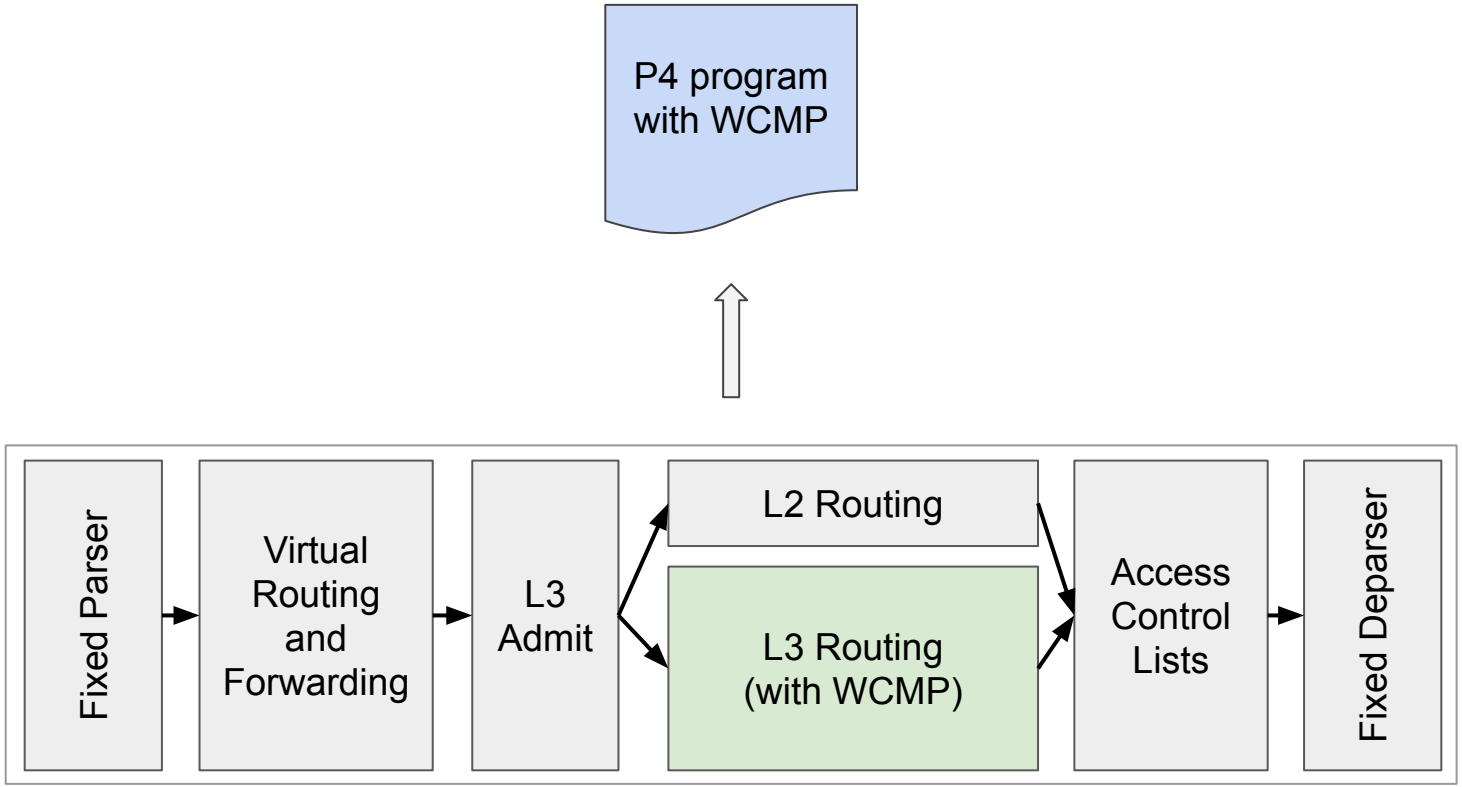
Using P4 to do WCMP: Fully-programmable switches



Using P4 to do WCMP: Fully-programmable switches

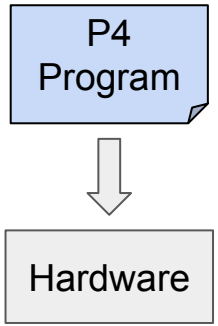


Using P4 to do WCMP: Fixed-Function Switch

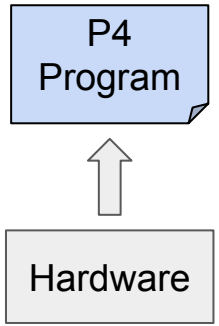


Using P4 at Google

P4 program determines what the hardware does



Hardware determines what the P4 program does



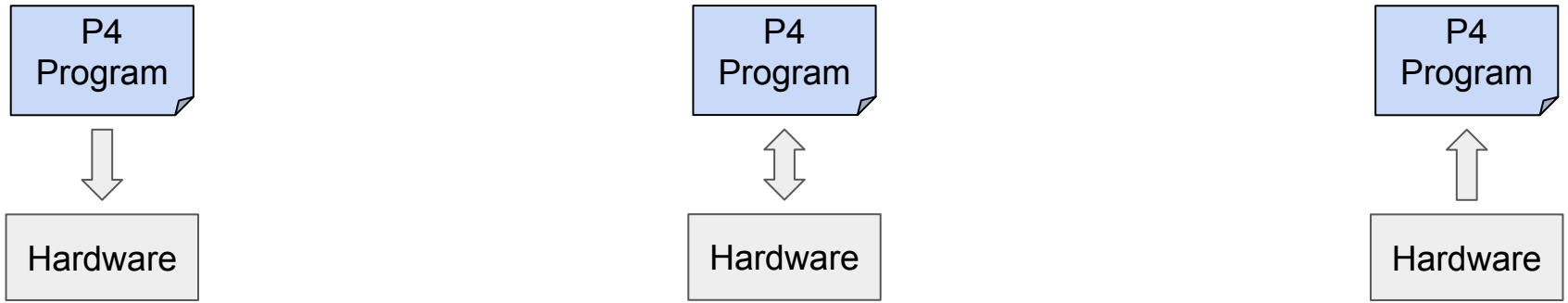
Using P4 at Google

Hardware limits what P4 program can do, but only model our **use case**:

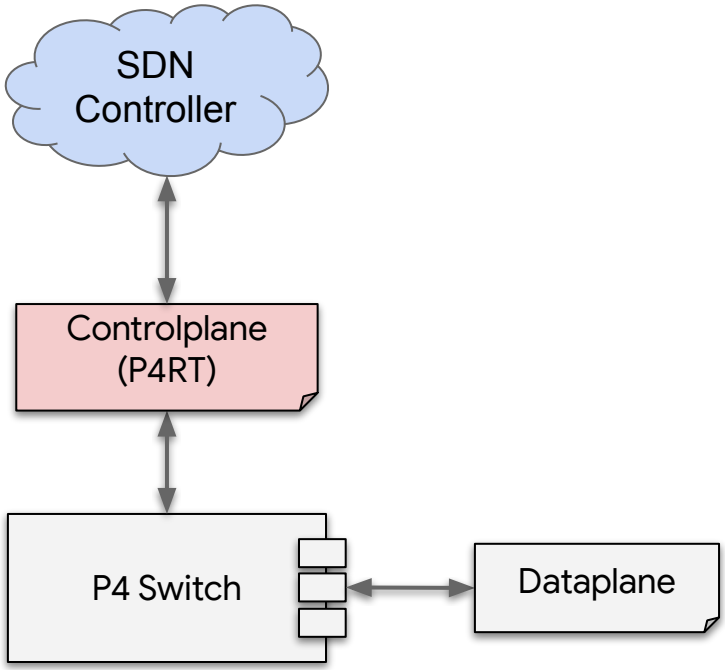
- Only tables we use (e.g. no L2)
- Only match keys we use
- Logical tables that have semantic meaning (abstraction)

P4 program determines what the hardware does

Hardware determines what the P4 program does

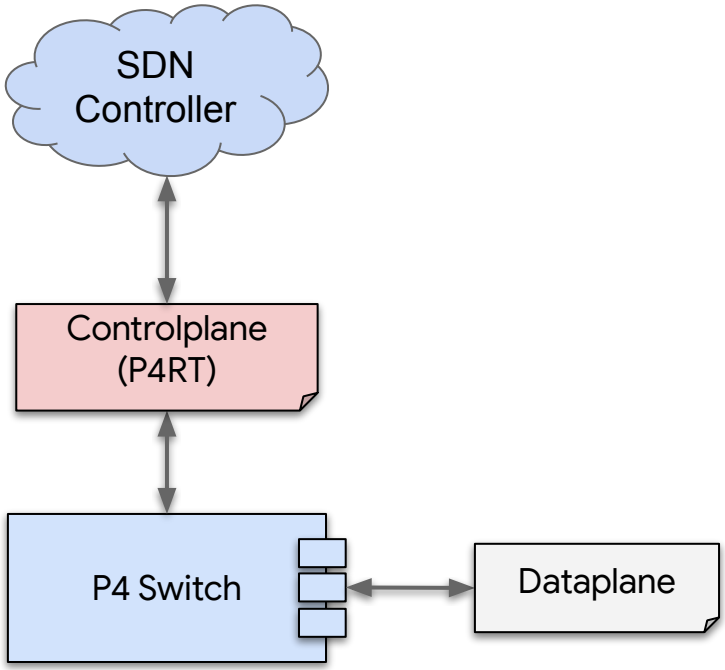


Why P4?



Model heterogeneous fleet in a single language

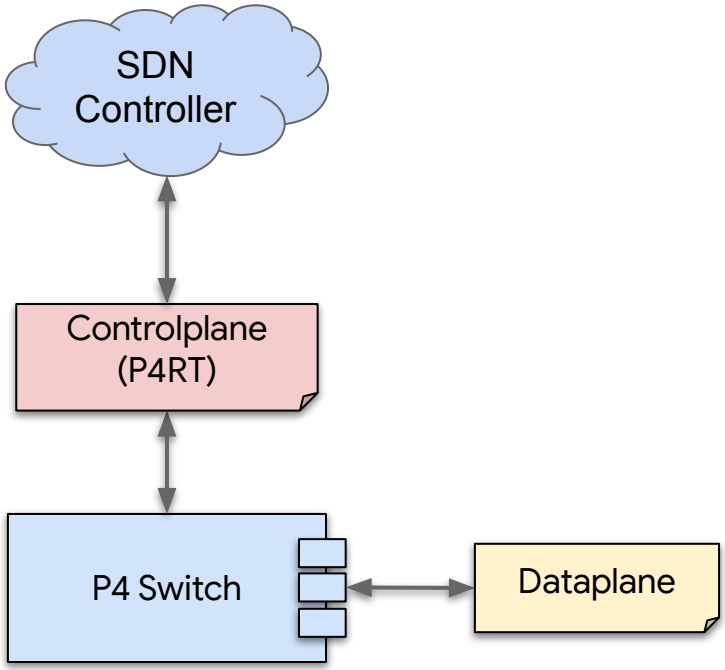
Why P4?



Model heterogeneous fleet in a single language

Clear interface between controller and switch

Why P4?

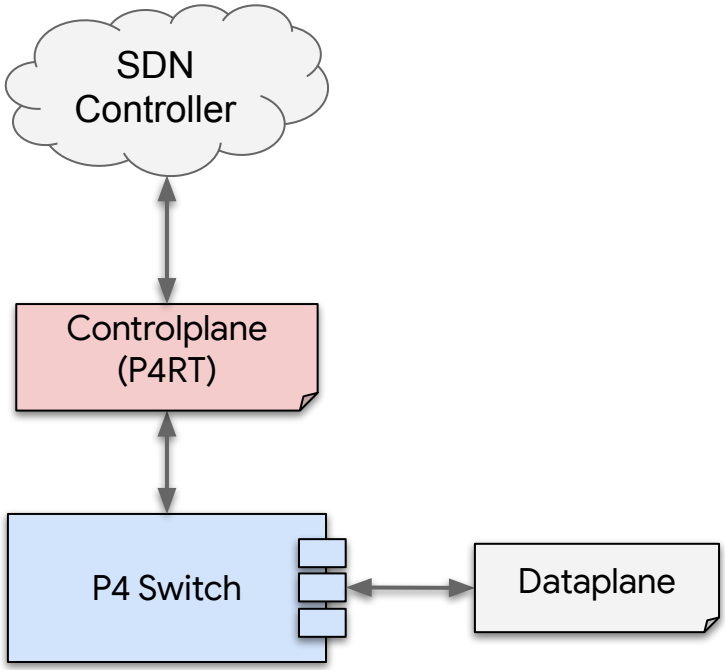


Model heterogeneous fleet in a single language

Clear interface between controller and switch

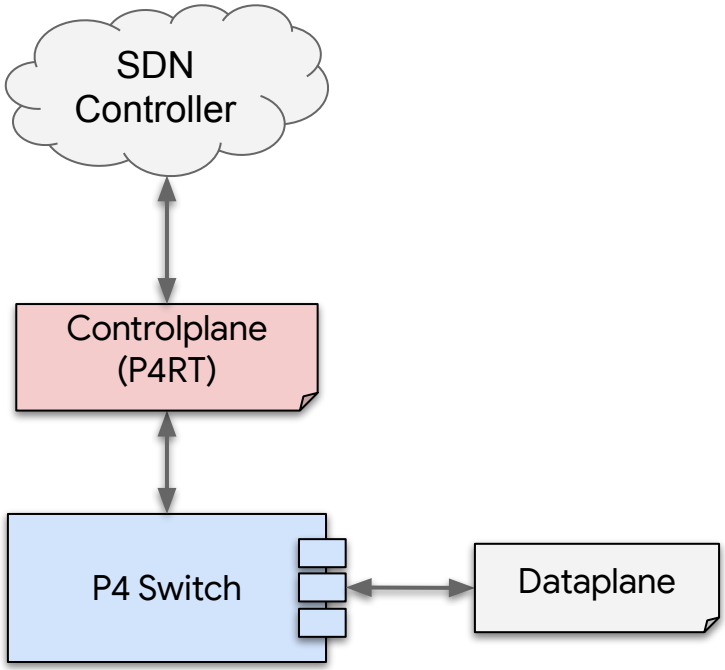
Precise specification of the switch behavior

Validation on the control plane side



Replay production table entries

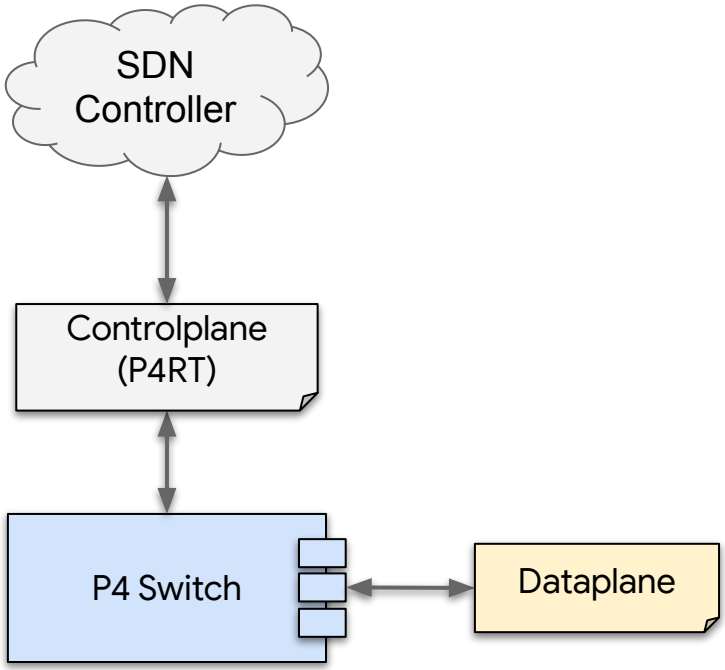
Validation on the control plane side



Replay production table entries

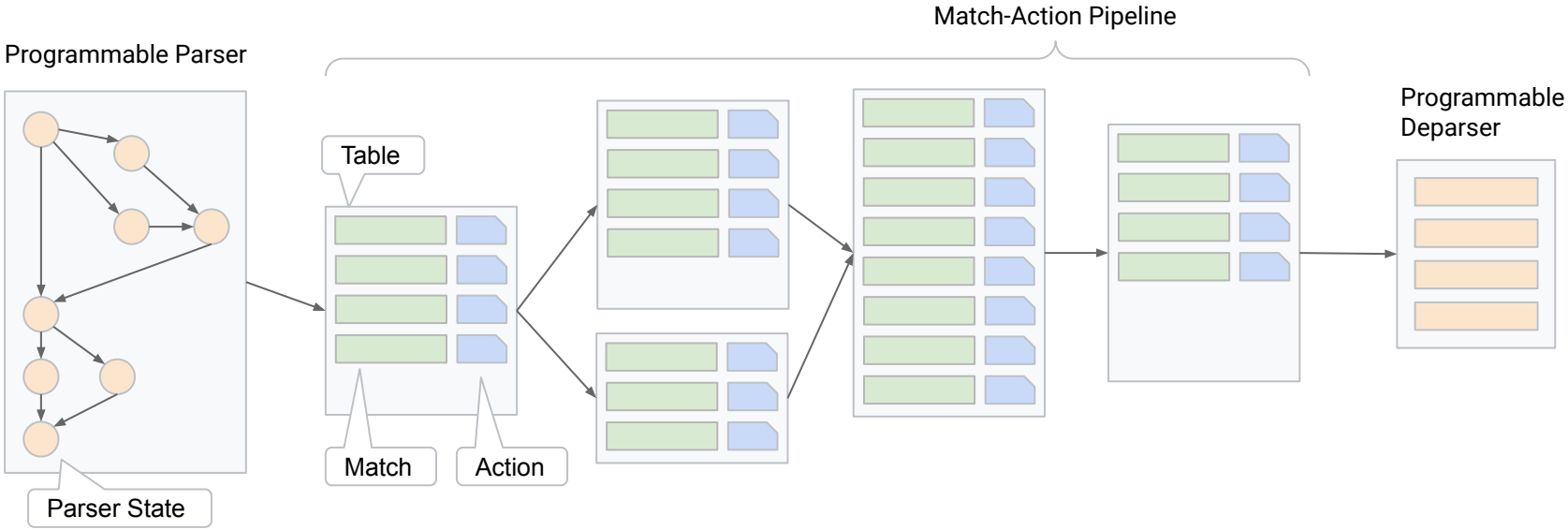
Fuzzer to randomly create table entry insert/delete requests

Validation on the data plane side

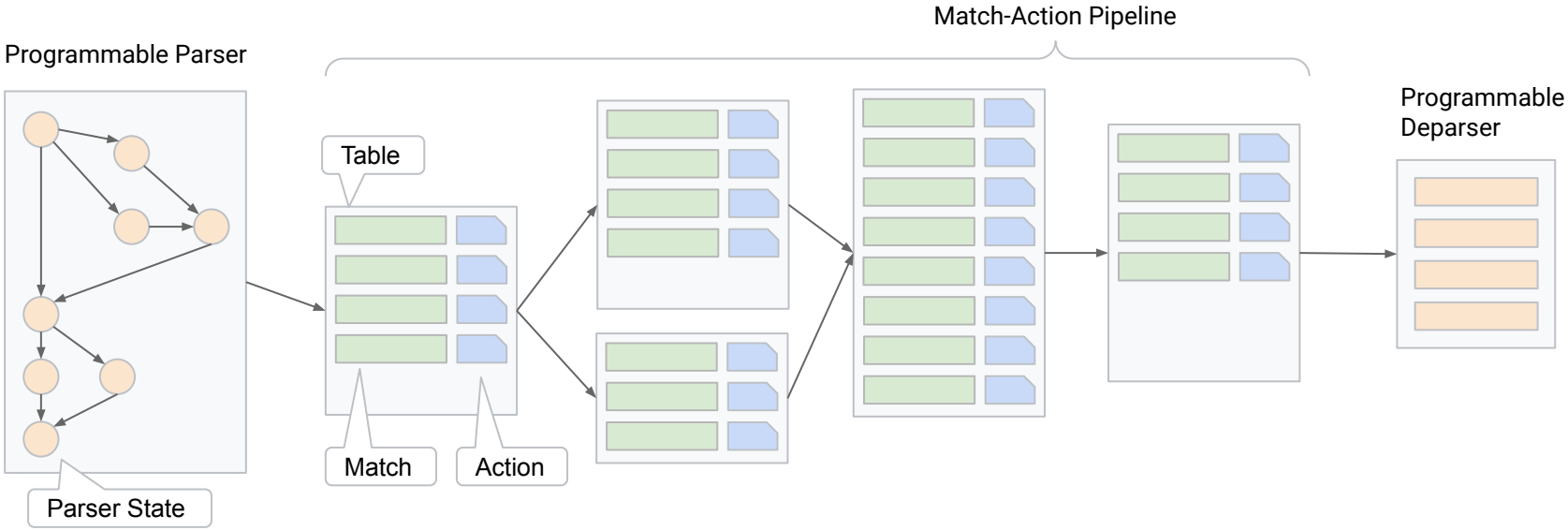


Dataplane testing requires packets that trigger all switch behaviors

Automatic Packet Generation

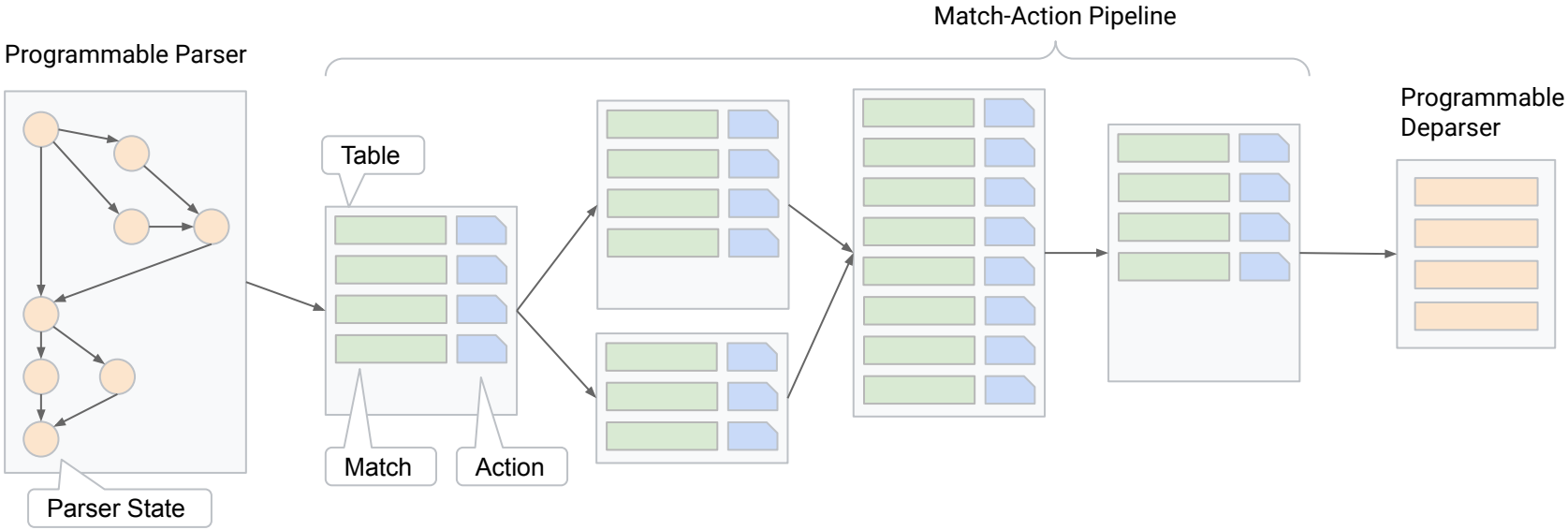


Automatic Packet Generation



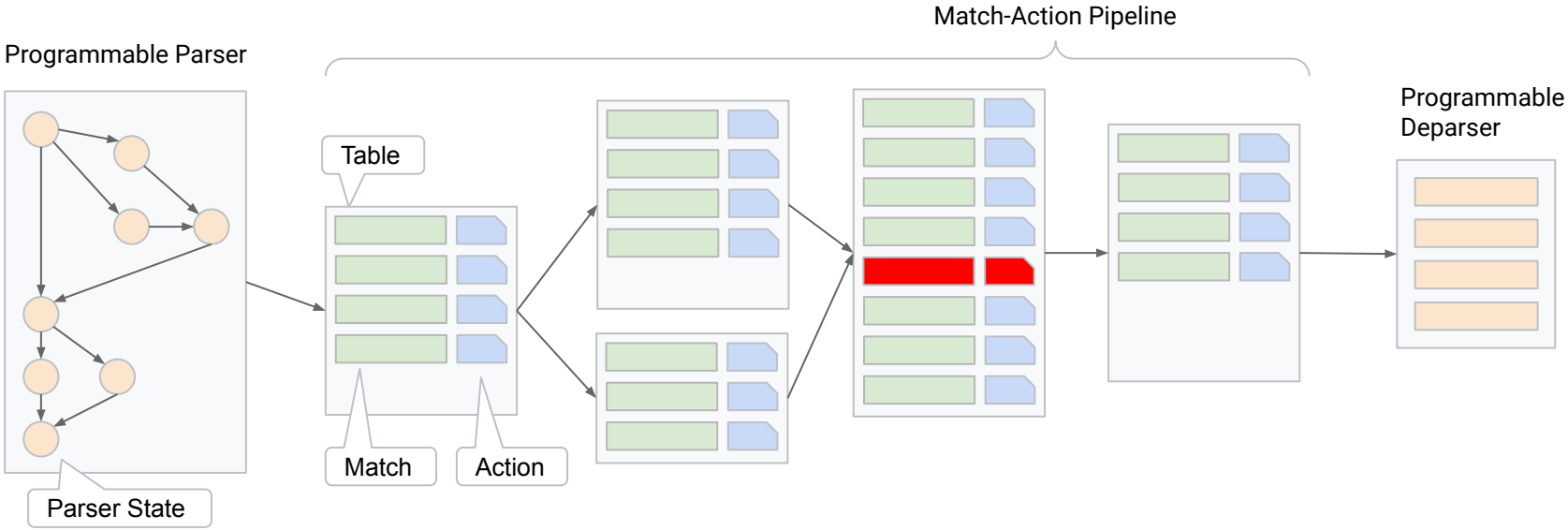
Execute P4 Program

Automatic Packet Generation



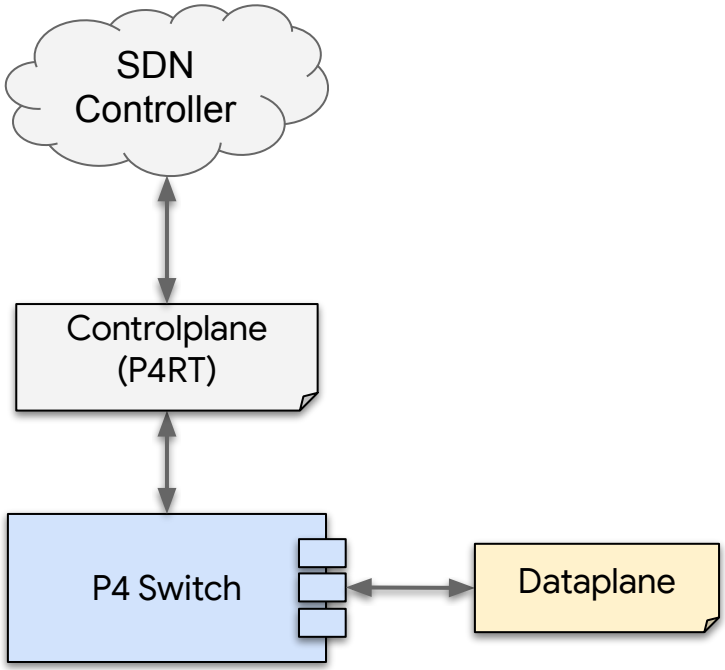
Reverse Execution Using Solver

Automatic Packet Generation



Reverse Execution Using Solver

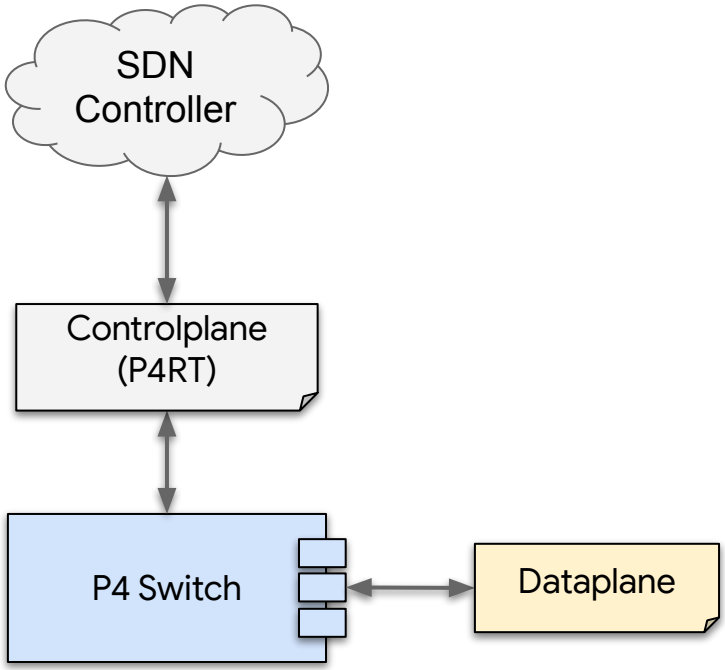
Validation on the data plane side



Dataplane testing requires packets that trigger all switch behaviors

- Automatically generate packet to hit every table entry

Validation on the data plane side



Dataplane testing requires packets that trigger all switch behaviors

- Automatically generate packet to hit every table entry

Test hashing, meters, counters

Validation Summary

All tests are parametric in the P4 program.

Updated P4 program or new switch can automatically re-validate all switches and controller software

P4 program is source of truth

Thank you

Stefan Heule <heule@google.com>

Google Cloud