State4: State-preserving Reconfiguration of P4-programmable Switches

Chenxing Ji
(Gabe)

TUDelft

Supervisor: Prof. Fernando Kuipers

Evolution Toward Programmable Data-plane

No Programmability

Control-plane
+
Data-plane

Traditional Switches

Programmability

Control-plane

Data-plane

SDN + fixed-
function data-plane

Programmability

Control-plane

Data-plane

SDN + Programmable
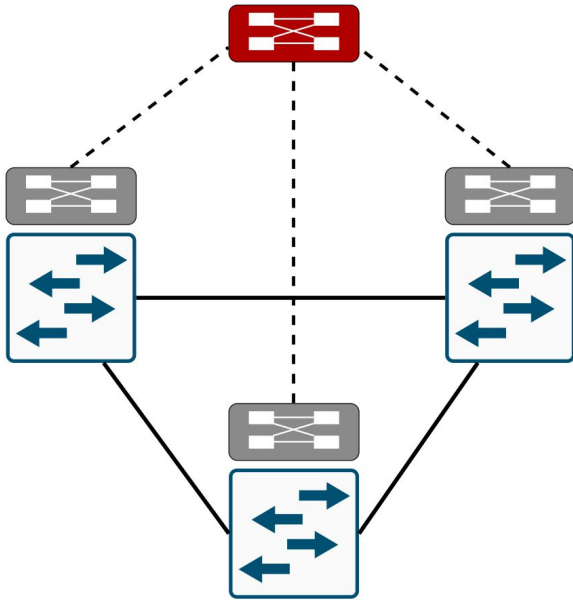Data-plane

TUDelft
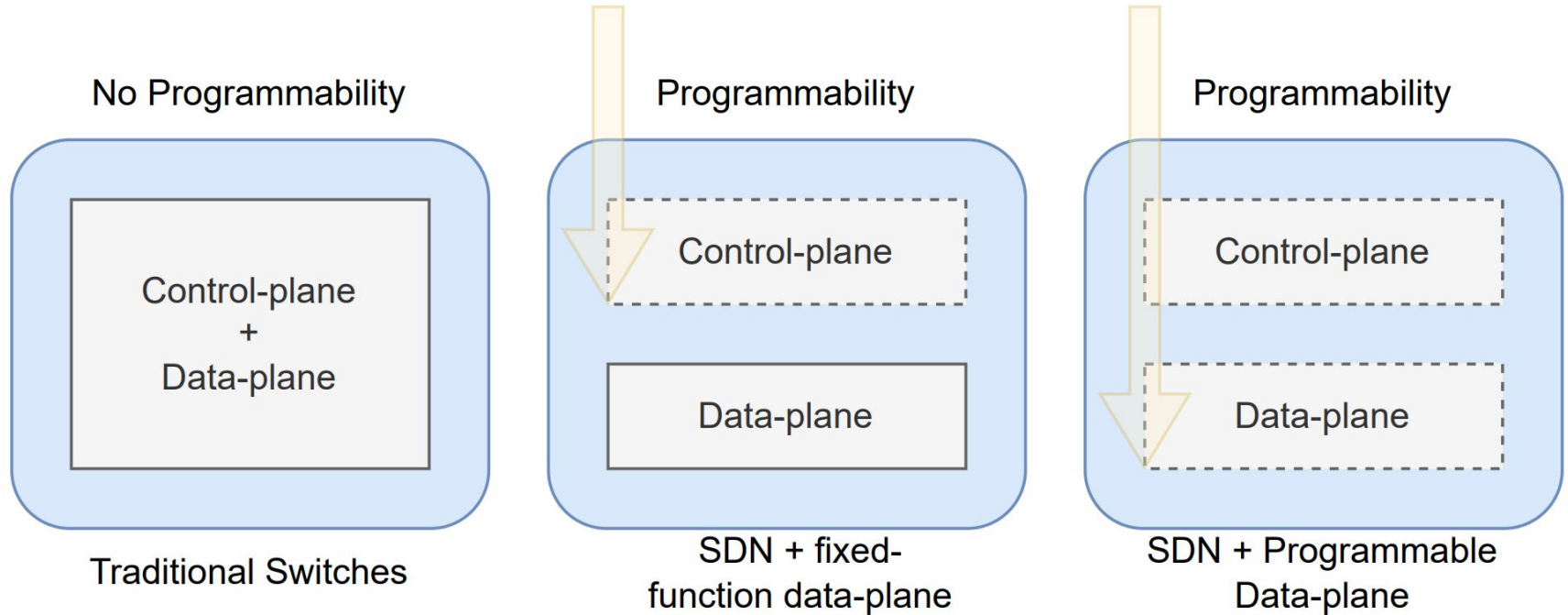
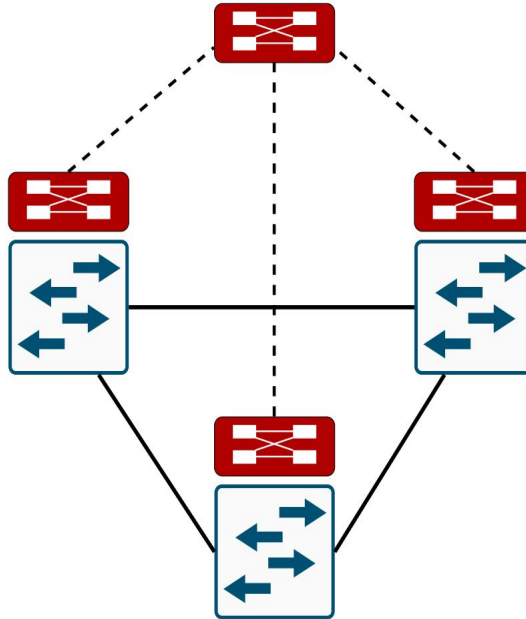# Programmable Data-plane

Benefits:

- Flexible Data-plane

- Improved Telemetry

- Offload Network Functions

SDN and Stateful Data-plane

Traditional Software Defined Networks

Programmable Data-plane + SDN

Advantages:

1. Reduced communication with centralized controller
2. Improved throughput

Disadvantages:

1. Hardware constraints limits complex stateful applications
2. Difficulties in locating state positions

TUDelft

Stateful Data-plane Examples:
Firewalls, Static-NAT, and Load Balancers

1. Stateful Firewalls:
   - Switch filters unrecognized inbound TCP connections
   - Data-plane stored the states of established connections
   - Application example: Port-Knocking Firewall [1]

2. Static-NAT:
   - Keeps track of state for established connections
   - Use stateful tables to establish connections

3. Load-Balancer:
   - HULA Stateful Load-Balancer [2]:
     Adopts ECMP and used local stateful information to determine the next best hop.
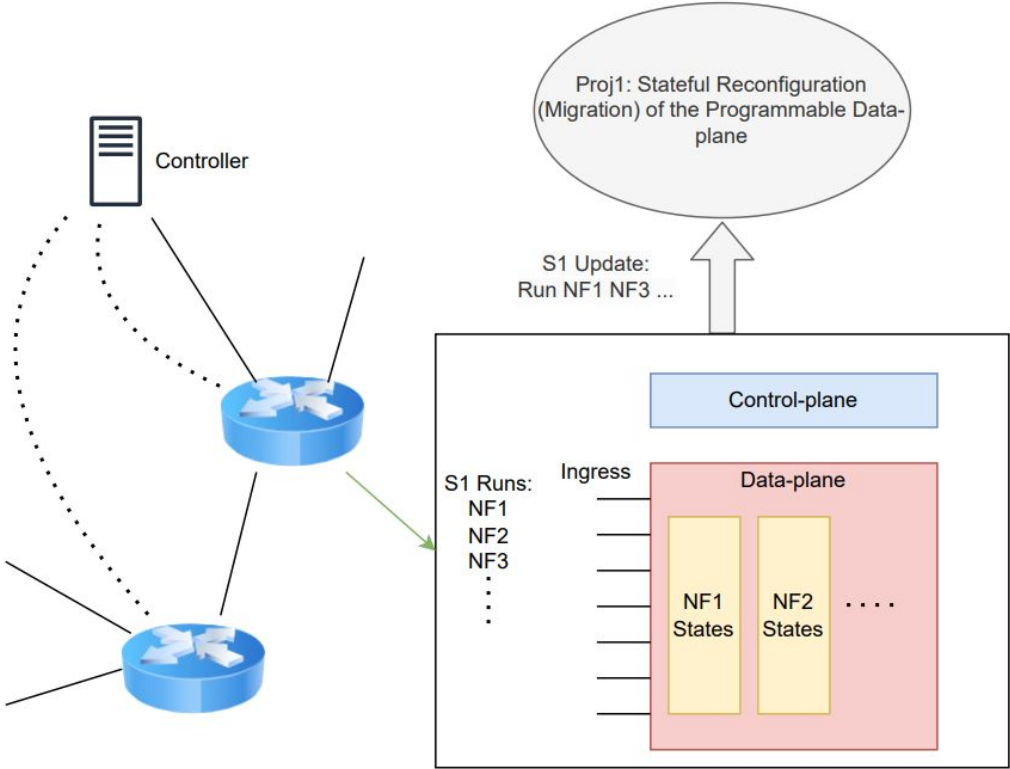
TUDelft

Reference:
[1] E. O. Zaballa, et al. "P4Knocking: Offloading host-based firewall functionalities to the network"
[2] Naga Katta et al. 2016. HULA: Scalable Load Balancing Using Programmable Data Planes

Motivation

State-loss problems:

- Flow loss

- Function inconsistency

State-preserving Switch Reconfiguration, State4

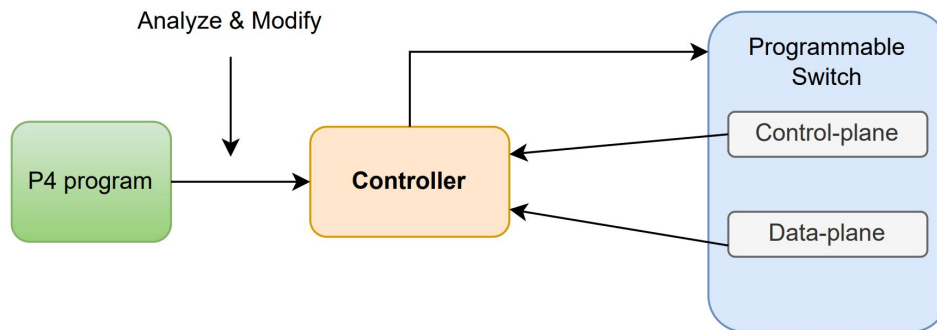Challenge -> Hard to maintain state:

- High data-rate: Tbps (Tofino)
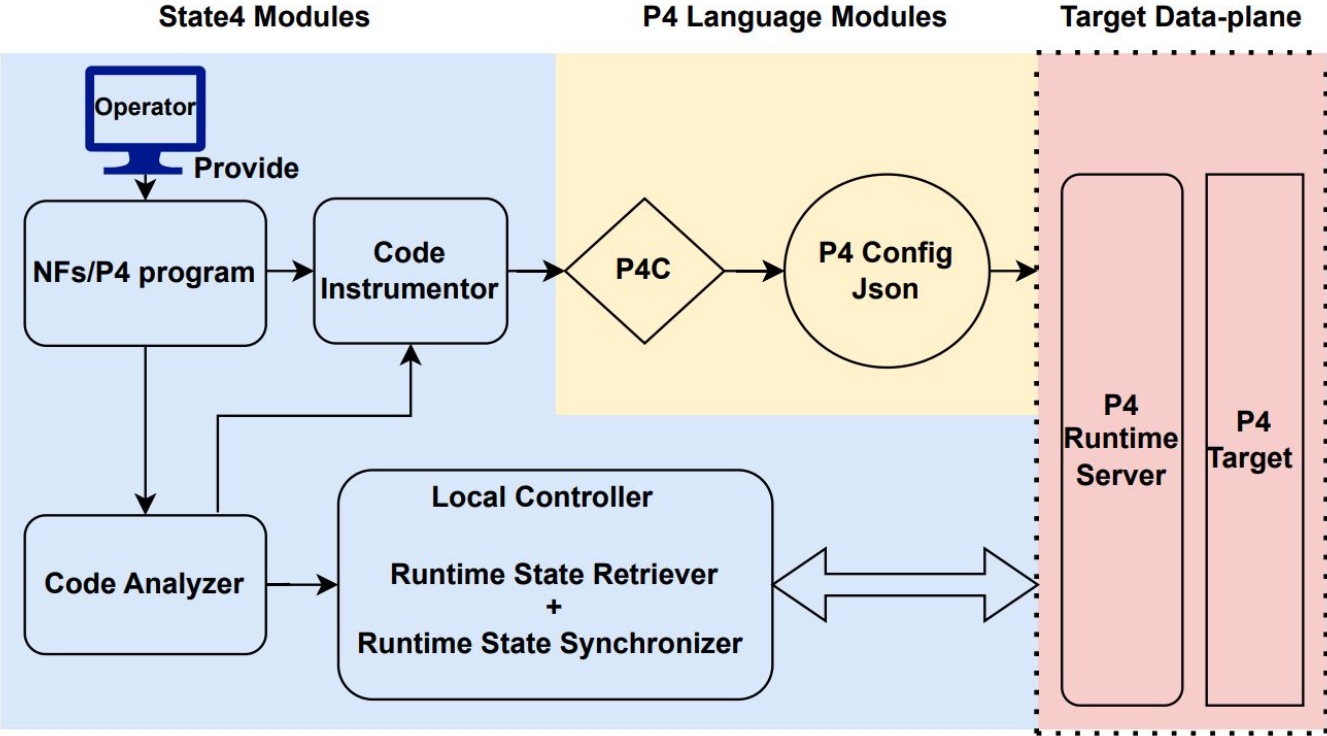
- Fast changing state

Approach:
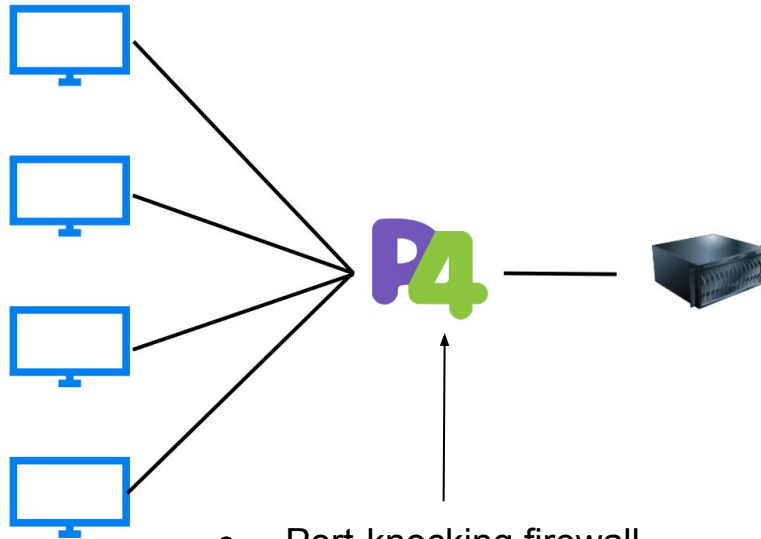Control-plane Pulling
+
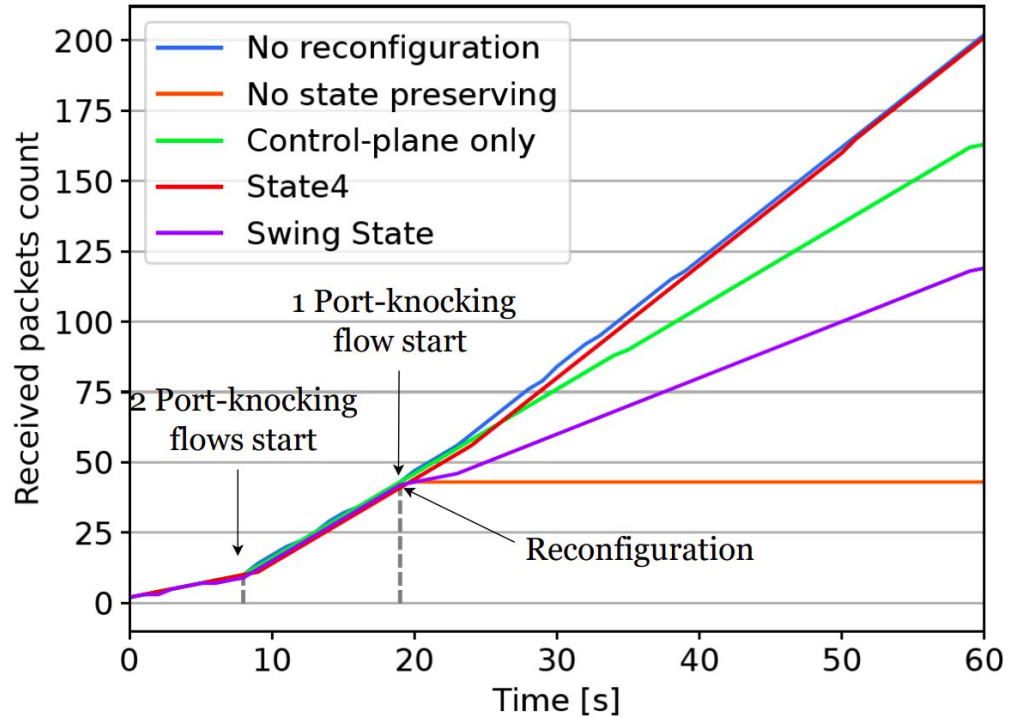Data-plane Cloning

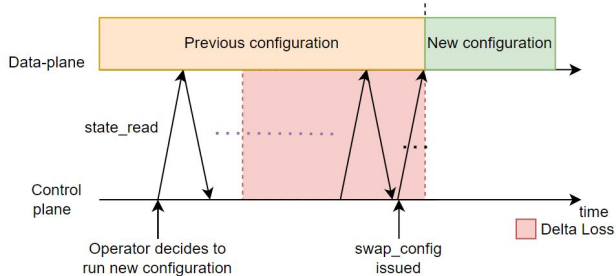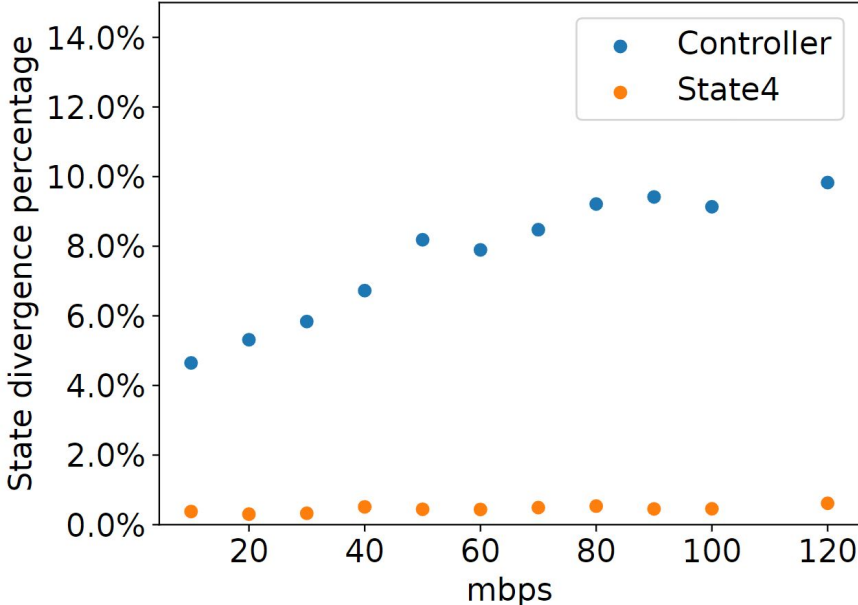# System Workflow

Results of State4

- Port-knocking firewall
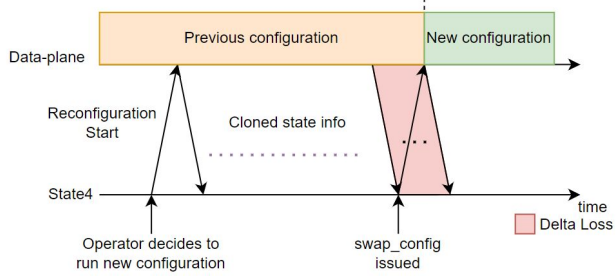- HULA load-balancer
- L2 forwarding

Swing State uses Data-plane cloning only

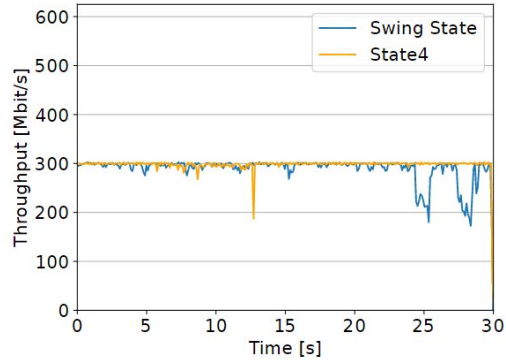# State4 Results continued - Part 1: Comparison with Controller

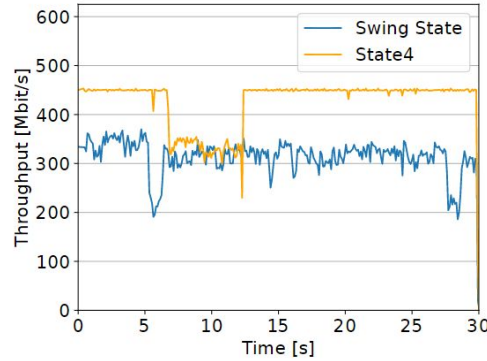(a) State loss in the controller approach.

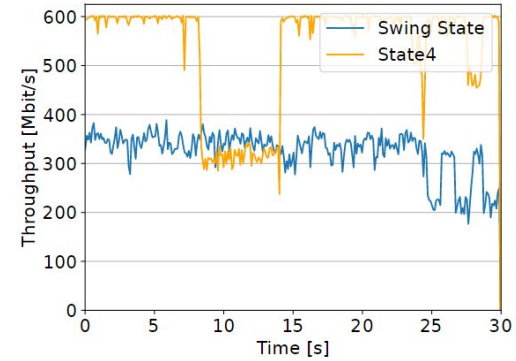(b) State loss in the State4 approach.

State4 Results continued - Part 2: Comparison to Swing State

(a) 300 Mbps throughput.

(b) 450 Mbps throughput.

(c) 600 Mbps throughput.

Swing States requires constant cloning to synchronize.

Compared to Swing State, State4 introduces less overhead, and impact the traffic less

Summary

Key Contributions:

1. State preserving reconfiguration framework.

2. Enabling **stateful switch reconfiguration** with minimal loss

3. Results show our approach incur less overhead compared to State-of-the-art

TUDelft

Questions?

Any questions? ;)

Thank you!