Taming Latency in Software Defined Networks

Keqiang He, Junaid Khalid, Sourav Das, Aaron Gember-Jacobson, Chaithan Prakash,
Aditya Akella, Li Erran Li, Marina Thottan

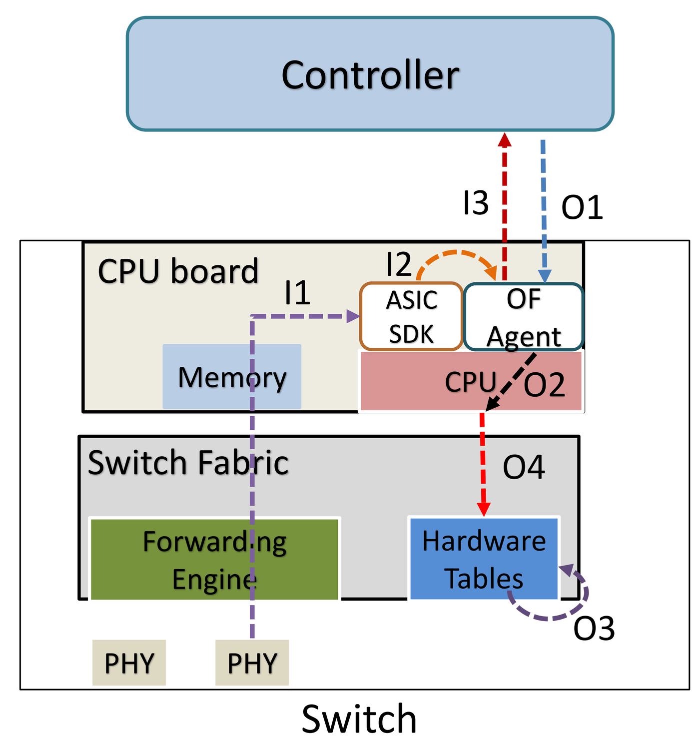
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Latency in SDN

Timely interaction between an SDN controller and switches is crucial to many applications like MicroTE, Fast Failover, Mobility, etc. These applications assume that the latency in interacting with the network switches is constant and negligible. However our measurement studies shows that this latency is significant. Moreover, it varies with the switch platforms, type of operations performed, table occupancy and concurrent operations on the switches.

Using grey-box probing, we narrow down the key factors for these latencies to be TCAM Organization, Low power switch CPU and software implementation inefficiencies. To overcome the latencies and achieve responsive control, we develop a systematic framework leveraging both the logically central view and global control in SDN, and the dissection of latencies from our measurement study.

Elements of Latency



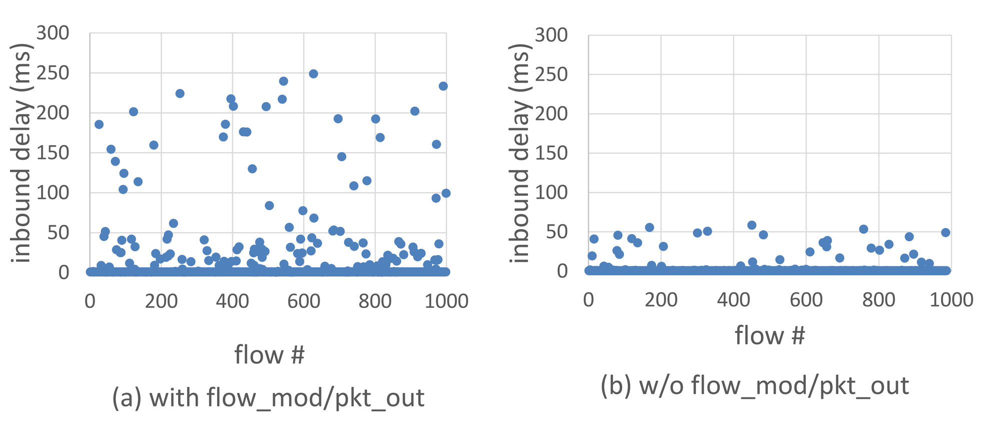
Inbound Latency
- **I1:** Send to ASIC SDK
- **I2:** Send to OF Agent
- **I3:** Send to Controller

Outbound Latency
- **O1:** Parse OF Message
- **O2:** Software schedules the rule
- **O3:** Reordering of rules in table
- **O4:** Rule is updated in table

Inbound Latency

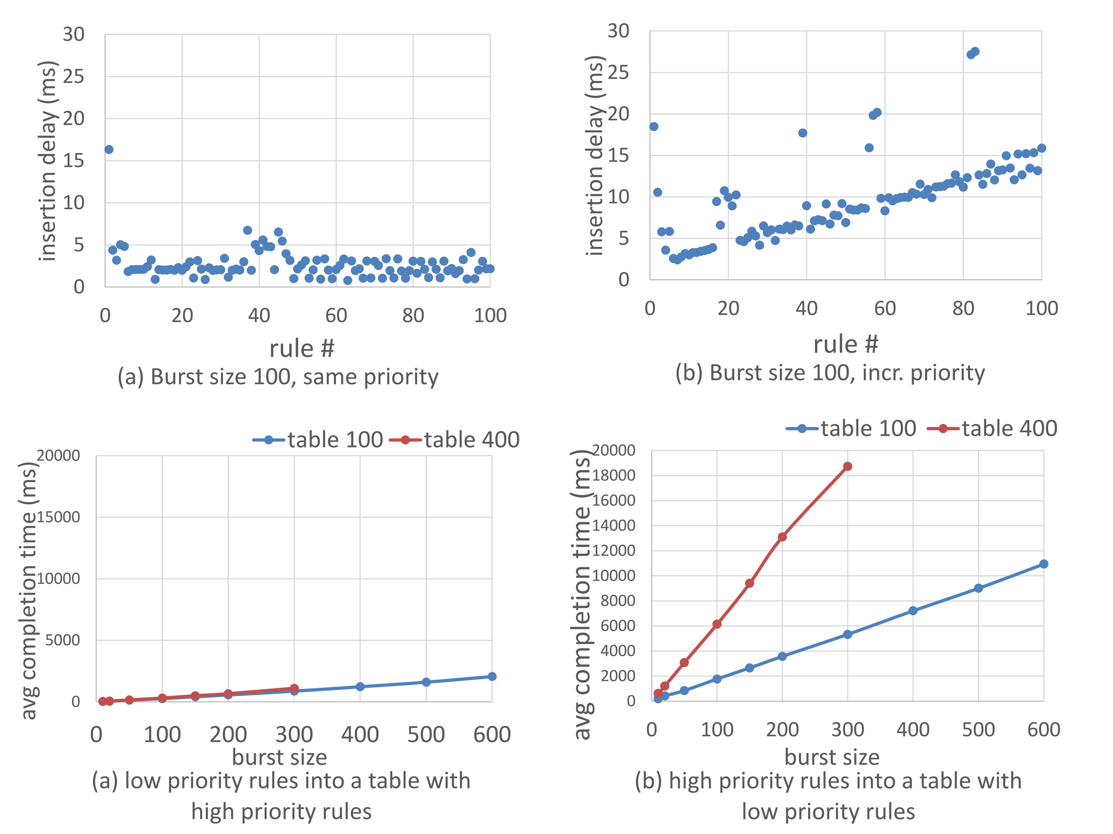
- Increases with flow arrival rate
- Increases with interference from outbound msgs

Flow Arrival Rate (packets/sec)	Mean Delay per packet_in (msec)
100	3.32
200	8.33



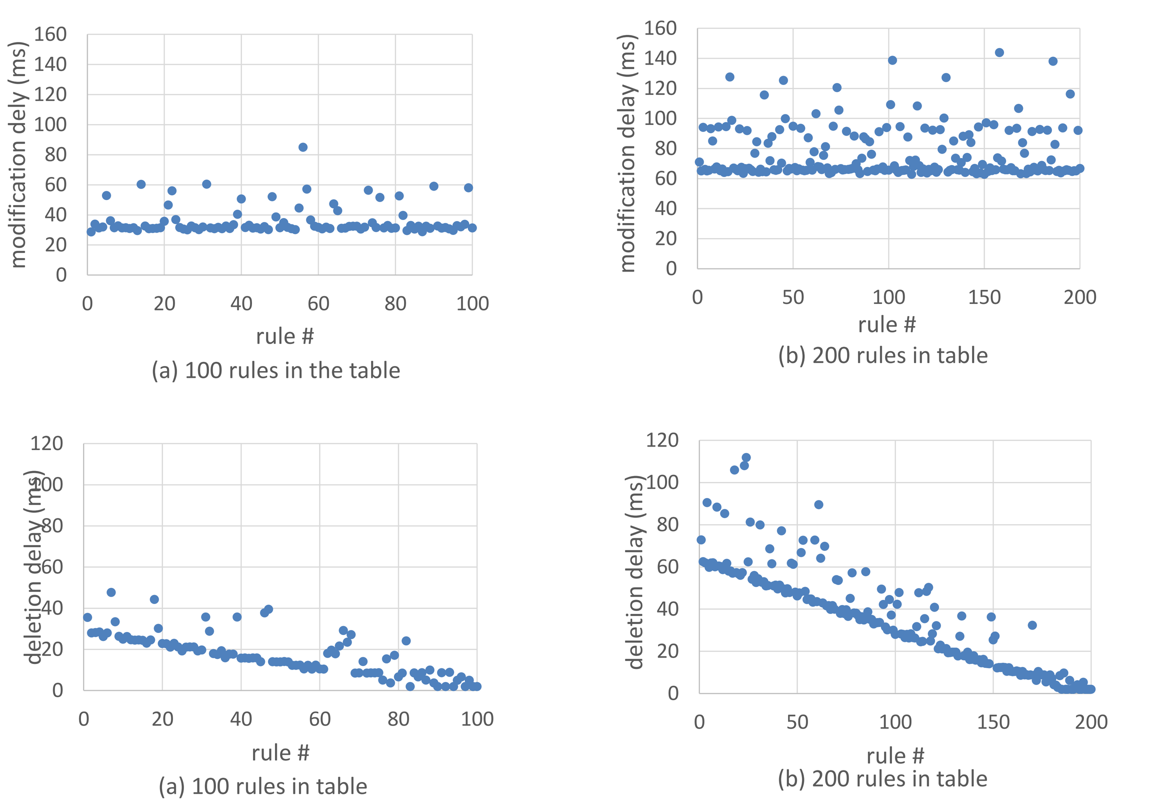
Outbound Latency - Insertion

- Affected by priority insertion patterns
- Affected by the table occupancy

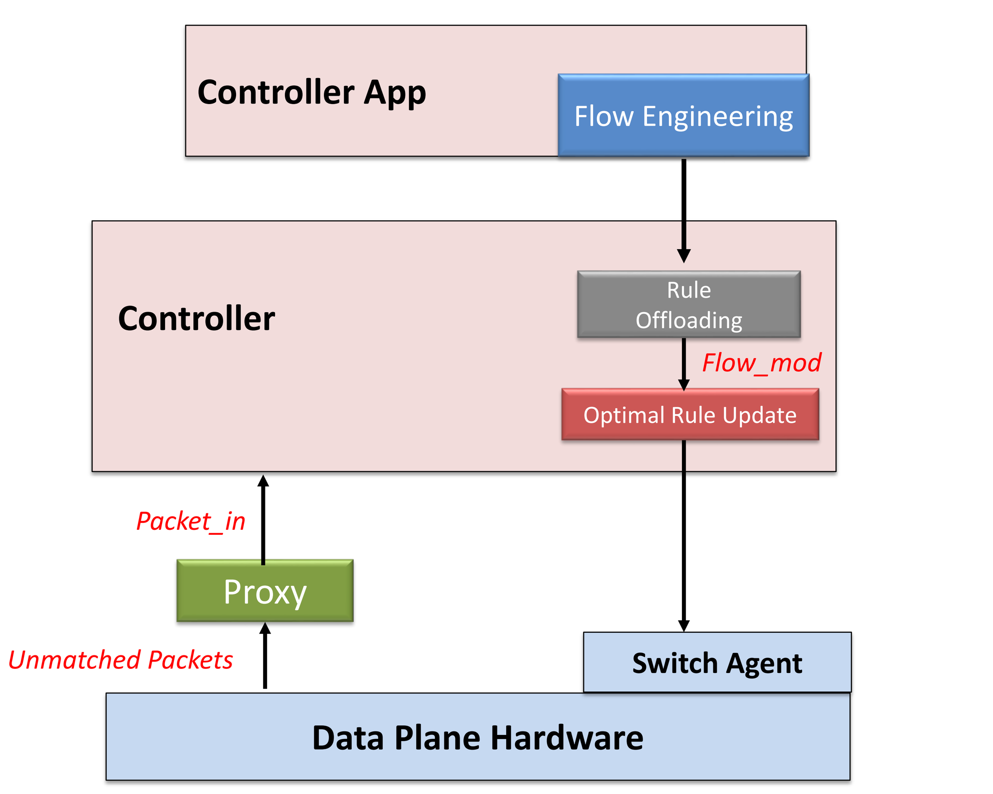


Outbound Latency – Modify/Delete

- Higher than Insertion latency
- Not affected by rule priority but affected by table occupancy

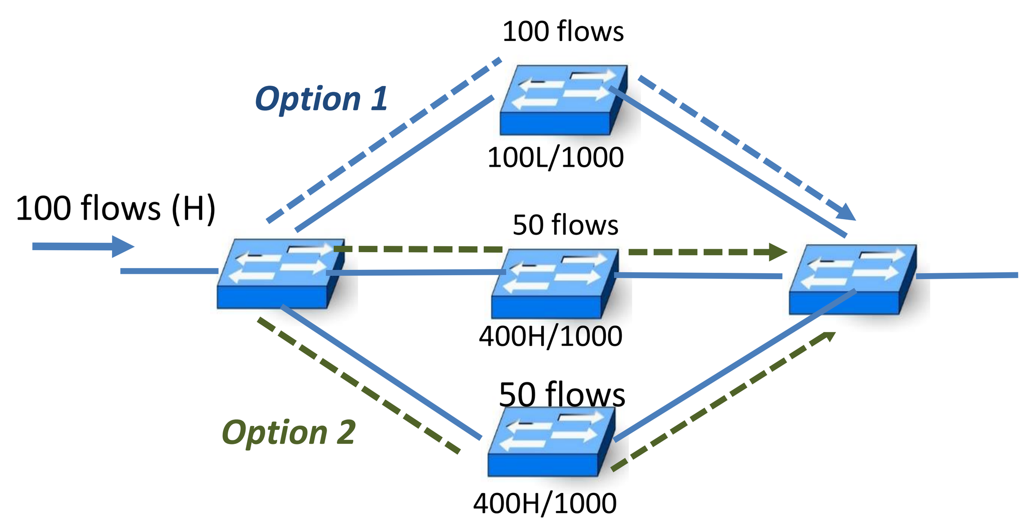


Taming Latency



Four Modules:
1. Proxy
2. Flow Engineering
3. Rule Offloading
4. Optimal Rule Update

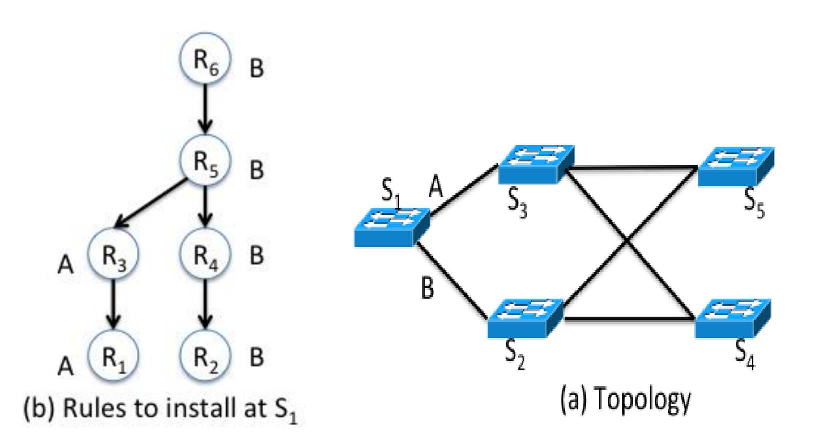
Flow Engineering



- Goal: Select paths across the network such that ***installation delay is minimized*** and *the network objective is satisfied*
- *Minimizes the aggregate impact* of both **rule displacement** in TCAM and **total number** of rules

Rule Offloading

- Networks with tunnels typically sees less *churn in forwarding state* in underlay network as compare to the end points
- Leverages this characteristic to *offload rules*
- Goal: Minimizes the installation latency by *offloading rules to* **underlay switches**



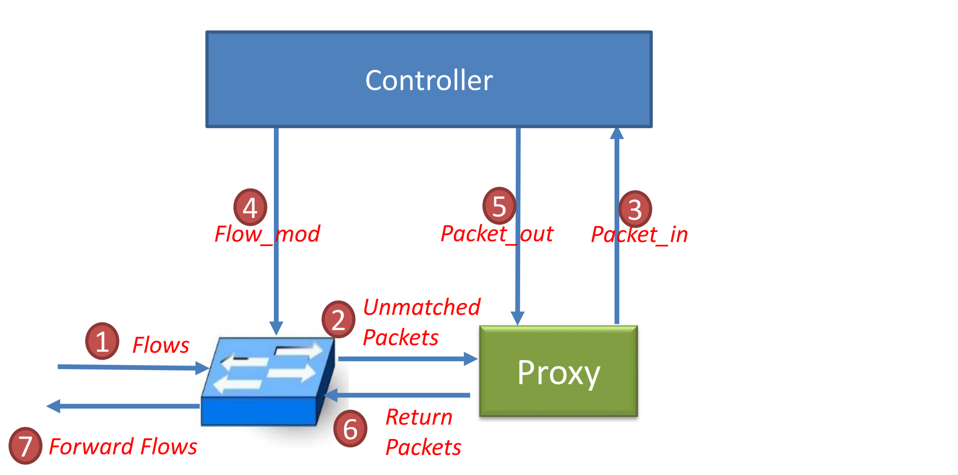
Switch	Rule
S1	R5
	R6
	RA
	RB
S3	R1
	R3
S2	R2
	R4

Optimal Rule Update

- Measurements show that *optimal order* of rule insertion *varies with switch platform*
- Goal: Control the actual rule insertion using the *pattern* that is *optimal* for the switch

Proxy

Goal: Physically decouple the switch's handling of *packet_in and packet_out* messages from *flow_mod*



Performance

Inbound

- Prototyped on a commodity host(Intel quad core, 2.66Ghz, 8GB RAM)
- Proxy almost completely eliminates the inbound delay

Flow Arrival Rate (per sec)	Delay w/o Proxy (msec)	Delay with Proxy (msec)
200	8	0.01
2000	-	0.02

Flow Arrival Rate (per sec)	99th percentile delay w/o Proxy (msec)	99th percentile delay with Proxy (msec)
200	192	0.07
2000	-	3.5

Outbound

- Simulated failover scenario in a tunneled WAN Network
- *Topology:* Full mesh with 25 nodes
- *Traffic matrix:* Assign a popularity index to each node
- *Table occupancy:* Assume switches have some pre-installed rules
- *Workloads:* 6 workloads which have different table occupancies and traffic volumes