

FOCUS: Function Offloading from a Controller to Utilize Switch Power

Ji Yang^{*+}, Zhenyu Zhou^{*}, Theophilus Benson^{*}, Xiaowei Yang^{*}, Xin Wu^{*}, Chengchen Hu⁺
^{*}Duke University, ⁺Xi'an Jiaotong University, ^{*}Big Switch Networks



Motivation

An SDN controller faces scalability challenges in a large network.

Existing Solutions

- Hardware optimization: inflexible.
- Distributed controller: control traffic overhead.
- Turn on legacy functions: loss of visibility and control.
- Execute arbitrary code in switches: heavyweight.

Our Approach

- Delegate local stable functions with a simple API.
- Example applications: ARP, LLDP and elephant flow detection.

Challenges

Global Visibility

- The controller should keep the identical visibility as before.

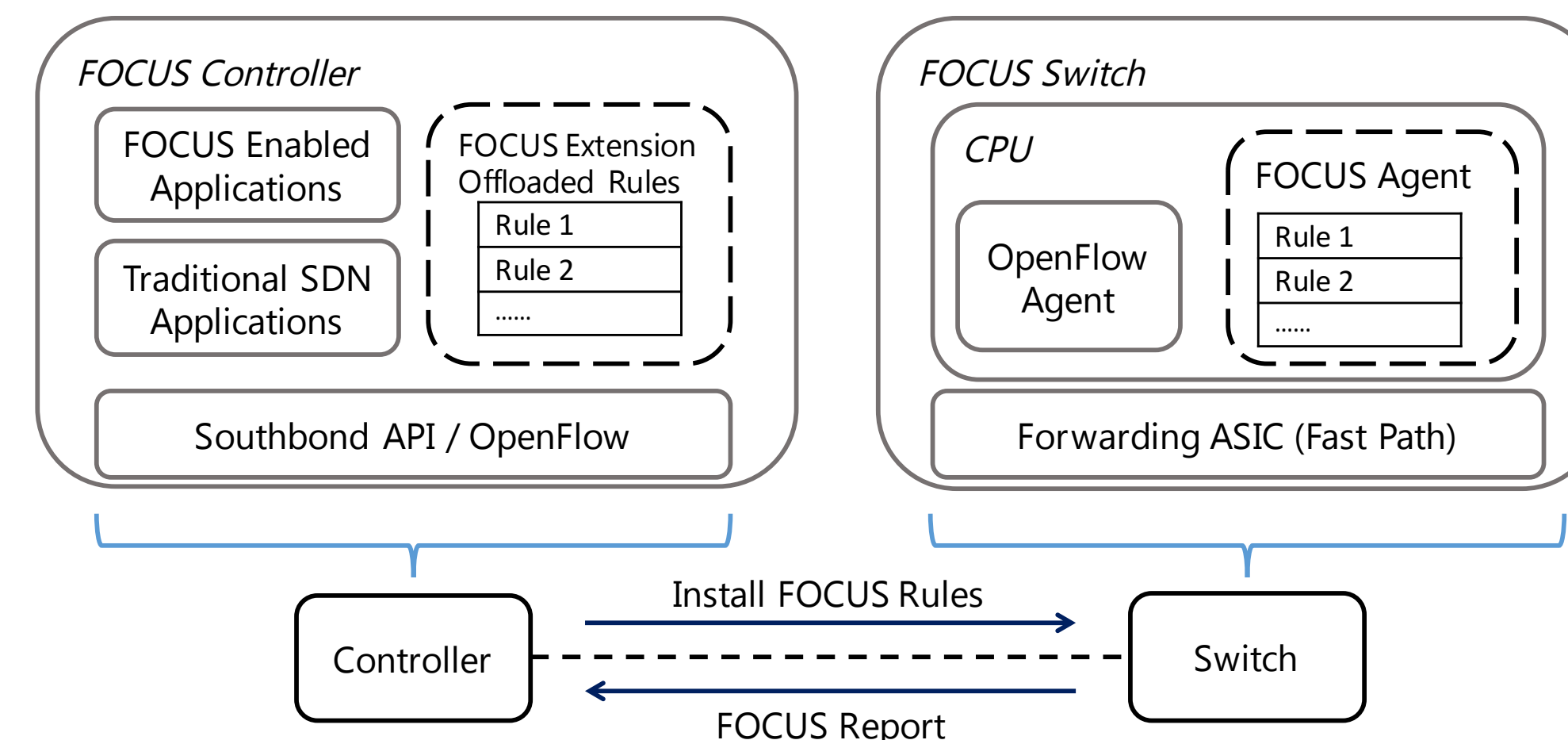
Local Decisions

- The abstraction should act solely on local information.

No Hardware Modifications

- Implement the solution using switch software.

Architecture



FOCUS Rules

Trigger	Action List	Timeout
---------	-------------	---------

Triggers

- Timer-based: for periodically polling and sending packets.
- Packet matching predicate: flexible TLV packet matching.

Action-List

- Packet operations: for accessing fields of the input packets.
- Flow entry operations: for accessing the flow table entries.
- Message operations: for communicating with the controller.

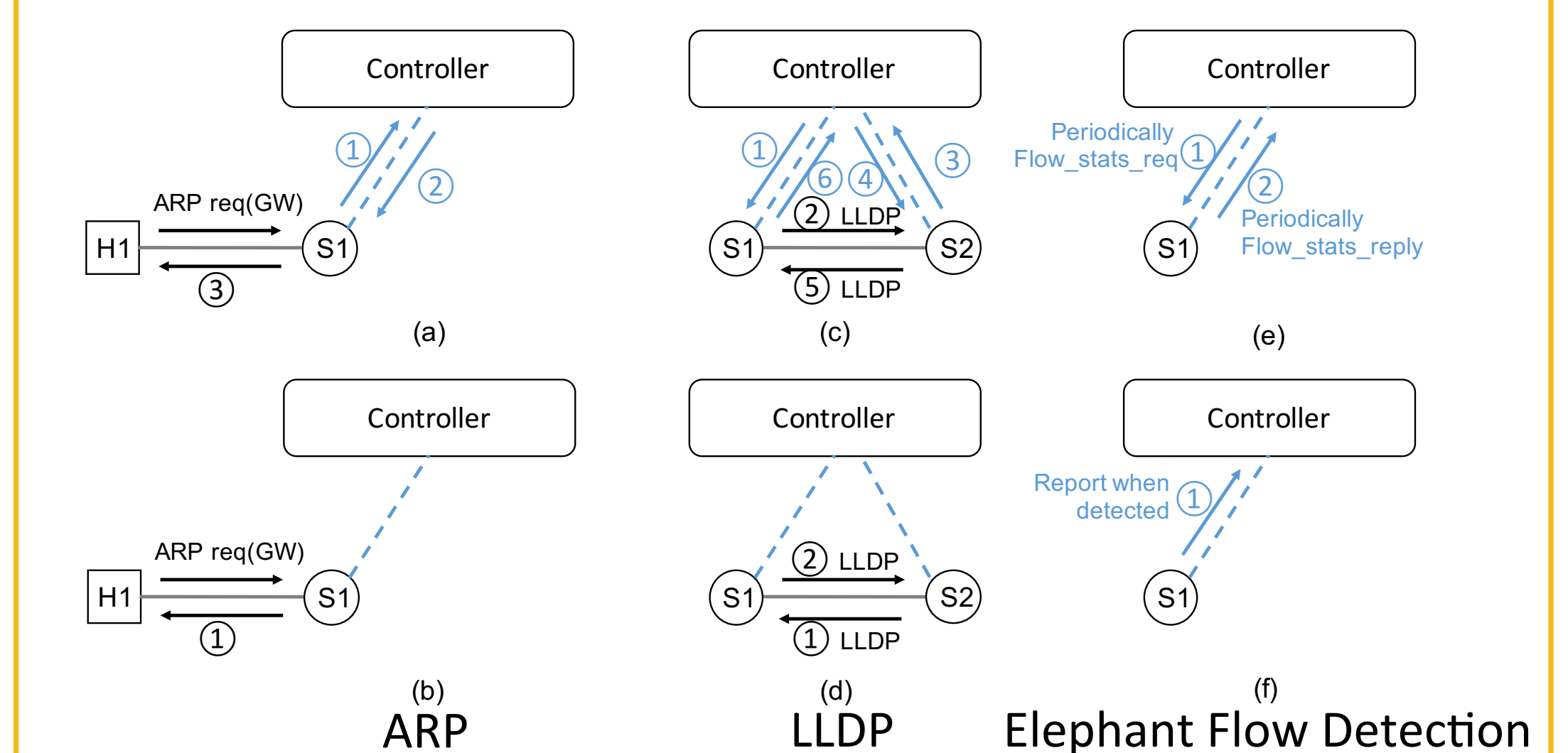
Timeout

- Informs the controller of whether a rule is still active.

Examples

Comparison of OpenFlow with FOCUS Workflow

- Host Discovery (ARP, ICMP for TTL expiration)
- Topology Maintenance (LLDP)
- Traffic Statistics (elephant flow detection)



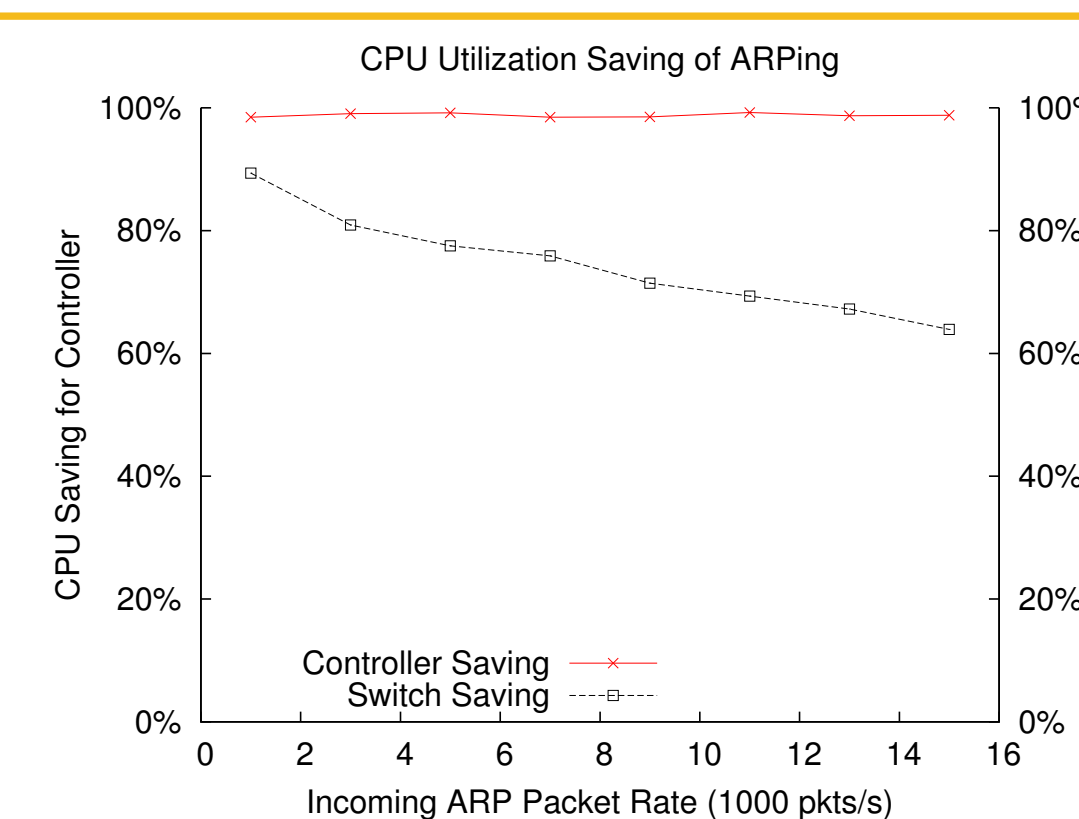
API Example (ARP Reply for Default Gateway)

Trigger	Actions
ARP target_IP=GW_IP	<code>pkt_compose(ARP)</code>
	<code>get_field(src_MAC)</code>
	<code>set_field(dst_MAC, ret¹)</code>
	<code>set_field(target_MAC, ret)</code>
	<code>get_field(src_IP)</code>
	<code>set_field(target_IP, ret)</code>
	<code>pkt_output(in_port)</code>

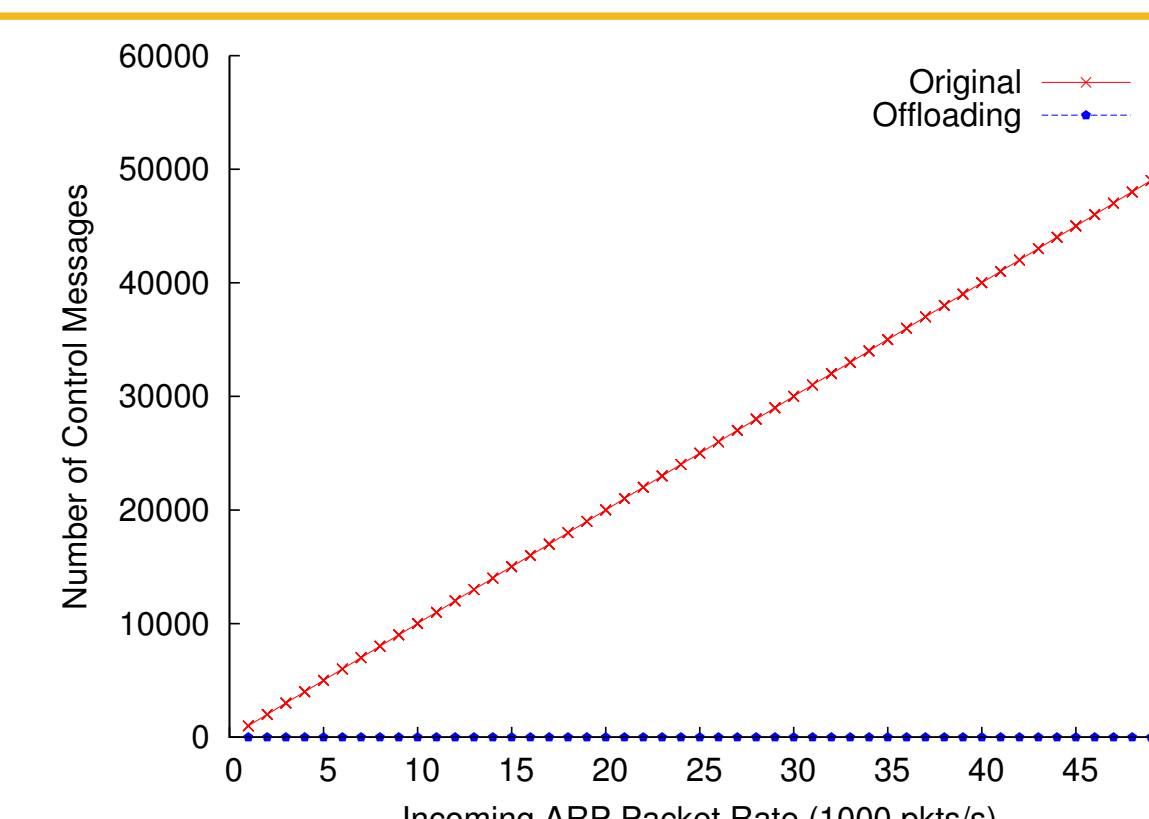
Evaluation

Performance Improvement for ARP

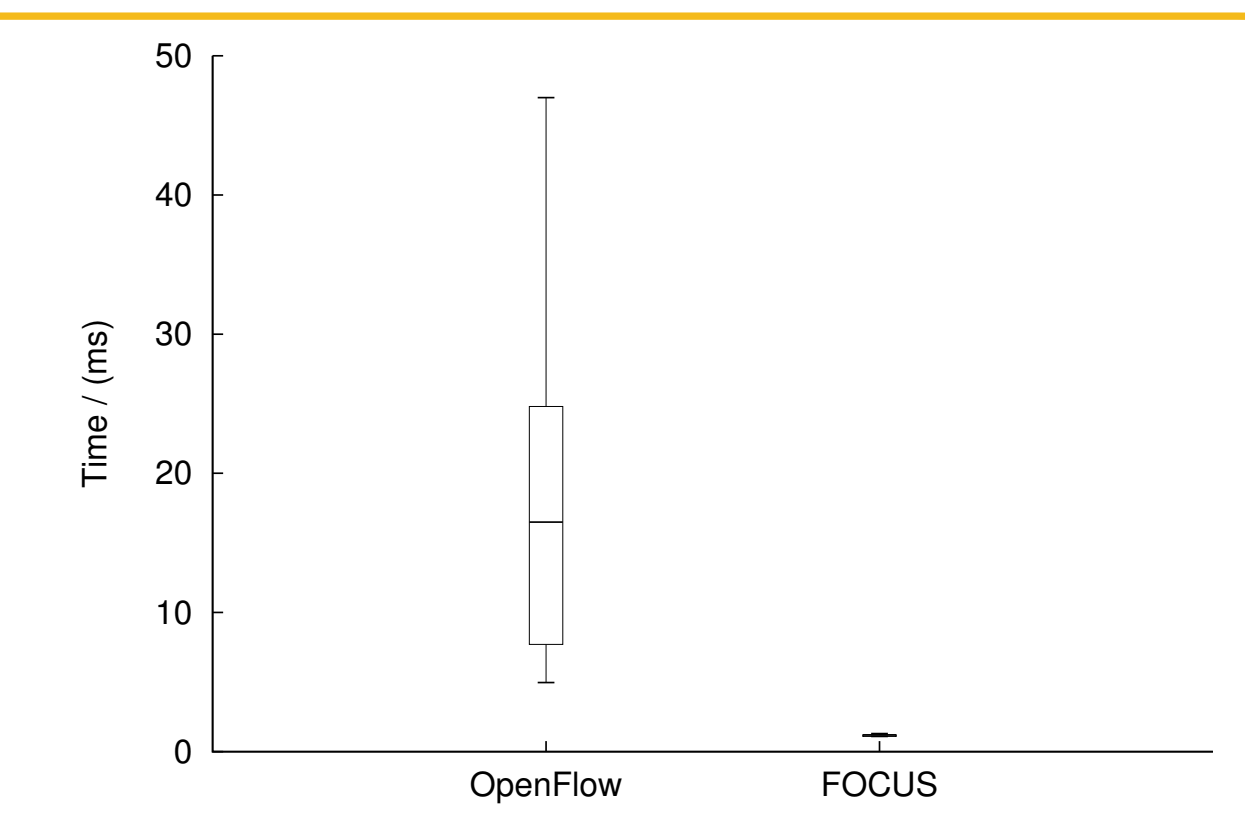
- Computational overhead is reduced by 80% – 98%
- Communication overhead is reduced by 50% – nearly 100%
- ARP response time is shortened by 18ms
- More results for other protocols can be found in our technical report:
cs.duke.edu/~zzy/file/focus-report-2016-001.pdf



Controller CPU Utilization (ARP)



Control Message (ARP)



Data Plane Response Time (ARP)