A traffic analysis use case for stateful SDN data plane

D. Moro, D. Sanvito, A. Capone
Politecnico di Milano

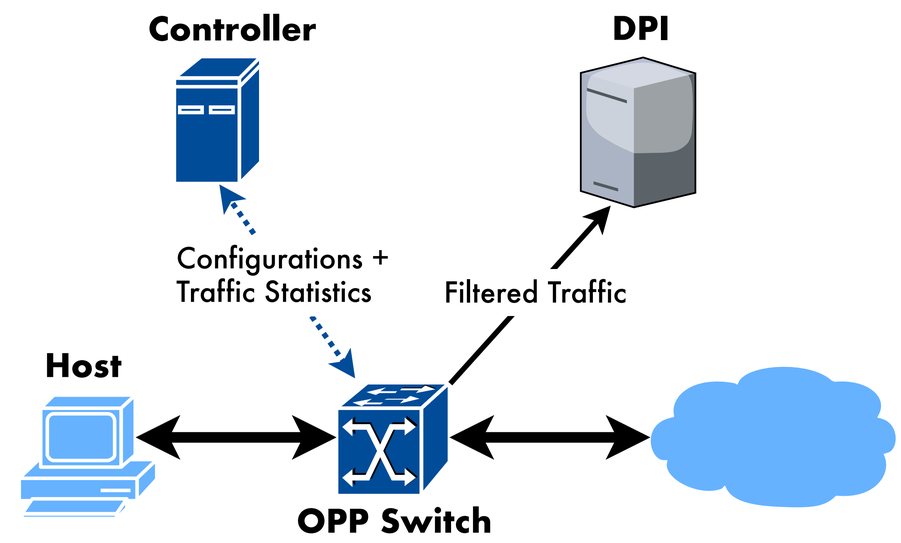
POLITECNICO MILANO 1863

Beba
BEhavioural BAsed forwarding
www.beba-project.eu

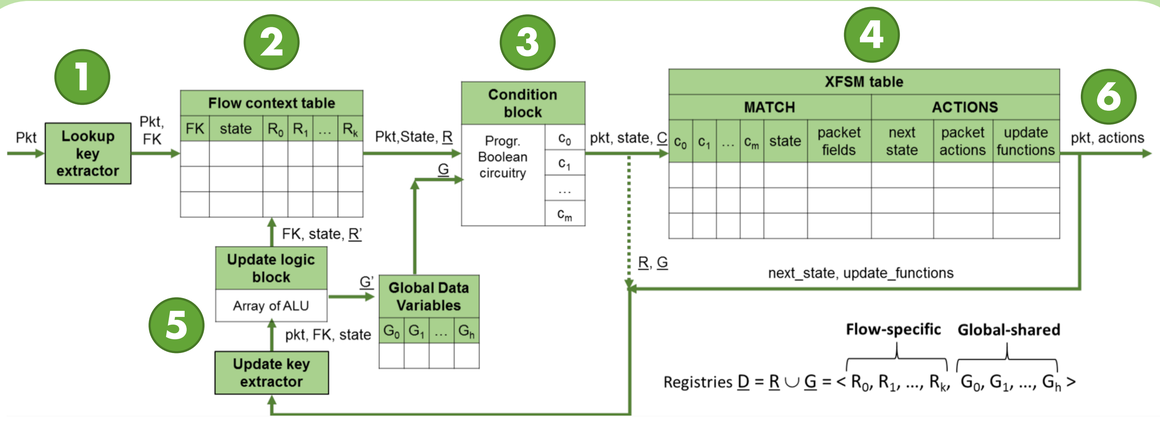
Introduction

Open Packet Processor (OPP) is a stateful OpenFlow extension that defines a data plane architecture for Software-Defined Networking, implementing an Extended Finite State Machine. It allows packets to be processed in a stateful fashion directly on the fast path: forwarding decisions are based on "flow-states" and "flow-registers", updated by the data plane itself as a consequence of packet-level events (i.e. table match) and timers. The poster presents an application that performs traffic filtering and statistics collection in order to offload software-based Deep Packet Inspection (DPI). The application provides a filtering scheme to reduce the quantity of traffic forwarded and analyzed by the DPI, demonstrating that 100% classification accuracy can be reached offloading more than 98% of traffic from the DPI itself.

Topology Example



Open Packet Processor Architecture



Steps:

1) A Flow Key (FK) is extracted from packet according to "lookup scope" header fields
2) Exact matching on FK to extract the "flow context" (state + registers)
3) Conditions on packet header and flow context are evaluated
4) Wildcard matching to extract the proper EFSM transition composed by next state, packet actions, registers update actions
5) Flow context table updated
6) Packet sent to the next stage in the pipeline

Flow Context table

Flow Key	State	Registers [R_0,R_1,…R_k]	Timeouts
A,B,w,z	1	[1,12,…,0]	Idle, hard, rollback state
…	…	…	
* (any)	0 (default)	[0,0,...0]	

Updated by packet actions or controller.

Flow Key: Exact match on the fields defined by the update and lookup scope; configured by the controller

State: All flows start with default state (last row). Entries are populated by means of set-state action or by the controller

Timeouts expiration: entry removed or state updated

EFSM Table

Updated by the controller
Extended OpenFlow Match+Action Table
Match: Packet fields + **State** + **Conditions result**
Actions: Packet actions + **Next State** + **Update Actions**
Timeouts expiration: entry removed

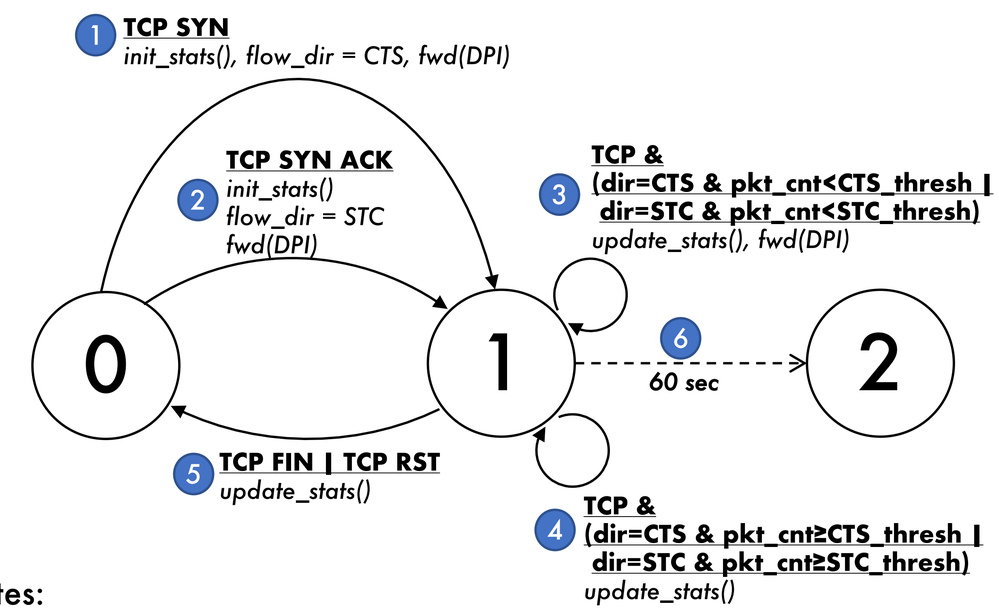
Application Example: Traffic Classification Offloading

Reducing the quantity of traffic forwarded and analyzed by a DPI can lead to 100% classification accuracy, with a great reduction of required computing power with respect to the analysis of the unfiltered traffic; however the lack of complete visibility on flows would prevent the DPI from computing traffic metrics for data analytics.

With this application we perform not only traffic filtering, but also statistics collection directly on the fast path of the network element.

Classification task is kept on the DPI, while traffic statistics is computed by the switch and collected by the controller.

TCP State Machine Example

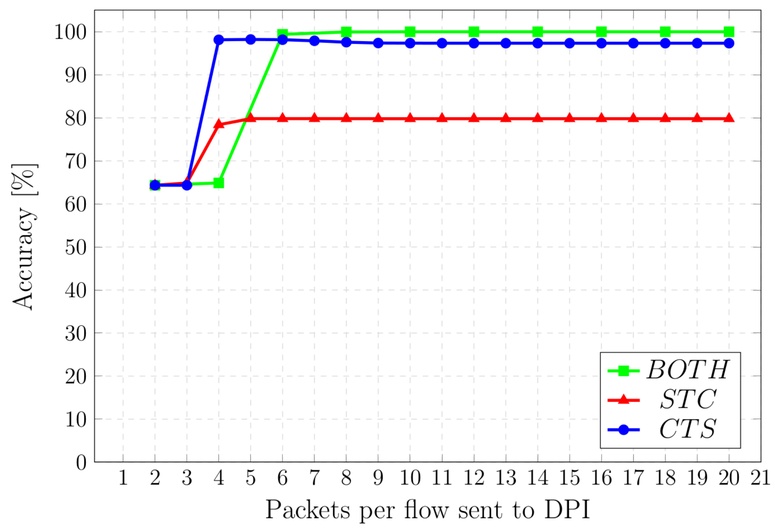


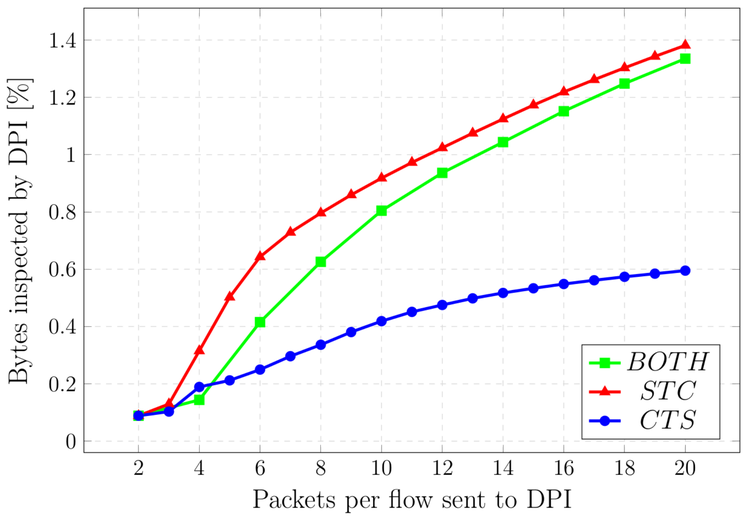
States:

0) Default state
1) Active flow state
2) "To Be Collected" state: controller will explicitly collect statistics after timeout expiration

Experimental Validation

Impact on classification accuracy of the filtering perfomed by the network element.
100% accuracy reached analyzing only 10 packets per flow





Impact of filtering on the percentage of packets analyzed by the DPI node.
10 packets per flow threshold leads to 0.8% traffic in terms of bytes analyzed by the DPI

- *BOTH*: filtering performed on both directions
- *STC*: filtering performed only on the Server to Client direction + 1 pkt on the other direction
- *CTS*: filtering performed only on the Client to Server direction + 1 pkt on the other direction

References

[1] G. Bianchi, M. Bonola, S. Pontarelli, D. Sanvito, A. Capone, C. Cascone, "Open packet processor: A programmable architecture for wire speed platform-independent stateful in-network processing" 2016.

[2] D. Sanvito, D. Moro, A. Capone, "Towards traffic classification offloading to stateful SDN data planes", NEAF-IO 2017, Bologna July 2017.