
# Towards traffic classification offloading to stateful SDN data planes

**Davide Sanvito, Daniele Moro, Antonio Capone**
Dipartimento di Elettronica, Informatica e Bioningegneria, Politecnico di Milano, Italy

Bologna, 3 Luglio 2017
NEAF-IO - Workshop on NEtwork Accelerated FunctIOns

# HTTPS and Encryption



- Encrypted traffic is growing
    - North America: ~40%
    - Europe: >60%
- HTTPS
    - from sensitive transactions to HTTPS «everywhere»
- Impact of Encryption on DPIs
    - Payload not inspectable, only headers and in clear text

HTTPS:// EVERYWHERE

Let's Encrypt

Free SSL For All

https://www.yourdomain.com

# Motivation



- Propose a solution to optimize DPI (or other classifier) usage from the network
- Classify the same amount of traffic with less computational resources
- No direct classification into the network element

# Solution - Idea



**Scheme to filter and collect statistics directly on the data plane of a SDN network**

- Standard OpenFlow Implementation possible but not scalable
- Stateful data plane (Open Packet Processor) is the right option to program a stateful application within the datapath with no overload on the controller

# Related works



- Several works evaluated the effect of traffic sampling on classification accuracy ([1] [2] [3])
- Most optimization operate on the DPI implementation itself
- [4] propose a stateful SDN approach (based on OpenState [5]) to make offloading. Their approach is not completely decoupled from the controller

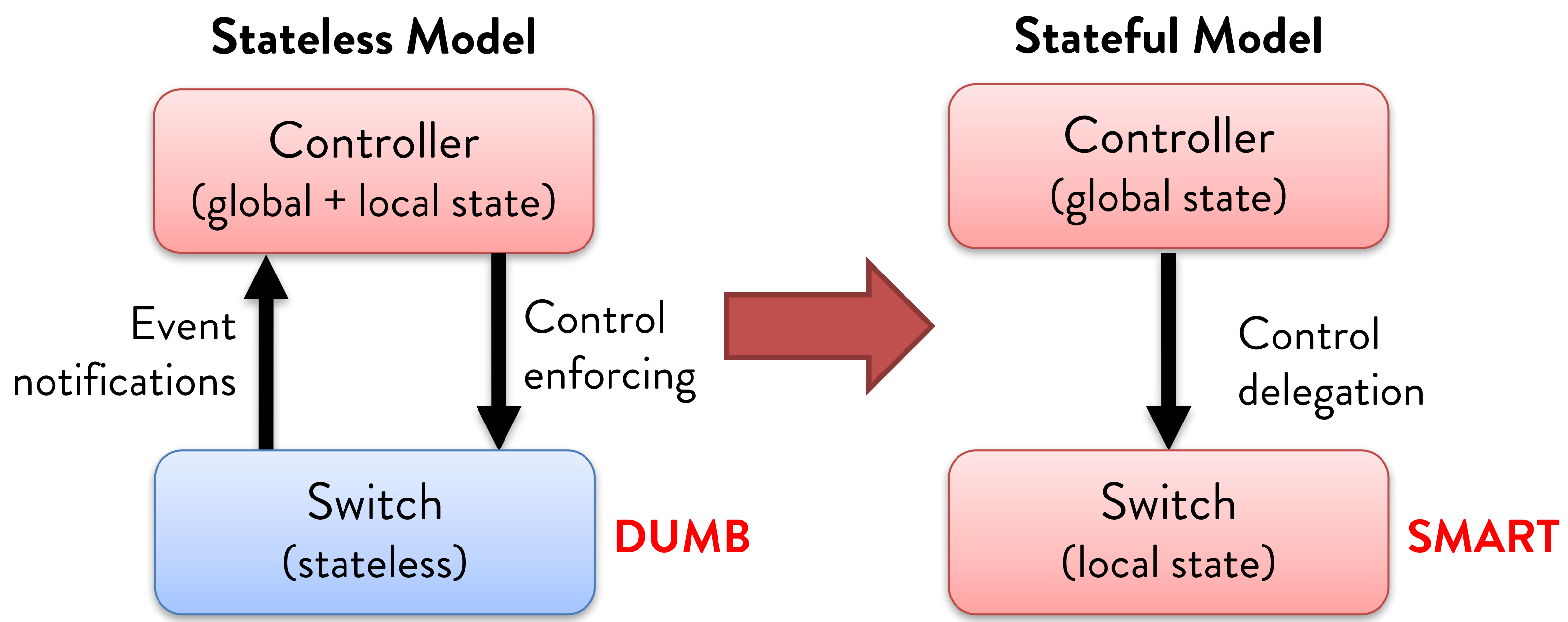
**We propose an offloading mechanism independent from the classifier (that can be DPI or Machine Learning -based) and completely decoupled from the controller**

[1] S. Fernandes *et al.* "Slimming down deep packet inspection systems." *INFOCOM Workshops 2009, IEEE*. IEEE, 2009.
[2] N. Cascarano *et al.* "Improving cost and accuracy of DPI traffic classifiers." *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010.
[3] L. Bernaille *et al.*"Early application identification." *Proceedings of the 2006 ACM CoNEXT conference*. ACM, 2006.
[4] T. Zhang *et al.* "On-the-fly Traffic Classification and Control with a Stateful SDN approach."
[5] G. Bianchi *et al.* "OpenState: programming platform-independent stateful openflow applications inside the switch." ACM SIGCOMM Computer Communication Review 44.2 (2014): 44-51.

# Stateful Data Plane



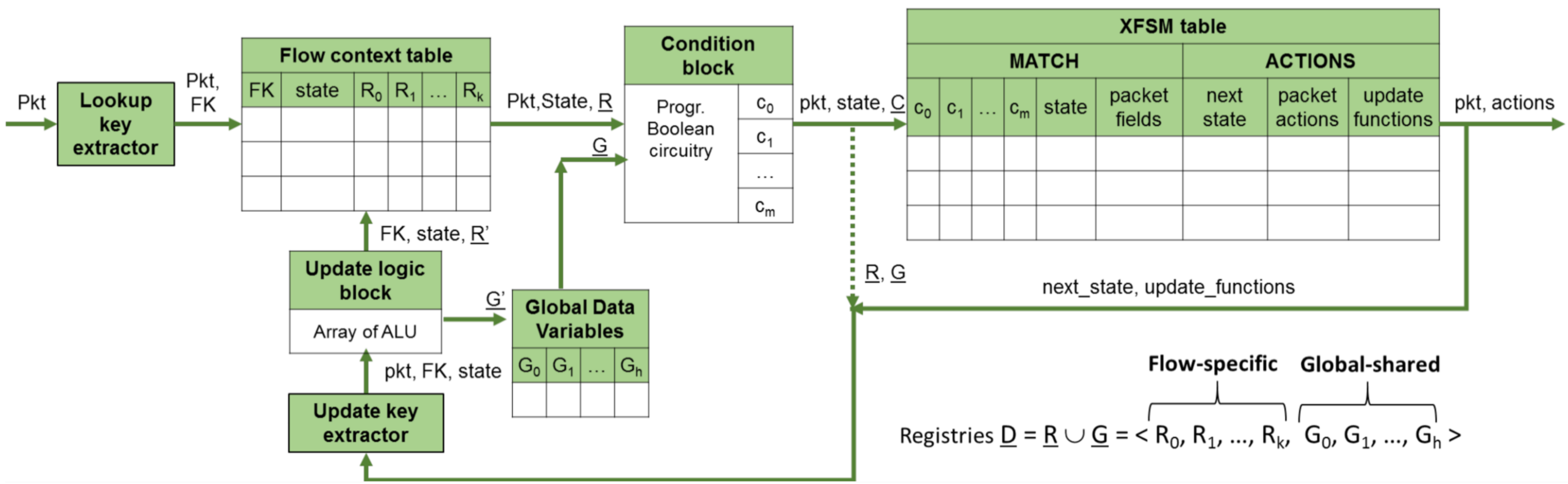
- From dumb to ”local smartness” on switch
- Memory associated to flow on switch
- History-dependent decision on flow



# Open Packet Processor (OPP)



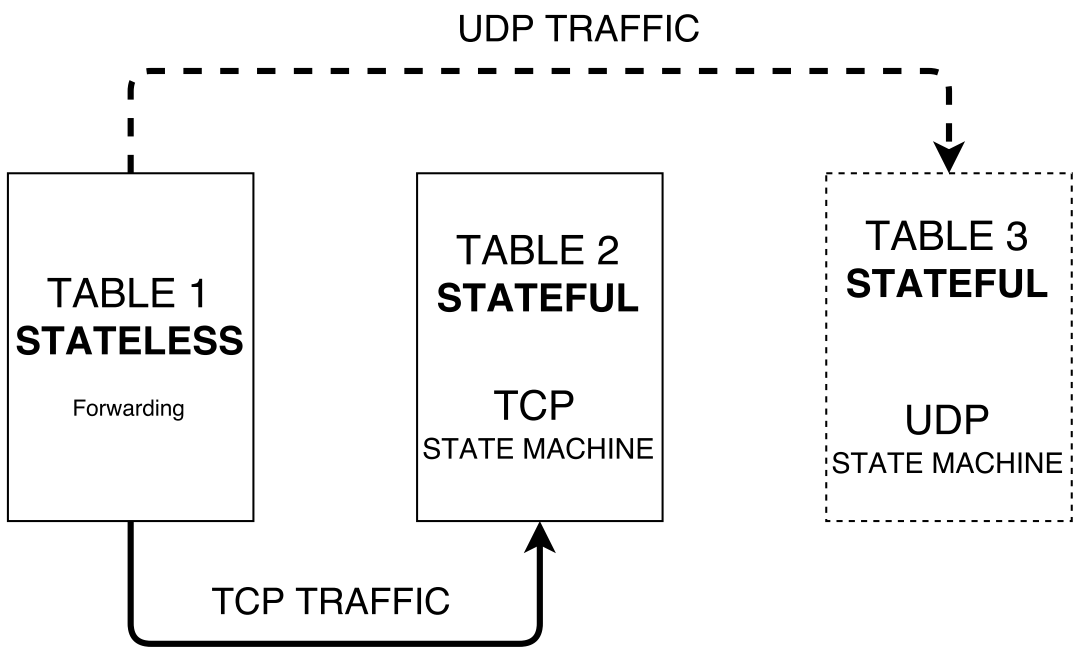
- OpenFlow stateful extension
- Flow states and flow registers
- Extended Finite State Machine model
    - **States:** Forwarding policy
    - **Transitions:** packet-level events, time-based events, conditions



# Proposed application

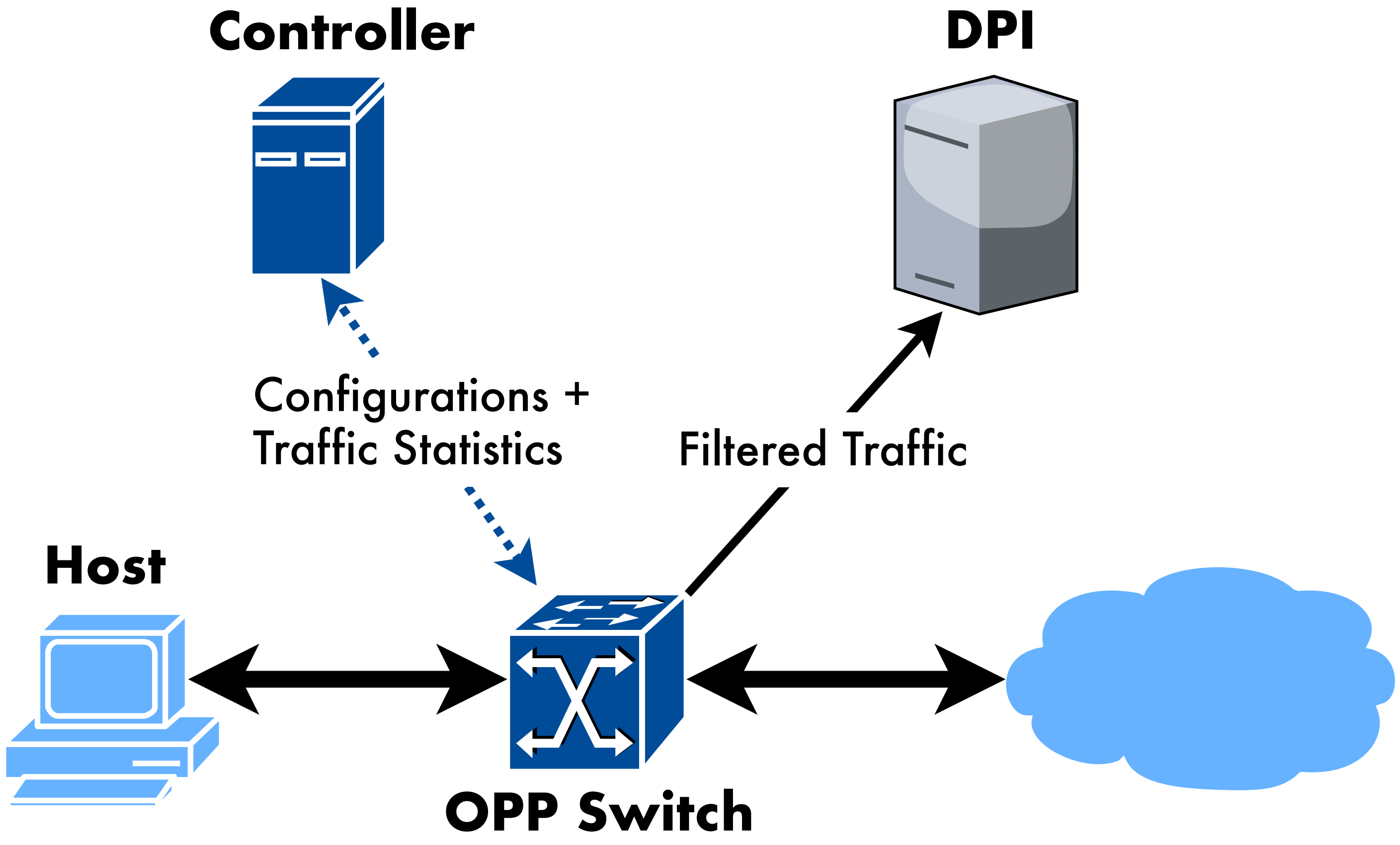


- Tables pipeline:
  1. Stateless table: select output port
  2. Stateful table: filtering, DPI forwarding, statistics collection
- Keep memory for each flow (e.g. TCP connections)

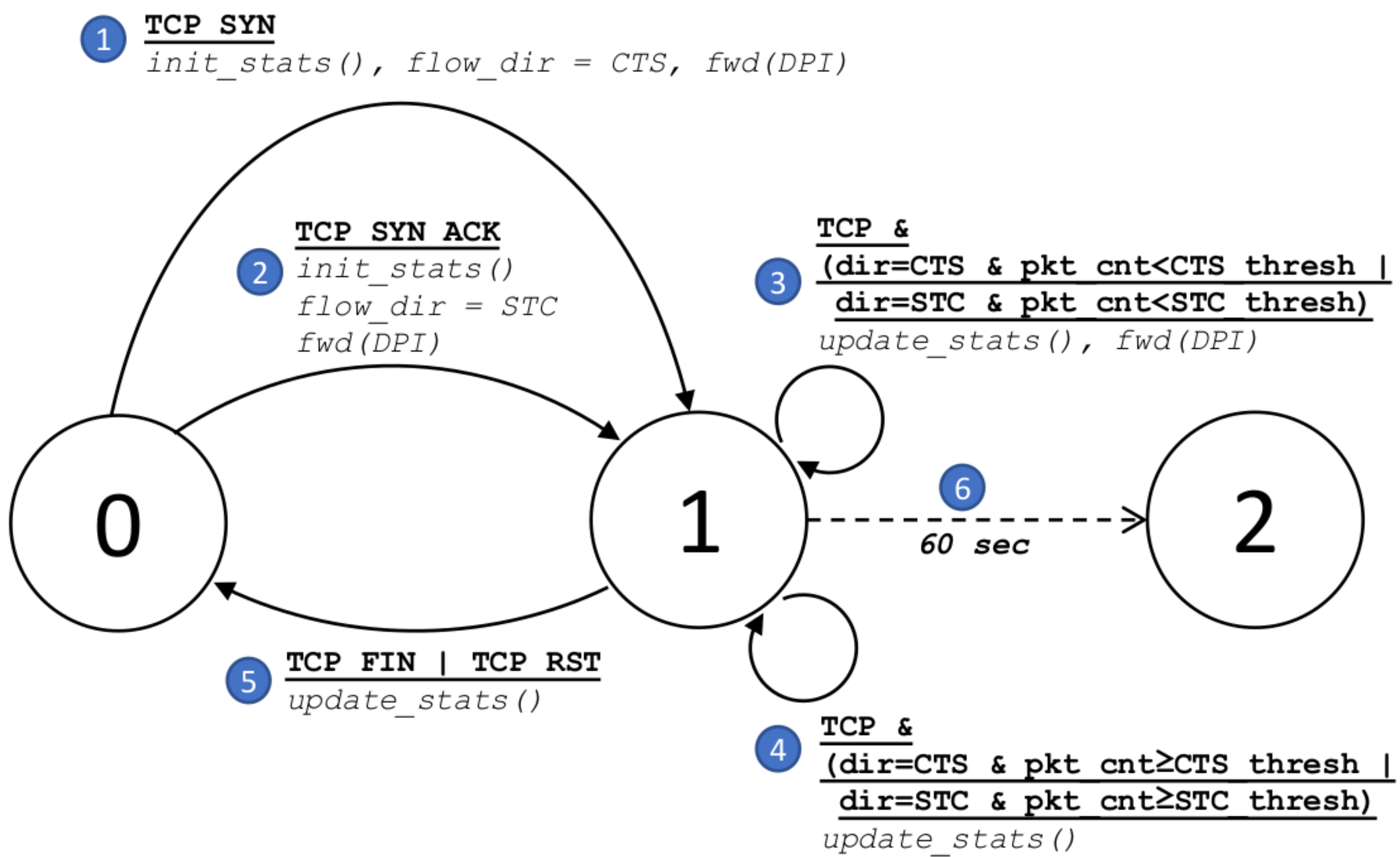


# Topology Example





# TCP State Machine



# Table configuration



**Header Field Extractors**

```
HF[0]=PKT.TS
HF[1]=PKT.LEN
```

**Flow Data Variables**

```
FDV[0]=pkt_cnt
FDV[1]=TS_start
FDV[2]=TS_stop
FDV[3]=byte_cnt
FDV[4]=flow_dir
```

**Global Data Variables**

```
GDV[0]=CTS_thresh
GDV[1]=STC_thresh
GDV[2]=0
```

**Conditions**

```
C[0]: FDV[0]≥GDV[0]?
C[1]: FDV[0]≥GDV[1]?
C[2]: FDV[4]>GDV[2]?
```

# Results



Domestic trace (12h trace of domestic traffic)

- Classification Accuracy
- Filtering impact in terms of packets analyzed by the DPI
- Filtering impact in terms of offloaded bytes from the DPI

CAIDA trace from 10GbE backbone link (7.2TB of traffic with 56M TCP connections):

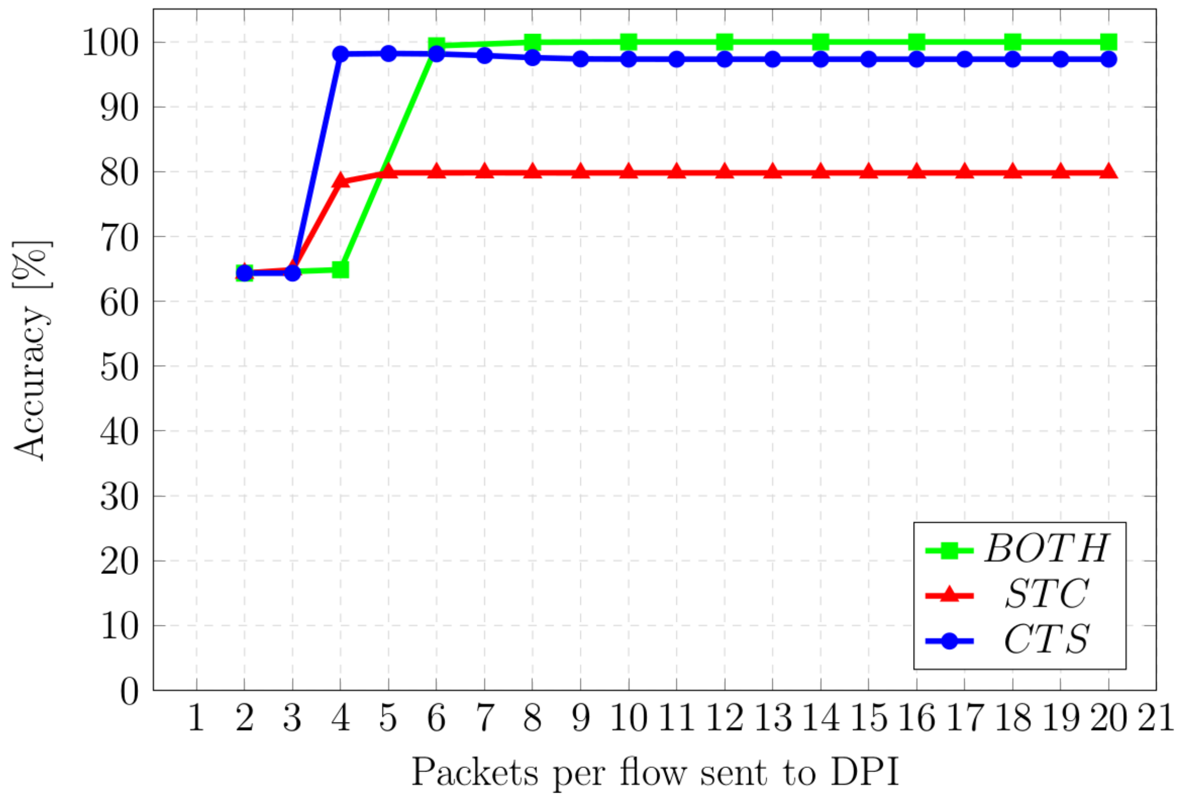
- Filtering impact

# Results – Classification Accuracy



Negligible loss of classified flows

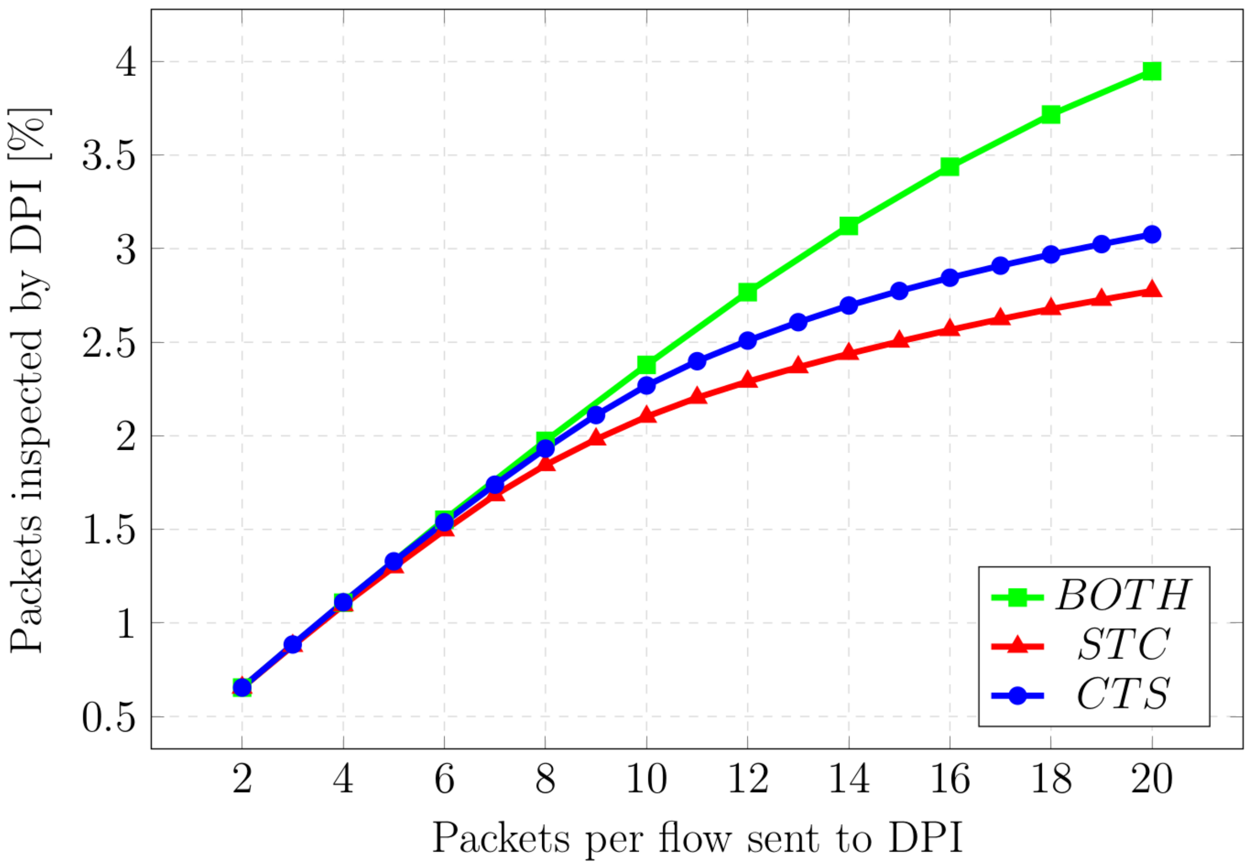
**10 pkts: 100% accuracy**




- BOTH: filtering both direction
- STC: filtering the Server-To-Client direction + 1 packet in the other
- CTS: filtering the Client-To-Server direction + 1 packet in the other

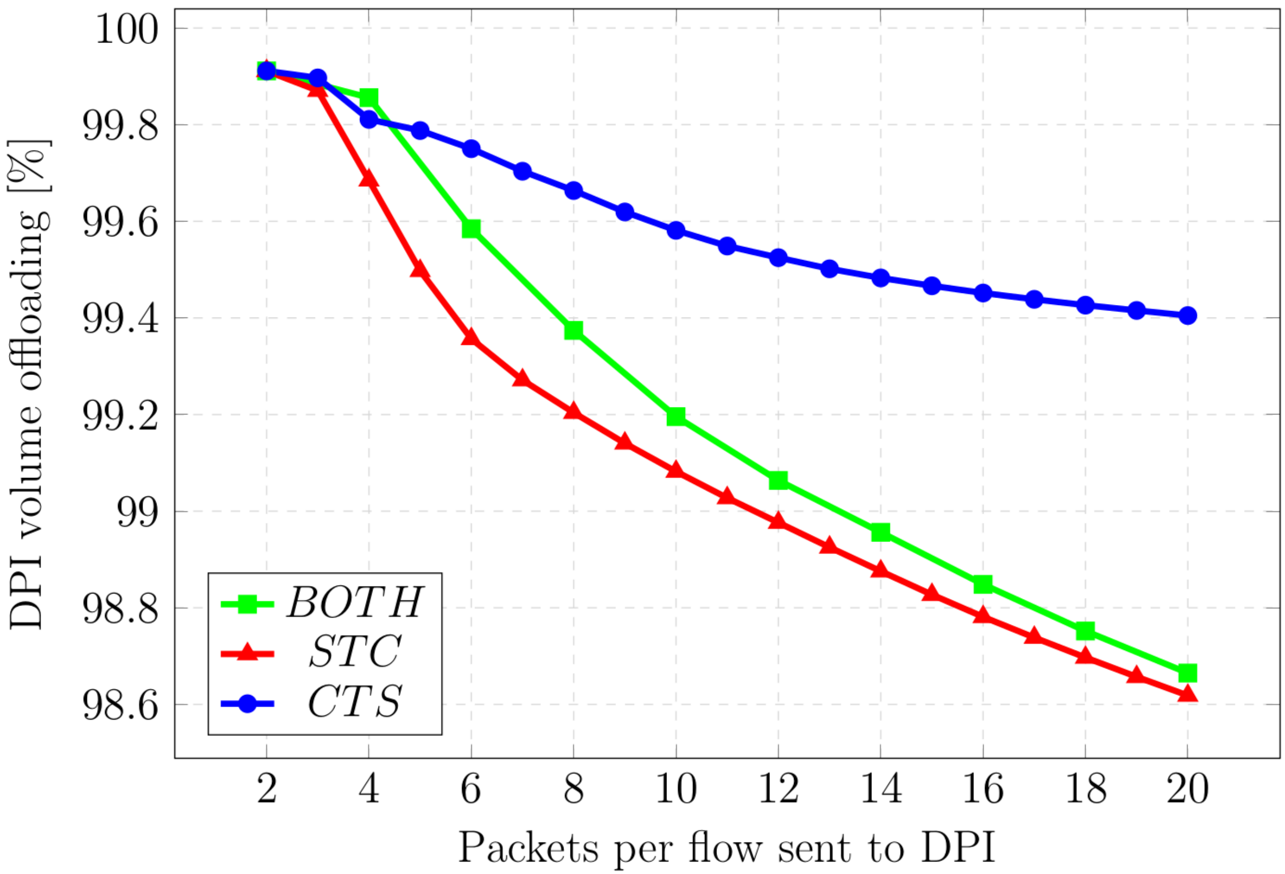
# Results – Filtering Impact (Packets)





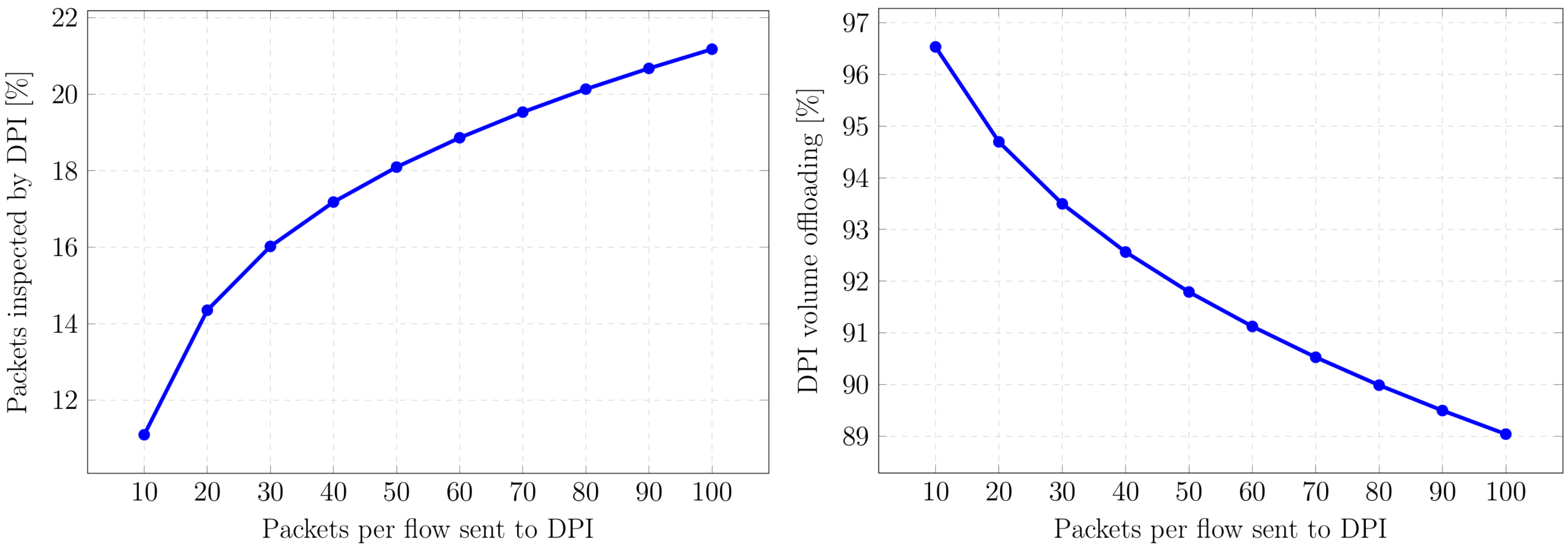

100% classification accuracy can be reached inspecting only 2.3% of the total packets

# Results – Filtering Impact (Bytes)




100% classification accuracy can be reached offloading more than 99% of traffic from the DPI

# Results – Filtering Impact on CAIDA




- Only BOTH case (most promising in terms of accuracy)
- Threshold bigger than previous evaluation (also with bigger threshold we can reach high traffic offloading)

# In-switch statistics collection



Despite the DPI's lack of complete visibility of flows, the switch is still able to compute useful flow metrics such as:

- **Start** and **end timestamp** of flows
- **Number of packets** per flow per direction
- **Byte quantity** per flow per direction

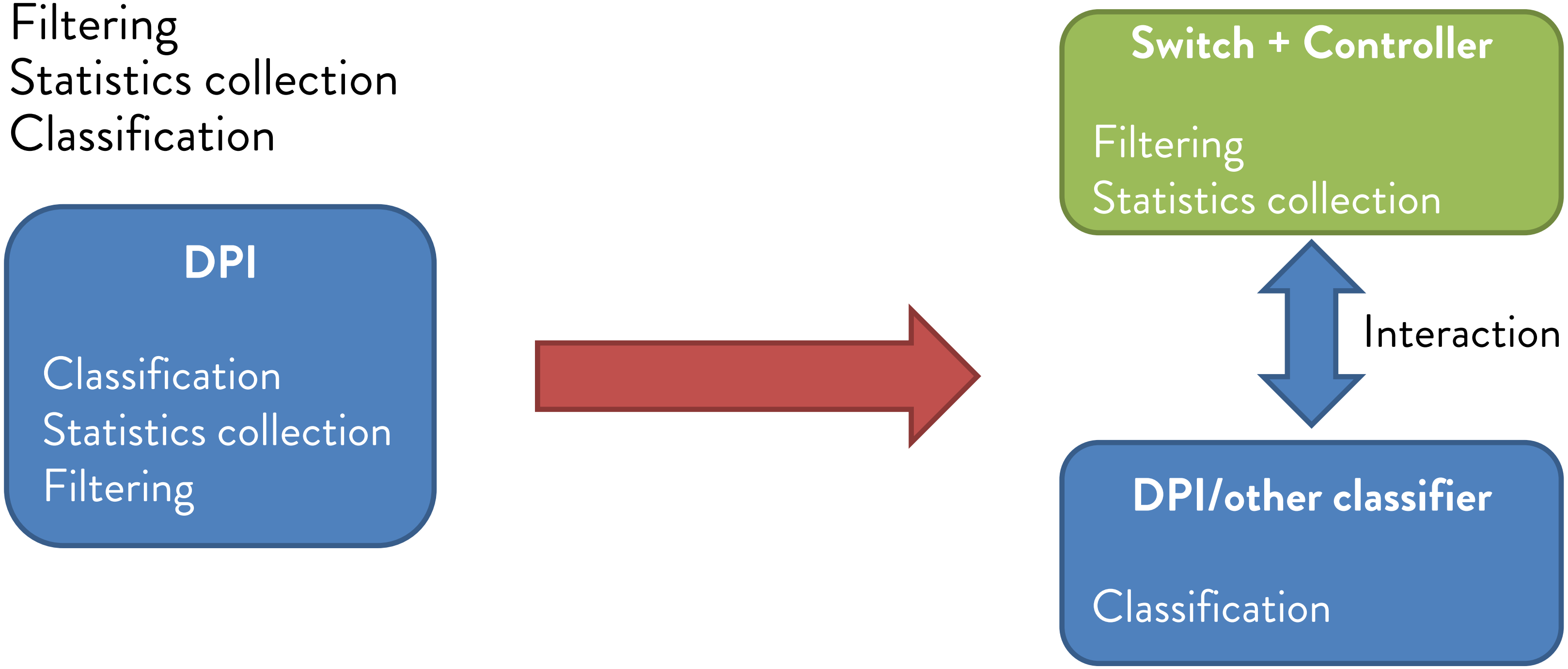
# Conclusion (1)



We propose a scheme to optimize DPI usage exploiting data plane programmability

With this solution we try to separate:

- Filtering
- Statistics collection
- Classification



# Conclusion (2)



Extensive numerical evaluation of the solution showed that our proposal can lead to:

- Zero-classification accuracy loss
- Huge reduction in traffic volume and number of packets to the DPI
- Offload the DPI dramatically reduce the required computational power

The solution is flexible and programmable:

- We can disable the DPI forwarding and make only in-switch statistics collection
- We can integrate feedback from DPI to:
  - provide application-aware forwarding
  - further lowering the filtering threshold
- System can be integrated with ML classifier or other type of classifier

# Future work/Open discussion



- Computation of other flow-related statistics on network element
- Other stateful application that we can offload down to the data plane (e.g. flow sampling to collect statistics)
- Stateful data plane SDN network:
  - Role of the controller
  - Amount of "smartness" in the switch
- Boundaries between offloading tasks to the network and leave the work to dedicated middleboxes or end-hosts

**daniele1.moro@mail.polimi.it**

# THANKS

for your attention

# ANY QUESTIONS?

**daniele1.moro@mail.polimi.it**

# BACKUP SLIDES

# Open Packet Processor: Tables description

## Flow Context Table

- Updated by packet actions or controller
- **Flow Key:** exact match on fields defined by update and lookup scope, configured by the controler
- **State:** flows start with default state. Entries populated by means of set-state action or by the controller

| 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] | |

# Open Packet Processor: Tables description

## EFSM Table

- Updated by the controller
- Extended OpenFlow match+action table
- **Match:** packet fields + **state** + **conditions result**
- **Actions:** packet actions + **next state** + **update actions**

| MATCH | | | | | | ACTIONS | | |
|---|---|---|---|---|---|---|---|---|
| $C_0$ | $C_1$ | ... | $C_m$ | State | Packet fields | Next state | Packet actions | Update functions |
| | | | | | | | | |

# Memory Requirements

- **8 EFSM entries**
- With this application we need 2 entries in the flow context table for each connection
- **CAIDA Trace:** require ~900K flow context entries
- An OPP ASIC implementation can support 256 EFSM entries and 1 Million entries for the flow context tables

# Deep Packet Inspection (DPI)

- Traffic analysis and classification
- Use cases
  - Network security (IDS, IPS, DLP)
  - Bandwidth management
  - User profiling
  - Government Surveillance and Censorship
- Problems:
  - Encrypted traffic
  - High Computational cost