consorzio nazionale
interuniversitario
per le telecomunicazioni

Demo: implementing advanced network functions with stateful programmable data planes

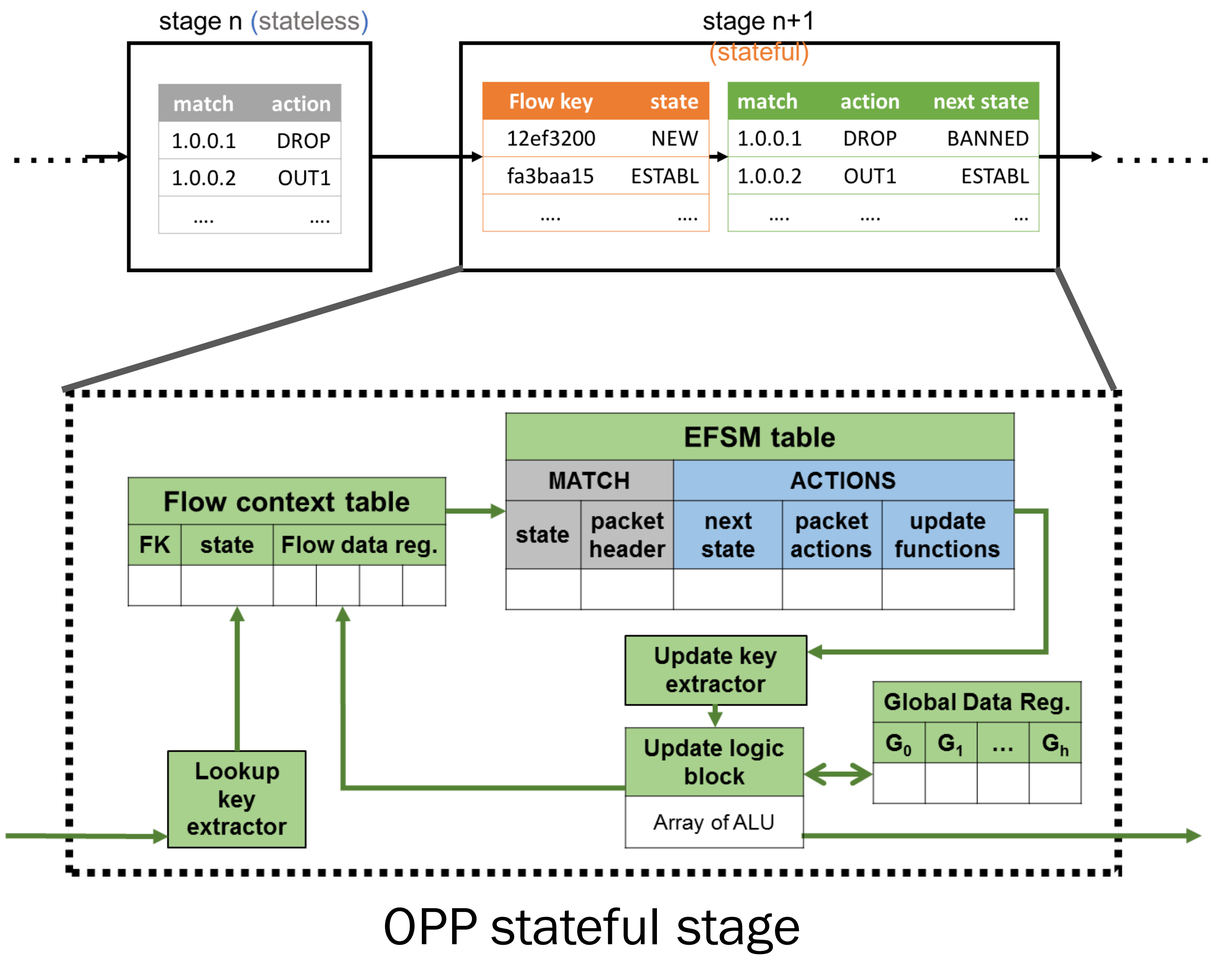
Marco Bonola*, Roberto Bifulco+, Luca Petrucci*, Salvatore Pontarelli*, Angelo Tulumello*, Giuseppe Bianchi*
* CNIT/University of Rome Tor Vergata +NEC Laboratories Europe

1. Motivation

- Prove the use of stateful SDN for implementing complex network functions
- Demonstrate OPP [1] flexibility re-implementing existing applications
    1. a distributed data driven network topology discovery mechanism proposed in[2]
    2. a distributed load balancing mechanism based on an in-band switch telemetry mechanism (inspired by [3])
    3. a server load balancing with static NAT, a well known mechanism used in data centers (e.g.[4]).

2. OPP pipeline at a glance

- **OPP** enables the execution of **Extended Finite State Machines** (EFSM) machines in the data plane
- An OPP pipeline consists of an arbitrary set of stateless (OpenFlow-like) and stateful stages that share data through metadata fields



OPP stateful stage

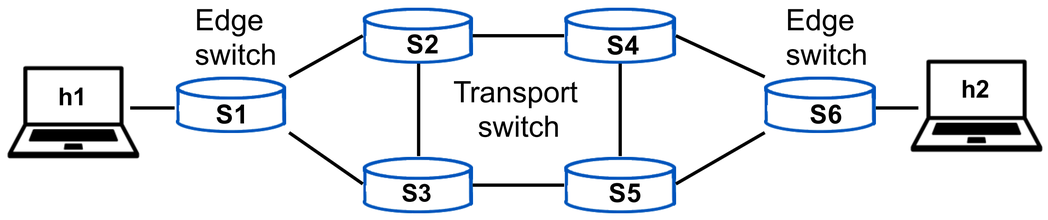
4. Demo

- Two flavor demo:
    1. ***Software prototype*** of OPP in a mininet emulation environment (all use cases)
    2. ***Hardware FPGA*** OPP PoC in lab (use case 2)

References

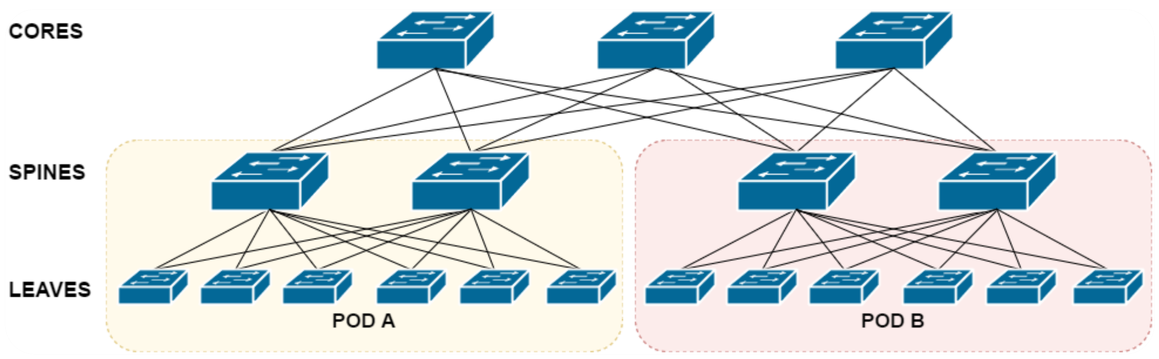
[1] Bianchi, G., Bonola, M., Pontarelli, S., Sanvito, D., Capone, A., & Cascone, C. (2016). Open Packet Processor: a programmable architecture for wire speed platform-independent stateful in-network processing. arXiv preprint arXiv:1605.01977.
[2] McCauley, James, et al. "Taking an AXE to L2 Spanning Trees." Proceedings of the 14th ACM Workshop on Hot Topics in Networks. ACM, 2015.
[3] Kim, Changhoon, et al. "In-band network telemetry via programmable dataplanes." ACM SIGCOMM. 2015.
[4] "AWSElasticLoadBalancer" https://aws.amazon.com/elasticloadbalancing

UC1: data driven network topology discovery



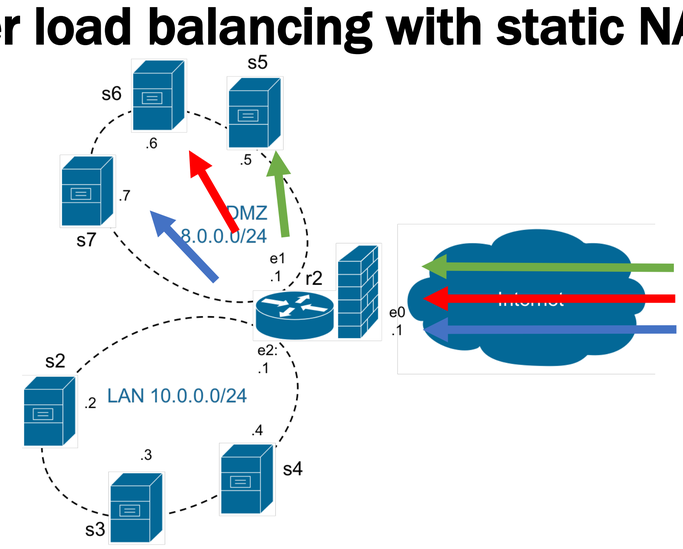
- This mechanism is used to find the best path in a switched LAN with redundant links and possible loops, without requiring the exchange of control messages.
- Data packets are embedded in a custom header to transport a few information used by the algorithm:
    1. the sum of the link costs for the entire path from the source to destination
    2. a unique id used to discriminate duplicated packets
- Without requiring any intervention by external controllers and any ad-hoc control packets all switches learn the best path for all destinations (avoiding loops)

UC2: in-band switch telemetry for load balancing



- Telemetry probes are periodically generated by leaves and replicated by spines (data packet + MPLS header)
  Format: ***leafId*** (mpls_tc); ***utilization*** (mpls_label)
- The utilization field contains the sum of the link loads in the packet's opposite direction
- Each leaf has a best hop for every destination
- Leaves are capable of dynamically re-route packets through the best hop toward the destination

UC3: server load balancing with static NAT



- Each first packet of a flow is "assigned" to 1 of the servers in the cluster (in RR policy) and DNATed toward it accordingly.
- All subsequent packets will be forwarded and DNATed toward the proper server.

Partly funded by EU H2020 projects:
"BEBA" grant agreement 644122
http://www.beba-project.eu
"SUPERFLUIDITY" grant agreement 671566
http://superfluidity.eu