Building a Multi-Layer HCoS Router with P4

P4 Developer Days

Miroslaw Walukiewicz

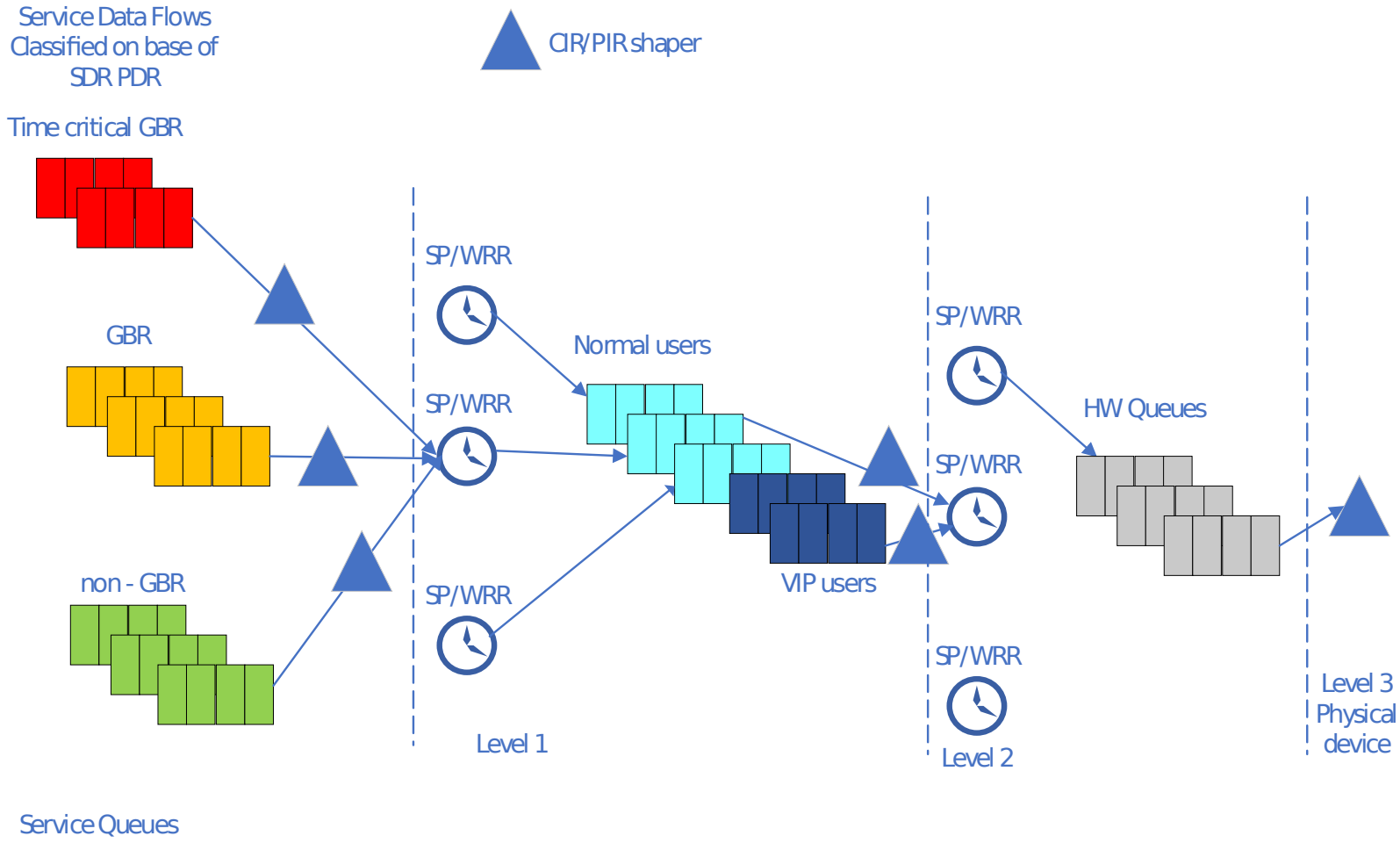
mirek@pantherun.com

08/12/2026

QoS overview

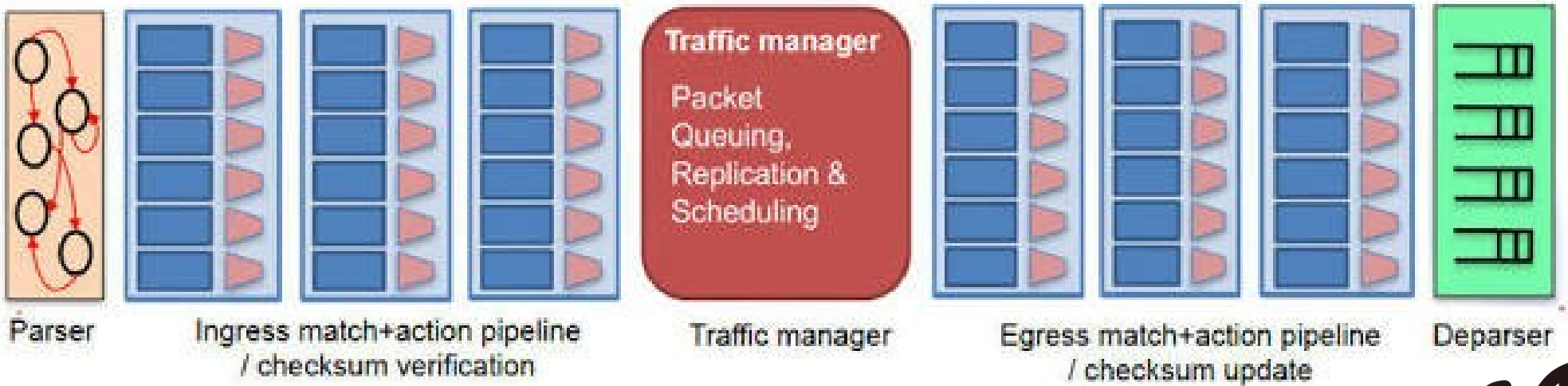
- QoS is a very complex networking topic ... in both configuration and implementation space.
- Everyone can understand (more or less ⏪) how it works, but almost nobody is able to make it working perfectly.
- There is no single definition of QoS.
- QoS definition depends on hardware silicon capabilities:
  - It contains buffering, scheduling and prioritization engines.

Traditional HQoS example (UPF)

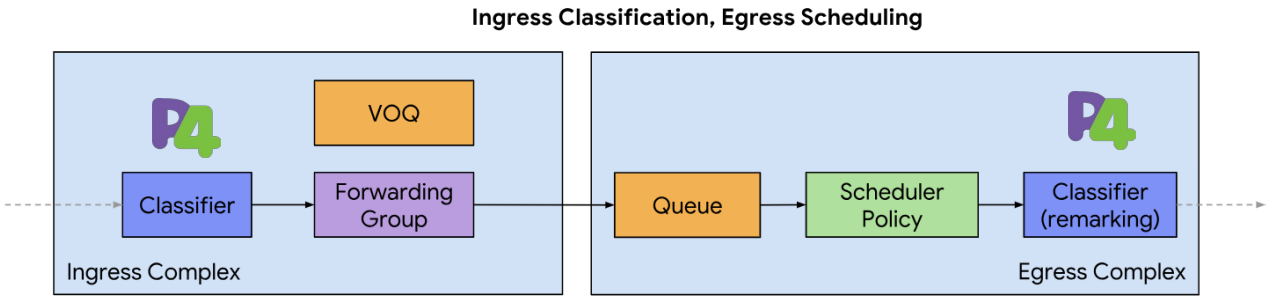


How the P4 language can be used?

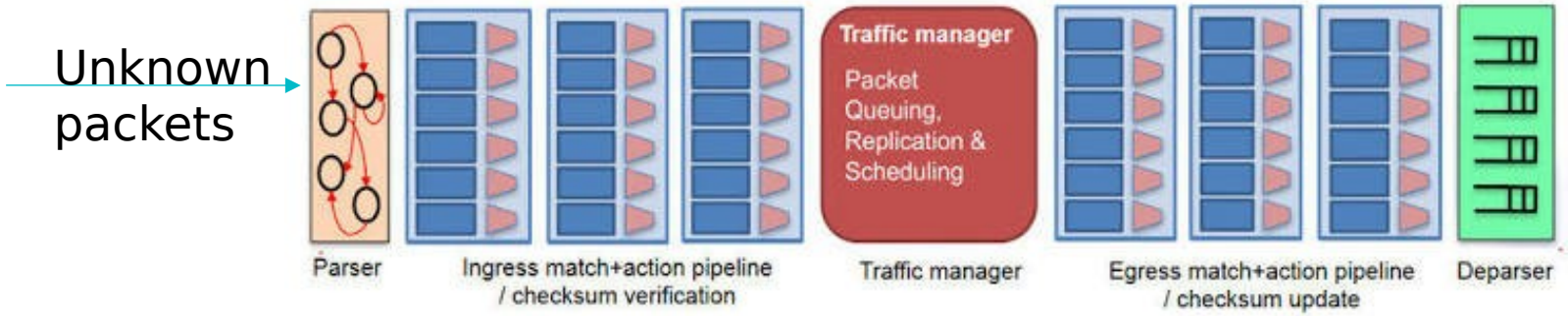
- Long time ago some people tried to define full P4 defined QoS engine and they failed.
- This enormous effort learned others that P4 language can be used for controlling QOS engines/externs using metadata.
  - See PSA or TNA intrinsic metadata definitions describing some specific QoS model.



P4 helps in defining QoS policies



[OpenConfig - QoS model overview]



- QoS trust tables
- QoS mapping to forwarding groups

- QoS Remarking

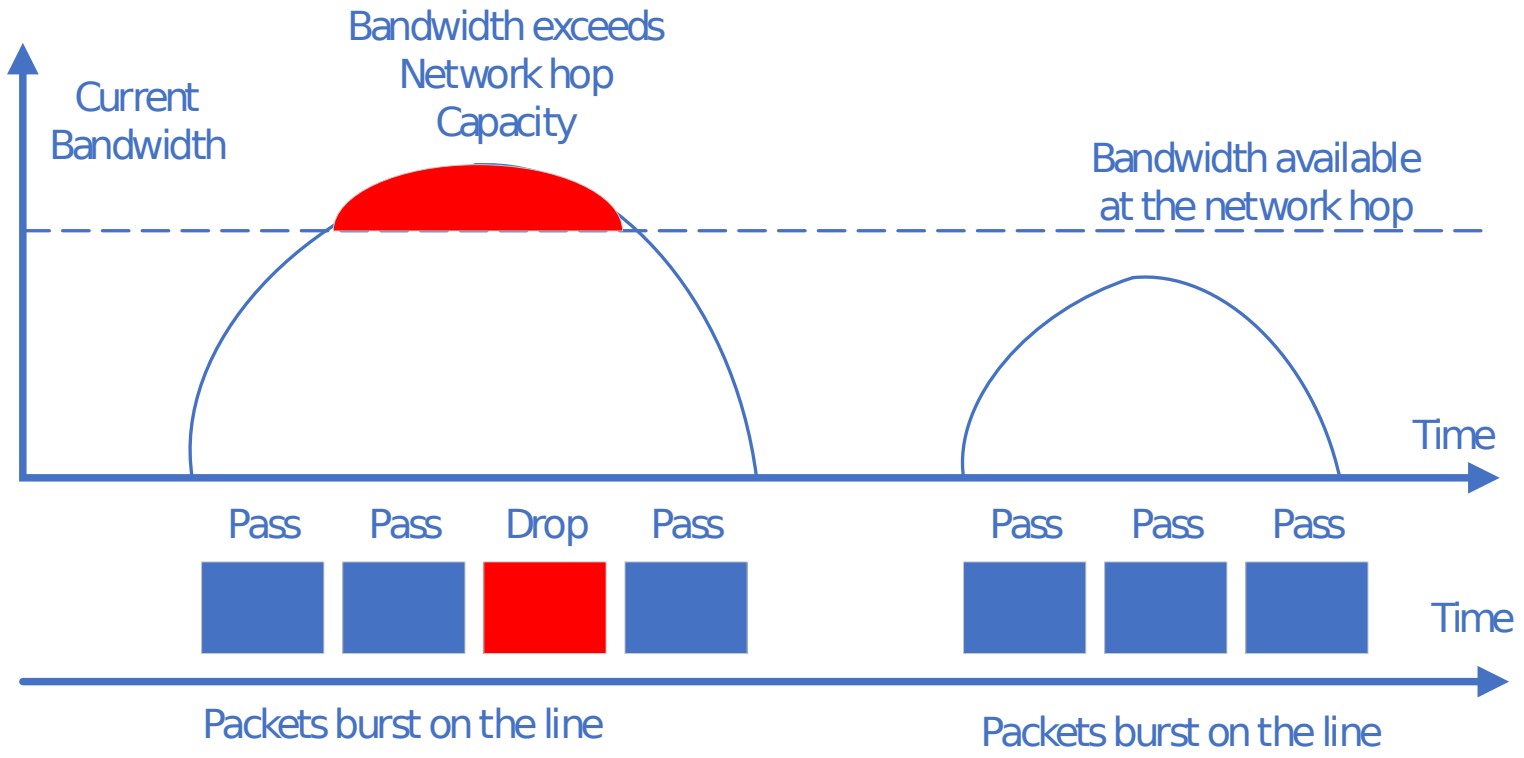
New challenges in scheduling

- Deterministic scheduling:
  - TSN for industrial, robotics
  - professional Audio Video
  - Safety in space and airplanes
  - Asynchronous Traffic Scheduling
- Complex policy based shaping (bursts handling)

Traditional QOS does not help much:

- possible packet drops,
- extended latency and complexity

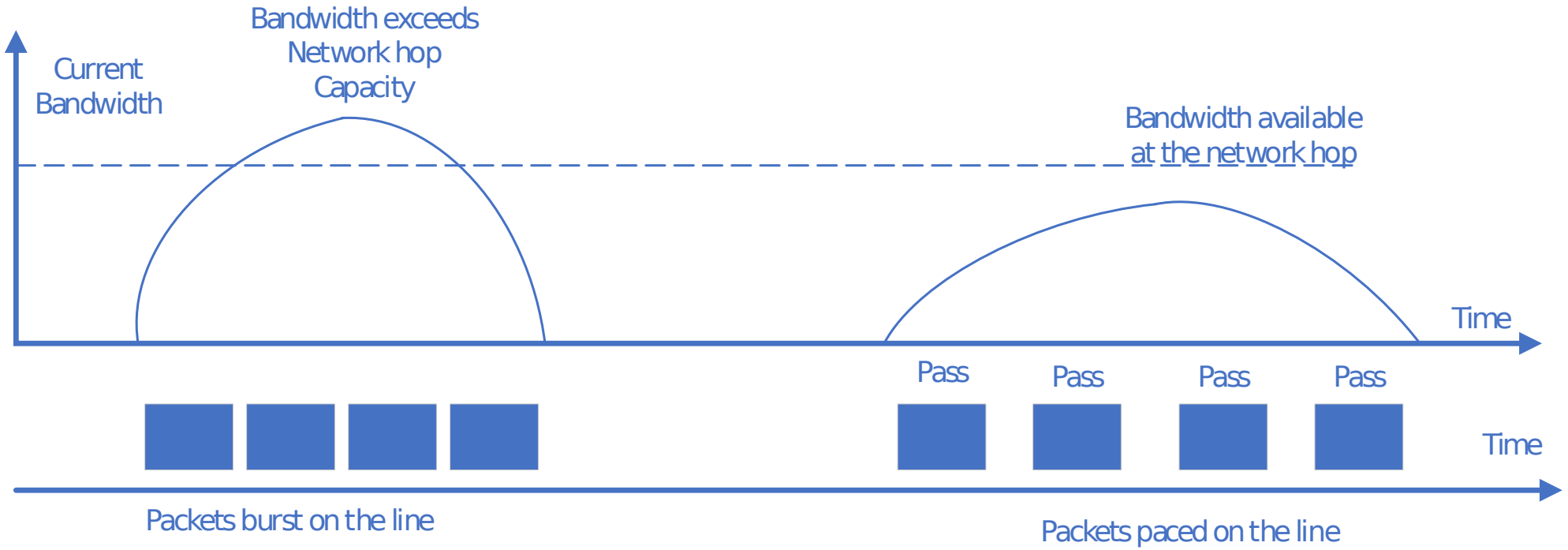
Shaping problem



Packet drops when BW exceeds limits

ANTHERUN

What is expected?

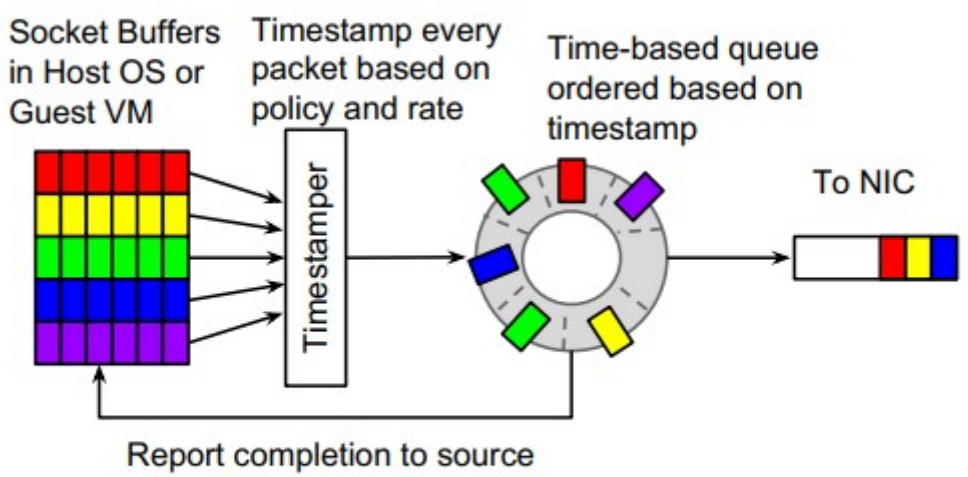


Delay packets instead of dropping

PANTHERUN

New approach in scheduling

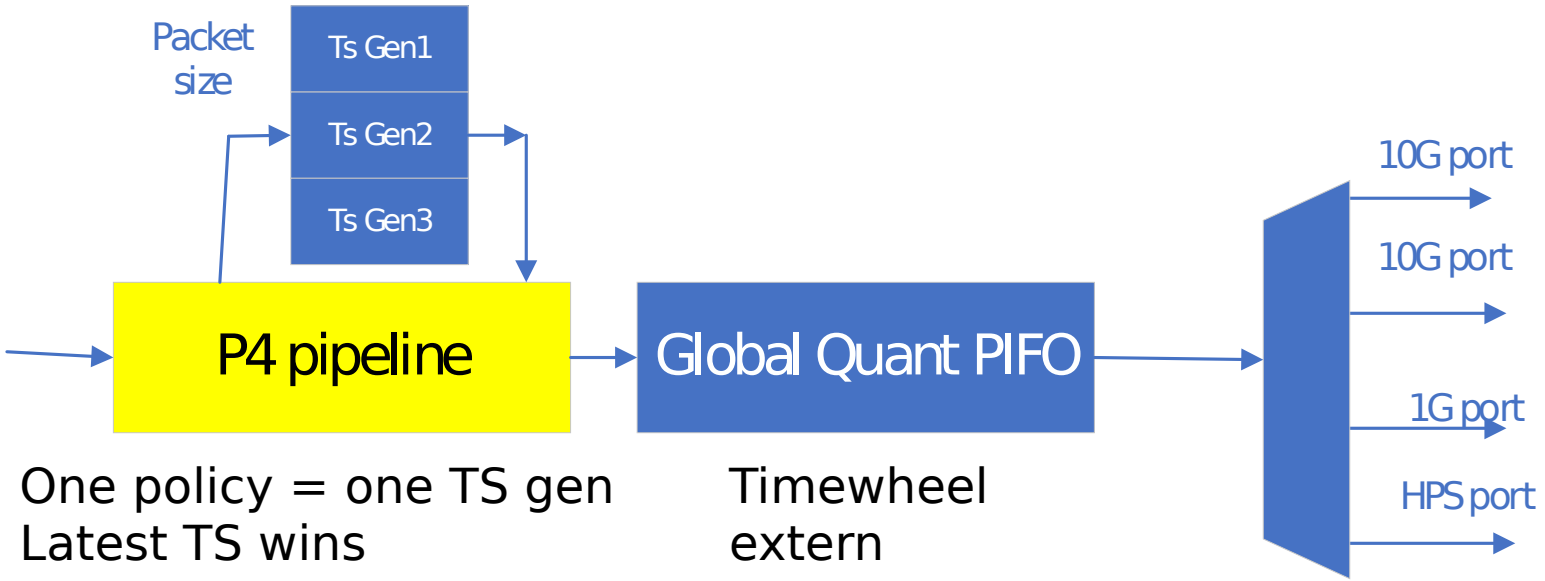
- Predict time when packet needs to be released
- Wait in the queue, until packet’s release time happens
- Release it...



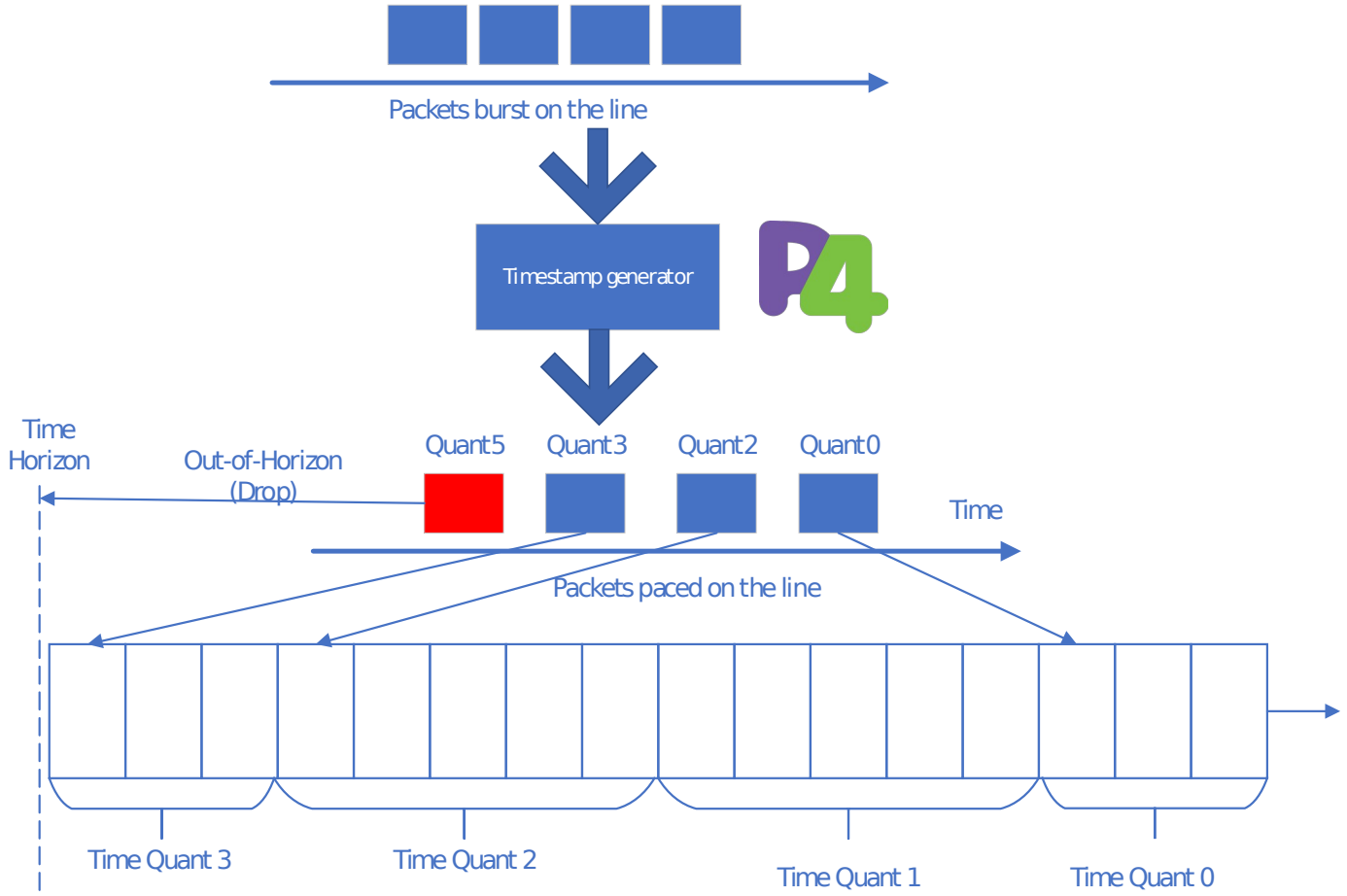
Carousel: Scalable Traffic Shaping at End-Hosts

Ahmed Saeed · Nandita Dukkipati · Valas Valancius · Terry Lam · Carlo Contavalli · Amin Vahdat · ACM SIGCOMM 2017

Real world HW implementation



How it works?



ANTHERUN

Real P4 code challenges

- Timestampers are P4 registers based.
- P4 register update is made as read-modify-write that is always challenging in HW.
- Timestamp computation can be performed parallelly.

The P4 code assumes always pipelining it may introduce additional cycles in system latency.

  - Maybe we can think about some P4 controls parallelization?

Thank you

Contact: mirek@pantherun.com