DESIRE6G

RESOURCE SHARING POLICY ENFORCEMENT IN ACCESS AGGREGATION NETWORKS

Sándor Laki, Assistant professor

Communication Networks Laboratory
ELTE Eötvös Loránd University
Budapest, Hungary

Web: https://lakis.web.elte.hu

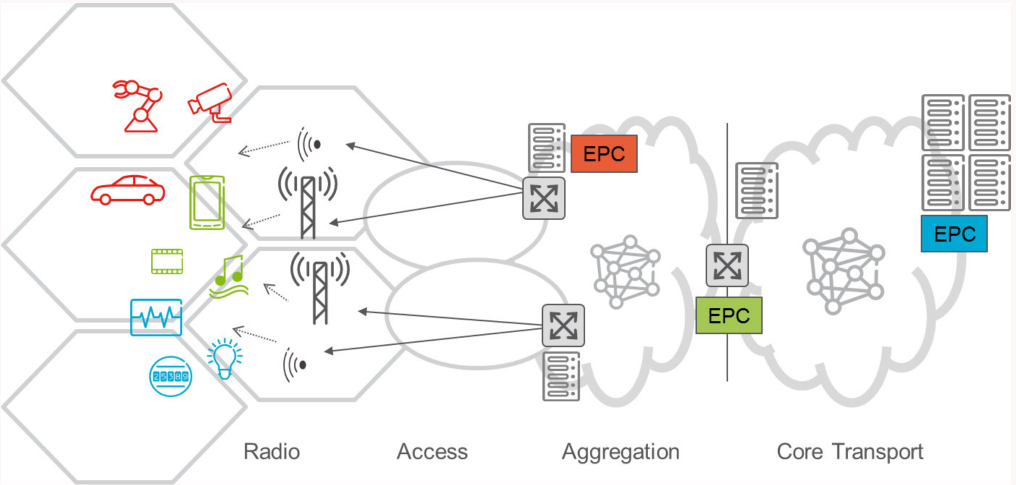
Co-funded by the European Union

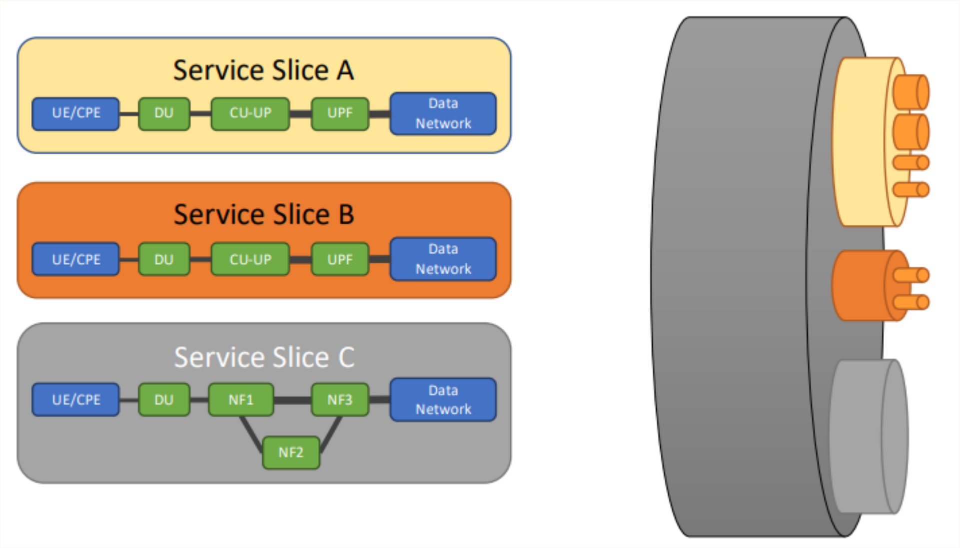
6GSNS

6G Programmable Deterministic Webinar Series, 10/22/2024

THIS TALK

- Access Aggregation Network
  - Uplink/downlink policies
- Hierarchical resource sharing policies
  - HQoS among MNOs, slices, users
  - Performance isolation
  - Data plane vs control/management plane aspects
- Queueing latency requirements
  - UltraLow <1ms
  - L4S ~ 1ms
  - Normal
  - Congestion controlled or not?





LOW DELAY

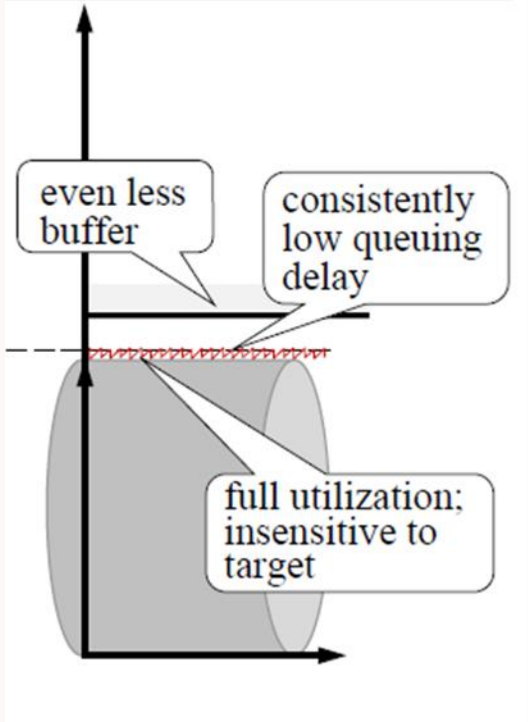
- Not only non-queue-building traffic
  - DNS, gaming, voice, SSH, ACKs, HTTP requests, etc
- Capacity-seeking traffic as well
  - TCP, QUIC, RMCAT for WebRTC
  - web, HD video conferencing, interactive video,
  - cloudrendered, virtual reality, augmented reality,
  - remote presence, remote control,
  - interactive light-field experiences, …





L4S

- Low Latency, Low Loss, Scalable Throughput Internet service
  - ECN-based congestion notification
  - Reaction to the extent of congestion
  - Scalable congestion control
  - Small buffer space
- AQM support is needed
  - ECN marking



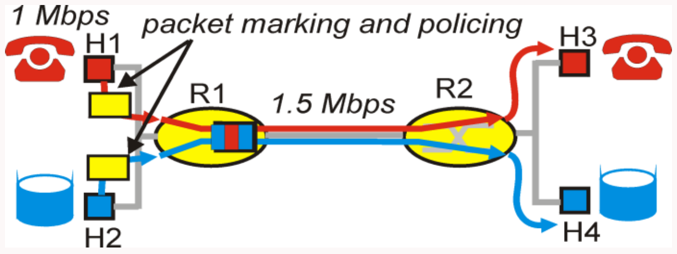
IMPLEMENTING HQOS TM AT SCALE

Traditional Hardware	Software Only	Programmable Hardware
Expensive	Cheap	
Rigid	Flexible	?
Scalable	Non scalable	

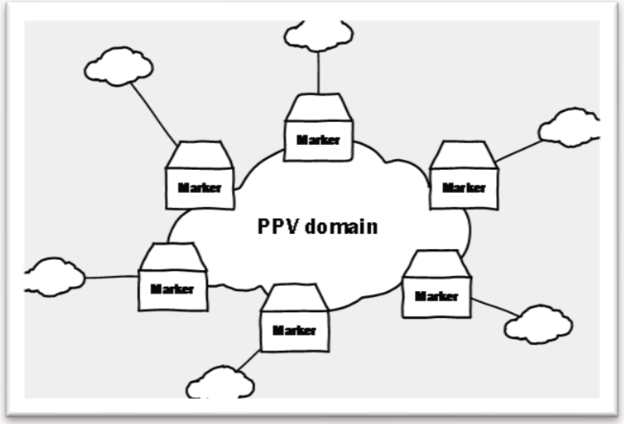
IMPLEMENTING HQOS TM AT SCALE

Traditional Hardware	Software Only	Programmable Hardware
Expensive	Cheap	Limited TM engine HQoS not possible
Rigid	Flexible	… but it can perform computations
Scalable	Non scalable	… but it can handle new headers
		… but it can do all these at line rate

PER PACKET VALUE (PPV) RESOURCE SHARING

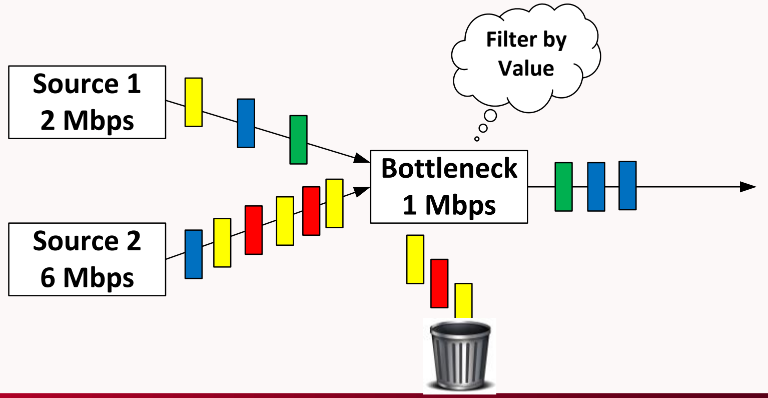


- Our approach is based on the **Per Packet Value** framework
  - Core-stateless approach as CSFQ, RFQ, AIFO, RIFO, etc.



- **Packet Marker** at the edge of the network
  - **Stateful, but highly *distributed***
  - *Assigning values to packets*
  - Packet values are **incentives** helping to decide which packet to forward/drop in case of congestion

- **Resource Nodes** (e.g., routers) aim at maximizing the total transmitted Packet Value.
  - **Stateless and *simple***
  - *Drop packets with **minimum value first strategy** if packet arrives at a full buffer*

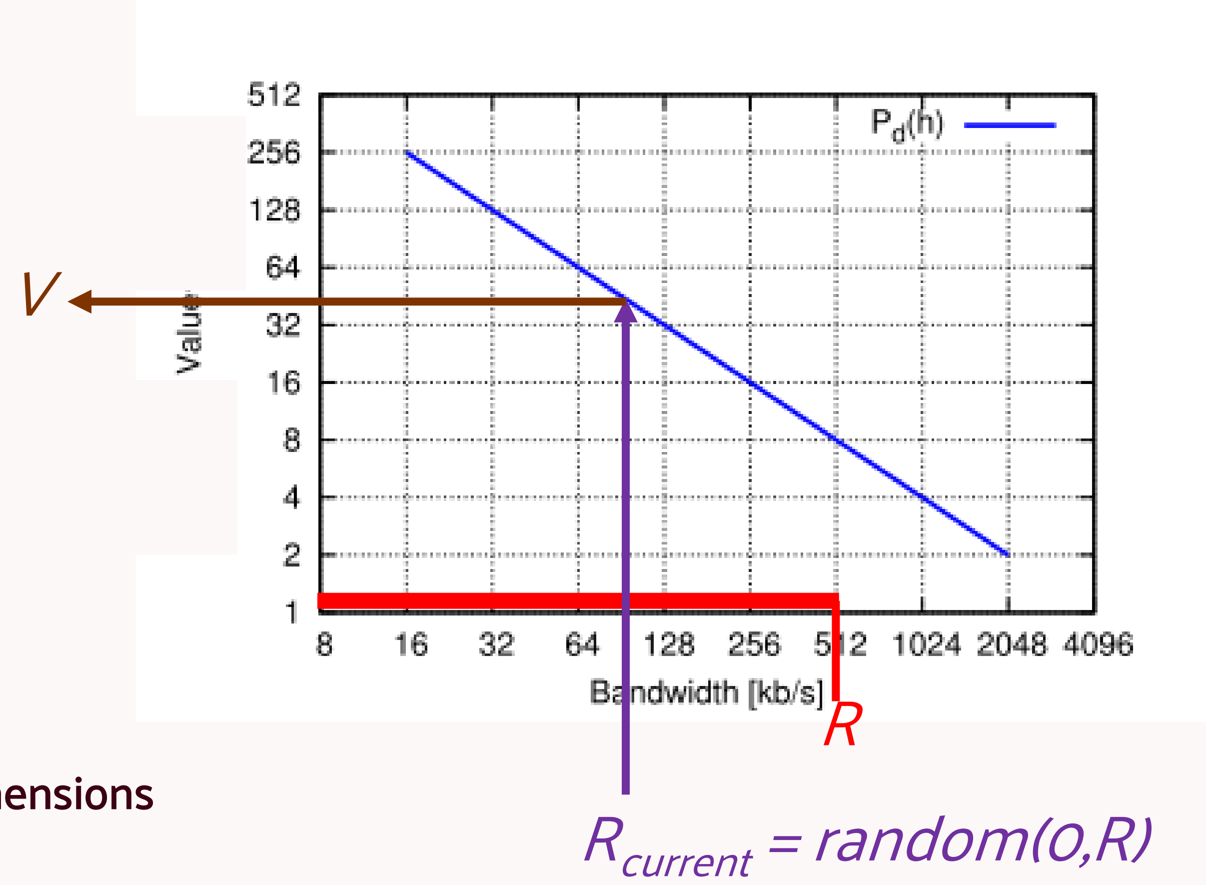


PACKET MARKING - SIMPLIFIED

- **Independent markers** for each traffic aggregate (TA)
- **Resource sharing policy** expressed by a *Throughput-Value Function*
  - Marginal utility, strictly monotne decreasing

- **Packet Value** based on a given *TVF*
  - Continously measures the rate R of TA
  - $R_{current} = random(0,R)$
  - $V = TVF(R_{current})$

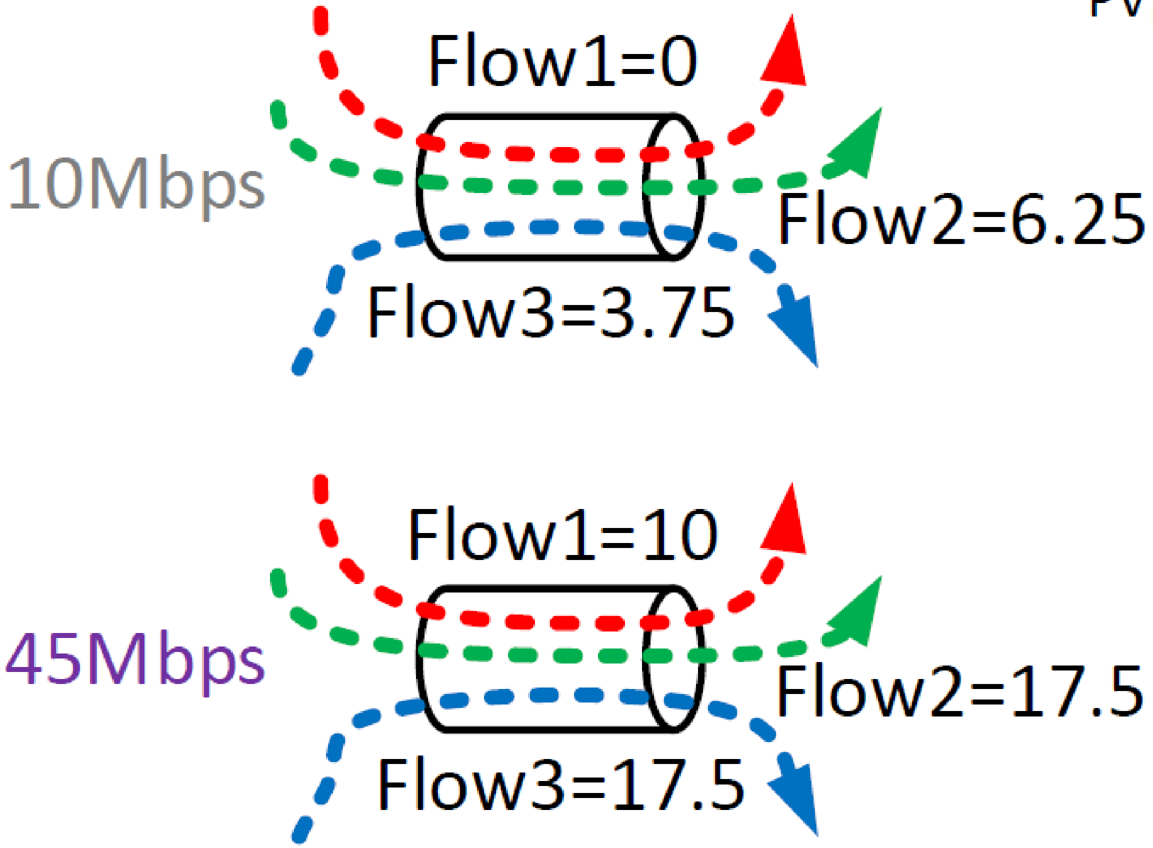
- **Other possible markers/incentives as orthogonal dimensions**
  - Traffic class L4S vs Classic
  - Delay class
  - etc.

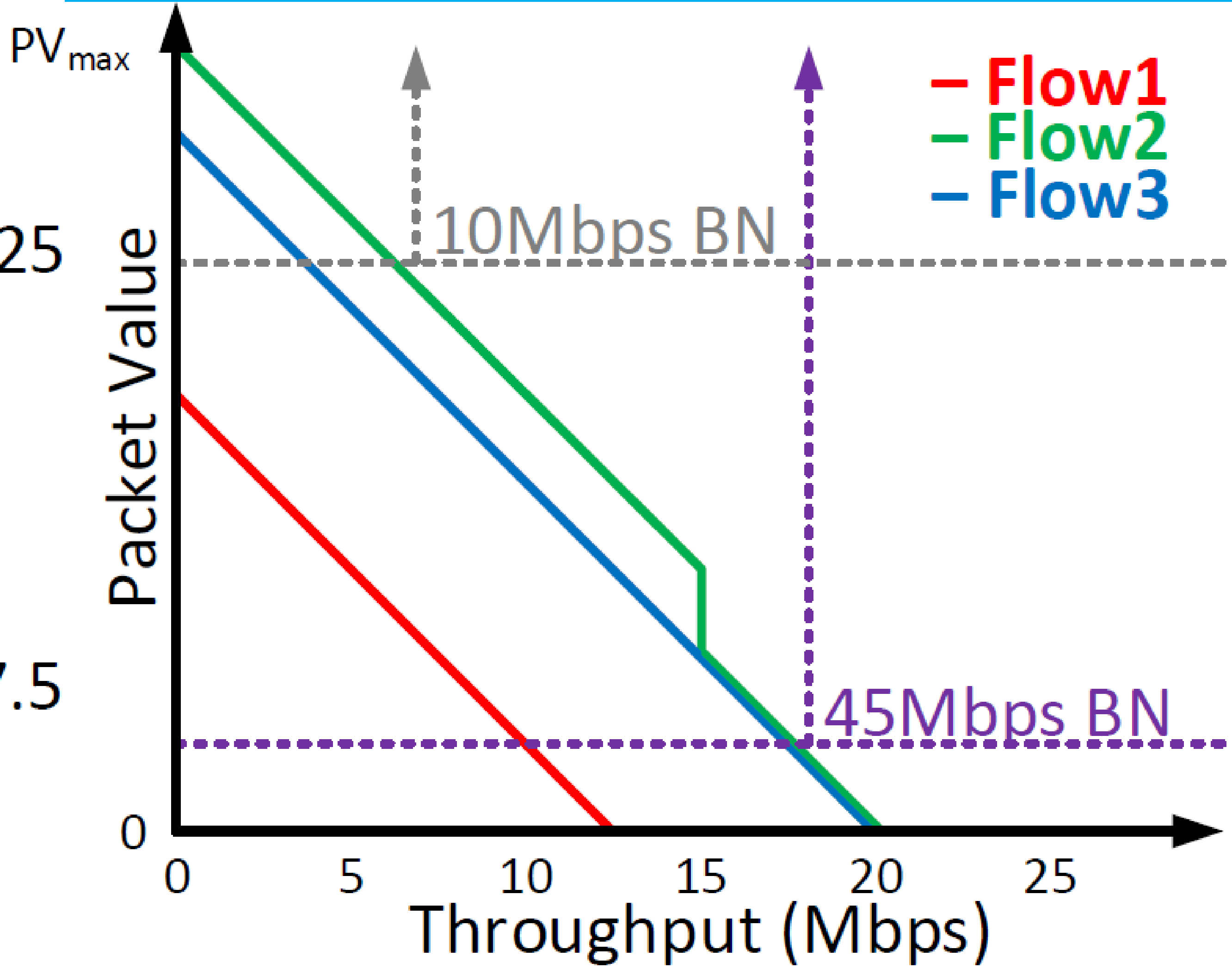


RESEARCH SHARING WITH PER PACKET VALUE

DESIRE6G

Throughput-Value Functions (TVF) defining resource sharing strategies





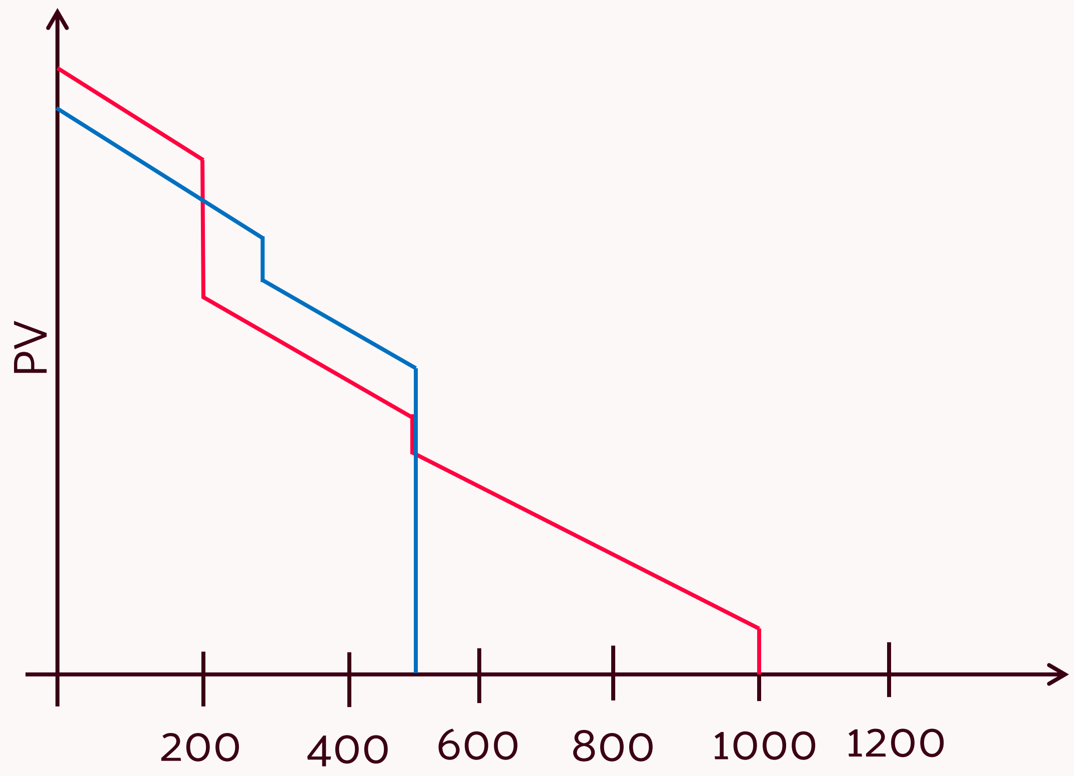
EXAMPLE POLICY DESIGN

- Policy group:
  - Applied on a flow group at one of the aggregation levels
- Policy configuration:
  - List of guaranteed and best effort segments
  - Estimated maximum demand (the traffic is shaped at this demand level)
- Policy configuration segments:
  - One guaranteed segment: bandwidth and weight
    - The priority is to ensure the guaranteed bandwidth requirements for all flows
    - If it cannot be satisfied, the weight expresses how different entities share the bottleneck
  - 1st Best effort segment: bandwidth and weight
    - Above the guaranteed bandwidth until the best effort bandwidth the available resources are shared according to the weight
  - 2nd Best effort segment: bandwidth and weight
    - Above the 1st best effort segment until the 2nd best effort bandwitdth...

EXAMPLE POLICY DESIGN

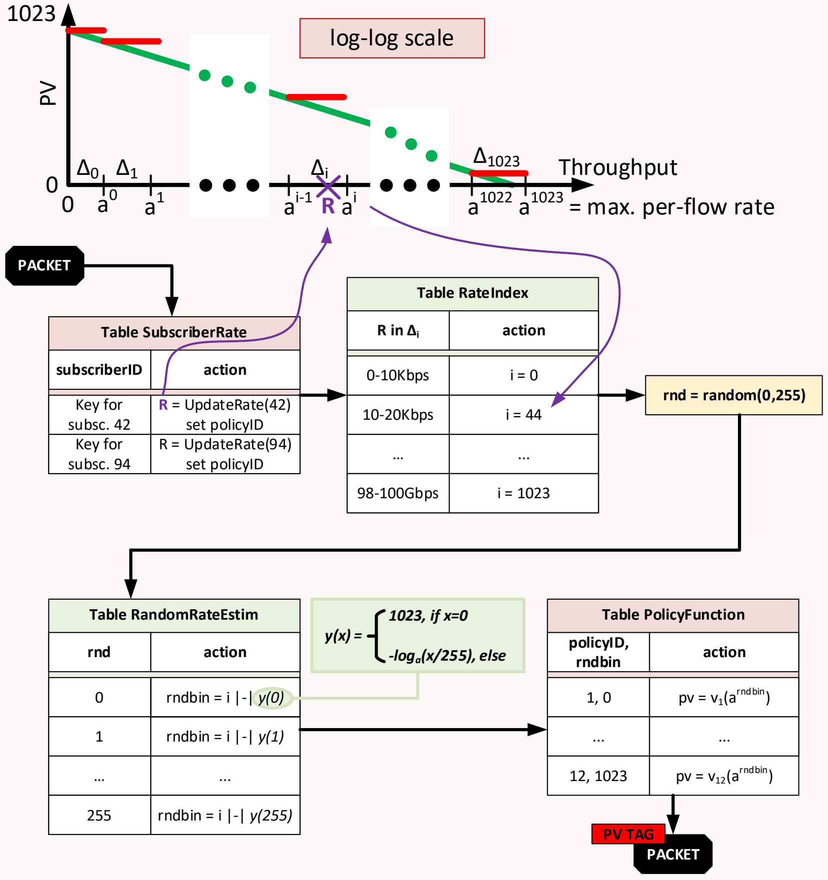
- Policy RED
  - Guarenteed segment: 200 Mbps, w=10
  - 1st Best effort segment: 200-500 Mbps, w=2
  - 2nd Best effort segment: 500-1000 Mbps, w=1

- Policy BLUE
  - Guaranteed segment: 300 Mbps, w=9
  - 1st Best effort segment: 300-500Mbps, w=3



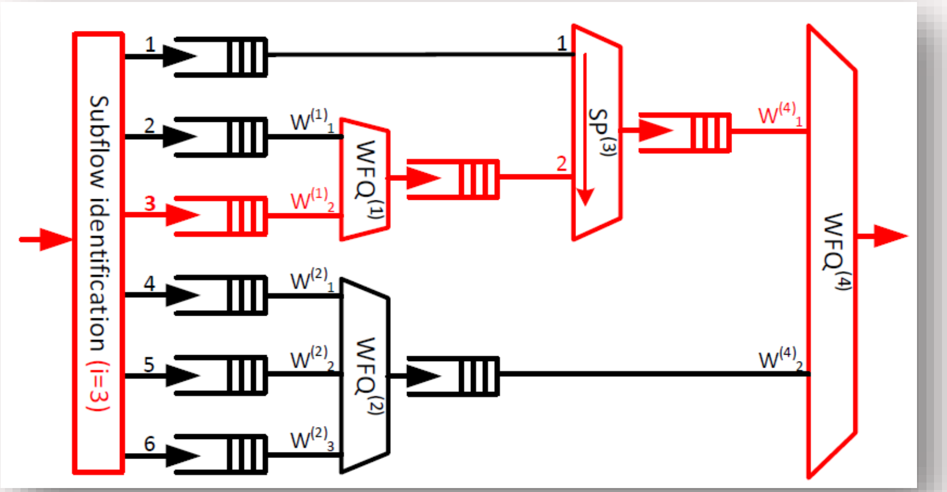
LINE RATE MARKING ON INTEL TOFINO FOR 35K+ USERS

›DESIRE6G‹



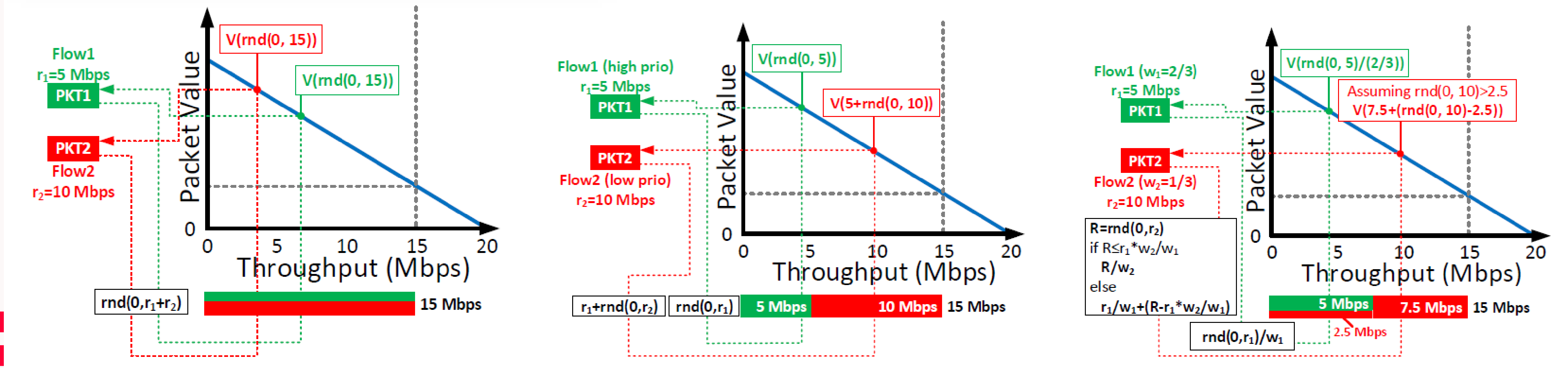
- The policy function is quantized logarithmically
- 1 marker instance marks the traffic of a single subscriber
  - Logarithm tables to solve
    - Calculate packet value from rate measurement
    - Random number generation between [0 rate]
      - Limited to [0 (2^n)-1]
        - "n" is an integer constant
- Data plane only implementation.
  - Policy functions are configured by the control plane
- We run 35000 marker instances in parallel

HQOS POLICY MARKING



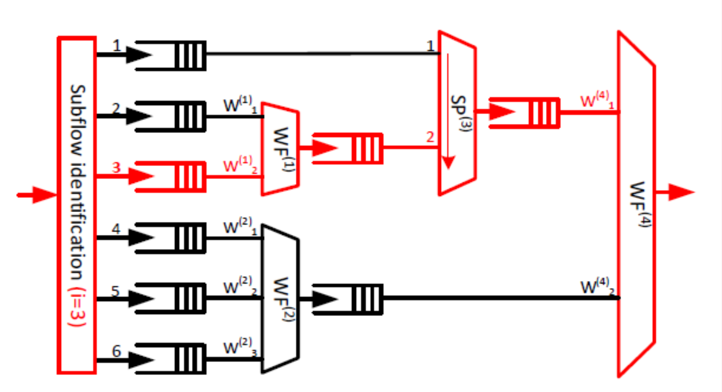
Marking a sub-hierarchy in a single point – less resource intensive

- Such that the outgoing PV distribution is according to the aggregate's TVF
- It also implements the resource sharing of the hierarchy

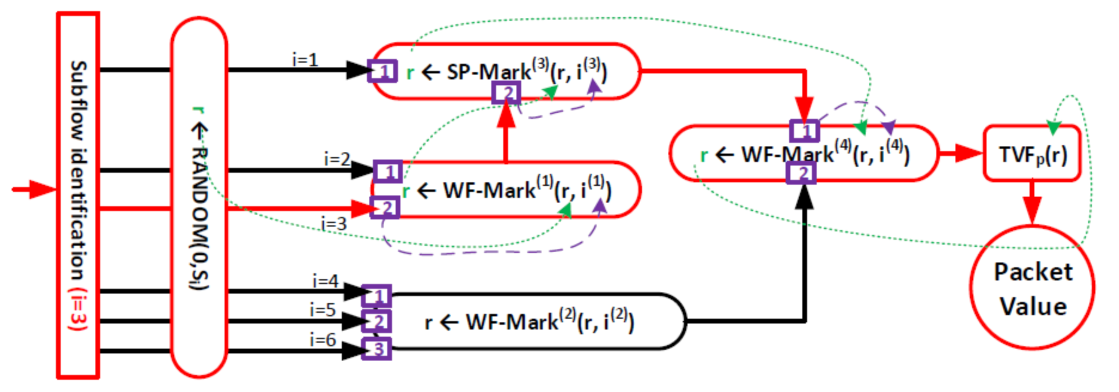


(a) Without flow differentiation. (b) Emulated strict priority scheduling. (c) Emulated weighted-fair scheduling.

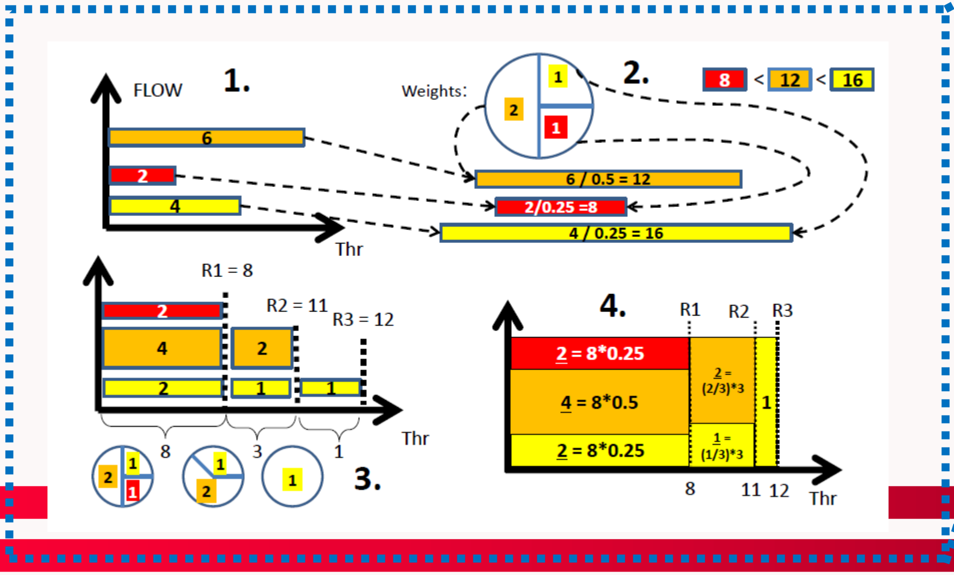
HQOS POLICY MARKING DPDK PROTOTYPE

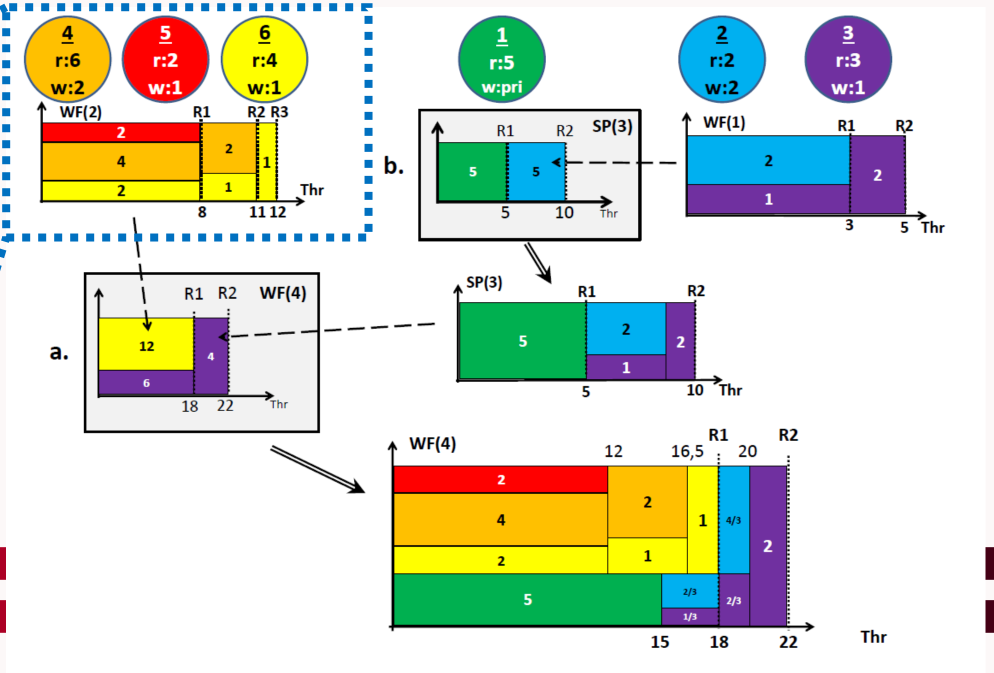


(a) Traditional HQoS.

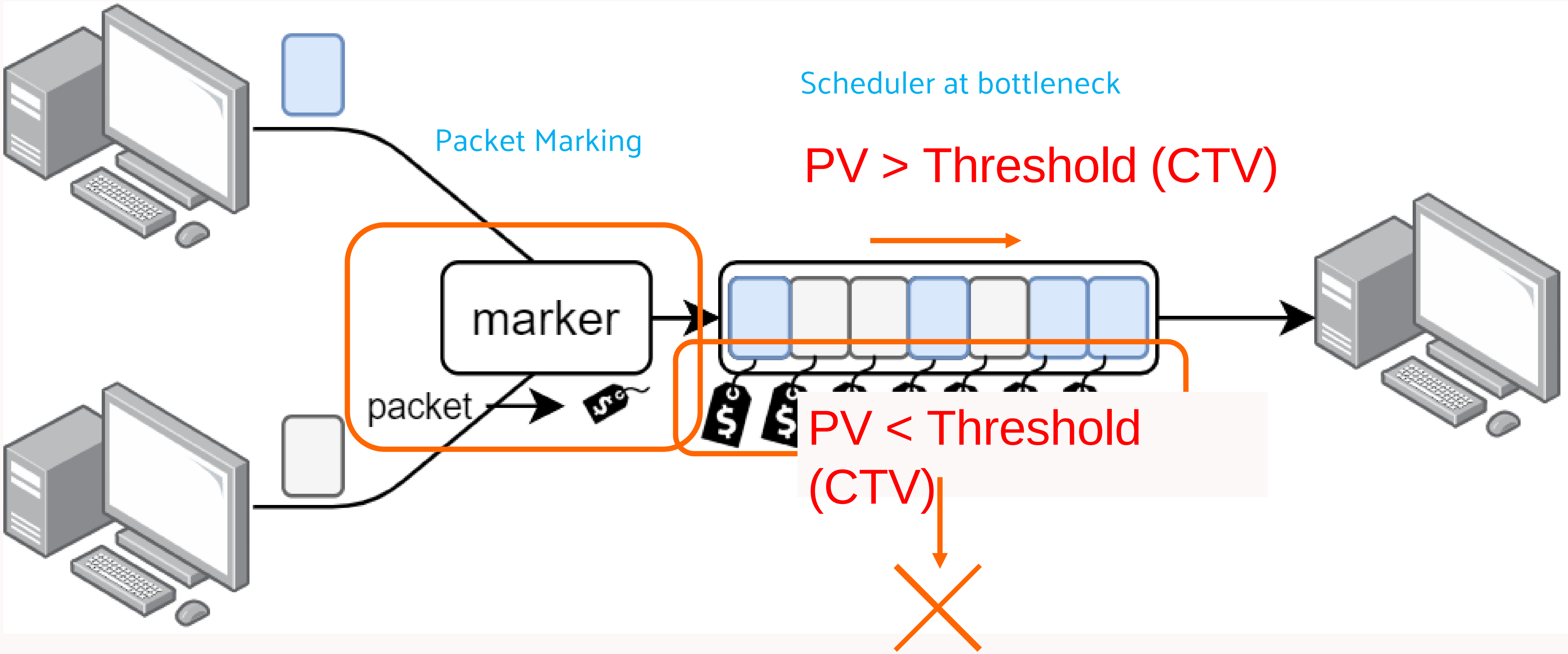


(b) HQoS Packet Marker.



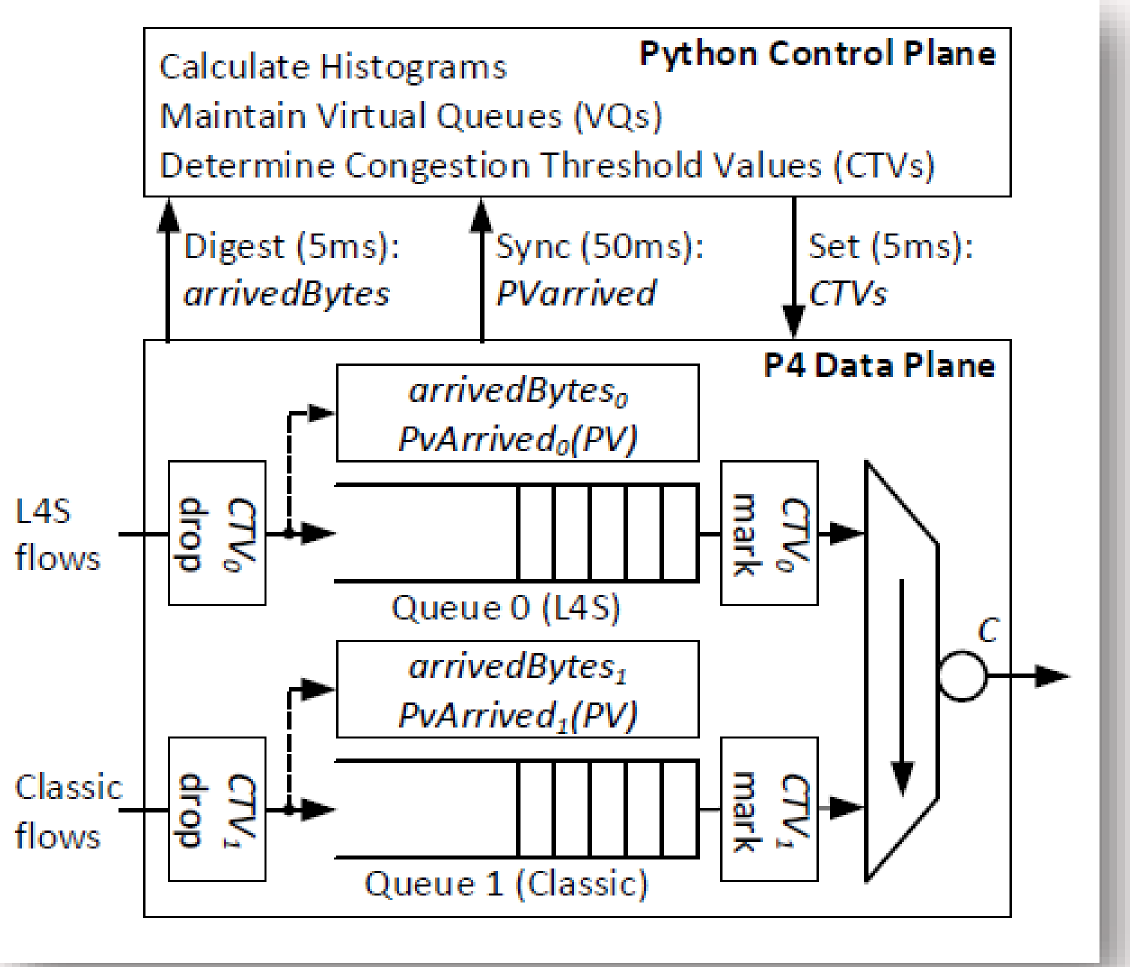


CORE-STATELESS AQM



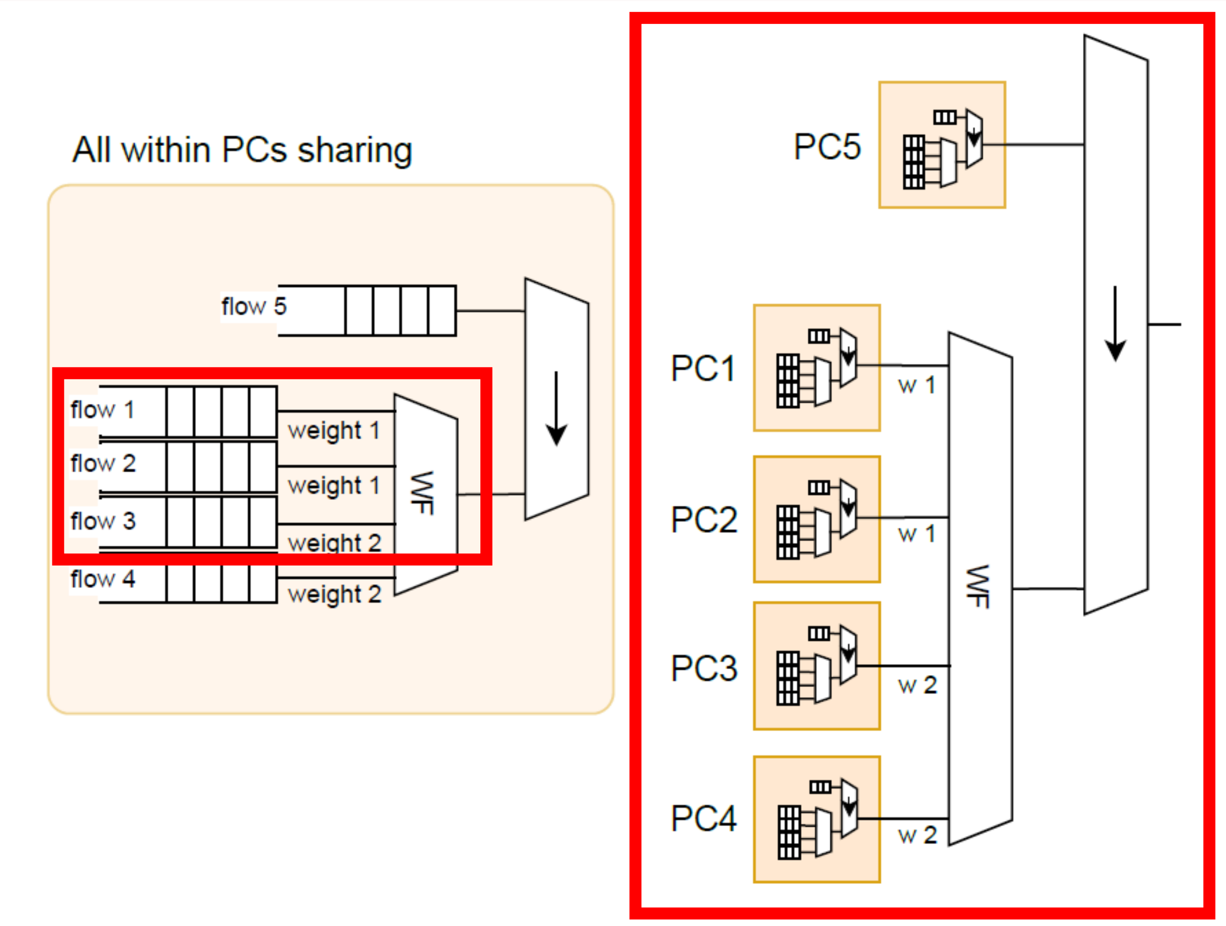
CORE-STATLESS AQM FOR INTEL TOFINO

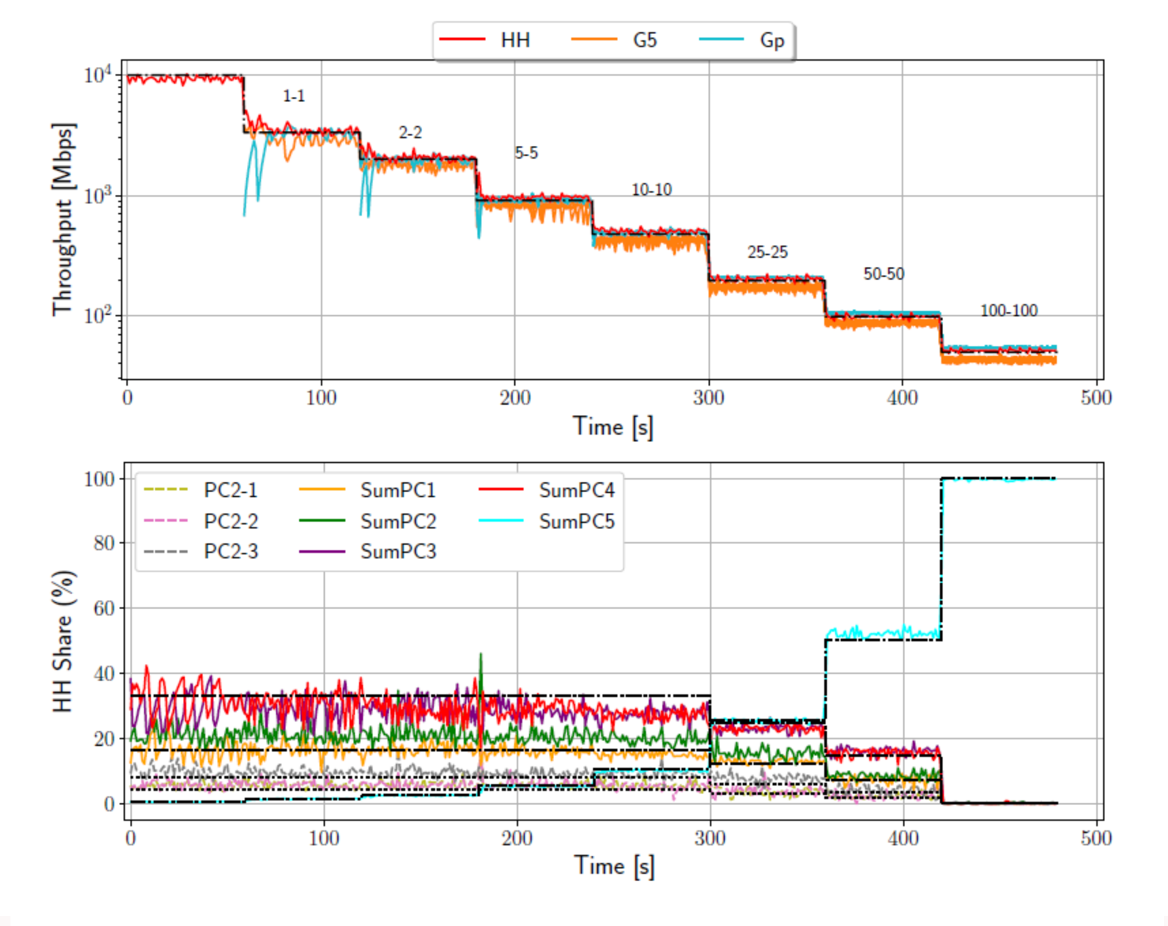
DESIRE6G



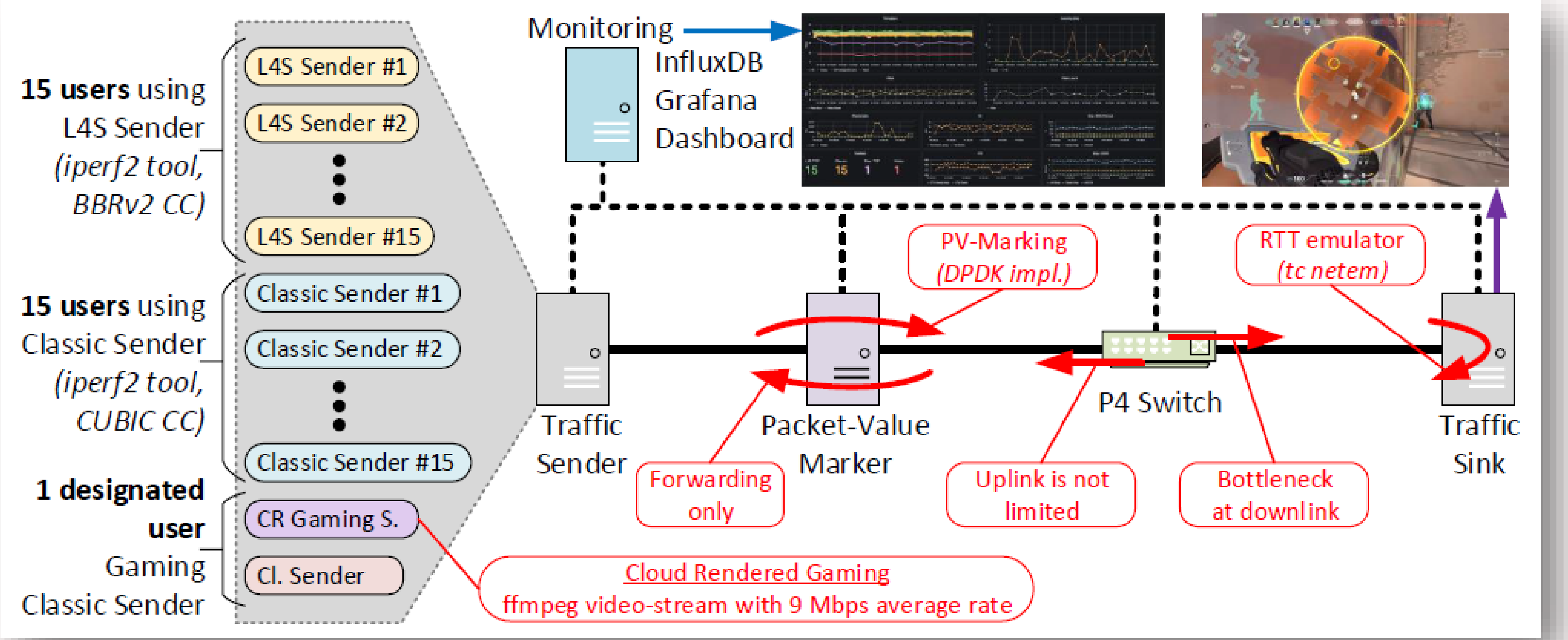
- PV distribution-based cutoff value (CTV) calculation
  - Too complex
- Control plane – Data plane interactions
  - >= 1ms
  - C/C++ CP with BFRT API
- Imbalance between ingress and egress
  - Ingress is overloaded
  - Egress is empty
- 3 stages at both ingress and egress
  - Dropping at ingress
  - ECN marking at egress
- TM engine
  - supports Strict Priority
  - 8 queues per egress
  - Shallow buffers 24MB buffer space in total
  - Non-visible packet losses in th TM engine

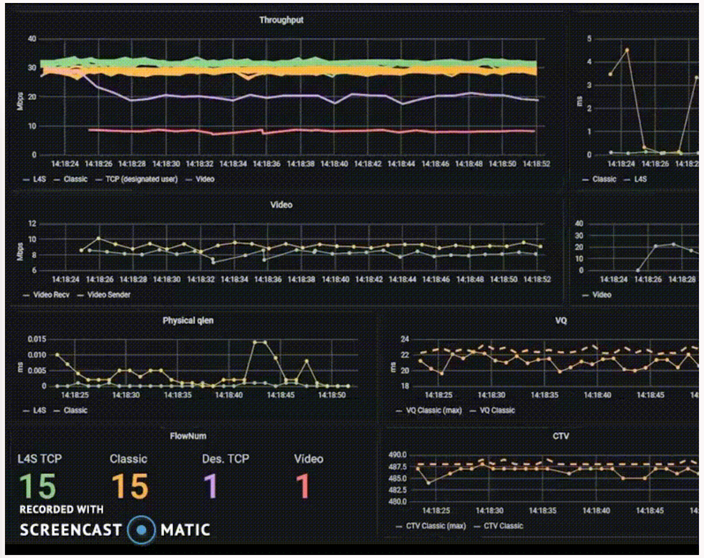
HQOS-MARKING RESULTS





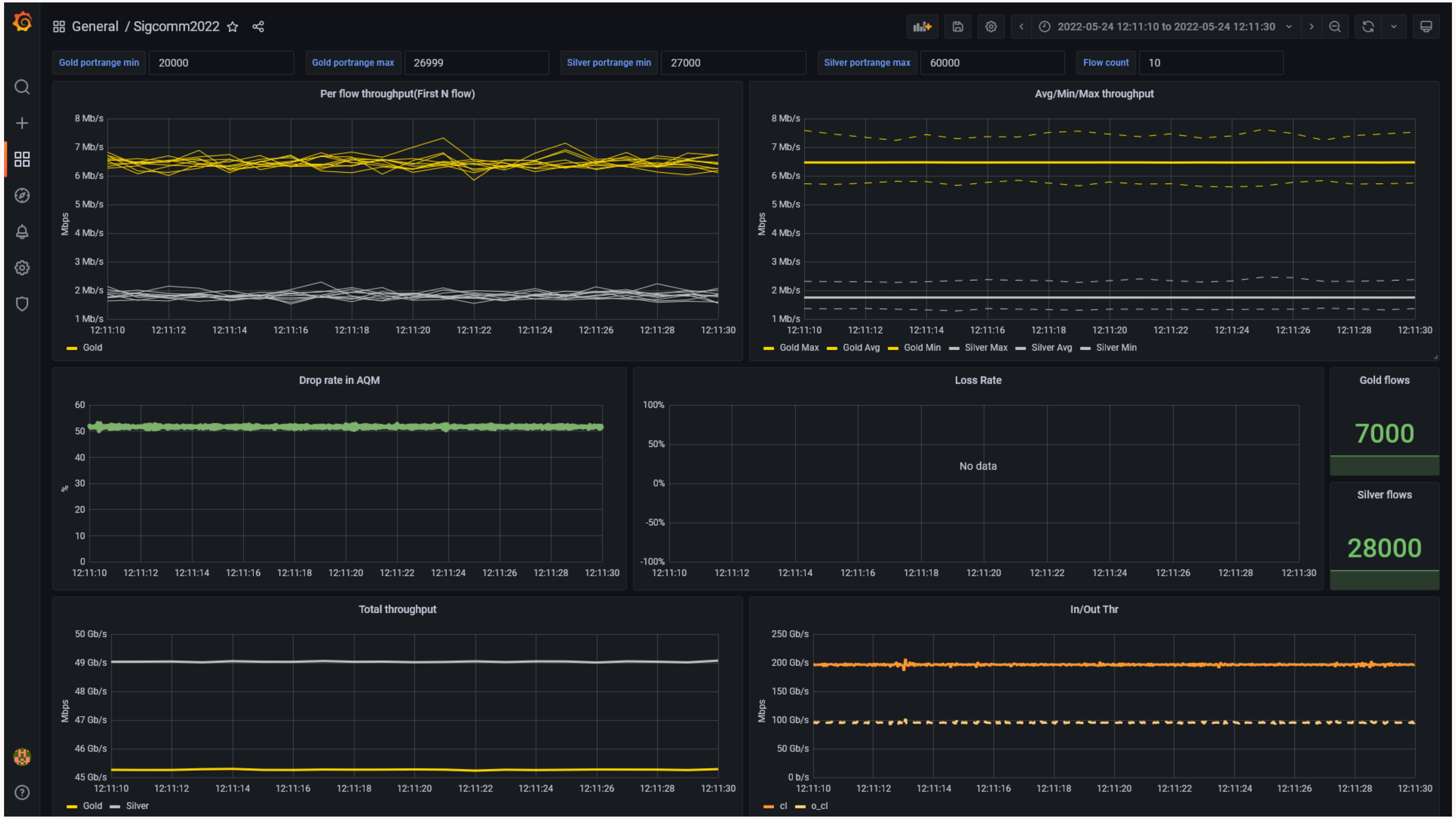
HQOS DEMO EXAMPLE





Queuing delay
L4S avg: 0.1ms, max: 0.4ms
Normal avg: 2ms, max: 3.2ms

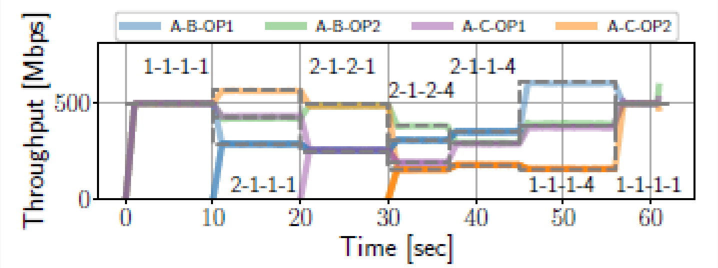
SCALABILITY DEMO EXAMPLE – 35K USERS



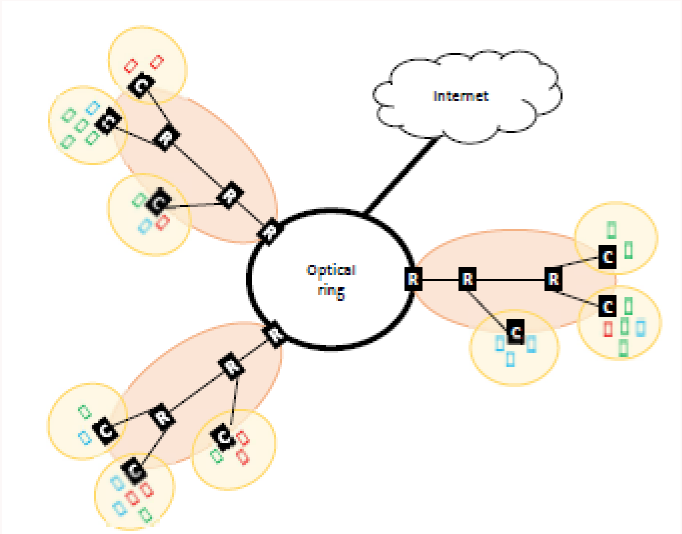
FURTHER PROPERTIES

- Ensures max-min utility fairness on arbitrary topology
- Problem with greedy, non-congestion controlled flows
  - Dead packets occupy network resources
  - Shaping close to the source
- Packet Marking implementation
  - Additional compute resource
  - Distributed SW-based implementation (eBPF/XDP, DPDK) or hardware (P4)
- Specialized AQM
  - All the nodes need to implement (programmable network)
    - Kernel modul, P4/Tofino implementation

FUTURE PLANS



- Extension to non-programmable network domains
  - Mapping to DSCP drop precedences and applying WRED AQM with different profiles
  - Hierarchical max-min utility fairness
    - Ensuring max-min utility fair allocation among slices, subslices of each slice, users of each subslice
- HQoS Marker has high complexity – implemented in DPDK
  - Simplified, less-generic HQoS rules have lower complexity -> Tofino or smartNIC implementation
- Trust in packet values and markings
  - Interface for applications



THANKS!

Sandor Laki

lakis@inf.elte.hu

Co-funded by the European Union

6GSNS

DESIRE6G has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096466.
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.