Towards the Integration of TAPRIO-based Scheduling with Centralized TSN Control

TENSOR 2023

Panagiotis Papadimitriou
University of Macedonia, Greece

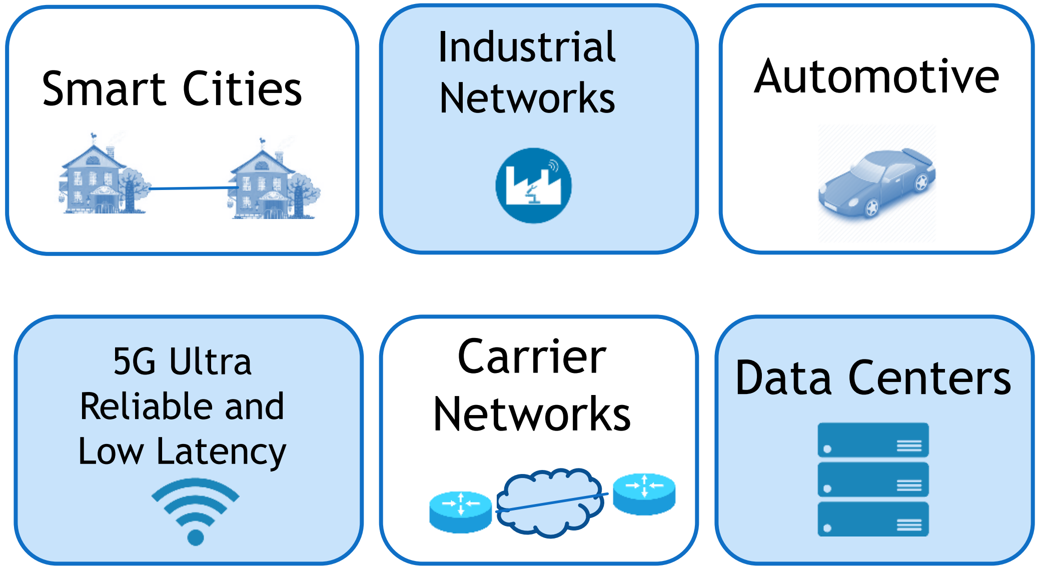
In collaboration with:

George Papathanail, Lefteris Mamatas (University of Macedonia, Greece)

Introduction

- Diverse requirements:
  - Support for diverse operational requirements
  - High throughput
  - Delay guarantees

- Challenges:
  - Complex sender-receiver relationship
  - Wide range of traffic patterns
  - Extreme dynamicity
    - On-demand resource allocation for new user requests
    - Dynamic topology changes
  - Radically different business models



Time-Sensitive Networking (TSN)

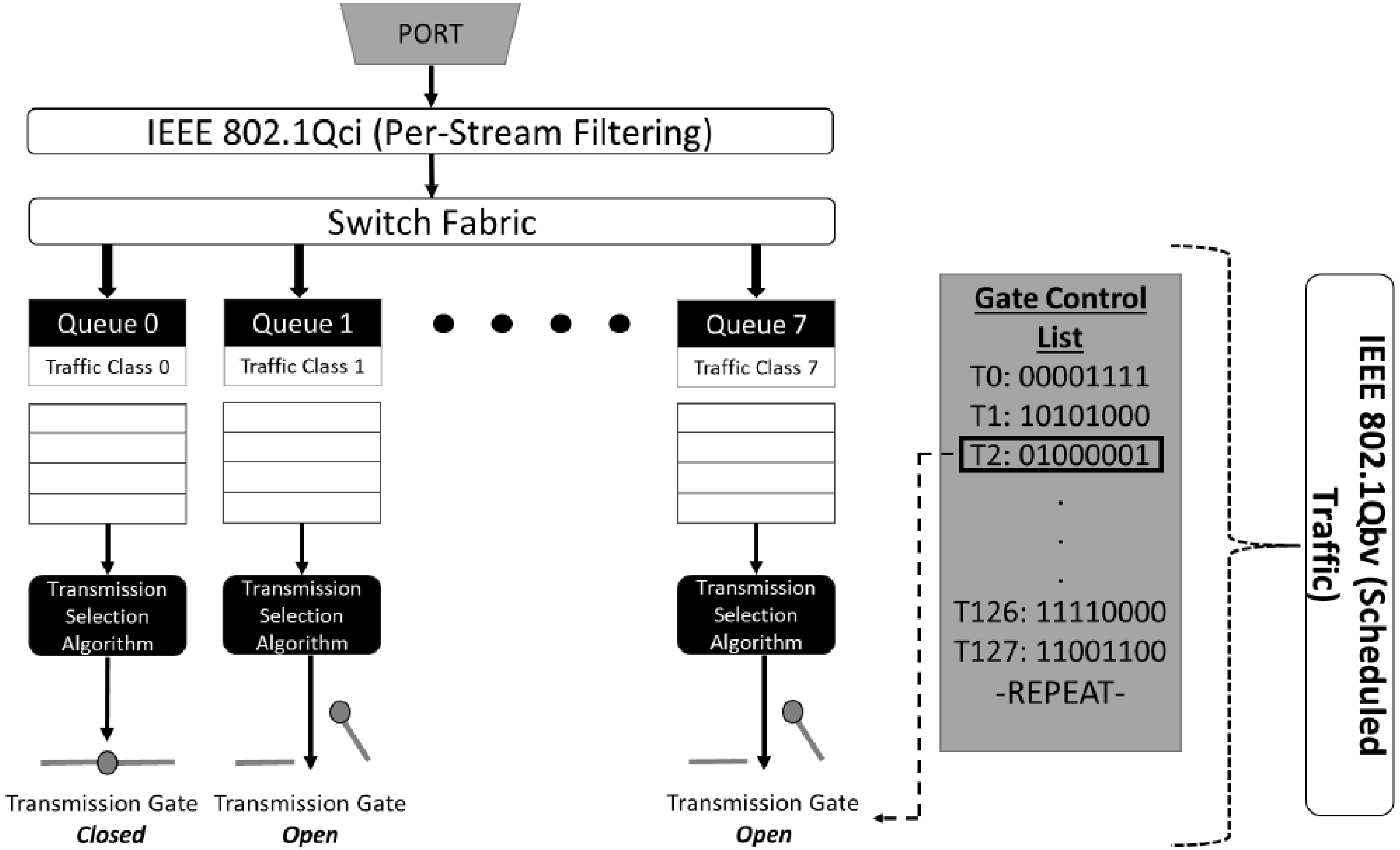
- TSN is a set of IEEE 802 Ethernet sub-standards
  - Network and link layer techniques to achieve:
    - Bounded latency
    - Low delay variation (jitter)
    - Low loss

Category	Standards
Time Synchronization Providing network wide precise synchronization of the clocks of all entities at Layer 2.	IEEE 802.1AS & IEEE 802.1AS-Rev (Network Timing & Synchronization)
Latency & Jitter Separating traffic into traffic classes and efficiently forwarding & queuing the frames in accordance to these traffic classes.	IEEE 802.1Qav (Credit Based Shaping) IEEE 802.1Qbv (Scheduled Traffic) IEEE 802.3br & IEEE 802.1Qbu (Frame Preemption) IEEE 802.1Qch (Cyclic Queuing) IEEE 802.1Qcr (Asynchronous Traffic Shaping)
Reliability & Redundancy Maintaining network wide integrity by ensuring path redundancy and ingress queue policing.	IEEE 802.1CB (Frame Replication & Elimination) IEEE 802.1Qca (Path Control & Reservation) IEEE 802.1Qci (Per-Stream Filtering)
Resource Management Providing dynamic discovery, configuration and monitoring of network in addition to resource allocation & registration.	IEEE 802.1Qat & IEEE 802.1Qcc (Stream Reservation) IEEE 802.1Qcp (YANG Models) IEEE 802.1CS (Link-Local Reservation)

- Our focus on this study:
  - Scheduled Traffic (**IEEE 802.1Qbv**)
    - Guarantees worst case latency

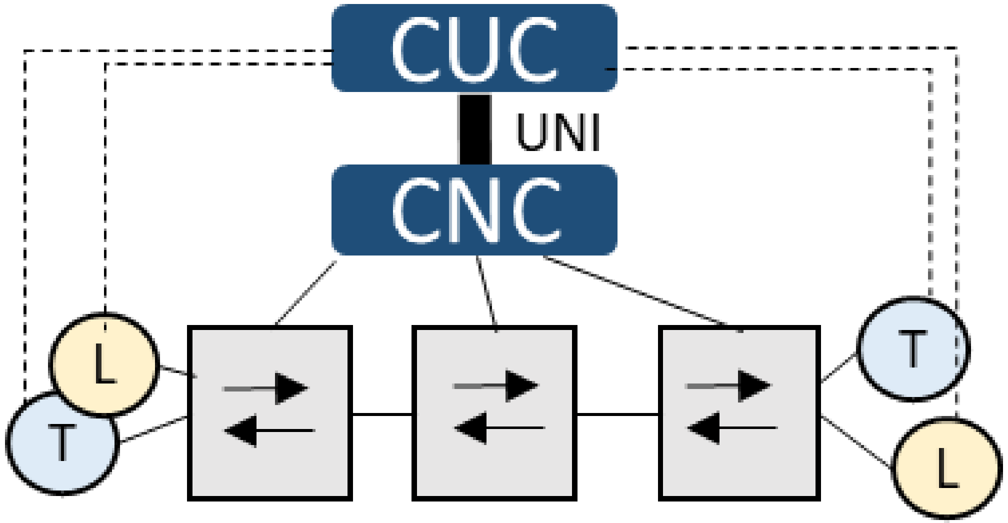
IEEE 802.1Qbv

- IEEE 802.1Qbv introduces a transmission gate operation for each queue
- Transmission gates are controlled by a Gate Control List (GCL)

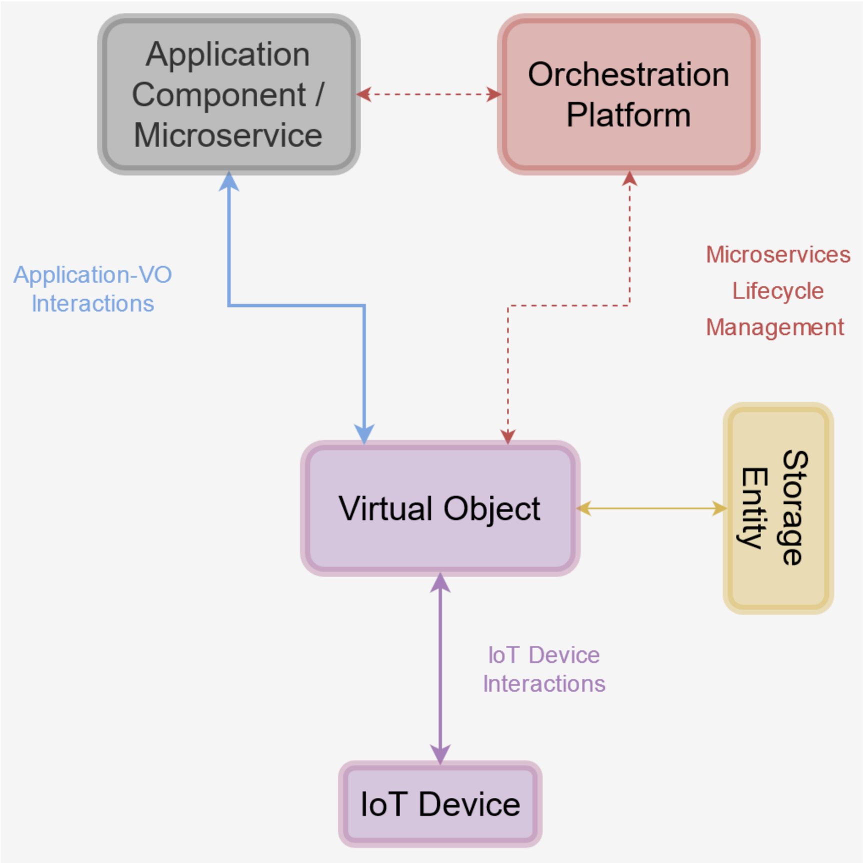


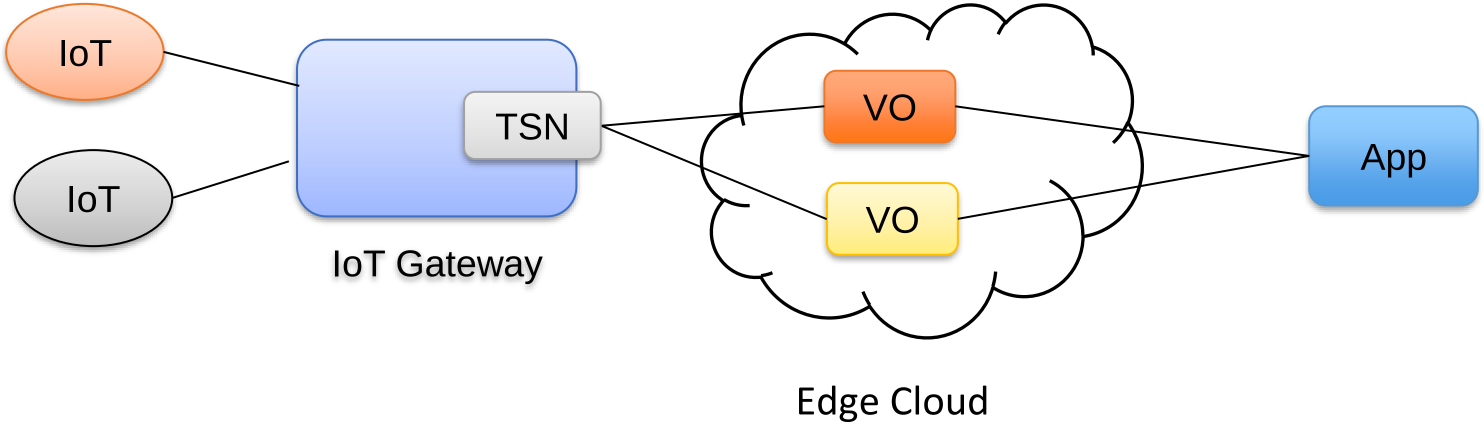
Centralized Network Controller (CNC)

- Fully centralized model of TSN control plane
  - CUC collects and conveys all flow requirements from talkers (IEEE 802.1Qdj)
  - CNC is responsible for TSN schedule generation based on a network-wide view



TSN for IoT-to-VO Communication





nephele

Objectives

- Various TSN aspects mandate TSN platforms for experimentation:
  - Interaction of TSN with network orchestrators
    - Translation of high-level flow requirements or intents into GCL
  - Synchronization among talker and TSN bridges

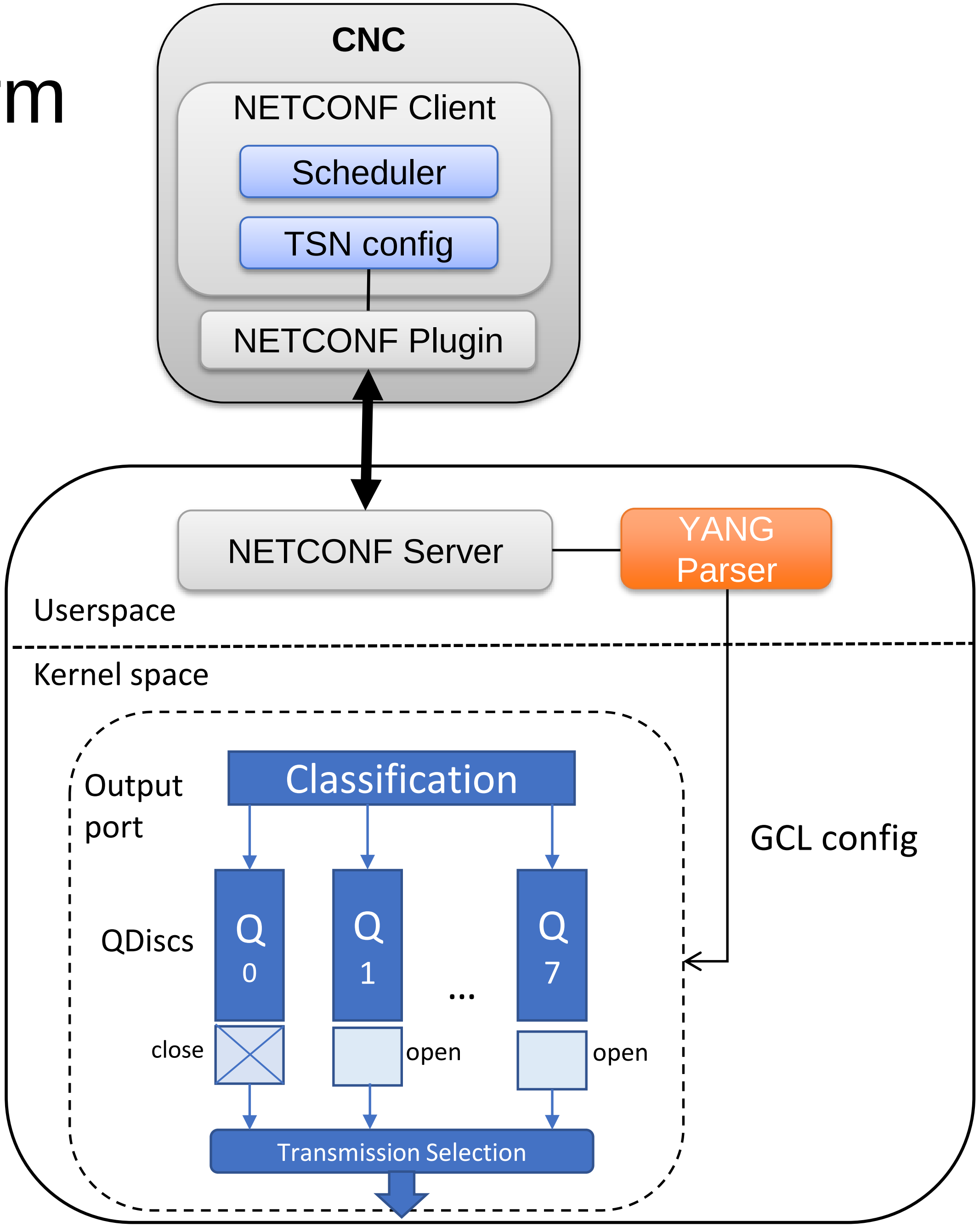
- Experimental environments under consideration:
  - TSN testbed
  - Mininet

- Main Goal:
  - Integration of TSN scheduler with centralized TSN control
    - TAPRIO
    - CNC

TSN Platform

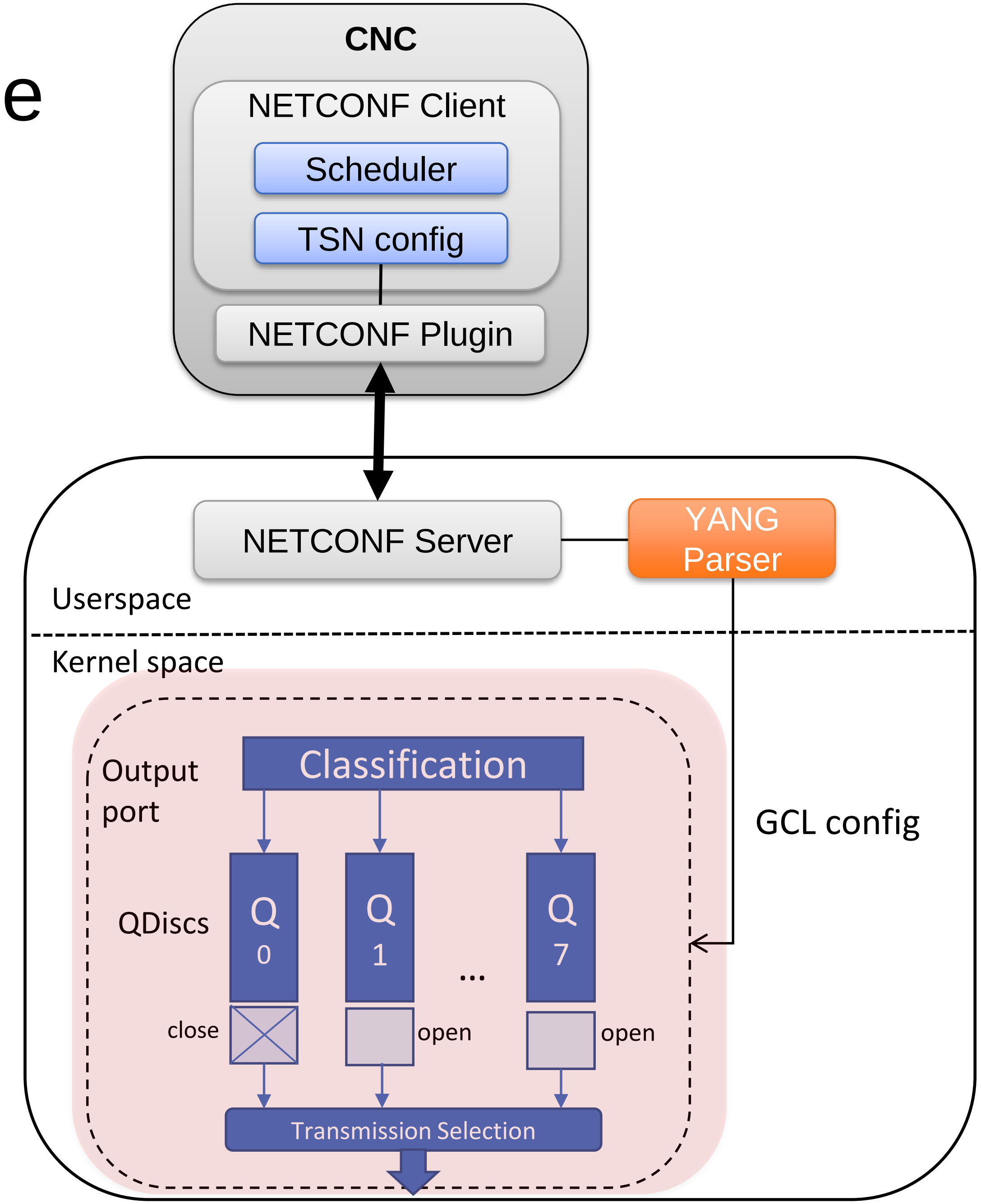
TSN Platform

- TSN platform components:
  - TAPRIO-based TSN datapath
  - TSN control plane (CNC)
  - NETCONF for CNC-TSN interactions



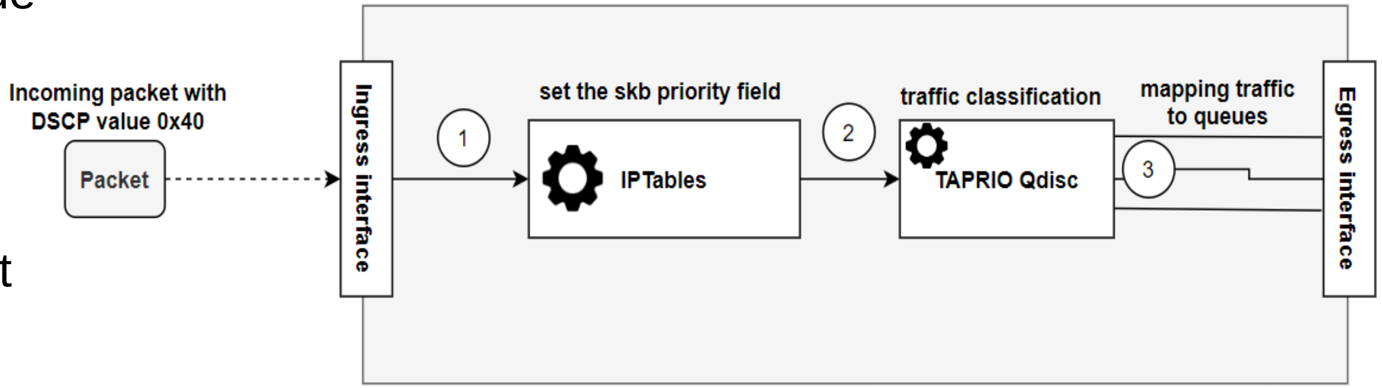
Data Plane

- TAPRIO-based datapath:
  - Packet classification to a specific traffic class via the **priority** field of the socket buffer (**skb**)
- Traffic class-to-queue mapping:
  - **DSCP** field of the packet header using IPv6
- TAPRIO activation:
  - Linux **tc qdisc**
  - Modification of **skb** priority field through **iptables**



Data Plane

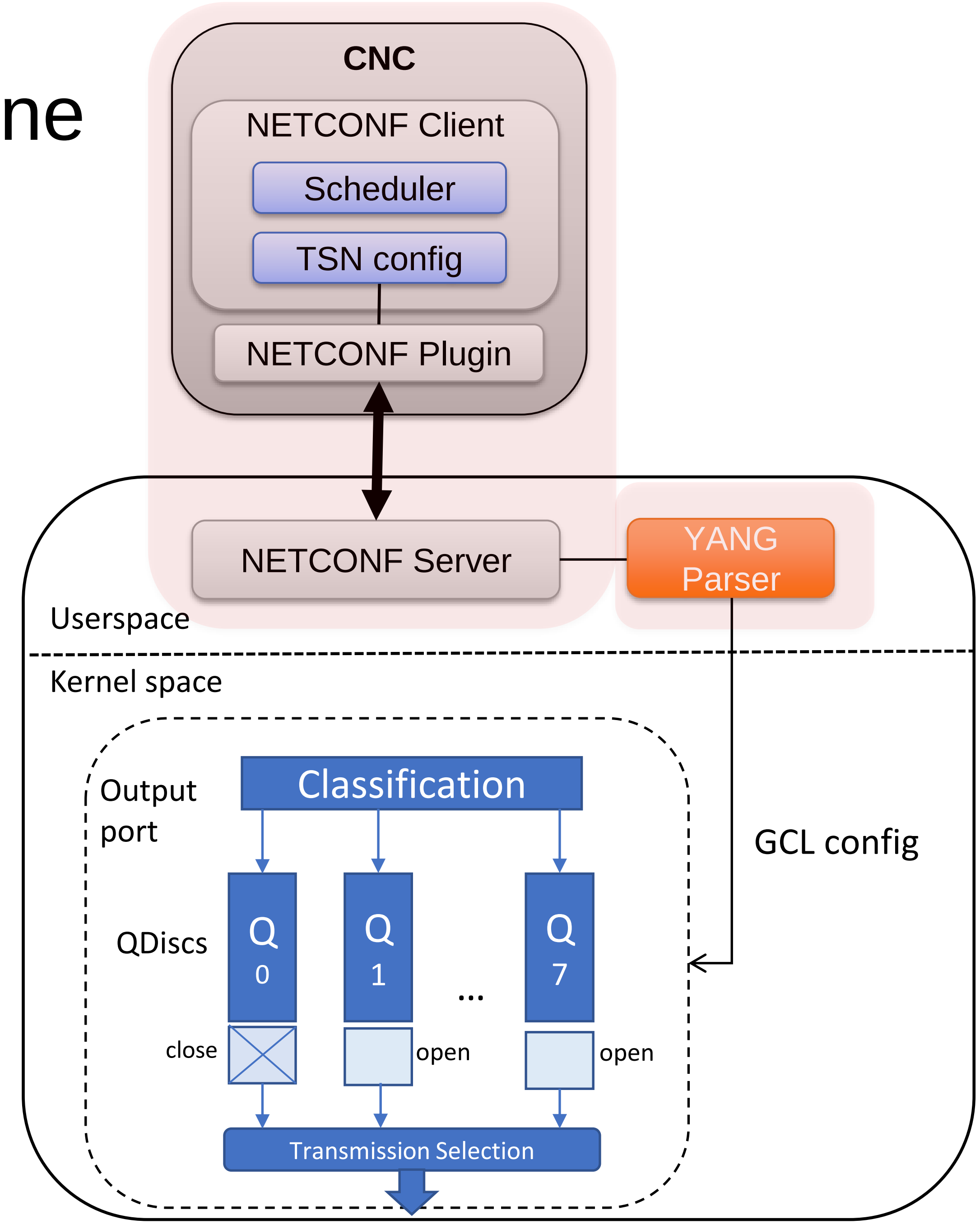
- Workflow:
  - Incoming packet marked with DSCP value 0x40 reach the ingress interface
  - First classification using IPTables
    - set the skb priority field (0x40)
  - TAPRIO qdisc maps the incoming packet to queues



Chain POSTROUTING (policy ACCEPT)					
Target	Prot	Source	Destination	Match	Action
CLASSIFY	All	Anyware	Anyware	DSCP match 0xXX	CLASSIFY set 0:Y

Control Plane

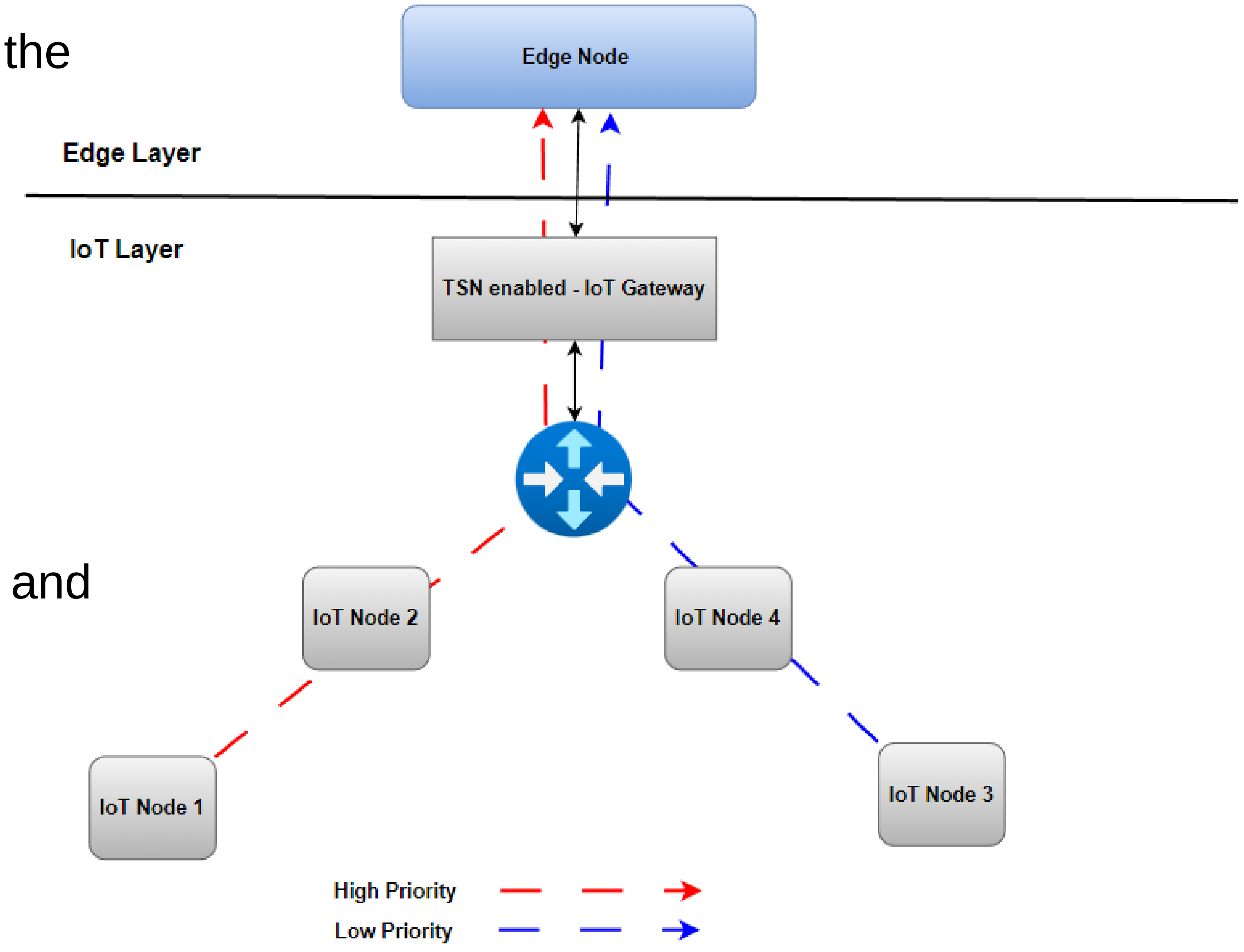
- CNC:
  - computes TSN 802.1Qbv schedules
  - communicates with TAPRIO via the NETCONF plugin
    - YANG-TSN model
- YANG parser:
  - parses YANG-TSN models to a set of actions that can be applied directly to the queuing disc layer of the Linux kernel



Experimental Evaluation

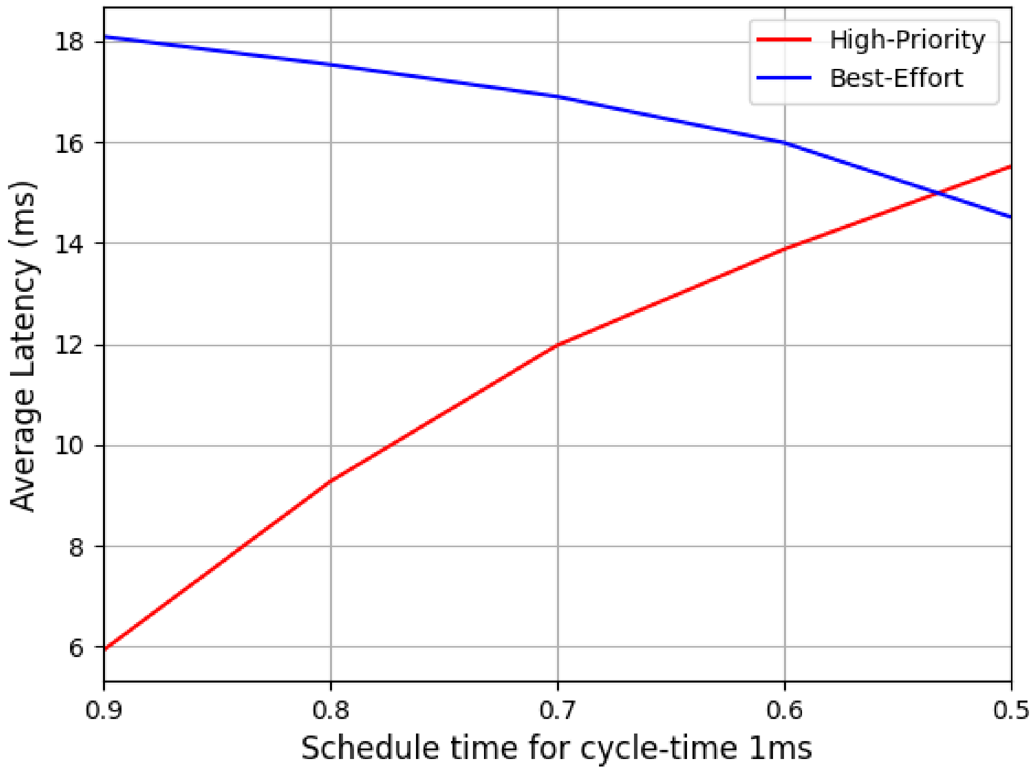
Experimental Setup

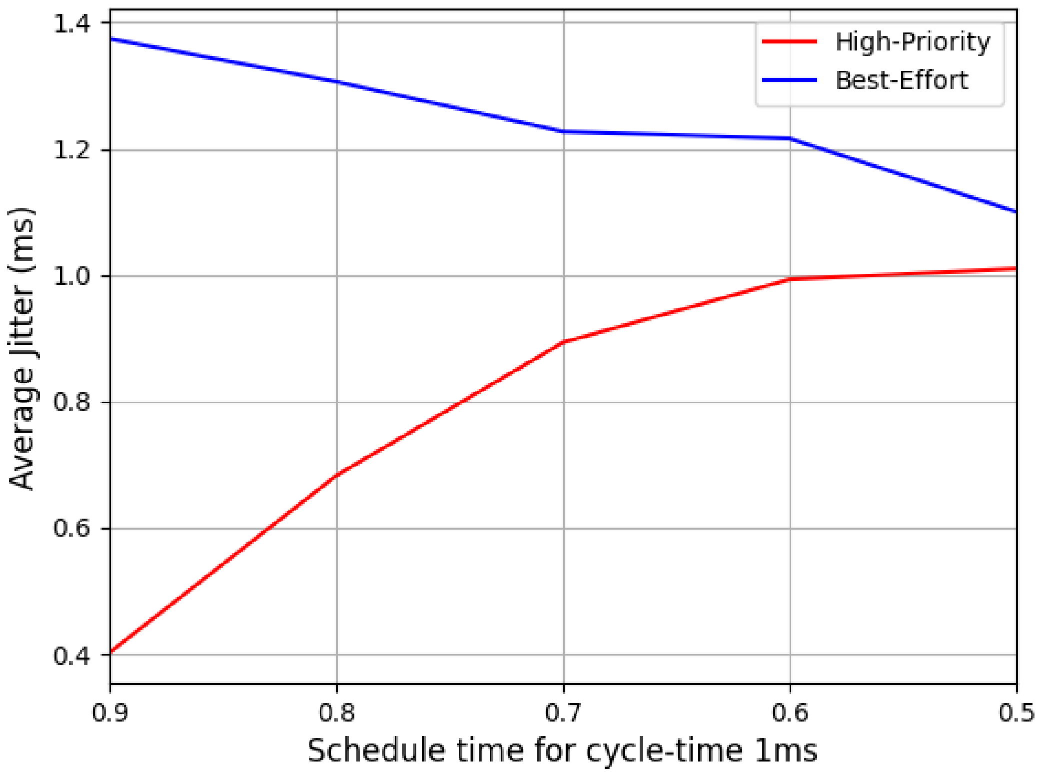
- TAPRIO activated on the egress port of the IoT Gateway.
- Extended version of Mininet on Ubuntu 20.04.1 LTS:
  - Support for up to 8 TX/RX queues
  - IPMininet for IPv6 addressing
- Tests conducted on a VM with 8 vCPUs and 8GB RAM



Impact of Diverse TAPRIO Schedules on Latency/Jitter

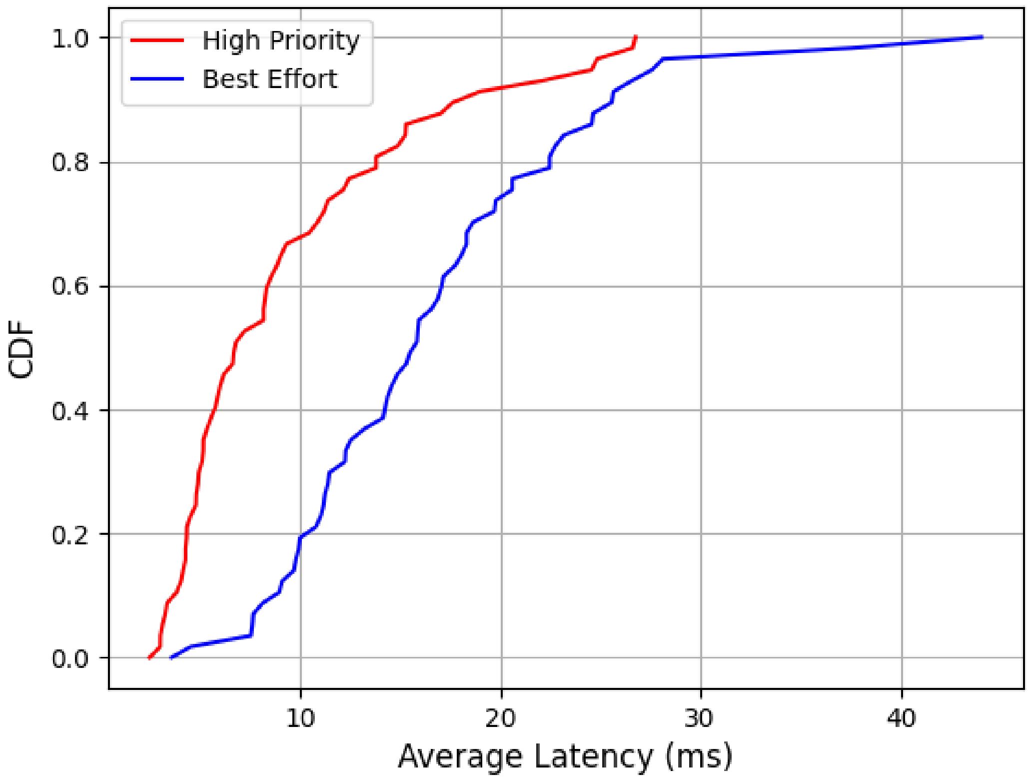
- **High-priority** and **best-effort** traffic with 1440-byte packets at 2000 packets/sec using iperf
  - High-priority traffic matched on DSCP field 0x40
  - Best-effort traffic matched on DSCP field 0x00
- Tests with diverse schedules on a cycle time of 1 ms

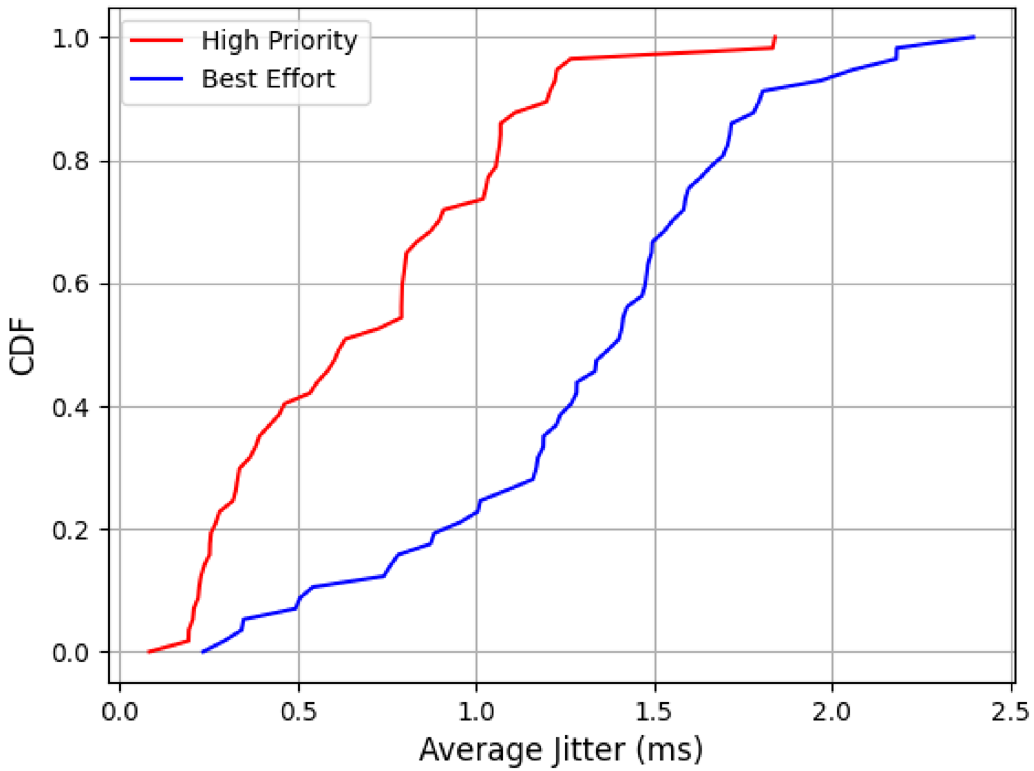




Impact of TAPRIO 800:200

- High priority and best-effort traffic with 1440-byte packets at 2000 packets/sec using Iperf
  - High-priority traffic matched on DSCP field 0x40
  - Best-effort traffic matched on DSCP field 0x00
- Tests with TAPRIO 800:200 a cycle time of 1 ms

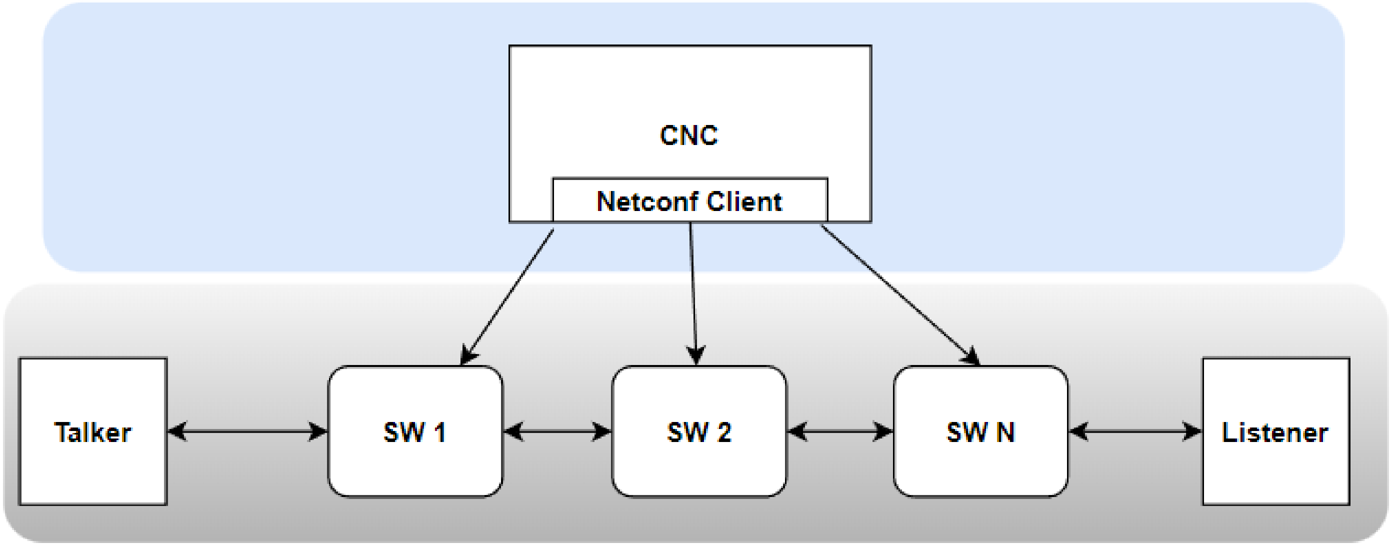




Control Communication Overhead

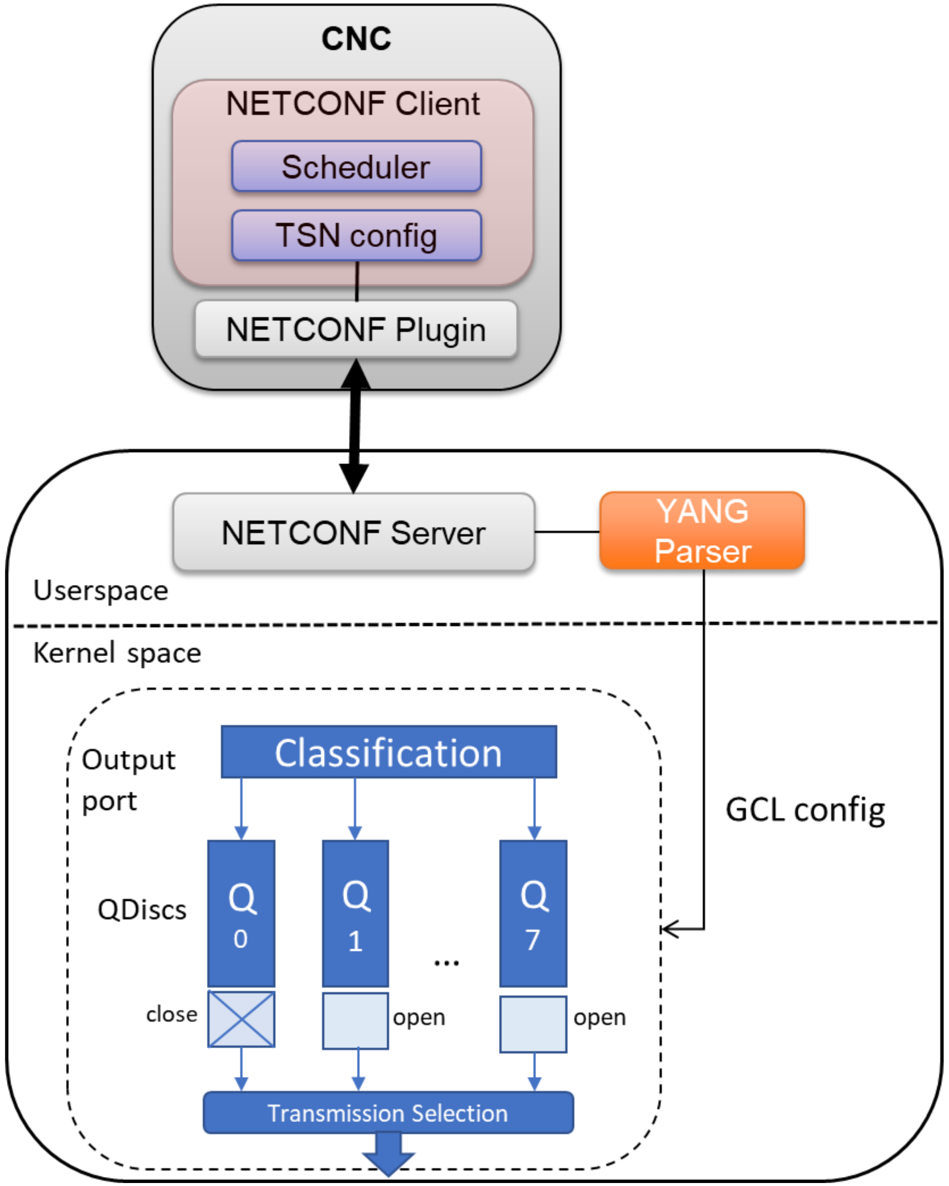
- Interaction between CNC and TAPRIO
  - Communication overhead for TSN schedule population into TAPRIO

- Experiments with a Talker-Listener pair and 1-10 TSN bridges



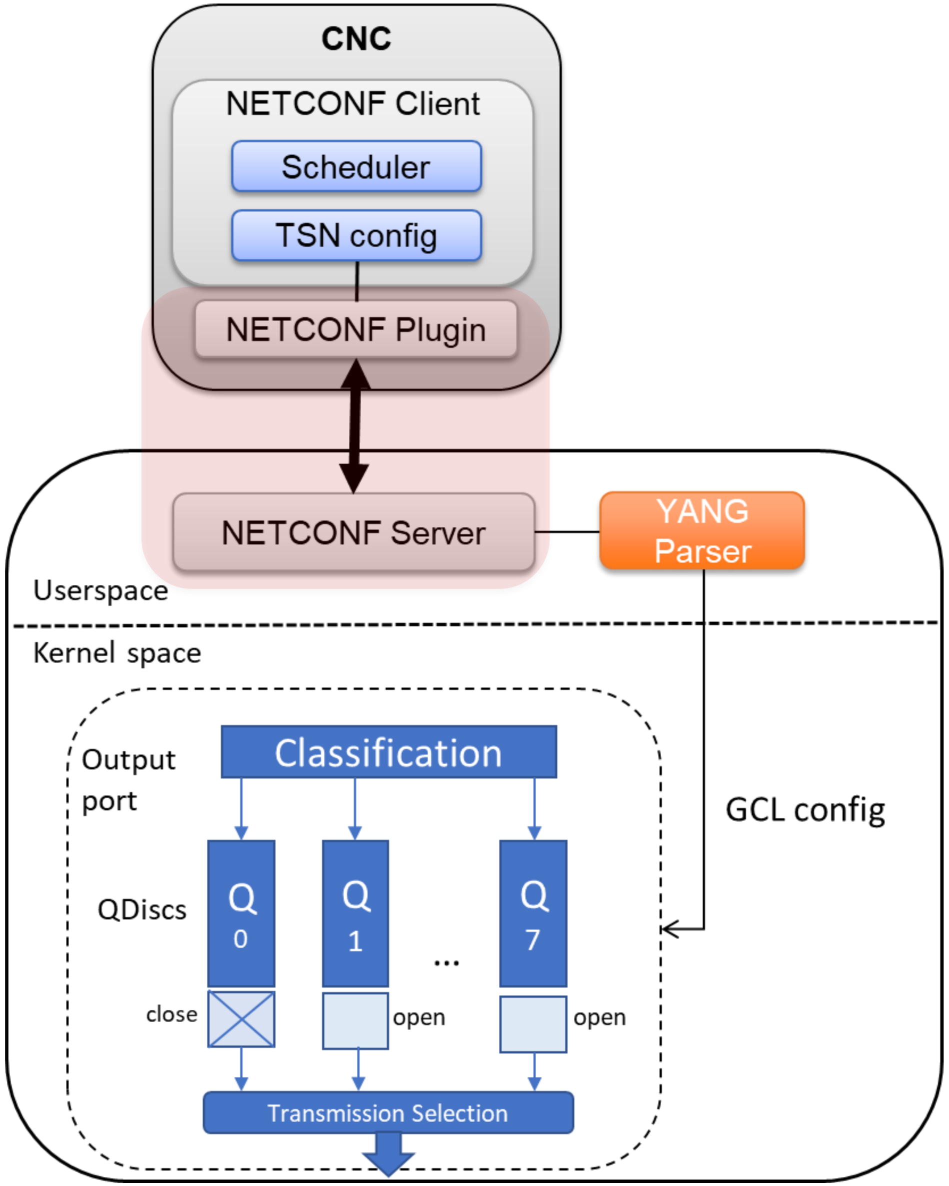
Control Communication Overhead

- Main steps for CNC-TAPRIO interaction
  - TSN schedule configuration



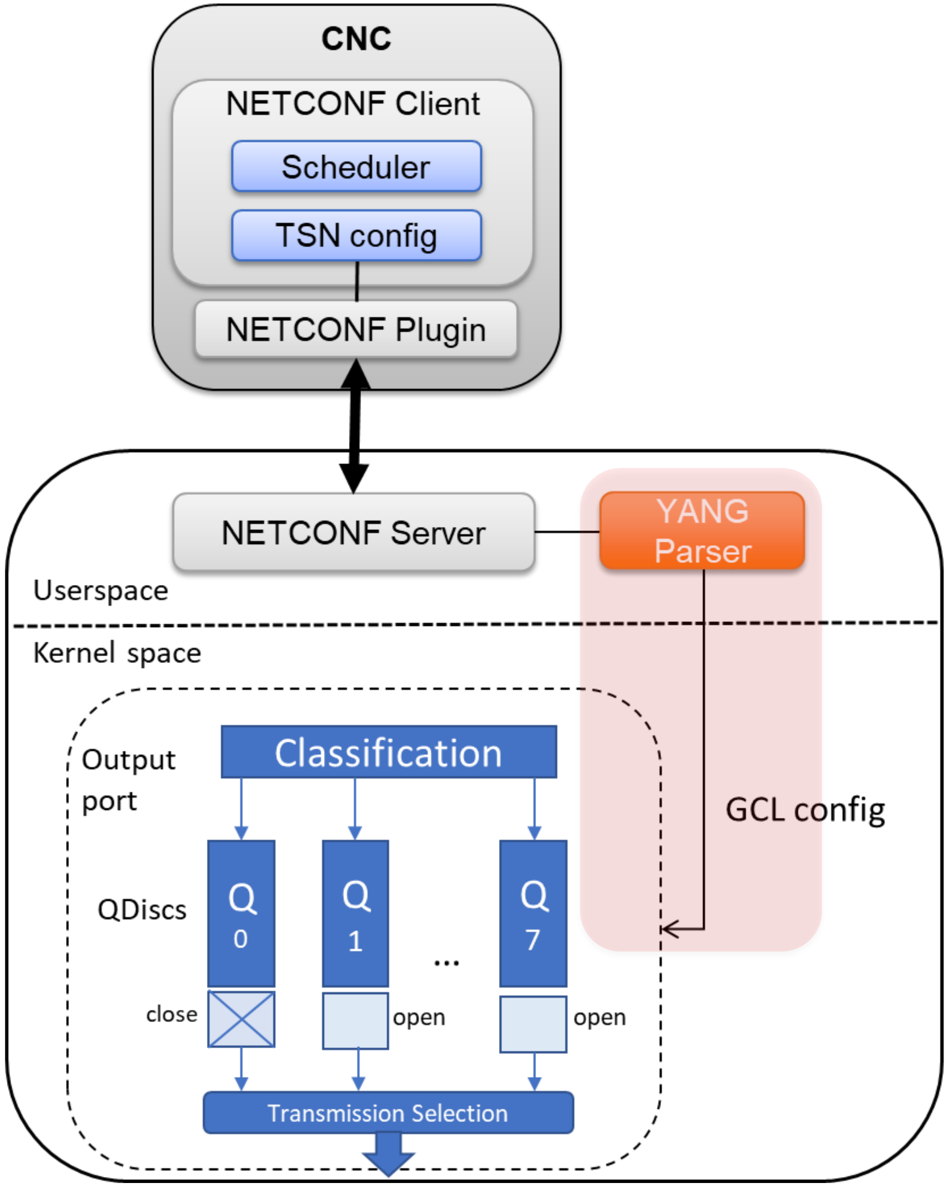
Control Communication Overhead

- Main steps for CNC-TAPRIO interaction
  - TSN schedule configuration
  - Communication via NETCONF



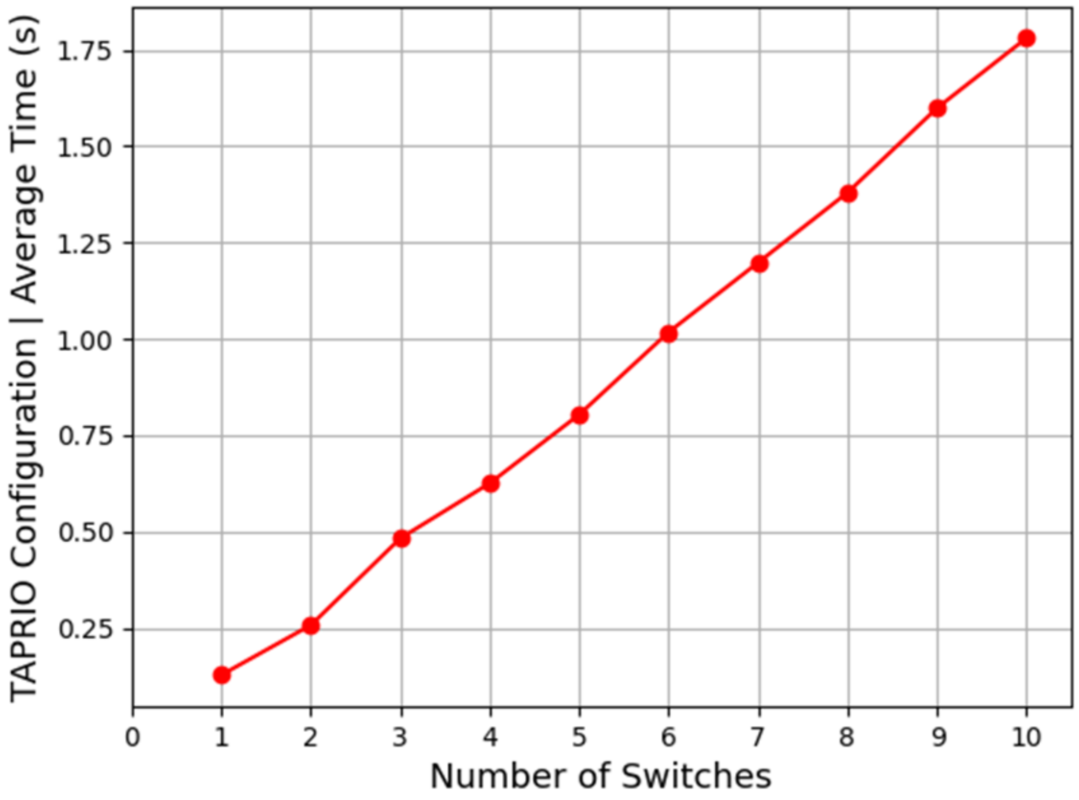
Control Communication Overhead

- Main steps for CNC-TAPRIO interaction
  - TSN schedule configuration
  - Communication via NETCONF
  - YANG-TSN parsing



Control Communication Overhead

- Control communication delay is dominated by TSN schedule generation within CNC
  - Total delay is low and scales well with the number of TSN switches



Switches	TAPRIO Configuration Generation (sec)	NETCONF Communication (sec)	YANG Parsing (sec)
1	0,128	0,026	0,019
2	0,257	0,024	0,019
3	0,482	0,025	0,020
4	0,625	0,025	0,020
5	0,803	0,022	0,021
6	1,017	0,024	0,021
7	1,198	0,025	0,020
8	1,379	0,024	0,020
9	1,599	0,025	0,022
10	1,781	0,027	0,021

Conclusions

Conclusions

- TSN platform for experimentation with TSN mechanisms
  - TAPRIO-based bridge for prioritization of scheduled traffic
  - CNC for TSN schedule generation

- Initial performance/feasibility tests
  - Prioritized traffic experiences reduced latency and jitter
  - Low control communication overhead during the interaction of CNC with TAPRIO

- Future work:
  - Interoperability of TSN with orchestration platforms (e.g., NEPHELE) for the deployment of hyper-distributed applications
    - Translation of high-level network requirements/intents into low-level GCL configurations

Thank you!

Panagiotis Papadimitriou

E-mail: papadimitriou@uom.edu.gr
WWW: http://netcloud.uom.gr/