Universität Rostock Traditio et Innovatio

On Hard Realtime Traffic in Converged Time-Sensitive Networks

Willi Brekenfelder, Helge Parzyjegla, Peter Danielis
Gero Mühl, Fabian Kummer, Eike Schweissguth,
Frank Golatowski

Architecture of Application Systems
Institute of Computer Science
University of Rostock

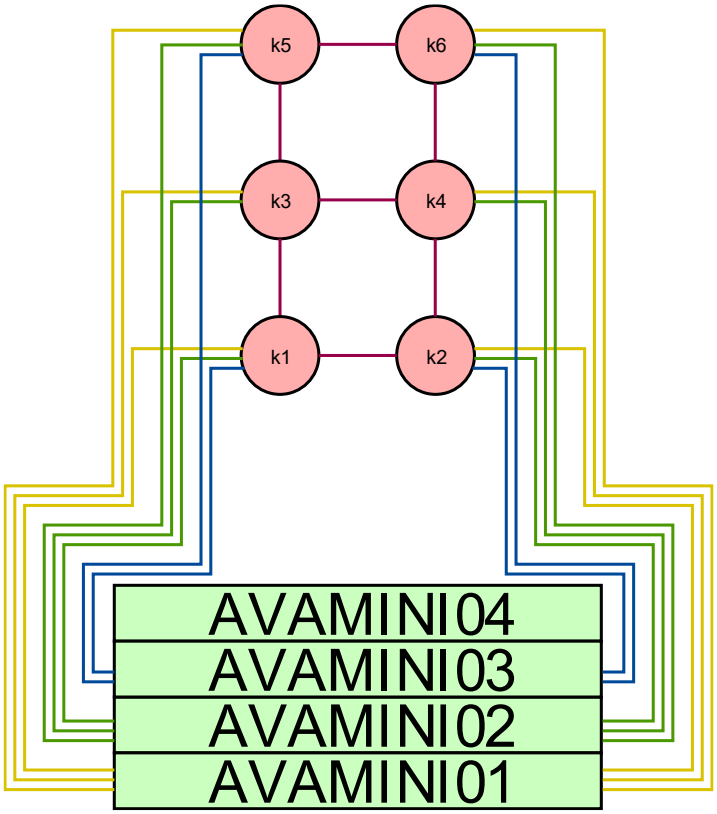
Motivation

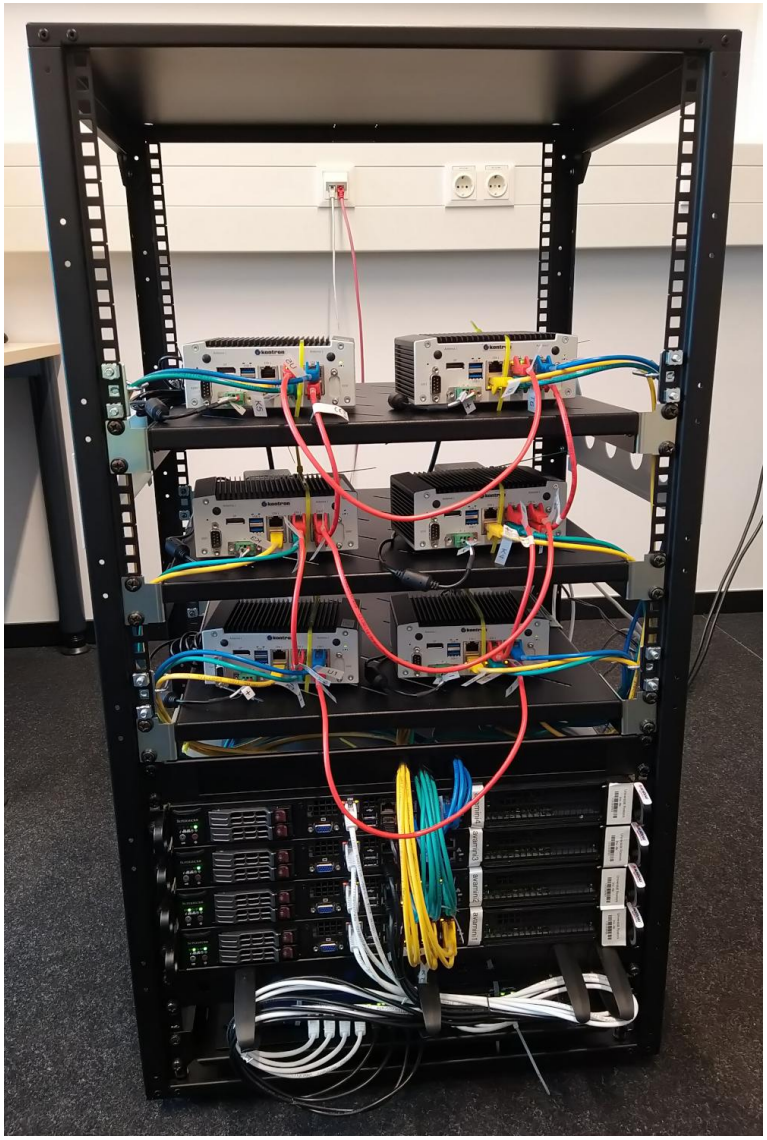
- Multiple different TSN-switches from different vendors
  - Advantech, Cisco, FibroLAN, Hirschmann, InnoRoute, Kontron, Marvell…
- In theory those switches should work according IEEE 802.1Q

- Practice
  - Time synchronization between different hardware
  - Accuracy of planned and actual sending time at source
  - Practical function of theoretical planned schedules
  - Standard compliance of consumer of the shelf switches

- Agenda
  1. Testbed Hardware
  2. Software
  3. Case Study
  4. Results

Hardware





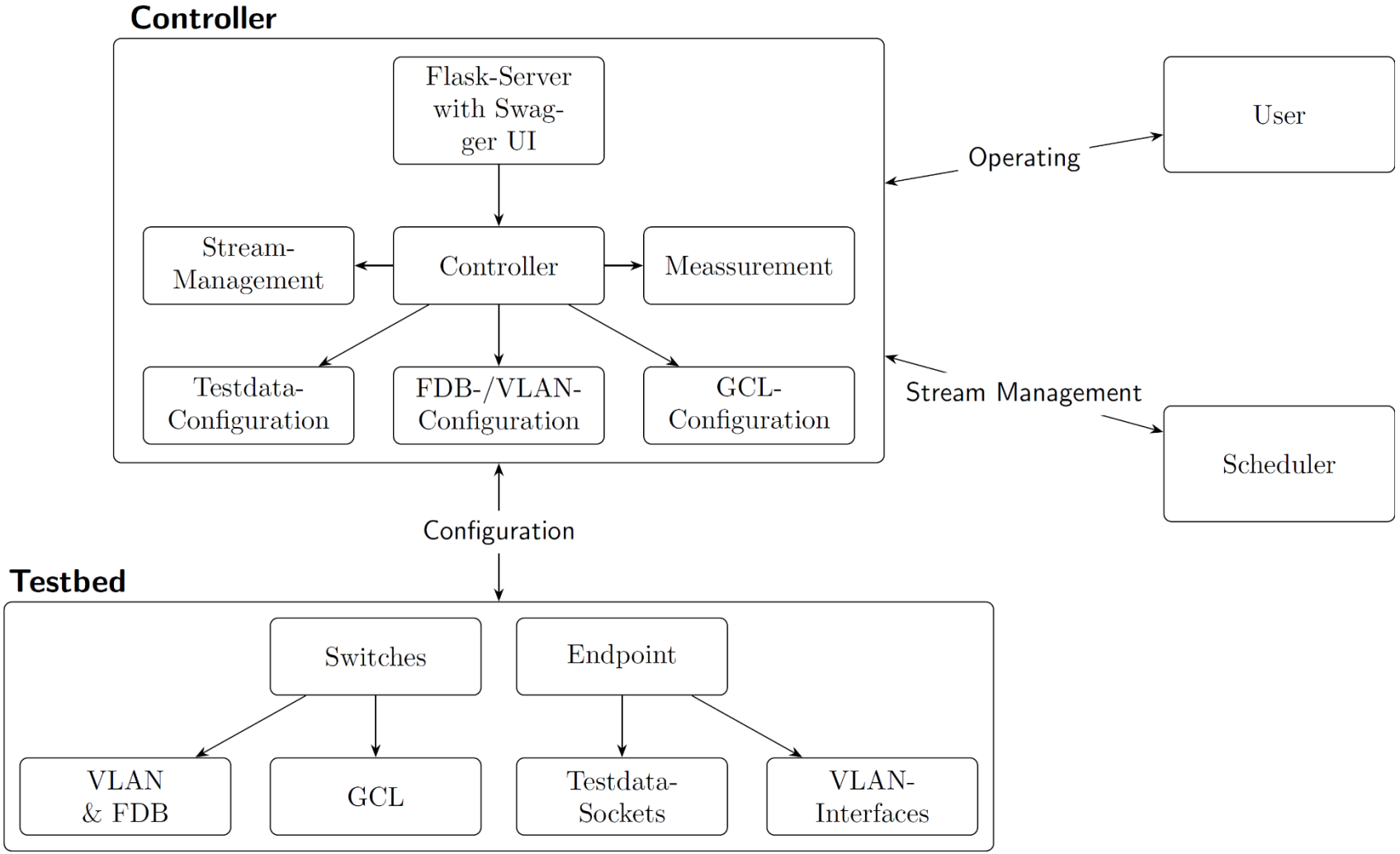
- 6 switches (Kontron KBox A-230-LS)
- 4 servers (Supermicro SYS-1019C-FHTN8 with Intel I210)

Hardware

- Controller on server AVAMINI01
- Sending and receiving servers AVAMINI02 & AVAMINI03
- Spare server AVAMINI04

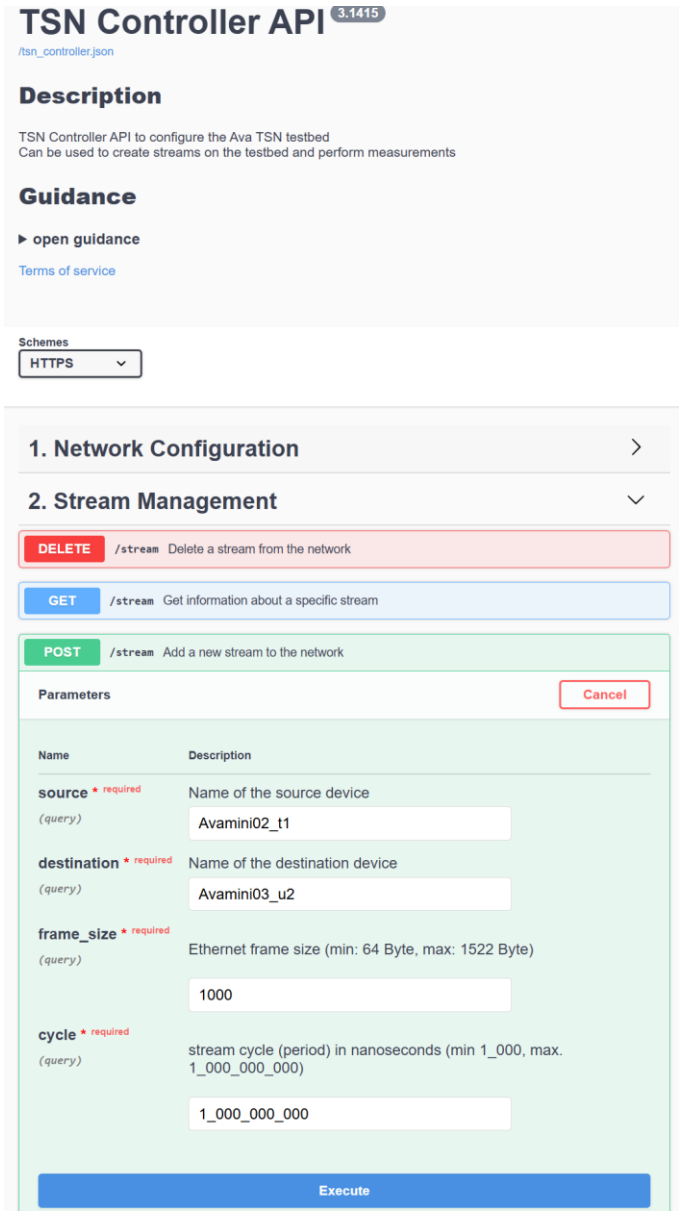
- Advantages of this topology
  - Sending and receiving with same or different devices
  - Easy setup for different topologies (subset of ring with intersection)
  - Redundant devices
  - Physical separation of different traffic generators possible
  - Paths with multiple hops over TSN-devices

Software


Controller
Flask-Server with Swag-ger UI
Stream-Management
Controller
Meassurement
Testdata-Configuration
FDB-/VLAN-Configuration
GCL-Configuration
Operating
User
Stream Management
Scheduler
Configuration
Testbed
Switches
Endpoint
VLAN & FDB
GCL
Testdata-Sockets
VLAN-Interfaces

Software – Swagger UI

- Network Configuration
  - Get / Set Topology
  - Mode of switches
  - Reservation mode
  - Guard bands
- Stream Management
  - Add / delete streams
- Controller Configuration
  - Get active scheduler
  - Mode of switches
- GCL Monitoring
- Measurement
  - Start / delete / download measurements

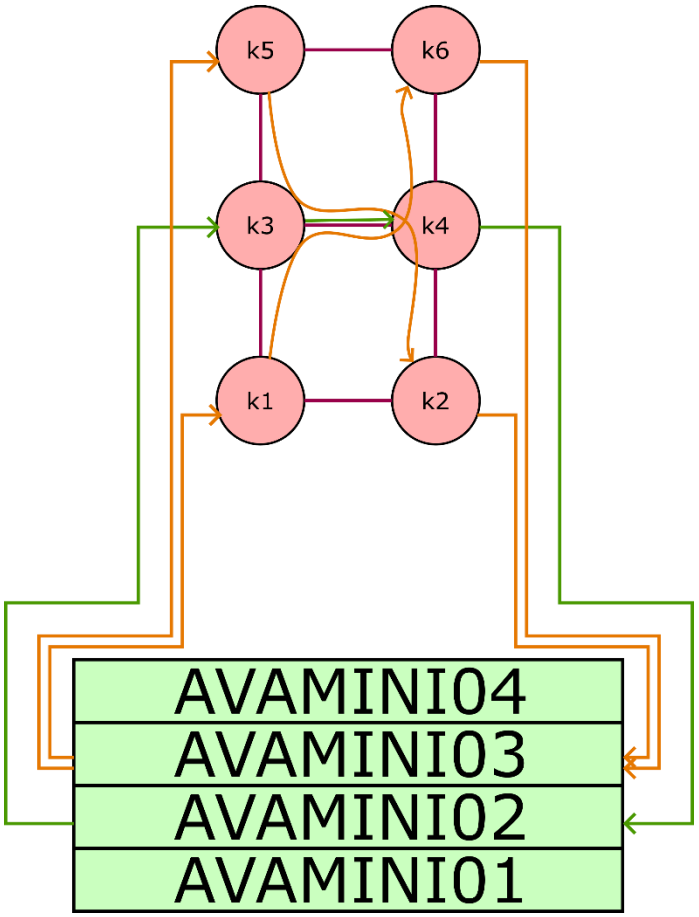


Software – Main Difficulties

- Sending TT-Traffic on pre-planned time
  - Earliest TxTime First support of Intel I210
- Accurate timestamps for measuring delays
  - From hardwareclock of NIC when sending or receiving frame
- Time Synchronization
  - AVAMINI01 as grandmaster
  - Synchronization:
    - To Kontron switches via control links
    - To servers via backbone
    - Device internal via phc2sys

Case Study

- TT-Stream
  - AVAMINI02 → k3 → k4 → AVAMINI02
  - 1500 Byte Payload
  - 500 µs cycletime
- BE-Traffic
  - 500 Mbit/s
    AVAMINI03 → k5 → k3 → k4
    → k2 → AVAMINI03
  - 500 Mbit/s
    AVAMINI03 → k1 → k3 → k4
    → k6 → AVAMINI03
- Different switch configurations
  - PCP
  - GCL
  - GCL + GB
  - Full Path



Case Study – PCP

- Different priority code point (PCP) values
  - TT-Stream → PCP = 7
  - BE-Traffic → PCP = 0

GCL k3 → k4			GCL k4 →AVAMINI02		
	Gates	Time		Gates	Time
T01	OOOOOOOO	500 000ns	T01	OOOOOOOO	500 000ns

Case Study – GCL

- Different priority code point (PCP) values
  - TT-Stream → PCP = 7
  - BE-Traffic → PCP = 0
- Timeslot for Current Part of Path where only Queue 7 is open

GCL k3 → k4			GCL k4 →AVAMINI02		
	Gates	Time		Gates	Time
T01	CCCCCOOO	27 160ns	T01	CCCCCOOO	54 320ns
T02	OCCCCCCC	12 160ns	T02	OCCCCCCC	12 160ns
T03	CCCCCOOO	460 680ns	T03	CCCCCOOO	433 520ns

Case Study – GCL + GB

- Different priority code point (PCP) values
  - TT-Stream → PCP = 7
  - BE-Traffic → PCP = 0
- Timeslot for Current Part of Path where only Queue 7 is open
- Guard-Bands are activated

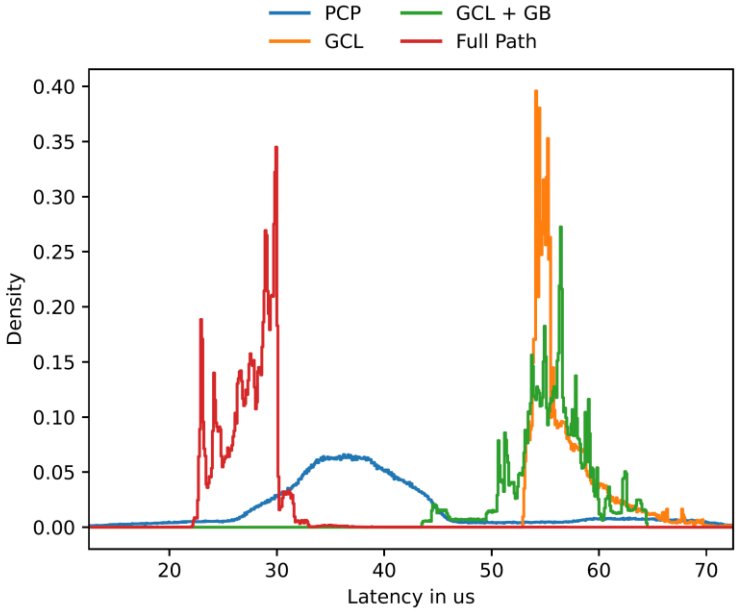
GCL k3 → k4			GCL k4 →AVAMINI02		
	Gates	Time		Gates	Time
T01	CCCCCOOO	27 160ns	T01	CCCCCOOO	41 984ns
T02	CCCCCCCC	12 336ns	T02	CCCCCCCC	12 336ns
T03	OCCCCCCC	12 160ns	T03	OCCCCCCC	12 160ns
T04	CCCCCOOO	460 680ns	T04	CCCCCOOO	433 520ns

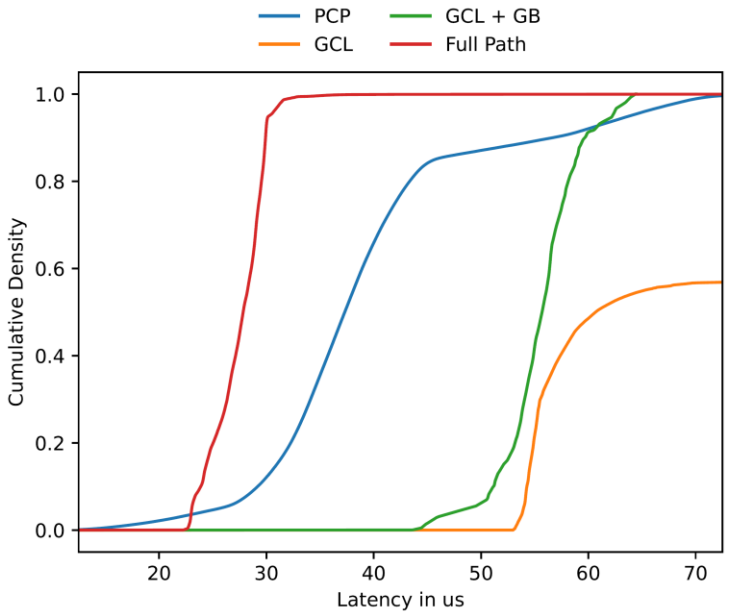
Case Study – Full Path

- Different priority code point (PCP) values
  - TT-Stream → PCP = 7
  - BE-Traffic → PCP = 0
- Timeslot for Full Path where only Queue 7 is open
- Guard-Bands are activated

GCL k3 → k4			GCL k4 →AVAMINI02		
	Gates	Time		Gates	Time
T01	OCCCCCCC	66 480ns	T01	OCCCCCCC	66 480ns
T02	CCCCCOOO	421 184ns	T02	CCCCCOOO	421 184ns
T03	CCCCCCCC	12 336ns	T03	CCCCCCCC	12 336ns

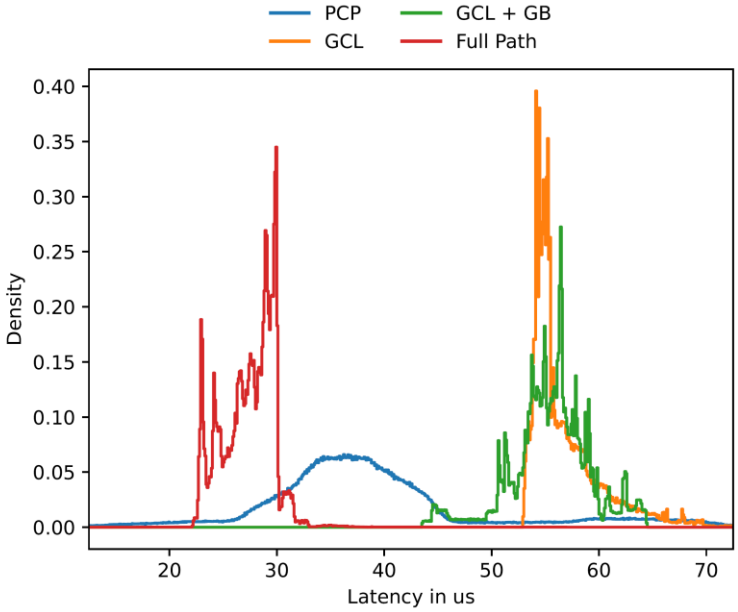
Results

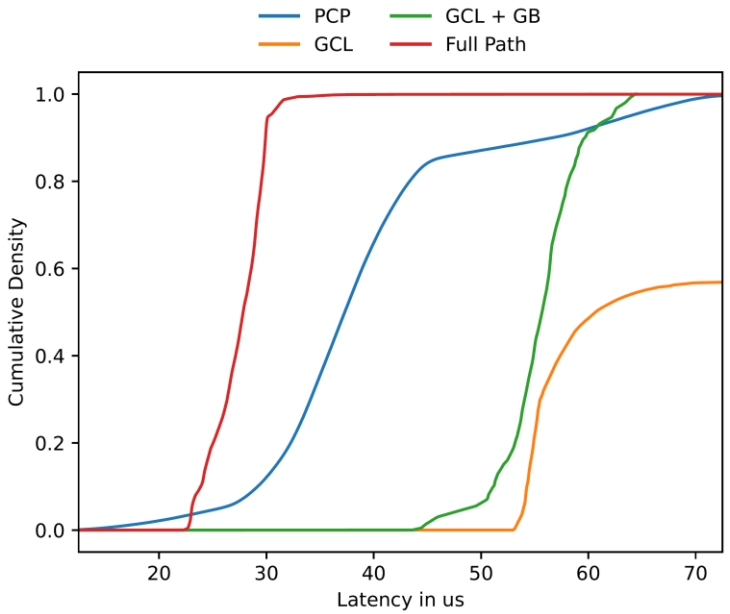




- Meets expectations
- Sending and receiving works under full network utilization

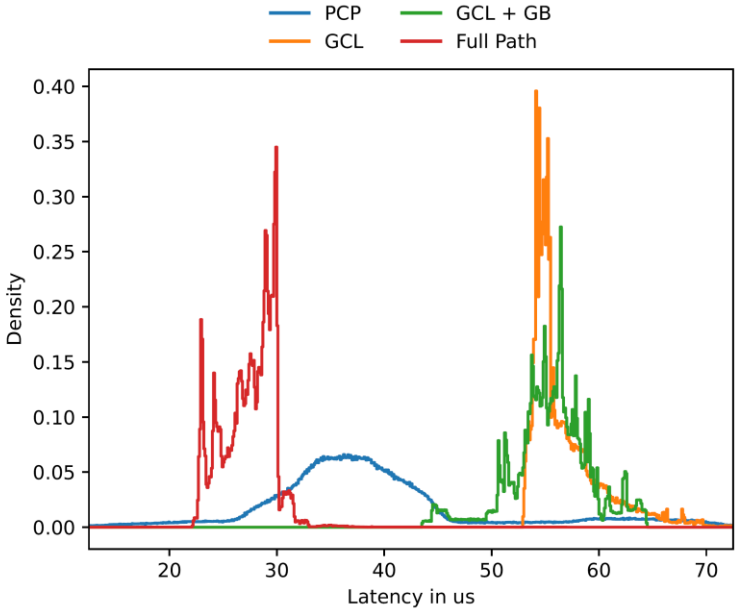
Results – PCP

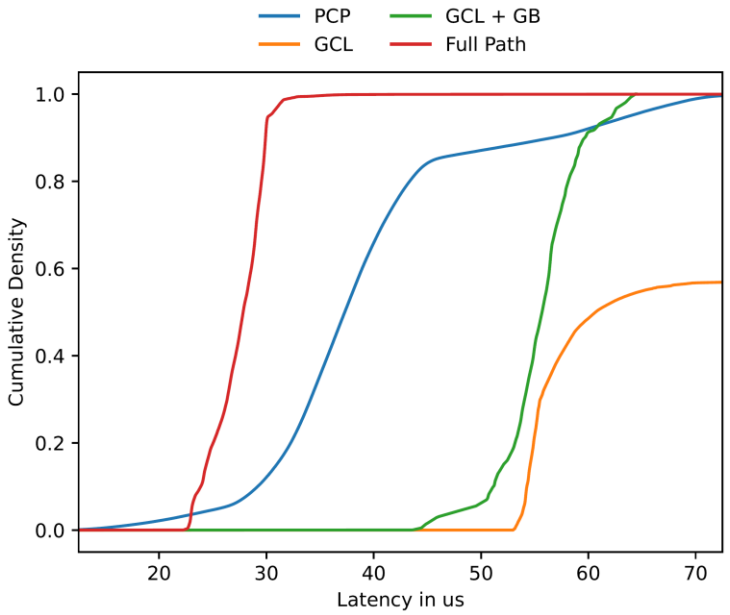




- Widely distributed with most delays between 30µs and 45µs
- High jitter

Results – GCL

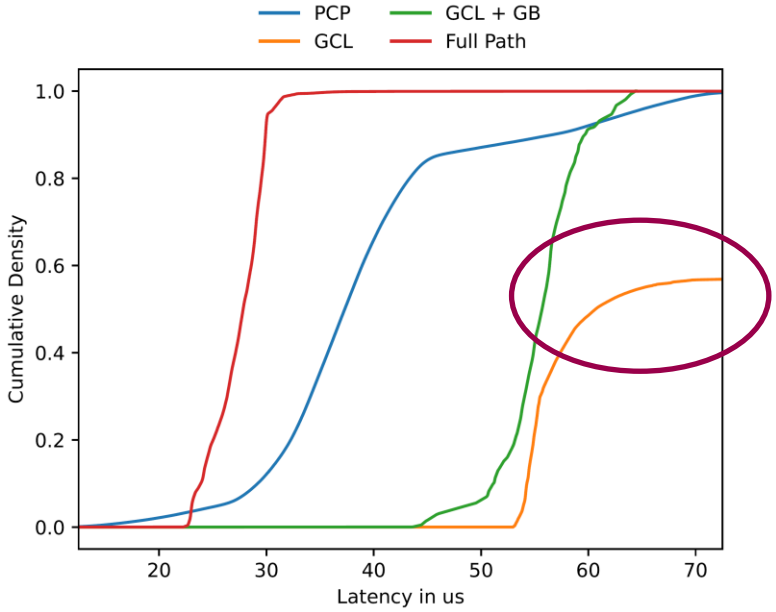


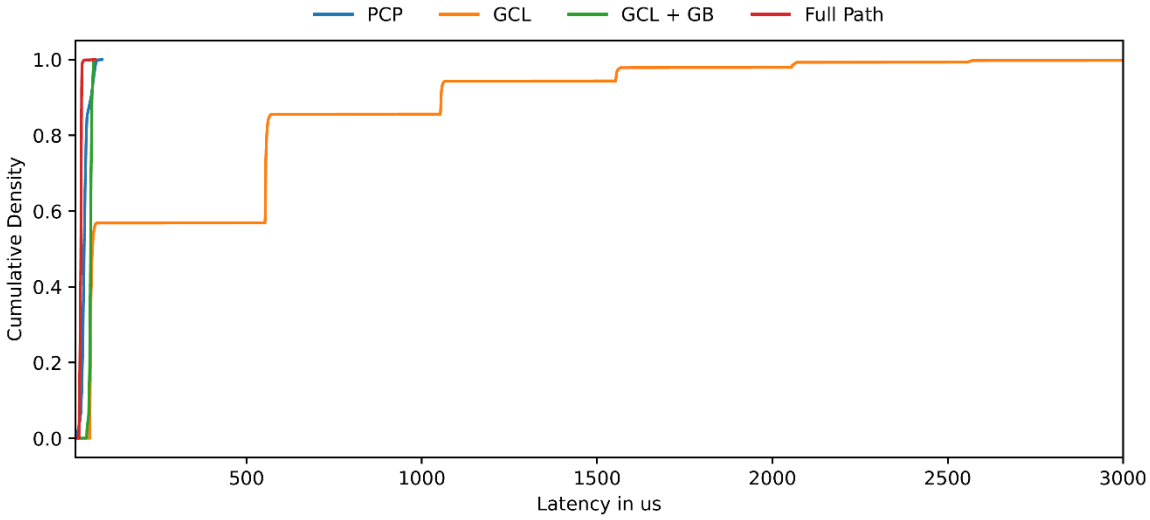


- Lowest latency at 54µs
- High peak
- Similar to GCL + GB

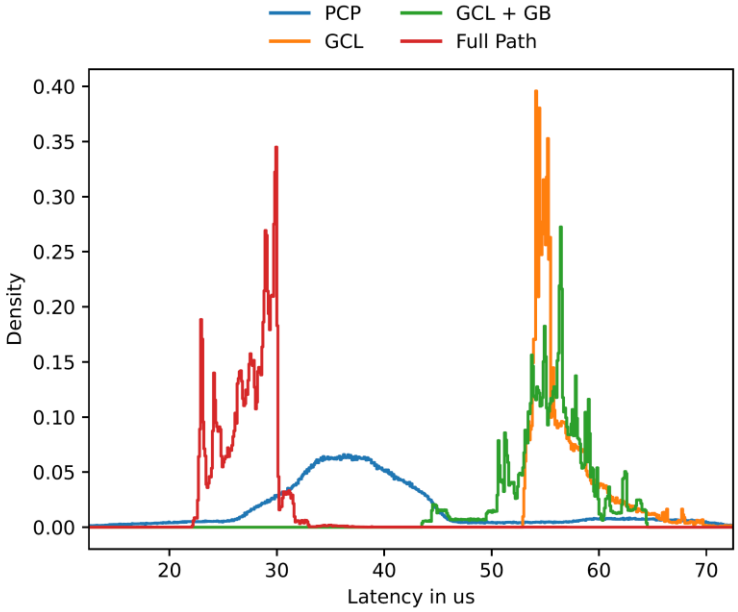
Results – GCL

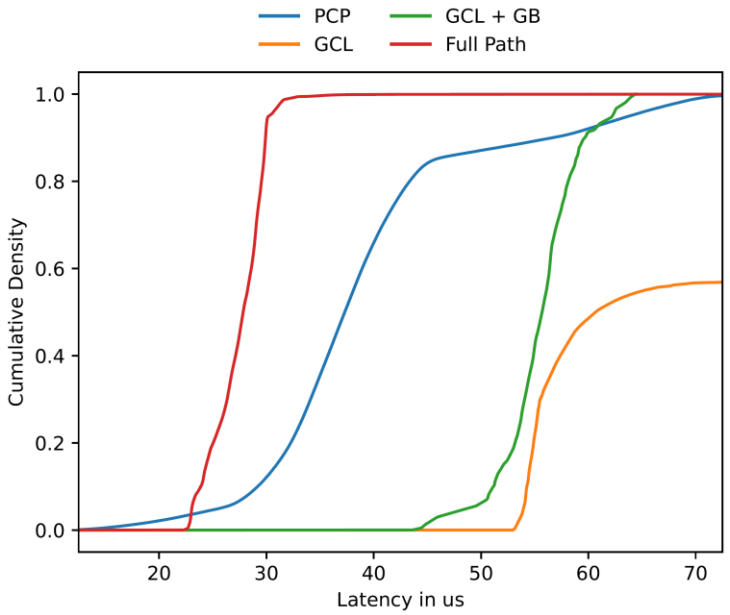
- According to IEEE 802.1Q
  - GCL should perform better then GCL + GB
  - Definition of GB not necessary
- Reality
  - KBox switches do not deliver protected windows by their own
  - Frames are not delivered within cycle time





Results – GCL + GB

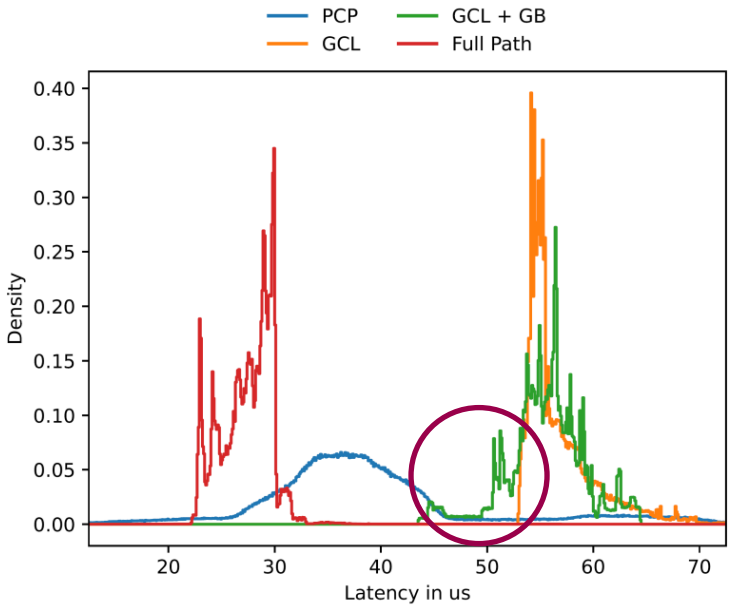




- Similar to GCL
- All frames are received within 70µs

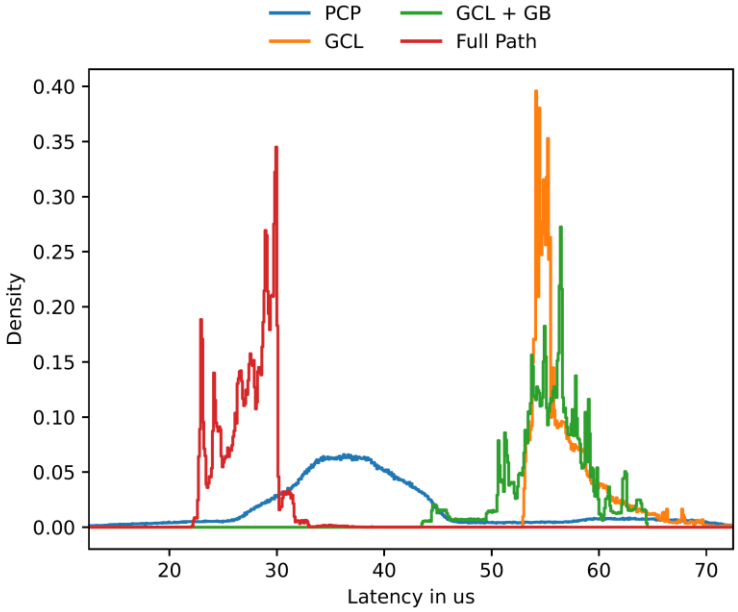
Results – GCL + GB

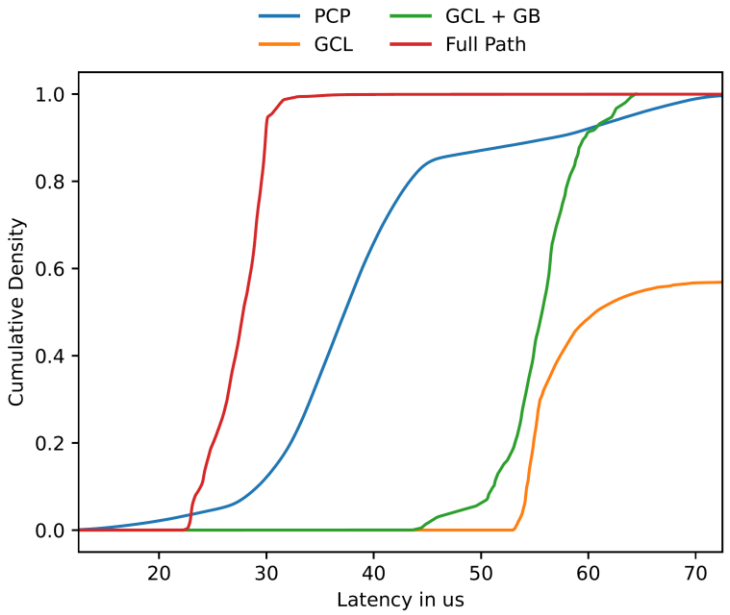
- last gate is opened at 54.2µs
- No frame should reach destination before
- Reason:
  - Poor time synchronization



GCL k3 → k4			GCL k4 →AVAMINI02		
	Gates	Time		Gates	Time
T01	CCCCCOOO	27 160ns	T01	CCCCCOOO	41 984ns
T02	CCCCCCCC	12 336ns	T02	CCCCCCCC	12 336ns
T03	OCCCCCCC	12 160ns	T03	OCCCCCCC	12 160ns
T04	CCCCCOOO	460 680ns	T04	CCCCCOOO	433 520ns

Results – Full Path





- Low latency
- Lowest jitter
- Almost all frames are received within 35µs

Conclusions

- Testbed
  - Hardware setup with 6 switches and 4 servers
  - Software for automatic network and endpoint configuration
  - Integrated scheduler for automatic traffic planning
- Advantages
  - Different topologies without physical changes
  - Multiple traffic classes at once via separated servers
- Leveraging practical knowledge at working with real TSN networks
  - Manage cyclic topologies
  - Time synchronization with gPTP
- Challenges
  - Not all hardware works according the standard
    - Testing different hardware necessary
  - Poor time synchronization with ptp4l and phc2sys
    - Synchronization via TSN network

Thank You for your attention!

Willi Brekenfelder
willi.brekenfelder@uni-rostock.de
https://www.ava.uni-rostock.de