Common Implementation and Interoperability Challenges: Focus on Clock Control for Qbv on Linux-based End Stations

Presentation at TSN/A conference, October 2nd, 2024

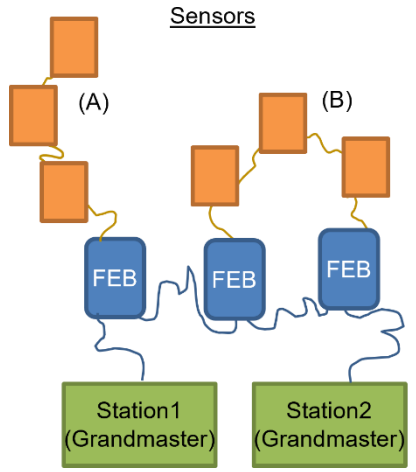
Kilian Brunner, Martin Ostertag (ZHAW)

Florian Frick (ISW)

Introduction

- **Interoperability and usability of TSN is in the focus of the IIC TSN Testbed at the ISW in Stuttgart**
  - Hands-on testing is complementing standardization and product development
  - Learnings based on experience of a large number of vendors and frequent plugfests

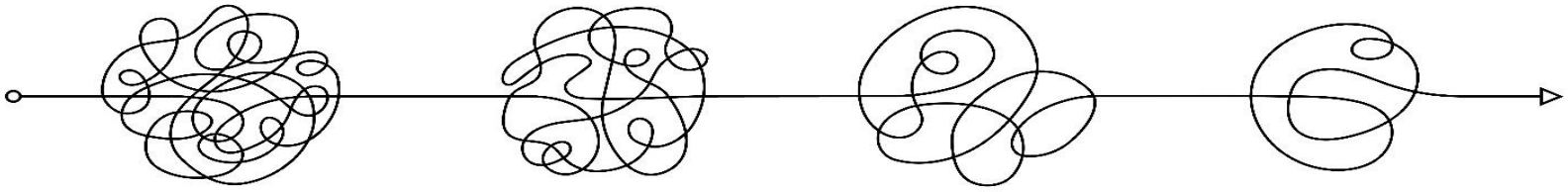
- **ZHAW developed a portable IEEE 802.1AS-2020 conformant software stack**
  - Used mainly for synchronized measurements, no Qbv and other TSN features
  - To improve the SW quality, we join the IIC plugfests in Stuttgart since March 2023



Observed Phenomena with “Real Devices”

- **Observations**
  - Startup and frequent unexpected grandmaster changes
  - The time synchronization tree would split from time to time, resulting in two active GMs in the network
  - Throughput of streams with scheduled traffic was far from expected

- **Analysis**
  - Joint analysis revealed different interpretation of error conditions and error handling, and misaligned settings:

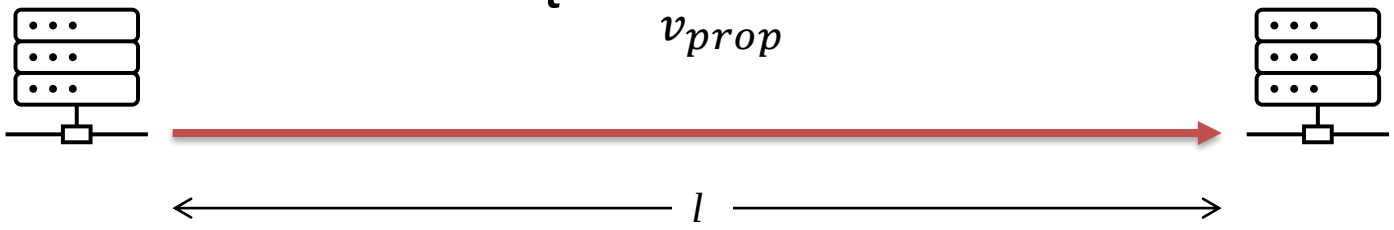


☹ Negative values for link delay measurements
☹ Traffic class allocation for time synchronization messages
☹ Control of LocalClock to support time triggered transmission and TAS (802.1Qbv) gating

Negative link delays

- **meanLinkDelay**
  - Is the physical propagation delay between two devices

$$t = \frac{l}{v_{prop}}$$



- **How would you handle the following situation in your code:**
  - Your measurement tells you, the propagation delay is negative
  - The standard tells you: *the data type is “unsigned nanoseconds”*

Negative link delays (2)

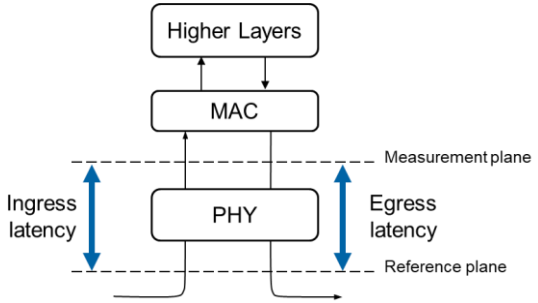
- **Observed reactions to negative delays include:**
  - **Set value to 0, stop operation**, **restart** the stack, **disable the port** for a certain time
  - The first will reduce accuracy and the others of these actions will re-trigger synchronization tree establishment (BTCA)

- **Two reasons why meanLinkDelay may be negative**

At startup (NRR unknown)

$$d = \frac{(t_4 - t_1) * NRR - (t_3 - t_2)}{2}$$

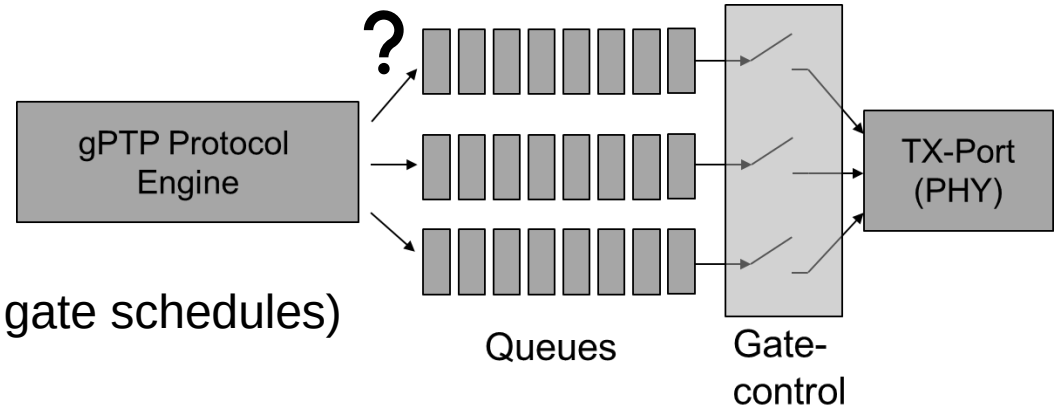
Incorrect PHY latency compensation



- **Mitigation: Prepare for the unexpected**
  - IEC/IEEE 60802 identified the issue and explicitly addresses it in a note
  - Maintenance items to AS-2020 have been submitted at IEEE

Traffic class assignment for IEEE 802.1AS

- **IEEE 802.1AS-2020 does not specify a traffic class for messages**
  - “shall be transmitted in expedited manner compared to other traffic (e.g., best effort)”



- **Observations under high load**
  - Synchronization disrupted (overload or misaligned gate schedules)
- **IEC/IEEE 60802 d3.0 describes (non-normative) a bit clearer**
  - “Synchronization is part of network control”
    In the example config shown there, this has the highest PCP value
- **Mitigations**
  - Have traffic class assignment for 802.1AS messages configurable in your device.
  - Network management and engineering need to consider this in the planning and schedule definitions, the work on engineering processes and tools is absolutely crucial for TSN success.

Clocks in devices and in IEEE 802.1AS

- **IEEE 802.1AS operates on the assumption that message time stamps are generated by a single free running “LocalClock” for all ports**
  - Correction of network delays and clock frequency rely on this assumption
- **Many devices today have a single HW clock for**
  - PTP time stamping
  - HW assisted time-triggered transmission
  - Qbv gate control
- **Controlling frame transmission times requires this clock to be synchronized to the network cycle**
  - The time received via IEEE Std 802.1AS

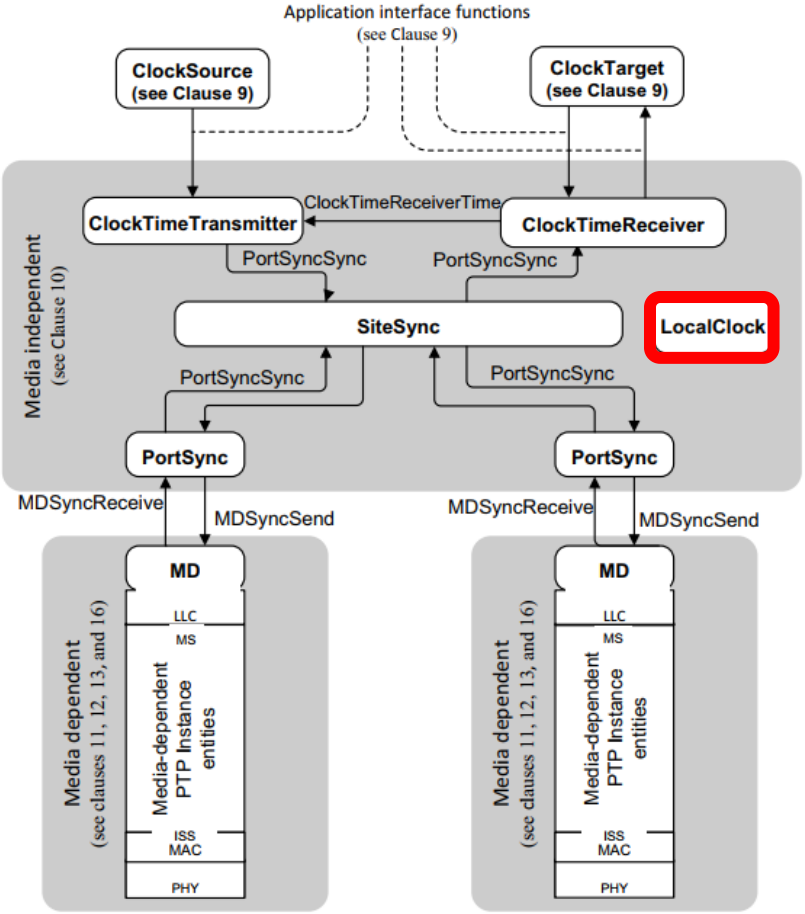
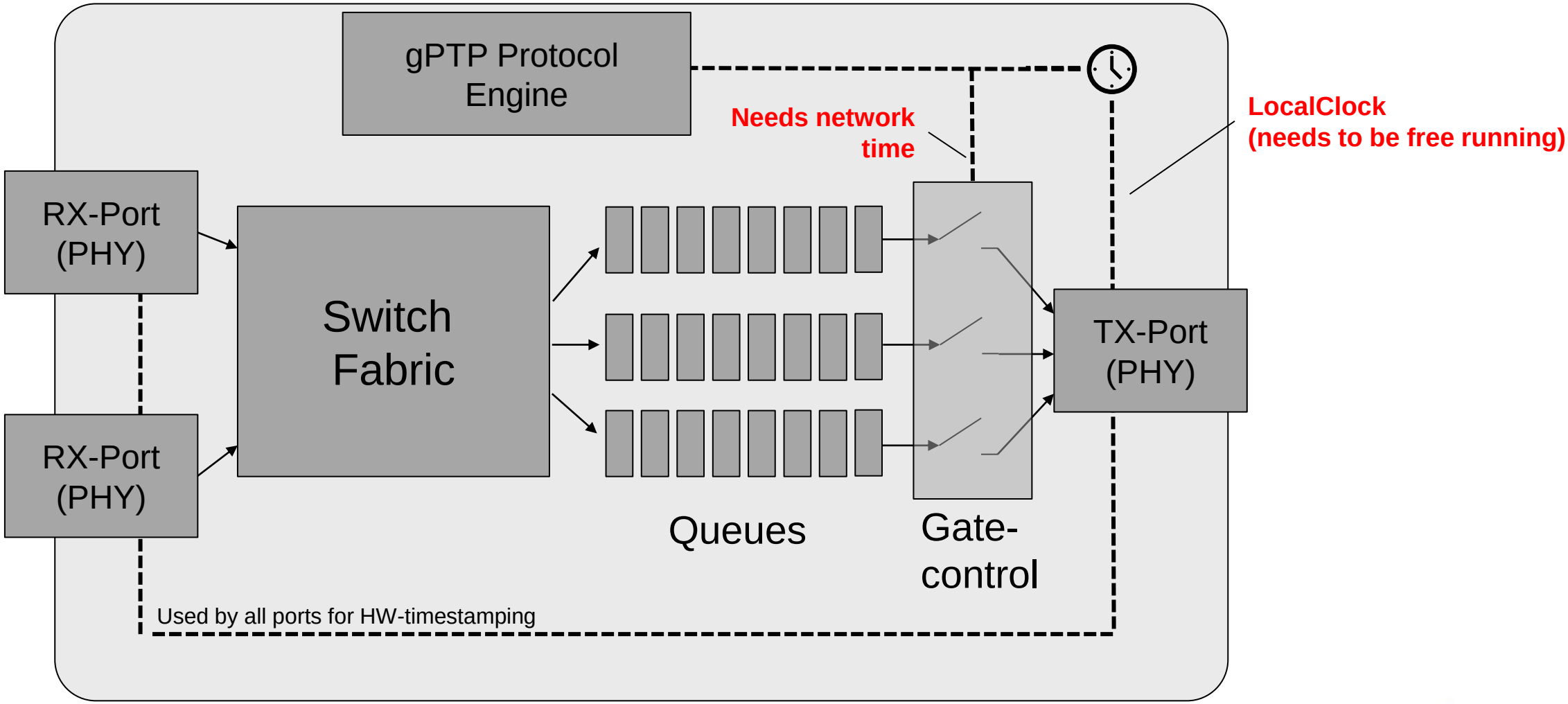


Figure 10-1—Model for media-independent layer of PTP Instance

Source: IEEE 802.1ASdr™-2024

Single Clock Dilemma



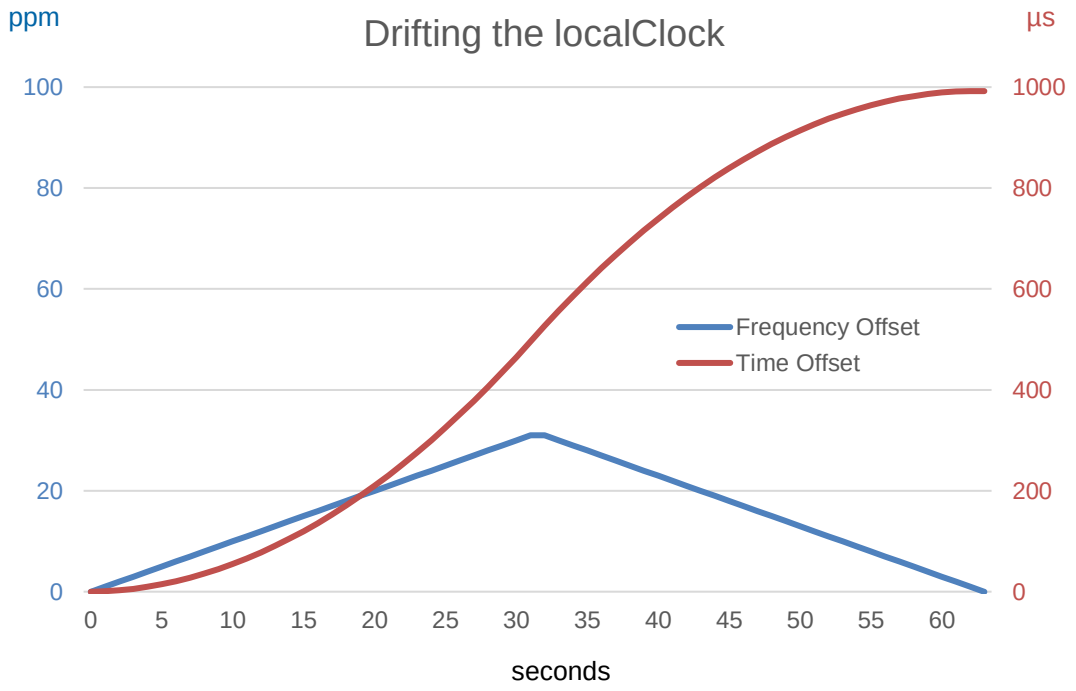
Why drifting may not be sufficient

- **It is “OK” for the LocalClock to drift within limits**
  - “… The physical adjustment of the frequency … is allowed but not required” (802.1AS-2020)
  - Maximum frequency change: 1 ppm of nominal frequency per second (IEC/IEEE 60802 d3.0)

- **Drifting 1 millisecond takes ~1 minute, which is hardly acceptable**

- **Options?**
  - Jump the local clock
  - Syntonize the local clock and handle the time offset in the driver or on application level

- **Often, approach 1 is chosen**



And what IF we jump?

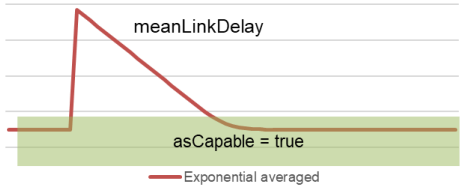
Universität Stuttgart ISW

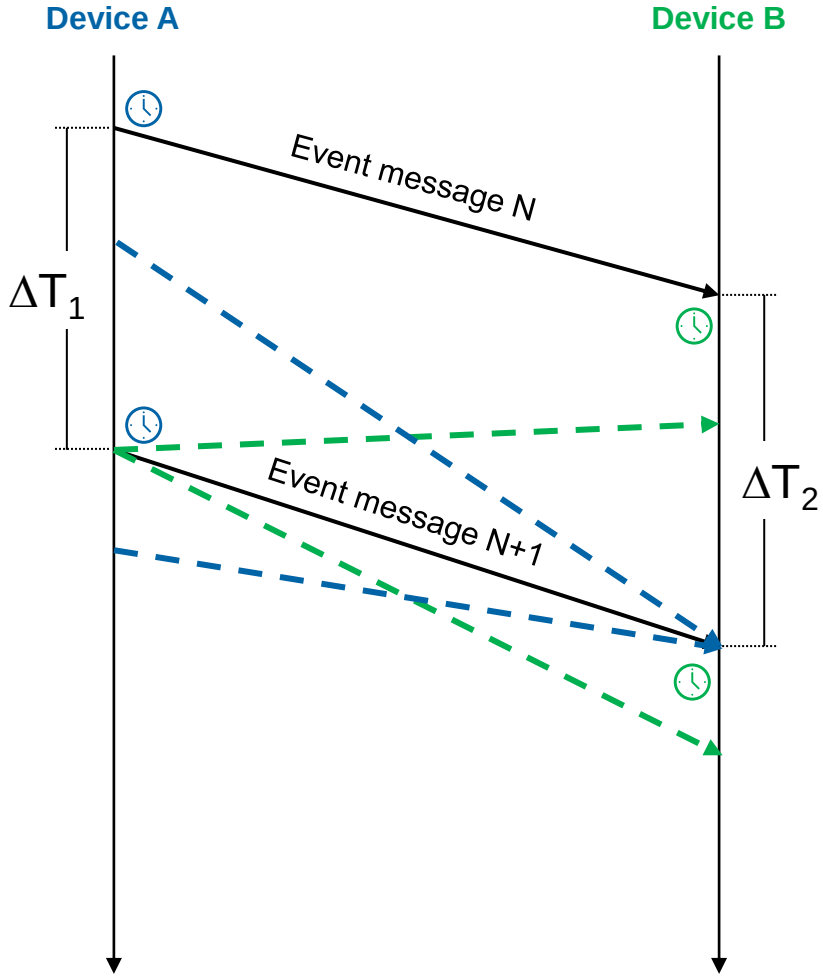
zhaw School of Engineering InES Institute of Embedded Systems

$$\text{NeighborRateRatio } NRR = \frac{\Delta T_1}{\Delta T_2}$$

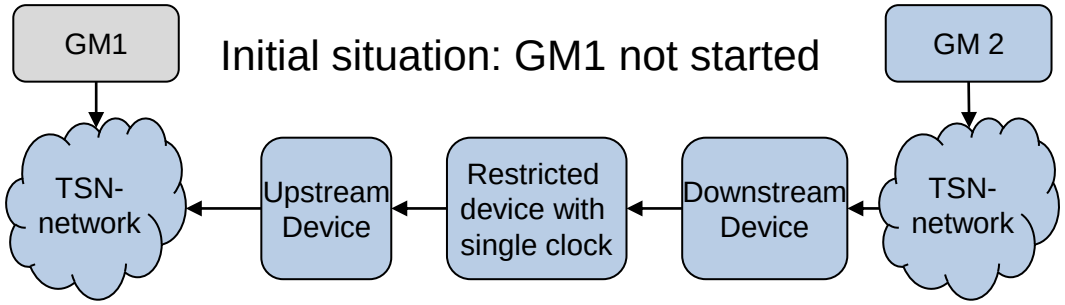
- **If either LocalClock jumps, calculations are messed up**
  - Blue: Device A jumps
  - Green: Device B jumps

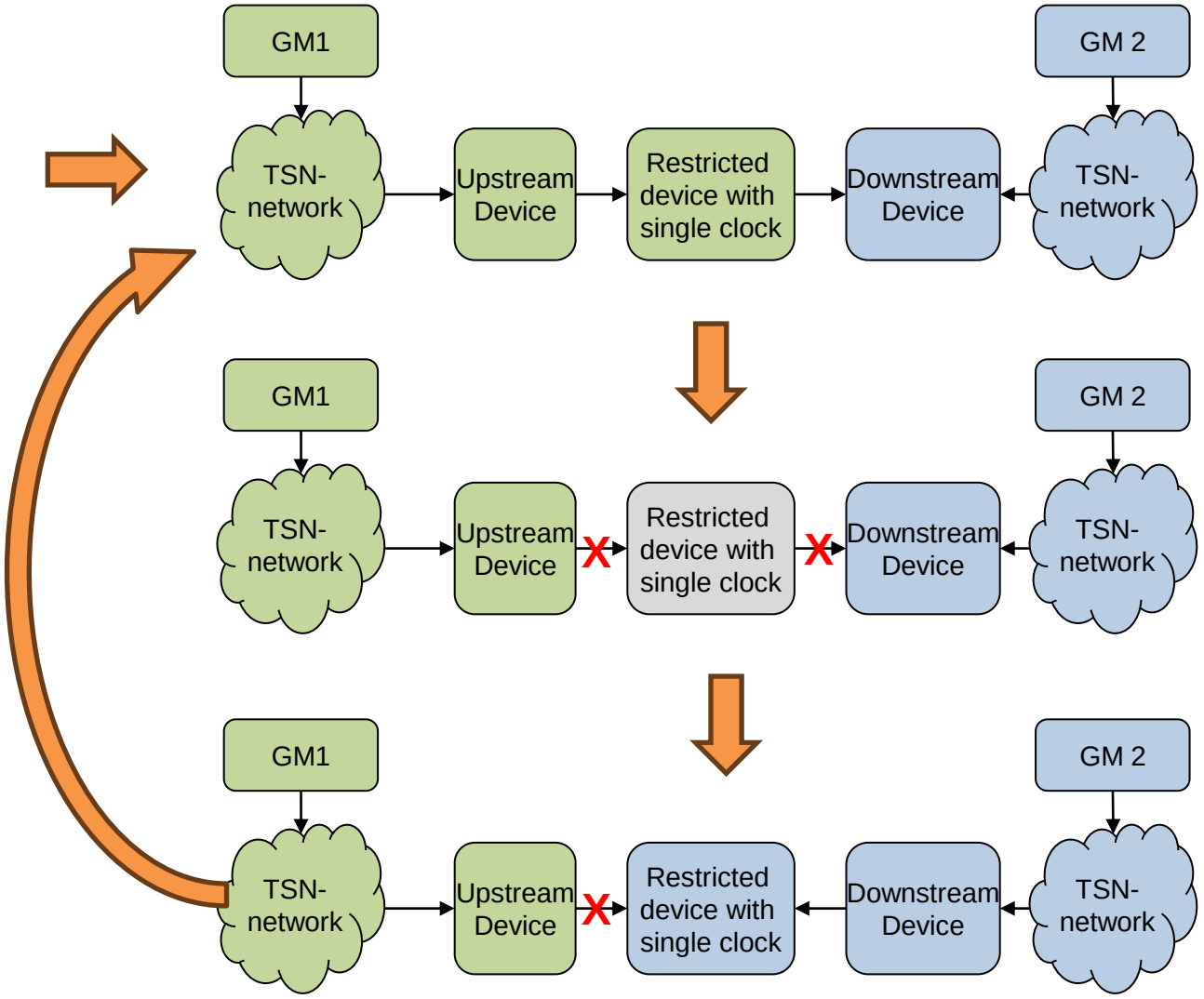
- **If the jumps are large, NRR will be far off and impact meanLinkDelay calculation**
  - NRR (in ppm) may take very large negative or positive values
  - Range checks will flag the link as non asCapable
    - Link will be removed from the sync tree
  - AS-2020 proposes (not mandatory) an IIR filter for link delay averaging
    - There is a memory effect





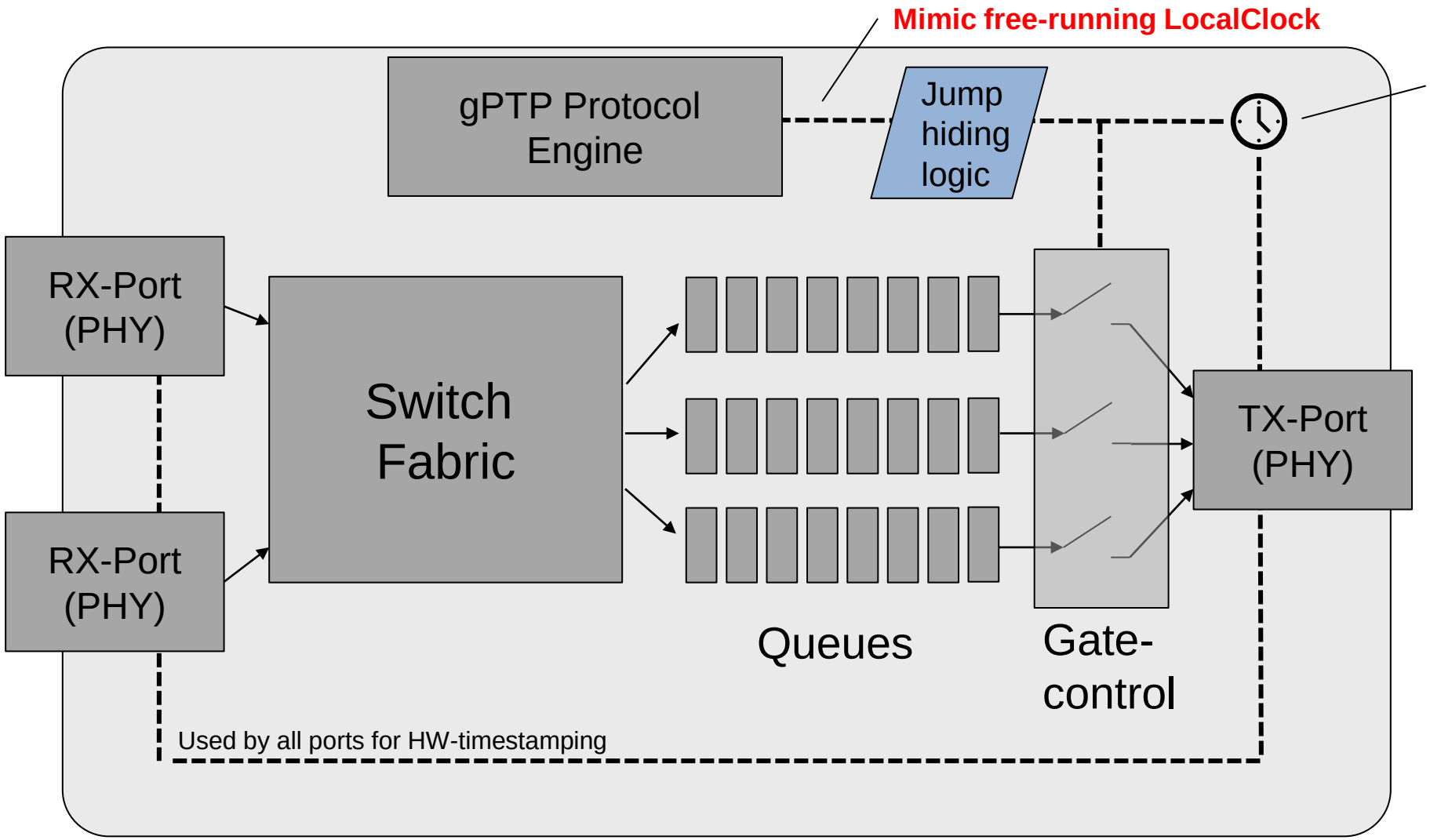
Mixed vendor system observations





- **Inhomogeneous device landscape**
- **Some running LinuxPTP**
- **Some running other implementations**

Proposed Mitigation



- **Correct timestamp values by subtracting the (accumulated) jumps before forwarding them to the gPTP protocol engine**
- **Add heuristics (median filters or similar) to cope with misbehaving neighbors**

LocalClock: Mid- and Long-term solution

- **Device vendors and implementors need to make sure that any timestamps visible outside the device follow the “free running clock” paradigm**
  - Some of the implementations used LinuxPTP:
    - When Sync messages are received, the clock jumps
    - Thereafter, the clock frequency jumps

- **With the proliferation of TSN technology**
  - New components are being developed by component vendors that resolve the described dependency and provide independent clocks for time stamping and egress timing

- **Hardening the implementation to cope with “misbehaving” neighbors is always a good idea**

Conclusion

- **Interoperability between devices of different vendors, using different components has its challenges**
  - Interoperability tests beyond protocol conformance testing can increase product quality, identify challenges and feed back findings to IEEE
  - Not only IIC (part of Digital Twin Consortium), also AVNU conducts interoperability plugfests
  - Consider extending existing test suites beyond protocol conformance
  - Publication with IEEE Open Access is ongoing

- **Component and device vendors are already addressing these challenges in new and upcoming components**

- **IEC/IEEE 60802 requires rock-solid time synchronization as a base**

Thank you

Universität Stuttgart ISW

zhaw School of Engineering
InES Institute of Embedded Systems

https://www.zhaw.ch/de/engineering/institute-zentren/ines/communication-network-engineering

https://www.isw.uni-stuttgart.de/en/research/

kilian.brunner@zhaw.ch
martin.ostertag@zhaw.ch
florian.frick@isw.uni-stuttgart.de