Evaluating Requirements of High Precision Time Synchronisation Protocols using Simulation

Lazar T. Todorov **Till Steinbach**
Franz Korf Thomas C. Schmidt

Hamburg University of Applied Sciences
{lazar.todorov, till.steinbach, korf, schmidt}@informatik.haw-hamburg.de

6th International Workshop on OMNeT++ (OMNeT++ 2013)
5 March 2013, Cannes, France

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

SPONSORED BY THE

Federal Ministry of Education and Research

Agenda

1. Problem Statement & Motivation
2. Background & Related Work
3. Concept, Evaluation & Use-Cases
4. Conclusion & Outlook

Problem Statement

Why analysing clock synchronisation protocols

- High precision time synchronisation is required in many domains of distributed real-time systems
  - Cars, trains, airplanes, industrial installations, ...
- Design and configuration of synchronisation protocols is very challenging
- Many parameters per node
  - Clock precision, maximum drift, buffer-sizes, timeouts, tasks, ...
- Complex equations to analytically evaluate the setup
- After deployment, synchronised system is a black-box

Motivation
Why simulation is the right tool

Simulation helps to obtain or analyse:

- Expected system behaviour of a given hardware configuration
- Required hardware or oscillator precision for given timing requirements
- Startup-time until system is operational
- Environmental influences on synchronisation
  - Voltage, temperature, ageing
- Failover scenarios

Challenges
For a precise simulation model

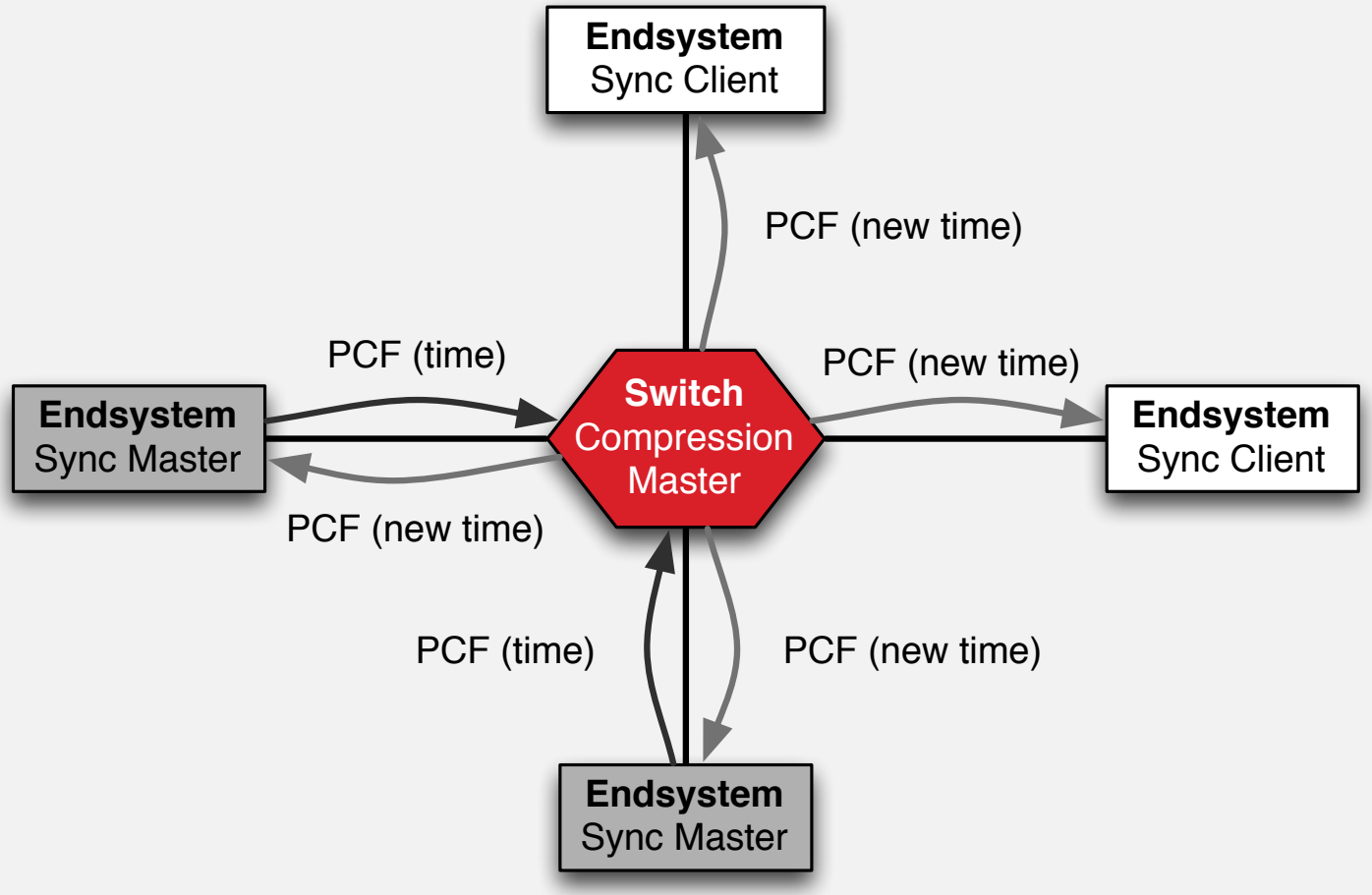
- Correct results require high accuracy
  - Clock synchronisation precision can be in the range of nanoseconds
  - Clock (or frequency) drift is in range of fractions of a picosecond
- Suitable simulation model of oscillator behaviour
- Simulation of synchronisation is extremely resource heavy
  - Simulation of dedicated time for each component
  - Simulation of oscillator for each component
  - Rescheduling of future events for every change regarding clocks
- Run simulation very fast (real-time)

- Approach focuses on Ethernet-based synchronisation
- Most parts can still be used in other systems
- There are several protocols
  - IEEE 1588 PTP
  - IEEE 802.1AS
  - AS6802 Time-triggered Ethernet
  - ...

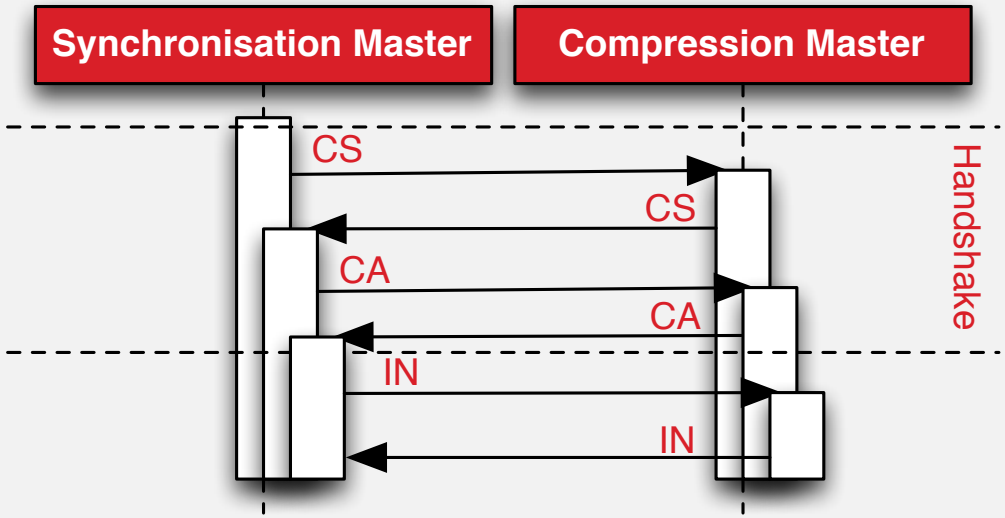
- AS6802 Specification
  - Society of Automotive Engineers
  - Standardised in 2011[1]
- Compatible extension to IEEE 802.3
- Master-/slave protocol
- Fault-tolerant

[1] Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee: *Time-Triggered Ethernet AS6802*. Nov. 2011.

- Synchronisation uses Protocol Control Frames (PCF)
- AS6802 defines 3 roles:

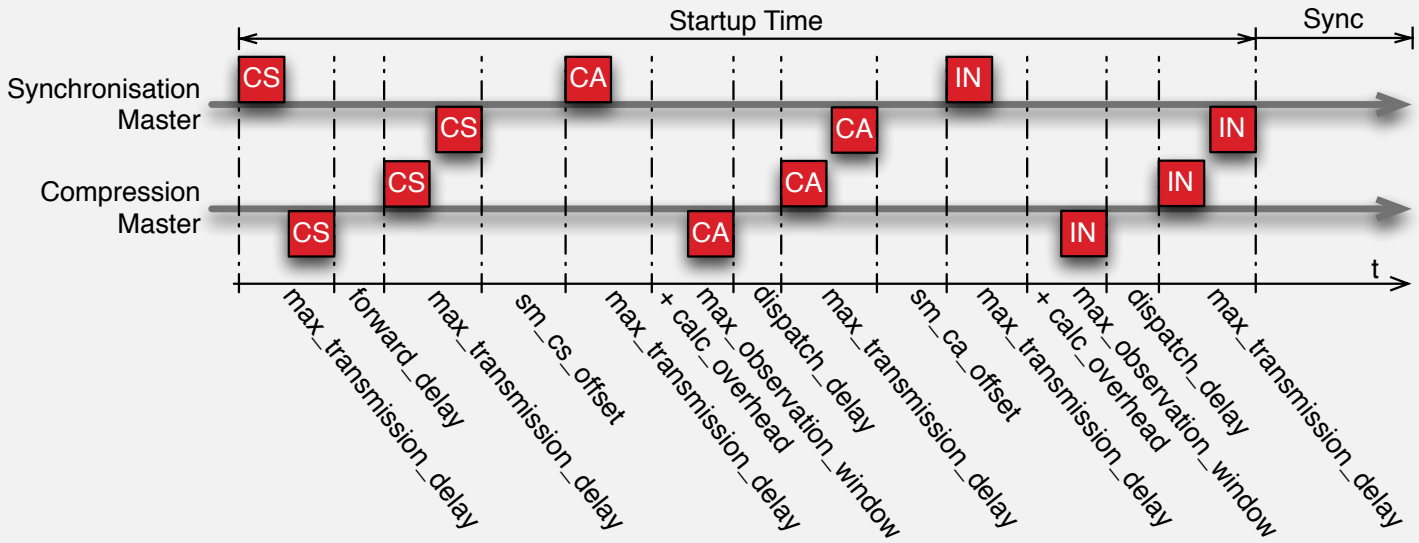


- Startup is controlled by an initial handshake



- Defines **startup time** until global time is established

- It is possible to estimate the startup time:



- Only a rough estimation. Scheme does not respect:
  - Actual clock drift
  - Rounding due to clock resolution

Related Work & Previous Work

Related Time Synchronisation Models in OMNeT++

- Praus/Granzer/Gaderer/Sauter[2]
  - Concept for synchronisation in industrial automation
  - Separates hardware and software components
- Liu/Yang[3]
  - IEEE 1588 PTP simulation in OMNeT++
  - Shows influences of topology / operating conditions
  - Less complex oscillator model
- TTE4INET[4]
  - Provides real-time protocol infrastructure for the AS6802 synchronisation model

[2] Fritz Praus et al.:"A simulation framework for fault-tolerant clock synchronization in industrial automation networks". Sept. 2007.

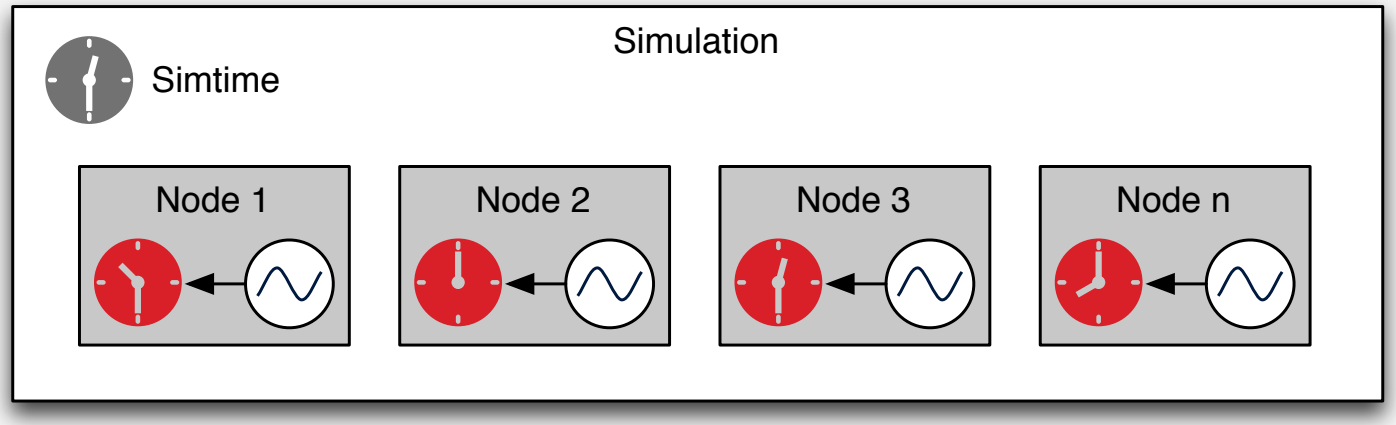
[3] Yingshu Liu and Cheng Yang:"OMNeT++ based modeling and simulation of the IEEE 1588 PTP clock". Sept. 2011.

[4] Till Steinbach et al.:"An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy". Mar. 2011.

Local Clocks

An independent simulation time for each node

- Each simulated node requires local clock
- Local clock must run independent from clocks of other devices
- Simulation time cannot be used (as it is simulation global!)
- Behaviour of local clock depends on oscillator model



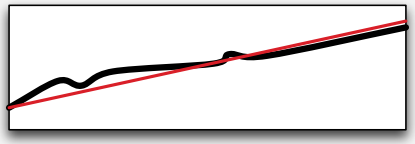
Oscillator Model
Modelling oscillator behaviour

- An oscillator is never precise
- Clock (frequency) drift affects the length of each tick[5]
- Oscillator model must regard inaccuracy for precise results
- Best oscillator model depends on use-case
- Make oscillators interchangeable and let user choose

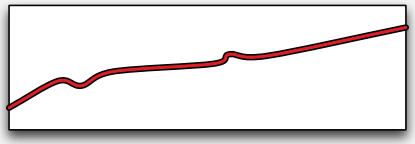
[5] International Telecommunication Union - Telecommunication Standardization Sector: *G.810 Definitions and Terminology for Synchronization Networks*. Aug. 1996.

Oscillator Model
Some Examples

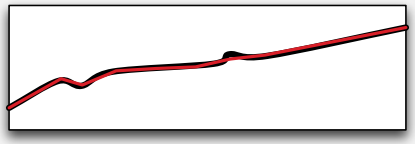
- Imprecise model:
  - Assumes constant clock drift
  - Clock drift is simulated only once per device



- Very precise model:
  - Each tick as separate Event
  - Clock drift for each tick is simulated

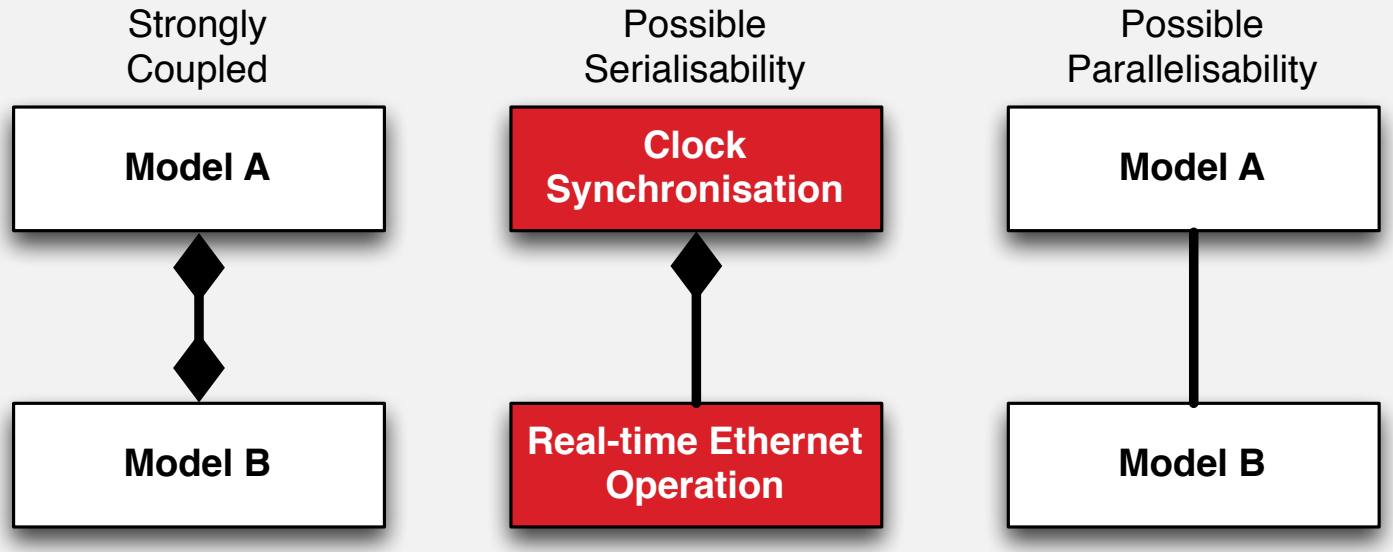


- Best trade-off:
  - Determines the time for that the drift can be assumed constant
  - Clock drift is typically simulated once per cycle



Increasing Performance

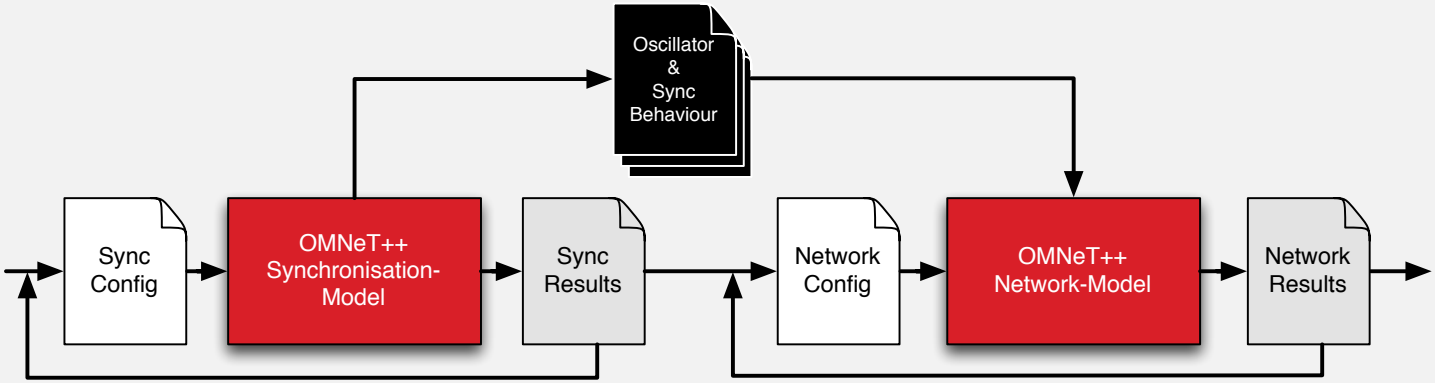
An evolutionary approach for simulation problems



- We call this *evolutionary approach*
- It uses knowledge of previous runs to accelerate following simulation

Increasing Performance

An evolutionary approach for simulation problems



1. simulate synchronisation
2. simulate operation

Simulation of large Networks

Performance gain when dividing the Model

- Performance increase when using evolutionary approach

network	nodes	Simsec/sec	accel. factor
with sync.	5	≈0.67	-
without sync.	5	≈8.50	≈12.68
with sync.	9	≈0.22	-
without sync.	9	≈4.40	≈20.00
with sync.	23	≈0.04	-
without sync.	23	≈1.63	≈40.75

- *All simulations run in real-time on COTS-Hardware!*

Hardware Configuration Analysis

Use-Case

- Used to determine network and synchronisation metrics of a given . . .
  - hardware profile
  - network configuration and topology
- Provides worst case imprecision and start-up duration

Hardware Configuration Analysis

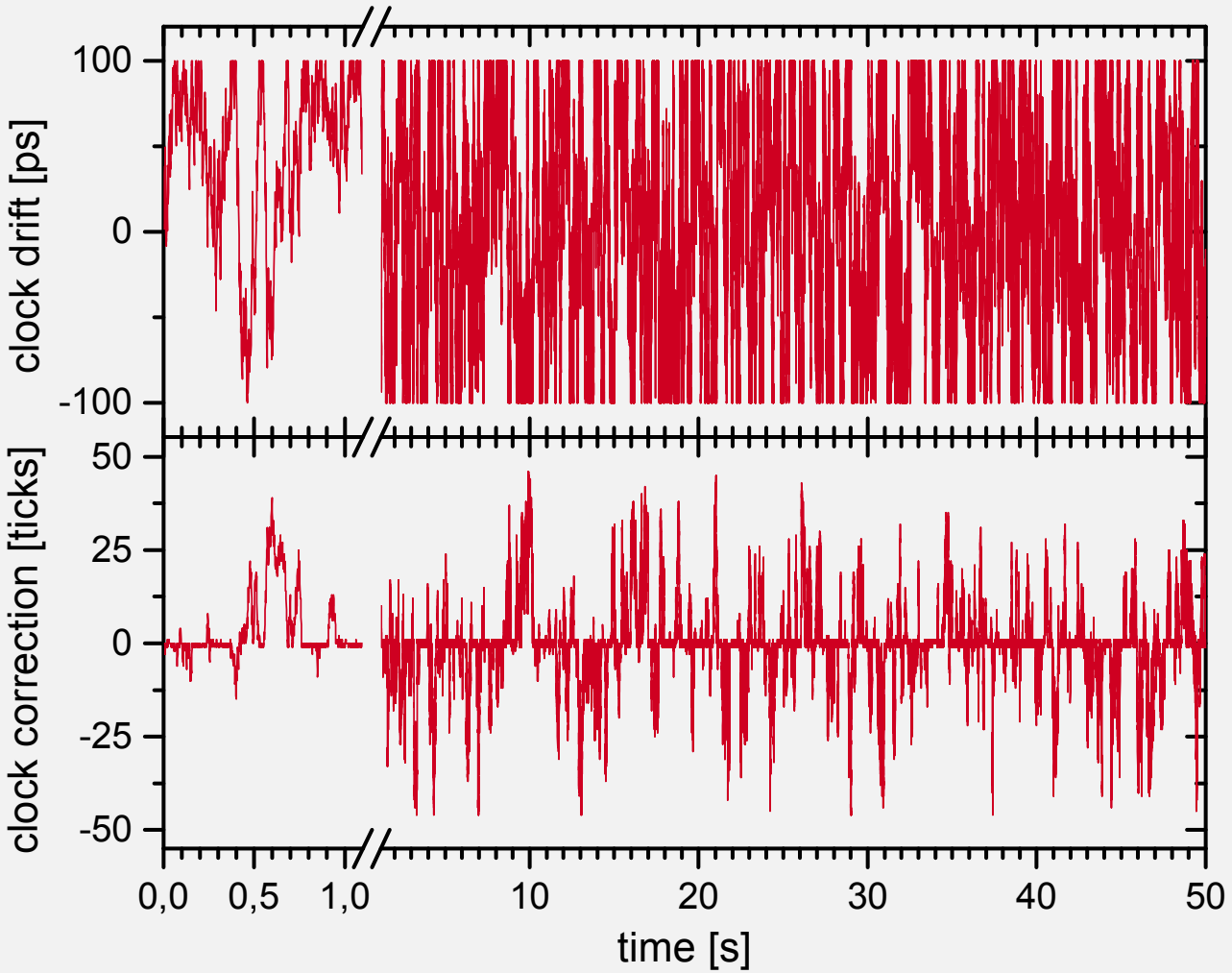
An example configuration

sync config	max drift	drift change	time to sync [µs]
Sync. Master	±100ps	20ps	209.84
Sync. Master	±300ps	34ps	209.92
Sync. Master	±200ps	20ps	209.92
Sync. Client	±150ps	3ps	209.92
Comp. Master	±70ps	14ps	202.88

- Estimated start-up duration for the configuration: 209.44 µs
  - max. 480 ns (6 ticks) difference between analytical and simulation result

Hardware Configuration Analysis

Clock correction made visible



- maximum clock correction below 50 ticks

Timing Requirement Analysis
Use-Case

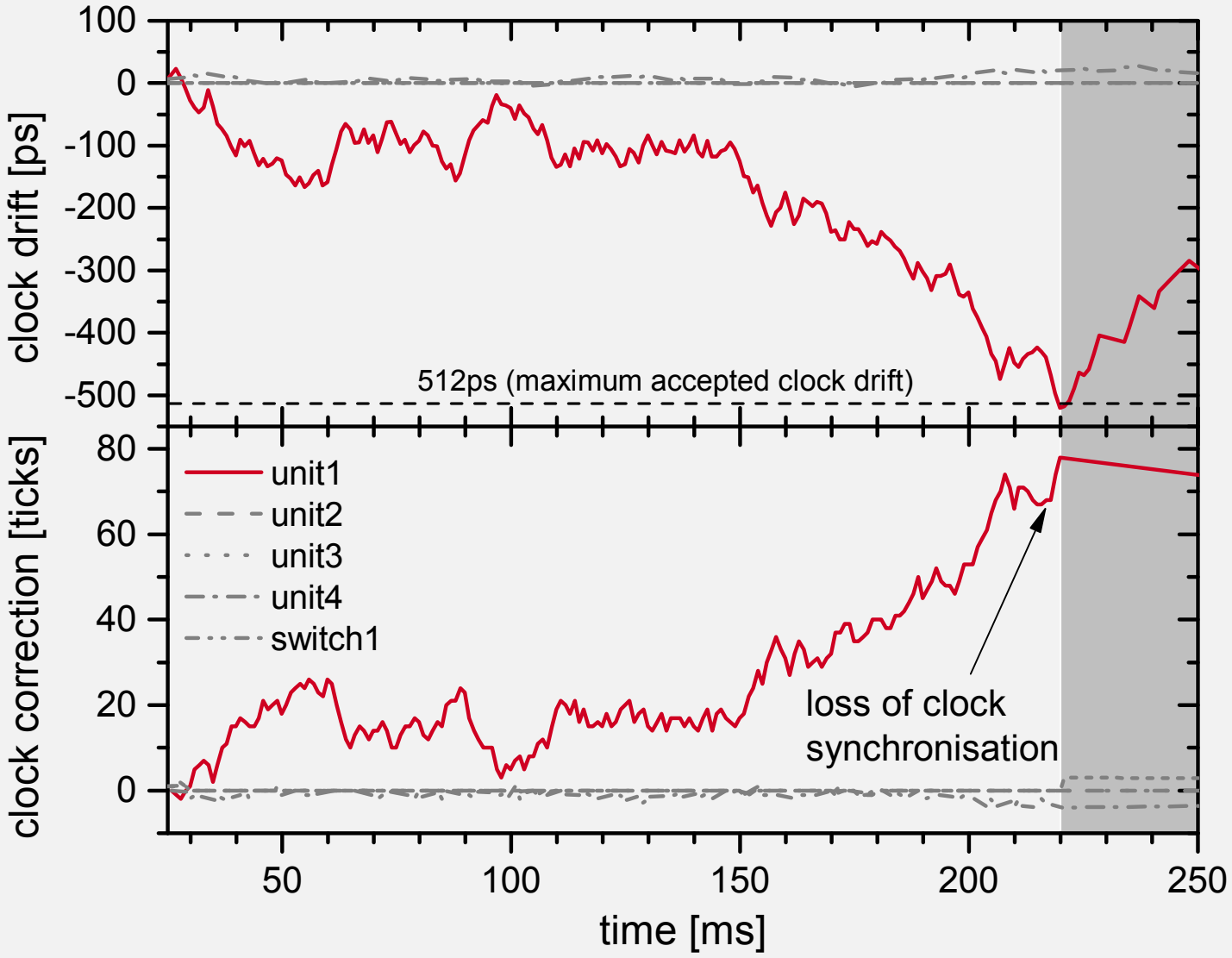
- Used to determine suitability of a component to fulfil given requirements

Example:

- Configuration similar to previous example, but one node with imprecise oscillator
- Shows the simulation of loss of clock synchronisation

Timing Requirement Analysis

Loss of clock synchronisation



Conclusion

- Real-time clock synchronisation gains importance
- For design and configuration, simulation is eligible to obtain parameters
- Significant increase of simulation performance by separation of clock synchronisation and network operation model
- For large networks simulation speed increased by over 40-times without affecting the results

Outlook
Ongoing and future work

- Compare simulation results with AS6802 hardware
- Design more detailed clock models
  - Artificially synchronised oscillators for worst-case analyses
  - Oscillator with temperature and voltage model
  - Ageing of oscillator
- Further improve performance by dividing oscillator and clock model

Evaluating Time Synchronisation Protocols using Simulation

Thank you for your attention!

- Website of simulation model: http://tte4inet.realmv6.org
- Website of CoRE research group: http://www.haw-hamburg.de/core

SPONSORED BY THE

Federal Ministry of Education and Research

References I

[1] Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee. *Time-Triggered Ethernet AS6802*. SAE Aerospace. Nov. 2011. URL: http://standards.sae.org/as6802/.

[2] Fritz Praus et al. "A simulation framework for fault-tolerant clock synchronization in industrial automation networks". In: *IEEE Conference on Emerging Technologies and Factory Automation 2007*. Sept. 2007, pp. 1465–1472. DOI: 10.1109/EFTA.2007.4416962.

[3] Yingshu Liu and Cheng Yang. "OMNeT++ based modeling and simulation of the IEEE 1588 PTP clock". In: *International Conference on Electrical and Control Engineering (ICECE), 2011*. Piscataway, NJ, USA: IEEE Press, Sept. 2011, pp. 4602–4605.

References II

[4] Till Steinbach et al. "An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy". In: *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. Barcelona, Spain: ACM-DL, Mar. 2011, pp. 375–382.

[5] International Telecommunication Union - Telecommunication Standardization Sector. *G.810 Definitions and Terminology for Synchronization Networks*. Tech. rep. Series G: Transmission Systems and Media. Geneva, Switzerland: Institution, Aug. 1996.

[6] D. B. Sullivan et al., eds. *Characterization of Clocks and Oscillators*. technical note 1337. Boulder, Colorado: National Institute of Standards and Technology, 1990.