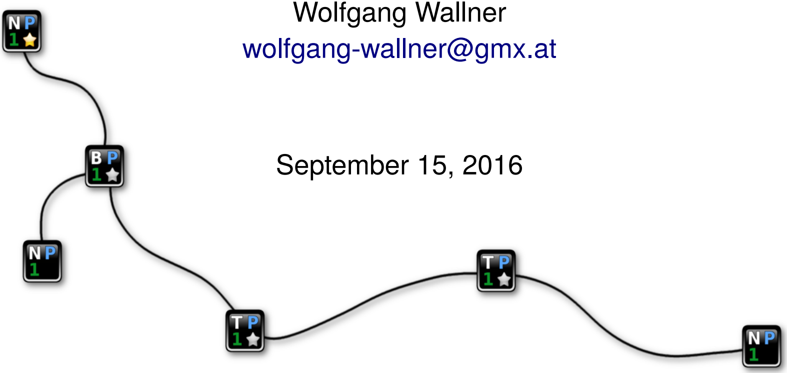
FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Simulation of the IEEE 1588 Precision Time Protocol in OMNeT++

Wolfgang Wallner
wolfgang-wallner@gmx.at

September 15, 2016



Presentation Outline

Introduction
- Motivation
- Problem statement

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Presentation Outline

- Introduction
  - Motivation
  - Problem statement
- Basic concepts
  - Clock Model
  - Precision Time Protocol
- Implementation
  - Clock Model for OMNeT++
  - PTP in OMNeT++

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

Motivation

Motivation:

- Distributed real-time systems need a **global time base**

Motivation

Motivation:

- Distributed real-time systems need a **global time base**
- Requirements depend on the application
  - Precision
  - Cost
  - Fault tolerance
  - ...

Motivation

Motivation:

- Distributed real-time systems need a **global time base**
- Requirements depend on the application
  - Precision
  - Cost
  - Fault tolerance
  - …

- IEEE 1588 specifies the **Precision Time Protocol (PTP)**

Motivation

- PTP provides a **large feature set**
  - Design space exploration is challenging

Motivation

- PTP provides a **large feature set**
  - Design space exploration is challenging
- Experimenting with real hardware is **expensive**
  - Prohibitive for experiments with larger networks

Motivation

- PTP provides a **large feature set**
  - Design space exploration is challenging
- Experimenting with real hardware is **expensive**
  - Prohibitive for experiments with larger networks

$\Rightarrow$ Use **simulation** to avoid costs and provide flexibility

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

Problem statement

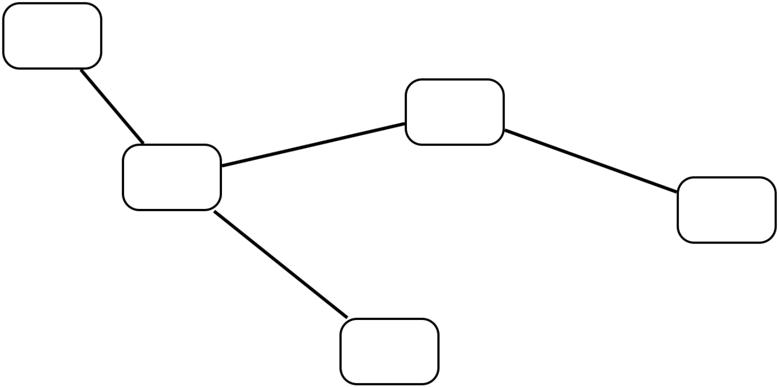
- **Simulation goal**
  Provide a tool for PTP design space exploration

Problem statement

- **Simulation goal**
  Provide a tool for PTP design space exploration
- Requirements:
  - simple
  - efficient
  - realistic

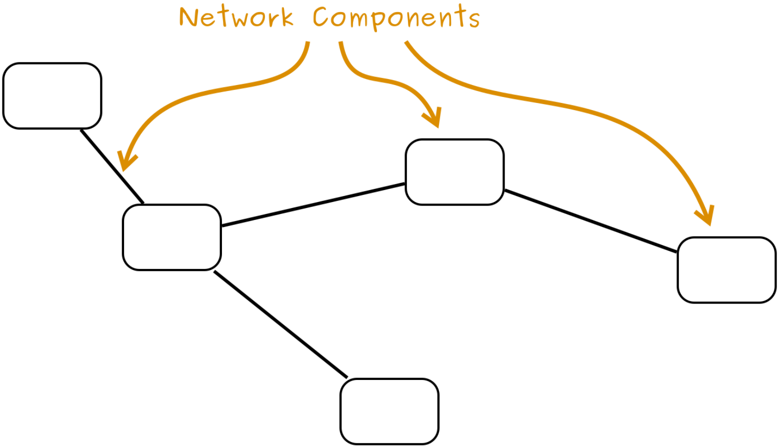
Problem statement

Simulation components:



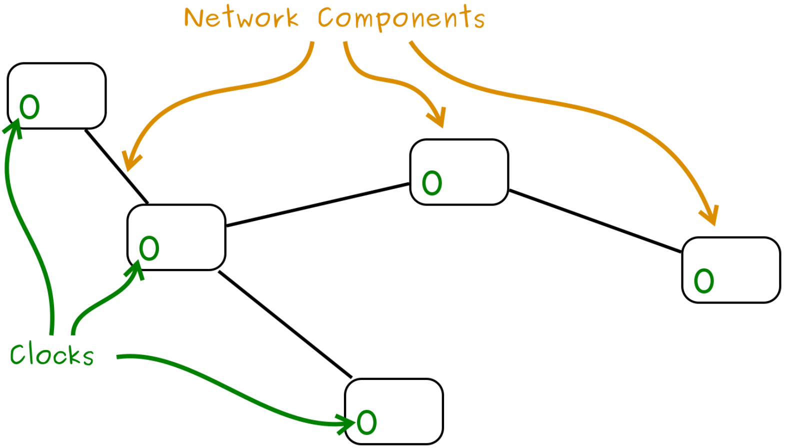
Problem statement

Simulation components:



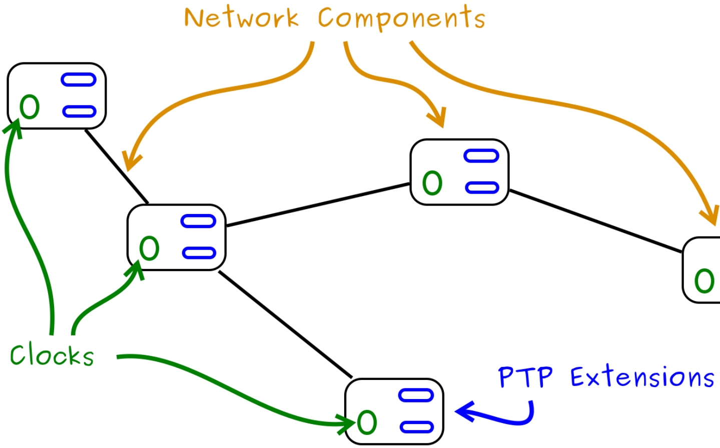
Problem statement

Simulation components:



Problem statement

Simulation components:



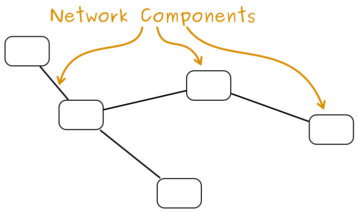
Simulation Components: Network

Network components

- Prior work suggets usage of

  OMNeT++

- **INET** library provides common network components

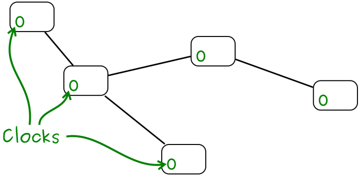


Simulation Components: Clocks

Clocks

Various noise sources:

- Random noise
- Deterministic influences
  - Environment, Aging, Drift, ...

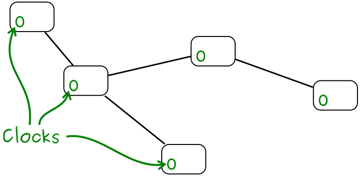


Simulation Components: Clocks

Clocks

Various noise sources:

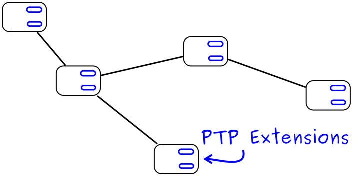
- **Random noise**
- ~~Deterministic influences~~
  - ~~Environment, Aging, Drift, ...~~



Simulation Components: PTP

PTP Components

- PTP Hardware
  - Timestamping NICs, . . .
- Software Components
  - PTP Stack
  - Clock Servo



Problem statement: Summary

Problem Statement (revised)

- Implement **PTP** in OMNeT++
- Provide realistic **clock noise**

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

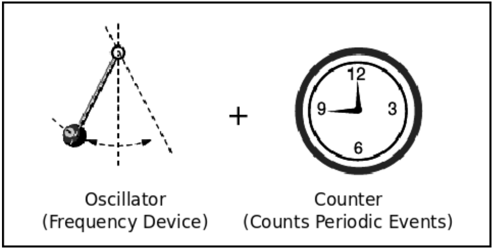
Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

Clock model I

The model of a **digital clock** consists of two parts:

- Oscillator
- Counter

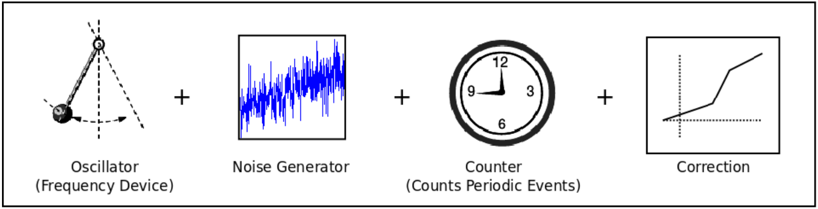


- Both parts may suffer from noise

Clock model II

Modified clock model:

- Oscillator and counter are **perfect**
- Additional components:
  - Noise generator
  - Correction stage



Basic definitions

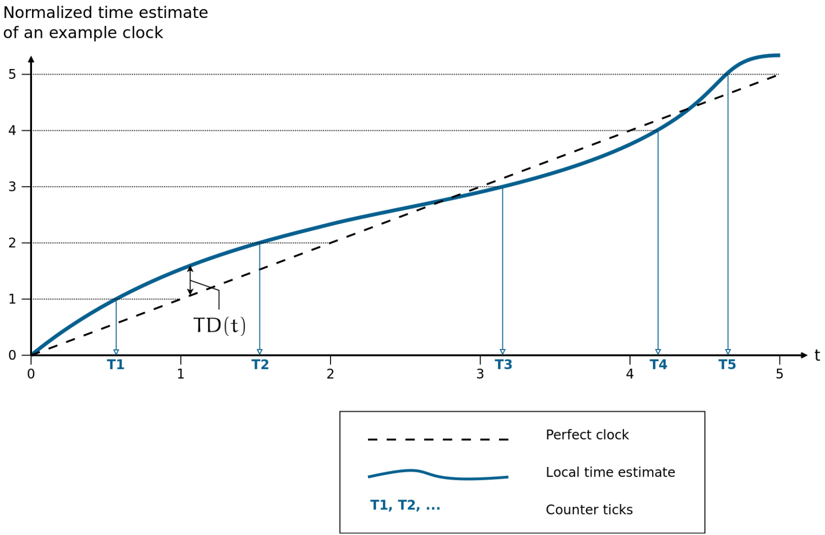
Frequency Stability Analysis
: Discipline of judging clock stability

Important attribute:

Time Deviation (TD)
: Instantaneous time departure from a nominal time

$\rightarrow$ *How wrong is this clock (now)?*

Visualization of Time Deviation



Combined Powerlaw Noise

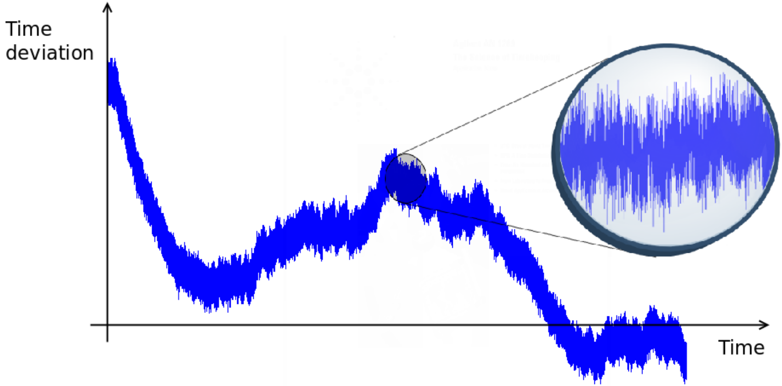


Figure: Combination of different PLNs

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

PTP - Intro

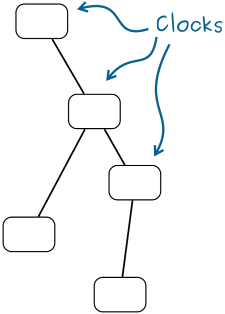
Precision Time Protocol

- Network based clock synchronization
- Compromise between cost and precision

PTP - Concepts

PTP Concepts

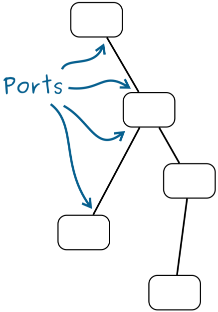
- Network nodes are called **clocks**



PTP - Concepts

PTP Concepts

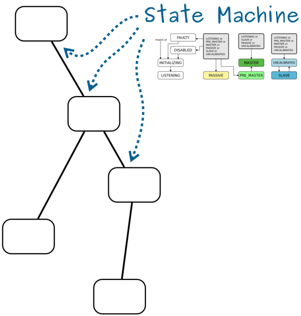
- Network nodes are called **clocks**
- Clocks have **ports**



PTP - Concepts

PTP Concepts

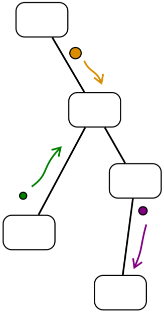
- Network nodes are called **clocks**
- Clocks have **ports**
- Ports have **states**



PTP - Concepts

PTP Concepts

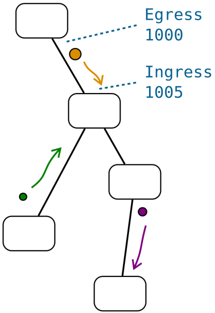
- Network nodes are called **clocks**
- Clocks have **ports**
- Ports have **states**
- Ports communitate via messages



PTP - Concepts

PTP Concepts

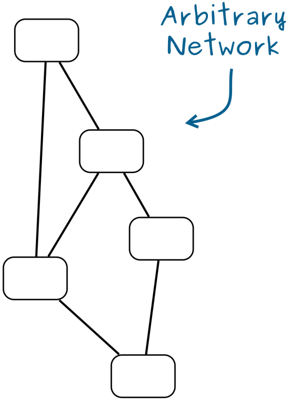
- Network nodes are called **clocks**
- Clocks have **ports**
- Ports have **states**
- Ports communitate via messages
- Nodes can
  - **timestamp** messages
  - **scale** their local clock



PTP - Services

PTP Services

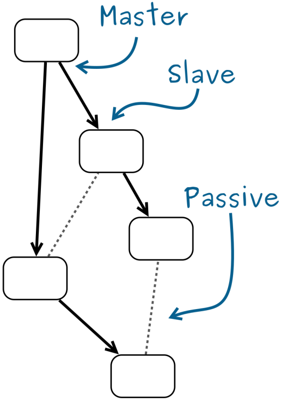
- Clock hierarchy
- Offset estimation
- Configuration



PTP - Services

PTP Services

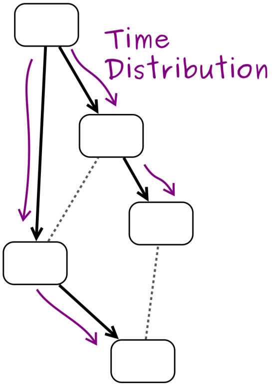
- **Clock hierarchy**
  - Best Master Clock algorithm
  - Root is called **grand master**
- Offset estimation
- Configuration



PTP - Services

PTP Services

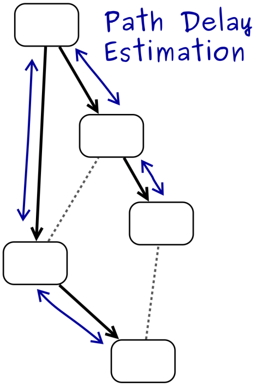
- Clock hierarchy
  - Best Master Clock algorithm
  - Root is called **grand master**
- Offset estimation
  - **Timestamp broadcast**
  - Path Delay measurement
- Configuration



PTP - Services

PTP Services

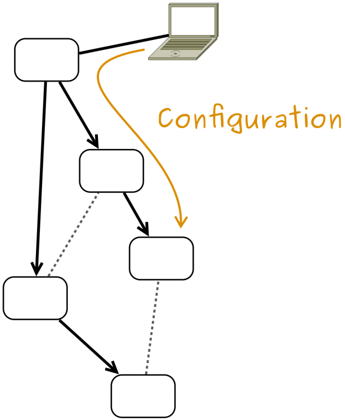
- Clock hierarchy
  - Best Master Clock algorithm
  - Root is called **grand master**
- Offset estimation
  - Timestamp broadcast
  - **Path Delay measurement**
- Configuration



PTP - Services

PTP Services

- Clock hierarchy
  - Best Master Clock algorithm
  - Root is called **grand master**
- Offset estimation
  - Timestamp broadcast
  - Path Delay measurement
- **Configuration**



Presentation Outline

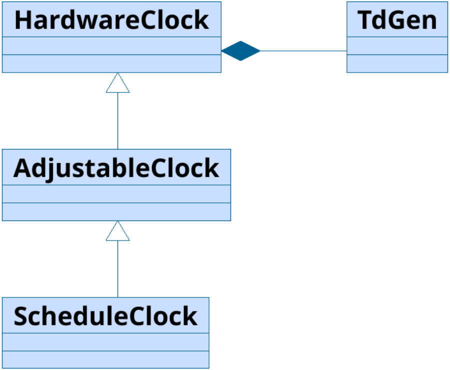
Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

Clock model for OMNeT++



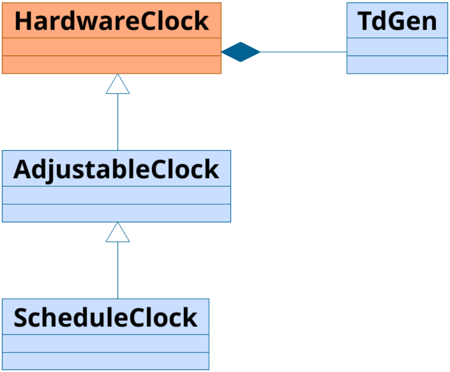
Clock model hierarchy

- Different modules for different tasks

Clock model for OMNeT++

HardwareClock

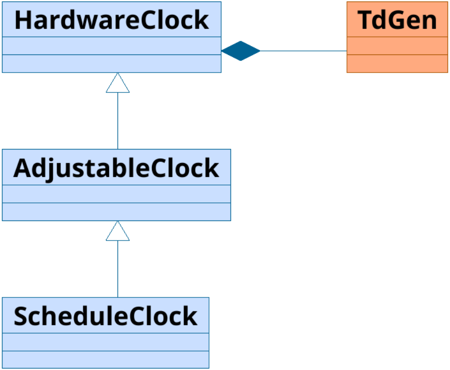
- Convert global real-time to locale estimate
  - **Real-time:** perfect, continous
  - **Estimate:** non-perfect, discrete



Clock model for OMNeT++

TdGen

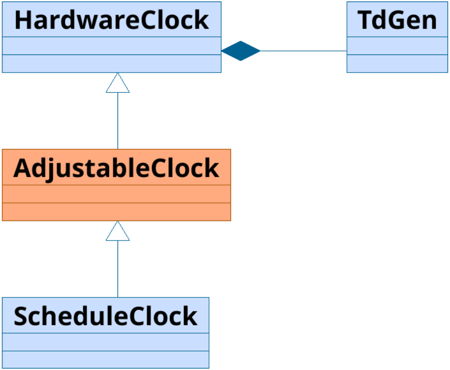
- Provide the Time Deviation (TD) for a given point in time
- Future TD values may be **estimates**



Clock model for OMNeT++

AdjustableClock

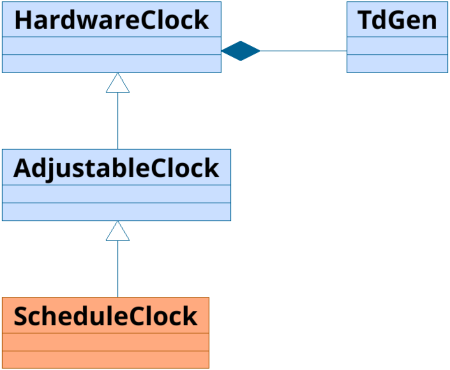
- Provide abstraction on top of `HardwareClock`
- Add **linear correction**



Clock model for OMNeT++

ScheduleClock

- Locale alternative to `scheduleAt()`
- Internal event queue



Presentation Outline

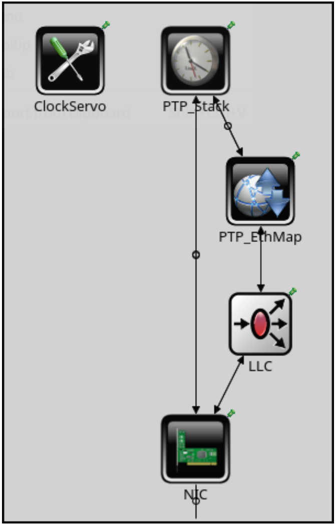
Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

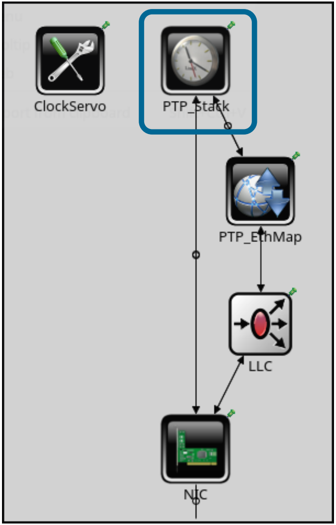
PTP Stack



Basic PTP node

- Architecture is based on `StandardHost` and `EthernetSwitch` from INET library

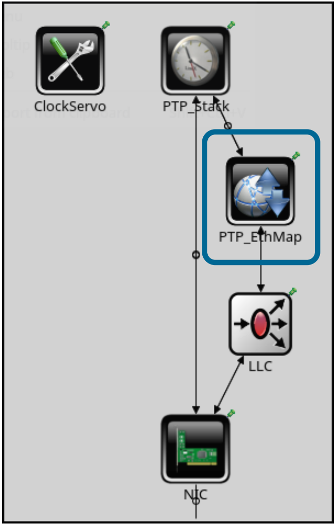
PTP Stack



PTP stack

- Implements core of IEEE 1588
  - Message types
  - BMC algorithm
  - Port states
  - Data sets
  - Clock types
  - Delay mechanisms
  - ...

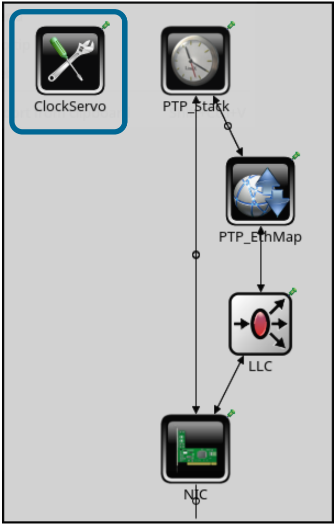
PTP Ethernet Mapping



PTP Ethernet Mapping

- Annex F of IEEE 1588
- PTP over Ethernet

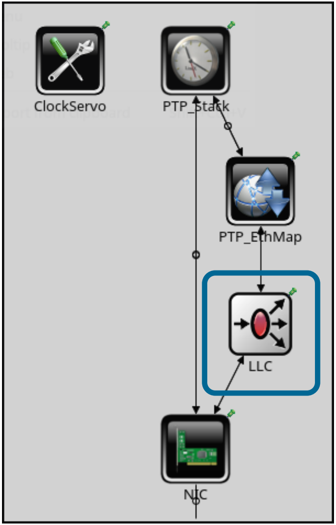
PTP Clock servo



Clock servo

- Generic interface
- 1 implementation:
  - PI controller

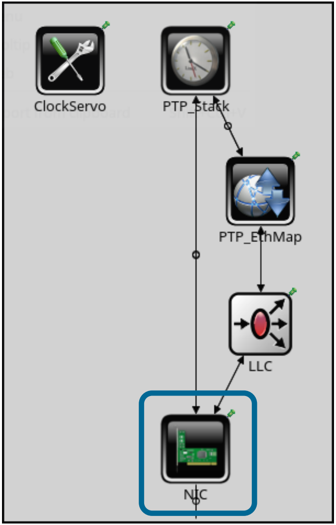
Logical Link Control



Logical Link Control

- Layer 2 access
- Move frames to correct application based on **EtherType**

PTP NIC



PTP NIC

- Clock
- PHY
- MAC

Presentation Outline

Introduction
- Motivation
- Problem statement

Basic concepts
- Clock Model
- Precision Time Protocol

Implementation
- Clock Model for OMNeT++
- PTP in OMNeT++

Conclusion

Project State

Current project state

- Code released as GPL
  - github.com/w-wallner
- Thesis finished
  - *Simulation of time-synchronized networks using IEEE 1588-2008* [7]
- Papers
  - ISPCS[1] 2016, Stockholm
  - OMNeT++ Community Summit 2016, Brno

[1]Conference with a focus on Precision Time Protocol (PTP)

Outlook

Future work

- PTP features
- Hardware properties
  - Deterministic clock influences
  - Switch models (queues)
- More clock servos

Conclusion

- Simulation approach **is feasible**
  - Clocks can be implemented efficiently
  - Assembling PTP networks is easy with Graphical User Interface (GUI)
- Simulation has already been useful for **teaching PTP**
- There is **strong interest** for such a simulation

Questions

- Do you have any questions?

Thanks

Thanks for your attention!

Acronyms I

AAS	Austrian Academy of Sciences
ADEV	Allan Deviation
AVAR	Allan Variance
BC	Boundary Clock
BMC	Best Master Clock
DES	Discrete Event Simulation
E2E	End-to-End
FFM	Flicker Frequency Modulation
FPM	Flicker Phase Modulation
FSA	Frequency Stability Analysis
GUI	Graphical User Interface
LLC	Logical Link Control
NIC	Network Interface Card
OC	Ordinary Clock
OMNeT++	Objective Modular Network Testbed in C++
P2P	Peer-to-Peer

Acronyms II

PI	proportional-integral
PLN	Powerlaw Noise
PSD	Power Spectral Density
PTP	Precision Time Protocol
RW	Random Walk
TC	Transparent Clock
TD	Time Deviation
WFM	White Frequency Modulation
WPM	White Phase Modulation

References I

- John C. Eidson
  Measurement, Control, and Communication Using IEEE 1588
  Springer, 2006
- Georg Gaderer, et al
  An Oscillator Model for High-Precision Synchronization Protocol Discrete Event Simulation
  Proceedings of the 39th Annual Precise Time and Time Interval Meeting, 2007
- IEEE Std 1139-2008
  IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology - Random Instabilities
  2009

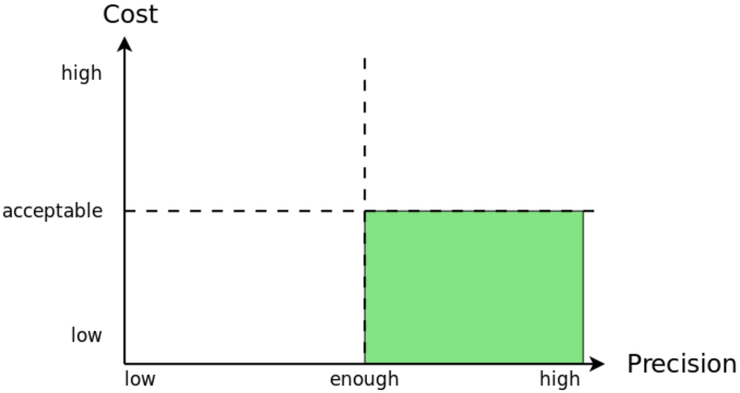
References II

- N. Jeremy Kasdin and Todd Walter
  Discrete Simulation of Power Law noise
  Proceedings of the 1992 IEEE Frequency Control Symposium, 1992
- William J. Riley
  Handbook of Frequency Stability Analysis
  NIST Special Publication 1065, 2008
- Enrico Rubiola
  The Leeson effect - Phase noise in quasilinear oscillators
  ArXiv Physics e-prints, 2005
- W. Wallner
  Simulation of Time-synchronized Networks using IEEE 1588-2008
  Master's thesis, Faculty of Informatics, Vienna University of Technology, 2016

Dropped Slides

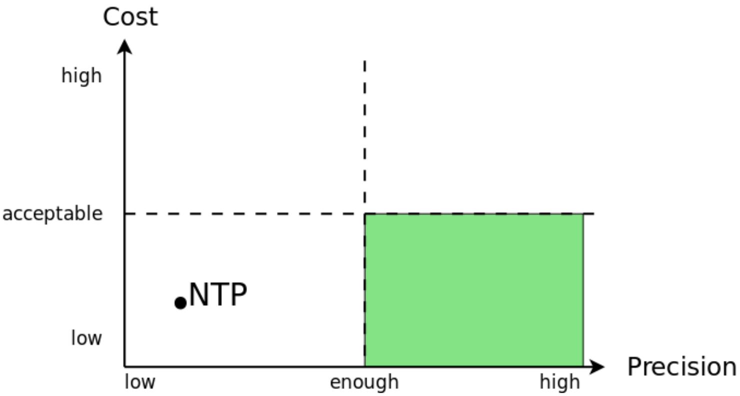
Motivation

PTP: Compromise between **cost** and **precision**



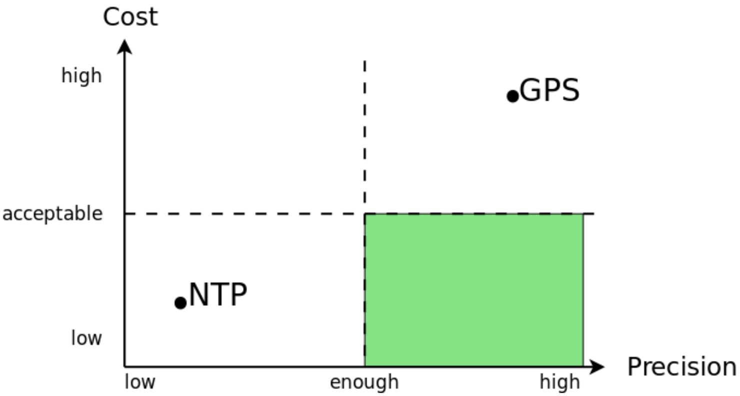
Motivation

PTP: Compromise between **cost** and **precision**



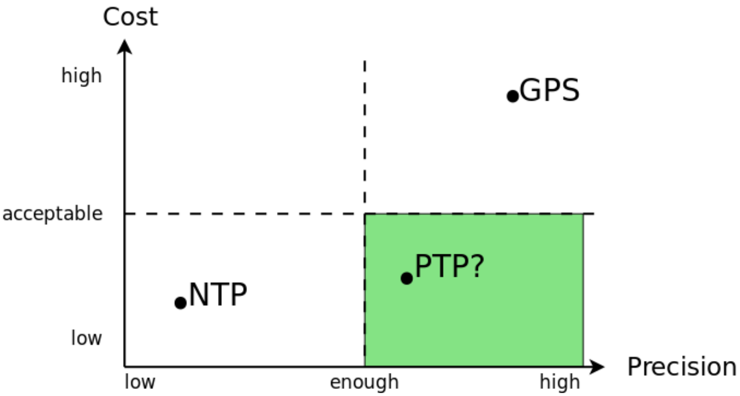
Motivation

PTP: Compromise between **cost** and **precision**



Motivation

PTP: Compromise between **cost** and **precision**



Problem statement

Reminder: Any simulation is just as good as its models.

Possible risks of simulation include:

- Too naive clock model $\rightarrow$ **false positives**
- Clumsy control loop $\rightarrow$ **false negatives**

Frequency Stability Analysis - Intro

- We need to justify the stability of clocks
- This discipline is called **Frequency Stability Analysis (FSA)**
- Literature:
  - *Handbook of Frequency Stability Analysis*[5]
  - *IEEE 1139* [3]
    (Standard definitions for random instabilities)

Clock Noise: Description

Two important measures for description of noise:

$S_y(f)$ Power Spectral Density (PSD)

One-sided PSD of $y(t)$

Useful for **frequency domain analysis**

Clock Noise: Description

Two important measures for description of noise:

$S_y(f)$ **Power Spectral Density (PSD)**

One-sided PSD of $y(t)$

Useful for **frequency domain analysis**

$\sigma_y^2(\tau)$ **Allan Variance (AVAR)**

Special variance to measure stabilty of clocks

Useful for **time domain analysis**

Powerlaw Noise I

Random noise in oscillators has a **special PSD shape**:

$$S_y(f) \propto f^{\alpha}$$

Definition

This is called **Powerlaw Noise (PLN)**

Powerlaw Noise II

Special cases for α:

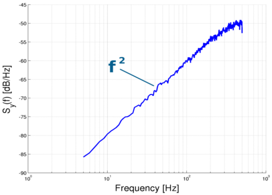
2	WPM	White Phase Modulation
1	FPM	Flicker Phase Modulation
0	WFM	White Frequency Modulation
-1	FFM	Flicker Frequency Modulation
-2	RW	Random Walk

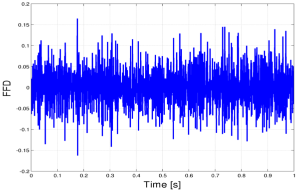
Powerlaw Noise examples

Power Spectral Density

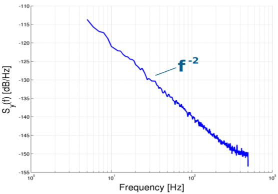
Fractional Frequency Deviation

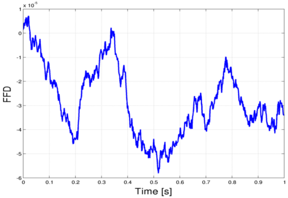
WPM





RW





Allan Variance

- Problem: standard variance does not converge

Allan Variance

- Problem: standard variance does not converge
- Alternative: **Allan Variance (AVAR)**

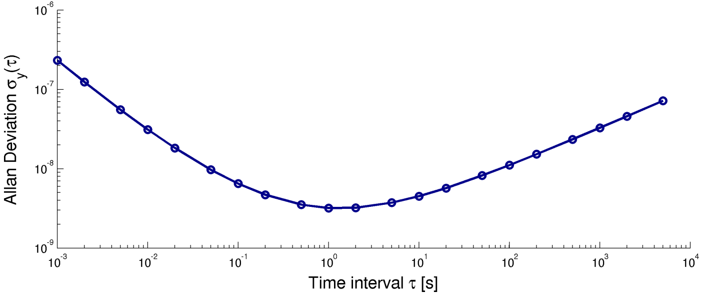
Allan Variance

Allan Variance

- Problem: standard variance does not converge
- Alternative: **Allan Variance (AVAR)**
- Equally widespread: **Allan Deviation (ADEV)**

Allan Variance

- Problem: standard variance does not converge
- Alternative: **Allan Variance (AVAR)**
- Equally widespread: **Allan Deviation (ADEV)**
- Example:



Relationship AVAR/PSD I

PLNs have characteristic AVAR:

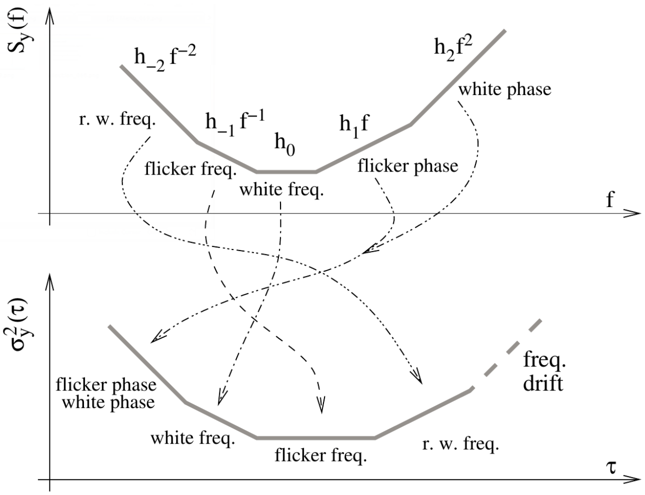


Image was taken from [6].

Relationship AVAR/PSD II

Table B.2 of IEEE 1139[3]:

PLN	$S_y(f)$	$\sigma_y^2(\tau)$
RW	$h_{-2} \cdot f^{-2}$	$A \cdot h_{-2} \cdot \tau^{1}$
FFM	$h_{-1} \cdot f^{-1}$	$B \cdot h_{-1} \cdot \tau^{0}$
WFM	$h_{0} \cdot f^{0}$	$C \cdot h_{0} \cdot \tau^{-1}$
FPM	$h_{1} \cdot f^{1}$	$D \cdot h_{1} \cdot \tau^{-2}$
WPM	$h_{2} \cdot f^{2}$	$E \cdot h_{2} \cdot \tau^{-2}$

- A, B and C are constants
- D and E depend on certain parameters

PTP - Timestamp modes

Different timestamp modes:

- 1-step clocks
- 2-step clocks

PTP - Timestamp modes

Different timestamp modes:

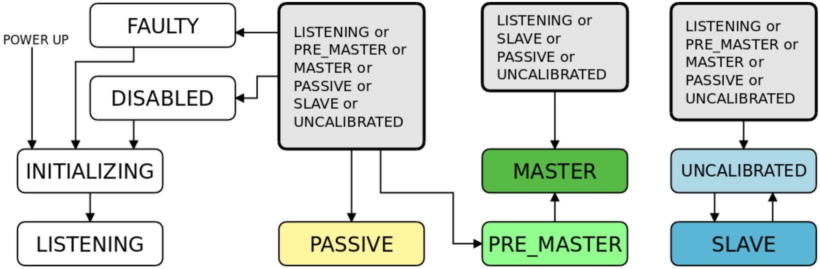
- 1-step clocks
  - Capable of timestamping outgoing frames **on-the-fly**
  - Needs explicit hardware support
- 2-step clocks

PTP - Timestamp modes

Different timestamp modes:

- 1-step clocks
  - Capable of timestamping outgoing frames **on-the-fly**
  - Needs explicit hardware support
- 2-step clocks
  - Not capable to timestamp on-the-fly
  - Use **`FollowUp`** messages

PTP - State machine



3 non-transient states:

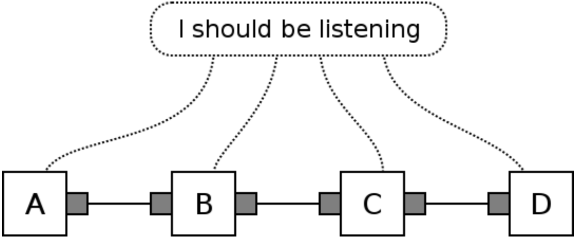
- MASTER
- SLAVE
- PASSIVE

PTP - Best Master Clock algorithm

- Clocks decide periodically about port states
- Next port state depends on
  - received `Announce` messages
  - timeouts
  - synchronization errors
  - ...
- Best Master Clock (BMC) is eventually consistent
- BMC results in a forest

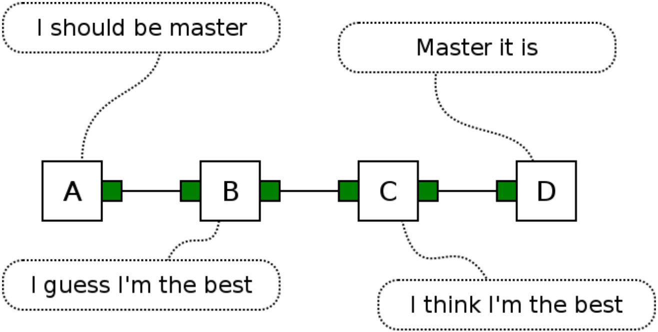
PTP - Simple BMC example I

- At first, all nodes start in `LISTENING`



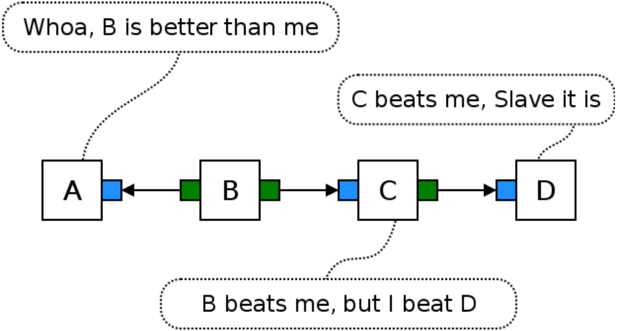
PTP - Simple BMC example II

- They see an idle PTP network, and try to become `MASTER`



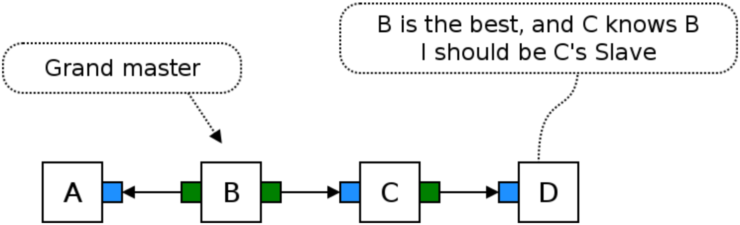
PTP - Simple BMC example III

- As the nodes start to see `Announce` messages, some ports change to `SLAVE`



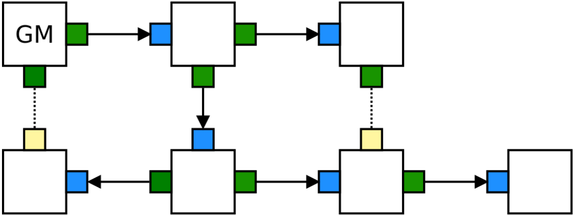
PTP - Simple BMC example IV

- Final hierarchy



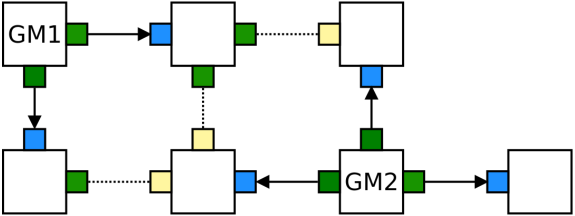
BMC example: rings

- Example network with 1 good clock
- Passive states break rings



BMC example: 2 excellent clocks

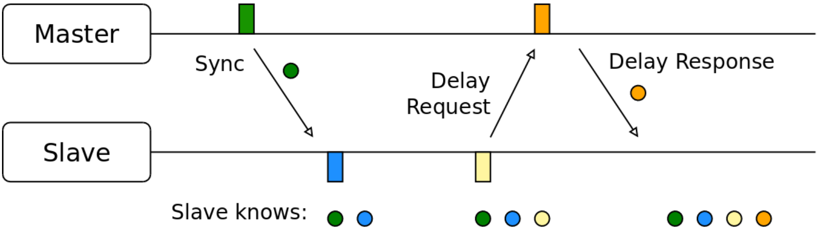
- Example network 2 excellent clocks
- Passive states divide network



PTP - Synchronization principle

Two tasks:

- Timestamp distribution
- Delay estimation



PTP - Clock types I

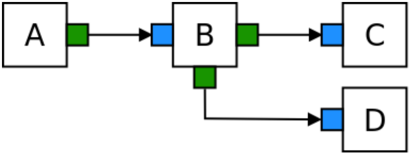
Ordinary Clock (OC)

- 1 port
- typical end node

PTP - Clock types II

Boundary Clock (BC)

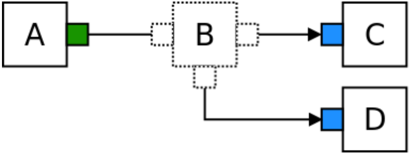
- multiple ports
- otherwise similar to OC



PTP - Clock types III

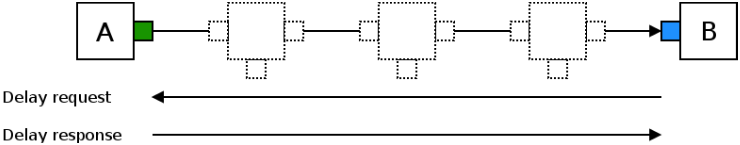
Transparent Clock (TC)

- multiple ports
- tries to not influence the PTP network
  - **residence time correction**
- introduced in IEEE 1588-2008

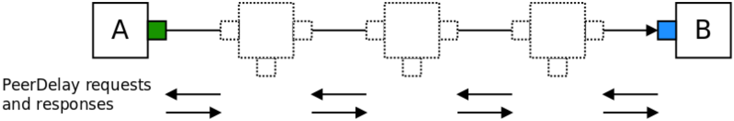


PTP - Delay mechanisms I

- End-to-End (E2E)



- Peer-to-Peer (P2P)



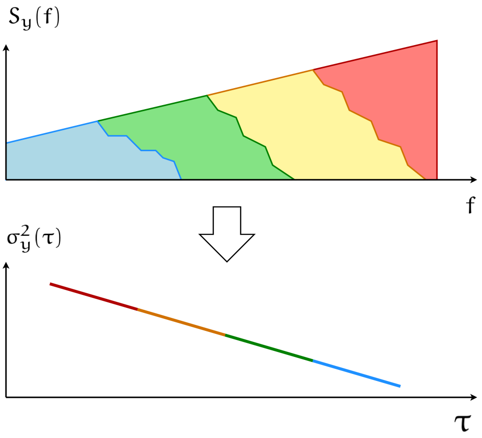
PTP - Delay mechanisms II

- E2E: Slave measures and corrects full distance
- P2P: Each nodes measures and corrects small part

- Advantages E2E:
  - Expected precision
- Advantages P2P:
  - Reduced overhead
  - Fast reaction on path change

PLN simulation - Combining PSDs I

- Combined PSD results in expected AVAR



PLN simulation - Prior work

Austrian Academy of Sciences (AAS)

Prior work:

- Was engaged in PTP and PLN simulation
- Several publications, e.g.
  - Gaderer, et al
    An Oscillator Model for High-Precision Synchronization Protocol Discrete Event Simulation, 2007[2]
- Served as **inspiration**

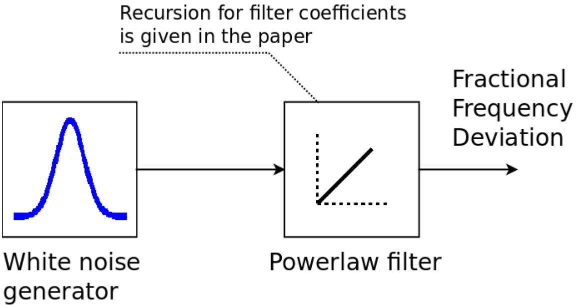
PLN simulation - Prior work

Prior work: **Kasdin/Walter Method**

- N. Jeremy Kasdin and Todd Walter,
  Discrete Simulation of Power Law noise, 1992[4]
- Generic method for PLN generation
- Basis for AAS papers
- Approach: **Filtering of white noise**

PLN simulation - Prior work

Kasdin/Walter approach: Filtering of white noise



PLN simulation - Challenges

- Problem **solved theoretically** by KW-approach
- **Too complex** for practical simulation purpose
  - Maximum simulation time is limited
  - Inefficient for Discrete Event Simulation (DES)

PLN simulation - Approaches

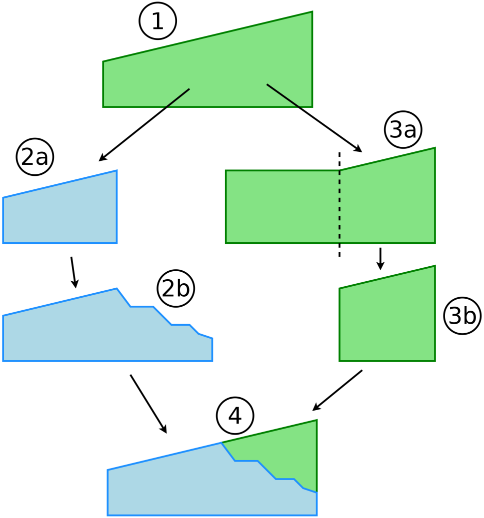
Maximum simulation time

- Combining PSDs with different f_s
- IIR filters for even α

Efficiency

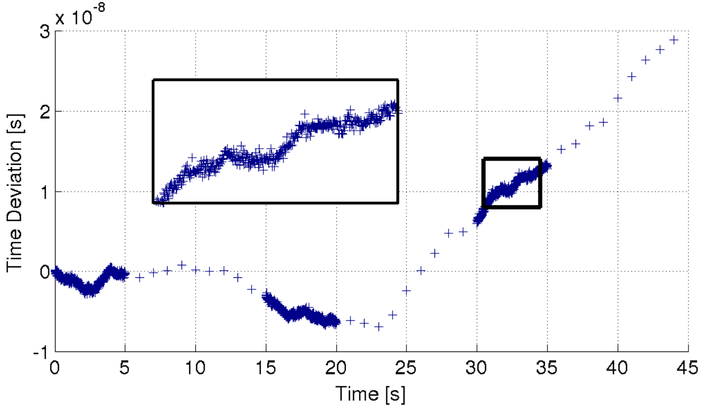
- Skip unneeded PSD contributions

Combining PSDs



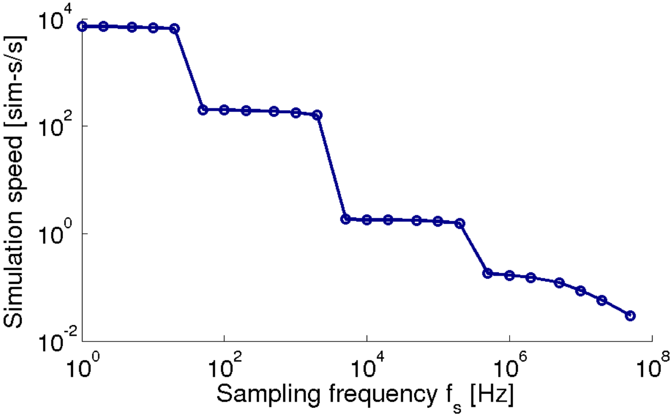
PLN simulation - Benchmark I

- Time Deviation at **different sampling rates**
  - Overall clock wander determined by Random Walk (RW)
  - High frequency noise is there when needed



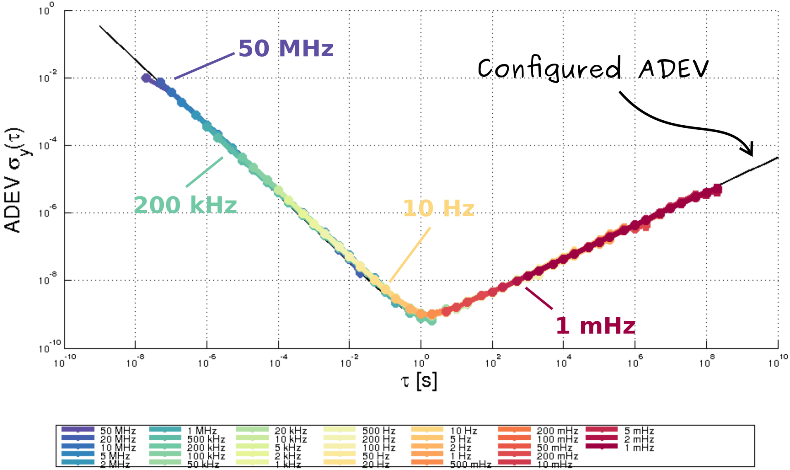
PLN simulation - Benchmark II

- Simulation speed $\propto 1/f_s$
- Results on my systems (Intel Core i7 2.00GHz):

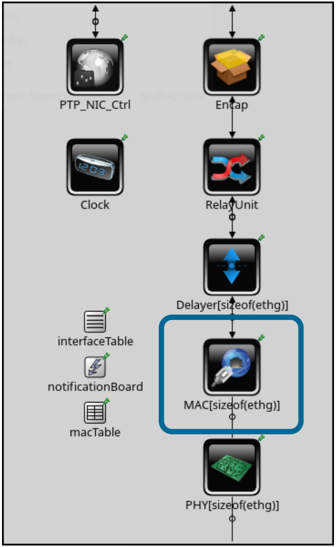


PLN simulation - Benchmark III

- Correct ADEV for **any sampling rate**



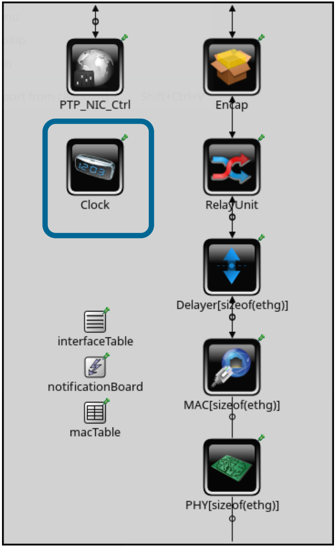
PTP NIC - MAC



PTP MAC

- **Timestamps**
  Event messages need ingress and egress timestamps
- **Residence time correction**
  When acting as a TC, the MAC must correct the residence time of outgoing frames

PTP NIC - Clock



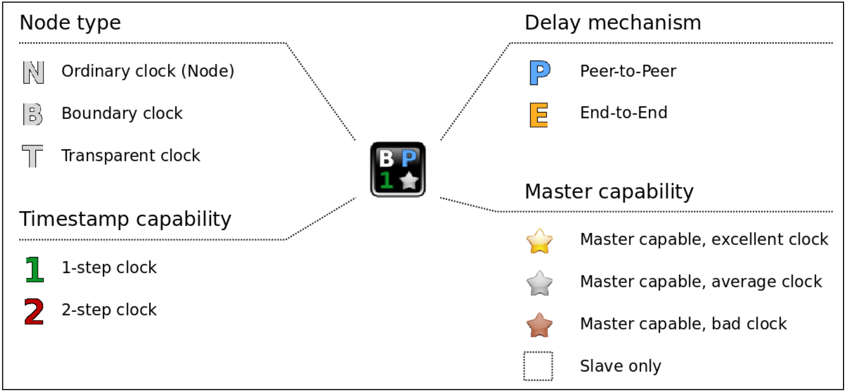
Clock

- **Timestamps**
  Used to timestamp events
- **Scalable**
  Controlled by Clock Servo
- **Event scheduling**
  PTP stack relies on it for timeouts

$\Rightarrow$ **Clock Noise**

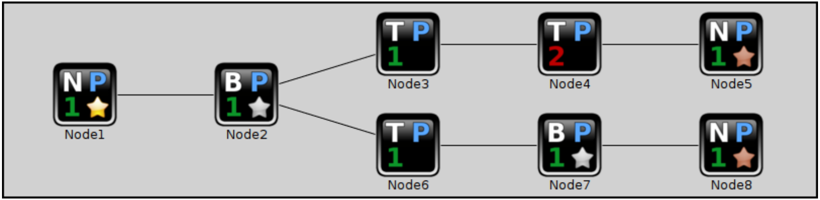
Node nodes

Node Symbols



Example network

Example network



Debugging and Logging

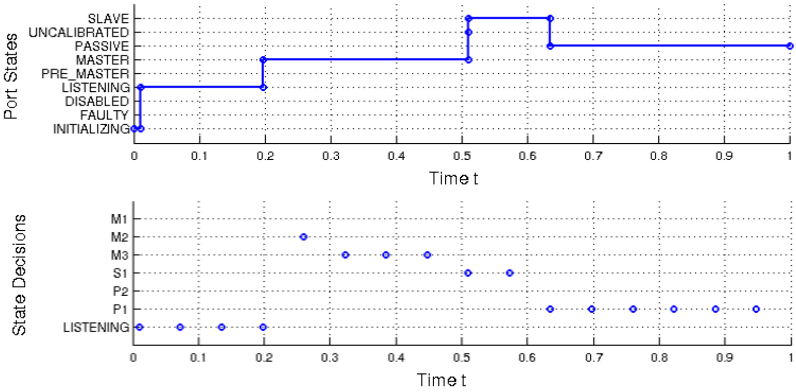


Figure: Port States and State Decisions

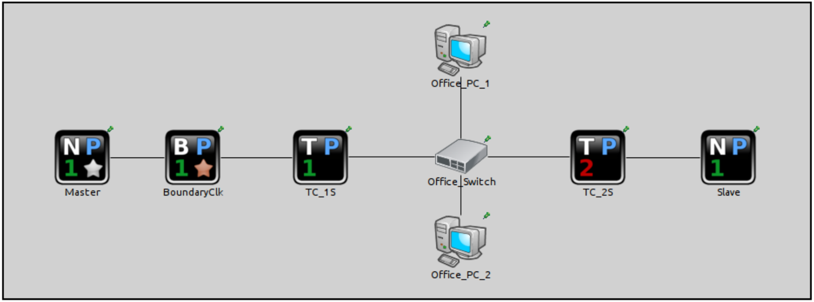
Message symbols

Message Symbols

- Announce
- Sync
- FollowUp
- DelayRequest
- DelayResponse
- PDelayRequest
- PDelayResponse
- PDelayResponseFU
- Management
- Signaling

Example network

Example network with PTP devices and standard office gear.



Experiment A1: Best Master Clock Algorithm

Experiment A1: Best Master Clock Algorithm

Best Master Clock Algorithm

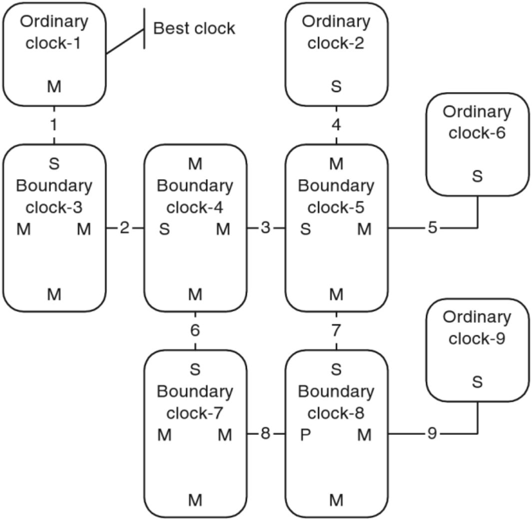


Figure: Example network from Eidson's book[1].

Best Master Clock Algorithm

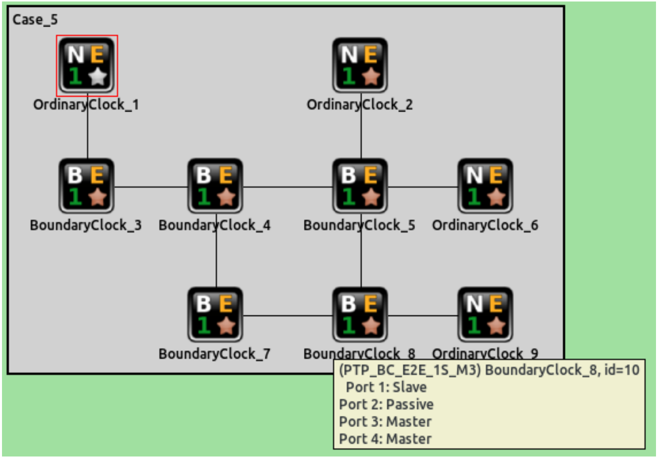
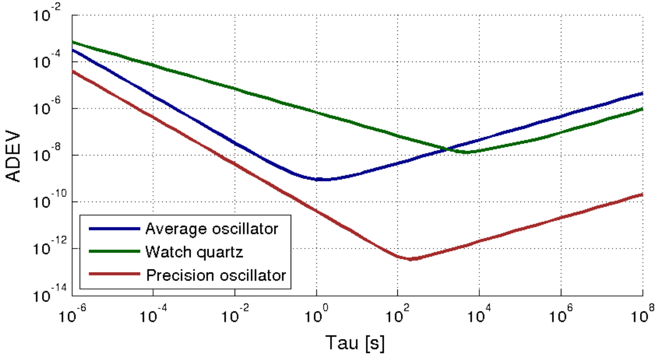


Figure: Simulation of the example network from Eidson's book.

Example Oscillators

- LibPLN implements 2 example oscillators

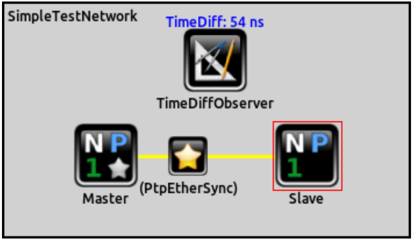


Experiment 1: Sync Interval

Experiment 1: Sync Interval

Simple Network

Simple test network



Parameter Study: Sync Interval

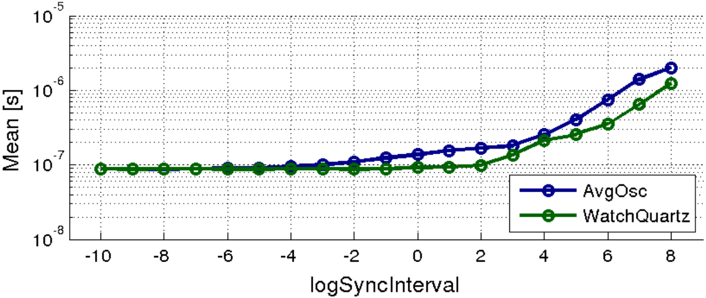


Figure: Mean value of the offset

Parameter Study: Sync Interval

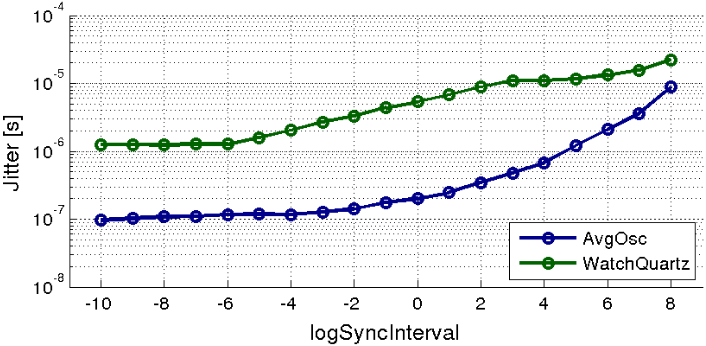


Figure: Jitter[2] of the offset

[2] $|Max - Min|$

Experiment: Path Asymmetry

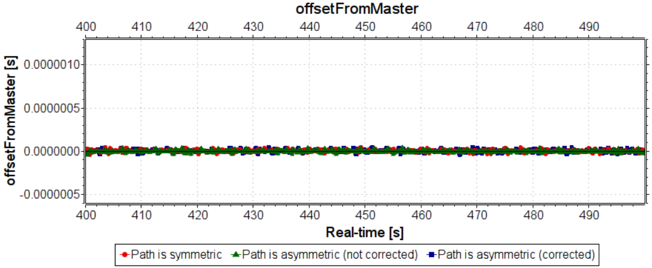
Experiment: Path Asymmetry

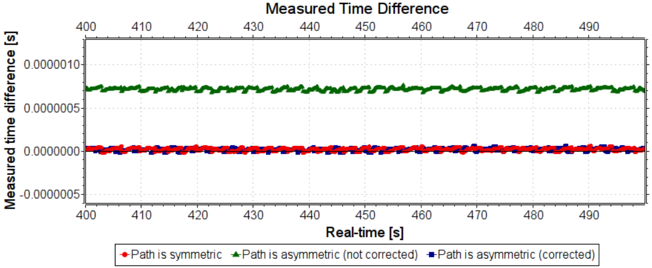
Asymmetry

Configuration

- Network with 2 PTP nodes
- 3 Configurations
  - No path asymmetry
  - Path asymmetry without correction
  - Path asymmetry with correction

Asymmetry


offsetFromMaster
offsetFromMaster [s]
0.0000010
0.0000005
0.0000000
-0.0000005
400
410
420
430
440
450
460
470
480
490
Real-time [s]
Path is symmetric
Path is asymmetric (not corrected)
Path is asymmetric (corrected)


Measured Time Difference
Measured time difference [s]
0.0000010
0.0000005
0.0000000
-0.0000005
400
410
420
430
440
450
460
470
480
490
Real-time [s]
Path is symmetric
Path is asymmetric (not corrected)
Path is asymmetric (corrected)