Simulation and emulation of real-time communication networks

Master Thesis

Franz Profelt

20.06.2016

General

Motivation

- Urge for testing of embedded systems
- Flexible testing scenarios
- Improved testing with Emulation and *HiL*

Tasks

- Fundamentals of OMNeT++, simulation, emulation, ...
- Design Evaluation
- Analysis of the openPOWERLINK stack
- Development of an OMNeT++ simulation representing a openPOWERLINK network

OMNeT++ Framework

General

- Object oriented modular discrete event network simulation
- Open Source simulation framework written in C++

Components

- Network, simple module, compound module
- channels
- messages, packets

Simulation types

- Real-time simulation
- Emulation
- Parallel simulation

Fundamental Designs

Monolithic

- Small number of modules
- Complex functionality within a single module
- Avoiding compound modules

Modular

- High number of modules
- Small functionality within a single module
- Combination of multiple modules to compound modules

Performance Measurement

Measurement Methods

runtime	Measurement of the runtime required to simulate a given amount of simulation time
created events	Measurement Measurement of the number of created events withing a fixed runtime
real-time	Observation of the real-time simulation indicator (performance ratio) during a parameter sweep for the data generation interval

Results

The average ratio of performance values using a modular design over a monolithic design.

Sequential

runtime	4.033
created events	3.506
real-time	1.592

Parallel

runtime	2.067

Conclusion

Using a monolithic design as sequential simulation is used for the openPOWERLINK simulation.

openPOWERLINK Stack

- Open Source implementation of POWERLINK.
  - Real-time communication protocol
- Distributed under the BSD license, available on GitHub and Sourceforge.
- Easy introduction in POWERLINK.
- Simple integration of POWERLINK in products.
- Improved development influenced by user requests and community contribution.

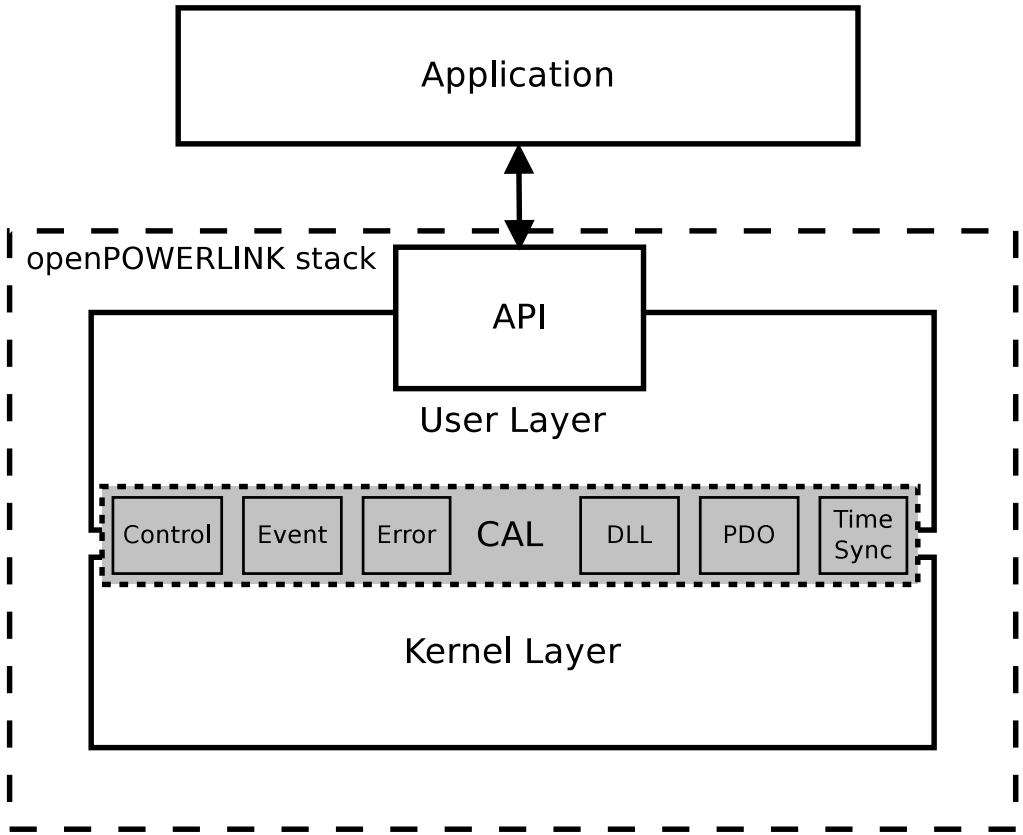
Structure

Architecture

User layer High level functionalities,API, Asynchronous transmission

Kernel layer Time critical functionalities, synchronization, drivers

CAL Connection between user layer and kernel Layer



Platform Dependency

Realized via common header files and platform specific implementations.

Implemented modules for minimal dependency

target	General target specific functionalities (Led, IP Address, Default Gateway, Tickcount, sleep)
edrv	Ethernet driver
hrestimer	High resolution timer
sdoudp	Service Data Object (*SDO*) transmission via UDP
trace	Trace output for debugging

Simulation Stub

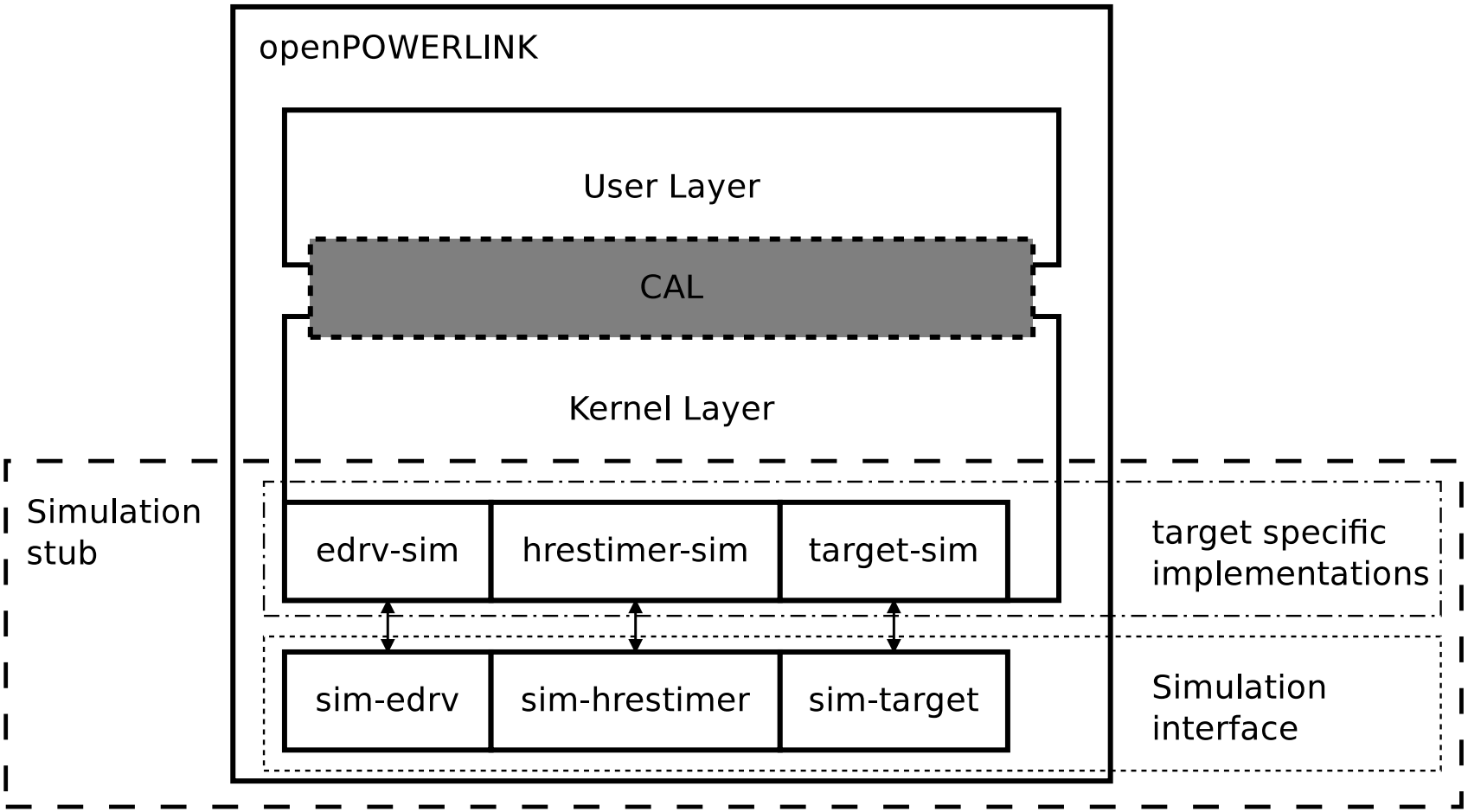
Target-specific implementation

- Implementation of all platform dependent modules for *sim* target
- Function forwarding to simulation interface
- Simple parameter conversions

Simulation interface

- Separate interface module for each platform dependent module
- Store function pointer to external simulation environment
- Calling of function pointer with stored instance handle

Simulation Stub



openPOWERLINK Simulation

- Separation of connection to openPOWERLINK stack from simulation modules.
- Reusability for different used stack configurations.
- Modularity for developing different Demo applications.

Multiple Instances

- Data and states stored in static variables
- Usage of openPOWERLINK library as shared library
- Multiple instances of shared library within memory
- Manual resolving and handling of different copies of shared library

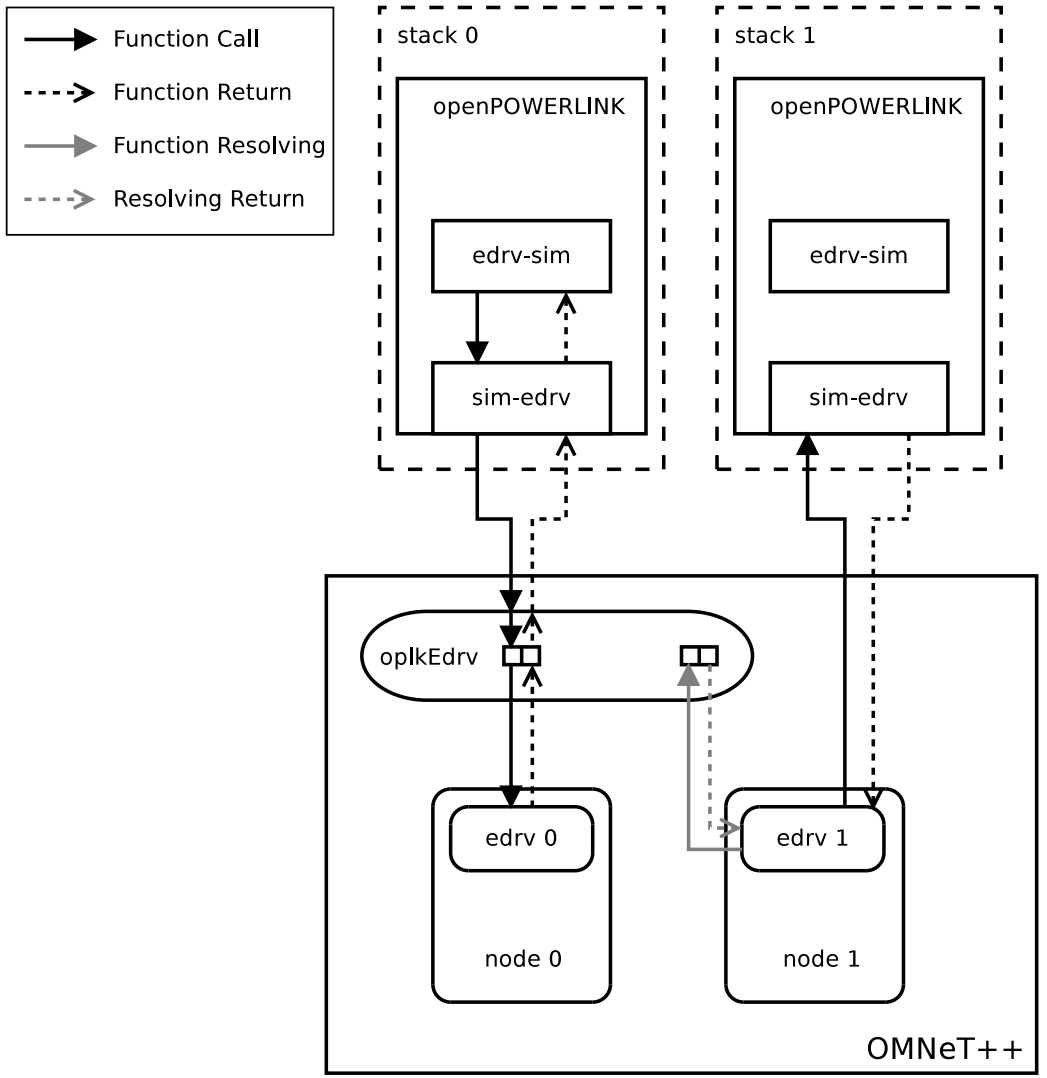
Simulation Hierarchy

Simulation interface

- Connection to stack
- Handling of multiple stack instances
- Static functions for function pointer

Simulation modules

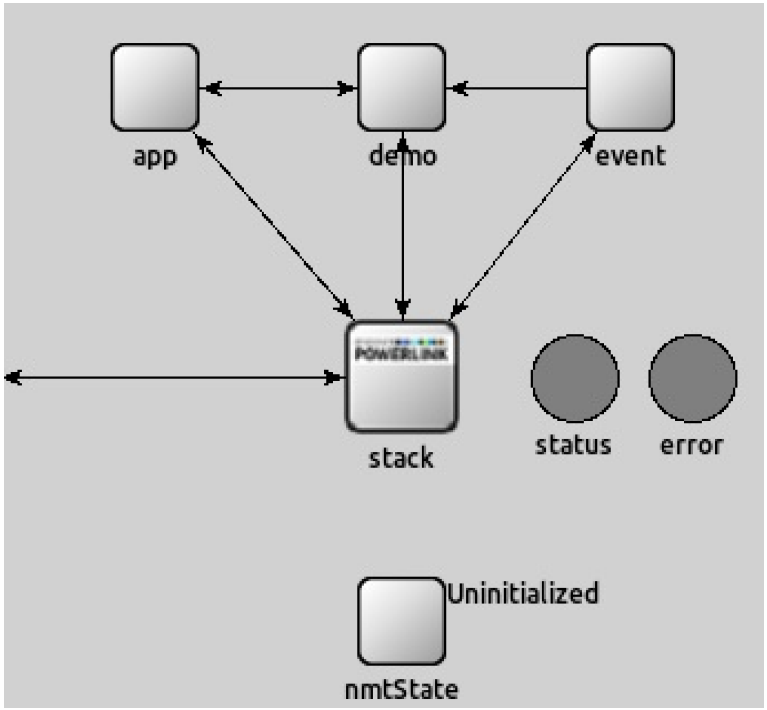
- Representation of stack structure
- Implementation of required functionalities



Generic node, MN, CN

Stack module

- Structure of simulated openPOWERLINK stack
- Basic functionalities for each node (MN, CN)
- Message handling in between modules
- Base classes for specific implementations



Further Development

Enhancements

- Usage of INET functionalities
- Implementation of different modularities
- Implementation of multiple demo networks/applications

Publication

- Integration of the simulation stub within the openPOWERLINK stack 2.5.0
- Hosting on GitHub
  `https://github.com/OpenAutomationTechnologies/openPOWERLINK_omnetpp`