# Porting Power Line Protocols from Hardware into ns-3 DCE

Andreas M. Lehmann
Chair of Information Transmission
Friedrich-Alexander-Universität Erlangen-Nürnberg
andreas.lehmann@fau.de

Jörg Deutschmann, Johannes Hampel
iAd GmbH
Unterschlauersbacher Hauptstr. 10
D-90613 Großhabersdorf, Germany
software@iad-de.com

Johannes B. Huber
Chair of Information Transmission
Friedrich-Alexander-Universität Erlangen-Nürnberg
huber@lnt.de

## ABSTRACT

Protocol development for large communication networks with many communication nodes requires reliable information about the networking behavior of the nodes. Usually network simulations support this development. We propose an approach to integrate the protocol stack of a power line communication (PLC) device into the network simulator 3 (ns-3) via its direct code execution (DCE) framework. The approach is not limited to PLC, but universal for different communication media and different protocol stacks. As an example we use the PLC channel and iAd GmbH's proprietary DLC®-3000 Single Frequency Network (SFN) PLC protocol stack. The machine code only needs minor adjustments to run in the DCE environment including the medium access layer and thus avoids remodeling all higher layers within the simulator. A proof of concept compares the log outputs of a hardware testbed with the ones from ns-3 DCE.

## 1. INTRODUCTION

The power grid of the future, also known as "smart grid", will heavily depend on communication systems. Narrowband power line communications and its different standards, such as PRIME [1], G3 [2], or DLC®-3000 SFN [3], are of special interest. In order to efficiently research and develop such protocols and verify interoperability, large networks with many communication nodes have to be investigated. For development of higher layers three major approaches exist, namely simulation, emulation and testbeds, further discussion of which can be found in [5], [6] and [7]. As all approaches have strengths and weaknesses of their own, usually a mixed approach is used. Simulations are based on adequate modeling. This requires simplifications and assumptions, but delivers control, scalability and reproducibility of results. Insights obtained by simulation are then translated into "machine code" which is tested and verified by emulation and/or testbeds.

To avoid the translation of model-based code from within the simulation into machine code that runs on dedicated hardware we use the direct code execution (DCE) framework, described in [6] and [7], to integrate existing machine code that is written in C/C++ into ns-3. This simplifies development and verification significantly as only one codebase for higher layers has to be maintained. Using iAd GmbH's proprietary DLC®-3000 Single Frequency Network (SFN) protocol we were already able to show that it is possible to integrate a power line protocol into DCE and ns-3 [4].

We outline the structure, which will enable us to integrate further protocols into the simulation environment. With this setup the advantages of simulation, namely scalability, reproducibility and debuggability are combined with the ones of emulation, i.e. verification of machine code and maintenance of only one code base.


*WNS3* 2015 Barcelona, SPAIN


## 2. FRAMEWORK DESCRIPTION

A network simulation framework needs to take into account a multitude of communication nodes that share the same communication medium, but of course not necessarily the same protocol stacks. In order to have comparable results it is important to model the channel influences (impairment, channel conditions, delay, etc.) independently of the protocols. Furthermore the impact of these influences is of course different for different protocols, which also has to be reflected in the modeling. Figure 1 shows the basic structure we are employing for our simulation environment. The common communication medium is modeled by the Universal Shared Medium Handler (USMH). In this module all impairments that are specific to the shared medium are modeled. The USMH also provides interfaces for sending and receiving packets as well as detection of packet collisions. In order for the USMH to determine signal propagation information on the topology is needed, which is not included in the picture here. As the ns-3 topology and propagation modules are not able to model all PLC specific channel conditions we employ one that is custom fit to our specific needs. With the topology dependency of the impairment and signal propagation every communication node requires a module that handles sending and receiving for it. We call this module specific physical layer handler (SPLH) , and for every protocol an own SPLH module has to be implemented that models for example the packet error ratio depending on the impairment and channel as well as interference situation and decides whether a packet reception was successful. Unlike the typical ns-3 NetDevice, the SPLH does not provide Layer 2 functionalities, which are in our case part of

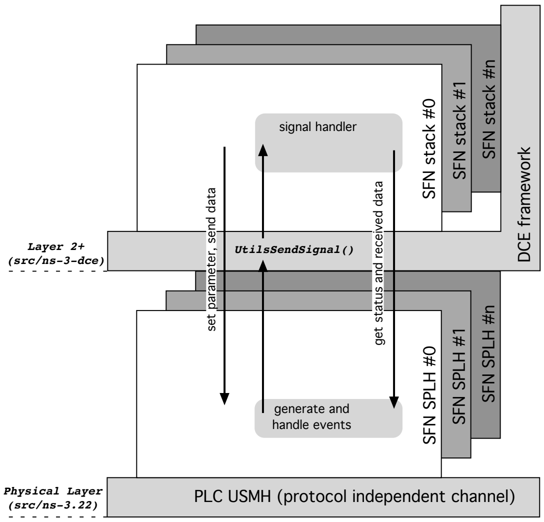


**Figure 1: Interface between physical layer simulation (ns-3) and SFN stack (DCE)**

the protocol stack running with its actual machine code in ns-3 DCE. The protocol for which the machine code integration has been carried out is iAd GmbH's DLC®-3000 SFN Protocol. The approach is universal and can be applied to many different communication protocols and shared mediums. Therefore we will not go into detail about the protocol at hand and refer to [3] for further information.

In order to integrate the low layer functionalities of the SFN protocol, the interfaces as shown in Figure 1 are used. The SFN stack running within the DCE framework is mapped to the SFN SPLH providing access to the physical layer simulation. The SFN SPLH forwards send requests from the SFN stack to the PLC channel and receive events in the opposite direction. Moreover it handles per-device physical layer parameters like frequency settings, which can affect the probability of successful transmission. In order to signal events from the simulation to a node running an SFN stack, we use the Linux signaling mechanism. This is similar to an interrupt handler on the hardware devices, which is invoked and then interacts with hardware, registers, memory, or other components of the operating system. In the simulation, a signal handler obtains all relevant information from the ns-3 driven SFN SPLH. A major advantage of this approach is that nearly all parts of the machine code including interrupt handlers are integrated into the simulation. With the described integration of the SFN stack into DCE, we have a well-defined interface between the ns-3 simulation representing the physical layer of a PLC system and higher layers running machine code within the DCE framework.

# 3. RESULTS

To confirm the proper integration of the SFN stack into DCE we provide a proof of concept testbed setup that is compared to the described ns-3 DCE approach. Furthermore we offer some insights on the performance of the simulation as well as exemplary results that can be obtained from the USMH tracer. Output that would not be as easily available in a testbed approach.

# 4. CONCLUSIONS

The proposed approach together with the ns-3 DCE framework offers a good opportunity to employ the same higher layer protocol stacks on hardware as well as within a network simulation environment. We proposed a structure for the integration of the physical, MAC, and higher layers into the framework and provide a proof of concept by integrating the existing machine code of the SFN protocol stack. Furthermore the additional simulator output as compared to testbeds enhances verification and debugging possibilities. Ns-3 DCE together with our suggested approach provides a flexible and scalable environment that also enables coexistence simulation between protocol stacks with the added value to provide reproducible results. The proposed structure is not limited to PLC protocols and the PLC channel, but can also be employed for other protocols or other communication media.

# 5. ACKNOWLEDGEMENTS

The authors would like to thank the iAd GmbH, Großhabersdorf, Germany, for the ongoing cooperation and support.

# 6. REFERENCES

[1] *Draft Specification for PoweRline Intelligent Metering Evolution*, Apr. 2014.

[2] *G3-PLC Open Standard for Smart Grid Implementation*, 2014.

[3] G. Bumiller. Influence of single frequency network transmission on the physical layer of a multi carrier modulation system. In *International Symposium on Power Line Communications and Its Applications*, pages 80–84, April 2005.

[4] J. Deutschmann, A. Lehmann, J. Hampel, and J. Huber. Network simulation for powerline protocols with direct code execution applied to dlc-3000 sfn. In *IEEE International Conference on Smart Grid Communications*, pages 464–468, Nov 2014.

[5] G. Kunz, O. Landsiedel, and G. Wittenburg. From simulations to deployments. In K. Wehrle, M. Günes, and J. Gross, editors, *Modeling and Tools for Network Simulation*, pages 83–97. Springer Berlin Heidelberg, 2010.

[6] M. Lacage. *Outils d'Expérimentation pour la Recherche en Réseaux (Experimentation Tools for Networking Research)*. PhD thesis, Université de Nice-Sophia Antipolis, 2010.

[7] H. Tazaki, F. Urbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous. Direct code execution: Revisiting library os architecture for reproducible network experiments. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pages 217–228, New York, NY, USA, 2013. ACM.