

Poster: Connecting Simulation and Real World: IEEE 802.11p in the Loop

Florian Klingler, Gurjashan Singh Pannu, Christoph Sommer, Falko Dressler
Heinz Nixdorf Institute and Dept. of Computer Science
Paderborn University
Paderborn, Germany
klingler@ccs-labs.org, pannu@ccs-labs.org, sommer@ccs-labs.org, dressler@ccs-labs.org

ABSTRACT

We present LAN Radio, an Open Source prototyping system to couple simulation and real world for the wireless channel of IEEE 802.11p based communication. Our main focus is the use in a vehicular networking environment, where field testing and real world experimentation is becoming more relevant these days. Large Field Operational Tests (FOTs) are expensive and difficult to handle in terms of reproducibility and controllability – particularly for application development and integration tests for automotive Electronic Control Units (ECUs). To support such experiments, we developed an approach to integrate a real system to test into a large scale simulation scenario without the need to change the physical and access layer parts of the communication stack of this system. We closely followed the Hardware In The Loop (HIL) simulation concept but also integrated a wireless communication channel into the picture. LAN Radio is building upon the well established and Open Source development platform OpenWRT to provide optimal extensibility.

ACM Reference Format:

Florian Klingler, Gurjashan Singh Pannu, Christoph Sommer, Falko Dressler. 2017. Poster: Connecting Simulation and Real World: IEEE 802.11p in the Loop. In *Proceedings of MobiCom '17, Snowbird, UT, USA, October 16–20, 2017*, 3 pages. <https://doi.org/10.1145/3117811.3131265>

1 INTRODUCTION

Research in the context of vehicular communication has evolved from pure theoretical studies to fully integrated prototypes ready for deployment on the road [2]. Many manufacturers have developed systems to allow inter-vehicle communication for safety and efficiency applications, however their operation is only evaluated in terms of simulation studies. In order to validate the correct behavior of such prototypes, realistic experiments are necessary, which can be scaled both in terms of abstraction detail and traffic density. One approach to such validation is to deploy a large set of vehicles in real world field tests [1], each equipped with prototype hardware for vehicular communication. This has two major downsides: first, the setup is very cost intensive; second, the environment is very difficult to control in terms of reproducibility. Another option

is to first record traces of real wireless traffic of communicating vehicles [5], and to then replay those traces on the system under test. However, most of the time this is no option, as other vehicles depend on the behavior of the target system, i.e., the surrounding traffic and the target system form a closed-loop control system.

HIL simulation can fill this gap between field tests and simulation by bidirectionally coupling the simulation framework together with the system under test [4, 6]. It has all the advantages of simulation studies (such as scalability, debug-ability, and reproducibility) and can integrate the system under test as a participant in the simulation study. This enables the system under test to participate in the network traffic available in the simulation, react to information received, and transmit own information to vehicles inside the simulation framework. Yet, current implementations of HIL simulations either have a special focus in mind, e.g., investigating lower layer characteristics of communication devices [4], or require specialized hardware with custom developed software like Software Defined Radios (SDRs) [3, 6]. Recently, investigations have been made to develop a framework for interoperability testing of ECUs used for cooperative ITS systems to allow over the Internet conformance tests [8]. However, a fundamental limitation is that the physical and access layers of the network stack need to be replaced at the system under test.

In this paper, we present *LAN Radio*,¹ a system which fills this gap by coupling the wireless channel modeled in a simulation framework with the wireless channel in the real world without the need to change the physical and access layers of the system under test. We therefore call it IEEE 802.11p in the Loop following the HIL terminology. The simulation and prototype hardware can be physically separated by a network, potentially even the Internet, simplifying field trials. The goal is to offer a flexible framework to allow prototypes in hardware to interact with nodes in a simulated environment using standardized IEEE 802.11p based protocol suites such as ETSI ITS-G5 and IEEE WAVE. Our implementation is using low cost hardware and will be made available as Open Source to the research community.

In brief, our main contributions can be summarized as follows.

- We introduce a new interface for realizing an IEEE 802.11p in the Loop system.
- We primarily focus on the higher layer protocols to enable investigation of application layer performance without the need to modify the system under test.
- Our system is easily extensible, based on commodity WLAN hardware, and freely available as Open Source.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MobiCom '17, October 16–20, 2017, Snowbird, UT, USA

© 2017 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4916-1/17/10.

<https://doi.org/10.1145/3117811.3131265>

¹<http://www.ccs-labs.org/software/>

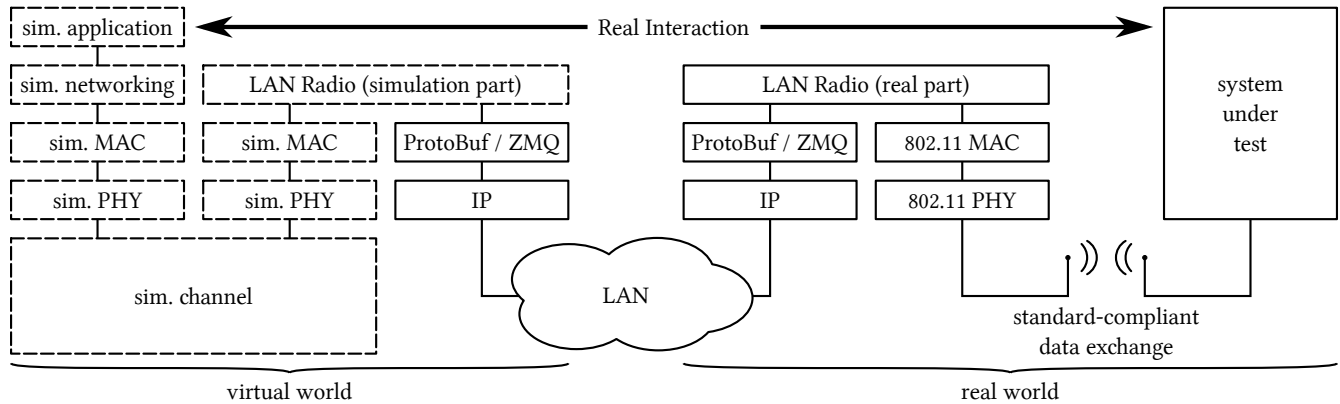


Figure 1: System model of LAN Radio

2 IEEE 802.11p IN THE LOOP

The system model of our LAN Radio system is outlined in Figure 1. The objective of our system is to allow applications running in a simulation environment to interact in real time with an application in the real world (e.g., running on an ECU equipped with radio communication) while providing a standard-compliant interface for data exchange to the *system under test*. Our system consists of two parts, namely (a) a model in the simulation framework to represent an *external* wireless network interface which connects to (b) a prototype in the real world to receive and transmit the information via an IEEE 802.11 compatible wireless interface. We want to highlight that the communication between the simulation and the real world is bidirectional, therefore, we can both receive frames from the real world in the simulation, as well as transmit information from the simulation to the real world.

2.1 Challenges

Combining the wireless channels of a virtual world (simulation) with the real world brings up two main challenges.

First, wireless signals exchanged inside the simulation have to be modeled to exhibit a realistic behavior by using adequate path loss models for signal propagation as well as realistic models for frame reception and decoding. The decoded messages then need to be sent to the real world side via our hardware prototype in order to be transmitted over the air using the standardized IEEE 802.11p communication protocol. We solve this first challenge by integrating LAN Radio into a discrete event simulation framework to model the wireless communication in a realistic way, and use encoding and decoding functionality to generate standards-compliant payload for frames transmitted via a dedicated hardware prototype.

Second, to transfer radio messages from the real world into the simulation, it is necessary to change the simulation state upon an external event. In our case, that means that whenever the hardware prototype sends a decoded frame from the real world to the simulator, the internal state of the simulation needs to be modified to process the frame. LAN Radio approaches this challenge by using a separate thread in the simulation model dedicated to listening for received messages from the hardware, and then interacting with the simulation kernel to update the state of the simulation.

2.2 Messaging

We rely on Google Protocol buffers (Protobuf) for serialization of messages sent between the simulation model and the hardware prototype. In particular, we include the standards-compliant encoded payload of the frame to be transmitted as well as meta information like source and destination MAC addresses in each Protobuf message. Further, we use ZeroMQ (ZMQ) as a messaging library to exchange the Protobuf encoded messages with the hardware.

The advantage of our approach is that all payload sent via the LAN Radio system is generated in the simulation framework as a MAC service data unit (MSDU). The generation of an MSDU within the simulation leads to full flexibility in protocol design, and allows easy testing of new applications, since the hardware prototype can be reused without modification. The MSDU, together with protocol control information like addressing information as well as optionally data rate and transmit power, is then transmitted via an IP based network to our prototype, which transmits the MSDU to the system under test. When receiving information from our system under test, the LAN Radio prototype decodes the frame and sends the information as a MAC SDU together with lower layer protocol information back to the LAN Radio model inside the simulation framework. This lower layer information includes addressing information as well as optional received signal strength and data rate information.

3 SIMULATION WORLD

For our implementation of LAN Radio on the simulation side, we selected the Veins vehicular network simulation framework [7]. Veins is based on the OMNeT++ discrete event simulator and provides realistic models for signal propagation and medium access according to the IEEE 802.11p standard, and evolved to a widely accepted toolkit in the vehicular networking research community. Moreover, Veins connects to SUMO to provide realistic modeling of road traffic. LAN Radio extends Veins by providing a module such that a node can connect its radio interface to be coupled with the real world. This way, all messages received by this node in the simulation world are forwarded to the hardware connecting to the real world prototype and transmitted over the air to the system under test. Further, the simulation model provides an interface to

configure the hardware prototype to the needs of the system under test such as the used carrier frequency and channel bandwidth.

In the following, we outline the operation of LAN Radio in detail using a simple example. In particular, we consider a scenario of some nodes inside a simulation, each within communication distance and running an example application generating beacon messages at periodic intervals (as used, for example, for cooperative awareness messages). All nodes are configured to use IEEE 802.11p in combination with realistic channel modeling as supported by the Veins framework.

The transmission of frames from the simulation to the real world is modeled as follows:

(1) Whenever a node generates a message, it is handed down to the simulated networking layer, and then further to the MAC for transmission. After competing for channel access, the frame is handed to the PHY module of all nodes within interference range to detect (but not necessarily decode) the frame.

(2) Any node that detects this frame tries to decode it, and if successful hands it up to the networking layer, and then further to the application. If this receiving node is configured to use our LAN Radio interface, a copy of this frame is handed to the LAN Radio module as well, which then encodes the message according to the protocol specifications of the respective message. The encoding is done using the corresponding ASN.1 packet encoding rules.

(3) Additional information such as higher layer message headers are prepended to the frame, such that it corresponds to a standards-compliant MSDU. This MSDU together with protocol control information, e.g., source and destination MAC Address and Ethertype, is then embedded inside a Protobuf message and sent via ZMQ to the hardware prototype for transmission.

The transmission of frames from the real world to the simulation is modeled as follows:

(1) A dedicated thread within the simulation model continuously listens for incoming messages from the hardware prototype. Whenever the hardware prototype receives an IEEE 802.11p frame, the MSDU is encoded inside a Protobuf/ZMQ message together with protocol control information and sent to our simulation model.

(2) Within the simulation model, the payload is decoded according to the ASN.1 specification. Then a message is generated following the standard procedure of the underlying network simulator. The corresponding frame is then inserted by the LAN Radio thread into a queue of frames to be processed by the simulation scheduler for transmission.

(3) Our model periodically checks for the presence of frames in this queue. If available, they are handed down to the PHY layer, which delivers it to all nodes within the simulation environment. Receiving nodes which are able to decode the frame then hand it up to the networking layer for further processing.

4 REAL WORLD

Our hardware prototype implementation builds upon the *LEDE* fork of *OpenWRT* for developing embedded systems based on Linux. OpenWRT allows to develop small firmware images for a variety of embedded systems (such as many cheap Wi-Fi routers) as well as commodity x86/x64 platforms. These firmware images contain the operating system adapted to unattended operation as well as

all necessary applications for simple deployment and maintenance. Images are usually no larger than a few MB.

At boot time, our application listens for configuration information from the simulation model via the Protobuf/ZMQ interface in order to correctly set up all wireless interfaces. Finally, one thread each is created to send and to receive IEEE 802.11p frames. Upon the successful reception of a frame from the wireless interface, it is encapsulated into a Protobuf message together with status information, e.g., addressing, and sent via Protobuf/ZMQ to the simulation model. When receiving a Protobuf message via ZMQ from the simulation, its MSDU is decapsulated and sent via the wireless interface to the system under test.

Based on the Outside the Context of a BSS (OCB) mode functionality of IEEE 802.11p our platform supports standard vehicular network on 5.9 GHz using 10 MHz channels.

5 CONCLUSION AND FUTURE WORK

We presented LAN Radio, a system to bidirectionally couple the wireless channel of a simulation framework with the wireless channel of the real world. Applications of such a system can be found in the validation of automotive ECUs for vehicular networking (or ECUs interacting with such). This way, the test of application behavior becomes better controllable in fully user customizable scenarios. Our system builds upon the OpenWRT framework and allows transmission and reception of data frames based on the IEEE 802.11p protocol as used in vehicular communication standards. In future work we will investigate the effects when integrating the system with a complete ETSI ITS G5 networking stack.

ACKNOWLEDGEMENTS

This work has been supported in part by the EU/EFRE.NRW Hy-Nets project.

REFERENCES

- [1] Bin Cheng, Ali Rostami, and Marco Gruteser. 2016. Experience: Accurate Simulation of Dense Scenarios with Hundreds of Vehicular Transmitters. In *22nd ACM International Conference on Mobile Computing and Networking (MobiCom 2016)*. ACM, New York, NY, 271–279. <https://doi.org/10.1145/2973750.2973779>
- [2] Falko Dressler, Hannes Hartenstein, Onur Altintas, and Ozan K. Tonguz. 2014. Inter-Vehicle Communication – Quo Vadis. *IEEE Communications Magazine* 52, 6 (June 2014), 170–177. <https://doi.org/10.1109/MCOM.2014.6829960>
- [3] Norman Franchi, Georg Fischer, and Robert Weigel. 2013. Radio Hardware In-the-Loop Emulation for Testing Vehicular Communication Systems. In *5th IEEE International Symposium on Wireless Vehicular Communications (WiVeC 2013)*. IEEE, Dresden, Germany. <https://doi.org/10.1109/wivec.2013.6698242>
- [4] Paul Fuxjaeger and Stefan Ruehrup. 2015. A Software-Defined Radio Testbed for Car-to-X Communication Devices and Protocols. In *3rd GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*. Ulm University, Ulm, Germany, 5–8.
- [5] Javier Gozalvez, Miguel Sepulcre, and Ramon Bauza. 2012. IEEE 802.11p Vehicle to Infrastructure Communications in Urban Environments. *IEEE Communications Magazine* 50, 5 (May 2012), 176–183. <https://doi.org/10.1109/MCOM.2012.6194400>
- [6] Matej Kloc, Robert Weigel, and Alexander Koelpin. 2016. Making Real-Time Hardware-in-the-Loop Testing of Automotive Electronic Control Units Wireless. In *International Conference on Advanced Technologies for Communications (ATC 2016)*. IEEE, Hanoi, Vietnam, 407–412. <https://doi.org/10.1109/ATC.2016.7764816>
- [7] Christoph Sommer, Reinhard German, and Falko Dressler. 2011. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing* 10, 1 (Jan. 2011), 3–15. <https://doi.org/10.1109/TMC.2010.133>
- [8] Alexey Voronov, Cristofer Englund, Hoai Hoang Bengtsson, Lei Chen, Jeroen Ploeg, Jan de Jongh, and Jacco van de Sluis. 2015. Interactive Test Tool for Interoperable C-ITS Development. In *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*. IEEE, Las Palmas, Spain, 1713–1718. <https://doi.org/10.1109/ITSC.2015.278>