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 Jonhg, 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